



HAL
open science

Development of new software tools for phase equilibria modelling of open systems

Matthew Mayne

► **To cite this version:**

Matthew Mayne. Development of new software tools for phase equilibria modelling of open systems. Earth Sciences. Université de Lyon, 2018. English. NNT : 2018LYSES038 . tel-02316944

HAL Id: tel-02316944

<https://theses.hal.science/tel-02316944>

Submitted on 15 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N°d'ordre NNT : 2018LYSES038

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'Université Jean Monnet, Saint Etienne

Ecole Doctorale N° accréditation 488
Sciences, Ingénierie, Santé

Spécialité / discipline de doctorat : Sciences de la terre, Pétrologie

Soutenue publiquement le 03/12/2018, par :
Matthew Jason Mayne

Titre de la thèse :
**Développement de nouveaux outils
informatiques permettant de modéliser
les équilibres de phases en système
ouvert**

Devant le jury composé de :

White, Richard
Tajcmanová, Lucie
Lanari, Pierre
Dziggel, Annika

Professeur, University of St Andrews
Professeur, Heidelberg University
Docteur, University of Bern
Docteur, Aachen University

Président du jury, rapporteur
Rapporteur
Examineur
Examineur

Stevens, Gary
Moyen, Jean-François

Professeur, Stellenbosch University
Professeur, Université Jean Monnet

Co-directeur.rice de thèse
Co-directeur.rice de thèse

ABSTRACT

The investigation of metamorphic processes in the Earth's crust is integral to understanding the formation and evolution of the Earth. These processes control the preservation potential of the geochronological rock record and give us insight into, amongst others, the pressure and temperature conditions of the Earth's interior. Further, they control fluid generation and consumption within the crust which influences global geochemical cycles within the lithosphere, hydrosphere and atmosphere. This has important implications on the global climate and the creation of conditions conducive to life. The dominant mechanism of change both within and between these systems are compositional changes invoked by processes of mass transfer. Modern quantitative phase equilibrium modelling allows the calculation of the stable phase assemblage of a rock system at equilibrium given its pressure, temperature and bulk chemical composition. However, current software programs have limited functionalities for the sophisticated handling of a changing bulk composition. A new software tool (Rcrust) has been developed that allows the modelling of points in pressure–temperature–bulk composition space in which bulk compositional changes can be passed between points as the system evolves. This new methodology enables quantitative process-oriented investigation of the evolution of rocks with a focus on mass transfer. Mass transfer within the crust generally occurs through partial melting and melt redistribution. These processes themselves are highly dependent on the chemical distribution of the crust in particular the water content of rocks undergoing metamorphism, thus the fluid state of rocks undergoing partial melting is investigated. It is found that metapelitic rocks begin melting at the fluid-saturated solidus regardless of their fluid state unless the subsolidus P - T path of the rock is sufficiently steep such that total water loss from the hydrous silicates of the rock cause the system to become less than fully-hydrated.

Further, as Earth's crustal differentiation proceeds largely by production and emplacement of granitic magmas (many of which are sourced from the melting of pelitic protoliths), a number of source controls are investigated for their relative effect on the bulk composition of melts produced by sequential melting and melt loss of a metapelitic starting bulk composition. It is found that the fluid state of the system is shown to have the strongest control on melt compositions with the pressure-temperature path having subordinate control on the volume and composition of melts produced.

This thesis presents key examples of the use of Rcrust to model the behaviour of open systems as well as the software developed in order to solve these problems and the graphical

user interface built to enable its distribution and reuse in the petrology community.

Keywords: Rcrust, metamorphism; mass transfer, crustal differentiation, phase equilibrium modelling

OPSOMMING

Metamorfisme is die sleutel dryfkrag van die mineraalverspreiding binne en stabilisering van die kontinentale kors deur middel van chemiese differensiasie. As sodanig ondersoekende metamorfe prosesse in die kors is integraal om die vorming en evolusie van die Aarde te verstaan. Hierdie prosesse beheer die bewaringspotensiaal van die geochronologiese rotsrekord en gee ons insig in onder andere die druk en temperatuurtoestande van die Aarde se binnekant. Verder beheer hulle vloeistofopwekking en verbruik binne die kors wat globale geochemiese siklusse in die litosfeer, hidrosfeer en atmosfeer beïnvloed. Dit het belangrike implikasies op die globale klimaat en die skep van lewensverwante omstandighede.

Die oorheersende meganisme van verandering binne en tussen hierdie stelsels is komposisionele veranderinge wat deur massaproesse toegepas word. Moderne kwantitatiewe fase-ewewigsmoedellering laat die berekening van die stabiele fase-samestelling van 'n rotsstelsel teen ewewig toe, gegee sy druk, temperatuur en grootmaat chemiese samestelling. Huidige sagteware programme het egter beperkte funksionaliteite vir die gesofistikeerde hantering van 'n veranderende grootmaat samestelling. 'n Nuwe sagteware-instrument (Rcrust) is ontwikkel wat die moedellering van punte in druk-temperatuur-massa-samestellingsruimte toelaat waarin grootmaat komposisionele veranderinge tussen punte verloop kan word as die stelsel ontwikkel. Hierdie nuwe metodologie stel kwantitatiewe proses-georiënteerde ondersoek in na die evolusie van rotse met die fokus op massa-oordrag. Massa-oordrag binne die kors kom gewoonlik voor deur middel van gedeeltelike smelt en smeltherverdeling. Hierdie prosesse self is hoogs afhanklik van die chemiese verspreiding van die kors, veral die waterinhoud van gesteentes wat metamorfose ondergaan. Die vloeistandstaat van rotse wat gedeeltelike smelting ondergaan, word dus ondersoek. Daar word bevind dat metapelitiese gesteentes begin smelt by die vloeistofversadigde solidus, ongeag hul vloeistand, tensy die subsolidus P–T-pad van die rots voldoende steil is, sodat totale waterverlies uit die waterige silikate van die rots veroorsaak dat die stelsel minder as ten volle gehidreer word.

Aangesien Aard se korsdifferensiasie hoofsaaklik deur produksie en inplanting van granietmagmas ontstaan (waarvan baie afkomstig is van die smelt van pelitiese protoliete), is 'n aantal bronkontroles ondersoek vir hul relatiewe effek op die grootmaat samestellings van smeltprodukte wat deur opeenvolgende smelting en smelt verlies van 'n metapeliet begin massa samestelling geproduseer word. Daar word bevind dat die vloeistoestand van die stelsel getoon word om die sterkste beheer op smeltkomposisies te hê, met die druk-

temperatuurpad ondergeskikte beheer op die volume en samestelling van smelt wat geproduseer word.

Hierdie proefskrif bied sleutel voorbeelde vir die gebruik van Rcrust om die gedrag van oop stelsels te modelleer, sowel as om die sagteware wat ontwikkel is om hierdie probleme op te los en die grafiese gebruikerskoppelvlak wat opgebou is om sy verspreiding en hergebruik in die petrolgemeenskap moontlik te maak, op te los.

Sleutelwoorde: Rcrust, metamorfose; massa-oordrag, korsdifferensiasie, fase-ewewigsmodellering

RÉSUMÉ

Le métamorphisme est un phénomène majeur affectant la distribution des phases minérales au sein de la croûte continentale et participant à sa stabilisation. L'étude des processus métamorphiques est donc essentielle pour comprendre la formation et l'évolution de la Terre. Ces processus exercent un contrôle sur le potentiel de préservation des roches à sa surface et nous renseignent entre autres sur les conditions de pression-température régnant en profondeur. Ils contrôlent également la production et le stockage de fluides au sein de la croûte ce qui influence les cycles géochimiques au sein de la lithosphère, de l'hydrosphère et de l'atmosphère et a, de fait, des implications importantes sur le climat et l'apparition de la vie sur Terre.

La principale source de variabilité au sein de ces systèmes correspond à des changements de composition chimique résultant eux-mêmes de transferts de matière. Les techniques modernes de modélisation quantitative des équilibres de phases permettent de calculer l'assemblage minéralogique stable au sein d'un système à l'équilibre pour lequel les paramètres pression, température et composition chimique sont connus. Ceci étant, les programmes informatiques actuels ne possèdent que de fonctionnalités limitées pour modéliser et appréhender les conséquences de changements de composition chimique du système au cours du métamorphisme. Un nouvel outil informatique (Rcrust) a été développé pour permettre de calculer l'assemblage minéralogique stable dans un système soumis à des variations de composition lors de son évolution dans l'espace multidimensionnel pression-température-composition chimique. Cette nouvelle approche méthodologique permet d'étudier de manière quantitative les processus métamorphiques et en particulier les conséquences des transferts de matière. Au sein de la croûte, ces derniers sont généralement associés à la fusion partielle des roches et à la migration des liquides magmatiques. Ces phénomènes sont tous deux fortement contrôlés par la composition chimique des roches crustales, en particulier leur teneur en eau. Ainsi, ce travail s'intéresse au rôle et à l'état des fluides présents dans les roches lors des phénomènes de fusion partielle. Il est ainsi montré que les métapélites commencent à fondre en franchissant leur solidus saturé en fluide, et ce, quel que soit leur état de saturation vis-à-vis de ce fluide. Une exception à ce comportement s'observe dans le cas où le trajet pression-température suivi par la roche comprend une importante décompression quasi-isotherme en conditions subsolidus. Ceci a pour conséquence d'entraîner une perte en eau du système qui présentera ainsi un assemblage minéralogique différent de celui observé lorsque la teneur en eau est maximisée au solidus

sans atteindre la saturation (conditions dites d'hydratation maximale).

La différenciation de la croûte continentale résulte de la production et migration de magmas granitiques dont une partie importante est issue de la fusion de protolithes pélitiques. Ainsi, les conséquences de variations compositionnelles au sein de telles sources sur la chimie des liquides produits par fusion partielle ont été appréhendées en modélisant une séquence d'évènements de fusion/extraction, reproduisant un comportement en système ouvert. Il est montré que l'état du système vis-à-vis des fluides est le paramètre qui exerce le plus fort contrôle sur la composition chimique des liquides produits.

Ce travail de thèse présente une série d'exemples illustrant l'utilisation de Rcrust pour modéliser les systèmes ouverts tout en précisant la méthodologie suivie pour l'élaboration de l'outil ainsi que l'interface graphique construite afin de permettre sa distribution et son utilisation au sein de la communauté pétrologique.

Mots-clés: Rcrust, métamorphisme, transfert de matière, différenciation crustale, modélisation des équilibres de phases

ACKNOWLEDGEMENTS

First off I would like to say a massive thank you to my supervisors Gary Stevens, Jeff Moyen and Tim Johnson. I have learnt so much by working with each of you and in this way you have shaped my abilities and future potentials as a scientist. The time you spent on me is much appreciated. Thank you also needs to go to all the incredible role models I have met in the petrological community, there are too many to mention here so I will just make a special mention to Chris Yakymchuck: for your meticulous reviews and extensive discussions; and Lars Kaislaniemi for your help with the wrapper and teaching me to compile.

Further special mention needs to go to my two abstract translators and all round life coaches: Simon Couzinié, thank you for our intense discussions on every rock face and all the crazy adventures we managed to squeeze in amongst those, thank you for welcoming me into your family and always jumping to help me at a moment's notice. Kirstin Cilliers thank you for being an absolute inspiration, your energy and enthusiasm for life is contagious, I wouldn't have gotten here without it.

I also would not have been able to get anywhere without the bureaucratic and administrative support of Tannie Loxie, Gillian Strydom, George Olivier, Emmanuelle Beretta, Narine Van Den Berg, Jannike Bergh and the staff at ResidHotel. Also thank you Wendy Knott-Craig from IOM3 for the incredible opportunities you afforded me and for the friendship we developed.

Thank you to everyone who opened up their lab (and often their couch) to me during my travels with special mention to Martin Hand, Renée Tamblyn, Kiara Alessio and Laura Morrissey at the University of Adelaide; Erin Martin, Kate Glasson, Michael Hartnady, Jo Moore, Eleanore Blereau and Denis Fougerouse at Curtin University; Jeroen van Hunen and Ian Chaplin at Durham University; Robyn Ormond at University of Johannesburg; Tawnee Britt at the University of Pretoria; Matthias Schmitz and Kathrin Schneider at Friedrich-Schiller-Universität Jena. Thank you all for going above and beyond.

Un gros bisous to the people of the LMV: Adrien Vezinet for all the climbing, running, camping, via ferrata, stupeflip and so much more, you made my stay an adventure; to Bogdana Radu for letting me sleep on your couch and constantly expanding my music horizons; to Gautier Nicoli for our in depth discussions and "beers till the bull"; to Cynthia Sanchez-Garrido for your constant patience; to Cyril Chelle-Michou for introducing me to the best galette in town; to Oscar Laurent and Arnaud Villaros you are the best braaing

Frenchmen I have ever met; to Lucile Dessimoulie and Théo Mouhrier for sharing those “times well wasted” and forest adventures; to Damien Fourure for instantly including me in everything; to Delphine Smittarello for the games of pool and Antonin Laurent for the hikes; to Maud Herbelin and Ratiba Kared for so willingly opening your office to me; to Michael Mintrone for translating the Zodiac knight; to Camille Dusséaux for Montpellier ramblings; to Capucine Albert for our random meetings and finally to the honorary LMV man Lucas Cassini thank you for your incredible company, for every time you joined me running up that hill and every time we visited Lipopette.

To the South African lab old and new thank you: to Valby van Schijndel and Jeanne Taylor for hiring me in your adventures; to Marcos García Arias for teaching me the way of Perple_X; to Shawn Kitt, Tomas Hedin and Duncan Hall for including me in the STDM experience and the tennis games; to Andrew Watson for the food lovers burgers and Johnny Gold, to Lunga Bam for the imperative advice on family; to Sara Burness for helping me limp down Lion’s Head and your constant stream of infectious happiness; to Nonkuselo Madlakana for your daily motivations and that ferry crossing; to Tolene Kruger for those custard danishes; to Kathryn Cutts for the park runs; to Shane Doggart for the crazy geology talks, flat earth theorists and that pegmatite; to Tsitsi Emmanuel for our existential conversations; to Joshua Chisambi for being the neatest and nicest office neighbour; to Marelize Grobbelaar for the best hand delivered sandwich in town; to Raphael Gordilho for introducing me to the strangest Brazilian food; to Roxanne Soorajlal for including me in your work and braving Rcrust; to Moses Angombe for those after work beers and that orthophoto; to Lebogang Babedi for the late night lab discussions; to Alice Petronio for your music suggestions; to Schalk Walters for teaching me the meaning of kindness; to Robyn Symons for the belly dancing lessons and Louis Jonk for the magic lessons; to Perpetua Okoye for your constant motivations in checking up on me; and to Kirstyn Barrat for always showing me there is a reason to laugh. To the new lab: Ian Weir, Jared Van Rooyen, Jean Looock, Ryan Cloete, Ismael Kanguuchi; Charl Cilliers and Marisa Storm thank you for providing and continuing such a fun and welcoming work environment this place has always felt like home. And finally a special mention to the next wave of Co-tutellers to Helena Fest for those roadtrips, balcony beers and fusebal games; to Moritz Muhlberg for Canyon, always saying yes to every adventure and for leading the hike when I needed you to and to Tahnee Otto for immediately becoming one of us on arrival I wish you all the best of luck for the future and am confident you will succeed.

To my UCT family: Antónia de Carvalho, Simon Schorn; Lorena Tafur Martin Giger and

Tristan Stuck thank you for always making me feel welcome in the house of the “rival”. How do I even mention “the pack”: to Ben Whitehead, Anya Eilers and Sabrina Skeat you were with me through every late night NARGA session, every cold morning packing down tents and will always remember to not mention “front-left”. You supported me from the moment I chose who I wanted to be, and incredibly could see this moment coming before I knew it myself. You built me up every step of the way along our rapidly diverging lives and for that I am forever in your debt.

To Sabrina Kumschick an incredibly special thank you for your “unbiased” support and dedication. You calmed me when things were mad and psyched me up when I felt down. You inspire me every day and give me the strength to be more than I am.

Finally, to my incredible family who have suffered through it all my whole life, you were behind me in every step and unconditionally believed in me especially when you weren’t quite sure what I was doing: Corbin, Karen and Alan Mayne I will never be able to describe what your love means to me.

Then of course thank you to Stellenbosch University and my funder: The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF. Funding by the French Embassy of South Africa to M.J.Mayne and by the South African Research Chairs Initiative (SARChI) to G. Stevens is gratefully acknowledged. M.J. Mayne acknowledges support from the European Research Council (project MASE, ERC StG 279828 to J. van Hunen).

TABLE OF CONTENTS

| | Page |
|---|-------------|
| Declaration | ii |
| Abstract | iii |
| Opsomming | v |
| Résumé | vii |
| Acknowledgements | ix |
| Table of Contents | xii |
| List of Figures | xiii |
| List of Tables | xvi |
| List of Abbreviations | xvii |
| Chapter 1: Introduction | 1 |
| Chapter 2: Presentation of research paper 1: Performing process-oriented investigations involving mass transfer using Rcrust: a new phase equilibrium modelling tool | 15 |
| Chapter 3: Presentation of research paper 2: The trajectory of the P-T path controls the onset of melting in metasedimentary rocks | 38 |
| Chapter 4: Presentation of research paper 3: A phase equilibrium investigation of selected source controls on the composition of melt batches generated by sequential melting of an average metapelite | 59 |
| Chapter 5: Conclusion and future perspectives | 89 |
| Bibliography | 97 |
| Addendum A: Electronic copy of Rcrust software | A:1 |
| Addendum B: User manual for Rcrust | B:1-37 |
| Addendum C: Presentation of the programming code for Rcrust | C:1-83 |
| Addendum D: Supplementary data for Research Paper 3 | D:1 |

LIST OF FIGURES

| | Page |
|------------------|---|
| Chapter 1 | |
| Fig. 1 | Example of a petrogenetic grid taken from Spear et al. (1999) 2 |
| Fig. 2 | Example of a pressure-temperature isochemical phase diagram taken from Powell et al. (1998) 3 |
| Fig. 3 | Modelling of melt extraction using THERMOCALC taken from Yakymchuk and Brown (2014) 5 |
| Fig. 4 | Modelling of garnet fractionation using Theriak/Domino taken from De Capitani and Petrakakis (2010) 6 |
| Fig. 5 | Modelling of melt extraction using Perple_X taken from (Connolly, 2018) 6 |
| Fig. 6 | Old and new Graphical user interface built for Rcrust 11 |
| Chapter 2 | |
| Fig. 1 | Division of the bulk composition of the Full System into portions representing the reactive subsystem, isolated subsystem and extract subsystem respectively 20 |
| Fig. 2 | Rcrust modelling procedure 22 |
| Fig. 3 | Modelling of subsolidus water loss 24 |
| Fig. 4 | Modelling melt loss events 26 |
| Fig. 5 | Modelling entrainment of phases during melt loss 28 |

| | Page |
|---|-------------|
| Fig. 6 Modelling fractional crystallisation of cumulative extract subsystem | 30 |
| Fig. 7 Modelling limited availability of feldspar | 32 |
| Chapter 3 | |
| Fig. 1 <i>P–T</i> diagrams illustrating three modelled path arrays | 44 |
| Fig. 2 Phase abundance and water abundance in the reactive subsystem for fluid-saturated, fluid-absent and fluid restricted rocks | 48 |
| Fig. 3 Melt abundance and solidi in <i>P–T</i> space produced by different path arrays | 50 |
| Fig. 4 Details of the fluid absent system projected from the <i>P–T</i> origin | 52 |
| Chapter 4 | |
| Fig. 1 Starting bulk compositions produced by varying Mg# and Ca# | 67 |
| Fig. 2 Phase abundance and cumulative melt produced along <i>P–T–X</i> paths for fluid-saturated, fluid-absent and fluid restricted rocks | 69 |
| Fig. 3 Bulk compositions of natural data array of s-type granites and modelled melt extracts for the source controls: Mg#, Ca#, H ₂ O, <i>P–T</i> and Melt threshold expressed as wt.% oxide versus wt.% SiO ₂ | 72 |

| | Page |
|--|-------------|
| Fig. 4 Bulk compositions of natural data array of s-type granites and modelled melt extracts for the source controls: Mg#, Ca#, H ₂ O, <i>P-T</i> and Melt threshold expressed as MgO + FeO _t ; Mg#; A/CNK and Na/Ca all versus wt.% SiO ₂ | 74 |
| Fig. 5 Bulk compositions of natural data array of s-type granites and modelled melt extracts grouped by fluid state and temperature intervals | 78 |
| Chapter 5 | |
| Fig. 1 Comparison of isochemical phase diagrams and phase assemblage maps created in Rcrust | 92 |

LIST OF TABLES

| | Page |
|---|------|
| Chapter 1 | |
| Table. 1. Suitability of commonly used thermodynamic modelling programs for phase equilibrium modelling in open systems | 7 |
| Chapter 2 | |
| Table. 1. Starting bulk composition for calculations | 24 |
| Table. 2. Bulk composition of the cumulative extract subsystem formed by subsolidus water loss, supersolidus melt loss and entrainment | 28 |
| Chapter 3 | |
| Table. 1. Modelling modes for handling of bulk H ₂ O content | 46 |
| Table. 2. Starting bulk composition for calculations | 46 |
| Chapter 4 | |
| Table. 1. Initial bulk compositions used in the construction of path dependent <i>P-T-X</i> paths with magnesium number (<i>Mg#</i>) and calcium number (<i>Ca#</i>) altered where applicable. | 66 |
| Table. 2. Summary of the source controls considered and each of their 3 cases. | 66 |
| Chapter 5 | |
| Table. 1. Suitability of the software presented in this thesis for phase equilibrium modelling in open systems | 89 |

LIST OF ABBREVIATIONS

Minerals:

Abbreviations for rock forming minerals were taken from Whitney & Evans (2010) as:

| | |
|------------------|-----------------------------------|
| Bt | Biotite |
| Crd | Cordierite |
| Fsp | Feldspar (plagioclase and alkali) |
| Grt | Garnet |
| H ₂ O | Water |
| Ilm | Ilmenite |
| Ky | Kyanite |
| Opx | Orthopyroxene |
| Qz | Quartz |
| Sil | Sillimanite |
| Spl | Spinel |
| St | Staurolite |

Terminology:

| | |
|--------------------------------|---|
| * | Multiplication |
| / | Division |
| $\Delta P/\Delta T$ or dP/dT | Pressure versus temperature gradient |
| A/CNK | $Al_2O_3/(CaO+Na_2O+K_2O)$ by mol.% |
| aH ₂ O | Water activity |
| An% | Anorthite percentage of feldspar as $Ca/(Ca+Na)*100$ by mol.% |

| | |
|--------|--|
| Ca# | Calcium number of the bulk as $\text{Ca}/(\text{Ca} + \text{Na}) * 100$ by mol.% |
| cos | trigonometric function cosine |
| IS | Isolated subsystem |
| E-HPG | Eclogite-high-pressure granulite |
| ES | Extract subsystem |
| FS | Full system |
| G-UHT | Granulite-ultrahigh temperature |
| GNU | GNU's Not Unix |
| Al | Aluminium |
| Ca | Calcium |
| °C | Degrees Celsius |
| Fe | Iron |
| FeOt | Total Iron as $\text{FeO} + 0.89981 * \text{Fe}_2\text{O}_3$ by wt.% |
| GCDkit | Geochemical Data Toolkit |
| GUI | Graphical user interface |
| H | Hydrogen |
| IBH | Isobaric heating |
| ITD | Isothermal decompression |
| K | Potassium |
| kbar | kilobar |
| MAD | Metamorphic assemblage diagrams |
| MCT | Melt Connectivity Transition |
| Mg | Magnesium |

| | |
|--------------|---|
| Mg# | Magnesium number as $\text{Mg}/(\text{Fe} + \text{Mg}) * 100$ by mol.% |
| mol.% | molar percentage |
| Mn | Manganese |
| Na | Sodium |
| NCKFMASHTO | $\text{Na}_2\text{O}-\text{CaO}-\text{K}_2\text{O}-\text{FeO}-\text{MgO}-\text{Al}_2\text{O}_3-\text{SiO}_2-\text{H}_2\text{O}-\text{TiO}_2-\text{O}$ |
| O | Oxygen |
| ORCID | Open Researcher and Contributor ID |
| <i>P-T-X</i> | Pressure-Temperature-Bulk composition |
| pi | mathematical constant π |
| RS | Reactive subsystem |
| Si | Silicon |
| sin | trigonometric function sine |
| Ti | Titanium |
| vol.% | volume percentage |
| wt.% | weight percentage |
| x_i | an increment in the vector direction x |
| y_i | an increment in the vector direction y |

CHAPTER 1

INTRODUCTION

The formation of igneous and metamorphic rocks is fundamental to determining the chemistry of the Earth's crust. Processes of metamorphic devolatilisation, partial melting, magma emplacement, fractional crystallisation and hydrothermal alteration, amongst others, cause the redistribution of elements within the crust. Partial melting and melt migration into the upper crust, in particular, redistributes heat producing elements within the crust depleting the lower crust in these incompatible elements and enriching the upper crust. This in conjunction with the formation of the mantle lithospheric keel is fundamental to the stability of Earth's continents. These processes and their effects can extend to the Earth's surface through volcanic activity, hydrothermal venting and the tectonic exhumation of rocks leading to their exposure at the surface and weathering. Consequently, H₂O and CO₂ can be released to the atmosphere/hydrosphere or re-absorbed into the lithosphere by weathering, sedimentation and lithification. Thus lithospheric processes can directly affect the hydrosphere and atmosphere as evidenced by major disturbances in the global geochemical cycles (Friend, 1973; Amiotte Suchet and Probst, 1995). This has important implications for secular changes in the Earth's climate and the availability of nutrients to the biosphere.

Since the majority of crustal processes occurring at depth are inaccessible to direct observation, observations of these processes must come from the incomplete sample set of rocks brought to the surface through erosion, uplift or transport through volcanic plumbing systems. In order to extend this dataset, laboratory experiments are conducted which aim to reproduce the conditions of the Earth's interior and then expose either natural or artificially created samples to these conditions. Other types of experiments commonly functioning in simple systems or using individual minerals measure the range of physical and chemical parameters necessary to define the thermodynamic properties of minerals and fluids. This thermodynamic data can be used to infer the conditions of a natural rock's formation (Spear et al., 2016).

Building on this dataset by cross referencing geochemical studies of natural rocks and experimental studies allowed individual mineral reactions to be identified (Berman, 1991). Compilations of reactions in a simple chemical system were first displayed on petrogenetic grids (Albee, 1965)(e.g. Fig. 1). The compilation of internally consistent thermodynamic

datasets allowed the quantitative calculation of subsolidus phase equilibria (Helgeson et al., 1978; Powell and Holland, 1985, 1988; Gottschalk, 1997; Holland and Powell, 1998).

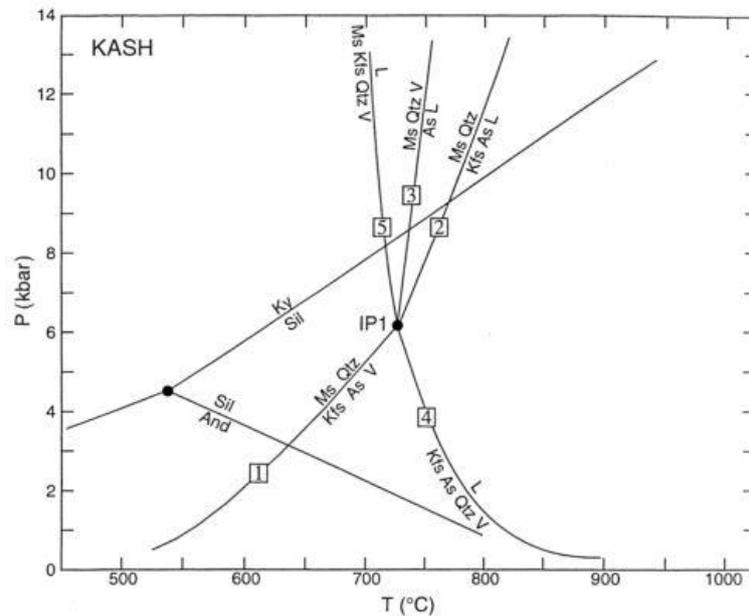


Fig. 1. Petrogenetic grid in pressure (P)–temperature (T) space showing the location of selected reactions in the system K_2O – Al_2O_3 – SiO_2 – H_2O (KASH) taken from Spear et al. (1999). Abbreviations are given as follows: And = andalusite, As = Al_2SiO_5 , IP1 = Invariant point, Kfs = K-feldspar, Ky = kyanite, L = liquid, Ms = muscovite, Qtz = quartz, Sil = sillimanite, V = H_2O . Note: Reaction number 5 will only be observed in sillimanite absent rocks

In order to investigate the possible reactions that a single rock could experience, compositionally relevant phase diagrams were created from petrogenetic grids by considering a single bulk composition thereby creating isochemical phase diagrams of P – T – X space (Hensen and Essene, 1971; Hensen and Harley, 1990) (e.g. Fig. 2). These diagrams were initially termed ‘pseudosections’ but that name has recently been argued to be removed from common usage in favour of the term “metamorphic assemblage diagrams” (Spear and Pattison, 2017). Expansion of these data sets to include fluids and melts along with the creation of activity-composition models which describe phases by proportions of respective compositional endmembers has allowed more complicated systems to be investigated and for pressure-temperature conditions above the solidus to be investigated (Berman, 1988; Powell and Holland, 1988; Ghiorso and Sack, 1995; Holland and Powell, 2011; White et al., 2014).

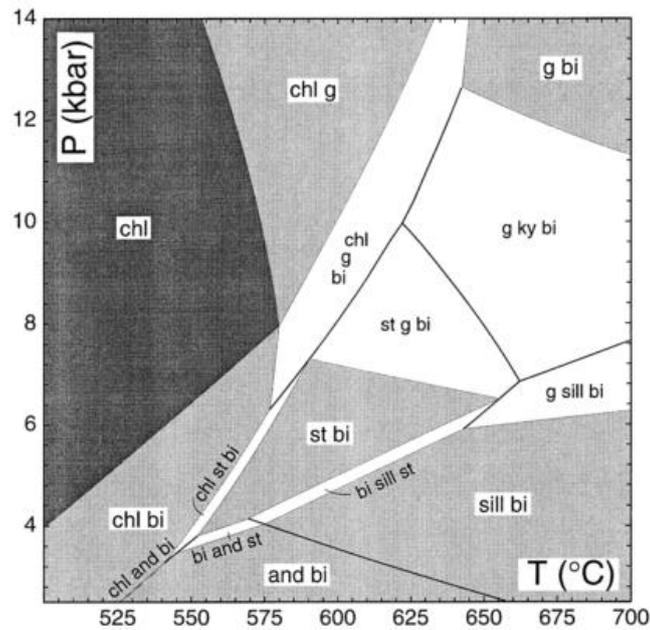


Fig. 2. Pressure (P)–temperature (T) isochemical phase diagram calculated in the system K_2O – FeO – MgO – Al_2O_3 – SiO_2 – H_2O for a pelite composition: $Al_2O_3=41.89$, $MgO=18.19$, $FeO=27.29$, and $K_2O=12.63$ (in mol.%) taken from Powell et al. (1998). Muscovite, quartz and water are in excess. Abbreviations are given as follows: and = andalusite; bi = biotite; chl = chlorite; g = garnet; ky = kyanite; sill = sillimanite; st = staurolite. Note: this diagram only shows phase assemblage fields applicable to this specific bulk composition.

Computer programs have been used to perform phase equilibrium modelling by either using the simultaneous solution of non-linear equations as is the case in THERMOCALC (Powell and Holland, 1988; Powell et al., 1998) or the minimization of Gibbs free energy of the system (ΔG) as is the case in Perple_X (Connolly and Kerrick, 1987; Connolly, 2005) and Theriak/Domino (De Capitani and Petrakakis, 2010). This modelling traditionally focuses on considering one bulk composition in isochemical phase diagrams or on the linear scaling between two bulk compositions in isobaric or isothermic phase diagrams. Lab experiments also primarily focus on systems occurring in equilibrium and those that consider melting generally only simulate equilibrium batch melting due to the technical limitations of performing phase fractionations in an experimental capsule and the challenge of diminishing volumes of the products produced if fractionation is induced. However, in some cases the modelling of natural systems mandates a more sophisticated handling of a changing bulk composition.

Recent studies have begun to focus on the effects of compositional changes in open systems (Brown and Korhonen, 2009; Yakymchuk and Brown, 2014; Stuck, 2016; Webb et al.,

2015). As rocks are buried and subjected to processes of metamorphism they can dehydrate (White and Powell, 2010; Webb et al., 2015) or experience periods of fluid infiltration (Johnson et al., 2003; Korhonen et al., 2012) and hydrothermal alteration (White et al., 2003; Riesco et al., 2004; Nicoli and Dyck, 2018). If these systems reach sufficiently high temperatures, they can begin to melt and experience melt loss (Vigneresse and Burg, 2000; White et al., 2002; White and Powell, 2010; Brown and Korhonen, 2009; Johnson et al., 2011; Diener and Fagereng, 2014; Yakymchuk and Brown, 2014; Mayne et al., 2016; Morrissey et al., 2016; Stuck and Diener, 2018). The movement of melts or magmas can form complicated systems involving entrainment of minerals to melt (Taylor and Stevens, 2010; Garcia-Arias and Stevens; Stevens et al., 2007; Clemens and Stevens, 2012) as well as magma mixing, mingling and hybridisation (Beard et al., 2005; Clemens and Stevens, 2012). At sites of emplacement magma systems may differentiate further through processes of fractional crystallisation or assimilation (Blevin and Chappell, 1992; Linnen and Keppeler, 2002). Throughout these processes chemical fractionation of major element, trace element and isotope systems occurs producing a complex array of chemical signatures which could provide insight into the mechanisms of compositional change in these rocks.

López-Carmona et al. (2014) classify the potential changes to the effective bulk composition of a rock (the portion of the rock that is available to the reacting assemblage of phases) into two mechanisms. “Open system behaviour” (discussed above) and “successive (re)equilibrations” which result from the growth of crystals with slow intracrystalline diffusion thereby temporarily removing their chemical constituents from the reactive system until diffusion may make them re-available. These two mechanisms can occur in conjunction with each other.

Lanari and Engi (2017) have presented a methodology (Lanari et al., 2018) and software tool (Lanari et al., 2014) for approximating and interrogating the volumes of equilibration in a rock which has compositionally zoned minerals. However, current thermodynamic modelling software has limited functionalities for dealing with the bulk compositional changes induced by “Open system behaviour”. Here I classify bulk compositional changes occurring in open systems as phase manipulations consisting of phase additions (e.g. fluid infiltration, wall rock assimilation) and phase extractions (e.g. dehydration, melt loss). The functionalities of the three most commonly used thermodynamic modelling programs for metamorphic studies in performing phase manipulations are detailed below and summarised in Table 1.

The use of current thermodynamic modelling software to model open system behaviour

THERMOCALC

A “read bulk info” script can be used to manually change the abundance of a phase and recalculate a bulk composition. This can be iteratively used to increase the proportion of a phase or to fractionate a phase e.g. (Zeh, 2006). By calculating individual panels each time the bulk composition changes a stitched model of open system processes can be created e.g. (Yakymchuk and Brown, 2014) (Fig.3).

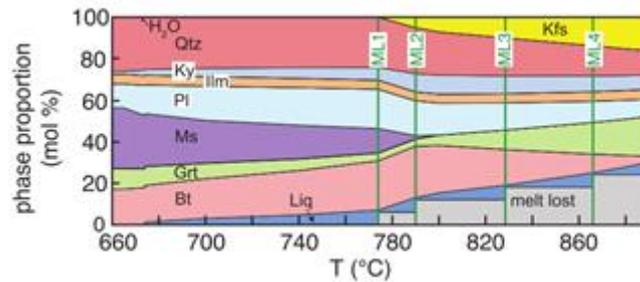


Fig. 3. Molar proportion of phases plotted against temperature for isobaric heating at 12 kbar starting with an average amphibolite-facies pelite composition calculated using THERMOCALC. The system is considered as conditionally open with melt loss induced by removing six-sevenths of the melt present wherever the P - T path intersects the 7 mol.% melt isopleth using the ‘read-bulk-info’ script. Abbreviations are as follows: Bt = biotite, Grt = garnet, Ilm = Ilmenite, Kfs = K-feldspar, Ky = kyanite, Liq = liquid, MLx = melt loss event, Ms = muscovite, Pl = plagioclase feldspar, Qtz = quartz. Taken from Yakymchuk and Brown (2014).

Theriak/Domino

Input commands in the “drv-file” can be specified to remove or add a set proportion of phase(s) from the system along a calculation path (De Capitani and Petrakakis, 2010) (Fig. 4). Commands can also be specified to remove or add set compositions along a path.

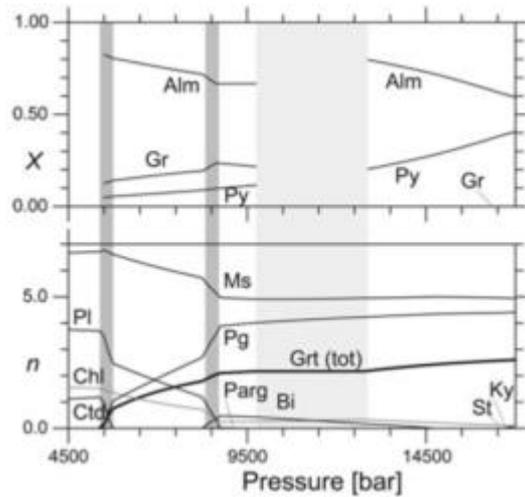


Fig. 4. Fractionation of garnet in a metapelite along a prograde P - T path calculated using Theriak/Domino. At incremental steps along the path the composition of the newly grown garnet is subtracted from the bulk composition. X shows the composition of the garnet in the reactive system as Alm = almandine, Gr = grossular and Py = pyrope. n shows the modal amount of phases in the reactive system as Bi = biotite, Chl = chlorite, Ctd = chloritoid, Grt = garnet, Ky = kyanite, Ms = muscovite, Parg = pargasite, Pg = paragonite, Pl = plagioclase, St = staurolite. The light-shaded area is a P - T segment where garnet is not growing and x is not defined. Taken from De Capitani and Petrakakis (2010).

Perple_X

“Phase fractionation calculations” can be performed by selecting this option during creation of the build file. This fractionation can be along isobaric, isothermic or a user defined P - T path (Connolly, 2005) (Fig. 5). Fractionation must remove all of a given phase(s) and the independent variable used to increment the calculation path must increase or decrease continuously.

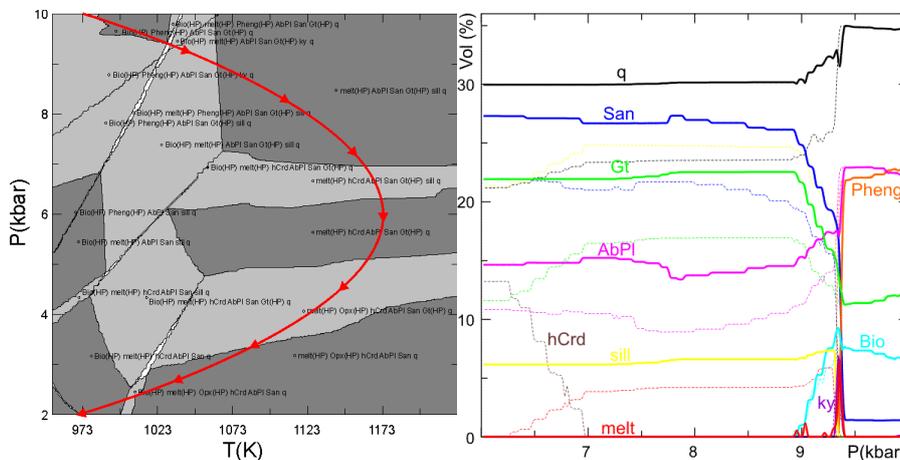


Fig. 5. Melt loss from an average metapelite bulk composition along a clockwise

decompression path (red arrow in left panel on top of an isochemical phase diagram of pressure–temperature space of the same composition) calculated using Perple_X. All melt is extracted from the system whenever it occurs as a stable phase. The volume percentage of the phases in the reactive system are shown in the right panel as AbPl = plagioclase feldspar, Bio = biotite, Gt = garnet, hCrd = cordierite, ky = kyanite, Pheng = phengite, q = quartz, San = sanadine, sill = sillimanite. Taken from (Connolly, 2018).

Table 1. Suitability of commonly used thermodynamic modelling programs for phase equilibrium modelling in open systems

| Software | Calculation method | Refinements | User Input | Phase manipulations |
|--------------------|---------------------------------|---|---------------------------------------|--|
| Thermocalc [1] | Solution of nonlinear equations | Not applicable (individual reaction lines are calculated) | Command line and scripts | At a single point, manually change proportion of phase(s) |
| Theriak Domino [2] | Gibbs free energy minimisation | Reaction boundary gridded refinement | Command line or text file input | Along a single path, fractionation or addition of a set proportion of phase(s) or set proportion of chemical components |
| Perple_X [3] | Gibbs free energy minimisation | Fixed regular grid refinement | Executable queries or text file input | Along a single path, fractionation of the full proportion of selected phase(s). Paths must be defined with one variable increasing or decreasing i.e. no loops |

[1] <http://www.metamorph.geo.uni-mainz.de/thermocalc/>; [2]

<https://titan.minpet.unibas.ch/minpet/theriak/theruser.html>; [3] <http://www.perplex.ethz.ch/>

The need for sophisticated handling of bulk compositional changes induced by phase manipulations

The presented thermodynamic modelling softwares have been shown to be able to model some open system processes however their use in this manner becomes restrictive when attempting to handle more complex open system behaviour that closer approximates the natural system. In such cases there may be a need to consider:

- Multiple phase fractionations concurrently (e.g. extraction of melt and peritectic minerals),
- Conditional phase manipulations (e.g. extraction of melt when a melt threshold is

exceeded)

- Phase manipulation proportions that are relative to the system (e.g. retention of melt on grain boundaries during melt extraction)
- Complex P – T paths (e.g. multi-stage metamorphism)
- A variety of conditional arguments in one modelling space (e.g. prograde devolatilisation of a subsolidus assemblage followed by melt loss of that assemblage above the solidus)
- Multiple potential P – T paths (e.g. the water content of a subsolidus rock is pressure dependent thus modelling in P – T space with a single water content introduces inherent inconsistencies)
- Compositional heterogeneity on the mineral, rock and crustal scale

Aims of this study

This study aims to present a methodology for the modelling of phase equilibrium in open systems. This methodology seeks to address the current limitations of thermodynamic software in the handling of bulk compositional changes induced by open system behaviour. Further, this study aims to use these new tools to build a model for crustal anatexis in which the effects of compositional change throughout this process is investigated. In order to accurately model anatexis this study will develop a methodology for the handling of the subsolidus fluid state of a system before anatexis. The combined effect of the pressure-temperature and bulk composition in this system, with sophisticated control of fluid state, will be investigated for its control on the position of the solidus. The controls of fluid state, P – T path and melt loss threshold will then be investigated for their effect on the resultant bulk composition of melts cumulatively extracted from a protolith. These source controls will be compared for their relative strength of influence and contrast with the effect of compositional heterogeneity in the source.

Structure of the thesis

This thesis presents the software program Rcrust (the name deriving from the fact that it is written in R and was developed to solve problems in crustal petrology) as a tool for the modelling of phase equilibrium in open systems.

Performing process oriented investigations

Chapter 2 presents the way in which the new functionalities of Rcrust allowed a methodology to be developed using mass transfer to perform process oriented investigations in open systems. These methodologies were built concurrently with further developments of Rcrust with the aim of creating a thermodynamically constrained model for crustal anatexis. The first challenge to this model was accurately determining the water content of a metamorphic rock. During the subsolidus prograde metamorphism of a rock the total water content of a rock may lie within the hydrous silicates of the rock, may exist as a free fluid phase in the limited pore space of the rock, or exceeding this, must leave the system. Rcrust models this behaviour by setting a maximum allowable amount of free water that can occur in the system and extracting all water that exceeds this amount. This allows the rock to progressively lose total water content from the reactive subsystem during its subsolidus evolution. Once the solidus is crossed melt is produced at the interface between reactant minerals. As this melt amount builds up an interconnected melt network may form and if it exceeds some critical threshold melt loss may occur. This process is thought to have a maximum critical threshold value by which it will occur estimated by the melt connectivity transition by Rosenberg and Handy (2005). Rcrust simulates this melt loss by extracting melt from the reactive subsystem whenever the melt threshold is exceeded and leaves a fraction of this melt behind to estimate melt retention on grain boundaries.

The effect of subsolidus water loss and its dependence on P - T path

Concurrent to building the methodology for performing process oriented investigations using Rcrust process-oriented investigations were undertaken to divulge the effects of compositional change on open systems. Chapter 3 presents a detailed study of the effect of subsolidus water loss on the position of the solidus in P - T space. When modelling subsolidus prograde water loss close to the solidus it is found that even fluid-absent rocks have the ability to melt at the fluid-saturated solidus as incremental heating before the solidus decreases the total water contained in the hydrous silicates freeing water to form a free phase in the porosity of the rock and participate in potential melting reactions. It seems that the fluid saturated solidus can only be crossed without melting if the subsolidus P - T path of the rock is sufficiently steep such that the increase in total water contained in the hydrous silicates as a function of pressure exceeds the decrease in total water contained in the hydrous silicates as a function of temperature, resulting in the rock becoming water deficient, i.e. not fully-hydrated.

The effect of source controls on open system melt compositions

These new tools and methodologies for modelling subsolidus water content, supersolidus melt threshold and varying P - T paths are used in Chapter 4 to investigate the effect of source controls on the composition of melt batches extracted sequentially from a metapelitic protolith. The resultant compositional effect of the source controls of fluid state, melt threshold and P - T path are compared and contrast to the effect of compositional variance in the protolith. It is found that the fluid state of the system is shown to have the strongest control on melt compositions with the pressure-temperature path having subordinate control on the volume and composition of melts produced.

A brief history of the development of Rcrust

Rcrust was developed following an iterative approach first implemented by Jean-François Moyen in response to extensive discussions with Arnaud Villaros and Gautier Nicoli which also culminated in a publication in the International Journal of Earth Sciences (Villaros et al., 2018). Phase stabilities are determined by calling and interpreting output from meemum from the Perple_X suite (Connolly and Kerrick, 1987; Connolly, 2009) via calls from the R environment. These calls were first developed by Jean-François Moyen during a research stay at Stellenbosch University in January 2013 and further refined by Jean-François Moyen and Gautier Nicoli in late 2013. In January 2014 the project was handed over to Matthew Mayne who became the sole programming developer from that point forward.

Compositional manipulations were split into two groups (phase additions and phase extractions) and sequenced within runs to allow path dependent processes with simultaneous additions and extractions. A graphical user interface was developed (Fig.3) using tcl/tk (Ousterhout and Jones, 2009) within R to handle user inputs and display processed outputs. In 2015 a new parsing method was employed in collaboration with Lars Kaislaniemi making use of a compiled form of meemum to speed up total calculation time (Mayne et al., 2016). A complete overhaul of the program began in 2016 in which a systematic methodology was employed to enable new modules to be activated or deactivated at points along the iterative loop. A new syntax was developed allowing a user defined modelling space to be built and conditions applied within this space to simulate the desired modelling output. The graphical user interface was redeveloped (Fig. 3) using the R package: Shiny (RStudio, 2014) in order to match these changes in functionality and to offer platform independence. File inputs of parameters were enabled and conditional inputs developed including phase thresholds and retention settings. A dependence solver was built to determine calculation order of points when points pass on compositional changes along user defined paths. Output routines were

built to process data and produce graphical outputs for contour diagrams of the modelling space, relative phase abundances along specific paths and for the creation of phase assemblage maps.

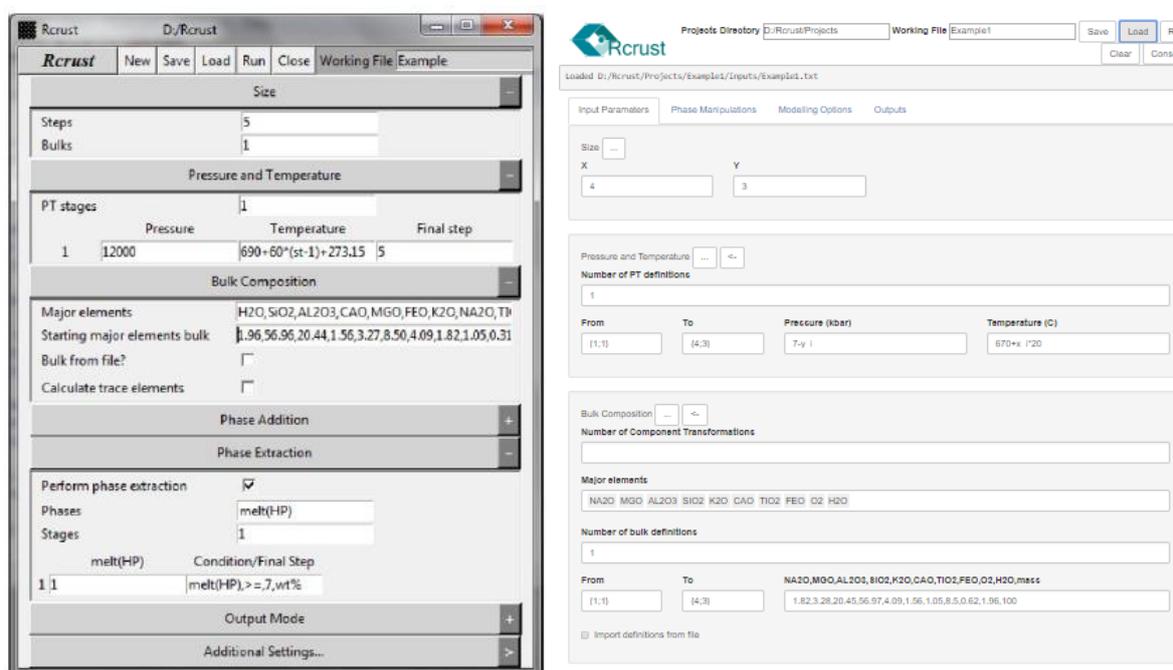


Fig. 6. Old Tcl/Tk interface (left) built on Microsoft Windows “widgets” and new Shiny interface (right) operating platform independently from a browser.

Presentation of the modelling tool: Rcrust

The software program Rcrust is attached to this thesis as Addendum A: an electronic appendix. Addendum B present a user manual for installing and using the program and Addendum C present a fully commented description and presentation of the programming code.

Supplementary data

Addendum D provides the supplementary data for Chapter 4

REFERENCES

- Albee, A.L., 1965, A petrogenetic grid for the Fe-Mg silicates of pelitic schists: *American Journal of Science*, v. 263, p. 512–536.
- Amiotte Suchet, P., and Probst, J.L., 1995, A global model for present-day atmospheric/soil CO₂ consumption by chemical erosion of continental rocks (GEM-CO₂): *Tellus*, v. 47B, p. 273–280.
- Beard, J.S., Ragland, P.C., and Crawford, M.L., 2005, Reactive bulk assimilation: A model for crust-mantle mixing in silicic magmas: *Geology*, v. 33, p. 681–684, doi: 10.1130/G21470.1.
- Berman, R.G., 1988, Internally-consistent thermodynamic data for minerals in the system Na₂O-K₂O-CaO-MgO-FeO-Fe₂O₃-Al₂O₃-SiO₂-TiO₂-H₂O-CO₂: *Journal of Petrology*, v. 29, p. 445–522.
- Berman, R.G., 1991, Thermobarometry Using Multi-Equilibrium Calculations: a New Technique, With Petrological Applications: *Canadian Mineralogist*, v. 29, p. 833–855.
- Blevin, P.L., and Chappell, B.W., 1992, The role of magma sources, oxidation states and fractionation in determining the granite metallogeny of eastern Australia: *Transactions of the Royal Society of Edinburgh: Earth Sciences*, v. 83, p. 305–316, doi: 10.1017/S0263593300007987.
- Brown, M., and Korhonen, F.J., 2009, Some Remarks on Melting and Extreme Metamorphism of Crustal Rocks, *in* *Physics and Chemistry of the Earth's Interior*, p. 67–87.
- De Capitani, C., and Petrakakis, K., 2010, The computation of equilibrium assemblage diagrams with Theriak/Domino software: *American Mineralogist*, v. 95, p. 1006–1016, doi: 10.2138/am.2010.3354.
- Clemens, J.D., and Stevens, G., 2012, What controls chemical variation in granitic magmas? *Lithos*, v. 134–135, p. 317–329, doi: 10.1016/j.lithos.2012.01.001.
- Connolly, J.A.D., 2005, Computation of phase equilibria by linear programming: A tool for geodynamic modeling and its application to subduction zone decarbonation: *Earth and Planetary Science Letters*, v. 236, p. 524–541, doi: 10.1016/j.epsl.2005.04.033.
- Connolly, J.A.D., 2018, Phase (Melt) Fractionation with Perple_X, http://www.perplex.ethz.ch/perplex_phase_fractionation.html (accessed October 2018).
- Connolly, J.A.D., 2009, The geodynamic equation of state: What and how: *Geochemistry, Geophysics, Geosystems*, v. 10, p. 1–19, doi: 10.1029/2009GC002540.
- Connolly, J.A.D., and Kerrick, D.M., 1987, An algorithm and computer program for calculating composition phase diagrams: *Calphad*, v. 11, p. 1–55, doi: 10.1016/0364-5916(87)90018-6.
- Diener, J.F.A., and Fagereng, Å., 2014, The influence of melting and melt drainage on crustal rheology during orogenesis: *Journal of Geophysical Research: Solid Earth*, v. 119, p. 6193–6210, doi: 10.1002/2014JB011088. Received.
- Friend, J.P., 1973, The Global Sulfur Cycle, *in* S.I., R. ed., *Chemistry of the Lower Atmosphere*, Springer, p. 177–178.
- Garcia-Arias, M., and Stevens, G. Melting behavior of a metasedimentary source: Consequences for S-type granite magma compositions.: *Journal of Chemical Information and Modeling*.
- Ghiorso, M.S., and Sack, R.O., 1995, Chemical mass transfer in magmatic processes IV. A revised and internally consistent thermodynamic model for the interpolation and extrapolation of liquid-solid equilibria in magmatic systems at elevated temperatures and pressures: *Contributions to Mineralogy and Petrology*, v. 119, p. 197–212.
- Gottschalk, M., 1997, Internally consistent thermodynamic data for rock-forming minerals in the system SiO₂-TiO₂-Al₂O₃-CaO-MgO-FeO-K₂O-Na₂O-H₂O-CO₂: *European Journal of Mineralogy*, v. 9, p. 175–223.
- Helgeson, H.C., Delany, J.M., and Nesbitt, H.W., 1978, Summary and critique of the thermodynamic properties of rock-forming minerals: *American Journal of Science*, v. 278A, p. 1–229.
- Hensen, B.J., and Essene, E.J., 1971, Stability of pyrope-quartz in the system MgO-Al₂O₃-SiO₂: *Contributions to Mineralogy and Petrology*, v. 30, p. 72–83.
- Hensen, B.J., and Harley, S.L., 1990, Graphical analysis of P–T–X relations in granulite facies metapelites, *in* *High-temperature metamorphism and crustal anatexis*, Netherlands, Springer, p. 19–56.
- Holland, T.J.B., and Powell, R., 2011, An improved and extended internally consistent thermodynamic dataset

- for phases of petrological interest, involving a new equation of state for solids: *Journal of Metamorphic Geology*, v. 29, p. 333–383, doi: 10.1111/j.1525-1314.2010.00923.x.
- Holland, T., and Powell, R., 1998, An internally consistent thermodynamic data set for phases of petrological interest: *Journal of Metamorphic Geology*, v. 16, p. 309–343.
- Johnson, T., Gibson, R.L., Brown, M., Buick, I., and Cartwright, I., 2003, Partial Melting of Metapelitic Rocks Beneath the Bushveld Complex, South Africa: *Journal of Petrology*, v. 44, p. 301–314, doi: 10.1093/petrology/44.5.789.
- Johnson, T.E., White, R.W., and Brown, M., 2011, A year in the life of an aluminous metapelite xenolith — The role of heating rates, reaction overstep, H₂O retention and melt loss: *Lithos*, v. 124, p. 132–143, doi: 10.1016/j.lithos.2010.08.009.
- Korhonen, F.J., Powell, R., and Stout, J.H., 2012, Stability of sapphirine + quartz in the oxidized rocks of the Wilson Lake terrane, Labrador: Calculated equilibria in NCKFMASHTO: *Journal of Metamorphic Geology*, v. 30, p. 21–36, doi: 10.1111/j.1525-1314.2011.00954.x.
- Lanari, P., Bovay, T., Airaghi, L., and Centrella, S., 2018, Quantitative compositional mapping of mineral phases by electron probe micro-analyser:.
- Lanari, P., and Engi, M., 2017, Local Bulk Composition Effects on Metamorphic Mineral Assemblages: *Reviews in Mineralogy and Geochemistry*, v. 83, p. 55–102, doi: 10.2138/rmg.2017.83.3.
- Lanari, P., Vidal, O., De Andrade, V., Dubacq, B., Lewin, E., Grosch, E.G., and Schwartz, S., 2014, XMapTools: A MATLAB©-based program for electron microprobe X-ray image processing and geothermobarometry: *Computers and Geosciences*, v. 62, p. 227–240, doi: 10.1016/j.cageo.2013.08.010.
- Linnen, R.L., and Keppler, H., 2002, Melt composition control of Zr/Hf fractionation in magmatic processes: *Geochimica et Cosmochimica Acta*, v. 66, p. 3293–3301, doi: 10.1016/S0016-7037(02)00924-9.
- López-Carmona, A., Gutiérrez-Alonso, G., Tishin, P.A., and Gertner, I.F., 2014, Thermodynamic modelling of metamorphic processes: state of the art in pseudosection approach: *IOP Conference Series: Earth and Environmental Science*, v. 110.
- Mayne, M.J., Moyen, J.F., Stevens, G., and Kaislaniemi, L., 2016, Rcrust: a tool for calculating path-dependent open system processes and application to melt loss: *Journal of Metamorphic Geology*, v. 34, p. 663–682, doi: 10.1111/jmg.12199.
- Morrissey, L.J., Hand, M., Lane, K., Kelsey, D.E., and Dutch, R.A., 2016, Upgrading iron-ore deposits by melt loss during granulite facies metamorphism: *Ore Geology Reviews*, v. 74, p. 101–121, doi: 10.1016/j.oregeorev.2015.11.012.
- Nicoli, G., and Dyck, B., 2018, Exploring the metamorphic consequences of secular change in the siliciclastic compositions of continental margins: *Geoscience Frontiers*, p. 1–9, doi: 10.1016/j.gsf.2017.12.009.
- Ousterhout, J.K., and Jones, K., 2009, Tcl and the Tk toolkit:.
- Powell, R., and Holland, T., 1988, An internally consistent dataset with uncertainties and correlations: 3. Applications to geobarometry, worked examples and a computer program: *Journal of Metamorphic Geology*, v. 6, p. 173–204, doi: 10.1111/j.1525-1314.1988.tb00415.x.
- Powell, R., and Holland, T.J.B., 1985, An internally consistent thermodynamic dataset with uncertainties and correlations: 1. Methods and a worked example.: *Journal of Metamorphic Geology*, v. 3, p. 327–342.
- Powell, R., Holland, T., and Worley, B., 1998, Calculating phase diagrams involving solid solutions via non-linear equations, with examples using THERMOCALC: *Journal of Metamorphic Geology*, v. 16, p. 577–588, doi: 10.1111/j.1525-1314.1998.00157.x.
- Riesco, M., Stüwe, K., Reche, J., and Martínez, F.J., 2004, Silica depleted melting of pelites. Petrogenetic grid and application to the Susqueda aureole, Spain: *Journal of Metamorphic Geology*, v. 22, p. 475–494, doi: 10.1111/j.1525-1314.2004.00527.x.
- Rosenberg, C.L., and Handy, M.R., 2005, Experimental deformation of partially melted granite revisited: Implications for the continental crust: *Journal of Metamorphic Geology*, v. 23, p. 19–28, doi: 10.1111/j.1525-1314.2005.00555.x.
- RStudio, 2014, Shiny: Easy web applications in R:, <http://shiny.rstudio.com>.

- Spear, F.S., Kohn, M.J., and Cheney, J.T., 1999, P-T paths from anatectic pelites: Contributions to Mineralogy and Petrology, v. 134, p. 17–32.
- Spear, F.S., and Pattison, D.R.M., 2017, The implications of overstepping for metamorphic assemblage diagrams (MADs): Chemical Geology, v. 457, p. 38–46, doi: 10.1016/j.chemgeo.2017.03.011.
- Spear, F.S., Pattison, D.R.M., and Cheney, J.T., 2016, The metamorphosis of metamorphic petrology: Geological Society of America Special Paper, v. 523, p. 31–74, doi: 10.1130/2016.2523(02).
- Stevens, G., Villaros, A., and Moyen, J.F., 2007, Selective peritectic garnet entrapment as the origin of geochemical diversity in S-type granites: Geology, v. 35, p. 9–12, doi: 10.1130/G22959A.1.
- Stuck, T.J., 2016, Mineral Equilibria Constraints on Open-System Melting and Consequences of Melt Loss in Metabasic Rocks: 61 p.
- Stuck, T.J., and Diener, J.F.A., 2018, Mineral equilibria constraints on open-system melting in metamafic compositions: Journal of Metamorphic Geology, v. 36, p. 255–281, doi: 10.1111/jmg.12292.
- Taylor, J., and Stevens, G., 2010, Selective entrainment of peritectic garnet into S-type granitic magmas: Evidence from Archaean mid-crustal anatectites: Lithos, v. 120, p. 277–292, doi: 10.1016/j.lithos.2010.08.015.
- Vigneresse, J.L., and Burg, J.P., 2000, Continuous vs. discontinuous melt segregation in migmatites: Insights from a cellular automaton model: Terra Nova, v. 12, p. 188–192, doi: 10.1046/j.1365-3121.2000.00299.x.
- Villaros, A., Laurent, O., Couzinié, S., Moyen, J.F., and Mintrone, M., 2018, Plutons and domes: the consequences of anatectic magma extraction—example from the southeastern French Massif Central: International Journal of Earth Sciences, p. 1–24, doi: 10.1007/s00531-018-1630-x.
- Webb, G., Powell, R., and McLaren, S., 2015, Phase equilibria constraints on the melt fertility of crustal rocks: The effect of subsolidus water loss: Journal of Metamorphic Geology, v. 33, p. 147–165, doi: 10.1111/jmg.12114.
- White, R.W., and Powell, R., 2010, Retrograde melt-residue interaction and the formation of near-anhydrous leucosomes in migmatites: Journal of Metamorphic Geology, v. 28, p. 579–597, doi: 10.1111/j.1525-1314.2010.00881.x.
- White, R.W., Powell, R., and Clarke, G.L., 2002, The interpretation of reaction textures in Fe-rich metapelitic granulites of the Musgrave Block, Central Australia: Constraints from mineral equilibria calculations in the system: Journal of Metamorphic Geology, v. 20, p. 41–55, doi: 10.1046/j.0263-4929.2001.00349.x.
- White, R.W., Powell, R., Holland, T.J.B., Johnson, T.E., and Green, E.C.R., 2014, New mineral activity-composition relations for thermodynamic calculations in metapelitic systems: Journal of Metamorphic Geology, v. 32, doi: 10.1111/jmg.12071.
- White, R.W., Powell, R., and Phillips, G.N., 2003, A mineral equilibria study of the hydrothermal alteration in mafic greenschist facies rocks at Kalgoorlie, Western Australia: Journal of Metamorphic Geology, v. 21, p. 455–468, doi: 10.1046/j.1525-1314.2003.00454.x.
- Yakymchuk, C., and Brown, M., 2014, Consequences of open-system melting in tectonics: Journal of the Geological Society, v. 171, p. 21–40, doi: 10.1144/jgs2013-039.
- Zeh, A., 2006, Calculation of garnet fractionation in metamorphic rocks, with application to a flat-top, Y-rich garnet population from the Ruhla Crystalline Complex, Central Germany: Journal of Petrology, v. 47, p. 2335–2356, doi: 10.1093/petrology/egl046.

CHAPTER 2

PRESENTATION OF RESEARCH PAPER 1: PERFORMING PROCESS-ORIENTED INVESTIGATIONS INVOLVING MASS TRANSFER USING RCRUST: A NEW PHASE EQUILIBRIUM MODELLING TOOL

This paper, first authored by Matthew Mayne, was accepted for publication in the Geological Society of London Special Issue: Making granites: petrogenetic processes, tectonic environments and secular variations.

The following aspects of the research were done independently by Matthew Mayne while receiving standard supervision by his supervisors Gary Stevens, Jean-François Moyen and Tim Johnson: (i) writing of programming code; (ii) conducting phase equilibrium modelling; (iii) generation of the figures; (v) writing of the manuscript

Performing process-oriented investigations involving mass transfer using Rcrust: a new phase equilibrium modelling tool

Matthew Jason Mayne ^{1,2,*}, **Gary Stevens** ¹, **Jean-François Moyen** ² & **Tim Johnson** ³

¹University of Stellenbosch, Department of Earth Sciences, Private Bag X1, 7602 Matieland, South Africa

²Université de Lyon, Laboratoire Magmas et Volcans, UJM-UCA-CNRS-IRD, 42023 Saint-Etienne, France

³Curtin University, Department of Applied Geology, WA 6845 Perth, Australia

*Corresponding author (e-mail: mmayne@sun.ac.za)

Authors' ORCiDs:

Matthew Jason Mayne, 0000-0002-2103-9647;

Gary Stevens, 0000-0003-1593-9419;

Jean-François Moyen, 0000-0002-0065-2442

Tim Johnson 0000-0001-8704-4396

Abbreviated title: “Performing process-oriented investigations with Rcrust”

ABSTRACT

Modern quantitative phase equilibria modelling allows the calculation of the stable phase assemblage of a rock system given its pressure, temperature and bulk composition. A new software tool (Rcrust) has been developed that allows the modelling of points in pressure–temperature–bulk composition space in which bulk compositional changes can be passed from point to point as the system evolves. This new methodology enables quantitative process-oriented investigation of the evolution of rocks. Procedures are outlined here for using this tool to model: 1) the control of the water content of a subsolidus system based on available pore space; 2) triggering of melt loss events when a critical melt volume threshold is exceeded, while allowing a portion of melt retention; 3) entrainment of crystals during segregation and ascent of granitic magmas from its source; 4) modification of the composition of granite magmas due to fractional crystallization and 5) progressive availability (through dissolution) of slow diffusing species and their control on the effective bulk composition of a system. These cases collectively illustrate thermodynamically constrained methods for modelling systems that involve mass transfer.

Keywords: Phase equilibria modelling; Rcrust; effective bulk composition; water content; anatexis; melt loss; peritectic entrainment; fractional crystallization; volume of equilibration; dissolution–precipitation.

INTRODUCTION

Given the pressure (P), temperature (T) and bulk composition (X) of a rock, quantitative phase equilibria modelling can predict the stable phase assemblages, the P – T fields of stability of these assemblages, phase modes and phase compositions, that characterize the system at equilibrium see THERMOCALC (Powell and Holland, 1988; Powell et al., 1998), Perple_X (Connolly and Kerrick, 1987) and Theriak/Domino (De Capitani and Petrakakis, 2010). This technique has become a standard tool in metamorphic studies, most commonly involving analysis via calculation of isochemical phase diagram sections known as *pseudosections* e.g. Yakymchuk (2017) or metamorphic assemblage diagrams (MADs) (Spear and Pattison, 2017). Recent studies have begun to consider the effects of bulk compositional change in rocks undergoing metamorphism, e.g. through melt loss (Yakymchuk and Brown, 2014; Morrissey et al., 2016; Vigneresse and Burg, 2000; Brown and Korhonen, 2009; White and Powell, 2002, 2010; Diener and Fagereng, 2014; Mayne et al., 2016; Johnson et al., 2011), fluid loss (White and Powell, 2010), fluid infiltration (Korhonen et al., 2012; Johnson et al., 2003) and/or metasomatism (White et al., 2003; Riesco et al., 2004; Nicoli and Dyck, 2018). In these studies, isochemical phase diagram sections are limiting and alternative methods have been employed. To account for a changing bulk composition via subtraction or addition of portions of one or more phases, isothermal P – X or isobaric T – X diagrams are commonly used, in which bulk composition is allowed to vary between two predetermined endmembers. In cases where more than one compositional change is required, a number of individually calculated panels are stitched together (Yakymchuk and Brown, 2014; White and Powell, 2002; Brown and Korhonen, 2009; Johnson et al., 2011). However, when investigating processes that sequentially alter the bulk composition of a system within P – T space, manual calculation and stitching of diagrams becomes restrictive. An alternative, to this calculation of predetermined phase changes, is to include parameters for physical mechanisms of mass transfer within a calculation such that bulk compositional changes occur within a system based on properties that the system obtains such as phase abundances or phase compositions (Connolly, 2005; Karpov et al., 1997; Ghiorso and Sack, 1995). The need for increasingly more sophisticated models that allow this style of mass transfer prompted the creation of a path dependent thermodynamic modelling tool (called ‘Rcrust’) which automates the handling of bulk compositional change in a modelled P – T – X space (Mayne et al., 2016). This paper serves as a summary of the functionalities of this program and demonstrates some applications in

performing process-oriented investigations that inherently involve bulk compositional change. This provides a useful tool for reconciling petrographic observations and chemical analyses with thermodynamic modelling at a variety of scales within complex systems in order to better understand phase equilibria.

RCRUST: A NEW PHASE EQUILIBRIUM MODELLING TOOL

Path dependence of the reactive subsystem

Rcrust operates by assigning P - T - X conditions to a number of points that define a modelling space. A `Perple_X` routine is then used which minimizes the Gibbs free energy of the system in order to predict the equilibrium phase assemblage at each point (Connolly, 2005, 2009). Modelling can be performed on compositions including pelites, greywackes, calc-silicates and mafic rocks as far as the available solution models in `Perple_X` allow within the full P - T range of crustal metamorphism and magmatism. Further information about modelling tolerances can be found in the solution model files and references therein. What is unique in Rcrust's methodology is that the conditions of points in the modelling space can be assigned with reference to other points. If dependencies are assigned, then calculations first solve the hierarchy of dependence of points, after which calculations proceed sequentially along paths that connect the points. This 'path dependence' allows bulk compositional changes to be passed on from point to point, thereby permitting investigation of processes that are dependent on the evolving assemblage as the rock moves through pressure-temperature-bulk compositional space.

To best model the behaviour of rock systems that gain or lose material, it is helpful to consider the system as consisting of a number of subsystems which are, at least temporarily, in chemical isolation from one another. For example, the bulk composition of a rock, or the full system (FS), can be thought of as consisting of: 1. a reactive subsystem (RS) which represents the portion that is currently in chemical equilibrium at the P - T conditions of the system; 2. an extract subsystem (ES) which represents the portion that has been extracted from the RS, during the current and prior steps, and is no longer in chemical equilibrium with the RS. The ES is not subject to further phase equilibrium modelling; and 3. an isolated subsystem (IS) which represents that portion of the FS that has not yet contributed to the reactive subsystem but may later become part of it (Fig. 1). In this way a rock undergoing anatexis, for example, can be thought of as a portion which is currently melting (the reactive

subsystem), a portion which has been extracted through melt loss (the extract subsystem) and a portion, for example, the cores of compositionally zoned pre-anatectic porphyroblasts that have yet to contribute to the effective bulk composition of the reactive subsystem (the isolated subsystem). By transferring bulk material between these systems as the path progresses we can effectively model processes that effect a rocks evolution.

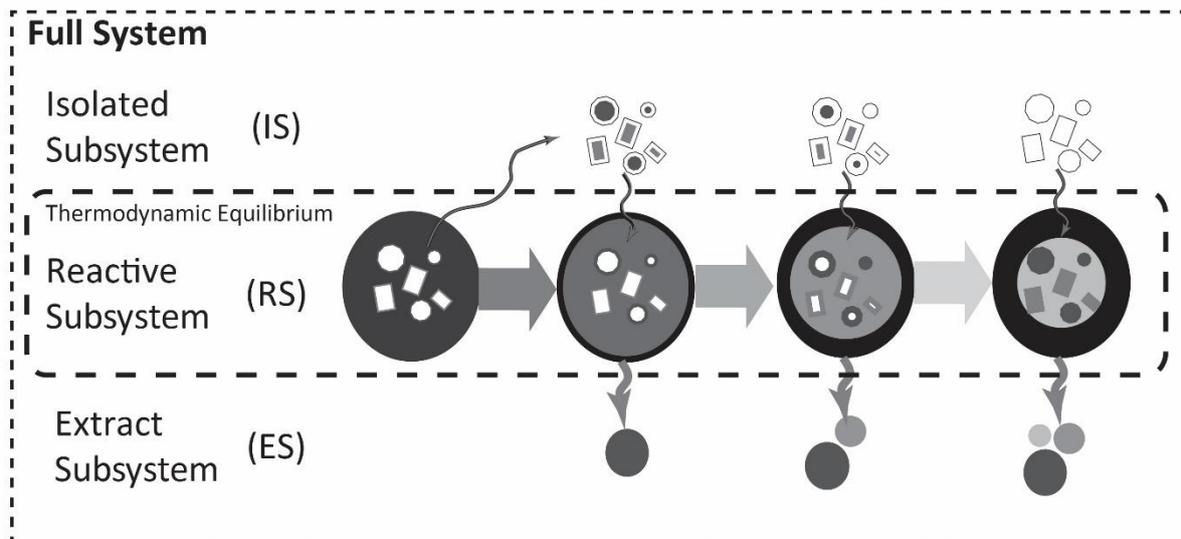


Fig. 1. Division of the bulk composition of the rock (full system) into a portion that is not yet incorporated into the chemical system (isolated subsystem); a portion that contains the chemical system currently in thermodynamic equilibrium (reactive subsystem) and a portion that is no longer part of the chemical system (extract subsystem). Effective bulk compositional change can be induced in the rock by moving crystals and/or melt and/or fluid between these subsystems.

Modelling setup

The Rcrust program is operated through a graphical user interface written in R Copyright© 2016 the R Foundation for Statistical Computing (R Core Team, 2016) and therefore requires a working version of R which can be downloaded freely from <http://www.R-project.org/>. The Rcrust interface writes a text document which is read by the calculation routine when run. This enforces reproducibility between calculations as a record of all input parameters is saved with each calculation. These input files can be shared between users to allow direct correlation between calculations. For advanced users the GUI can be bypassed

by manually editing input files and accessing Rcrust directly from the console.

In a standard calculation the user first defines the size of the modelling space by setting the number of points in the P – T window to be considered. Once the number of points has been defined, P – T conditions are then attributed to each point by defining functions (Fig. 2b). These definitions can be as simple as individually assigning P – T values for each point or may be complex functions that define, for example a non-linear P – T path (e.g. the clockwise P – T loop in Mayne et al. fig. 5). Importantly, the axes of the modelling space do not have to be P and T , but may be a function of bulk composition or the progression of P – T – X paths or any other variable of interest. The next parameter that the user needs to define is the bulk composition (Fig. 2c). Again, this can be simple, for example comprising the weight percentage of each of the oxides that define the reactive subsystem, or a more complex function of subsystems from a previously calculated point (e.g. the bulk composition of the reactive subsystem of the previous point along a P – T path). This functionality allows the user to create P – T paths that have path dependent bulk compositions.

At any point during a calculation, phase manipulations can be used to alter the bulk composition of subsystems by either adding or extracting phases to or from the reactive subsystem (Fig. 2d). Such manipulations can be statically defined prior to the calculation of phase assemblages (for example to add a set amount of water progressively into the system as a function of progress along a path) or dynamically defined in response to phase assemblage proportions (e.g. melt loss events triggered by exceeding a critical melt threshold). The latter also allows phase manipulations to respond to P – T dependent criteria, for example the volume of a phase in the phase assemblage (which itself is also dependent on pressure). The combination of phase manipulations and path dependence can be used to approximate mechanistic processes affecting the evolution of a rock. Output data can be viewed directly in the Rcrust interface or exported as text or postscript files for editing. Graphing routines are available in Rcrust, including an interpolation routine for projecting P – T paths onto a grid to allow contouring, and a routine for the creation of “phase assemblage maps”, a term proposed here to describe graphs that show the change in phase assemblage of a system where pressure, temperature and bulk composition can be independently variable and path-dependent within the diagram (i.e. not a *pseudosection*) from a grid of points (Fig. 2e).

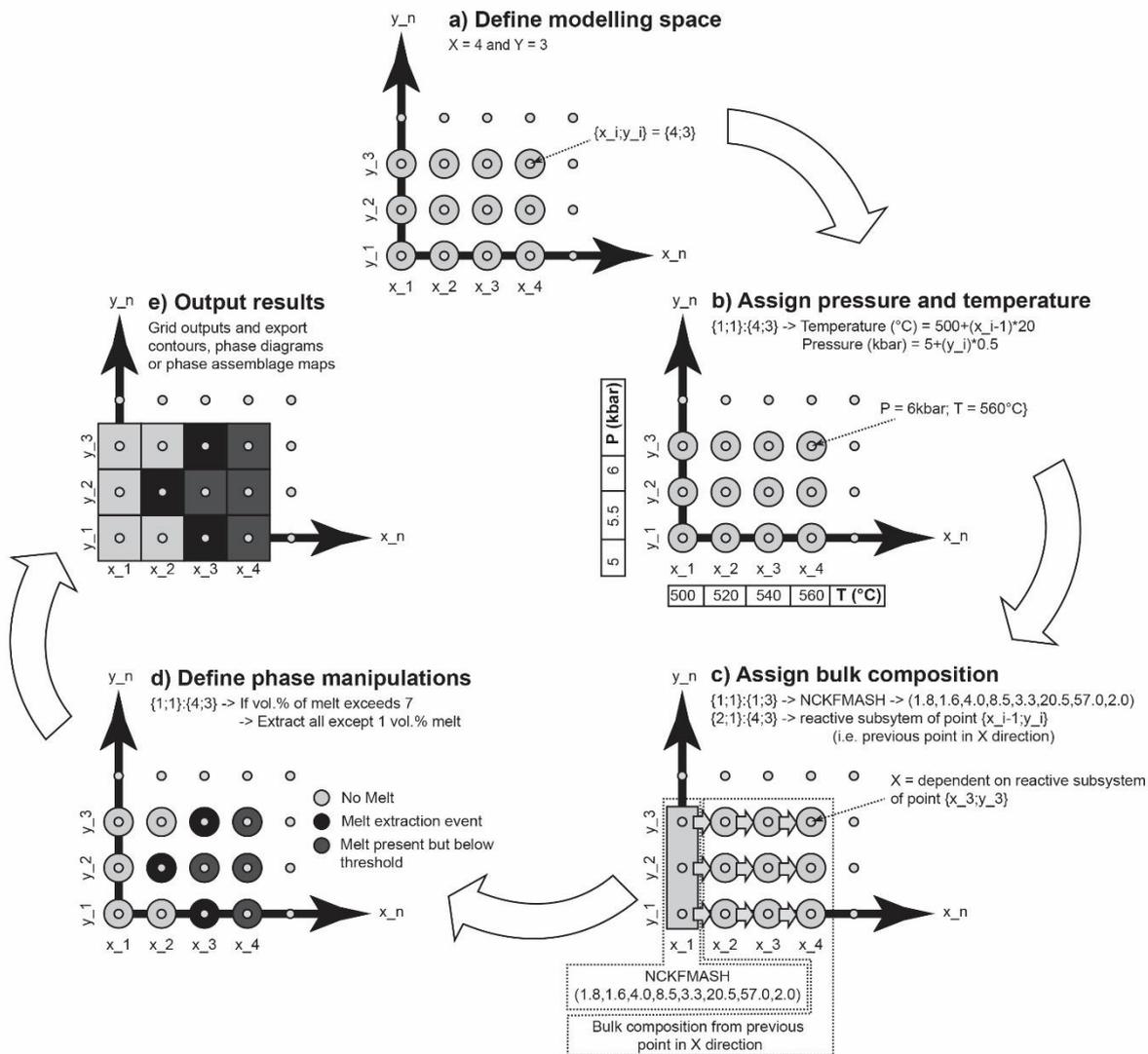


Fig. 2. Procedure for modelling using the phase equilibria tool Rcrust. (a) Define the number of points in the modelling space. (b) Assign pressure and temperature conditions. (c) Assign the bulk composition of the reactive subsystem. (d) Define any required phase additions or phase extractions. (e) project outputs to grids and export contour, phase diagram sections or phase assemblage maps.

PERFORMING PROCESS-ORIENTED INVESTIGATIONS

To illustrate the methodology by which process-oriented investigations can be undertaken modelling was performed in Rcrust version 2017-10-26 (Mayne et al., 2016) with a compiled form of the meemum function from the Perple_X suite of programs (Connolly,

2009). The NCKFMASHTO chemical system was utilized with the 2011 revised hp11ver.dat thermodynamic data file from the internally consistent dataset of (Holland and Powell, 2011). Solution models were chosen which are consistent with the slightly simplified chemistry of the bulk system (e.g. the chemical system does not account for Mn) yet takes into account substitutions that are important in stabilizing phases (e.g. Ti in biotite). The following solution models were used from solution_model_679.dat: Fsp(C1) for plagioclase and alkali-feldspar (Holland and Powell, 2003), Cpx(HP) for clinopyroxene (Holland and Powell, 1996), Ep(HP11) for epidote (Holland and Powell, 2011), Sp(WPC) for spinel (White et al., 2002), Ilm(WPH) for ilmenite (White et al., 2000) and the latest White *et al* (2014) set of models: Bi(W) for biotite, Crd(W) for cordierite, Gt(W) for garnet, melt(W) for melt, Mica(W) for micas, Opx(W) for orthopyroxene, St(W) for staurolite. Abbreviations for phases in figures except for melt and mica were taken from Whitney & Evans (2010) as Bt, Biotite; Crd, cordierite; Fsp, plagioclase and alkali-feldspar; Grt, garnet; H₂O, liquid water; Ilm, ilmenite; Qz, quartz; Sil, sillimanite; Spl, spinel; St, staurolite.

Dynamic setting of subsolidus water content of the Reactive System

The water content of the combined hydrous silicate phases of a subsolidus rock changes as a function of pressure and temperature as both the abundance and composition of those hydrous silicates changes. Any H₂O that cannot be accommodated by the hydrous phases must either be contained within the pore space or must leave the rock system. To model this behaviour, the bulk composition of the system can be set to start with an excess of water and phase extraction routines set such that free water is extracted whenever its volume exceeds some value that is considered to represent a reasonable estimate of the porosity. Figure. 3 shows the phase relations for an average pelite bulk composition (Table 1) from Ague (1991) which evolved along an isobaric 7 kbar heating path from 670 to 920 °C. The bulk composition begins with an excess of water (10 wt.%) and all water that exceeds 0.1 vol.% is extracted in each step. Water extracted from the reactive subsystem is kept in isolation from the reactive subsystem and the abundance of phases are only shown relative to the phases within the reactive subsystem. This process is described in further detail in Mayne *et al* (2017).

| Average amphibolite facies pelite composition from Ague (1991) | | | | | | | | | | |
|--|------------------|------------------|--------------------------------|------|------|------|-------------------|------------------|------|------------------|
| | SiO ₂ | TiO ₂ | Al ₂ O ₃ | FeO | MgO | CaO | Na ₂ O | K ₂ O | O | H ₂ O |
| wt.% | 52.15 | 0.96 | 18.81 | 7.82 | 3.11 | 1.43 | 1.68 | 3.76 | 0.28 | 10.00 |

Table 1. Bulk composition of an average amphibolite-facies pelite from Ague (1991)

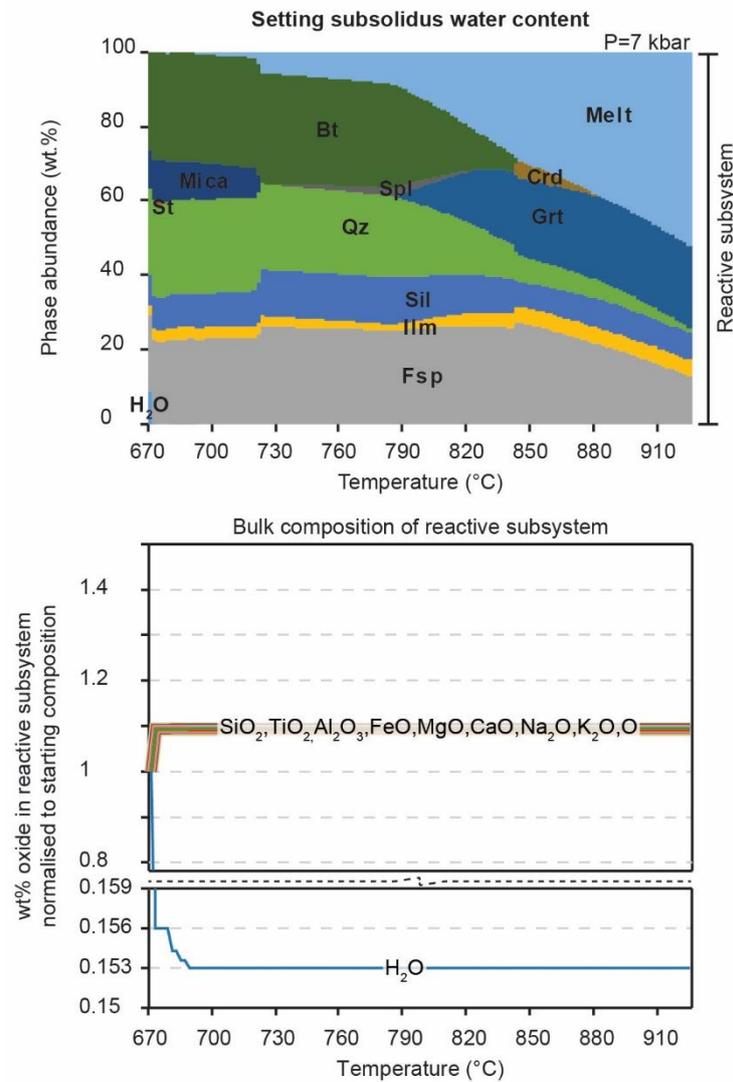


Fig. 3. Modelling subsolidus water content by starting with excess water and then extracting all water that exceeds 0.1 vol.% (approximating a pore space). Phase abundance in wt.% of the reactive subsystem versus temperature for the isobaric heating path at 7 kbar from 670-920 °C starting with the pelite bulk composition in Table 1. The wt.% oxide in the bulk composition of the reactive subsystem is shown normalised to the starting bulk composition of the path.

Melt loss through critical threshold values and retention amounts

When rocks melt, the melt initially forms along grain boundaries. As melting progresses, the volume of melt may reach some critical value such that it forms an interconnected network and can be extracted from the local rock system (Yakymchuk and Brown, 2014; Brown and Korhonen, 2009). To design a system which is open to melt loss, a critical melt threshold, upon which melt loss will occur, needs to be defined, for example when 80% of grain boundaries become melt bearing (the melt connectivity transition (MCT) proposed by (Rosenberg & Handy (2005)). These thresholds can be approximated by volumetric constraints (MCT achieved at c. 7 vol.%) and incorporated into modelling parameters. When melt loss events occur they are very unlikely to be 100% efficient; some melt will inevitably remain within isolated pores/pockets and/or on grain boundaries. To model this behaviour, phase extraction routines can be defined to trigger when the volume of melt exceeds a set value, and to extract some proportion of that melt, perhaps leaving a small amount (e.g. 1 vol.%) in the reactive subsystem to account for melt retention on grain boundaries. This process is extensively discussed in the literature (Yakymchuk and Brown, 2014; Morrissey et al., 2016; Vigneresse and Burg, 2000; Brown and Korhonen, 2009; White and Powell, 2002, 2010; Diener and Fagereng, 2014; Johnson et al., 2011). Figure 4 shows the phase relations for the isobaric heating path with water extraction routines as described in Figure 3 but with an additional extraction routine set to occur whenever 7 vol.% melt is exceeded extracting all melt except 1 vol.%. The cumulative melt extracted from the reactive subsystem is labelled as “Melt_es” and is isolated from the reactive subsystem. Rcrust’s abilities in this regard are described further in Mayne et al. (2016). Farina et al. (2017) use this methodology to correlate multi-pulsed magmatism in the Monte Capanne pluton with multimodal distribution of zircon ages in order to estimate the duration of partial melting and the timescale of melt extraction from the source.

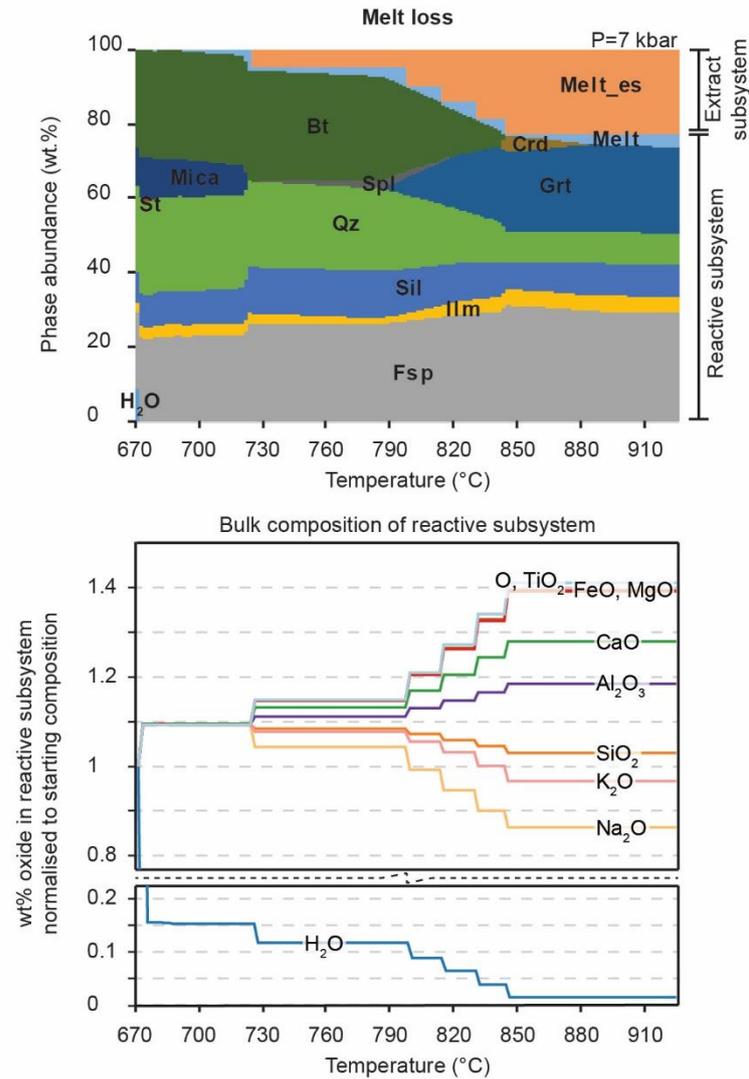


Fig. 4. Melt loss triggered when melt volume exceeds 7 vol.% and set to extract all melt except 1 vol.% (approximating melt retention on grain boundaries) for the isobaric heating path at 7 kbar from 670-920 °C. Handling of subsolidus water content and starting bulk composition as in Figure 3. The wt.% oxide in the bulk composition of the reactive subsystem is shown normalised to the starting bulk composition of the path.

Entrainment of phases to melt (magma loss)

Melts often entrain crystals such as peritectic minerals on segregation from the source (Stevens et al., 2007). This creates magmas whose chemical composition is some combination of the volume of segregated melt and the composition and abundance of the entrained crystals. When these magmas segregate, the bulk composition of the reactive anatectic system is changed, with implications for subsequent melting behaviour. Phase extraction routines can be modified to model this behaviour by extracting proportions of additional phases whenever melt loss events are triggered. Within modelling functions, some peritectic mineral products formed by melting reactions may have the same name as pre-existing minerals e.g. plagioclase feldspar. In order to distinguish between the pre-existing subsolidus phases and the newly formed peritectic phases a ‘delta function’ can be applied. The delta function allows the user to specify only the portion of a mineral phase that increased in mass during melting. This can be used, for example, to selectively extract peritectic minerals along with melt (Stevens et al., 2017). Figure 5 shows the phase relations for the isobaric heating path with water and melt extraction described in Figure 3 and Figure 4 respectively. Additional to these conditions, when melt extraction events occur 30% of the increase in mass, of any phase excluding melt, between the current point and either the last melt extraction event or the last melt absent point (whichever comes first) is extracted. The cumulative extract constituting the extract subsystem is shown in Figure 5 and its bulk composition is recorded in Table 2.

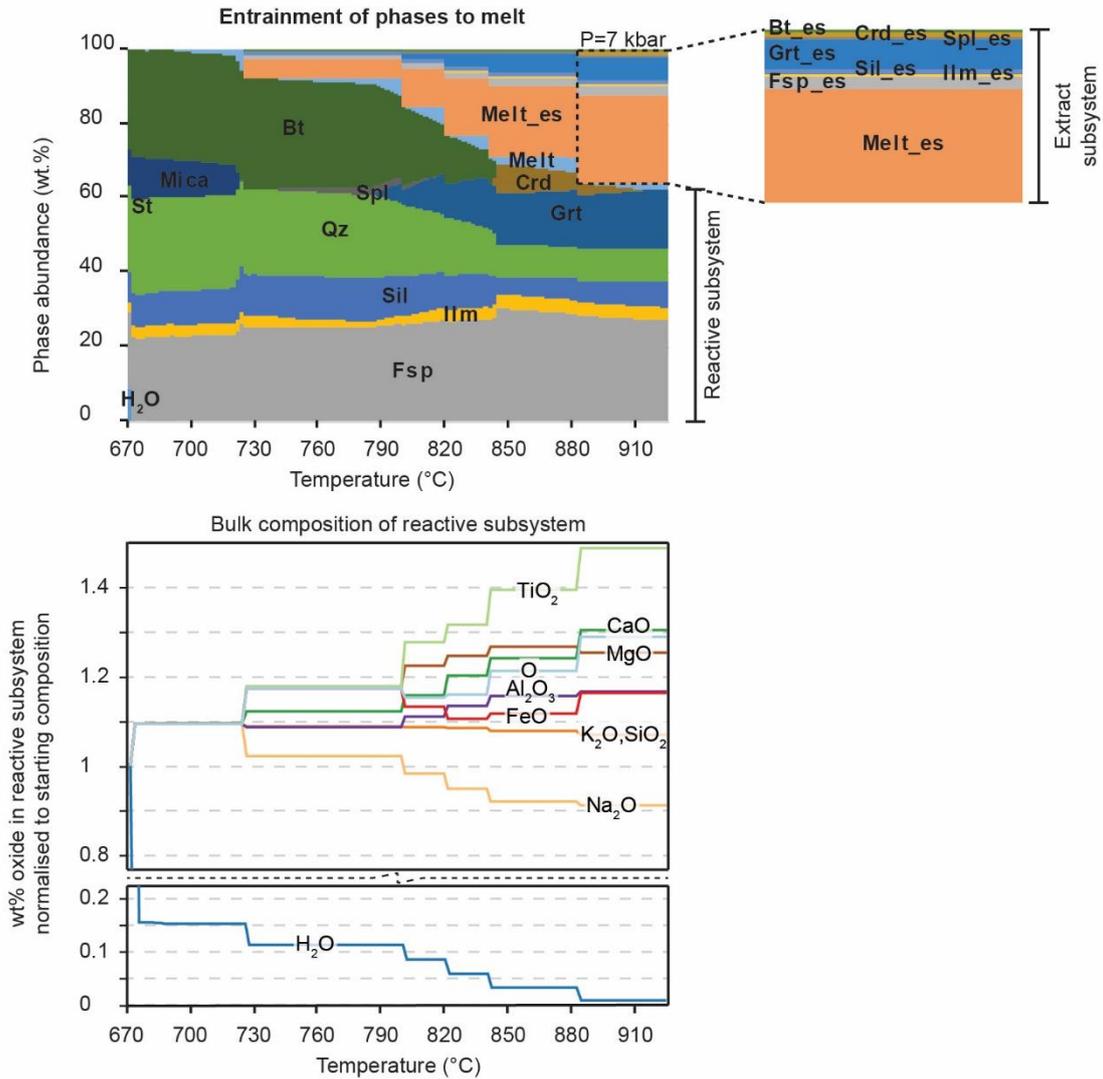


Fig. 5. Melt loss with entrainment of 30% of the increase in mass of all phases except melt between the current melt loss event and either the previous melt loss event or melt absent point (whichever is closer to the current event) for the isobaric heating path at 7 kbar from 670-920 °C. Melt threshold, handling of subsolidus water content and starting bulk composition as in Figure 4. The wt.% oxide in the bulk composition of the reactive subsystem is shown normalised to the starting bulk composition of the path.

| Cumulative extract subsystem formed in Figure 5 | | | | | | | | | | |
|---|------------------|------------------|--------------------------------|------|------|------|-------------------|------------------|------|------------------|
| | SiO ₂ | TiO ₂ | Al ₂ O ₃ | FeO | MgO | CaO | Na ₂ O | K ₂ O | O | H ₂ O |
| wt.% | 59.38 | 0.37 | 18.15 | 7.57 | 2.51 | 1.02 | 2.39 | 4.28 | 0.21 | 4.12 |

Table 2. Bulk composition of the cumulative extract subsystem formed by subsolidus water loss, supersolidus melt loss and entrainment described in Figure 5

Fractional crystallisation

Mechanisms of filter pressing or crystal settling may cause magma systems to fractionate, whereby a portion of crystallized phases are separated from the magma system (García-Arias and Stevens, 2017b). This loss of crystals changes the bulk composition of the magma and subsequent crystallization path followed by the remainder of the system. Depending on the mechanism involved, this process can be modelled by setting phase extraction routines to extract the crystals from the magma under several different circumstances. These extractions can be set relative to the cooling rate of the magma system or triggered when set criteria are met, such as when a threshold volume of crystals is exceeded (Bachmann and Bergantz, 2004; Stevens et al., 2017) or when the difference in density between a phase and melt exceeds a certain threshold. Figure 6 starts with the bulk composition of the cumulative extract system achieved in Figure 5 (Table 2) representing a magma emplaced at the same depth as extraction (7 kbar) which subsequently isobarically cools and crystallizes. To simulate a process of filter pressing, whenever, the total volume of solid phases (all phases excluding melt and H₂O) in the reactive subsystem exceeds 20 vol.%, 90% of all solid phases are extracted.

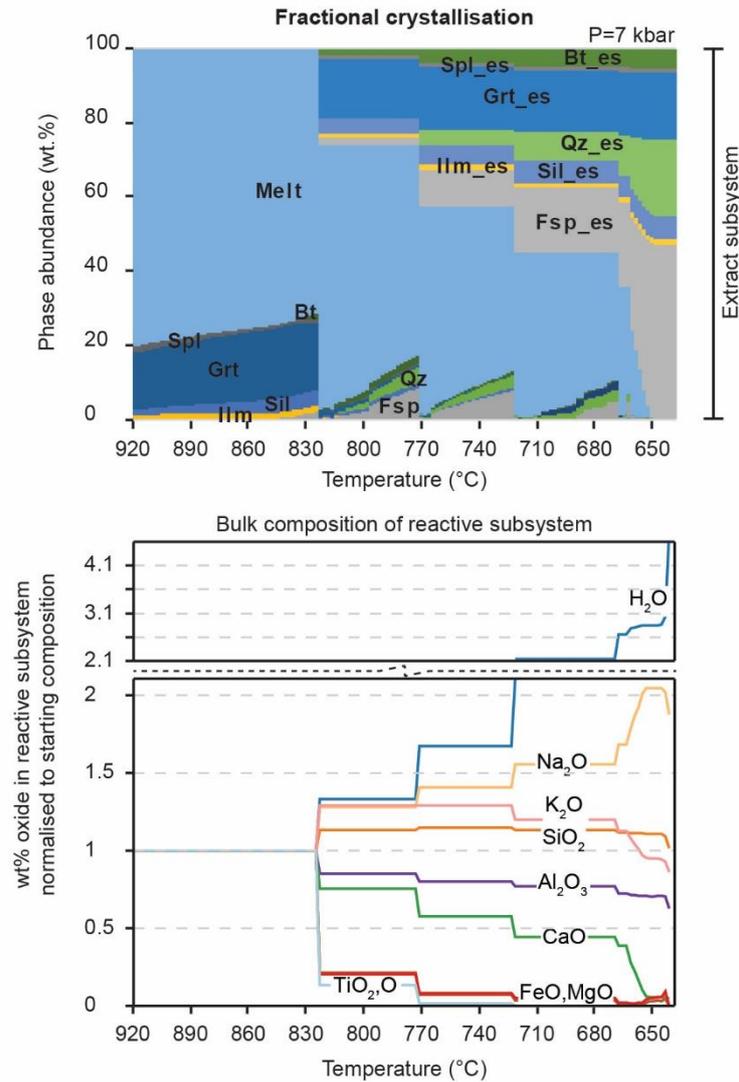


Fig. 6. Fractional crystallisation of the cumulative extract subsystem formed by Figure 5 (representing a magma source), 90% of all solid phases (all phases except melt and H₂O) are extracted whenever the cumulative solid phases exceed 20 vol.% of the reactive subsystem along the isobaric cooling path at 7 kbar from 920-640 °C. The wt.% oxide in the bulk composition of the reactive subsystem is shown normalised to the starting bulk composition of the path.

Limited availability of phases

The diffusion of elements within crystals such as plagioclase or garnet is extremely slow (Morse, 1984; Zuluaga et al., 2005; Johannes and Koepke, 2001; Tajčmanová et al., 2007). Within anatectic systems, a dissolution–reprecipitation mechanism may operate that is orders of magnitude faster than intra-crystalline diffusion and may represent the rate limiting control on the availability of components from such phases to participate in melting reactions (García-Arias and Stevens, 2017a; Johannes and Koepke, 2001; Taylor et al., 2014). The role of this behaviour in changing the bulk composition of anatectic systems can be approximated by dividing each crystal into an inner core, whose components are effectively isolated from contributing to the chemical equilibrium of the full system, and an outer rim that contains the chemically available components. With increasing time and temperature and decreasing crystal size, components formerly contained within the unreactive core become available to participate in reactions (Nicoli et al., 2017; Tajčmanová et al., 2007; Madlakana and Stevens, 2018). This process can be modelled by removing a portion of the bulk composition of the rock (the cores of grains) thereby creating an isolated subsystem. Portions of this isolated subsystem can then be transferred back into the reactive subsystem, for example as a function of P , T or melt volume (Stevens et al., 2017). Figure 7 shows the phase relations for the isobaric heating path at 7 kbar from 670 to 920°C where 95% of all feldspar present in the first step of the calculation is extracted and then added progressively back into the reactive subsystem. The mass of feldspar returned in each step is given as $0.01 \times (\text{melt}_{\text{vol}\%})^2$ until all feldspar is returned. Subsolidus water content is handled the same as in Figure 3.

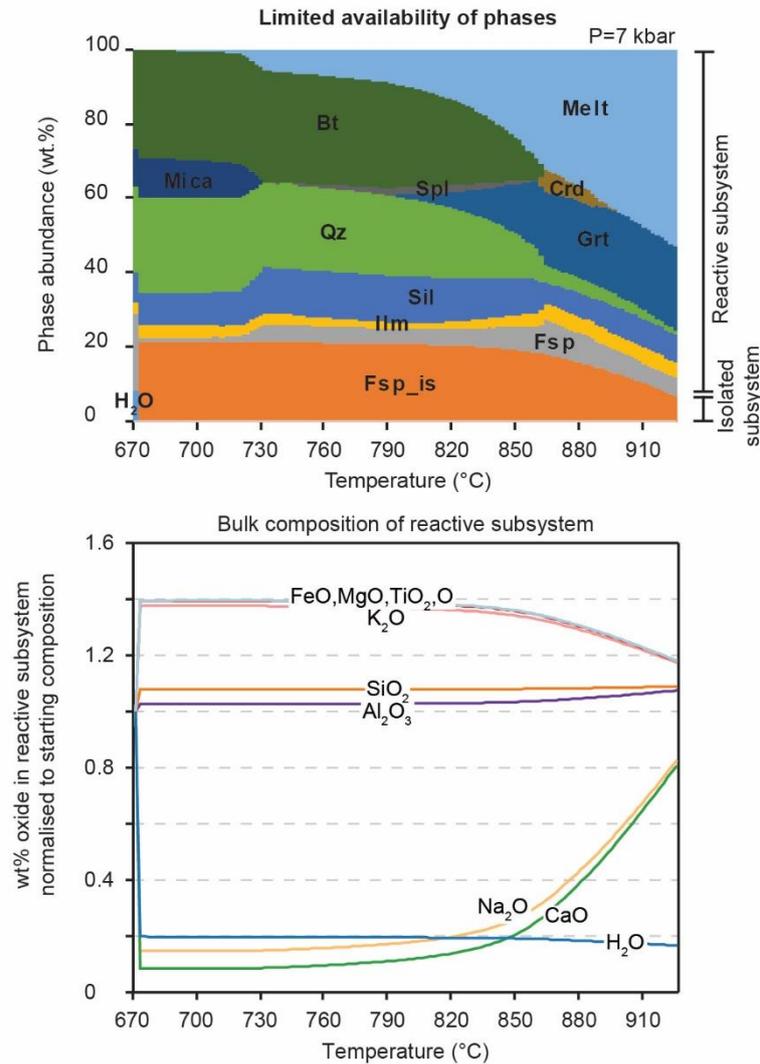


Fig. 7. Limited availability of feldspar modelled by extracting 95% of all feldspar occurring in the reactive subsystem at the start of the path and then progressively reintroduced to the reactive subsystem as a function of melt volume (where mass of feldspar integrated in each step = $0.01 \times (\text{melt}_{\text{vol.}\%})^2$) for the isobaric heating path at 7 kbar from 670-920 °C. Handling of subsolidus water content and starting bulk composition as in Figure 3. The wt.% oxide in the bulk composition of the reactive subsystem is shown normalised to the starting bulk composition of the path.

CONCLUSIONS

The presented methodology illustrates how process-oriented investigations can be performed by splitting the effective bulk composition of a system into a number of subsystems and then manipulating bulk compositional proportions between these systems. Rcrust's unique ability to assign bulk compositions by dependence and dynamic phase manipulations allows path dependence to exist between points in modelling space. The consequence of this path dependence is the ability to model in simultaneously changing pressure, temperature and bulk compositional space. Path dependence restricts modelling results to be directional (along paths). This can be used as an advantage as this new form of modelling is suited for investigating processes that effect the progressive evolution of a rock system. By setting appropriate phase manipulations, processes of mass transfer can be simulated to gain insights into the controlling factors behind this chemical evolution. This style of modelling introduces a unique set of phase diagram sections in which any variable, including bulk composition may change independently within a calculation. I suggest terming these diagrams "phase assemblage maps" in order to distinguish the independence of their pressure, temperature and chemical variability from traditional *pseudosections* which are either isobaric, isothermal, isochemical or composite panel versions of these. This would prevent misinterpretations of these diagrams that do not take into account their path dependence and implications thereof. As a matter of convention to aid understanding of these diagrams I further suggest that complete descriptions of the modelled P - T paths and starting bulk composition should always accompany such diagrams to the point of reproducibility. Caution should be exercised when interpreting the accuracy of these diagrams as they are intrinsically limited by the accuracy of the thermodynamic datasets that they rely on as well as the accuracy of the activity–composition models which are used to predict variations in the thermodynamic properties of the solution phases (solid and liquid) as they change composition.

The methods presented in this paper exemplify the novel ability of path dependent P - T - X calculations to investigate processes of compositional change in a thermodynamically constrained manner. Future work could use the combination of phase manipulations and path dependence to investigate the mechanistic processes that affect the evolution of a rock by building models based on mass transfer between bulk compositions of subsystems. As these models become more sophisticated they could begin to consider crucial controls on phase equilibria such as kinetic obstacles to equilibration reactions (Nicoli et al., 2017;

Tajčmanová et al., 2007; Madlakana and Stevens, 2018) local bulk composition effects (Lanari and Engi, 2017) or the effect of local stress on equilibration volumes (Wheeler, 2018, 2014). Further, component partitioning between phases could be used to provide estimates of trace element and isotope compositions (Yakymchuk et al., 2018; Gardiner et al., 2018). These additional considerations come with their own uncertainties that the reader must be mindful of. Incrementally building on these methodologies opens up new lines of investigation into the studying of petrogenetic processes.

ACKNOWLEDGEMENTS

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF. Funding by the French Embassy of South Africa to M.Mayne and by the South African Research Chairs Initiative (SARChI) to G. Stevens is gratefully acknowledged. The authors thank Gautier Nicoli and an anonymous reviewer for their helpful comments on the manuscript

SUPPLEMENTARY MATERIAL*

Files for reproducing figures 3-7 are contained in the zipped folder “Mayne_et_al_2018_1_modelling_files”. To open these download Rcrust from <https://www.sun.ac.za/english/faculty/science/earthsciences/rcrust> and replace the “Projects” and “data” folders with those in the zipped folder. Open each project by double clicking the appropriate x.RData file.

*For the thesis examiners’ convenience these files are already copied into the electronic copy of Rcrust provided in Addendum A, to open them simply install Rcrust following the user manual in Addendum B, open the “Projects” directory of Rcrust, open the project of choice and double click on the .RData file contained in the project file

REFERENCES

- Ague, J.J., 1991, Evidence for major mass transfer and volume strain during regional metamorphism of pelites: *Geology*, v. 19, p. 855–858.
- Bachmann, O., and Bergantz, G.W., 2004, On the origin of crystal-poor rhyolites: Extracted from batholithic crystal mushes: *Journal of Petrology*, v. 45, p. 1565–1582, doi: 10.1093/petrology/egh019.
- Brown, M., and Korhonen, F.J., 2009, Some Remarks on Melting and Extreme Metamorphism of Crustal Rocks, *in* *Physics and Chemistry of the Earth's Interior*, p. 67–87.
- De Capitani, C., and Petrakakis, K., 2010, The computation of equilibrium assemblage diagrams with Theriak/Domino software: *American Mineralogist*, v. 95, p. 1006–1016, doi: 10.2138/am.2010.3354.
- Connolly, J.A.D., 2005, Computation of phase equilibria by linear programming: A tool for geodynamic modeling and its application to subduction zone decarbonation: *Earth and Planetary Science Letters*, v. 236, p. 524–541, doi: 10.1016/j.epsl.2005.04.033.
- Connolly, J.A.D., 2009, The geodynamic equation of state: What and how: *Geochemistry, Geophysics, Geosystems*, v. 10, p. 1–19, doi: 10.1029/2009GC002540.
- Connolly, J.A.D., and Kerrick, D.M., 1987, An algorithm and computer program for calculating composition phase diagrams: *Calphad*, v. 11, p. 1–55, doi: 10.1016/0364-5916(87)90018-6.
- Diener, J.F.A., and Fagereng, Å., 2014, The influence of melting and melt drainage on crustal rheology during orogenesis: *Journal of Geophysical Research : Solid Earth*, v. 119, p. 6193–6210, doi: 10.1002/2014JB011088.Received.
- Farina, F., Stevens, G., Dini, A., and Schaltegger, U., 2017, Small-scale zircon age variability in the Monte Capanne pluton revealing the timescale of crustal melting and melt extraction, *in* *Goldschmidt 2017 Abstract*,.
- García-Arias, M., and Stevens, G., 2017a, Phase equilibrium modelling of granite magma petrogenesis : A . An evaluation of the magma compositions produced by crystal entrainment in the source: *Lithos*, v. 277, p. 131–153, doi: 10.1016/j.lithos.2016.09.028.
- García-Arias, M., and Stevens, G., 2017b, Phase equilibrium modelling of granite magma petrogenesis : B . An evaluation of the magma compositions that result from fractional crystallization: *Lithos*, v. 277, p. 109–130, doi: 10.1016/j.lithos.2016.09.027.
- Gardiner, N.J., Johnson, T.E., Kirkland, C.L., and Smithies, R.H., 2018, Melting controls on the lutetium–hafnium evolution of Archaean crust: *Precambrian Research*, v. 305, p. 479–488, doi: 10.1016/j.precamres.2017.12.026.
- Ghiorso, M.S., and Sack, R.O., 1995, Chemical mass transfer in magmatic processes IV . A revised and internally consistent thermodynamic model for the interpolation and extrapolation of liquid–solid equilibria in magmatic systems at elevated temperatures and pressures: *Contributions to Mineralogy and Petrology*, v. 119, p. 197–212.
- Holland, T., and Powell, R., 2003, Activity–compositions relations for phases in petrological calculations: An asymmetric multicomponent formulation: *Contributions to Mineralogy and Petrology*, v. 145, p. 492–501, doi: 10.1007/s00410-003-0464-z.
- Holland, T.J.B., and Powell, R., 2011, An improved and extended internally consistent thermodynamic dataset for phases of petrological interest, involving a new equation of state for solids: *Journal of Metamorphic Geology*, v. 29, p. 333–383, doi: 10.1111/j.1525-1314.2010.00923.x.
- Holland, T.J.B., and Powell, R., 1996, Thermodynamics of Order-Disorder in Minerals. II. Symmetric Formulism Applied to Solid Solutions: *American Mineralogist*, v. 81, p. 1425–1437.
- Johannes, W., and Koepke, J., 2001, Incomplete reaction of plagioclase in experimental dehydration melting of amphibolite: *Australian Journal of Earth Sciences*, v. 48, p. 581–590, doi: 10.1046/j.1440-0952.2001.00876.x.
- Johnson, T., Gibson, R.L., Brown, M., Buick, I., and Cartwright, I., 2003, Partial Melting of Metapelitic Rocks Beneath the Bushveld Complex , South Africa: *Journal of Petrology*, v. 44, p. 301–314, doi: 10.1093/petrology/44.5.789.
- Johnson, T.E., White, R.W., and Brown, M., 2011, A year in the life of an aluminous metapelite xenolith —

- The role of heating rates , reaction overstep , H₂O retention and melt loss: *Lithos*, v. 124, p. 132–143, doi: 10.1016/j.lithos.2010.08.009.
- Karpov, I.K., Chudnenko, K.V., and Kulik, D.A., 1997, Modeling chemical mass transfer in geochemical processes; thermodynamic relations, conditions of equilibria and numerical algorithms: *American Journal of Science*, v. 297, p. 767–806.
- Korhonen, F.J., Powell, R., and Stout, J.H., 2012, Stability of sapphirine + quartz in the oxidized rocks of the Wilson Lake terrane, Labrador: Calculated equilibria in NCKFMASHTO: *Journal of Metamorphic Geology*, v. 30, p. 21–36, doi: 10.1111/j.1525-1314.2011.00954.x.
- Lanari, P., and Engi, M., 2017, Local Bulk Composition Effects on Metamorphic Mineral Assemblages: *Reviews in Mineralogy and Geochemistry*, v. 83, p. 55–102, doi: 10.2138/rmg.2017.83.3.
- Madlakana, N., and Stevens, G., 2018, Plagioclase disequilibrium induced during fluid-absent biotite-breakdown melting in metapelites: *Journal of Metamorphic Geology*, doi: 10.1111/jmg.12429.
- Mayne, M.J., Moyen, J.F., Stevens, G., and Kaislaniemi, L., 2016, Rcrust: a tool for calculating path-dependent open system processes and application to melt loss: *Journal of Metamorphic Geology*, v. 34, p. 663–682, doi: 10.1111/jmg.12199.
- Mayne, M.J., Stevens, G., Moyen, J.F., and Johnson, T.E., 2017, The shape of the P-T path controls the onset of crustal anatexis, *in* Goldschmidt 2017 Abstract.
- Morrissey, L.J., Hand, M., Lane, K., Kelsey, D.E., and Dutch, R.A., 2016, Upgrading iron-ore deposits by melt loss during granulite facies metamorphism: *Ore Geology Reviews*, v. 74, p. 101–121, doi: 10.1016/j.oregeorev.2015.11.012.
- Morse, S.A., 1984, Cation Diffusion in Plagioclase Feldspar: *Science*, v. 225, p. 504–505.
- Nicoli, G., and Dyck, B., 2018, Exploring the metamorphic consequences of secular change in the siliciclastic compositions of continental margins: *Geoscience Frontiers*, p. 1–9, doi: 10.1016/j.gsf.2017.12.009.
- Nicoli, G., Stevens, G., Moyen, J.F., Vezinet, A., and Mayne, M.J., 2017, Insights into the complexity of crustal differentiation: K₂O-poor leucosomes within metasedimentary migmatites from the Southern Marginal Zone of the Limpopo Belt, South Africa.: *Journal of Metamorphic Geology*, doi: 10.1111/jmg.12265.
- Powell, R., and Holland, T., 1988, An internally consistent dataset with uncertainties and correlations: 3. Applications to geobarometry, worked examples and a computer program: *Journal of Metamorphic Geology*, v. 6, p. 173–204, doi: 10.1111/j.1525-1314.1988.tb00415.x.
- Powell, R., Holland, T., and Worley, B., 1998, Calculating phase diagrams involving solid solutions via non-linear equations, with examples using THERMOCALC: *Journal of Metamorphic Geology*, v. 16, p. 577–588, doi: 10.1111/j.1525-1314.1998.00157.x.
- R Core Team, 2016, R: A Language and Environment for Statistical Computing., <https://www.r-project.org/>.
- Riesco, M., Stüwe, K., Reche, J., and Martinez, F.J., 2004, Silica depleted melting of pelites. Petrogenetic grid and application to the Susqueda aureole, Spain: *Journal of Metamorphic Geology*, v. 22, p. 475–494, doi: 10.1111/j.1525-1314.2004.00527.x.
- Rosenberg, C.L., and Handy, M.R., 2005, Experimental deformation of partially melted granite revisited: Implications for the continental crust: *Journal of Metamorphic Geology*, v. 23, p. 19–28, doi: 10.1111/j.1525-1314.2005.00555.x.
- Spear, F.S., and Pattison, D.R.M., 2017, The implications of overstepping for metamorphic assemblage diagrams (MADs): *Chemical Geology*, v. 457, p. 38–46, doi: 10.1016/j.chemgeo.2017.03.011.
- Stevens, G., Clemens, J.D., and Mayne, M.J., 2017, The origins of geochemical trends in felsic igneous rocks : Insight into processes in the source., *in* Goldschmidt 2017 Abstract.
- Stevens, G., Villaros, A., and Moyen, J.F., 2007, Selective peritectic garnet entainment as the origin of geochemical diversity in S-type granites: *Geology*, v. 35, p. 9–12, doi: 10.1130/G22959A.1.
- Tajčmanová, L., Konopásek, J., and Connolly, J.A.D., 2007, Diffusion-controlled development of silica-undersaturated domains in felsic granulites of the Bohemian Massif (Variscan belt of Central Europe): *Contributions to Mineralogy and Petrology*, v. 153, p. 237–250, doi: 10.1007/s00410-006-0143-y.
- Taylor, J., Nicoli, G., Stevens, G., Frei, D., and Moyen, J., 2014, The processes that control leucosome

- compositions in metasedimentary granulites : perspectives from the Southern Marginal Zone migmatites , Limpopo Belt , South Africa: *Journal of Metamorphic Geology*, v. 32, p. 713–742, doi: 10.1111/jmg.12087.
- Vigneresse, J.L., and Burg, J.P., 2000, Continuous vs. discontinuous melt segregation in migmatites: Insights from a cellular automaton model: *Terra Nova*, v. 12, p. 188–192, doi: 10.1046/j.1365-3121.2000.00299.x.
- Wheeler, J., 2014, Dramatic effects of stress on metamorphic reactions: *Geology*, v. 42, p. 647–650, doi: 10.1130/G35718.1.
- Wheeler, J., 2018, The effects of stress on reactions in the Earth: Sometimes rather mean, usually normal, always important: *Journal of Metamorphic Geology*, v. 36, p. 439–461, doi: 10.1111/jmg.12299.
- White, R.W., and Powell, R., 2002, Melt loss and the preservation of granulite facies mineral assemblages: *Journal of Metamorphic Geology*, v. 20, p. 621–632, doi: 10.1046/j.1525-1314.2002.00206.x.
- White, R.W., and Powell, R., 2010, Retrograde melt-residue interaction and the formation of near-anhydrous leucosomes in migmatites: *Journal of Metamorphic Geology*, v. 28, p. 579–597, doi: 10.1111/j.1525-1314.2010.00881.x.
- White, R.W., Powell, R., and Clarke, G.L., 2002, The interpretation of reaction textures in Fe-rich metapelitic granulites of the Musgrave Block, Central Australia: Constraints from mineral equilibria calculations in the system: *Journal of Metamorphic Geology*, v. 20, p. 41–55, doi: 10.1046/j.0263-4929.2001.00349.x.
- White, R.W., Powell, R., Holland, T.J.B., Johnson, T.E., and Green, E.C.R., 2014, New mineral activity-composition relations for thermodynamic calculations in metapelitic systems: *Journal of Metamorphic Geology*, v. 32, doi: 10.1111/jmg.12071.
- White, R.W., Powell, R., Holland, T.J.B., and Worley, B., 2000, The effect of TiO₂ and Fe₂O₃ on metapelitic assemblages at greenschist and amphibolite facies conditions: mineral equilibria calculations in the system K₂O-FeO-MgO-Al₂O₃-SiO₂-H₂O-TiO₂-Fe₂O₃: *Journal of Metamorphic Geology*, v. 18, p. 497–511.
- White, R.W., Powell, R., and Phillips, G.N., 2003, A mineral equilibria study of the hydrothermal alteration in mafic greenschist facies rocks at Kalgoorlie, Western Australia: *Journal of Metamorphic Geology*, v. 21, p. 455–468, doi: 10.1046/j.1525-1314.2003.00454.x.
- Whitney, D.L., and Evans, B.W., 2010, Abbreviations for names of rock-forming minerals: *American Mineralogist*, v. 95, p. 185–187, doi: 10.2138/am.2010.3371.
- Yakymchuk, C., 2017, Applying phase equilibria modelling to metamorphic and geological processes: Recent developments and future potential: *Geoscience Canada*, v. 44, p. 27–46, doi: 10.12789/geocanj.2017.44.114.
- Yakymchuk, C., and Brown, M., 2014, Consequences of open-system melting in tectonics: *Journal of the Geological Society*, v. 171, p. 21–40, doi: 10.1144/jgs2013-039.
- Yakymchuk, C., Kirkland, C.L., and Clark, C., 2018, Th/U ratios in metamorphic zircon: *Journal of Metamorphic Geology*, v. 00, p. 1–23, doi: <https://doi.org/10.1111/jmg.12307>.
- Zuluaga, C.A., Stowell, H.H., and Tinkham, D.K., 2005, The effect of zoned garnet on metapelite pseudosection topology and calculated metamorphic P-T paths: *American Mineralogist*, v. 90, p. 1619–1628, doi: 10.2138/am.2005.1741.

CHAPTER 3

PRESENTATION OF RESEARCH PAPER 2: THE TRAJECTORY OF THE P - T PATH CONTROLS THE ONSET OF MELTING IN METASEDIMENTARY ROCKS.

This paper, first authored by Matthew Mayne, is under review for publication in the Journal of Metamorphic Geology.

The following aspects of the research were done independently by Matthew Mayne while receiving standard supervision by his supervisors Gary Stevens, Jean-François Moyen and Tim Johnson: (i) writing of programming code; (ii) conducting phase equilibrium modelling; (iii) generation of the figures; (v) writing of the manuscript

The trajectory of the P–T path controls the onset of melting in metasedimentary rocks.

Matthew Jason Mayne ^{1,2,*}, **Gary Stevens** ¹, **Jean-François Moyen** ² & **Tim Johnson** ³

¹University of Stellenbosch, Department of Earth Sciences, Private Bag X1, 7602

Matieland, South Africa

²Université de Lyon, Laboratoire Magmas et Volcans, UJM-UCA-CNRS-IRD, 42023 Saint-

Etienne, France

³Curtin University, Department of Applied Geology, WA 6845 Perth, Australia

*Corresponding author (e-mail: mmayne@sun.ac.za)

Authors' ORCiDs:

Matthew Jason Mayne, 0000-0002-2103-9647;

Gary Stevens, 0000-0003-1593-9419;

Jean-François Moyen, 0000-0002-0065-2442

Tim Johnson 0000-0001-8704-4396

Abbreviated title: “The *P–T* path controls the onset of melting”

ABSTRACT

The partial melting behaviour of rocks is critically dependant on the pressure–temperature (P – T) location of the solidus. As rocks lose water during their subsolidus evolution, the higher the temperature at which the solidus is encountered, the lower the water content available to drive future anatexis of any given rock. Consequently, understanding the location of the solidus in P – T space is critical to understanding the metamorphic evolution of anatectic rocks and their fertility for the production of granitic melt. Despite wide application of these techniques, conventional phase equilibrium modelling that is based on a single or a limited number of “stitched” bulk compositions cannot handle bulk-rock water content realistically. The reason for this is that the water content of any fully-hydrated rock changes as a function of pressure and temperature. This study uses Rcrust, a new software tool that allows calculation of phase equilibria in systems with a continually changing bulk composition, to investigate the partial melting of an average metapelite composition. Three different fluid states are considered: fluid-saturated with an abundant H_2O -rich fluid; fluid-absent, which has no fluid phase but begins calculations as fully-hydrated; and fluid-restricted which has a restricted quantity of free water (0.1 vol.%) filling pore spaces. The behaviour of the system under all three fluid states was investigated along a variety of linear prograde P – T paths that culminate in granulite and eclogite facies P – T conditions. Three fundamental aspects of partial melting behaviour were revealed that cannot be determined without the ability to continuously vary the bulk composition: (1) the fluid-absent solidus and the wet solidus are identical for all but steep P – T paths; (2) the bulk water content at the solidus of an average metapelite varies substantially as a function of pressure; (3) rocks following steep P – T paths ($\Delta P/\Delta T > 6$ kbar/100 °C) will melt at significantly higher temperature than the wet solidus, because the increase in total water contained in the hydrous silicates as a function of pressure exceeds the decrease in total water contained in the hydrous silicates as a function of temperature, resulting in the rock becoming water deficient, i.e. not fully-hydrated.

Keywords: Rcrust, anatexis; bulk water content; pressure-temperature paths; wet solidus

INTRODUCTION

The beginning of melting in the continental crust is controlled by pressure, temperature and bulk composition. Bulk compositions with sufficient H₂O to contain a discrete fluid phase will melt at the appropriate fluid-saturated solidus e.g. (Thompson, 1982; Spear et al., 1999; Johannes and Koepke, 2001), whilst rocks with insufficient water to allow a fluid phase to form are considered to melt at the lowest temperature fluid-absent incongruent melting reaction that the metamorphic assemblage allows e.g. (Stevens and Clemens, 1993). For aluminous metapelites this will commonly be the fluid-absent incongruent melting reaction involving muscovite e.g. (Huang and Wyllie, 1981), which is located at significantly higher temperature than the pelite wet solidus. The fluid-absent solidus for biotite-bearing rocks without muscovite is located at even higher temperatures within the granulite facies (Vielzeuf and Montel, 1994; Stevens et al., 1995). Thus, there is general consensus that the presence or absence of water in the subsolidus assemblage exerts the strongest control on the temperature at which partial melting begins in common crustal rocks e.g. (Waters, 1988; Weinberg and Hasalová, 2015; Clemens et al., 2016). However, the onset of fluid-absent melting is extremely difficult to define accurately.

Experimentally, the fluid-absent solidus is hard to identify because of the slow kinetics of the melting reactions, particularly those involving plagioclase e.g. (Johannes et al., 1994; Johannes and Koepke, 2001), and the difficulty of identifying small fractions of quenched melt in experimental run products. An additional challenge is the unavoidable inclusion of gas into the capsule (normally air), that must lower aH₂O on the grain boundaries of the charge. Crystallization experiments suffer similar kinetic problems and are likely to systematically underestimate solidus temperatures.

Phase equilibria studies also have difficulty in investigating the onset of anatexis in fluid-absent systems across a range of pressures because established software for phase equilibria modelling have limited abilities for the automated handling of bulk compositional changes within pressure–temperature space. Most studies that assume fluid-absent conditions of anatexis determine bulk rock H₂O content using isobaric T – $X(H_2O)$ sections and set bulk H₂O content at the maximum value possible before the appearance of melt without the presence of a free water phase. This commonly results in the first melt being produced through incongruent melting reactions that involve breakdown of micas, typically in the temperature range of 700 to 850 °C for mid– to lower crustal pressures (Vielzeuf and Holloway, 1988; Stevens et al., 1997; Brown and Korhonen, 2009; Yakymchuk and Brown,

2014). Other studies make the assumption that there is a small amount of H₂O available in pore spaces leading to minor H₂O-saturated melting at the wet solidus, and may also use isobaric $T-X(H_2O)$ sections or modes of hydrous minerals coupled with stoichiometric amount of H₂O in these minerals (Indares et al., 2008) to determine appropriate bulk rock H₂O content at a specific pressure. However, if one considers the variations in assemblage and mineral mode inherent to any relevant pseudosection, it is clear that the water content of the subsolidus mineral assemblage varies as a function of both pressure and temperature. Fixed-composition pseudosections can therefore only constitute an accurate representation of the behaviour of the system at the specific pressure used to constrain the H₂O content of the system; all pressures below and above that value must either over- or underestimate the H₂O content (Webb et al., 2015). These inaccuracies in bulk H₂O content are likely to cause significant discrepancies in melting behaviour as a function of pressure in pseudosections constructed in this way. Similarly, it is impossible for an experimental investigation of fully-hydrated, fluid-absent or fluid-restricted partial melting to be conducted at a range of pressures using the same starting material because the correct water content for defining this state is unique to a specific pressure. This study presents a new method for constraining the H₂O content of the subsolidus system through progressive water loss during prograde metamorphism. This method is used to investigate the position of the solidus in pressure-temperature space, the bulk rock water content at the solidus as a function of pressure and the influence of the trajectory of the $P-T$ path on the melting behaviour of such rocks under upper amphibolite, eclogite and granulite facies conditions for a single pelite bulk composition.

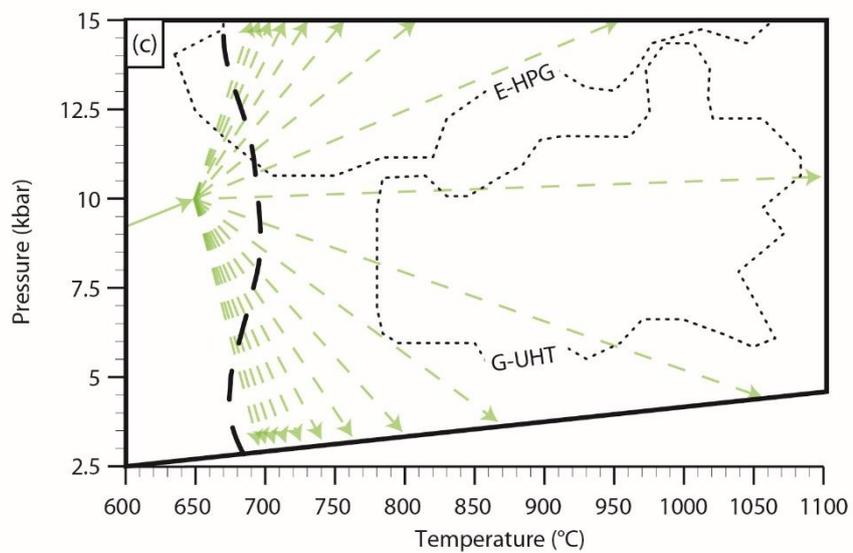
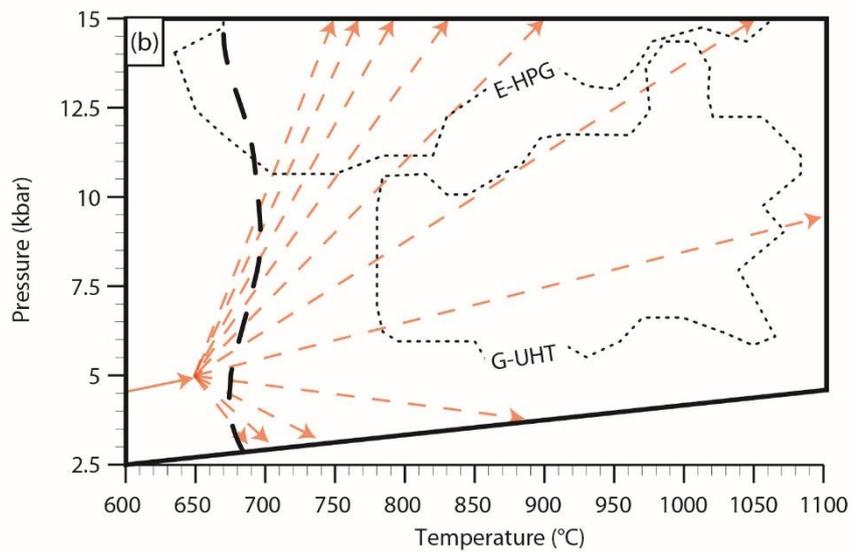
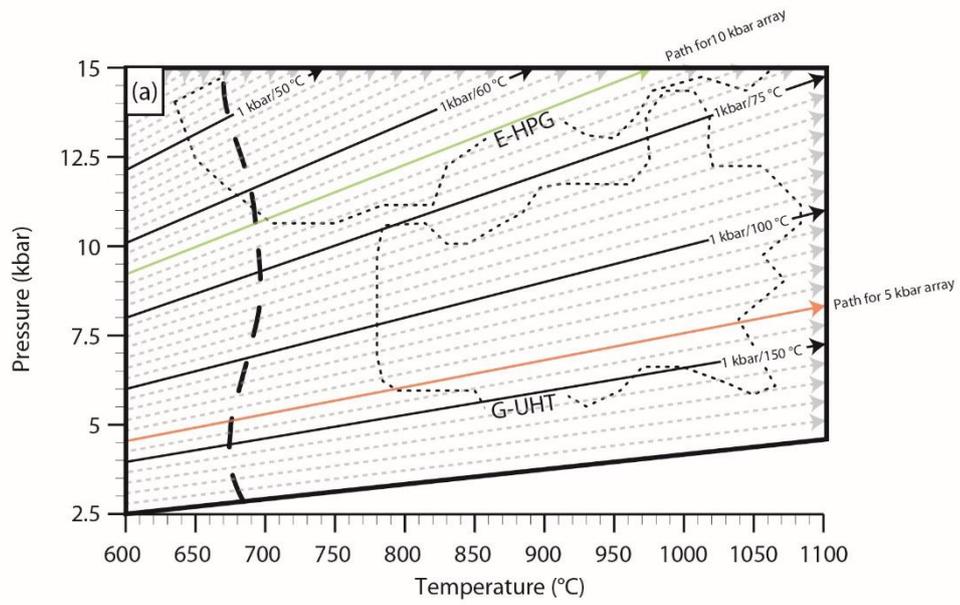
METHODOLOGY

The phase equilibria tool Rcrust (Mayne et al., 2016) allows modelling to be conducted along $P-T$ paths with bulk composition as a variable. It does this by breaking each $P-T$ path into a number of points in $P-T$ space. Each point passes on any bulk compositional changes induced by modelling functions to the next point on its respective path. Since these points are calculated in a progressive order, they have path dependent bulk compositions.

In order to systematically evaluate the effect of the steepness of the $P-T$ path on the partial melting behaviour of the system three different fanned arrays of linear prograde $P-T$ paths were modelled, one projecting a fanned array from the $P-T$ origin (0 °C; 0 kbar) (Fig. 1a), one starting at the origin and then following a $P-T$ path till a fanned projection from 650 °C

and 5 kbar (Fig. 1b) and one starting at the origin and then following a $P-T$ path till a fanned projection from 650 °C and 10 kbar (Fig. 1c). This produced $P-T$ paths that arrive at the same amphibolite, granulite and eclogite facies $P-T$ conditions by travelling along different trajectories through $P-T$ space.

The first set of paths start at the $P-T$ origin (0 °C; 0 kbar) and increment pressure (in kbar) as $\sin((y_{i-1}+6)*\pi/180)*10/140*(x_{i-1})$ and temperature (in °C) as $\cos((y_{i-1}+6)*\pi/180)*280/140*(x_{i-1})$ where y_i increments the dP/dT of the individual paths in the set and x_i increments the steps along each respective path. From this fanned array of paths two points were chosen (650 °C;5 kbar and 650 °C;10 kbar) from which to project inflections of the $P-T$ path. The bulk compositions achieved at these points by following the original $P-T$ array from the origin were used as the initial compositions for two further fanned arrays projecting from these points. This ensures that bulk compositional changes from the $P-T$ origin array are inherited by the paths originating at 650 °C;5 kbar and 650 °C;10 kbar.



- — — Fluid-saturated solidus
- E-HPG -- Eclogite-high-pressure granulite
- G-UHT -- Granulite-ultrahigh temperature
- — — 0 kbar array
- — — 5 kbar array
- — — 10 kbar array

Fig. 1. P – T diagrams illustrating the modelled path arrays and their intersections with the fields of eclogite–high–pressure granulite (E–HPG) and granulite–ultrahigh temperature (G–UHT) peak metamorphic conditions, as defined by (Brown, 2014). The 0 kbar array (grey dashed arrows) maintains a range of relatively flat P – T slopes from the origin across the modelled P – T space. The 5 kbar array (red lines) and 10 kbar array (green lines) begin with the bulk composition attained by the relative path on the 0 kbar array at 650 °C; 5 kbar (path for 5 kbar array) and 650 °C; 10 kbar (path for 10 kbar array) respectively. These arrays then propagate a range of changed $\Delta P/\Delta T$ gradients while modelling continues to remove water to ensure a fully-hydrated, fluid-absent, or fluid-restricted subsolidus assemblage. The position of the fluid-saturated solidus (calculated in this study) is shown as a thick dashed line for reference.

Modelling was conducted in three different fluid modes (Table 1), fluid-saturated, fluid-absent and fluid-restricted with 0.1 vol.% water (approximating a water filled pore space). In the fluid-saturated mode, H_2O content in the bulk rock was set to 17 wt.%, which produced a water-bearing assemblage under all P – T conditions investigated. In fluid-absent mode, phase assemblages were calculated for each P – T array with the fluid-saturated composition at the origin but with the software set to remove any free water and recalculate the composition at each step along the path. This maintains the subsolidus assemblage in a fully-hydrated, fluid-absent state, as long as subsolidus reactions encountered by prograde P – T paths do not consume H_2O . In the fluid restricted mode exactly the same procedure was followed but with the software set to retain water sufficient to fill a nominal pore space, here taken as 0.1 vol.% of the rock, which likely represents a maximum value for high-grade metamorphic rocks e.g.(Thompson, 1983; Stevens and Clemens, 1993; Yardley, 2009; Brown and Korhonen, 2009; Yakymchuk and Brown, 2014). In fluid-absent or fluid restricted mode, if subsolidus reactions encountered by prograde P – T paths consume H_2O the system may become fluid-absent as water is not added back in to the system.

Phase equilibrium modelling was performed on an average amphibolite–facies metapelite composition (Table 2) from (Ague, 1991) in the NCKFMASHTO (Na_2O – CaO – K_2O – FeO – MgO – Al_2O_3 – SiO_2 – H_2O – TiO_2 – O) chemical system. Calculations were performed using Rcrust (version 2017–02–14) (Mayne et al., 2016) based on a compiled form of *meemum* from Perplex 6.7.3 (Connolly, 2009) and the thermodynamic data file *hpl1ver.dat* (Holland and Powell, 2011). Activity–composition (a – x) models from *solution_model_673.dat* were

used as follows: Bi(W), Crd(W), Gt(W), Ilm(W), melt(W), Mica(W), St(W) and Opx(W) from the internally consistent set of solution models produced by (White et al., 2014) for biotite, cordierite, garnet, ilmenite, melt, white mica, staurolite and orthopyroxene respectively; Fsp(C1) for feldspars (Holland and Powell, 2003); Ep(HP11) for epidote (Holland and Powell, 2011) and Sp(WPC) for spinel (White et al., 2002).

| | Fluid-saturated | Fluid-absent | Fluid restricted |
|---|---|---|---|
| Starting bulk H ₂ O content (wt.%) | 17 | 17 | 17 |
| Phase extraction definition | None | Remove all water present in each step | Remove all except 0.1 vol.% of water present in each step |
| Fluid state | Water-bearing assemblage under all <i>P-T</i> conditions investigated | Subsolidus assemblage is fluid-absent. Subsolidus assemblage is fully-hydrated as long as subsolidus reactions encountered by prograde <i>P-T</i> paths do not consume H ₂ O | Subsolidus assemblage begins as water present, if subsolidus reactions encountered by prograde <i>P-T</i> paths consume H ₂ O the system may become fluid-absent |

Table 1. Modelling modes for handling of bulk H₂O content in this study as fluid-saturated, fluid-absent and fluid restricted.

| | Na ₂ O | MgO | Al ₂ O ₃ | SiO ₂ | K ₂ O | CaO | TiO ₂ | FeO | O | H ₂ O* |
|--|-------------------|------|--------------------------------|------------------|------------------|------|------------------|------|------|-------------------|
| wt.% | 1.55 | 2.78 | 17.36 | 48.14 | 3.47 | 1.32 | 0.89 | 7.22 | 0.26 | 17.00 |
| <i>Note:</i> Bulk composition taken from (Ague, 1991) as an average amphibolite-facies pelite. | | | | | | | | | | |
| *H ₂ O begins in excess (17 wt.%). | | | | | | | | | | |

Table 2. Starting bulk composition for modelling as the average amphibolite-facies pelite from Ague (1991) after H₂O adjustment to ensure water begins in excess (17 wt.%).

RESULTS

Melting reactions as a function of fluid mode

Phase relations for the path marked “Path for 5 kbar array” (Fig. 1a) are shown in Figure 2. This system begins at the P - T origin with a bulk H₂O content of 17 wt.%. In fluid-saturated mode this ensures a water-bearing assemblage for the entire length of the path with water abundance in the reactive subsystem beginning at ~ 15 wt.% rising to 16 wt.% at the solidus and then falling progressively during melting to 8 wt.% as the melt volume grows (Fig. 2a). In this system melting begins with a reaction involving white mica at 680 °C producing ~ 15 wt.% melt followed by melting involving biotite until 880 °C producing a further ~ 55 wt.% melt and finally melting involving cordierite producing ~ 13 wt.% melt, culminating in a total of 83 wt.% melt by 970 °C (Fig. 2d). When comparing this system to the other fluid modes it helps to note that the renormalized abundance of melt in the reactive subsystem excluding the abundance of water equates to approximately 17, 60 and 14 wt.% melt, respectively, producing a total of 91 wt.% melt in the reactive subsystem excluding the abundance of water. Further, note that in order to achieve this high degree of melting the average pelite composition used became quartz deficient by approximately 750°C.

In fluid-absent mode there is no free water in the reactive subsystem (Fig. 2b). Water is extracted whenever it forms as a free phase (Fig. 2g) which occurs intermittently below the solidus culminating in just above 15.7 wt.% water extracted cumulatively by the solidus at 682°C. No further water is formed or extracted above the solidus. Melting begins at 682 °C producing less than 1 wt.% melt up till 688 °C, after which white mica is exhausted producing ~ 5 wt.% melt (Fig. 2b). From 688 °C onwards melting involving biotite produces ~ 17 wt.% melt by 825 °C after which melting involving cordierite produces a further ~ 25 wt.% melt by 910 °C and finally melting involving feldspar and quartz produces a further ~ 19 wt.% melt by the end of the path. This results in a total of ~ 66 wt.% melt by 1000 °C. Quartz is depleted by 940°C.

Modelling in the fluid restricted mode produced short intervals along the path with a water filled porosity resulting in <0.025 wt.% water in the reactive subsystem. This equates with the maximum allowable 0.1 vol.% pore space set in the modelling. The water extraction events follow a similar pattern to those in the fluid absent mode resulting in a cumulative 15.7 wt.% water extracted by the solidus at 680 °C. Overall, the partial melting behaviour is near identical to that observed in the fluid absent, mode with only a 2 °C lower solidus and slightly more melt produced before the exhaustion of white mica (also less than 1 wt.% by 688 °C) (Fig. 2c).

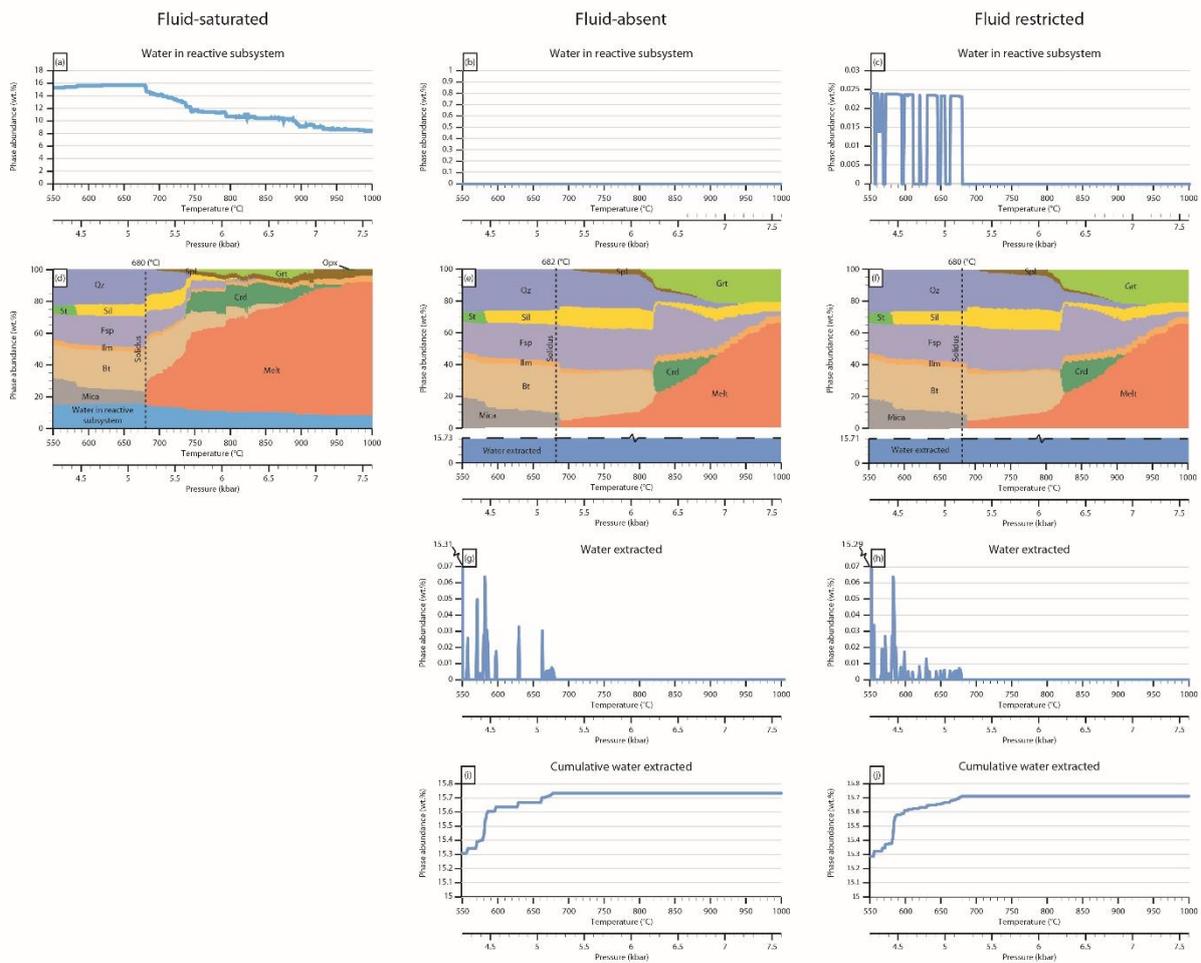


Fig. 2. Diagrams portraying the partial melting behaviour along the “Path for 5 kbar array” path shown in Figure 1a for the three fluid modes considered: (a-c) Water in the reactive subsystem (wt.%) as a function of temperature shows the difference between the three modelled fluid modes, i.e. fluid-saturated, fluid-absent and fluid restricted. Note the variation in scale for the y-axis. (d-f) Phase abundance as weight percentage of the reactive subsystem. (g-h) water extracted from the reactive subsystem as weight percentage of the full system. (i-j) cumulative water extracted from the reactive subsystem as weight percentage of the full system. Abbreviations for phases in figures except for melt, water and mica were taken from (Whitney and Evans, 2010) as Bt, Biotite; Crd, cordierite; Fsp, plagioclase and alkali-feldspar; Grt, garnet; Ilm, ilmenite; Opx, orthopyroxene; Qz, quartz; Sil, sillimanite; Spl, spinel; St, staurolite.

Effect of fluid state and P - T trajectory on the position of the solidus

The solidi produced by the 3 fluid modes in combination with the path array originating at 0 °C;0 kbar, are all within 2 °C of each other despite the inclusion of a fluid-absent rock in the modelling (Fig. 3). This is in agreement with the findings of (Webb et al., 2015) who demonstrated using isobaric T - X_{H_2O} sections, that subsolidus water loss from rocks undergoing prograde metamorphism migrates the rock composition towards the minimum saturation point on the boundary of the wet-melting field. This study follows the same approach but in P - T space because bulk composition may vary simultaneously with P and T . The findings indicate that fully-hydrated fluid absent rocks do melt at the wet solidus, even in fluid-absent systems. In order to assess the potential effect of the P - T path trajectory on partial melting behaviour, each of the 3 fluid modes described above are modelled along the 3 P - T path arrays described in Figure 1. The results indicate that, for the fluid saturated case, the wet solidus is unaffected by the trajectory of the prograde P - T path; the solidus is located at ~ 680 °C between 2.5 and 15 kbar (Fig. 3a), irrespective of the path taken by the rock to reach the solidus. The position of this calculated solidus is in broad agreement with the results of experimental studies and other phase equilibrium investigations using a conventional pseudosection approach (Thompson, 1982; Spear et al., 1999; Johannes and Koepke, 2001). This behaviour contrasts strongly with the solidus behaviour for the two modes where water content is handled in a more petrologically realistic manner. In both the fluid-absent and fluid-restricted modes, the results in terms of phase assemblage and positions of the path dependent solidi were identical, so only the fluid-absent system is shown and discussed further. For these systems, the three modelled path arrays each produced a different solidus (Fig. 3e). The path array originating at 0 °C;0 kbar, underwent melting at the wet solidus as described above (Fig. 3b). However, evolution along the P - T path arrays originating at 5 and 10 kbar and 650 °C produced very different behaviour. Both these arrays contain P - T paths that are substantially steeper than those of the array originating at 0 kbar (Fig.3 c & d) and the fluid-absent and fluid-restricted rocks evolving along paths with $\Delta P/\Delta T < 6$ kbar/100 °C melted at the wet solidus. However, for both systems, rocks evolving along steeper paths do not melt at the wet solidus, rather they map out a solidus which shifts to progressively higher temperature as pressure increases. For the array originating at 5 kbar this results in the solidus at 15 kbar being located at a temperature > 100 °C higher than the equivalent 0 kbar array's solidus. The portion of the solidus for the 5 kbar path arrays that is located at a temperature above the fluid-saturated solidus is located

within ~ 25 °C of the modelled white mica out boundary (Fig. 4c). Consequently, the steep P – T paths that produce this higher temperature solidus appear to record the fluid-absent incongruent melting reaction involving white mica at pressures above approximately 7 kbar. For both the fluid-restricted and fluid absent systems, the path array originating at 10 kbar records similar behaviour to the equivalent 5 kbar path array, matching the fluid-saturated solidus at lower pressure, and departing from it at higher pressure, but in this case the point of departure is located at ~ 11.5 kbar (Fig. 3d) because it is only above this pressure that the paths attain the prerequisite steepness. At 15 kbar the 10 kbar array is located >50 °C higher than the 0 kbar path array solidus.

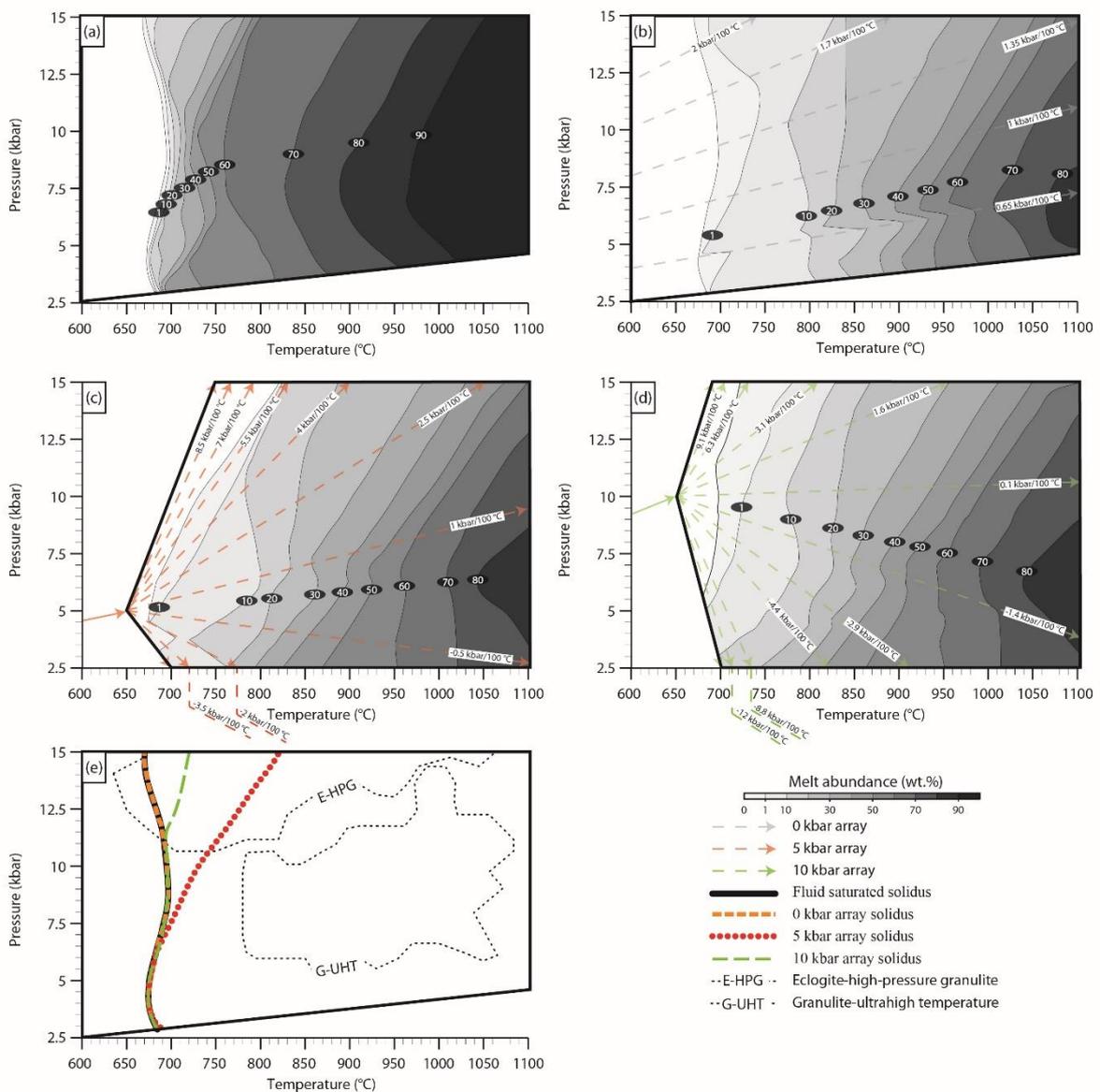


Fig. 3. Melt abundance as weight percentage of the reactive subsystem (renormalized to 100% after excluding the abundance of water) for a) the fluid-saturated system; b) the fluid-

absent system originating at the P – T origin; c) the fluid-absent system beginning at the P – T origin but inflecting to a fanned array at 650 °C, 5 kbar; d) the fluid-absent system beginning at the P – T origin but inflecting to a fanned array at 650 °C, 10 kbar. e) Relative positions of the solidi produced by the fluid-saturated system and the fluid-absent system from the three arrays considered with respect to the fields of eclogite–high–pressure granulite (E–HPG) and granulite–ultrahigh temperature (G–UHT) peak metamorphic conditions, as defined by (Brown, 2014)

Effect of fluid state and P – T trajectory on the melt production

Fluid saturated melting produces a large amount of melt close to the solidus with a tight grouping of melt wt.% contours near the solidus (calculated as melt abundance renormalized to 100% after excluding the abundance of water) (Fig. 3a). The system consists of up to 60 wt.% melt by 800 °C (Fig. 3a). For paths originating at the origin in the fluid-absent and fluid-restricted systems, the initial amount of melt produced is low with the 1 wt.% melt contour being separated from the solidus by 15 to >50 °C (Fig. 3b). A systematic increase in melt volume with temperature is observed in these paths above 800 °C with an average increase of approximately 10 wt.% melt per 30 °C.

Between approximately 3 and 4.5 kbar the solidus intersects a P – T field in which biotite is the only hydrous mineral, with white mica and cordierite present as an additional hydrous phase at higher and lower pressures respectively (Fig. 4 c- e). The H₂O content of the rock along this portion of the solidus is constant at ~ 1 wt.% (Fig. 4b), indicating that the biotite mode and water content do not vary as a function of pressure in this area of the phase diagram. The consequences of this are visible as a bulge to higher temperatures in the melt isopleths (up to 30 °C) for paths that travelled through the pressure–temperature field of this assemblage (Fig. 3b). A maximum melt abundance of 80 wt.% is obtained for this system at low pressures (<8.5 kbar).

Melt abundance contours for paths originating at 650 °C, 5kbar in fluid-absent and fluid restricted systems show similar features to those originating at the origin. Contours are strongly temperature dependent, show systematic increases above 800 °C and are deflected to higher temperature at high pressures. A maximum of 80 wt.% melt is produced for pressures between 3. and 8.5 kbar at 1 100 °C. Below 800 °C, however, marked differences are found with the solidus, the 1, and the 10 wt.% melt contour all deflected at high pressure to near coincidence at 825 °C, 15 kbar. Additionally, the anomalous shift to higher

temperatures experienced by the 0.65 kbar/100 °C path from the origin is erased by the inflection of the $P-T$ paths at 650 °C, 5kbar and an increase in temperature of contours is observed only for paths inflecting to a steepness less than -2 kbar/100 °C (Fig. 3c). Melt production for the paths originating at 650 °C, 10kbar in fluid-absent and fluid restricted systems are near identical to those produced by paths originating at 650 °C, 5kbar with the only significant differences being less deflected contours below 800 °C; at high pressure and a lack of any anomalous increase in temperature of contours for steeply downwards inflecting paths (Fig. 3d).

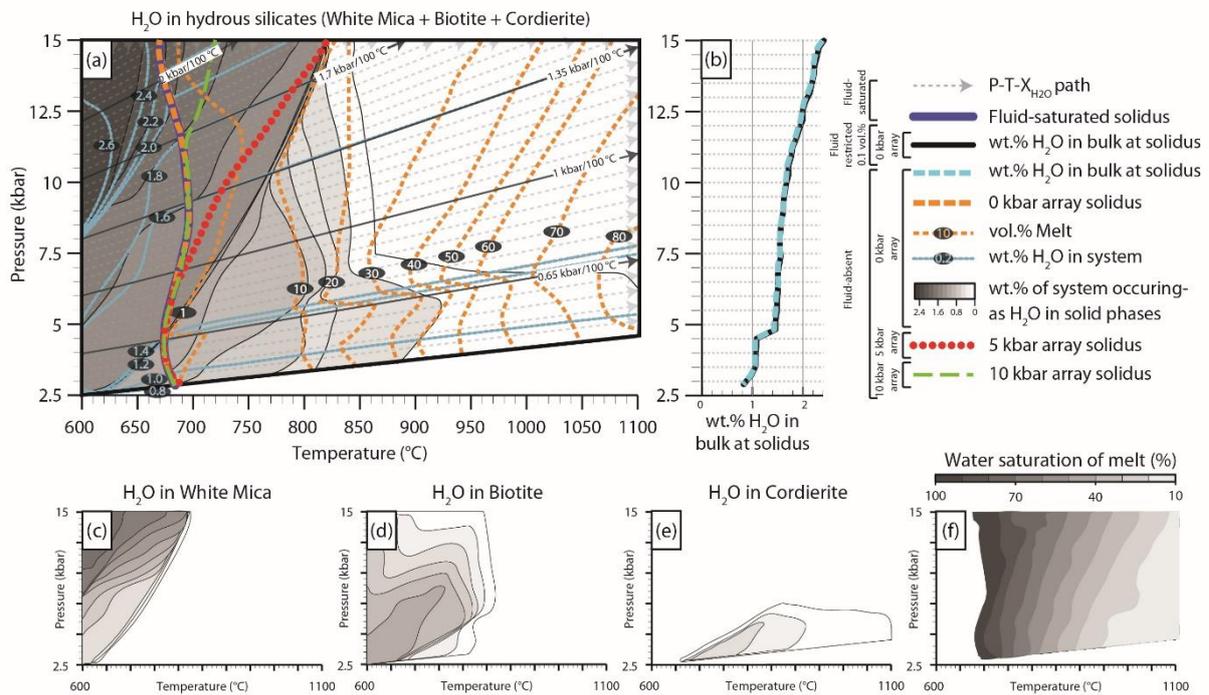


Fig. 4. $P-T$ diagram showing the $P-T$ path array originating at the $P-T$ origin (grey dashed arrows). a) The grey shading represents the total H₂O (as wt.% of the system) contained within the hydrous silicates white mica, biotite and cordierite, contoured from 0 (white) to 2.6 (dark grey) for the fluid absent system. Weight percentage of H₂O in the entire system (including melt) is contoured in 0.2 wt.% increments (blue dotted lines) and the weight percentage of melt in the reactive subsystem is contoured (orange dashed lines). b) Weight percentage of H₂O in the bulk system at the solidus for the fluid-absent and the fluid-restricted systems as a function of pressure. (c-e) Weight proportion of the solid system contained as H₂O in white mica, biotite and cordierite. f) Degree of water saturation of melt given by the H₂O in melt in the fluid-absent mode divided by the H₂O in melt in fluid-saturated mode, expressed as a percentage.

Rock water content at the solidus as a function of pressure

The H₂O content of the fluid-absent system at the solidus produced by paths from the origin varies by 1.6 wt.% (from 0.8 to 2.4 wt.%) (Fig. 4b) over the pressure range investigated and generally increases as a function of pressure. This substantial range in water content clearly illustrates the deficiencies in trying to model anatexis over a pressure range using a single bulk composition. The water content of the fluid restricted system at this solidus is only minutely higher than the fluid-absent rock at the same pressure (Fig.4b). For both fluid-absent and fluid restricted systems above the solidus, bulk H₂O content remains constant along each path as no supersolidus free water can form due to the presence of water-undersaturated melt (Fig. 4f). Above the solidus all H₂O is accommodated either in melt or within melt and hydrous silicates (Fig. 4a). The hydrous silicates that are stable within this bulk composition in the relevant P - T range are white mica (Fig. 4c), biotite (Fig. 4d) and cordierite (Fig. 4e). With increasing pressure, the stability of the hydrous silicates accommodates larger total water content (up to 2.6 wt.% of the reactive subsystem) by increasing the phase abundance of the hydrous silicates and by preferentially forming hydrous silicates that contain more water (e.g. white mica rather than biotite) (Fig. 4a, c-e). If P - T paths are steeper than the H₂O in hydrous silicate contours in Figure 4a (for example paths with $\Delta P/\Delta T > 6$ kbar/100 °C in Figure 3c) then the abundance of white mica may increase and the abundance of biotite may decrease to such an extent that the rock system becomes fluid-deficient. If this happens the system may cross the position of the fluid-saturated solidus without melting and only begin melting at the first fluid absent melting reaction the path intersects

DISCUSSION AND CONCLUSIONS

The methodology presented allows phase equilibrium modelling to investigate the P – T path dependent partial melting of rocks with realistic H_2O contents, over a range of pressures and temperatures. The results demonstrate that for rocks undergoing heating without a substantial pressure increase, all metapelitic rocks will melt at the wet solidus, irrespective of fluid state. The reason for this is that in the fluid-absent system, biotite H_2O content decreases as a function of temperature (Fig. 4d). Once the rock reaches the position of the wet solidus, the decrease in biotite H_2O content with the next increment of temperature increase allows for melt formation without the appearance of a fluid-phase. This liberation of water from biotite can also be seen in the water extraction behaviour of single paths in the fluid-absent and fluid restricted systems (Fig. 2 g,h). This is almost congruent fluid-absent anatexis as it is accompanied by only minor shifts in biotite and plagioclase compositions and it results in low melt volumes. Thus, phase equilibrium modelling using the best available models for activity–composition relations in the solid– and liquid–solution phases, in combination with a correct handling of H_2O , demonstrates that the idea that fluid-absent rocks will only melt at the experimentally determined conditions of muscovite– and biotite–driven incongruent melting, is incorrect. As also detailed by (Webb et al., 2015), the main reasons why experimental studies have missed this behaviour are that experiments using a single composition cannot properly map out fluid-absent partial melting behaviour over a range of pressures, or the natural decrease in water content that occurs as subsolidus rocks evolve to higher temperature. The modelling predicts that melting at conditions at or close to the wet solidus is inevitable for most rocks. Possible examples of field areas where evidence of this type of low temperature anatexis are preserved are the Damara Belt of Namibia (Ward et al., 2008) and in the Famatinian Orogen of NW Argentina (Sola et al., 2017). With no melt loss, biotite bearing assemblages appear to melt progressively between the fluid-absent solidus and the point at which biotite disappears (Fig. 4a).

At common granulite facies temperatures, melt proportion appears to be a rather well defined function of temperature, with surprisingly little pressure influence at 800 °C the system contains ~ 10 wt.% melt, at 850 °C this figure is ~ 20 wt.%. Maximum melt proportions are attained in the fluid-saturated system at 90 wt.% with more than 60 wt.% of this melt being produced below 850 °C. The fluid-absent and fluid restricted system produces significantly less melt (with a maximum of 80 wt.%) only producing melt above 1 wt.% after 750 °C with a systematic increase in melt above 800 °C at an average of 10 wt.%

melt per 30 °C. Importantly, the melts present at relatively low temperatures are significantly water under-saturated (Fig. 4f) and consequently can ascend to achieve crustal differentiation if they are able to segregate from their sources. This implies that fluid-absent melting does make a substantial contribution to low-temperature melting (< 850 C) and that these melts have significant ascent potential due to their water-undersaturated character. Being able to adjust system water content incrementally to simulate fully-hydrated, fluid-absent conditions and fluid-restricted, water-present conditions over a range of pressures also allows the path dependency of partial melting behaviour to be investigated. This has demonstrated that if metapelitic rocks cross the position of the wet solidus in the field of white mica stability along a sufficiently steep positive P - T path, the solidus is displaced to significantly higher temperature than the fluid-saturated solidus. This appears to be a result of white mica mode growing at the expense of biotite as a function of pressure, as the rock progresses along the prograde path (Fig. 4 c,d). In the fluid-restricted case this results in the subsolidus rock consuming the 0.1 vol.% pore water through phengite producing reactions and the rock becoming fluid-absent before the water-saturated solidus is encountered (Fig. 4a). Under these conditions, the rocks in both fluid-absent and fluid restricted systems are less than fully-hydrated and the resultant decrease in water activity favours white mica stability over melt and melting is delayed to higher temperature. This phenomenon could affect a large number of rocks which undergo prograde evolution along steep P - T paths into the high pressure amphibolite facies and eclogite facies. This could provide a mechanism for such rocks to remain solid for a considerable time along these steep paths crossing the wet solidus after which shallowing of the P - T path and/or subsequent heating would produce substantial melt volumes (Fig. 3 c,d).

DOWNLOAD

Rcrust is free to download and use under the GNU copyleft. The latest version of the software can be found on either of the two institutional addresses at: <http://www.univ-st-etienne.fr/rcrust> or <http://www.sun.ac.za/english/faculty/science/earthsciences/rcrust>.

Alternatively, the software can be downloaded by contacting the corresponding author.

ACKNOWLEDGMENTS

The financial assistance of the National Research Foundation (NRF) is gratefully acknowledged, as is funding by the French Embassy of South Africa to M. Mayne and by the SARChI initiative to G. Stevens. The authors would like to thank Frank Spear, Dave Kelsey and an anonymous reviewer for their comments on an earlier form of this

manuscript. Special thanks are given to Chris Yakymchuk for identifying the requirement of path dependent bulk H₂O contents for phase equilibria modelling in P - T space during his review of Mayne et al (2016). Your comment led us to produce the methodology presented in this paper as a possible solution.

SUPPLEMENTARY MATERIAL*

Files for reproducing the results presented are contained in the zipped folder “Mayne_et_al_2018_2_modelling_files”. PA0, PA5 and PA10 denote the path arrays originating from the P - T origin, from 650 °C; 5 kbar and from 650 °C; 10 kbar respectively. Fluid saturated calculations were performed using the PA0 array and fluid states of fluid-absent and fluid-restricted are shown for each of the possible path arrays. A process of gridded interpolation was performed on each array in order to plot a square regular gridded contour diagram, these files are presented with a suffix of “_grid” added to their project name. To open the files download Rcrust from <https://www.sun.ac.za/english/faculty/science/earthsciences/rcrust> and replace the “Projects” and “data” folders with those in the zipped folder. Open each project by double clicking the appropriate x.RData file.

*For the thesis examiners’ convenience these files are already copied into the electronic copy of Rcrust provided in Addendum A, to open them simply install Rcrust following the user manual in Addendum B, open the “Projects” directory of Rcrust, open the project of choice and double click on the .RData file contained in the project file

REFERENCES

- Ague, J.J., 1991, Evidence for major mass transfer and volume strain during regional metamorphism of pelites: *Geology*, v. 19, p. 855–858.
- Brown, M., 2014, The contribution of metamorphic petrology to understanding lithosphere evolution and geodynamics: *Geoscience Frontiers*, v. 5, p. 553–569.
- Brown, M., and Korhonen, F.J., 2009, Some Remarks on Melting and Extreme Metamorphism of Crustal Rocks, *in* *Physics and Chemistry of the Earth's Interior*, p. 67–87.
- Clemens, J.D., Buick, I.S., and Stevens, G., 2016, Fluids, melting, granulites and granites: A commentary: *Precambrian Research*, v. 278, p. 394–399, doi: 10.1016/j.precamres.2016.01.001.
- Connolly, J.A.D., 2009, The geodynamic equation of state: What and how: *Geochemistry, Geophysics, Geosystems*, v. 10, p. 1–19, doi: 10.1029/2009GC002540.
- Holland, T., and Powell, R., 2003, Activity-compositions relations for phases in petrological calculations: An asymmetric multicomponent formulation: *Contributions to Mineralogy and Petrology*, v. 145, p. 492–501, doi: 10.1007/s00410-003-0464-z.
- Holland, T.J.B., and Powell, R., 2011, An improved and extended internally consistent thermodynamic dataset for phases of petrological interest, involving a new equation of state for solids: *Journal of Metamorphic Geology*, v. 29, p. 333–383, doi: 10.1111/j.1525-1314.2010.00923.x.
- Huang, W.L., and Wyllie, P.J., 1981, Phase Relationships of S-Type Granite with H₂O to 35 kbar : Muscovite Granite From Harney Peak , South Dakota: *Journal of Geophysical Research*, v. 86, p. 10515–10529.
- Indares, A., White, R.W., and Powell, R., 2008, Phase equilibria modelling of kyanite-bearing anatectic paragneisses from the central Grenville Province: *Journal of Metamorphic Geology*, v. 26, p. 815–836, doi: 10.1111/j.1525-1314.2008.00788.x.
- Johannes, W., and Koepke, J., 2001, Incomplete reaction of plagioclase in experimental dehydration melting of amphibolite: *Australian Journal of Earth Sciences*, v. 48, p. 581–590, doi: 10.1046/j.1440-0952.2001.00876.x.
- Johannes, W., Koepke, J., and Behrens, H., 1994, Partial melting reactions of plagioclases and plagioclase-bearing systems, *in* *Feldspars and their Reactions*, Springer Netherlands, p. 161–194.
- Mayne, M.J., Moyen, J.F., Stevens, G., and Kaislaniemi, L., 2016, Rcrust: a tool for calculating path-dependent open system processes and application to melt loss: *Journal of Metamorphic Geology*, v. 34, p. 663–682, doi: 10.1111/jmg.12199.
- Sola, A.M., Hasalova, P., Weinberg, R.F., Suzano, N.O., Becchio, R.A., Hongn, F.D., and Botelho, N., 2017, Low-P melting of metapelitic rocks and the role of H₂O : Insights from phase equilibria modelling: *Journal of Metamorphic Geology*, v. 35, p. 1131–1159, doi: 10.1111/jmg.12279.
- Spear, F.S., Kohn, M.J., and Cheney, J.T., 1999, P -T paths from anatectic pelites: *Contributions to Mineralogy and Petrology*, v. 134, p. 17–32.
- Stevens, G., and Clemens, J.D., 1993, Fluid-absent melting and the roles of fluids in the lithosphere: a slanted summary? *Chemical Geology*, v. 108, p. 1–17.
- Stevens, G., Clemens, J.D., and Droop, G.T.R., 1995, Hydrous cordierite in granulites and crustal magma production: *Geology*, v. 23, p. 925–928, doi: 10.1130/0091-7613(1995)023<0925:HCIGAC>2.3.CO.
- Stevens, G., Clemens, and Droop, G.T.R., 1997, Melt production during granulite-facies anatexis: experimental data from “primitive” metasedimentary protoliths: *Contributions to Mineralogy and Petrology*, v. 128, p. 352–370.
- Thompson, A.B., 1982, Dehydration melting of pelitic rocks and the generation of H₂O-undersaturated granitic liquids: *American Journal of Science*, v. 282, p. 1567–1595.
- Thompson, A.B., 1983, Fluid-absent metamorphism: *Journal of the Geological Society, London*, v. 140, p. 533–547.
- Vielzeuf, D., and Holloway, J.R., 1988, Experimental determination of the fluid-absent melting relations in the pelitic system Consequences for crustal differentiation: *Contributions to Mineralogy and Petrology*, v. 98, p. 257–276, doi: 10.1007/BF00375178.

- Vielzeuf, D., and Montel, J.M., 1994, Partial melting of metagreywackes. Part I. Fluid-absent experiments and phase relationships: *Contributions to Mineralogy and Petrology*, v. 117, p. 375–393, doi: 10.1007/BF00307272.
- Ward, R.A., Stevens, G., and Kisters, A., 2008, Fluid and deformation induced partial melting and melt volumes in low-temperature granulite-facies metasediments, Damara Belt, Namibia: *Lithos*, v. 105, p. 253–271.
- Waters, D.J., 1988, Partial melting and the formation of granulite facies assemblages in Namaqualand, South Africa: *Journal of Metamorphic Geology*, v. 6, p. 387–404, doi: 10.1111/j.1525-1314.1988.tb00430.x.
- Webb, G., Powell, R., and McLaren, S., 2015, Phase equilibria constraints on the melt fertility of crustal rocks: The effect of subsolidus water loss: *Journal of Metamorphic Geology*, v. 33, p. 147–165, doi: 10.1111/jmg.12114.
- Weinberg, R.F., and Hasalová, P., 2015, Water-fluxed melting of the continental crust: A review: *Lithos*, v. 212–215, doi: 10.1016/j.lithos.2014.08.021.
- White, R.W., Powell, R., and Clarke, G.L., 2002, The interpretation of reaction textures in Fe-rich metapelitic granulites of the Musgrave Block, Central Australia: Constraints from mineral equilibria calculations in the system: *Journal of Metamorphic Geology*, v. 20, p. 41–55, doi: 10.1046/j.0263-4929.2001.00349.x.
- White, R.W., Powell, R., Holland, T.J.B., Johnson, T.E., and Green, E.C.R., 2014, New mineral activity-composition relations for thermodynamic calculations in metapelitic systems: *Journal of Metamorphic Geology*, v. 32, doi: 10.1111/jmg.12071.
- Whitney, D.L., and Evans, B.W., 2010, Abbreviations for names of rock-forming minerals: *American Mineralogist*, v. 95, p. 185–187, doi: 10.2138/am.2010.3371.
- Yakymchuk, C., and Brown, M., 2014, Consequences of open-system melting in tectonics: *Journal of the Geological Society*, v. 171, p. 21–40, doi: 10.1144/jgs2013-039.
- Yardley, B.W.D., 2009, The role of water in the evolution of the continental crust: *Journal of the Geological Society, London*, v. 166, p. 585–600, doi: 10.1144/0016-76492008-101.

CHAPTER 4

PRESENTATION OF RESEARCH PAPER 3: A PHASE EQUILIBRIUM INVESTIGATION OF SELECTED SOURCE CONTROLS ON THE COMPOSITION OF MELT BATCHES GENERATED BY SEQUENTIAL MELTING OF AN AVERAGE METAPELITE

This paper, first authored by Matthew Mayne, is accepted upon revision for publication in the Geological Society of London Special Issue: Making granites: petrogenetic processes, tectonic environments and secular variations.

The following aspects of the research were done independently by Matthew Mayne while receiving standard supervision by his supervisors Gary Stevens and Jean-François Moyen: (i) writing of programming code; (ii) conducting phase equilibrium modelling; (iii) generation of the figures; (v) writing of the manuscript

A phase equilibrium investigation of selected source controls on the composition of melt batches generated by sequential melting of an average metapelite

Matthew Jason Mayne ^{1,2,*}, **Gary Stevens** ¹, **Jean-François Moyen** ²

¹University of Stellenbosch, Department of Earth Sciences, Private Bag X1, 7602 Matieland, South Africa

²Université de Lyon, Laboratoire Magmas et Volcans, UJM-UCA-CNRS-IRD, 42023 Saint-Etienne, France

*Corresponding author (e-mail: mmayne@sun.ac.za)

Authors' ORCiDs:

Matthew Jason Mayne, 0000-0002-2103-9647;

Gary Stevens, 0000-0003-1593-9419;

Jean-François Moyen, 0000-0002-0065-2442

Abbreviated title: “Source controls on melting of a metapelite”

ABSTRACT

The ability of Rcrust (Mayne et al., 2016) to conduct path dependent phase equilibrium modelling with automated changing bulk compositions allows for a phase equilibrium approach to investigate an array of source controls for their effect on the bulk compositions of melts produced by sequential melting events. The following source controls of the rock system are considered: 1. Initial magnesium and iron content; 2. Initial sodium and calcium content; 3. fluid state; 4. Pressure-temperature path followed by the system and 5. threshold by which melt extractions in the system are triggered. The permutation of 3 cases for each of these source controls resulted in 243 different modelled pressure-temperature-bulk composition paths with which the melting of an average pelite composition is investigated. The resultant melt compositions are compared to that of a natural granite data set and provide a good fit for the incompatible elements Na_2O and K_2O with the allowance that granites most likely form as magmas consisting of melt and ferromagnesian rich crystals. The fluid state of the system is shown to have the strongest control on melt compositions with the pressure-temperature path having subordinate control on the volume and composition of melts produced.

Keywords: Phase equilibria modelling; Rcrust; source controls, fluid state, pressure-temperature path

INTRODUCTION

Granite plutons and batholiths (*sensu lato*) commonly exhibit a substantial range in major and trace element compositions, within co-magmatic rocks e.g. (Brown and Pressley, 1999; Stevens et al., 2007). Studies seeking to explain this commonly propose a range of different processes that occur in the magma plumbing systems or in magma chambers, and that are interpreted to modify magmas of originally homogenous composition. Processes such as unmixing of entrained residuum e.g. (Chappell et al., 1987), fractional crystallization e.g. (Foden et al., 2002; Breaks, Frederick and Moore, 1992), assimilation of wall rocks e.g. (Beard et al., 2005; DePaolo, 1981), or mixing with other magmas e.g. (Barbarin, 1988; Collins, 1996), all feature prominently in the literature, yet as discussed by (Clemens and Stevens, 2012), are unlikely to provide viable explanations for the major element, trace element and isotopic variations recorded within large granite bodies.

Some petrogenetic processes proposed to account for the observed range of magma compositions do occur in the source. For example, the variable degrees of entrainment of the peritectic assemblage proposed by Stevens et al. (2007), Clemens et al. (2011) and Villaros and Stevens (2009) to explain compositional variation in the S-type granites of the Cape Granite Suite. These studies propose that incompatible major element concentrations in granites reflect the compositions and proportions of reactant minerals in the source, whilst compatible element concentrations principally reflect the amount of peritectic assemblage entrained to the magma in the source, during biotite fluid-absent melting. When combined with efficient extraction of magma from the source as different batches that do not mingle in the plumbing systems e.g. (Brown and Pressley, 1999), the changes in melting reaction stoichiometry that occur with increasing temperature and decreasing water content are recognised to produce a range of magma compositions e.g. (Farina and Stevens, 2011). The water content of rocks, as well as the absence or availability of a water-rich fluid phase, both have a profound effect on melt composition and the temperature of melt generation. The role of water content of the system in determining melt productivity is also dependant on the shape of the P - T path, through the control of pressure on the solubility of water in melt. Despite the fact that source compositional variation, including water content, P - T condition of anatexis, and the volume of melt that is retained in the source prior to magma extraction, all clearly may have a profound effect on magma chemistry, relatively few studies recognise that these factors may play a role in shaping the compositional range recorded in granites. The majority of studies that have attempted to interpret the origins of major element

chemical variation within granites assume the involvement of large volumes of melt of homogenous composition. For even relatively small sized bodies this is a highly unrealistic assumption for two reasons. Firstly, even if the source rocks retain melt until a single large batch is segregated and delivered to construct the pluton, it is very unlikely that the source rocks, and thus the melt, will be compositionally homogenous. This conclusion is based on the observations that all rock sequences that may represent fertile sources of granite magma, such as hydrated volcanic sequences of mafic to intermediate composition, as well as clastic sediments with a considerable hydrous mineral fraction, are typically compositionally variable on a metre to sub-metre scale; additionally, the volume of source rocks must be significantly larger than the volume of the magmatic rocks produced. Thus, for even small plutons, melt must drain from a sufficiently large volume of source rocks that compositional variation in the source is unavoidable. Secondly, it is very unlikely that the source rocks deliver melt in a single large batch (Clemens et al., 2016), potentially facilitating homogenization by mixing and diffusion after segregation. Both exposed granulite facies source rocks and granites contain compositional and textural evidence to support incremental or continual drainage of melt from the source e.g. (Coleman et al., 2004; de Saint Blanquat et al., 2011). Examples of restitic granulites that have yielded at least 25 vol.% melt, contain compositional layering at a decimetre to metre scale that is interpreted to represent compositional layering within the protolith e.g. (Taylor and Stevens, 2010). Contacts between layers are sharp and mineral assemblages within different layers are not in equilibrium e.g. orthopyroxene- and sillimanite-bearing assemblages in adjacent layers (Stevens and Reenen, 1992). Such rocks must have lost melt episodically, in small batches, or continuously. If not, they would have developed diatexitic textures, with disruption of regular layering. Evidence from some S-type leucogranites indicates that melt batches are indeed very rapidly transported from source to pluton and do not remain in the source for any significant time after formation. These rocks have bulk compositions that are zircon undersaturated at reasonable estimates for magma temperature, yet contain zircon inherited from the source, supporting estimates of melt residency in the source of no more than 500 years (Villaros et al., 2009). Continuous or episodic melt extraction will result in different melt compositions leaving the source, as the melting reactions change as anatexis of the source progresses e.g. (Farina and Stevens, 2011).

From the arguments above, granites derived from partial melting of the crust can be assumed to be principally constructed from melt batches that had different compositions at the point of segregation from the source. Investigation of the role of source composition in

controlling melt compositions is hampered by the fact that almost all available data from experimental petrology represent the consequences of batch melting, mapped out over a range of pressures and temperatures. Typically, a single composition or a small number of different compositions is investigated e.g. (Stevens et al., 1997). The advent of Rcrust (Mayne et al., 2016), software which allows path dependant phase equilibrium modelling with changing bulk composition, allows the source control on the compositions of individual batches of melt leaving the source to be investigated using a phase equilibrium modelling approach. As there are an almost infinitely large number of potential variables that could be investigated. However, this study has aimed to investigate the control of Mg#, Na/Ca ratio and fluid state, as well as the effect of melt volume prior to melt extraction and the steepness of the P - T path on the compositions of melt batches produced from a metapelite undergoing progressive melting.

METHODOLOGY

The source parameters that were investigated in this study can be divided into compositional parameters and physical parameters. For the compositional parameters, the compositions used were based on the average pelite composition of Ague (1991) (Table 1). In order to approximate chemical heterogeneity in the metapelite source, the magnesium number ($Mg\#$) = $Mg/(Fe + Mg) * 100$ [mol.%] and calcium number ($Ca\#$) = $Ca/(Ca + Na) * 100$ [mol.%] of the bulk composition was varied. The $Mg\#$ of the Ague (1991) average is 20, two additional bulk compositions were generated by changing the atomic ratio of Mg to Fe to create $Mg\#$ = 41 and 60. Similarly, the $Ca\#$ for the Ague (1991) average is 19 and the atomic proportions of Ca and Na were varied for each of the 3 $Mg\#$ bulk compositions to produce additional bulk compositions with $Ca\#$ = 32 and 42. Thus, excluding variations in water content, 9 different bulk compositions were used: the permutation of 3 magnesium numbers and 3 calcium numbers. At conditions just below the water-present solidus, these rock compositions are all characterized by assemblages dominated by quartz, plagioclase, biotite and muscovite, and at a reference P - T condition of 640 °C and 7 kbar, the mineral modes and plagioclase anorthite contents are indicated in Figure 1. Note that the plagioclase anorthite contents are higher than the $Ca\#$'s of the rocks and also vary as a function of the $Mg\#$. This is because Na is taken up by both muscovite and biotite, but in different amounts, and the mode of biotite and muscovite vary as a function of $Mg\#$, whilst at this P - T condition, Ca is accommodated almost completely within plagioclase.

Water content was varied between three different fluid states (H_2O) across the 9 compositions described above. Firstly, each of the rocks was modelled as having 10 wt.% water in the bulk composition. This was sufficient to keep the assemblage water saturated to a temperature above 920 °C, the upper limit of the modelling conducted in this study. These compositions are termed water in excess. An additional set of compositions was modelled as fully-hydrated, fluid-absent. This was achieved by starting with the water in excess composition and setting the software to remove all water present as a free phase at each step of the prograde evolution of the rock. A final situation considered was for water to be porosity limited that is: water to be present at the water-saturated solidus only as a pore fluid filling a nominal porosity. In this case the software was set to start with water in excess and at each increment of prograde evolution to remove all free water except for 0.1 vol.%. The combination of the three different fluid states, with the variations in bulk rock $Mg\#$ and $Ca\#$, produces 27 different sets of compositional parameters that were modelled.

The 27 different sets of bulk compositional parameters were considered to evolve along three possible pressure-temperature paths (P - T). The first of these was an isobaric heating path at 7 kbar originating at 640 °C. The second, originating at the same point was a linear path to 920 °C and 9.5 kbar. The third, also originates at the same point and is a linear path to 920 °C and 12 kbar. In the discussion that follows these paths are referred to respectively as isobaric, gentle and steep. A second physical parameter considered in the modelling was the degree of melt retention in the source prior to melt extraction (*Melt threshold*). Values of 1, 7 and 10 vol.% melt were used as the trigger for melt extraction. Upon this threshold being exceeded, melt was extracted down to 1 vol.% which was assumed to represent the amount of melt retained on grain boundaries and therefore inaccessible to extraction. The combination of compositional parameters, with three P - T path scenarios and three *Melt thresholds* resulted in a total of 5 different source controls being investigated, each of which have 3 possible cases, culminating in 3^5 different models (Table 2).

| | SiO ₂ | TiO ₂ | Al ₂ O ₃ | FeO | MgO | CaO | Na ₂ O | K ₂ O | O | H ₂ O |
|------|------------------|------------------|--------------------------------|------|------|------|-------------------|------------------|------|------------------|
| wt.% | 52.15 | 0.96 | 18.81 | 9.59 | 1.34 | 0.93 | 2.18 | 3.76 | 0.28 | 10 |

Table 1. Initial bulk composition used in the construction of path dependent *P-T-X* paths in wt.% as the average amphibolite-facies pelite from Ague (1991). The composition shown forms the baseline for calculating the 9 starting bulk compositions shown in Figure 1 with magnesium number (*Mg#*) and calcium number (*Ca#*) altered where applicable.

| | | Source controls | | | | |
|-------|--------|---------------------------------|-------------------------------|--|---|--|
| | | Compositional parameters | | | Physical parameters | |
| | | Magnesium number (<i>Mg#</i>) | Calcium number (<i>Ca#</i>) | Fluid state (<i>H₂O</i>) | Pressure-temperature path (<i>P-T</i>) | Melt retention before extraction (<i>Melt threshold</i>) |
| Cases | Low | 20 | 19 | Fhfa: Fully-hydrated, fluid absent; water starting in excess but all liquid water present as a free phase removed from the reactive subsystem at each step | Isobaric: {640 °C;7 kbar}-> {920 °C;7 kbar} | 1 vol.% |
| | Medium | 41 | 32 | Porosity limited: water starting in excess but all liquid water exceeding 0.1 vol.% of the system removed at each step | Gentle: {640 °C;7 kbar}-> {920 °C;9.5 kbar} | 7 vol.% |
| | High | 60 | 42 | Water in excess: 10 wt.% H ₂ O; water present as a free phase throughout | Steep: {640 °C;7 kbar}-> {920 °C;12 kbar} | 10 vol.% |

Table 2. Summary of the *source controls* considered and each of their 3 *cases*.

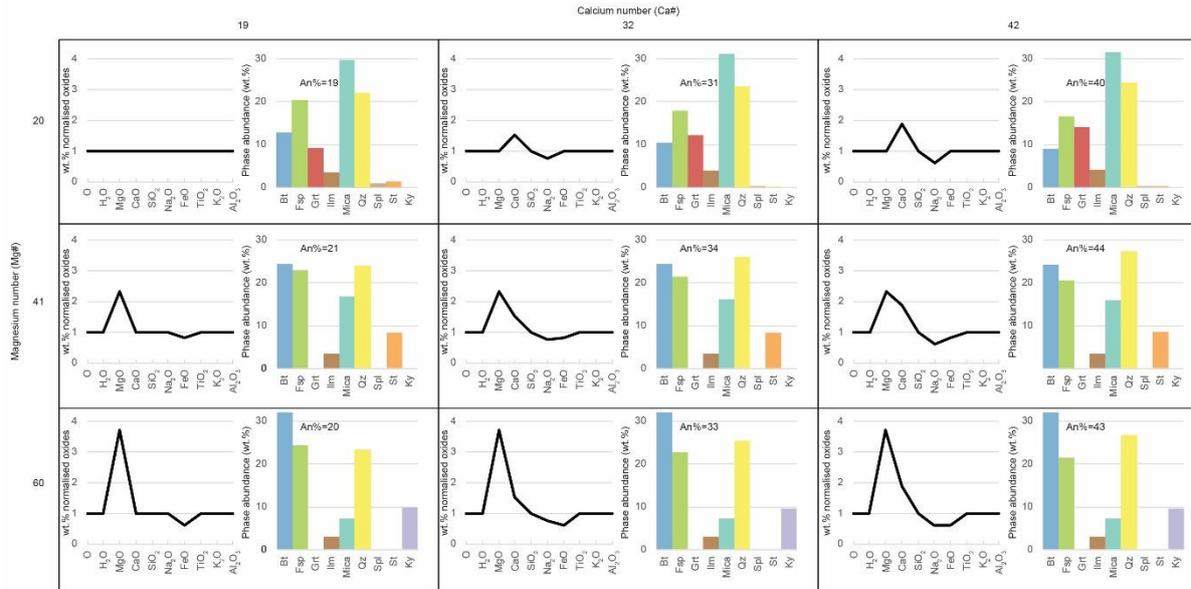


Fig. 1. Starting 9 bulk compositions produced by altering the *Mg#* and *Ca#* of the average pelite composition of Ague (1991)(Table 1) expressed as wt.% oxides normalized to the Ague (1991) average with the phase assemblage produced by each bulk composition at 640 °C; 7 kbar expressed as wt.% of each phase present. The anorthite percentage as $An\% = Ca / (Ca + Na) * 100$ [mol.%] of feldspar is reported above each phase assemblage. Abbreviations for phases except for melt and mica were taken from Whitney and Evans (2010) as Bt, Biotite; Fsp, plagioclase and alkali-feldspar; Grt, garnet; Ilm, ilmenite; Ky, kyanite; Qz, quartz; Spl, spinel; St, staurolite.

Calculations were performed in Rcrust version 2017-10-26 (Mayne et al., 2016) using a compiled form of the meemum function (version 6.7.9) from the Perple_X suite of programs (Connolly, 2009) in the NCKFMASHTO chemical system utilizing the 2011 revised hp11ver.dat thermodynamic data file from the internally consistent dataset of Holland and Powell (2011). Solution models were chosen which are consistent with the slightly simplified chemistry of the bulk system (e.g. the chemical system does not account for Mn) yet takes into account substitutions that are important in stabilizing phases (e.g. Ti in biotite). The following solution models were used from solution_model_679.dat: Fsp(C1) for plagioclase and alkali-feldspar (Holland and Powell, 2003), Cpx(HP) for clinopyroxene (Holland and Powell, 1996), Ep(HP11) for epidote (Holland and Powell, 2011), Sp(WPC) for spinel (White et al., 2002), Ilm(WPH) for ilmenite (White et al., 2000) and the latest White et al. (2014) set of models: Bi(W) for biotite, Crd(W) for cordierite, Gt(W) for garnet,

melt(W) for melt, Mica(W) for micas, Opx(W) for orthopyroxene, St(W) for staurolite. All the *P-T* paths modelled start at 640 °C and 7 kbar with 10 wt.% H₂O. This results in a free water phase being present at the beginning of each path, with water lost as appropriate to the model situation from the 642 °C step onwards.

RESULTS

The phase assemblage plots and melt production information as a function of temperature, for the Ague (1991) average *Mg#* and *Ca#* bulk composition, with 7 vol.% melt extraction threshold, evolving along the gentle prograde heating path (640°C; 7kbar to 920°C; 9.5 kbar) for three fluid cases are illustrated in Figure 2. The diagram illustrates some important fundamentals regarding the role of water during partial melting. The case with water limited to a nominal porosity and the fully-hydrated, fluid-absent case yield similar results because the bulk rock water contents of each of these water restricted cases at the solidus are similar. Each of the three cases considered undergoes melting at 684 °C, but the water restricted case predicts less than 3 vol.% melt in the system between 684 and approximately 735 °C, where there is a pulse of melt to >7 vol.% triggering the extraction of melt. This pulse of melt is the consequence of muscovite being consumed by an incongruent melting reaction. Sillimanite mode increases distinctly at the muscovite-out temperature as sillimanite is the primary crystalline product of the incongruent melting reaction which consumes muscovite. Following the disappearance of muscovite and the melt extraction event, melt modes increase very slightly as a function of temperature between 735 and 770 °C, where biotite mode begins to decrease as a function of temperature, with a corresponding increase in melt abundance. Biotite incongruent melting results in the appearance of peritectic garnet and triggers three melt extraction events between 770 and 850 °C, the temperature at which biotite disappears. Heating between 850 and 920 °C produces a single further melt extraction event. This last pulse of melt extraction occurs at a slightly higher temperature in the fully-hydrated, fluid-absent system, relative to the rock with 0.1 vol.% pore-water in the subsolidus assemblage. Thus, the melting of both water restricted rocks is controlled by the incongruent melting behaviour of the micas. Quartz and plagioclase remain in significant proportions in the residuum, despite the rocks having lost 20 vol.% melt.

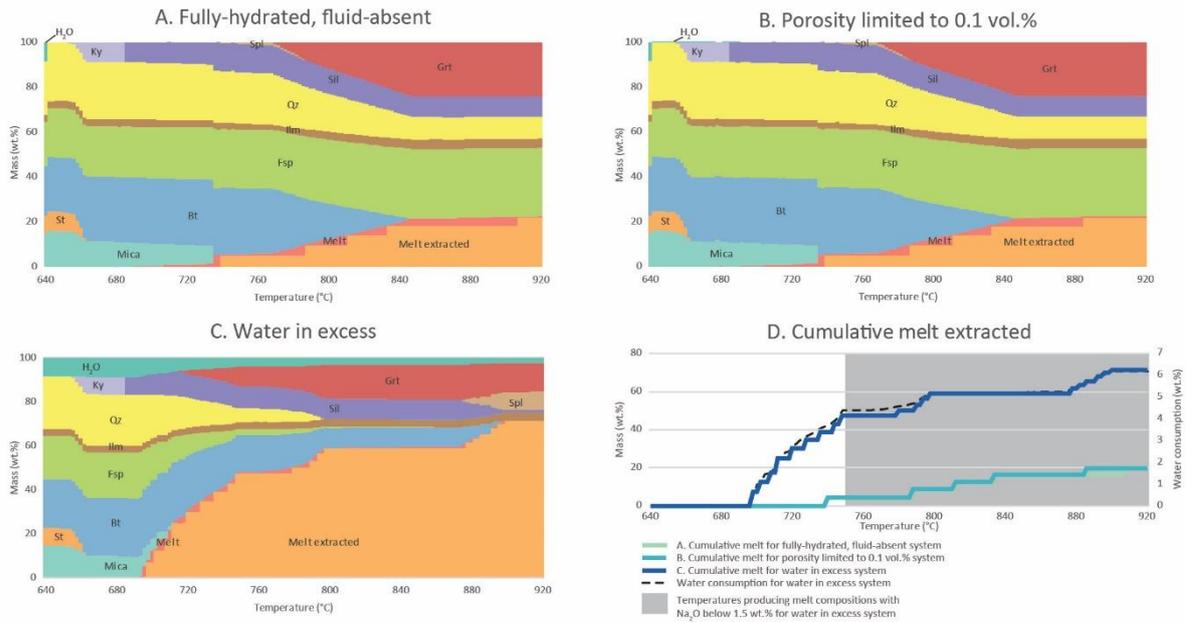


Fig. 2. Phase abundance and cumulative melt (both in wt.% relative to the full system) produced along P - T - X paths investigating water content (H_2O) as a source control on melt production considering 3 cases: a) fully-hydrated, fluid-absent; b) porosity limited to 0.1 vol.% and c) water in excess. For the 3 paths shown here $Mg\#$ and $Ca\#$ are taken as the Ague (1991) average with melt extraction events triggered by a 7 vol.% melt threshold along the gentle prograde heating path (640°C; 7kbar to 920°C; 9.5 kbar). Step resolution of paths is set at 2°C. Abbreviations for phases except for melt and mica were taken from Whitney and Evans (2010) as Bt, Biotite; Fsp, plagioclase and alkali-feldspar; Grt, garnet; H_2O , liquid water; Ilm, ilmenite; Ky, kyanite; Qz, quartz; Sil, sillimanite; Spl, spinel; St, staurolite.

In contrast, the availability of water controls the melting behaviour of the rock in the water in excess case; there is always as much melt in the system as the temperature and melt depleted composition will allow. Muscovite mode reduces markedly at the solidus, as do the modes of quartz and plagioclase. Muscovite disappears at approximately 700 °C and sillimanite mode grows with the decrease in muscovite, reflecting the incongruent melting of muscovite. Peritectic garnet appears at approximately 710 °C, marking this temperature as the point at which biotite decomposition makes a significant contribution to incongruent melting. The mode of biotite, quartz and plagioclase decrease substantially from this temperature to approximately 750 °C. Melt extraction events occur with a frequency of approximately one per 10 °C between the solidus and 750 °C. At this temperature the plagioclase composition in the residuum becomes anorthite, suppressing further biotite

melting to approximately 780 °C. Significant reduction in biotite and quartz mode occurs between 780 and 800 °C, with a number of corresponding melt extraction events. At 800 °C quartz is depleted and no further melt extraction events are triggered until approximately 870 °C, where biotite and sillimanite participate in an incongruent melting reaction which produces peritectic spinel which ends when biotite is exhausted at 900 °C.

The compositions of each melt extraction event produced along each of the possible *P-T-X* paths are plotted in Figures 3-4. The compositions of 548 bulk compositions reported for naturally occurring S-type granites (Euzen, 1993; Goad and Cerny, 1981; Scaillet et al., 1990; Nabelek et al., 1992; Solgadi et al., 2007; Williamson et al., 1996, 1997; Hine et al., 1978; Maas et al., 1997; Castro et al., 1999; Scheepers and Armstrong, 2002; Scheepers and Poujol, 2002; Bea et al., 1994; Ayres and Harris, 1997; Downes et al., 1990; Downes and Duthou, 1988; Chappell, 1978; Chappell and White, 1992; Birch, 1978; Bourne and Danis, 1987; Day and Weiblen, 1986; Georget and Martineau, 1986; Holtz and Johannes, 1991; Inger and Harris, 1993; Villaros, 2004; Sandeman and Clark, 2003; Scheepers, 1990) (Supplementary data: presented in Addendum D) are plotted for reference, although comparison between these rock compositions and the modelled melt compositions is only presented in the discussion. This dataset is filtered to only include samples that had been reported as being fresh and where possible thin section analysis was performed to ensure that only fresh samples were considered. Both the natural data and modelled data are normalised to the NCKFMAST chemical system where F is given as $\text{FeOt} = \text{FeO} + 0.89981 * \text{Fe}_2\text{O}_3$ by wt.% when FeO/Fe₂O₃ data was available (all the modelled data and a subset of the natural data) or else assuming all iron is ferrous. Harker plots are shown for major elements in the chemical system modelled with exceptions of H₂O and O as there is insufficient data on these components in the natural array considered to provide a comparison and TiO₂ as this component is not included in the current melt(W) solution model.

Modelled data is grouped by each of the 3 possible cases for each source control with colours assigned by Colorbrewer (Brewer, 2002). Alphahull approximation (Pateiro-López and Rodríguez-Casal, 2010) is performed on each case using an alpha value of 150 in order to calculate the bounding area for each subset of the modelled data in y vs SiO₂ space for y as CaO, Na₂O, K₂O, Al₂O₃, MgO + FeOt, Mg#, A/CNK and Na/Ca using GCDkit (Janoušek et al., 2006). A vector connecting the centroid of each bounding area is drawn as an arrow pointing towards the highest valued case (e.g. highest water content considered). The length of the vector gives an indication of the magnitude by which the bounding areas

are shifted relative to one another by the source control of interest and therefore its relative strength in controlling the resultant melt composition. These vectors of source control are reproduced on the right of the diagram centred to a dimensionless panel for comparison.

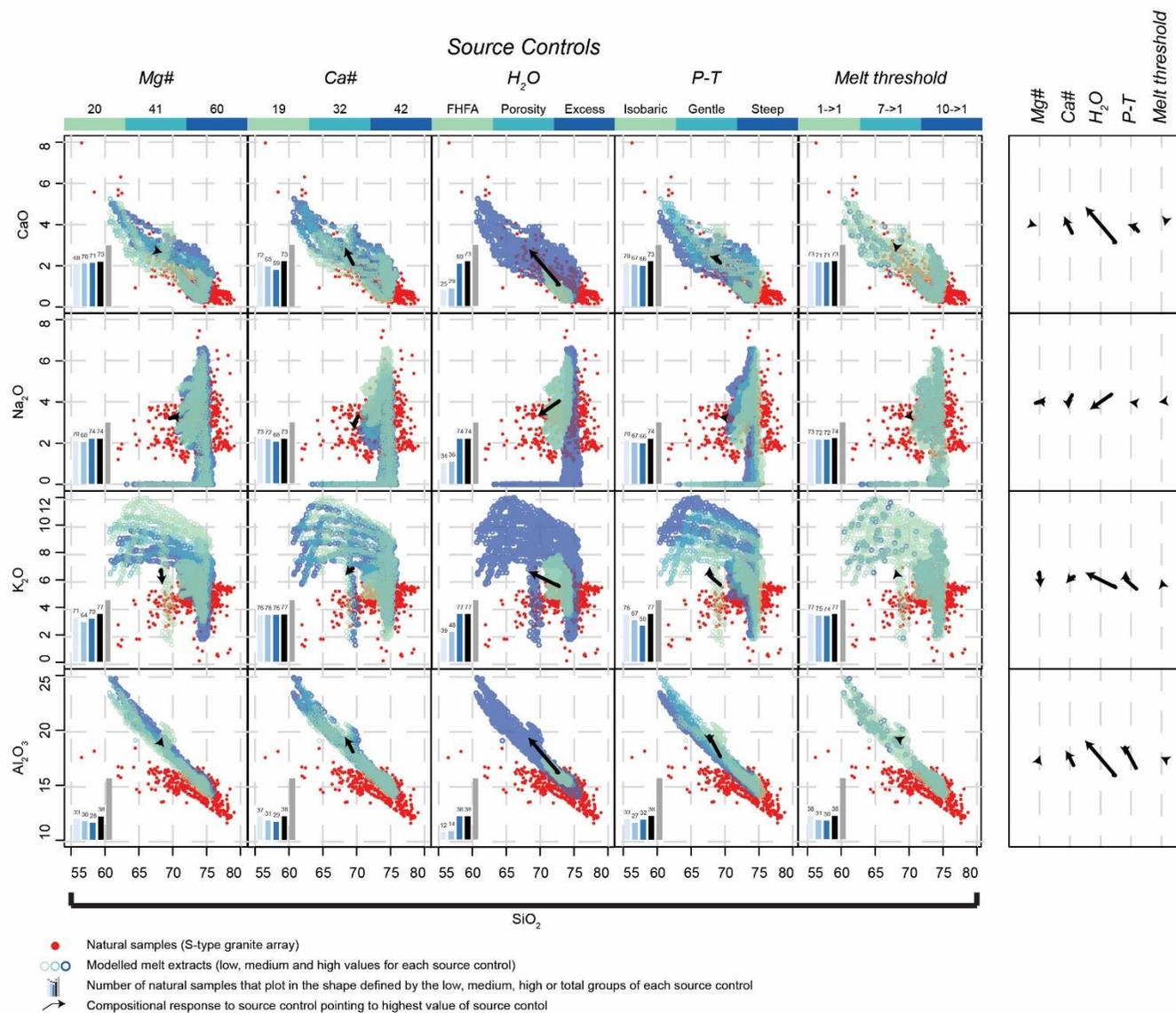


Fig. 3. Bulk compositions of natural data array of s-type granites and modelled melt extracts for the *source controls*: $Mg\#$, $Ca\#$, H_2O , $P-T$ and *Melt threshold* expressed as wt.% oxide versus wt.% SiO_2 . Weight percentages are normalised to the NCKFMAST chemical system. Modelled points are coloured according to the *case* considered for each *source control* (see Table 2) as low (light blue), medium (blue) and high (dark blue). Arrows link the central points of the space defined by each modelling *case* and point towards the highest valued *case*. Arrows produced by each source control are copied and centred on a unit-less diagram to the right of the figure for comparison.

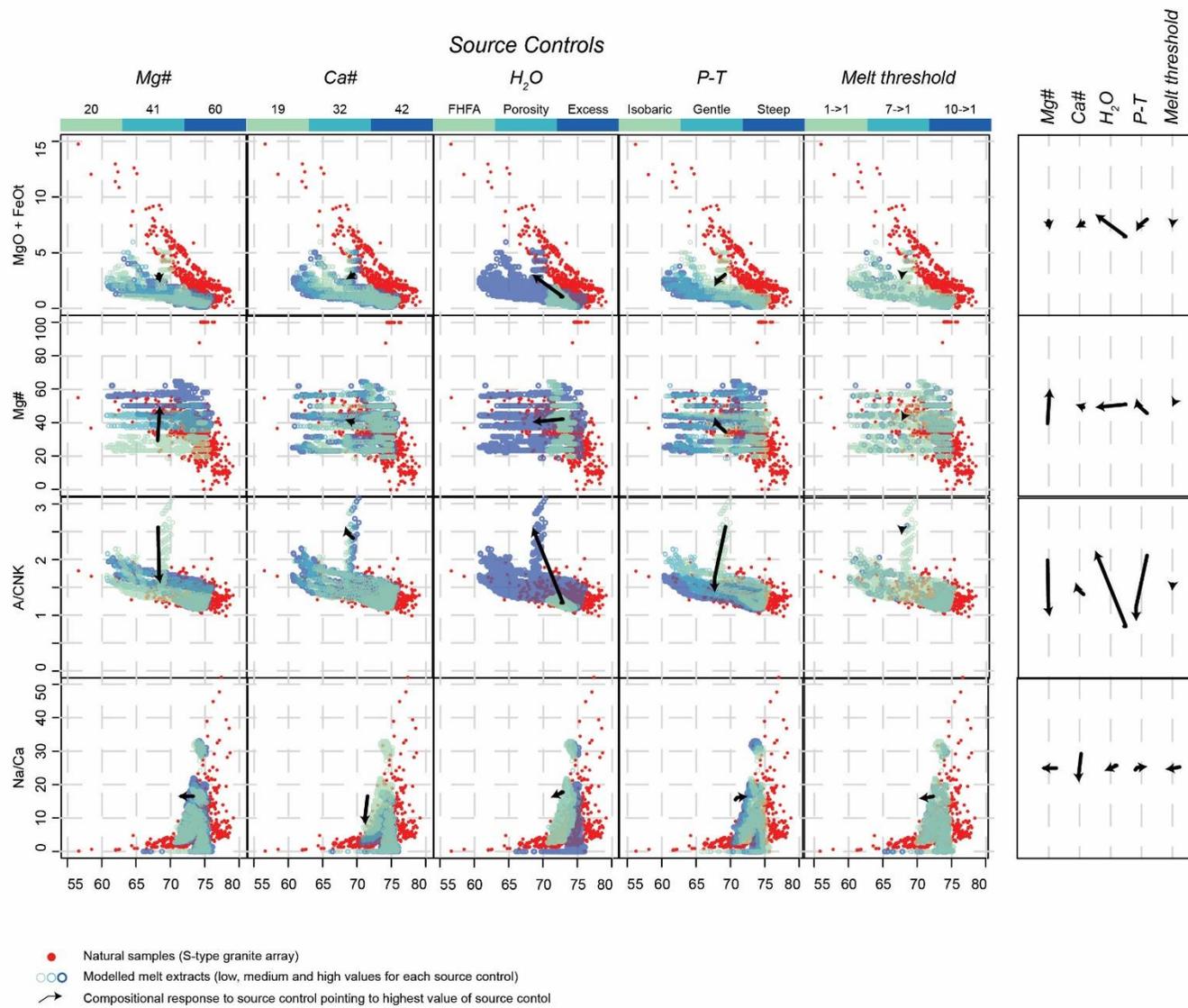


Fig. 4. Bulk compositions of natural data array of s-type granites and modelled melt extracts for the *source controls*: *Mg#*, *Ca#*, *H₂O*, *P-T* and *Melt threshold* expressed as $\text{MgO} + \text{FeOt} = \text{MgO} + \text{FeO} + 0.89981 * \text{Fe}_2\text{O}_3$ [wt.%]; $\text{Mg\#} = \text{MgO}/(\text{FeOt} + \text{MgO}) * 100$ [mol.%]; $\text{A/CNK} = \text{Al}_2\text{O}_3/(\text{CaO} + \text{Na}_2\text{O} + \text{K}_2\text{O})$ [mol.%] and Na/Ca [mol.%] all versus wt.% SiO_2 . Weight percentages are normalised to the NCKFMAST chemical system. Modelled points are coloured according to the *case* considered for each *source control* (see Table 2) as low (light blue), medium (blue) and high (dark blue). Arrows link the central points of the space defined by each modelling *case* and point towards the highest valued *case*. Arrows produced by each source control are copied and centred on a unit-less diagram to the right of the figure for comparison.

In both the water in excess and water restricted cases, the details of the melting behaviour and assemblage evolution vary significantly with P - T path steepness and rock composition. This is because pressure has a strong control on water solubility in the melt and on the mineralogy of the anhydrous ferromagnesian assemblage, whilst $Mg\#$ and $Ca\#$ have a strong control on the relative proportions of biotite, muscovite and plagioclase in the rocks (Fig. 1). However, allowing for these variations, all the modelling conducted in this study produces behaviour which is generally consistent with that described above. Each of the water in excess rocks produce three pulses of melt production, irrespective of the melt extraction threshold. For the gentle prograde heating example illustrated in Figure 2, the first of these occurs between the solidus and 740 °C. In this temperature interval, a large volume of melt (~50 vol.%) is produced through water fluxed incongruent reactions that consume muscovite and biotite along with quartz and plagioclase. These melts have SiO_2 contents very close to 75 wt.% and are characterised by Na_2O and K_2O contents that, respectively, vary inversely and proportionally with temperature (Fig. 5). This reflects the change in the stoichiometry of the low-temperature water present incongruent melting reactions to consume increasing amounts of mica as temperature increases. A second pulse of melt production occurs between 770 and 790 °C due to biotite + quartz + anorthite + H_2O incongruent melting. The melts have SiO_2 contents that range from 75 down to 63 wt.%, as well as very low Na_2O contents (<0.5 wt.%) and high K_2O contents (> 6 wt.%). A third pulse of melt production occurs between 870 and 900 °C, where biotite, sillimanite, garnet and H_2O melt incongruently to produce melt and peritectic spinel. This produces melts which range in SiO_2 content from 72 to 60 wt.% SiO_2 , with close to 0 wt.% Na_2O , K_2O between 6 and 12 wt.%, CaO between 2 and 5 wt.%, with very high Al_2O_3 , between 17 and 25 wt.%. Note that the exceptions to this are the portion of the light blue data array in Figure 5 which extend to low K_2O contents. These data points represent low $Mg\#$ rocks which evolved along the isobaric heating path at 7 kbar with melt extraction at a 1 vol.% melt threshold (thus recording small melt extraction amounts). In this situation biotite is depleted at relatively low temperature and these melt compositions are produced by incongruent melting of garnet, sillimanite and H_2O .

The water restricted models display considerably smaller compositional ranges of melt composition and two pulses of melt production (Fig. 5). The first is the consequence of incongruent melting reactions involving muscovite, the second is the result of incongruent melting reactions involving biotite. For both melt pulses, the melts range in composition between 72 and 75 wt.% SiO_2 , with K_2O between 3 and 6.5 wt.%, Na_2O between 3 and 6

wt.% and CaO < 1wt.%. The models with restricted water always produce approximately one third of the total melt volume produced by the water in excess rock at 920 °C. In the examples highlighted above, cumulative melt extracted from the system equates to 19.8 wt.% for the fully-hydrated, fluid-absent system; 20.0 wt.% for the porosity limited system and 71.4 wt.% for the water in excess system.

Progressive melting and melt extraction

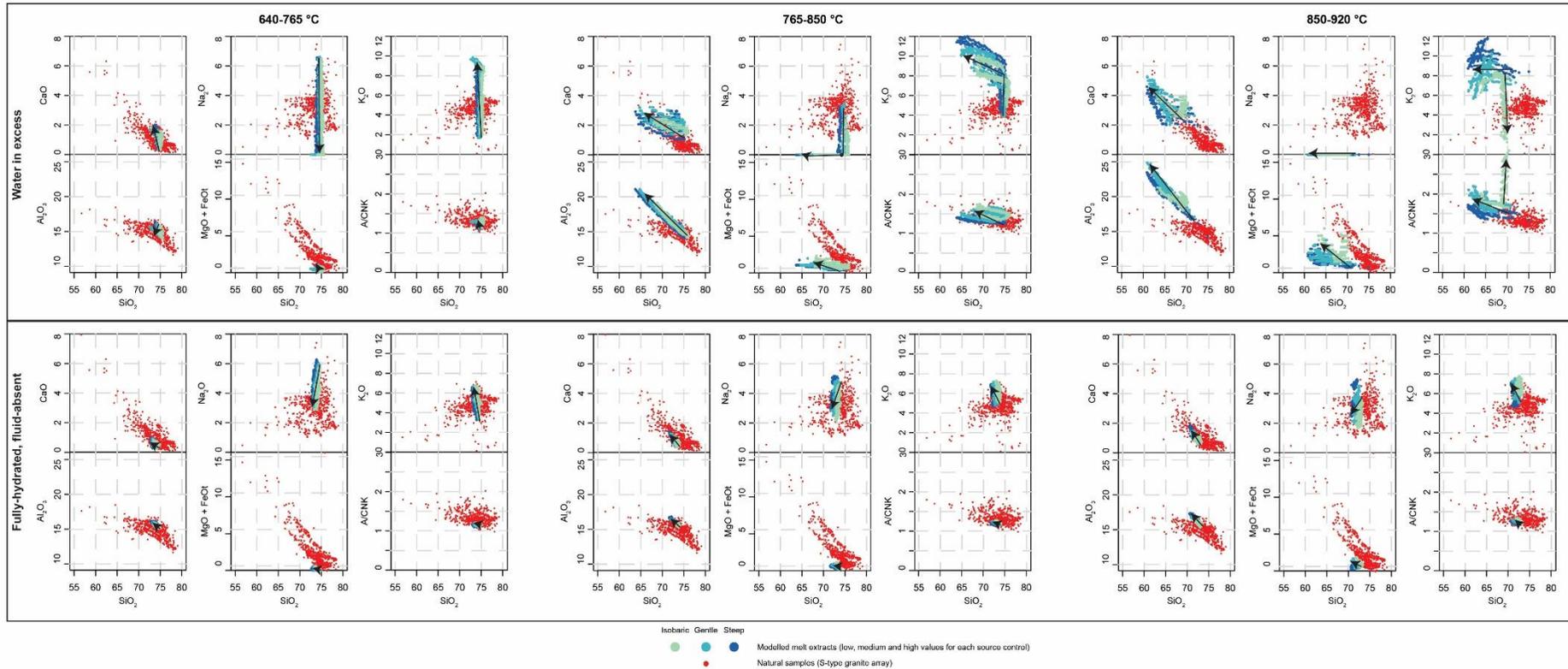


Fig. 5. Bulk compositions of natural data array of s-type granites and modelled melt extracts produced by varying the *source controls*: *Mg#*, *Ca#*, *P-T* and *Melt threshold* for the fluid case of water in excess (top) and fully-hydrated, fluid-absent (bottom). Modelled points are coloured according to the *P-T* path as isobaric (light blue), gentle (blue) and steep (dark blue). Melt extracts are grouped into 3 progressive temperature intervals with increasing temperature to the right of the figure. Weight percentages are normalised to the NCKFMAST chemical system.

DISCUSSION

The different source controls investigated in this study do not have equally significant influences on the composition of melt batches produced by progressive melting and melt extraction. The results indicate the hierarchy of influence on melt composition to be: 1. the presence or absence of water; 2. the pressure of melting; 3. the *Ca#* of the rock; 4. the *Mg#* of the rock; 5. *Melt threshold* (see size of compositional vectors on the right of Figs. 3&4). Ignoring the presence or absence of water, the nine bulk compositions investigated in this study vary substantially in terms of mode and composition of the reactant minerals, although all are metapelitic, with both muscovite and biotite present in the subsolidus assemblage along with quartz, plagioclase and aluminous minerals such as staurolite and/or aluminium silicates. Given the range in compositions it is noteworthy that on Figure 5, the different *P-T* path arrays are closely grouped, yet clearly discernible on almost every Harker plot. As the data for each *P-T* path include data on the compositions of the melt batches produced by all nine rock compositions, the variation in melt compositions produced by the compositional range covered is clearly subordinate to that induced by pressure of melting.

In making comparisons between the modelled melt compositions and the natural granite compositions, it needs to be considered that the granites are likely to represent mixtures of minerals and melts e.g. (Clemens and Stevens, 2012). Most hypotheses seeking to explain the petrogenesis of S-type granites can account for this. Restite entrainment and peritectic assemblage entrainment propose that a mineral load is carried out of the source during magma mobilization out of the source. Wall rock assimilation by melting can introduce peritectic minerals to the magma at the level of assimilation (Lackey et al., 2011). Mixing between hotter mafic magmas with a granitic melt will also produce an assemblage of ferromagnesian mineral in the resultant mixture. The results of this study, which covered the full range of temperature of anatexis that may be commonly attained in the mid to deep crust, also confirm the fact that common granites are mixtures of melt and a mineral assemblage. Despite the large number of permutations considered and the relatively high maximum temperatures used in the modelling, no melts are produced that have MgO and FeO contents as high as common granites (Fig. 4). Within the natural rock dataset FeO + MgO are inversely correlated with SiO₂ content ($R^2 = 0.73$) and FeO and MgO are positively correlated with TiO₂ ($R^2 = 0.74$). The modelled behaviour of TiO₂ could not be explored in this study because the activity-composition model for melts does not include TiO₂. These correlations are considerably better if the population considered is a single co-magmatic

suite e.g. (Stevens et al., 2007). Stevens et al. (2007) have demonstrated that as K_2O and TiO_2 are poorly correlated, and K_2O decreases as a function of TiO_2 and $FeO + MgO$, it is extremely unlikely that these major element geochemical characteristics in granites are a function of mixing with mafic magma or biotite accumulation. Stevens et al. (2007) and Clemens and Stevens (2012) argue that such well correlated positive associations between compatible elements are a consequence of entrainment of the peritectic assemblage produced principally by incongruent melting reactions involving biotite in the case of S-type granites, and incongruent melting reactions involving biotite and hornblende in the case of I-type granites. In the case of S-type granites the peritectic assemblage would consist principally of garnet and ilmenite or garnet, ilmenite and plagioclase e.g. (Madlakana and Stevens, 2018), whereas in I-type granites it would consist of clinopyroxene, orthopyroxene and plagioclase. Consequently, on Harker plots such as FeO vs SiO_2 , modelled melt compositions do not coincide with the natural granite population. In contrast, where the element of interest is an incompatible element that is readily accommodated in the melt, e.g. Na_2O or K_2O , the agreement between the range of concentrations defined by the natural rocks and that defined by the modelled melts may be expected to be good, where the modelled scenario is appropriate to the genesis of the granites.

All of the permutations of physical parameters and rock chemistry considered in the case of the water-restricted modelling produce melt compositions that could conceivably be involved in S-type granite genesis. The melt compositions change slightly as a function of P - T path and slightly as a function of temperature (Fig. 5), but generally occupy space on major element bivariate plots that is within the range of compositions defined by the granites. Entrainment of an appropriate peritectic assemblage produces compositional trends that are very similar to those portrayed by the natural rocks e.g. (García-Arias and Stevens, 2017a). Very importantly, because mica decomposition regulates the reactions, the melt compositions map out very similar ranges in Na_2O and K_2O as defined by the granites. If dilution by a peritectic assemblage is also taken into account (García-Arias and Stevens, 2017a) the modelled magma compositions will agree very well with the natural granites with $SiO_2 < 76$ wt.%. In the water in excess examples a substantial number of the melt compositions defined have no natural rock equivalents. These are compositions with SiO_2 in the range of granites and granodiorites, but with $Na_2O < 1$ wt.% and/or with $K_2O > 6$ wt.%. The low Na_2O compositions are produced by melting once plagioclase in the residuum has become strongly enriched in anorthite, whilst the high K_2O melts result from this melting, as well as further melting of biotite in the absence of quartz. The melt volumes produced are

substantial (> 20 vol.%) thus if such melting occurred in nature the results would be discernible in the rock record. Consequently, it can be concluded that such melting does not occur in nature. In Figure 5, water in excess melting of metapelites at relatively low temperature can be observed in terms of Na₂O and K₂O content in the melt as melting proceeds with increasing temperature. The melts produced close to the solidus have Na₂O > 6 wt.% and K₂O < 2 wt.%, with SiO₂ = 75 wt.%. Such compositions do occur in the natural rock population but they are rare. As melting progresses Na₂O in successive melt batches decreases and K₂O increases, as the stoichiometry of the melting reaction changes to consume less plagioclase and more muscovite. The melt compositions thus approach the compositions of common granites better. For melting along the gentle heating *P-T* path at 7 kbar, once approximately 40 vol.% melt has been extracted, the composition of the melt batches is sufficiently low in Na₂O (< 1.5 wt.%) and high in K₂O (> 7wt.%) that it can be concluded that no natural correlatives exist. This melt volume correlates with approximately 3.7 wt.% water consumed by the melting process and a temperature of only 740 °C. Further melting produces only increasingly more unusual extremely high K₂O melt compositions (Fig. 5).

The porosity of amphibolite facies metamorphic rocks is such that only a small amount of pore fluid may exist in the source prior to anatexis, as has been modelled in this study. The inclusion of this small amount of free water has a negligible effect on the melting history of the rock. Thus, melting with water in excess requires addition of water by some process. As it is not possible to move water through rocks above the water saturated solidus, such melting must occur at the interface between the hot rocks undergoing anatexis and the source of H₂O. This suggests that this process is likely to be most efficient at the solidus. The melt compositions produced are well characterised in this study, yet are rare in the natural rock record, leading to the conclusion that water fluxed melting of metapelites has not produced a significant proportion of the rocks reflected in the natural rock dataset used in this study.

The melt compositions modelled in this study attain a maximum SiO₂ content of approximately 76 wt.%; many granites have higher SiO₂ contents than this. This is a function of the fact that this study has examined melt compositions in the source, at lower pressure melts are more SiO₂ rich. Alternatively, fractional crystallization in the upper crust could provide a mechanism for producing granites with SiO₂ between 76 wt.% and 79 wt.% SiO₂ from melts similar to those predicted in this study.

CONCLUSIONS

This study explored the compositions of melt batches produced by progressive melting for 243 permutations of bulk composition, P - T path steepness, fluid state and details of melt extraction. Of these 81 cases were for water in excess, with the other 2/3 of cases reflecting water restricted or internally buffered conditions. The findings of this study provide valuable insights into the role of source controls in creating major element geochemical diversity in granites. The presence or absence of a water rich fluid in the source and the P - T path followed by the rock exert the strongest controls on melt compositions. Within metapelitic sources, compositional factors such as bulk rock Ca# and Mg# exert a minor influence, as does the amount of melt retention in the source prior to melt extraction, assuming the threshold is not greater than 10 vol.%. The greatest impact of P - T path steepness, for the range investigated, is that steeper paths produce slightly lower melt SiO₂ contents. This occurs for sources that are both quartz-saturated and for melt depleted sources where all quartz has been consumed. Thus it is a consequence of pressure on the modelled melt composition, not solely a function of $a\text{SiO}_2$.

Given that granite magmas predominantly represent mixtures of melts and crystals, with Fe, Mg, Ti, Ca and Al enriched in the crystal assemblage over melt, the most informative element concentrations in the modelled melts to compare with granite compositions are Na₂O and K₂O. These incompatible element concentrations in granites are likely to most closely approximate the compositions of the melts from which the granites formed. Water in excess melting of metapelites produces high melt yields (> 60 vol.%). A substantial volume of melt is produced at relatively low temperature (20 vol.% by 710 °C) and these melts have compositions that are unlike common granites as their K₂O/Na₂O ratios are low. Between 710 and 750 °C, water in excess melting occurs via the incongruent melting of muscovite and biotite and produced melt batches with compositions similar to normal granites. At higher temperature a number of incongruent melting reactions involving assemblages such as biotite, quartz and anorthite; biotite and quartz; biotite and sillimanite, produce melt batches with very low Na₂O and high K₂O that are unlike any common granites. In natural rocks, porosity considerations dictate that water in excess melting required addition of water to the rocks. Assuming geological circumstances that would allow for this, e.g. the thrusting of hot rocks over cold and then heating of both rock units to drive devolatilisation of the underlying rocks and melting of the overlying rocks, this modelling suggests that metapelites in the overlying unit would produce at least 20% of melts that are peraluminous

plagiogranites with K_2O as low as 1 wt.%. That such rocks are uncommon in the rock record suggests that such water in excess melting does not contribute significantly to the inventory of granites. At moderate temperatures, 710 to 750 °C, this process does produce melts and peritectic assemblages that could lead to reasonable granite compositions. However, this portion of the melting behaviour cannot be accessed without prior melt depletion, so the paucity of low K peraluminous plagiogranites suggests that rocks never get there. Fluid restricted partial melting of metapelites in the P - T ranges investigated in this study produces melt compositions that generally fit well with the petrogenesis of common peraluminous granites, if allowance is made for a process such as peritectic assemblage entrainment to increase FeO, MgO and A/CNK and produce the characteristic good correlation between FeO + MgO and TiO_2 . The modelled melts have maximum $SiO_2 = 76$ wt.% and this is lower than the value for many peraluminous leucocratic granites. These granites were either shaped by melt batches generated at substantially lower pressures than 7 kbar, or they have been shaped by crystal fractionation at low pressures. Some relatively low K_2O/Na_2O melt compositions also define the water restricted melt compositions produced at low temperature. This is a consequence of the fact that both the fully-hydrated, fluid-absent and pore water bearing rock compositions used do melt at the water-saturated solidus. However, the resultant melts that fall outside the range of normal granites make up a small fraction of the volume of melt produced from these rocks (< 3 vol.%)

DOWNLOAD

Rcrust is free to download and use under the GNU copyleft. The latest version of the software can be found on either of the two institutional addresses at: <http://www.univ-st-etienne.fr/rcrust> or <http://www.sun.ac.za/english/faculty/science/earthsciences/rcrust>. Alternatively, the software can be downloaded by contacting the corresponding author.

ACKNOWLEDGEMENTS

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF. Funding by the French Embassy of South Africa to M.Mayne and by the South African Research Chairs Initiative (SARChI) to G. Stevens is gratefully acknowledged.

SUPPLEMENTARY MATERIAL*

The full dataset of modelled melt compositions produced in this study and the array of s-type granite data used for comparison are contained in Addendum D. Files for reproducing figures 3-5 are contained in the zipped folder “Mayne_et_al_2018_3_modelling_files”. To open these download Rcrust from <https://www.sun.ac.za/english/faculty/science/earthsciences/rcrust> and replace the “Projects” and “data” folders with those in the zipped folder. Open each project by double clicking the appropriate x.RData file.

*For the thesis examiners’ convenience these files are already copied into the electronic copy of Rcrust provided in Addendum A, to open them simply install Rcrust following the user manual in Addendum B, open the “Projects” directory of Rcrust, open the project of choice and double click on the .RData file contained in the project file

REFERENCES

- Ague, J.J., 1991, Evidence for major mass transfer and volume strain during regional metamorphism of pelites: *Geology*, v. 19, p. 855–858.
- Ayres, M., and Harris, N., 1997, REE fractionation and Nd-isotope disequilibrium during crustal anatexis: Constraints from Himalayan leucogranites: *Chemical Geology*, v. 139, p. 249–269, doi: 10.1016/S0009-2541(97)00038-7.
- Barbarin, B., 1988, Field evidence for successive mixing and mingling between the Piolard Diorite and the Saint-Julien-la-Vêtre Monzogranite (Nord-Forez, Massif Central, France): *Canadian Journal of Earth Sciences*, v. 25, p. 49–59.
- Bea, F., Pereira, M.D., Corretgé, L.G., and Fershtater, G.B., 1994, Differentiation of strongly peraluminous, perphosphorus granites: The pedrobernardo pluton, central Spain: *Geochimica et Cosmochimica Acta*, v. 58, p. 2609–2627, doi: 10.1016/0016-7037(94)90132-5.
- Beard, J.S., Ragland, P.C., and Crawford, M.L., 2005, Reactive bulk assimilation: A model for crust-mantle mixing in silicic magmas: *Geology*, v. 33, p. 681–684, doi: 10.1130/G21470.1.
- Birch, W.D., 1978, Petrogenesis of some palaeozoic rhyolites in Victoria: *Australian Journal of Earth Sciences*, v. 25, p. 75–87, doi: 10.1080/00167617808729015.
- Bourne, J., and Danis, D., 1987, A proposed model for the formation of reversely zoned plutons based on study of the Lacorne Complex, Superior Province, Quebec: *Canadian Journal of Earth Sciences*, v. 24, p. 2506–2520.
- Breaks, Frederick, W., and Moore, J.M., 1992, The Ghost Lake Batholith, Superior Province of northwestern Ontario: a fertile, s-type, peraluminous granite - rare-element pegmatite system: *Canadian Mineralogist*, v. 30, p. 835–875.
- Brewer, C., 2002, ColorBrewer: www.ColorBrewer.org (accessed May 2018).
- Brown, M., and Pressley, R.A., 1999, Crustal Melting in Nature: Prosecuting Source Processes: *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, v. 24, p. 305–316.
- Castro, A., Patiño Douce, A.E., Corretgé, L.G., De La Rosa, J.D., El-Biad, M., and El-Hmidi, H.H., 1999, Origin of peraluminous granites and granodiorites, Iberian massif, Spain: An experimental test of granite petrogenesis: *Contributions to Mineralogy and Petrology*, v. 135, p. 255–276, doi: 10.1007/s004100050511.
- Chappell, B.W., 1978, Granitoids from the moonbi district, new england batholith, eastern Australia: *Journal of the Geological Society of Australia*, v. 25, p. 267–283, doi: 10.1080/00167617808729035.
- Chappell, B.W., and White, A.J.R., 1992, I-Type and S-Type granites in the Lachlan Fold Belt: *Transactions of the Royal Society of Edinburgh-Earth Sciences*, v. 83, p. 1–26, doi: 10.1017/S0263593300007720.
- Chappell, B.W., White, A.J.R., and Wyborn, D., 1987, The importance of residual source material (restite) in granite petrogenesis: *Journal of Petrology*, v. 28, p. 1111–1138.
- Clemens, J.D., Buick, I.S., and Stevens, G., 2016, Fluids, melting, granulites and granites: A commentary: *Precambrian Research*, v. 278, p. 394–399, doi: 10.1016/j.precamres.2016.01.001.
- Clemens, J.D., and Stevens, G., 2012, What controls chemical variation in granitic magmas? *Lithos*, v. 134–135, p. 317–329, doi: 10.1016/j.lithos.2012.01.001.
- Clemens, J.D., Stevens, G., and Farina, F., 2011, The enigmatic sources of I-type granites : The peritectic connexion: *Lithos*, v. 126, p. 174–181, doi: 10.1016/j.lithos.2011.07.004.
- Coleman, D.S., Gray, W., and Glazner, A.F., 2004, Rethinking the emplacement and evolution of zoned plutons: Geochronologic evidence for incremental assembly of the Tuolumne Intrusive Suite , California: *Geology*, v. 32, p. 433–436, doi: 10.1130/G20220.1.
- Collins, W.J., 1996, Lachlan Fold Belt granitoids: products of three component mixing: *Earth and Environmental Science Transactions of The Royal Society of Edinburgh*, v. 87, p. 171–181.
- Connolly, J.A.D., 2009, The geodynamic equation of state: What and how: *Geochemistry, Geophysics, Geosystems*, v. 10, p. 1–19, doi: 10.1029/2009GC002540.
- Day, W., and Weiblen, P., 1986, Origin of late Archean granite: geochemical evidence from the Vermilion Granitic Complex of northern Minnesota: *Contributions to Mineralogy and Petrology*, v. 93, p. 283–296.

- DePaolo, D.J., 1981, Trace element and isotopic effects of combined wallrock assimilation and fractional crystallization: *Earth and Planetary Science Letters*, v. 53, p. 189–202.
- Downes, H., Dupuy, C., and Leyreloup, A.F., 1990, Crustal evolution of the Hercynian belt of Western Europe: Evidence from lower-crustal granulitic xenoliths (French Massif Central): *Chemical Geology*, v. 83, p. 209–231, doi: 10.1016/0009-2541(90)90281-B.
- Downes, H., and Duthou, J.L., 1988, Isotopic and trace-element arguments for the lower-crustal origin of Hercynian granitoids and pre-Hercynian orthogneisses, Massif Central (France): *Chemical Geology*, v. 68, p. 291–308.
- Euzeu, T., 1993, Pétrogenèse des granites de collision post-épaississement : Le cas des granites crustaux et mantelliques du complexe de Pontivy-Rostrenen (Massif Armorica France) : Université de Rennes, 341 p.
- Farina, F., and Stevens, G., 2011, Source controlled $^{87}\text{Sr}/^{86}\text{Sr}$ isotope variability in granitic magmas : The inevitable consequence of mineral-scale isotopic disequilibrium in the protolith: *Lithos*, v. 122, p. 189–200, doi: 10.1016/j.lithos.2011.01.001.
- Foden, J.D., Elburg, M.A., Turner, S.P., Sandiford, M., O'Callaghan, J., and Mitchell, S., 2002, Granite production in the Delamerian orogen, South Australia: *Journal of the Geological Society*, v. 159, p. 557–575.
- García-Arias, M., and Stevens, G., 2017, Phase equilibrium modelling of granite magma petrogenesis : A . An evaluation of the magma compositions produced by crystal entrainment in the source: *Lithos*, v. 277, p. 131–153, doi: 10.1016/j.lithos.2016.09.028.
- Georget, Y., and Martineau, F., 1986, Age tardi-hercynien et origine crustale du granite de Brignogan (Finistère, France). Conséquences sur l'interprétation des granites Nordarmoricains: *Comptes rendus de l'Académie des sciences. Série 2, Mécanique, Physique, Chimie, Sciences de l'univers, Sciences de la Terre*, v. 302, p. 237–242.
- Goad, B.E., and Cerny, P., 1981, Peraluminous pegmatitic granites and their pegmatite aureoles in the Winnipeg River district, southeastern Manitoba: *Canadian Mineralogist*, v. 19, p. 177–194.
- Hine, R., Williams, I.S., Chappell, B.W., and White, A.J.R., 1978, Contrasts between i- and s-type granitoids of the Kosciusko batholith: *Journal of the Geological Society of Australia*, v. 25, p. 219–234, doi: 10.1080/00167617808729029.
- Holland, T., and Powell, R., 2003, Activity-compositions relations for phases in petrological calculations: An asymmetric multicomponent formulation: *Contributions to Mineralogy and Petrology*, v. 145, p. 492–501, doi: 10.1007/s00410-003-0464-z.
- Holland, T.J.B., and Powell, R., 2011, An improved and extended internally consistent thermodynamic dataset for phases of petrological interest, involving a new equation of state for solids: *Journal of Metamorphic Geology*, v. 29, p. 333–383, doi: 10.1111/j.1525-1314.2010.00923.x.
- Holland, T.J.B., and Powell, R., 1996, Thermodynamics of Order-Disorder in Minerals. II. Symmetric Formulism Applied to Solid Solutions: *American Mineralogist*, v. 81, p. 1425–1437.
- Holtz, F., and Johannes, W., 1991, Genesis of peraluminous granites I. Experimental investigation of melt compositions at 3 and 5 kb and various H₂O activities: *Journal of Petrology*, v. 32, p. 935–958.
- Inger, S., and Harris, N., 1993, Geochemical constraints on leucogranite magmatism in the Langtang Valley, Nepal Himalaya: *Journal of Petrology*, v. 34, p. 345–368.
- Janoušek, V., Farrow, C.M., and Erban, V., 2006, Interpretation of whole-rock geochemical data in igneous geochemistry: Introducing Geochemical Data Toolkit (GCDkit): *Journal of Petrology*, v. 47, p. 1255–1259, doi: 10.1093/petrology/egl013.
- Lackey, J.S., Erdmann, S., Hark, J.S., Nowak, R.M., Murray, K.E., Clarke, D.B., and Valley, J.W., 2011, Tracing garnet origins in granitoid rocks by oxygen isotope analysis: Examples from the South Mountain Batholith, Nova Scotia: *The Canadian Mineralogist*, v. 49, p. 417–440, doi: 10.3749/canmin.49.2.417.
- Maas, R., Nicholls, I.A., and Legg, C., 1997, Igneous and metamorphic enclaves in the S-type Deddick Granodiorite, Lachlan fold Belt, SE Australia: Petrographic, geochemical and Nd-Sr isotopic evidence for crustal melting and magma mixing: *Journal of Petrology*, v. 38, p. 815–841, doi: 10.1093/ptroj/38.7.815.
- Madlakana, N., and Stevens, G., 2018, Plagioclase disequilibrium induced during fluid-absent biotite-

- breakdown melting in metapelites: *Journal of Metamorphic Geology*, doi: 10.1111/jmg.12429.
- Mayne, M.J., Moyen, J.F., Stevens, G., and Kaislaniemi, L., 2016, Rcrust: a tool for calculating path-dependent open system processes and application to melt loss: *Journal of Metamorphic Geology*, v. 34, p. 663–682, doi: 10.1111/jmg.12199.
- Nabelek, P.I., Russ-Nabelek, C., and Haeussler, G.T., 1992, Stable isotope evidence for the petrogenesis and fluid evolution in the Proterozoic Harney Peak leucogranite, Black Hills, South Dakota: *Geochimica et Cosmochimica Acta*, v. 56, p. 403–417, doi: 10.1016/0016-7037(92)90141-5.
- Pateiro-López, B., and Rodríguez-Casal, A., 2010, Generalizing the Convex Hull of a Sample : The R Package Alphahull: *Journal of Statistical software*, v. 34, p. 1–28, doi: <http://dx.doi.org/10.18637/jss.v034.i05>.
- de Saint Blanquat, M., Horsman, E., Habert, G., Morgan, S., Vanderhaeghe, O., Law, R., and Tikoff, B., 2011, Multiscale magmatic cyclicality , duration of pluton construction, and the paradoxical relationship between tectonism and plutonism in continental arcs: *Tectonophysics*, v. 500, p. 20–33, doi: 10.1016/j.tecto.2009.12.009.
- Sandeman, H.A., and Clark, A.H., 2003, Glass-rich, Cordierite–Biotite Rhyodacite, Valle Ninahuisa, Puno, SE Peru: Petrological Evidence for Hybridization of “Lachlan S-type” and Potassic Mafic Magmas: *Journal of Petrology*, v. 44, p. 355–385, doi: 10.1093/petrology/44.2.355.
- Scailliet, B., France-Lanord, C., and Le Fort, P., 1990, Badrinath-Gangotri plutons (Garhwal, India): petrological and geochemical evidence for fractionation processes in a high Himalayan leucogranite: *Journal of Volcanology and Geothermal Research*, v. 44, p. 163–188, doi: 10.1016/0377-0273(90)90017-A.
- Scheepers, R., 1990, Magmatic association and radioelement geochemistry of selected Cape Granites with special reference to subalkaline and leucogranitic phases: Stellenbosch University, 151 p.
- Scheepers, R., and Armstrong, R., 2002, New U-Pb SHRIMP zircon ages of the Cape Granite Suite: implications for the magmatic evolution of the Saldania Belt: *South African Journal of Geology*, v. 105, p. 241–256.
- Scheepers, R., and Poujol, M., 2002, U-Pb zircon age of Cape Granite Suite ignimbrites: characteristics of the last phases of the Saldanian magmatism: *South African Journal of Geology*, v. 105, p. 163–178, doi: 10.2113/105.2.163.
- Solgadi, F., Moyen, J.F., Vanderhaeghe, O., Sawyer, E.W., and Reisberg, L., 2007, The role of crustal anatexis and mantle-derived magmas in the genesis of synorogenic Hercynian granites of the Livradois area, French massif central: *Canadian Mineralogist*, v. 45, p. 581–606, doi: 10.2113/gscanmin.45.3.581.
- Stevens, G., Clemens, and Droop, G.T.R., 1997, Melt production during granulite-facies anatexis: experimental data from “primitive” metasedimentary protoliths: *Contributions to Mineralogy and Petrology*, v. 128, p. 352–370.
- Stevens, G., and Reenen, D. van R., 1992, Partial melting and the origin of metapelitic granulites in the Southern Marginal Zone of the Limpopo Belt, South Africa: *Precambrian Research*, v. 55, p. 303–319.
- Stevens, G., Villaros, A., and Moyen, J.F., 2007, Selective peritectic garnet entainment as the origin of geochemical diversity in S-type granites: *Geology*, v. 35, p. 9–12, doi: 10.1130/G22959A.1.
- Taylor, J., and Stevens, G., 2010, Selective entrainment of peritectic garnet into S-type granitic magmas: Evidence from Archaean mid-crustal anatexites: *Lithos*, v. 120, p. 277–292, doi: 10.1016/j.lithos.2010.08.015.
- Villaros, A., 2004, Fusion partielle d’une croute hétérogène (metapelites et orthogneiss) à moyenne to haute pression: Grenoble University, 60 p.
- Villaros, A., and Stevens, G., 2009, The trace element compositions of S-type granites: evidence for disequilibrium melting and accessory phase entrainment in the source: *Contributions to Mineralogy and Petrology*, v. 158, p. 543–561, doi: 10.1007/s00410-009-0396-3.
- Villaros, A., Stevens, G., and Buick, I.S., 2009, Tracking S-type granite from source to emplacement : Clues from garnet in the Cape Granite Suite: *Lithos*, v. 112, p. 217–235, doi: 10.1016/j.lithos.2009.02.011.
- White, R.W., Powell, R., and Clarke, G.L., 2002, The interpretation of reaction textures in Fe-rich metapelitic

- granulites of the Musgrave Block, Central Australia: Constraints from mineral equilibria calculations in the system: *Journal of Metamorphic Geology*, v. 20, p. 41–55, doi: 10.1046/j.0263-4929.2001.00349.x.
- White, R.W., Powell, R., Holland, T.J.B., Johnson, T.E., and Green, E.C.R., 2014, New mineral activity-composition relations for thermodynamic calculations in metapelitic systems: *Journal of Metamorphic Geology*, v. 32, doi: 10.1111/jmg.12071.
- White, R.W., Powell, R., Holland, T.J.B., and Worley, B., 2000, The effect of TiO₂ and Fe₂O₃ on metapelitic assemblages at greenschist and amphibolite facies conditions: mineral equilibria calculations in the system K₂O-FeO-MgO-Al₂O₃-SiO₂-H₂O-TiO₂-Fe₂O₃: *Journal of Metamorphic Geology*, v. 18, p. 497–511.
- Whitney, D.L., and Evans, B.W., 2010, Abbreviations for names of rock-forming minerals: *American Mineralogist*, v. 95, p. 185–187, doi: 10.2138/am.2010.3371.
- Williamson, B.J., Downes, H., Thirlwall, M.F., and Beard, A., 1997, Geochemical constraints on restite composition and unmixing in the Velay anatectic granite, French Massif Central: *Lithos*, v. 40, p. 295–319, doi: 10.1016/S0024-4937(97)00033-9.
- Williamson, B.J., Shaw, A., Downes, H., and Thirlwall, M.F., 1996, Geochemical constraints on the genesis of Hercynian two-mica leucogranites from the Massif Central, France: *Chemical Geology*, v. 127, p. 25–42, doi: 10.1016/0009-2541(95)00105-0.

CHAPTER 5

CONCLUSION AND FUTURE PERSPECTIVES

Preliminary results from the creation and implementation of the Rcrust software tool appears to confirm that open system processes can be effectively investigated through path dependent modelling in $P-T-X$ space. By altering the bulk composition of a number of subsystems in a defined modelling space, where compositional changes are inherited along dependent paths, processes of mass transfer have been simulated. This new methodology and its software implementation in Rcrust addresses the need for thermodynamically constrained modelling of complex open system behaviour (Table 1).

Table 1. Summary of the capabilities of Rcrust for comparison with information presented on other phase equilibrium modelling softwares presented in Chapter 1 Table 1 detailing the suitability of each respective software program for performing phase equilibrium modelling in open systems

| Software | Calculation method | Graphical refinements | User Input | Phase manipulations |
|------------|-------------------------|-----------------------|--|---|
| Rcrust [1] | Inherited from Perple_X | None | Graphical user interface, text file input or R console input | Along multiple paths, multiple concurrent phase manipulations as conditional phase additions and extractions that can be set proportions or defined relative to system properties |

[1] <https://www.sun.ac.za/english/faculty/science/earthsciences/rcrust>

The methodology presented can deal with phase fractionations or additions concurrently. These phase manipulations can be triggered by conditions on the reactive system and can effect multiple phases by proportional changes that are set prior to calculation or can be defined relative to system properties (e.g. melt retention). The procedure for defining dependent paths allows multiple complex $P-T-X$ paths to be investigated simultaneously with calculations running sequentially through points within a path. Through these functionalities this new tool enables lines of investigation previously inaccessible to phase equilibrium modelling. The creation of these functions has focused on dealing with the most common chemical fluxes in metamorphic systems. These are the loss of subsolidus fluids

during prograde evolution; and at higher temperatures the loss of melts or magmas. Both of these processes have been accounted for in the modelling and the current software can handle their contributions to the effective bulk composition of the system concurrently in modelling space.

The presented papers, that constitute this thesis, document methodologies for using Rcrust to perform process-oriented investigations that involve mass transfer. These methodologies address the dynamic setting of subsolidus water content in a system undergoing prograde metamorphism; the triggering of melt loss events in a conditionally open system with melt retention on grain boundaries; the entrainment of phases to melt during melt loss (including the possibility of phase selective entrainment); the fractional crystallisation of magmas whereby the proportion of solid crystals in a cooling magma can trigger crystal loss from that magma; and the limiting of the availability of slow diffusing phases from contributing to the bulk composition of the chemically reactive subsystem.

The methodology for dynamic setting of subsolidus water loss is investigated further for its effect on the onset of anatexis and the control of the P – T path on this behaviour. It is found that the trajectory of the P – T path of a rock can influence the conditions under which melting begins and can effect the melt fertility of a metasedimentary protolith. This is a result of the progression of the P – T path controlling which subsolidus reactions are experienced by the rock on its prograde path and thus determining the total water content of the rock once anatexis begins. The combination of fluid state and P – T path appear to be the dominant source control on the compositions of melts produced by sequential melt loss from a metasedimentary protolith. These controls appear stronger than both the effects of compositional heterogeneity of the source as well as of volumetric controls on melt loss events.

Potential limitations

The main limitation for all metamorphic studies following a phase equilibrium approach is the delineation of equilibrium domains on the mineral and rock scale: the successive (re)equilibrations described by López-Carmona et al. (2014) as also evidence by the common observation of chemically zoned minerals. To determine what portions of the rock are in mutual equilibrium and what each of these domains could represent in the rock's complex P – T – X history presents the greatest challenge to its petrological interpretation (Lanari and Engi, 2017; Yakymchuk et al., 2017; Yakymchuk, 2017; Spear et al., 2016). Rcrust offers potential solutions for modelling this behaviour which are difficult to

implement with other software programs. By considering each domain in the rock as a subsystem in Rcrust we may be able to unravel some of the complex relations inherent in these systems and find constraints for the mechanisms by which they interact. Further possible limitations to all phase equilibrium modelling lie in the restraints imposed by the compositional range of the available thermodynamic data and the activity–composition models developed from this data. In Rcrust since chemical fluxes can be large, for example when considering continuous melt loss from rocks attaining high temperatures (~1000 °C) the bulk compositional shifts experienced by the residuum could take the composition of the effective bulk composition out of the range that the thermodynamic models were designed to handle.

New path dependent diagrams and a suggested protocol for reporting

Rcrust presents a functionally unique form of phase equilibrium modelling as its methodology of setting path dependence allows the pressure temperature and bulk composition of the system to all vary simultaneously in modelling space. If composition is set to be constant traditional isochemical phase diagrams (metamorphic assemblage diagrams) of P – T space can be produced (e.g. Fig. 1a). However, if phase manipulations are applied to the system, then bulk compositional changes can occur within the defined P – T space and the resultant diagrams can no longer be read in the same way as isochemical phase diagrams (e.g. Fig. 1b). Instead the reader must be made aware of the direction of the constituent paths that produced the diagram and must take care to only interpret petrological changes invoked in these systems along each respective path rather than across paths. It is proposed that these compositionally variable diagrams be termed “phase assemblage maps” in order to avoid the confusion of them being interpreted as isochemical phase diagrams. Further it is proposed that by convention these diagrams must be accompanied by indicators of their paths’ directions and positions in P – T space (e.g. the symbol above Fig.1b and the overprint arrows in Chapter 3: Figs 1, 3 & 4).

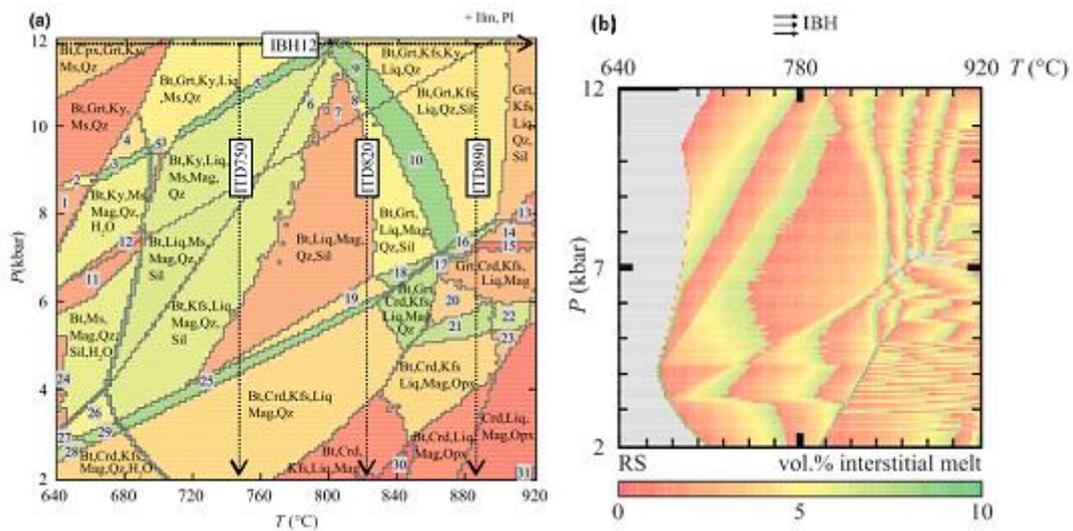


Fig. 1. (a) Isochemical phase diagram calculated using Rcrust in the system $\text{Na}_2\text{O}-\text{CaO}-\text{K}_2\text{O}-\text{FeO}-\text{MgO}-\text{Al}_2\text{O}_3-\text{SiO}_2-\text{H}_2\text{O}-\text{TiO}_2-\text{O}_2$ for an average amphibolite facies metapelite.

Individual isobaric heating and isothermal decompression paths in this $P-T$ space are labelled as IBH and ITD respectively. (b) Phase assemblage diagram calculated in Rcrust by compiling multiple isobaric heating paths that all start with the same bulk composition described above. Melt extraction is set to occur along each path whenever a 7 vol.% threshold is exceeded and remove all melt present except 1 vol.% in order to approximate melt retention on grain boundaries. The bulk composition of each point within a path is passed to the next point in sequence thereby creating a path dependent bulk composition which inherits compositional changes invoked by previous melt loss events Both diagrams are taken from Mayne et al. (2016).

Documented uses of Rcrust

Rcrust has been used by the following authors for phase equilibrium investigations. The projects I was personally involved in are identified with a *. In these projects I programmed functional improvements to Rcrust in response to the respective author's requests. All other projects are solely the work of the respective authors and are listed here only to document the scope of the current use of the Rcrust software.

Entrainment of peritectic phases*

Nicoli et al (2017) used Rcrust to simulate melt loss with entrainment of peritectic garnet. They used this model to explain the formation of K_2O -poor leucosomes in the Southern Marginal Zone of the Limpopo Belt, South Africa as the result of multi-step disequilibrium partial melting. Rcrust allowed melt extractions to be triggered by a melt threshold and extract garnet in combination with melt at each of these events.

Fractional crystallisation*

Soorajlal (2017) proposed a low temperature filter pressing model for Buddusò I-type granites and mafic enclaves in Sardinia, Italy in which melt is displaced from a crystal mush in order to produce a cumulate portion with granodioritic composition and an extract portion with leucogranitic composition. Soorajlal modelled the sequential extraction of melt from the reactive subsystem in equilibrium at a number of fixed P - T conditions in order to simulate a process of melt loss through filter pressing. Rcrust allowed melt loss events to be triggered by the cumulative proportion of crystals in the system. Work by Farina (2018) further develops the phase equilibrium constraints on this process of fractional crystallisation.

Crystallisation sequences

Holness et al (2018) used Rcrust in their investigation of the conditions of alkali-feldspar crystallisation. They found an apparent disconnect between phase equilibrium modelling predicting alkali-feldspars to begin crystallising late at relatively low temperatures in contrast to the field evidence in many granites containing large subhedral-to-euhedral K-feldspar phenocrysts.

Compositional constraints in planetary sciences

Mohit Melwani of the Jet Propulsion Laboratory of the Californian Institute of Technology used Rcrust to investigate melt formation and migration on early Mars in order to constrain the mass of volatile species (including SO_2) degassed from volcanoes over time. Mohit has also used Rcrust to investigate melt production and migration from the melt source regions within Europa's silicate interior (Melwani Daswani and Vance, 2018a, 2018b)

Estimating the duration of partial melting

Farina et al (2017) used Rcrust to correlate multi-pulsed magmatism in the Monte Capanne pluton with multimodal distribution of zircon ages in order to estimate the duration of partial melting and the timescale of melt extraction from the source

Identifying fluid state during crustal melting

Schwindinger (2018) and Schwindinger et al (2018) used Rcrust to simulate a conditionally open system effected by melt loss events. They found that the main indicator for water present melting of metasedimentary sources undergoing anatexis is the melt fraction produced, with the composition of melt and peritectic phases appearing to be insensitive to the fluid state of the system.

The role of thermal gradients in crustal melting

Juan David Montenegro of the National University of Colombia used Rcrust to model melt loss along different thermal gradients in order to assess the production and evolution of

tonalite–trondhjemite–granodiorite magmas in the Archean and its implications for the generation of continental crust.

Future work

The above projects demonstrate the broad scope of Rcrust's use and highlight its contribution to the petrological community. Up till now the development of Rcrust has largely focused on addressing compositional changes invoked by open system behaviour. Future work will build on expanding functionalities to the user and further developing these methodologies. In addition, there are a number of new directions that Rcrust can further develop:

Kinetic restraints on equilibrium

The limiting of the availability of a phase from contributing to the effective bulk composition of the reactive system is only partly addressed in Chapter 2. Future work could build models to consider the effect of the kinetic speed of reactions in limiting the equilibration volume of a system (Lanari and Engi, 2017). Here the diffusion rate of chemical components within slow diffusing phases such as garnet (Caddick et al., 2010) or plagioclase feldspar (Tajčmanová et al., 2007) could be integrated into modelling procedures. This could control the rate at which chemical components from a chemically zoned phase are available to the reactive bulk.

Consideration of trace elements in thermodynamic modelling

A major hesitation for igneous petrologists to adopt phase equilibrium modelling has been in the inability of current software to model trace element behaviour especially in magmatic systems. Including trace element data in thermodynamic databases and in the activity–composition models would require immense time, effort and resources. For processes occurring above the solidus, however, an easier alternative would be to use partition coefficients to approximate the fractionation of trace elements as the magmatic system evolves.

Saturation model for zircon

Another major limitation to the modelling of trace elements is the lack of appropriate thermodynamic models for the accessory phases which are often the largest sinks for trace elements. Some of the accessory phases for example zircon could be approximated using saturation models which assume that all of a given trace element e.g. zirconium above that which the current stable phases of the system can accommodate must exist in a given

accessory phase here zircon. This allows a thermodynamically unconstrained abundance of zircon to be approximated and manipulated by the modelling software as if it were determined phase.

Building on each of these procedures in an open source software platform will allow cumulative functionalities to be integrated into future modelling packages. In this way I aim to create an internally consistent, thermodynamically and kinetically constrained model for crustal anatexis.

REFERENCES

- Caddick, M.J., Konopásek, J., and Thompson, A.B., 2010, Preservation of garnet growth zoning and the duration of prograde metamorphism: *Journal of Petrology*, v. 51, p. 2327–2347, doi: 10.1093/petrology/egq059.
- Farina, F., 2018, Petrogenesis of the zoned Budduso Pluton (Sardinia, Italy): phase equilibria constraints on fractional crystallization: *Geological Society of London Special Issue: Making granites: petrogenetic processes, tectonic environments and secular variations*,.
- Farina, F., Stevens, G., Dini, A., and Schaltegger, U., 2017, Small-scale zircon age variability in the Monte Capanne pluton revealing the timescale of crustal melting and melt extraction, *in* Goldschmidt 2017 Abstract,.
- Holness, M.B., Clemens, J.D., and Vernon, R.H., 2018, How deceptive are microstructures in granitic rocks? Answers from integrated physical theory, phase equilibrium, and direct observations: *Contributions to Mineralogy and Petrology*, v. 173, p. 62, doi: 10.1007/s00410-018-1488-8.
- Lanari, P., and Engi, M., 2017, Local Bulk Composition Effects on Metamorphic Mineral Assemblages: *Reviews in Mineralogy and Geochemistry*, v. 83, p. 55–102, doi: 10.2138/rmg.2017.83.3.
- Mayne, M.J., Moyen, J.F., Stevens, G., and Kaislaniemi, L., 2016, Rcrust: a tool for calculating path-dependent open system processes and application to melt loss: *Journal of Metamorphic Geology*, v. 34, p. 663–682, doi: 10.1111/jmg.12199.
- Melwani Daswani, M., and Vance, S.D., 2018a, Melt production, eruption and degassing from Europa's interior: effects on composition over time, *in* Europa Deep Dive 2: Composition 2018,.
- Melwani Daswani, M., and Vance, S.D., 2018b, Thermal-Compositional Evolution of Europa's Interior and Ocean Since Accretion, *in* American Geophysical Union,.
- Nicoli, G., Stevens, G., Moyen, J.F., Vezinet, A., and Mayne, M.J., 2017, Insights into the complexity of crustal differentiation: K₂O-poor leucosomes within metasedimentary migmatites from the Southern Marginal Zone of the Limpopo Belt, South Africa.: *Journal of Metamorphic Geology*, doi: 10.1111/jmg.12265.
- Schwindinger, M., 2018, The Birthplace of Granite. Mechanisms controlling the evolution of granite magma within the source : Examples from Kangaroo Island, South Australia: 135 p.
- Schwindinger, M., Weinberg, R.F., and Clos, F., 2018, Wet or dry? The difficulty of identifying the presence of water during crustal melting: *Journal of Metamorphic Geology*, v. Submitted.
- Soorajlal, R., 2017, An investigation of the petrogenesis of the Buddusò I-type granites and its mafic enclaves in Sardinia, Italy.
- Spear, F.S., Pattison, D.R.M., and Cheney, J.T., 2016, The metamorphism of metamorphic petrology: *Geological Society of America Special Paper*, v. 523, p. 31–74, doi: 10.1130/2016.2523(02).
- Tajčmanová, L., Konopásek, J., and Connolly, J.A.D., 2007, Diffusion-controlled development of silica-undersaturated domains in felsic granulites of the Bohemian Massif (Variscan belt of Central Europe): *Contributions to Mineralogy and Petrology*, v. 153, p. 237–250, doi: 10.1007/s00410-006-0143-y.
- Yakymchuk, C., 2017, Applying phase equilibria modelling to metamorphic and geological processes: Recent developments and future potential: *Geoscience Canada*, v. 44, p. 27–46, doi: 10.12789/geocanj.2017.44.114.
- Yakymchuk, C., Clark, C., and White, R.W., 2017, Phase Relations, Reaction Sequences and Petrochronology: *Reviews in Mineralogy and Geochemistry*, v. 83, p. 13 LP – 53, doi: 10.2138/rmg.2017.83.2.

BIBLIOGRAPHY

- Ague, J.J., 1991, Evidence for major mass transfer and volume strain during regional metamorphism of pelites: *Geology*, v. 19, p. 855–858.
- Albee, A.L., 1965, A petrogenetic grid for the Fe-Mg silicates of pelitic schists: *American Journal of Science*, v. 263, p. 512–536.
- Amiotte Suchet, P., and Probst, J.L., 1995, A global model for present-day atmospheric/soil CO₂ consumption by chemical erosion of continental rocks (GEM-CO₂): *Tellus*, v. 47B, p. 273–280.
- Ayres, M., and Harris, N., 1997, REE fractionation and Nd-isotope disequilibrium during crustal anatexis: Constraints from Himalayan leucogranites: *Chemical Geology*, v. 139, p. 249–269, doi: 10.1016/S0009-2541(97)00038-7.
- Bachmann, O., and Bergantz, G.W., 2004, On the origin of crystal-poor rhyolites: Extracted from batholithic crystal mushes: *Journal of Petrology*, v. 45, p. 1565–1582, doi: 10.1093/petrology/egh019.
- Barbarin, B., 1988, Field evidence for successive mixing and mingling between the Piolard Diorite and the Saint-Julien-la-Vêtre Monzogranite (Nord-Forez, Massif Central, France): *Canadian Journal of Earth Sciences*, v. 25, p. 49–59.
- Bea, F., Pereira, M.D., Corretgé, L.G., and Fershtater, G.B., 1994, Differentiation of strongly peraluminous, perphosphorus granites: The pedrobernardo pluton, central Spain: *Geochimica et Cosmochimica Acta*, v. 58, p. 2609–2627, doi: 10.1016/0016-7037(94)90132-5.
- Beard, J.S., Ragland, P.C., and Crawford, M.L., 2005, Reactive bulk assimilation: A model for crust-mantle mixing in silicic magmas: *Geology*, v. 33, p. 681–684, doi: 10.1130/G21470.1.
- Berman, R.G., 1988, Internally-consistent thermodynamic data for minerals in the system Na₂O-K₂O-CaO-MgO-FeO-Fe₂O₃-Al₂O₃-SiO₂-TiO₂-H₂O-CO₂: *Journal of Petrology*, v. 29, p. 445–522.
- Berman, R.G., 1991, Thermobarometry Using Multi-Equilibrium Calculations: a New Technique, With Petrological Applications: *Canadian Mineralogist*, v. 29, p. 833–855.
- Birch, W.D., 1978, Petrogenesis of some palaeozoic rhyolites in Victoria: *Australian Journal of Earth Sciences*, v. 25, p. 75–87, doi: 10.1080/00167617808729015.
- Blevin, P.L., and Chappell, B.W., 1992, The role of magma sources, oxidation states and fractionation in determining the granite metallogeny of eastern Australia: *Transactions of the Royal Society of Edinburgh: Earth Sciences*, v. 83, p. 305–316, doi: 10.1017/S0263593300007987.
- Bourne, J., and Danis, D., 1987, A proposed model for the formation of reversely zoned plutons based on study of the Lacorne Complex, Superior Province, Quebec: *Canadian Journal of Earth Sciences*, v. 24, p. 2506–2520.
- Breaks, Frederick, W., and Moore, J.M., 1992, The Ghost Lake Batholith, Superior Province of northwestern Ontario: a fertile, s-type, peraluminous granite - rare-element pegmatite system: *Canadian Mineralogist*, v. 30, p. 835–875.
- Brewer, C., 2002, ColorBrewer: www.ColorBrewer.org (accessed May 2018).
- Brown, M., 2014, The contribution of metamorphic petrology to understanding lithosphere evolution and geodynamics: *Geoscience Frontiers*, v. 5, p. 553–569.
- Brown, M., and Korhonen, F.J., 2009, Some Remarks on Melting and Extreme Metamorphism of Crustal Rocks, *in* *Physics and Chemistry of the Earth's Interior*, p. 67–87.
- Brown, M., and Pressley, R.A., 1999, Crustal Melting in Nature: Prosecuting Source Processes: *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, v. 24, p. 305–316.
- Caddick, M.J., Konopásek, J., and Thompson, A.B., 2010, Preservation of garnet growth zoning and the duration of prograde metamorphism: *Journal of Petrology*, v. 51, p. 2327–2347, doi: 10.1093/petrology/egq059.
- De Capitani, C., and Petrakakis, K., 2010, The computation of equilibrium assemblage diagrams with Theriak/Domino software: *American Mineralogist*, v. 95, p. 1006–1016, doi: 10.2138/am.2010.3354.
- Castro, A., Patiño Douce, A.E., Corretgé, L.G., De La Rosa, J.D., El-Biad, M., and El-Hmidi, H.H., 1999, Origin of peraluminous granites and granodiorites, Iberian massif, Spain: An experimental test of granite petrogenesis: *Contributions to Mineralogy and Petrology*, v. 135, p. 255–276, doi:

10.1007/s004100050511.

- Chappell, B.W., 1978, Granitoids from the moonbi district, new england batholith, eastern Australia: *Journal of the Geological Society of Australia*, v. 25, p. 267–283, doi: 10.1080/00167617808729035.
- Chappell, B.W., and White, A.J.R., 1992, I-Type and S-Type granites in the Lachlan Fold Belt: *Transactions of the Royal Society of Edinburgh-Earth Sciences*, v. 83, p. 1–26, doi: Doi 10.1017/S0263593300007720.
- Chappell, B.W., White, A.J.R., and Wyborn, D., 1987, The importance of residual source material (restite) in granite petrogenesis: *Journal of Petrology*, v. 28, p. 1111–1138.
- Clemens, J.D., Buick, I.S., and Stevens, G., 2016, Fluids, melting, granulites and granites: A commentary: *Precambrian Research*, v. 278, p. 394–399, doi: 10.1016/j.precamres.2016.01.001.
- Clemens, J.D., and Stevens, G., 2012, What controls chemical variation in granitic magmas? *Lithos*, v. 134–135, p. 317–329, doi: 10.1016/j.lithos.2012.01.001.
- Clemens, J.D., Stevens, G., and Farina, F., 2011, The enigmatic sources of I-type granites : The peritectic connexion: *Lithos*, v. 126, p. 174–181, doi: 10.1016/j.lithos.2011.07.004.
- Coleman, D.S., Gray, W., and Glazner, A.F., 2004, Rethinking the emplacement and evolution of zoned plutons: Geochronologic evidence for incremental assembly of the Tuolumne Intrusive Suite , California: *Geology*, v. 32, p. 433–436, doi: 10.1130/G20220.1.
- Collins, W.J., 1996, Lachlan Fold Belt granitoids: products of three component mixing: *Earth and Environmental Science Transactions of The Royal Society of Edinburgh*, v. 87, p. 171–181.
- Connolly, J.A.D., 2005, Computation of phase equilibria by linear programming: A tool for geodynamic modeling and its application to subduction zone decarbonation: *Earth and Planetary Science Letters*, v. 236, p. 524–541, doi: 10.1016/j.epsl.2005.04.033.
- Connolly, J.A.D., 2018, Phase (Melt) Fractionation with Perple_X:, http://www.perplex.ethz.ch/perplex_phase_fractionation.html (accessed October 2018).
- Connolly, J.A.D., 2009, The geodynamic equation of state: What and how: *Geochemistry, Geophysics, Geosystems*, v. 10, p. 1–19, doi: 10.1029/2009GC002540.
- Connolly, J.A.D., and Kerrick, D.M., 1987, An algorithm and computer program for calculating composition phase diagrams: *Calphad*, v. 11, p. 1–55, doi: 10.1016/0364-5916(87)90018-6.
- Day, W., and Weiblen, P., 1986, Origin of late Archean granite: geochemical evidence from the Vermilion Granitic Complex of northern Minnesota: *Contributions to Mineralogy and Petrology*, v. 93, p. 283–296.
- DePaolo, D.J., 1981, Trace element and isotopic effects of combined wallrock assimilation and fractional crystallization: *Earth and Planetary Science Letters*, v. 53, p. 189–202.
- Diener, J.F.A., and Fagereng, Å., 2014, The influence of melting and melt drainage on crustal rheology during orogenesis: *Journal of Geophysical Research : Solid Earth*, v. 119, p. 6193–6210, doi: 10.1002/2014JB011088.Received.
- Downes, H., Dupuy, C., and Leyreloup, A.F., 1990, Crustal evolution of the Hercynian belt of Western Europe: Evidence from lower-crustal granulitic xenoliths (French Massif Central): *Chemical Geology*, v. 83, p. 209–231, doi: 10.1016/0009-2541(90)90281-B.
- Downes, H., and Duthou, J.L., 1988, Isotopic and trace-element arguments for the lower-crustal origin of Hercynian granitoids and pre-Hercynian orthogneisses, Massif Central (France).: *Chemical Geology*, v. 68, p. 291–308.
- Euzen, T., 1993, Pétrogenèse des granites de collision post-épaississement : Le cas des granites crustaux et mantelliques du complexe de Pontivy-Rostrenen (Massif Armorica France) : Université de Rennes, 341 p.
- Farina, F., 2018, Petrogenesis of the zoned Budduso Pluton (Sardegna, Italy): phase equilibria constraints on fractional crystallization: *Geological Society of London Special Issue: Making granites: petrogenetic processes, tectonic environments and secular variations*,.
- Farina, F., and Stevens, G., 2011, Source controlled $87\text{Sr}/86\text{Sr}$ isotope variability in granitic magmas : The inevitable consequence of mineral-scale isotopic disequilibrium in the protolith: *Lithos*, v. 122, p. 189–200, doi: 10.1016/j.lithos.2011.01.001.
- Farina, F., Stevens, G., Dini, A., and Schaltegger, U., 2017, Small-scale zircon age variability in the Monte Capanne pluton revealing the timescale of crustal melting and melt extraction, *in* Goldschmidt

2017 Abstract.

- Foden, J.D., Elburg, M.A., Turner, S.P., Sandiford, M., O'Callaghan, J., and Mitchell, S., 2002, Granite production in the Delamerian orogen, South Australia: *Journal of the Geological Society*, v. 159, p. 557–575.
- Friend, J.P., 1973, The Global Sulfur Cycle, *in* S.I., R. ed., *Chemistry of the Lower Atmosphere*, Springer, p. 177–178.
- Garcia-Arias, M., and Stevens, G. Melting behavior of a metasedimentary source: Consequences for S-type granite magma compositions.: *Journal of Chemical Information and Modeling*.
- García-Arias, M., and Stevens, G., 2017a, Phase equilibrium modelling of granite magma petrogenesis : A . An evaluation of the magma compositions produced by crystal entrainment in the source: *Lithos*, v. 277, p. 131–153, doi: 10.1016/j.lithos.2016.09.028.
- García-Arias, M., and Stevens, G., 2017b, Phase equilibrium modelling of granite magma petrogenesis : B . An evaluation of the magma compositions that result from fractional crystallization: *Lithos*, v. 277, p. 109–130, doi: 10.1016/j.lithos.2016.09.027.
- Gardiner, N.J., Johnson, T.E., Kirkland, C.L., and Smithies, R.H., 2018, Melting controls on the lutetium–hafnium evolution of Archaean crust: *Precambrian Research*, v. 305, p. 479–488, doi: 10.1016/j.precamres.2017.12.026.
- Georget, Y., and Martineau, F., 1986, Age tardi-hercynien et origine crustale du granite de Brignogan (Finistère, France). Conséquences sur l'interprétation des granites Nordarmoricaux: *Comptes rendus de l'Académie des sciences. Série 2, Mécanique, Physique, Chimie, Sciences de l'univers, Sciences de la Terre*, v. 302, p. 237–242.
- Ghiorso, M.S., and Sack, R.O., 1995, Chemical mass transfer in magmatic processes IV . A revised and internally consistent thermodynamic model for the interpolation and extrapolation of liquid-solid equilibria in magmatic systems at elevated temperatures and pressures: *Contributions to Mineralogy and Petrology*, v. 119, p. 197–212.
- Goad, B.E., and Cerny, P., 1981, Peraluminous pegmatitic granites and their pegmatite aureoles in the Winnipeg River district, southeastern Manitoba: *Canadian Mineralogist*, v. 19, p. 177–194.
- Gottschalk, M., 1997, Internally consistent thermodynamic data for rock-forming minerals in the system SiO₂-TiO₂-Al₂O₃-CaO-MgO-FeO-K₂O-Na₂O-H₂O-CO₂: *European Journal of Mineralogy*, v. 9, p. 175–223.
- Helgeson, H.C., Delany, J.M., and Nesbitt, H.W., 1978, Summary and critique of the thermodynamic properties of rock-forming minerals: *American Journal of Science*, v. 278A, p. 1–229.
- Hensen, B.J., and Essene, E.J., 1971, Stability of pyrope-quartz in the system MgO-Al₂O₃-SiO₂: *Contributions to Mineralogy and Petrology*, v. 30, p. 72–83.
- Hensen, B.J., and Harley, S.L., 1990, Graphical analysis of P—T—X relations in granulite facies metapelites, *in* High-temperature metamorphism and crustal anatexis, Netherlands, Springer, p. 19–56.
- Hine, R., Williams, I.S., Chappell, B.W., and White, A.J.R., 1978, Contrasts between i- and s-type granitoids of the kosciusko batholith: *Journal of the Geological Society of Australia*, v. 25, p. 219–234, doi: 10.1080/00167617808729029.
- Holland, T., and Powell, R., 2003, Activity-compositions relations for phases in petrological calculations: An asymmetric multicomponent formulation: *Contributions to Mineralogy and Petrology*, v. 145, p. 492–501, doi: 10.1007/s00410-003-0464-z.
- Holland, T.J.B., and Powell, R., 2011, An improved and extended internally consistent thermodynamic dataset for phases of petrological interest, involving a new equation of state for solids: *Journal of Metamorphic Geology*, v. 29, p. 333–383, doi: 10.1111/j.1525-1314.2010.00923.x.
- Holland, T., and Powell, R., 1998, An internally consistent thermodynamic data set for phases of petrological interest: *Journal of Metamorphic Geology*, v. 16, p. 309–343.
- Holland, T.J.B., and Powell, R., 1996, Thermodynamics of Order-Disorder in Minerals. II. Symmetric Formulism Applied to Solid Solutions: *American Mineralogist*, v. 81, p. 1425–1437.
- Holness, M.B., Clemens, J.D., and Vernon, R.H., 2018, How deceptive are microstructures in granitic rocks? Answers from integrated physical theory, phase equilibrium, and direct observations: *Contributions to Mineralogy and Petrology*, v. 173, p. 62, doi: 10.1007/s00410-018-1488-8.
- Holtz, F., and Johannes, W., 1991, Genesis of peraluminous granites I. Experimental investigation of melt

- compositions at 3 and 5 kb and various H₂O activities: *Journal of Petrology*, v. 32, p. 935–958.
- Huang, W.L., and Wyllie, P.J., 1981, Phase Relationships of S-Type Granite with H₂O to 35 kbar : Muscovite Granite From Harney Peak , South Dakota: *Journal of Geophysical Research*, v. 86, p. 10515–10529.
- Indares, A., White, R.W., and Powell, R., 2008, Phase equilibria modelling of kyanite-bearing anatectic paragneisses from the central Grenville Province: *Journal of Metamorphic Geology*, v. 26, p. 815–836, doi: 10.1111/j.1525-1314.2008.00788.x.
- Inger, S., and Harris, N., 1993, Geochemical constraints on leucogranite magmatism in the Langtang Valley, Nepal Himalaya: *Journal of Petrology*, v. 34, p. 345–368.
- Janoušek, V., Farrow, C.M., and Erban, V., 2006, Interpretation of whole-rock geochemical data in igneous geochemistry: Introducing Geochemical Data Toolkit (GCDkit): *Journal of Petrology*, v. 47, p. 1255–1259, doi: 10.1093/petrology/egl013.
- Johannes, W., and Koepke, J., 2001, Incomplete reaction of plagioclase in experimental dehydration melting of amphibolite: *Australian Journal of Earth Sciences*, v. 48, p. 581–590, doi: 10.1046/j.1440-0952.2001.00876.x.
- Johannes, W., Koepke, J., and Behrens, H., 1994, Partial melting reactions of plagioclases and plagioclase-bearing systems, *in* Feldspars and their Reactions, Springer Netherlands, p. 161–194.
- Johnson, T., Gibson, R.L., Brown, M., Buick, I., and Cartwright, I., 2003, Partial Melting of Metapelitic Rocks Beneath the Bushveld Complex , South Africa: *Journal of Petrology*, v. 44, p. 301–314, doi: 10.1093/petrology/44.5.789.
- Johnson, T.E., White, R.W., and Brown, M., 2011, A year in the life of an aluminous metapelite xenolith — The role of heating rates , reaction overstep , H₂O retention and melt loss: *Lithos*, v. 124, p. 132–143, doi: 10.1016/j.lithos.2010.08.009.
- Karpov, I.K., Chudnenko, K.V., and Kulik, D.A., 1997, Modeling chemical mass transfer in geochemical processes; thermodynamic relations, conditions of equilibria and numerical algorithms: *American Journal of Science*, v. 297, p. 767–806.
- Korhonen, F.J., Powell, R., and Stout, J.H., 2012, Stability of sapphirine + quartz in the oxidized rocks of the Wilson Lake terrane, Labrador: Calculated equilibria in NCKFMASHTO: *Journal of Metamorphic Geology*, v. 30, p. 21–36, doi: 10.1111/j.1525-1314.2011.00954.x.
- Lackey, J.S., Erdmann, S., Hark, J.S., Nowak, R.M., Murray, K.E., Clarke, D.B., and Valley, J.W., 2011, Tracing garnet origins in granitoid rocks by oxygen isotope analysis: Examples from the South Mountain Batholith, Nova Scotia: *The Canadian Mineralogist*, v. 49, p. 417–440, doi: 10.3749/canmin.49.2.417.
- Lanari, P., Bovay, T., Airaghi, L., and Centrella, S., 2018, Quantitative compositional mapping of mineral phases by electron probe micro-analyser:.
- Lanari, P., and Engi, M., 2017, Local Bulk Composition Effects on Metamorphic Mineral Assemblages: *Reviews in Mineralogy and Geochemistry*, v. 83, p. 55–102, doi: 10.2138/rmg.2017.83.3.
- Lanari, P., Vidal, O., De Andrade, V., Dubacq, B., Lewin, E., Grosch, E.G., and Schwartz, S., 2014, XMapTools: A MATLAB©-based program for electron microprobe X-ray image processing and geothermobarometry: *Computers and Geosciences*, v. 62, p. 227–240, doi: 10.1016/j.cageo.2013.08.010.
- Linnen, R.L., and Keppler, H., 2002, Melt composition control of Zr/Hf fractionation in magmatic processes: *Geochimica et Cosmochimica Acta*, v. 66, p. 3293–3301, doi: 10.1016/S0016-7037(02)00924-9.
- López-Carmona, A., Gutiérrez-Alonso, G., Tishin, P.A., and Gertner, I.F., 2014, Thermodynamic modelling of metamorphic processes: state of the art in pseudosection approach: *IOP Conference Series: Earth and Environmental Science*, v. 110.
- Maas, R., Nicholls, I.A., and Legg, C., 1997, Igneous and metamorphic enclaves in the S-type Deddick Granodiorite, Lachlan fold Belt, SE Australia: Petrographic, geochemical and Nd-Sr isotopic evidence for crustal melting and magma mixing: *Journal of Petrology*, v. 38, p. 815–841, doi: 10.1093/petroj/38.7.815.
- Madlakana, N., and Stevens, G., 2018, Plagioclase disequilibrium induced during fluid-absent biotite-breakdown melting in metapelites: *Journal of Metamorphic Geology*, doi: 10.1111/jmg.12429.
- Mayne, M.J., Moyen, J.F., Stevens, G., and Kaislaniemi, L., 2016, Rcrust: a tool for calculating path-dependent

- open system processes and application to melt loss: *Journal of Metamorphic Geology*, v. 34, p. 663–682, doi: 10.1111/jmg.12199.
- Mayne, M.J., Stevens, G., Moyen, J.F., and Johnson, T.E., 2017, The shape of the P-T path controls the onset of crustal anatexis, *in* Goldschmidt 2017 Abstract,.
- Melwani Daswani, M., and Vance, S.D., 2018a, Melt production, eruption and degassing from Europa’s interior: effects on composition over time, *in* Europa Deep Dive 2: Composition 2018,.
- Melwani Daswani, M., and Vance, S.D., 2018b, Thermal-Compositional Evolution of Europa’s Interior and Ocean Since Accretion, *in* American Geophysical Union,.
- Morrissey, L.J., Hand, M., Lane, K., Kelsey, D.E., and Dutch, R.A., 2016, Upgrading iron-ore deposits by melt loss during granulite facies metamorphism: *Ore Geology Reviews*, v. 74, p. 101–121, doi: 10.1016/j.oregeorev.2015.11.012.
- Morse, S.A., 1984, Cation Diffusion in Plagioclase Feldspar: *Science*, v. 225, p. 504–505.
- Nabelek, P.I., Russ-Nabelek, C., and Haeussler, G.T., 1992, Stable isotope evidence for the petrogenesis and fluid evolution in the Proterozoic Harney Peak leucogranite, Black Hills, South Dakota: *Geochimica et Cosmochimica Acta*, v. 56, p. 403–417, doi: 10.1016/0016-7037(92)90141-5.
- Nicoli, G., and Dyck, B., 2018, Exploring the metamorphic consequences of secular change in the siliciclastic compositions of continental margins: *Geoscience Frontiers*, p. 1–9, doi: 10.1016/j.gsf.2017.12.009.
- Nicoli, G., Stevens, G., Moyen, J.F., Vezinet, A., and Mayne, M.J., 2017, Insights into the complexity of crustal differentiation: K₂O-poor leucosomes within metasedimentary migmatites from the Southern Marginal Zone of the Limpopo Belt, South Africa.: *Journal of Metamorphic Geology*, doi: 10.1111/jmg.12265.
- Ousterhout, J.K., and Jones, K., 2009, Tcl and the Tk toolkit:.
- Pateiro-López, B., and Rodríguez-Casal, A., 2010, Generalizing the Convex Hull of a Sample : The R Package Alphahull: *Journal of Statistical software*, v. 34, p. 1–28, doi: <http://dx.doi.org/10.18637/jss.v034.i05>.
- Powell, R., and Holland, T., 1988, An internally consistent dataset with uncertainties and correlations: 3. Applications to geobarometry, worked examples and a computer program: *Journal of Metamorphic Geology*, v. 6, p. 173–204, doi: 10.1111/j.1525-1314.1988.tb00415.x.
- Powell, R., and Holland, T.J.B., 1985, An internally consistent thermodynamic dataset with uncertainties and correlations: 1. Methods and a worked example.: *Journal of Metamorphic Geology*, v. 3, p. 327–342.
- Powell, R., Holland, T., and Worley, B., 1998, Calculating phase diagrams involving solid solutions via non-linear equations, with examples using THERMOCALC: *Journal of Metamorphic Geology*, v. 16, p. 577–588, doi: 10.1111/j.1525-1314.1998.00157.x.
- R Core Team, 2016, R: A Language and Environment for Statistical Computing:., <https://www.r-project.org/>.
- Riesco, M., Stüwe, K., Reche, J., and Martinez, F.J., 2004, Silica depleted melting of pelites. Petrogenetic grid and application to the Susqueda aureole, Spain: *Journal of Metamorphic Geology*, v. 22, p. 475–494, doi: 10.1111/j.1525-1314.2004.00527.x.
- Rosenberg, C.L., and Handy, M.R., 2005, Experimental deformation of partially melted granite revisited: Implications for the continental crust: *Journal of Metamorphic Geology*, v. 23, p. 19–28, doi: 10.1111/j.1525-1314.2005.00555.x.
- RStudio, 2014, Shiny: Easy web applications in R:., <http://shiny.rstudio.com>.
- de Saint Blanquat, M., Horsman, E., Habert, G., Morgan, S., Vanderhaeghe, O., Law, R., and Tikoff, B., 2011, Multiscale magmatic cyclicity , duration of pluton construction, and the paradoxical relationship between tectonism and plutonism in continental arcs: *Tectonophysics*, v. 500, p. 20–33, doi: 10.1016/j.tecto.2009.12.009.
- Sandeman, H.A., and Clark, A.H., 2003, Glass-rich, Cordierite–Biotite Rhyodacite, Valle Ninahuisa, Puno, SE Peru: Petrological Evidence for Hybridization of “Lachlan S-type” and Potassic Mafic Magmas: *Journal of Petrology*, v. 44, p. 355–385, doi: 10.1093/petrology/44.2.355.
- Scailliet, B., France-Lanord, C., and Le Fort, P., 1990, Badrinath-Gangotri plutons (Garhwal, India): petrological and geochemical evidence for fractionation processes in a high Himalayan leucogranite: *Journal of Volcanology and Geothermal Research*, v. 44, p. 163–188, doi:

10.1016/0377-0273(90)90017-A.

- Scheepers, R., 1990, Magmatic association and radioelement geochemistry of selected Cape Granites with special reference to subalkaline and leucogranitic phases: Stellenbosch University, 151 p.
- Scheepers, R., and Armstrong, R., 2002, New U-Pb SHRIMP zircon ages of the Cape Granite Suite: implications for the magmatic evolution of the Saldania Belt: *South African Journal of Geology*, v. 105, p. 241–256.
- Scheepers, R., and Poujol, M., 2002, U-Pb zircon age of Cape Granite Suite ignimbrites: characteristics of the last phases of the Saldanian magmatism: *South African Journal of Geology*, v. 105, p. 163–178, doi: 10.2113/105.2.163.
- Schwindinger, M., 2018, The Birthplace of Granite. Mechanisms controlling the evolution of granite magma within the source : Examples from Kangaroo Island, South Australia: 135 p.
- Schwindinger, M., Weinberg, R.F., and Clos, F., 2018, Wet or dry? The difficulty of identifying the presence of water during crustal melting: *Journal of Metamorphic Geology*, v. Submitted.
- Sola, A.M., Hasalova, P., Weinberg, R.F., Suzano, N.O., Becchio, R.A., Hongn, F.D., and Botelho, N., 2017, Low-P melting of metapelitic rocks and the role of H₂O : Insights from phase equilibria modelling: *Journal of Metamorphic Geology*, v. 35, p. 1131–1159, doi: 10.1111/jmg.12279.
- Solgadi, F., Moyen, J.F., Vanderhaeghe, O., Sawyer, E.W., and Reisberg, L., 2007, The role of crustal anatexis and mantle-derived magmas in the genesis of synorogenic Hercynian granites of the Livradois area, French massif central: *Canadian Mineralogist*, v. 45, p. 581–606, doi: 10.2113/gscanmin.45.3.581.
- Soorajlal, R., 2017, An investigation of the petrogenesis of the Buddusò I-type granites and its mafic enclaves in Sardinia, Italy.
- Spear, F.S., Kohn, M.J., and Cheney, J.T., 1999, P -T paths from anatexitic pelites: *Contributions to Mineralogy and Petrology*, v. 134, p. 17–32.
- Spear, F.S., and Pattison, D.R.M., 2017, The implications of overstepping for metamorphic assemblage diagrams (MADs): *Chemical Geology*, v. 457, p. 38–46, doi: 10.1016/j.chemgeo.2017.03.011.
- Spear, F.S., Pattison, D.R.M., and Cheney, J.T., 2016, The metamorphism of metamorphic petrology: *Geological Society of America Special Paper*, v. 523, p. 31–74, doi: 10.1130/2016.2523(02).
- Stevens, G., and Clemens, J.D., 1993, Fluid-absent melting and the roles of fluids in the lithosphere: a slanted summary? *Chemical Geology*, v. 108, p. 1–17.
- Stevens, G., Clemens, J.D., and Droop, G.T.R., 1995, Hydrous cordierite in granulites and crustal magma production: *Geology*, v. 23, p. 925–928, doi: 10.1130/0091-7613(1995)023<0925:HCIGAC>2.3.CO.
- Stevens, G., Clemens, and Droop, G.T.R., 1997, Melt production during granulite-facies anatexis: experimental data from “primitive” metasedimentary protoliths: *Contributions to Mineralogy and Petrology*, v. 128, p. 352–370.
- Stevens, G., Clemens, J.D., and Mayne, M.J., 2017, The origins of geochemical trends in felsic igneous rocks : Insight into processes in the source., *in* Goldschmidt 2017 Abstract,.
- Stevens, G., and Reenen, D. van R., 1992, Partial melting and the origin of metapelitic granulites in the Southern Marginal Zone of the Limpopo Belt, South Africa: *Precambrian Research*, v. 55, p. 303–319.
- Stevens, G., Villaros, A., and Moyen, J.F., 2007, Selective peritectic garnet entrapment as the origin of geochemical diversity in S-type granites: *Geology*, v. 35, p. 9–12, doi: 10.1130/G22959A.1.
- Stuck, T.J., 2016, Mineral Equilibria Constraints on Open-System Melting and Consequences of Melt Loss in Metabasic Rocks: 61 p.
- Stuck, T.J., and Diener, J.F.A., 2018, Mineral equilibria constraints on open-system melting in metamafic compositions: *Journal of Metamorphic Geology*, v. 36, p. 255–281, doi: 10.1111/jmg.12292.
- Tajčmanová, L., Konopásek, J., and Connolly, J.A.D., 2007, Diffusion-controlled development of silica-undersaturated domains in felsic granulites of the Bohemian Massif (Variscan belt of Central Europe): *Contributions to Mineralogy and Petrology*, v. 153, p. 237–250, doi: 10.1007/s00410-006-0143-y.
- Taylor, J., Nicoli, G., Stevens, G., Frei, D., and Moyen, J., 2014, The processes that control leucosome compositions in metasedimentary granulites : perspectives from the Southern Marginal Zone

- migmatites, Limpopo Belt, South Africa: *Journal of Metamorphic Geology*, v. 32, p. 713–742, doi: 10.1111/jmg.12087.
- Taylor, J., and Stevens, G., 2010, Selective entrainment of peritectic garnet into S-type granitic magmas: Evidence from Archaean mid-crustal anatectites: *Lithos*, v. 120, p. 277–292, doi: 10.1016/j.lithos.2010.08.015.
- Thompson, A.B., 1982, Dehydration melting of pelitic rocks and the generation of H₂O-undersaturated granitic liquids: *American Journal of Science*, v. 282, p. 1567–1595.
- Thompson, A.B., 1983, Fluid-absent metamorphism: *Journal of the Geological Society, London*, v. 140, p. 533–547.
- Vielzeuf, D., and Holloway, J.R., 1988, Experimental determination of the fluid-absent melting relations in the pelitic system: Consequences for crustal differentiation: *Contributions to Mineralogy and Petrology*, v. 98, p. 257–276, doi: 10.1007/BF00375178.
- Vielzeuf, D., and Montel, J.M., 1994, Partial melting of metagreywackes. Part I. Fluid-absent experiments and phase relationships: *Contributions to Mineralogy and Petrology*, v. 117, p. 375–393, doi: 10.1007/BF00307272.
- Vigneresse, J.L., and Burg, J.P., 2000, Continuous vs. discontinuous melt segregation in migmatites: Insights from a cellular automaton model: *Terra Nova*, v. 12, p. 188–192, doi: 10.1046/j.1365-3121.2000.00299.x.
- Villaros, A., 2004, Fusion partielle d'une croûte hétérogène (metapelites et orthogneiss) à moyenne to haute pression: Grenoble University, 60 p.
- Villaros, A., Laurent, O., Couzinié, S., Moyen, J.F., and Mintrone, M., 2018, Plutons and domes: the consequences of anatectic magma extraction—example from the southeastern French Massif Central: *International Journal of Earth Sciences*, p. 1–24, doi: 10.1007/s00531-018-1630-x.
- Villaros, A., and Stevens, G., 2009, The trace element compositions of S-type granites: evidence for disequilibrium melting and accessory phase entrainment in the source: *Contributions to Mineralogy and Petrology*, v. 158, p. 543–561, doi: 10.1007/s00410-009-0396-3.
- Villaros, A., Stevens, G., and Buick, I.S., 2009, Tracking S-type granite from source to emplacement: Clues from garnet in the Cape Granite Suite: *Lithos*, v. 112, p. 217–235, doi: 10.1016/j.lithos.2009.02.011.
- Ward, R.A., Stevens, G., and Kisters, A., 2008, Fluid and deformation induced partial melting and melt volumes in low-temperature granulite-facies metasediments, Damara Belt, Namibia: *Lithos*, v. 105, p. 253–271.
- Waters, D.J., 1988, Partial melting and the formation of granulite facies assemblages in Namaqualand, South Africa: *Journal of Metamorphic Geology*, v. 6, p. 387–404, doi: 10.1111/j.1525-1314.1988.tb00430.x.
- Webb, G., Powell, R., and McLaren, S., 2015, Phase equilibria constraints on the melt fertility of crustal rocks: The effect of subsolidus water loss: *Journal of Metamorphic Geology*, v. 33, p. 147–165, doi: 10.1111/jmg.12114.
- Weinberg, R.F., and Hasalová, P., 2015, Water-fluxed melting of the continental crust: A review: *Lithos*, v. 212–215, doi: 10.1016/j.lithos.2014.08.021.
- Wheeler, J., 2014, Dramatic effects of stress on metamorphic reactions: *Geology*, v. 42, p. 647–650, doi: 10.1130/G35718.1.
- Wheeler, J., 2018, The effects of stress on reactions in the Earth: Sometimes rather mean, usually normal, always important: *Journal of Metamorphic Geology*, v. 36, p. 439–461, doi: 10.1111/jmg.12299.
- White, R.W., and Powell, R., 2002, Melt loss and the preservation of granulite facies mineral assemblages: *Journal of Metamorphic Geology*, v. 20, p. 621–632, doi: 10.1046/j.1525-1314.2002.00206.x.
- White, R.W., and Powell, R., 2010, Retrograde melt-residue interaction and the formation of near-anhydrous leucosomes in migmatites: *Journal of Metamorphic Geology*, v. 28, p. 579–597, doi: 10.1111/j.1525-1314.2010.00881.x.
- White, R.W., Powell, R., and Clarke, G.L., 2002, The interpretation of reaction textures in Fe-rich metapelitic granulites of the Musgrave Block, Central Australia: Constraints from mineral equilibria calculations in the system: *Journal of Metamorphic Geology*, v. 20, p. 41–55, doi: 10.1046/j.0263-4929.2001.00349.x.

- White, R.W., Powell, R., Holland, T.J.B., Johnson, T.E., and Green, E.C.R., 2014, New mineral activity-composition relations for thermodynamic calculations in metapelitic systems: *Journal of Metamorphic Geology*, v. 32, doi: 10.1111/jmg.12071.
- White, R.W., Powell, R., Holland, T.J.B., and Worley, B., 2000, The effect of TiO₂ and Fe₂O₃ on metapelitic assemblages at greenschist and amphibolite facies conditions: mineral equilibria calculations in the system K₂O-FeO-MgO-Al₂O₃-SiO₂-H₂O-TiO₂-Fe₂O₃: *Journal of Metamorphic Geology*, v. 18, p. 497–511.
- White, R.W., Powell, R., and Phillips, G.N., 2003, A mineral equilibria study of the hydrothermal alteration in mafic greenschist facies rocks at Kalgoorlie, Western Australia: *Journal of Metamorphic Geology*, v. 21, p. 455–468, doi: 10.1046/j.1525-1314.2003.00454.x.
- Whitney, D.L., and Evans, B.W., 2010, Abbreviations for names of rock-forming minerals: *American Mineralogist*, v. 95, p. 185–187, doi: 10.2138/am.2010.3371.
- Williamson, B.J., Downes, H., Thirlwall, M.F., and Beard, A., 1997, Geochemical constraints on restite composition and unmixing in the Velay anatectic granite, French Massif Central: *Lithos*, v. 40, p. 295–319, doi: 10.1016/S0024-4937(97)00033-9.
- Williamson, B.J., Shaw, A., Downes, H., and Thirlwall, M.F., 1996, Geochemical constraints on the genesis of Hercynian two-mica leucogranites from the Massif Central, France: *Chemical Geology*, v. 127, p. 25–42, doi: 10.1016/0009-2541(95)00105-0.
- Yakymchuk, C., 2017, Applying phase equilibria modelling to metamorphic and geological processes: Recent developments and future potential: *Geoscience Canada*, v. 44, p. 27–46, doi: 10.12789/geocanj.2017.44.114.
- Yakymchuk, C., and Brown, M., 2014, Consequences of open-system melting in tectonics: *Journal of the Geological Society*, v. 171, p. 21–40, doi: 10.1144/jgs2013-039.
- Yakymchuk, C., Clark, C., and White, R.W., 2017, Phase Relations, Reaction Sequences and Petrochronology: *Reviews in Mineralogy and Geochemistry*, v. 83, p. 13 LP – 53, doi: 10.2138/rmg.2017.83.2.
- Yakymchuk, C., Kirkland, C.L., and Clark, C., 2018, Th/U ratios in metamorphic zircon: *Journal of Metamorphic Geology*, v. 00, p. 1–23, doi: <https://doi.org/10.1111/jmg.12307>.
- Yardley, B.W.D., 2009, The role of water in the evolution of the continental crust: *Journal of the Geological Society*, London, v. 166, p. 585–600, doi: 10.1144/0016-76492008-101.
- Zeh, A., 2006, Calculation of garnet fractionation in metamorphic rocks, with application to a flat-top, Y-rich garnet population from the Ruhla Crystalline Complex, Central Germany: *Journal of Petrology*, v. 47, p. 2335–2356, doi: 10.1093/petrology/egl046.
- Zuluaga, C.A., Stowell, H.H., and Tinkham, D.K., 2005, The effect of zoned garnet on metapelite pseudosection topology and calculated metamorphic P-T paths: *American Mineralogist*, v. 90, p. 1619–1628, doi: 10.2138/am.2005.1741.

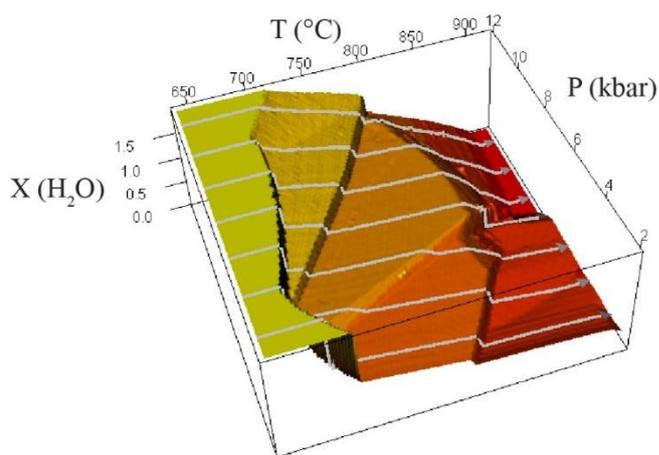
ADDENDA

ADDENDUM A: Electronic copy of Rcrust software

Included within the thesis files for evaluation is an electronic copy of the Rcrust program developed as part of the thesis by M.J. Mayne.

ADDENDUM B: User manual for Rcrust

Included within the thesis files for evaluation is the user manual for the Rcrust program.



Getting started

| | |
|--|----|
| Installation | 3 |
| Concept..... | 5 |
| Example1 – Simple..... | 7 |
| Example2 – Phase extraction..... | 12 |
| Example3 – Multi-path functionality..... | 15 |

Reference manual

| | |
|------------------------------|----|
| Rcrust file management | 18 |
| List of Parameters..... | 19 |
| Useful functions..... | 34 |
| Development | 36 |
| Troubleshooting | 37 |
| References | 37 |

GETTING STARTED

Installation

Rcrust was developed using version 3.3.0 (2016-05-03) of R. Copyright © 2016 the R Foundation for Statistical Computing. To install Rcrust perform the following steps:

1. Copy the Rcrust folder to a location of your choice (preferably a root directory for example C:\ or D:\). The result should be similar to the picture below with all the Rcrust files contained in single directory for example D:\\Rcrust\

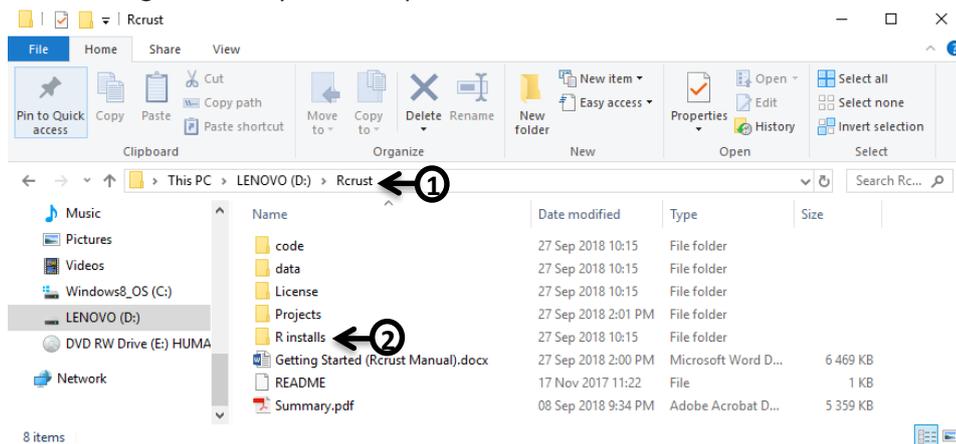


Figure 1 - Rcrust file structure located in the root directory D:\\Rcrust\

2. Install a working version of R on your system (at least version 3.3.0). The latest version of R used in the development of Rcrust is located in the folder "R installs" for your convenience. **Warning:** Rcrust requires the **64 bit** version of R. When installing R please ensure "64-bit Files" is ticked.

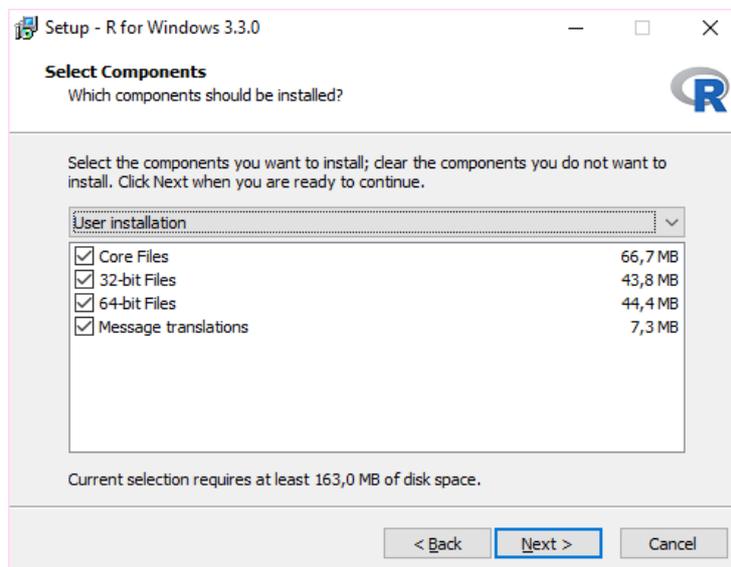


Figure 2 - R installation instruction ensuring at least "64-bit Files" is ticked

*Alternatively newer versions of R (which may not be compatible with Rcrust) can be downloaded from <http://www.r-project.org/> or for windows can be found directly at <http://cran.r-project.org/bin/windows/base/>

3. Open the folder called code in the Rcrust folder.

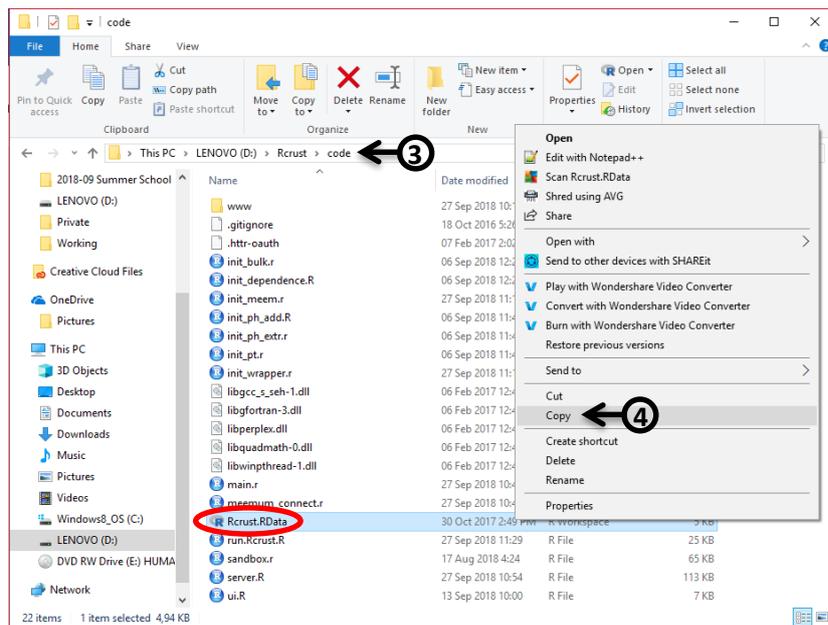


Figure 3 – Rcrust file (circled in red) within code folder. The Rcrust file location here is D:\\Rcrust\\code\\Rcrust.RData

4. Copy the Rcrust file found in the code folder (~\\Rcrust\\code\\Rcrust.RData). This can be done by right clicking on the file (circled in red above) and selecting “Copy” or by selecting the file and pressing “Ctrl”+”c”.
5. Paste the Rcrust file as a **shortcut** on the Desktop. This can be done by right clicking on the Desktop and selecting “Paste shortcut”.

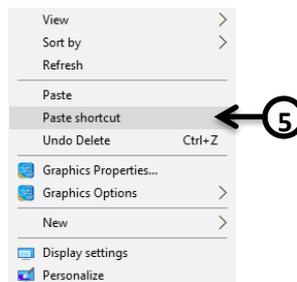


Figure 4 – Paste shortcut option selected for Rcrust file

6. Rename this file to “Rcrust”. Double click on this shortcut to open Rcrust.
7. Rcrust requires the R package called “shiny”, If this is not installed on your computer when you open Rcrust, Rcrust will try to install it (this requires an internet connection). Follow the prompts to complete installation of the package (it may ask you if you wish to create a personal library – choose yes). Alternatively, shiny can be downloaded here: <http://shiny.rstudio.com/>.

Each new project will be automatically saved in the “Projects” folder along with its associated inputs and outputs. To load a previously saved project simply double click the “xxx.RData” file in the associated project folder or open Rcrust from the desktop shortcut and load the project via the Rcrust GUI.

Concept

Rcrust is an R program aimed at modelling with path dependence. The program functions by calculating a number of points in P-T-X space where a bulk composition is passed between points. This creates path dependence as points within the path rely on the outcomes of previous points for their calculation. The bulk composition can be altered at each point by phase manipulations consisting of phase additions and/or phase extractions. Phase stabilities for each point are calculated by using a compiled form of *Perple_X* (Connolly & Kerrick, 1987; Connolly, 2005, 2009).

Rcrust manages calculations by splitting the full thermodynamic system (FS) into 3 subsystems: The reactive subsystem (RS) which contains the phases in thermodynamic equilibrium; The addition subsystem (AS) where phases are waiting to be added to the reactive subsystem; and the extract subsystem (ES) where phases extracted from the reactive subsystem are stored. The reactive subsystem is in thermodynamic equilibrium with the P-T-X conditions of each point and is re-equilibrated after each P-T-X change. The addition and extract subsystems are in thermodynamic isolation from other subsystems and from the P-T-X conditions of each point.

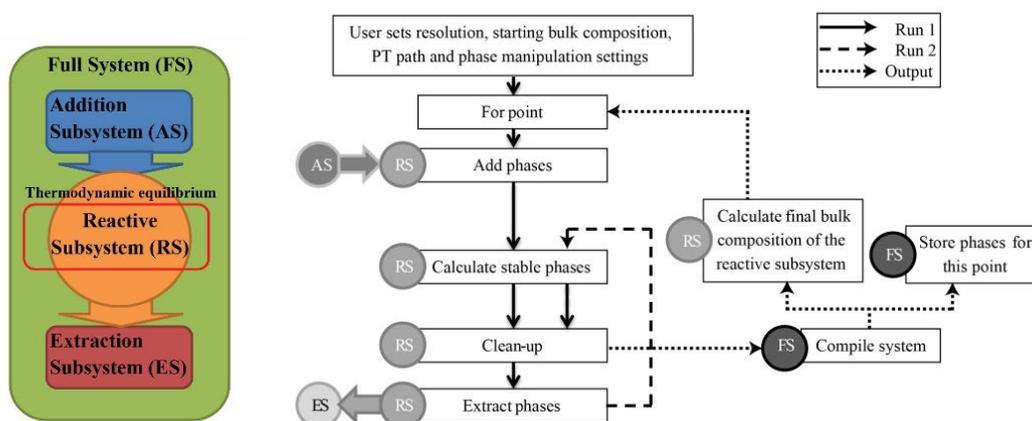


Figure 5 - Relationships between systems (left) and flow chart (right) illustrating the Rcrust program structure for a single path. The user inputs the calculation's resolution, starting bulk composition, P-T path and phase manipulation settings. Each step in a path consists of two runs and an output. The first run is shown in a solid line, the second run in a dashed line and the outputs in a dotted line. Circles show the system or subsystem involved in each step as AS (addition subsystem), ES (extract subsystem), FS (full system) or RS (reactive subsystem). Arrows show interactions between systems. From (Mayne et al., 2016)

Parameters for calculations are accessible to the user via the Rcrust Graphical User Interface (GUI). This GUI writes data to a text file which is then input to the program thus allowing the user to edit the file 'behind' the GUI as well as save inputs for re-use. The code files are extensively commented, and described in this document. The calculations routines are defined in several files, written in a modular way that should allow easy addition of features if required. For example, the Phase Extraction routine has been modified to suit the needs of magma extraction where additional capabilities allow melt extraction to leave a set melt retention amount behind.

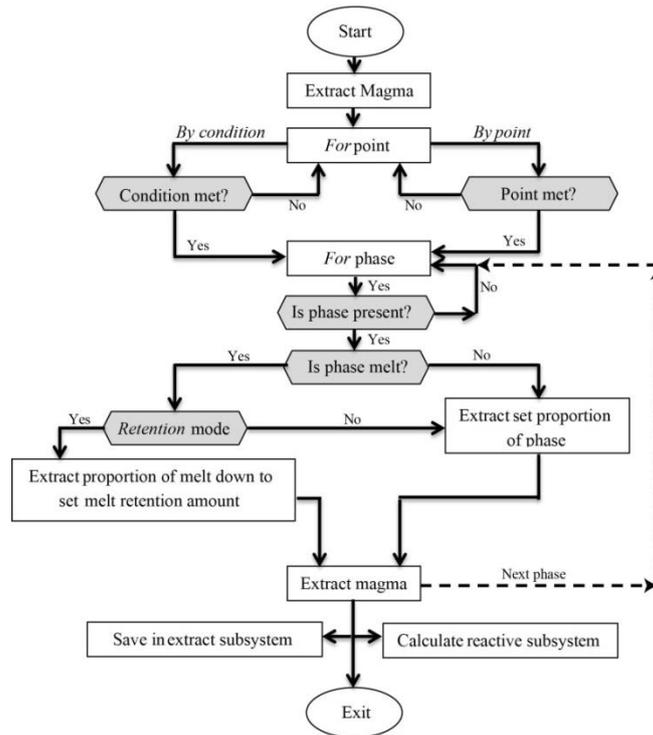


Figure 6 - Flow chart of the magma extraction routine. Grey hexagon shaped boxes are decision points. Coding variables are in italics. The for phase loop (dotted line) is repeated until each phase tagged for extraction has been considered. If Retention mode is active melt is considered last so that other phases extracted are accounted for in its calculation. From (Mayne et al., 2016)

Rcrust results should easily be loaded into GCDkit and examined from there.

It is important to remember a few things:

- Rcrust is in development. It is not mature software. It is very unstable at the best of times, and very unforgiving in terms of improperly formatted inputs, etc. When Rcrust fails, it will try to generate some human-readable error messages: read them! It may well give you hints at things you can correct in your inputs.
- Most of the errors you will see are related to incorrect input (files with incorrect number of lines etc.); or to exotic phases being produced by meemum.

Rcrust calls a set of binary files containing the thermodynamic equations thus relying on published databases (Holland and Powell typically). The output will never be better than the underlying thermodynamic model. Since we focus on melting, we are tied to the capacities (and limitations) of the melt models. For example, melt(HP) does not include Titanium therefore its use in a system containing Ti can over/under estimate melt abundance.

Examples

Below are 3 example simulations to get you started using Rcrust. All you need to do to complete the examples is to perform the actions written in bold numbered text. Explanations of what these actions achieve are given between steps.

Example1 – Simple

Follow the bold numbered steps

To begin the first example open Rcrust via the desktop shortcut.

1. Double click the Rcrust desktop shortcut

This will launch the R console and an empty Rcrust Graphical User Interface (GUI) in your default web browser. The **“Working File”** (circled in red) shows you which file is currently being worked on and the **“Projects Directory”** (circled in green) shows you where the projects folder is located. The Rcrust toolbar (in grey) contains buttons for file management.

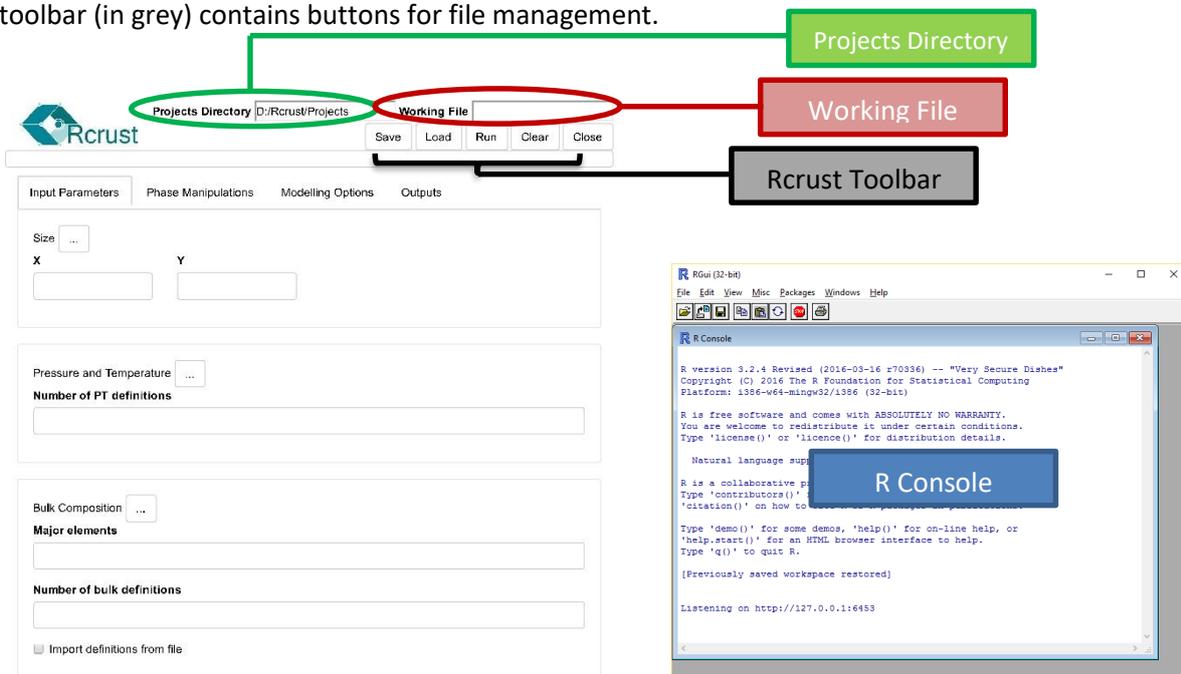


Figure 7 -Rcrust GUI and R Console (blue). Highlighted are the positions of the Projects Directory (green), Working File (red) and Rcrust toolbar (black).

2. Type **“Example1”** into the text box on the right of Working File and then click the Load button from the Rcrust toolbar

The data previously saved in the **“Example1”** file is now loaded into R and previously saved input parameters are loaded into the Rcrust GUI. To ensure that we do not overwrite any data lets rename the Working File.

3. Rename Example1 by typing **“Example_simple”** into the Working File textbox then click the Save button from the Rcrust toolbar

This will save the current Rcrust GUI inputs into a new file named **“Example_simple”**. The Rcrust GUI should now look similar to Figure 8.

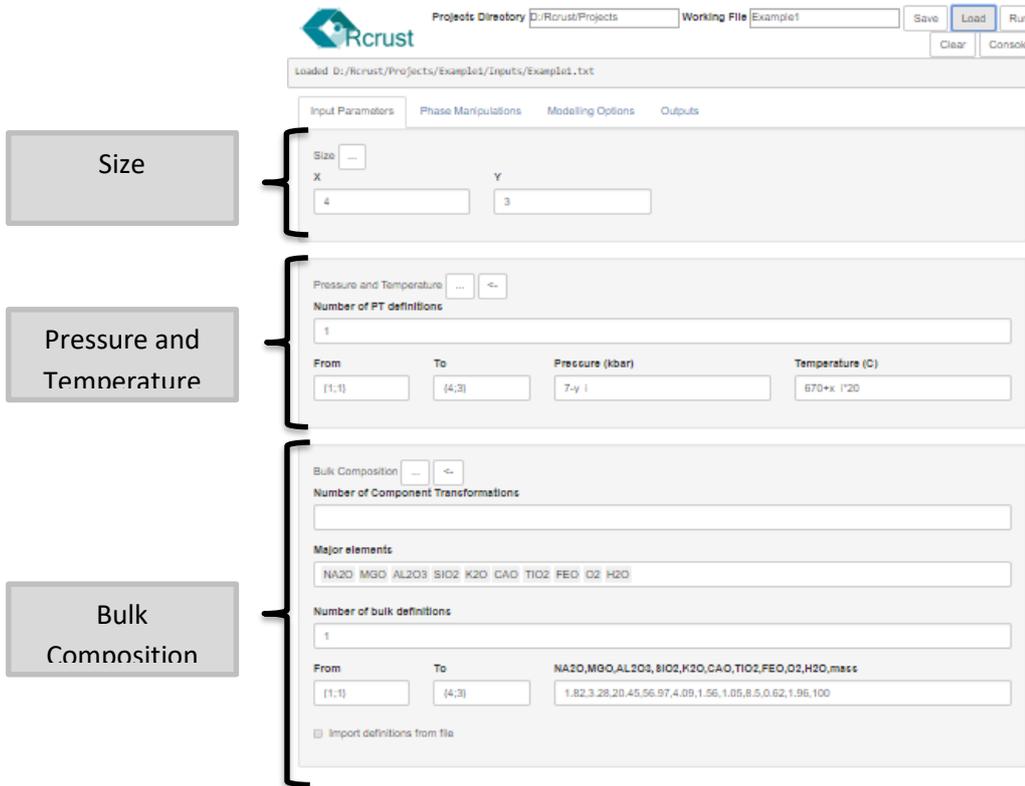


Figure 8 - Rcrust GUI with Example1 (Example_simple) parameters loaded. The GUI consists of a number of tabs. The Input Parameters tab sets the size, P-T conditions and bulk composition (X) of the simulation.

This example calculates the phases encountered at points in P-T-X space. Input parameters are grouped into collapsible panels:

- Size Panel

The **Size** panel sets the number of points in the simulation (here 4 points in the **X** direction multiplied by 3 in the **Y** direction). Points in the simulation space are identified by **tuples** written as $\{x_i ; y_i\}$ where *i* denotes the current point.

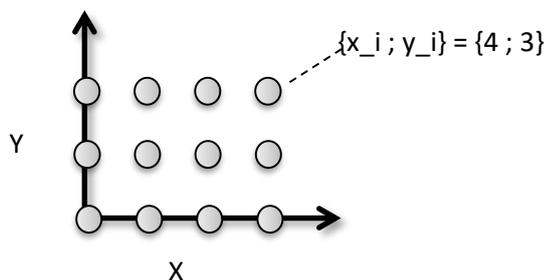


Figure 9 – Simulation space with point coordinates defined by tuples $\{x_i ; y_i\}$

- Pressure and Temperature Panel

Parameters in the simulation space are filled by a number of definitions. Each unique definition is applied over a range between the tuples **From** $\{x_a ; y_a\}$ and **To** $\{x_b ; y_b\}$ where *a* and *b* denote the start and end points of a rectangular range. Each definition sets its attributes as constants or as

functions of the point's position. **X** and **Y** positions of points are accessed by the variables x_i and y_i respectively. In the **Pressure and Temperature Panel** we set the Pressure in kilobars and Temperature in degrees Celsius for each point. In this example **Pressure** decreases along the y-axis by 1 kbar per point ($7-y_i$) and **Temperature** increases along the x-axis by 20 °C per point ($670+x_i*20$).

- Bulk Composition Panel

The **bulk composition** of the system is made up of a number of major elements expressed as wt.% elemental oxides. For this simple example we define the bulk composition to be constant across the full P-T space, and we give it the value of the average amphibolite-facies pelite composition considered in (Mayne et al., 2016).

| | Na ₂ O | MgO | Al ₂ O ₃ | SiO ₂ | K ₂ O | CaO | TiO ₂ | FeO | O ₂ | H ₂ O |
|------|-------------------|------|--------------------------------|------------------|------------------|------|------------------|-----|----------------|------------------|
| wt.% | 1.82 | 3.28 | 20.45 | 56.97 | 4.09 | 1.56 | 1.05 | 8.5 | 0.62 | 1.96 |

Table 1 – Average amphibolite facies pelite composition considered in (Mayne et al., 2016).

To get started let's run a reconnaissance simulation:

4. Click the Run button from the Rcrust toolbar

This will save inputs in the Rcrust GUI and launch the calculation procedure into the R console. Navigate to the R console now. The R console should now have a few lines of text in it (like the figure below): If your simulation successfully initialized like the one below then we are ready to start the calculation. If your console failed to initialize the program try reloading the original "Example1" file by closing Rcrust then starting from step 1 again, if problems persist try reinstalling Rcrust or report the problem to the developers (mjmayne@outlook.com).

```

RGui (32-bit)
File Edit View Misc Packages Windows Help

R Console
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

Listening on http://127.0.0.1:7418

Initializing bulk composition...
Bulk composition defined from inputs...
P-T-X space under investigation with x = 4 and y = 3
Creating bulk compositions from definitions in configuration file
Done with bulk composition preparation
.....
Creating meemum build file...
Created meemum build file as D:/Rcrust/bin/parse_meem.dat
.....
Initializing PT conditions...
Calculating PT conditions from inputs...
Done with PT conditions
.....
No phase addition.
Done with phase addition options
.....
No phase extraction.
Done with phase extraction options
.....
Done with dependence determination
.....
Initiation successful:
Please read the above lines and make sure this is what you wanted.
Choose "n" to abort or press [enter] to continue
|

```

Figure 10 - The Rcrust calculation is launched into the R Console which tracks the calculation progress and is currently waiting for a response to continue or to abort.

5. Click anywhere in the R console to activate it then press [enter] to continue

The calculation will run for 12 points :4 in the X direction by 3 in the Y direction. The results will automatically be saved to file and you will be prompted to select outputs.

6. Once the calculation is complete, navigate back to the Rcrust GUI and Select the Outputs tab

Here we see a compilation data file for the points in our simulation. To compare points, click the “Select Output” drop down and choose “Grid”.

7. Choose “Grid” under “Select Output”

Grid allows easy comparisons between points for example choose:

- Select Output = Grid
 - Variable = wt% and Melt_rs

8. Choose Variable=“wt%” and “Melt_rs”

The output should now match Figure 11. This shows us the amount of melt in the Reactive Subsystem (RS) over our point selection in P-T-X space. Remember we set temperature to increase along the x-axis and pressure to decrease along the y-axis. With that in mind, we see that the solidus (boundary between liquid and solid) at lower pressures is crossed by lower temperatures (this is a good illustration of the pressure dependence of melting). For further outputs you can deal directly with the data in the R console (hint: you can plot data directly into GCDkit). To access the data in R console, click the “Console” button on the Rcrust toolbar to launch a browser access. To return to the Rgui at any point type “c” then press [enter].

Melt_rs wt% on (X,Y) grid

| | V1 | V2 | V3 | V4 |
|---|-------|-------|-------|-------|
| 1 | 6.48 | 19.21 | 21.92 | 25.48 |
| 2 | 12.91 | 14.75 | 17.01 | 21.45 |
| 3 | 5.06 | 13.18 | 14.86 | 16.96 |

Figure 11 – Grid output Data for Example_simple showing weight percentage of melt in the reactive subsystem for the P-T-X points selected

To view a graphical output of this data toggle the “View” selection to “Plot”. This will plot a filled contour graph of the selected data which can be saved directly as a .ps file through the “Save To File” button at the bottom of the selection panel.

9. Toggle the “View” selection from “Data” to “Plot” and choose “Bottom Axis” as “Temperature”, “Left Axis” as “Pressure”

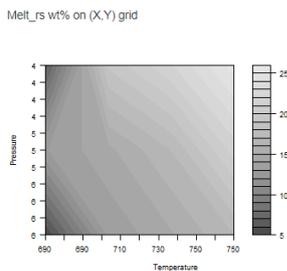


Figure 12 – Grid output Plot for Example_simple showing weight percentage of melt in the reactive subsystem for the P-T-X points selected as a filled contour plot



Example2 – Phase extraction

Phase extractions can remove phases from the reactive subsystem. This is used to simulate scenarios such as melt loss or fractional crystallization.

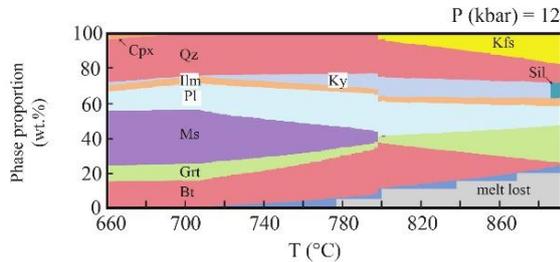


Figure 13 – Example of melt extraction along a P-T-X path from (Mayne et al., 2016): Weight percentage of phases versus temperature in degrees Celsius for a fixed pressure of 12 kbar. Starting composition taken as an average amphibolite facies pelite (Table 1). Melt is extracted whenever a 7 vol.% threshold is met.

Let's perform melt extraction along a path in the P-T-X space explored by Example1.

1. Load Example2 by opening Rcrust, typing 'Example2' in working file and clicking "Load" To ensure that we do not overwrite any data lets rename the Working File.
2. Rename the file by typing "Example_extract" into Working File then click the Save button from the Rcrust toolbar

This will save the current Rcrust GUI inputs into a new file named "Example_extract". The Rcrust GUI should now look like the images below:

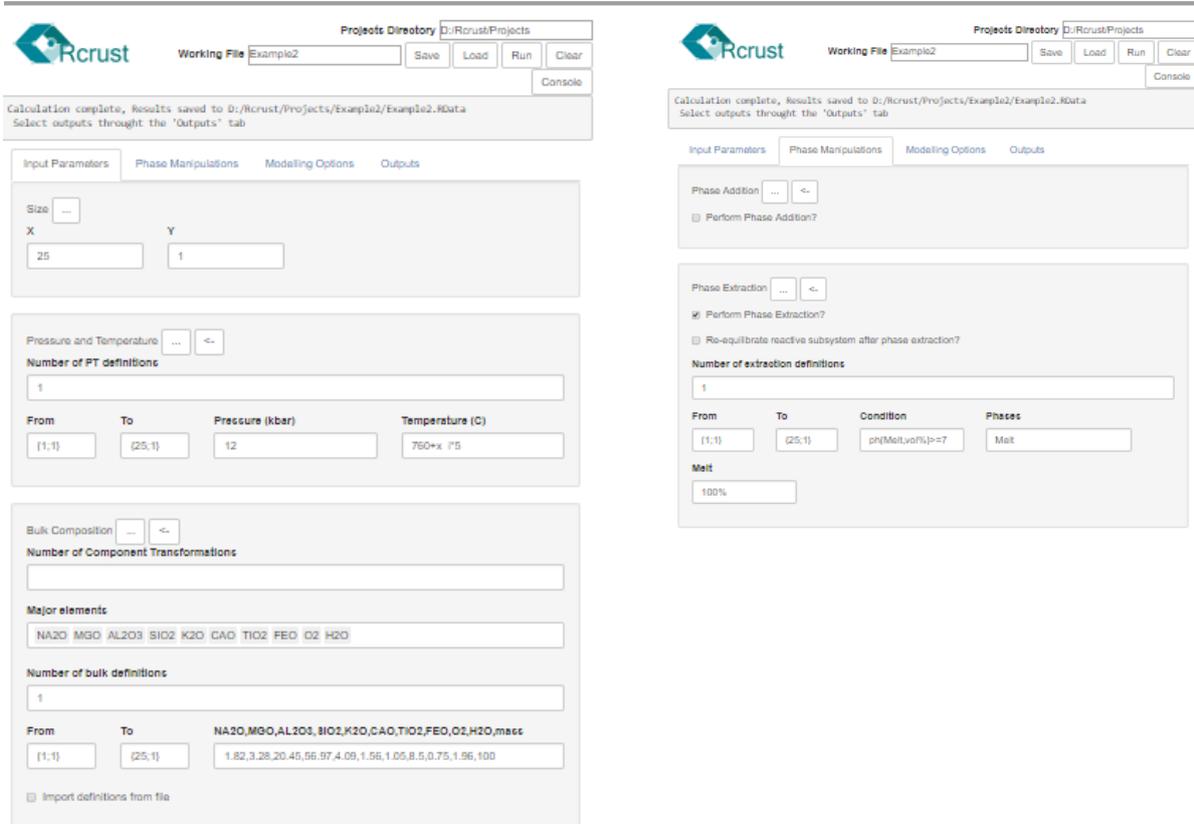


Figure 14 - Rcrust GUI inputs for Example_extract

This example calculates an open system dependent path in P-T-X space. The bulk composition of the Reactive Subsystem is altered by phase manipulations encountered along the path. Phase Manipulations are grouped into collapsible panels:

- Phase Addition

Phases such as intruding fluids, segregated melts or residual crystals can be incorporated into the reactive subsystem. These additions are defined by the major elemental oxides chosen in “Bulk composition”.

- Phase Extraction

Phases in the reactive subsystem can be extracted when set conditions are met. These conditions are defined as logical arguments such as “TRUE”/“FALSE” to extract for every point/no points respectively. Alternatively a logical argument can be built of the form “phase,operand,value,unit” where phase = name of the phase/solution model, operand = (<,<=,==,>=,>,>!), value = a number and unit = the phase property to test. In this example we want to trigger extraction whenever a melt threshold is reached so our conditional argument is “Melt,>=,7,wt%”. *note that each argument in the condition is separated by a comma.

When the condition is met phase extraction is triggered on the reactive subsystem. For each phase listed in **Phases for Extraction** we need to define the amount of the phase to extract. This can be a numeric value (interpreted as grams relative to the starting mass in grams defined in the Bulk Composition Panel), or a percentage of the current value. In this example we extract all melt (100%).

| From | To | Condition | Phases |
|-------|--------|------------------|--------|
| {1;1} | {25;1} | ph{Melt,vol%}>=7 | Melt |

Melt: 100%

Figure 15 - Phase extraction definition for Example_extract

To save you time we have pre-run this calculation so you can directly view the results by selecting the **Outputs** tab.

3. Select the Outputs tab in the Rcrust GUI

A custom output selection is available for viewing phase abundances along a path.

4. Choose “Phase Abundance Along Path” under “Select Output”

Here you can select which axis the path traverses (axis), which path you wish to consider (path), select the (Start Point) and (End Point) of the path and add a label for the column names.

5. Set “End Point” as 25

The output should now match Figure 16.

| Phase abundance vs Temperature(C) for {1;1} to {25;1} | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|
| | 765 | 770 | 775 | 780 | 785 | 790 | 795 | 800 | 805 | 810 | 815 | 820 | 825 | 830 | 835 | 840 | 845 | 850 | 855 | 860 | 865 | 870 | 875 | 880 | 885 | |
| Melt_rs | 3.90 | 4.04 | 4.37 | 4.71 | 5.09 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.20 | 0.59 | 0.95 | 1.35 | 1.66 | 2.32 | 2.89 | 3.48 | 4.38 | 0.00 | 0.73 | 1.62 | 2.37 | 2.44 | |
| Mica_rs | 22.53 | 22.34 | 21.74 | 21.14 | 20.48 | 19.96 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | |
| Bio_rs | 13.49 | 13.49 | 13.63 | 13.79 | 13.93 | 14.09 | 14.19 | 14.19 | 14.19 | 14.19 | 14.19 | 13.70 | 12.92 | 12.15 | 11.30 | 10.68 | 9.47 | 8.43 | 7.34 | 5.85 | 4.14 | 2.83 | 1.23 | 0.00 | 0.00 | |
| Pl_rs | 12.59 | 12.91 | 13.39 | 13.92 | 14.44 | 14.81 | 2.71 | 2.86 | 2.98 | 3.03 | 3.31 | 3.14 | 2.91 | 2.85 | 2.58 | 2.36 | 1.79 | 2.51 | 1.98 | 1.21 | 0.28 | 0.00 | 0.00 | 0.00 | 0.00 | |
| Cpx_rs | 6.27 | 6.05 | 5.72 | 5.35 | 5.00 | 4.74 | 7.84 | 7.74 | 7.64 | 7.62 | 7.36 | 7.27 | 7.08 | 6.76 | 6.61 | 6.49 | 6.39 | 5.86 | 5.80 | 5.81 | 5.80 | 5.55 | 5.19 | 5.02 | 4.74 | |
| Ilm_rs | 7.40 | 7.42 | 7.46 | 7.52 | 7.57 | 7.57 | 7.32 | 7.35 | 7.38 | 7.38 | 7.42 | 7.44 | 7.47 | 7.53 | 7.56 | 7.61 | 7.64 | 7.73 | 7.76 | 7.80 | 7.86 | 7.93 | 8.02 | 8.09 | 8.10 | |
| Ky_rs | 9.37 | 9.41 | 9.46 | 9.47 | 9.53 | 9.61 | 17.57 | 17.55 | 17.52 | 17.52 | 17.38 | 17.29 | 17.17 | 16.95 | 16.78 | 16.69 | 16.52 | 16.16 | 16.02 | 15.84 | 15.68 | 15.44 | 15.10 | 14.84 | 14.76 | |
| Qtz_rs | 24.46 | 24.34 | 24.22 | 24.10 | 23.95 | 23.80 | 19.91 | 19.89 | 19.88 | 19.87 | 19.85 | 19.66 | 19.30 | 18.97 | 18.61 | 18.34 | 17.60 | 17.32 | 16.83 | 16.16 | 15.38 | 14.78 | 14.05 | 13.48 | 13.45 | |
| Kf_rs | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 15.44 | 15.42 | 15.40 | 15.37 | 15.38 | 15.74 | 16.27 | 16.78 | 17.37 | 17.79 | 18.62 | 18.65 | 19.39 | 20.36 | 21.46 | 22.19 | 22.82 | 23.23 | 23.40 | |
| Gt_rs | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.08 | 0.54 | 1.29 | 2.03 | 2.80 | 3.36 | 4.44 | 5.45 | 6.39 | 7.59 | 8.98 | 10.15 | 11.55 | 12.57 | 12.69 | |
| Bulk_rs | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 94.48 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 84.98 | 79.58 | 79.58 | 79.58 | 79.58 | |
| Melt_es | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 5.51 | 9.50 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 5.41 | 0.00 | 0.00 | 0.00 | 0.00 | |
| Bulk_es | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 5.51 | 9.50 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 5.41 | 0.00 | 0.00 | 0.00 | 0.00 | |
| Melt_es_cumul | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 5.51 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 20.41 | 20.41 | 20.41 | 20.41 | 20.41 | |
| Bulk_es_cumul | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 5.51 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 15.01 | 20.41 | 20.41 | 20.41 | 20.41 | 20.41 | |

Phase abundance vs Temperature(C) for {1;1} to {25;1}

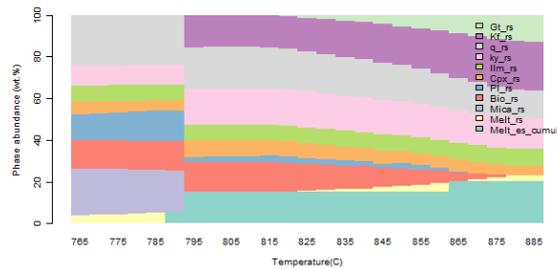


Figure 16 – Phase Abundance Along Path output for Example_extract showing mass of each phase in the full system (FS) across the P-T-X points selected.

This output highlights a few key features of the phase extraction function:

- Outputs in the Phase Abundance along path plotter are expressed as wt.%
- The second melt extraction event extracts more than 7 vol.% melt (relative to the full system)(red box) even though the melt extraction threshold was set to be 7 vol.%

Reason -> Evaluations are only performed at each point thus if the resolution (number of points) is low then large changes can occur between each point.

- Subsequent melt extraction events may appear less than the melt extraction threshold (green boxes)

Reason -> The melt extraction threshold is evaluated relative to the Reactive Subsystem (which itself is shrinking due to melt extraction events) thus equivalent proportions of melt equate to different proportions when compared to the full system (FS).

Example3 - Multi-path functionality

Multiple paths can be compiled in P-T-X space to produce path-dependent P-T mode diagrams. In these diagrams a plane in P-T space is filled with points originating from dependent paths.

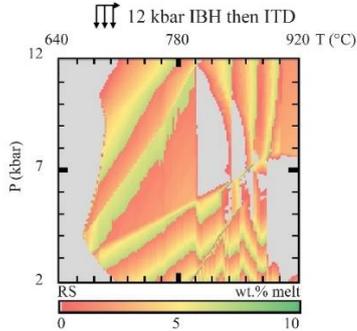


Figure 17 – Example of a composite path-dependent P-T mode diagram from (Mayne et al., 2016): Colours scale the weight percentage of melt in the reactive subsystem (RS). Starting composition at 640 °C and 12 kbar taken as an average amphibolite facies pelite (Table 1). Melt is extracted whenever a 7 wt.% threshold is met and leaves behind 1 wt.% approximating melt retention on grain boundaries. The simulation space is filled by a number of isothermal decompression paths that each originate from a 12 kbar isobaric heating path.

Let's create a path-dependent P-T mode diagram by decompressing off of the path investigated in Example2.

1. Load Example3 by opening Rcrust, typing 'Example3' in working file and clicking "Load" To ensure that we do not overwrite any data lets rename the Working File.

2. Rename the file by typing "Example_multi" into Working File then click the Save button from the Rcrust toolbar

This will save the current Rcrust GUI inputs into a new file named "Example_multi". The Rcrust GUI should now look like Figure 18.

Figure 18 - Rcrust GUI inputs for Example_multi

This example creates a composite path-dependent P-T mode diagram. It does this by first calculating an open system isobaric heating path at 12 kbar (IBH12) and then calculating a number of isothermal decompression paths that each originate from a point on IBH12. Points along IBH12 are each dependent on the reactive subsystem of the point one to the left of itself on the x-axis. Points along decompression paths are each dependent on the point one above itself on the y-axis. Melt loss is defined to occur whenever a 7 vol.% melt threshold is met and melt is extracted until 1 wt.% melt is left behind (this is achieved using the “retain(amount,unit)” function).

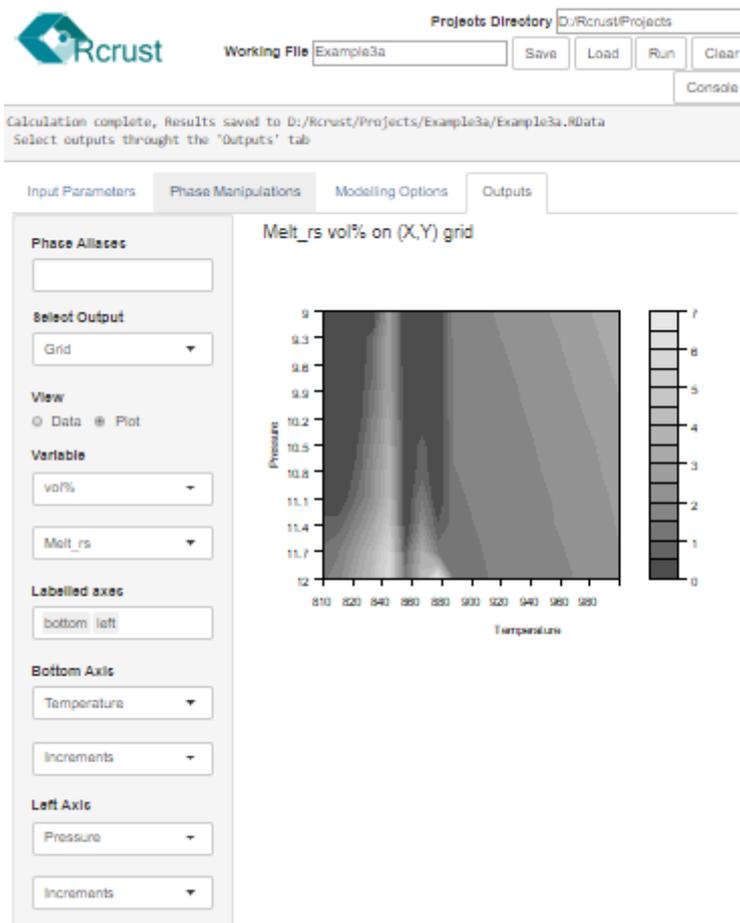


Figure 19 - Grid output for Example_multi showing volume percent of melt in the reactive subsystem (RS)



Projects Directory

Working File Save Load Run Clear

Console

File saved to D:/Rcrust/Projects/Example_multi/Inputs/Example_multi.txt

Input Parameters Phase Manipulations Modelling Options **Outputs**

Phase Aliases

Select Output
PAM

View
 Legend Plot

System
Reactive Subsystem

Field Labels
Numbers
Phases
Numbers

Bottom Axis
Temperature
Increments

Left Axis
Pressure
Increments

Phase Assemblage Map

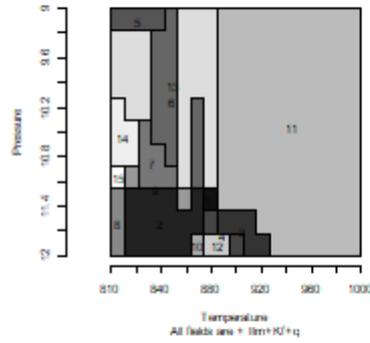


Figure 20 – Phase assemblage map for Example_multi showing field labels as numbers

REFERENCE MANUAL

Rcrust File Management

The top line of the Rcrust GUI hosts a toolbar of file management buttons. User inputs are saved in a text document (**Working File**) which is located in the Projects folder of Rcrust (**Projects Directory**). This file is written, read or run in Rcrust by the **Save**, **Load** and **Run** buttons.



Save

Saves the **Working File's** inputs and calculation results. Each working file is assigned its own project folder in the **Projects Directory**. Parameters currently in the Rcrust GUI are saved to the Inputs folder as a text document. Additional parameters can be passed to Rcrust by placing them in the text document after the line (# Additional Parameters). Calculation results are saved as an R workspace in the project folder.

Load

Loads the **Working File's** inputs and calculation results. Reads the working file from the inputs folder and loads its options in the Rcrust GUI. Replaces the current workspace with that of the **Working File's**.

Run

Saves the current Rcrust GUI inputs and runs the Rcrust calculation according to these parameters. Follow prompts in the R console to calculate the results. Once the results are complete you will be prompted to select outputs through the Rcrust GUI. Outputs written to file are saved in the Outputs folder of the project. Advanced users can access the results directly in the R console by pressing [esc] to activate the console (this is helpful for loading data into GCDkit). To relaunch the Rcrust GUI type 'runApp()' then press [Enter]

Clear

Clears current values in the Rcrust GUI

Console

Launches a browser in the R console giving you direct access to the coding environment and all calculated data

List of Parameters

User inputs are listed here in a systematic fashion for clarity. The parameter name (the name that appears in the Rcrust GUI) is listed first followed by the variable name (the name of the variable accessible in the R console). The data type required for the parameter is listed in the second box. The third box contains possible values for the parameter and identifies any default value. Below this is a description as to what the parameter controls.

| | | |
|--------------------------------------|---------|---------------------------------------|
| Example Parameter {ex_par} | Integer | 0 = closed 1 = open Default = 0 |
| Example definition for the parameter | | |

Annotations: Parameter name, Variable name, Data type, Possible values, Parameter description, Default value.

Input Parameters

Tuple definitions

| | | | |
|---|--|-------|---------------------------------|
| From {pt_from_#} | | Tuple | {1;1}<= pt_from_#<={x_n;y_n} |
| The beginning of the definition selection | | | |

| | | | |
|-------------------------------------|--|-------|-------------------------------|
| To {pt_to_#} | | Tuple | pt_from_#<=pt_to_#<={x_n;y_n} |
| The end of the definition selection | | | |

Size

Specify here the size of the simulation (resolution) you want to calculate: how many points in the X and Y directions.

Size

X Y

| | | |
|---|---------|---------|
| X {x_n} | Numeric | 1 < x_n |
| The total number of points in the X direction | | |

| | | |
|---|---------|---------|
| Y {y_n} | Numeric | 1 < y_n |
| The total number of points in the Y direction | | |

Pressure and Temperature

Pressure and Temperature

Number of PT definitions

From To Pressure (kbar) Temperature (C)

| | | |
|--|---------|--------------|
| Number of PT definitions {n_pt_def} | Numeric | 1 < n_pt_def |
| The number of definitions to use for assigning pressure and temperature values | | |

| | | |
|---|--------------------|-------|
| Pressure (kbar) {pressure_#} | Numeric/Expression | 0 < P |
| The pressure in kilobars of the Reactive Subsystem (RS). *(1 kbar = 0.1 GPa = 986,92 atm). This can be a constant or an expression built using the variables x_i, y_i, real numbers and Arithmetic Operators. e.g. 12-(y_i-1)*0.3 to decompress along the y-axis by 0.3kbar per step starting at 12 kbar. | | |

| | | |
|--|--------------------|-------|
| Temperature (°C) {temperature_#} | Numeric/Expression | 0 < T |
| The temperature in degrees Celsius of the Reactive Subsystem (RS). *(1 °C = 274.15 K = 33.8°F) This can be a constant or an expression built using the variables x_i, y_i, real numbers and Arithmetic Operators. e.g. 660+(x_i-1)*50 to heat along the x-axis by 50°C per step starting at 660°C. | | |

| | | |
|---|--------|----------------|
| PT definition {pt_def} | String | Options: input |
| Advanced setting toggling the PT definition mode. Used to allow PT definition from file. *pt definition from file still to come | | |

| | | |
|---|-------------------|--|
| {pt_definitions} | Listed Definition | |
| Pressure and temperature definitions of the form pt_definitions<-list("{x_a,y_a}_{x_b,y_b}"=c("pressure_#","temperature_#"), "{x_a,y_a}_{x_b,y_b}"=c("pressure_#","temperature_#")) | | |

Additional Settings (main.r)

| | | |
|---|-------------------------|--|
| {PT_restrictions} | Comma-separated-strings | |
| P-T conditions which should not be calculated. Use if projecting array through array where points should be ignored | | |

Bulk composition

| | | |
|---|---------|-------------------------|
| Number of Component Transformations {n_comp_trans} | Numeric | $0 \leq n_comp_trans$ |
|---|---------|-------------------------|

The number of component transformations to apply to the currently available chemical components: the possible components for transformation are set by the thermodynamic data file

| | | |
|-----------------------------------|--------|--|
| Replace component {old_comp_#} | String | |
|-----------------------------------|--------|--|

The current component to replace

| | | |
|-------------------------------|--------|-----------------------------|
| New component {new_comp_#} | String | <6 characters, All capitals |
|-------------------------------|--------|-----------------------------|

The name of the new component. This name must consist of less than 6 characters and must be all in capital letters.

| | | |
|--------------------------|-------------------------|--|
| NA2O,MGO,... {comp_#} | Comma-separated numeric | |
|--------------------------|-------------------------|--|

The value of the new component as a factor of the available components: This must be a string of comma separated numbers of which a total of no-more-than 11 can be non-zero. For example, if we wish to use the component O instead of O2 and our starting chemical components are NA2O,MGO,AL2O3,SiO2,K2O,CAO,TIO2,MNO,FEO,NIO,ZRO2,CL2,O2,H2O,CO2 we would use: 0,0,0,0,0,0,0,0,0,0,0,0,0.5,0,0 If we wish to use FE2O3 we would use: 0,0,0,0,0,0,0,0,2,0,0,0,0.5,0,0

| | | |
|------------------------------------|-------------------------|--|
| Major elements {major_elements} | Comma-separated-strings | |
|------------------------------------|-------------------------|--|

The major element components used to define the bulk composition.

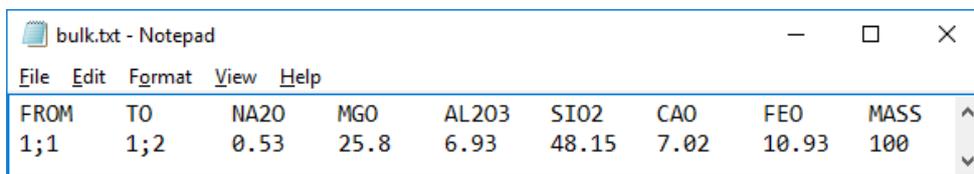
| | | |
|--|---------|-----------------------|
| Number of bulk definitions {n_bulk_def} | Numeric | $1 \leq n_bulk_def$ |
|--|---------|-----------------------|

The number of definitions to use for assigning bulk compositional values.

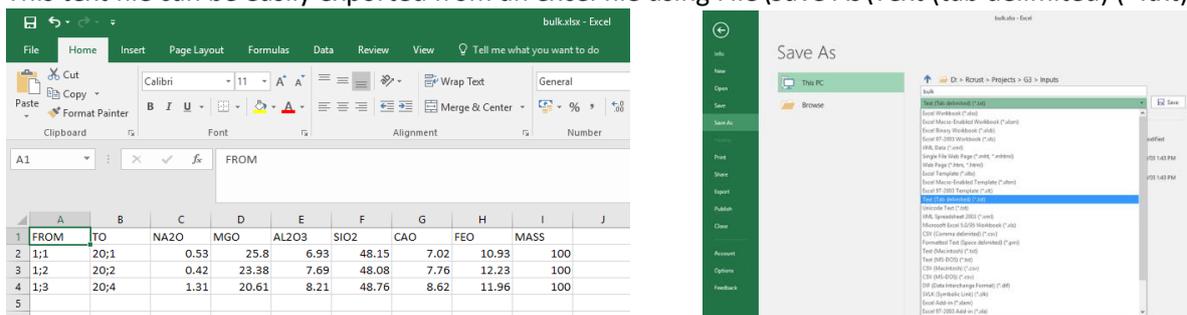
| | | |
|--|------------------------------------|---|
| NA2O,MGO,...,mass | Comma-separated-values/expressions | e.g. 1.5, 20.2, 40.8, 100 e.g. rs{x_i-1;y_i} |
| <p>The wt.% of each elemental oxide listed in “Major elements” (above) as well as the relative starting mass (in grams) of the Reactive Subsystem (RS).</p> <p>Expressions can use real numbers, Arithmetic Operators, x_i, y_i, x_n, y_n and any tuples of the form $xs\{\#\};\#$ where xs is any of rs,as,es,fs and $\#$ uses real numbers, Arithmetic Operators, x_i, y_i, x_n and/or y_n</p> <p>If $xs\{\#\};\#$ is of length 1 it will be repeated to fill the required terms.</p> <p>i.e. To make the full bulk composition of one point dependent on another first initialise the starting composition and then set the dependent points as $rs\{x_i-1; y_i\}$ for dependence on the x-axis or $rs\{x_i ; y_i-1\}$ for dependence on the y-axis</p> | | |

| | | |
|---|---------|---|
| Import definitions from file {bulk_def_file} | Boolean | TRUE = Import from file FALSE = definition via input |
| Choose whether bulk definitions are read from input or imported from a text file (.txt) located in the Inputs folder. | | |

| | | |
|---|------------------|---------------|
| Bulk file {bulk_file} | Character string | e.g. bulk.txt |
| The name of the input file containing the bulk composition definitions (the name must end with .txt). The bulk file must be a tab delimited text file (.txt) containing the columns “From”, “To”, your selection of major elements and “Mass”. For example: | | |



This text file can be easily exported from an excel file using File\Save As\Text (tab delimited) (*.txt)



The input file must provide bulk definitions for all points in the chosen modelling space $\{x_n;y_n\}$ with numbers in the “From” and “To” arguments separated by a semicolon (;). The bulk definitions describe the wt.% of each elemental oxide as well as the relative starting mass (in grams) of the Reactive Subsystem (RS). Expressions can use real numbers, r operators, x_i , y_i , x_n , y_n and any tuples of the form $xs\{\#\};\#$ where xs is any of rs,as,es,fs and $\#$ uses real numbers, r operators, x_i , y_i , x_n and/or y_n

Phase Manipulations

Phase Addition

| | | |
|---|---------|------------|
| Perform Phase Addition? {ph_add} | Boolean | TRUE/FALSE |
| Add phases/components into the Reactive Subsystem (RS) at specified points? | | |

| | | |
|---|---------|--|
| Number of addition definitions {n_ph_add_def} | Numeric | |
| The number of definitions to use for assigning phase additions. | | |

| | | |
|--|--------------------|---|
| Condition {ph_add_con_#} | Logical/expression | Options: TRUE FALSE ph{Melt,vol%}>=7 |
| A conditional argument that evaluates to a Boolean answer of TRUE or FALSE. Note: if directly providing argument as TRUE or FALSE ensure all capitals. | | |
| For example to add phases whenever melt exceeds a 7 vol% threshold you would use the following condition: ph{Melt,vol%}>=7 if "Melt" is the alias assigned in the solution model file used | | |

| | | |
|---|-------------------------|--|
| Phases for addition {ph_add_phs_#} | Comma-separated-strings | |
| Phases to be considered for phase addition in each definition. This can be any name as it is only the respective chemical components that are added to the system not the phase itself, thus the phase name is just a place holder for the user to remember what they are adding. | | |

| | | |
|---|------------------------|--|
| #,#,mass | Comma-separated-values | |
| The wt.% of each elemental oxide listed in "Major elements" (Bulk composition tab) as well as the relative mass (in grams) of the phases/components to add. | | |

Phase Extraction

Phase Extraction

Perform Phase Extraction?

Re-equilibrate reactive subsystem after phase extraction?

Number of extraction definitions

| From | To | Condition | Phases |
|---|---|---|--|
| <input style="width: 50px;" type="text" value="{1;1}"/> | <input style="width: 50px;" type="text" value="{4;1}"/> | <input style="width: 100px;" type="text" value="ph{Melt,wt%}>=7"/> | <input style="width: 50px;" type="text" value="Melt"/> |

Melt

| | | |
|--|---------|------------|
| Perform Phase Extraction? {ph_extr} | Boolean | TRUE/FALSE |
| Extract phases from the Reactive Subsystem (RS) when specified criteria are met? | | |

| | | |
|--|---------|------------|
| Re-equilibrate reactive subsystem after phase extraction? {reequilibrate_steps} | Boolean | TRUE/FALSE |
| Use the bulk composition at the end of extraction to recalculate phase stabilities | | |

| | | |
|---|---------|--|
| Number of extraction definitions {n_ph_extr_def} | Numeric | |
| The number of definitions to use for assigning phase extractions. | | |

| | | |
|--|--------------------|---|
| Condition {ph_extr_con_#} | Logical/expression | Options: TRUE FALSE ph{Melt,vol%}>=7 |
| <p>A conditional argument that evaluates to a Boolean answer of TRUE or FALSE. Note: if directly providing argument as TRUE or FALSE ensure all capitals.</p> <p>Expressions should be of the form described in “Extraction expressions” below.</p> <p>For example to extract phases whenever melt exceeds a 7 vol% threshold you would use the following condition: ph{Melt,vol%}>=7 if “Melt” is the alias assigned in the solution model file used</p> | | |

| | | |
|--|-------------------------|--|
| Phases for extraction {ph_extr_phs_#} | Comma-separated-strings | |
| <p>Phases to be considered for phase extraction in each definition. These can be any alias in the solution model file chosen (Phase Models tab) or any pure phase output by Perple_X (see Perple_X http://www.perplex.ethz.ch/ Perple_X Solution Model Glossary) and THERMOCALC's list of mineral abbreviations).</p> <p>*Can use "any_phase" to set generic arguments</p> | | |

| | | |
|---|-----------------------------------|---|
| [Phase extraction proportions] {ph_extr_phs_#_phase} | Numeric / percentage / expression | e.g. 10% 5 (mass relative to a starting mass of 100 for the full system) retain{2;vol%;Melt} delta{Melt ; x_i-1 ; y_i ; mass}*0.5 |
| <p>Define the proportion of phase to extract for each phase listed in "Phases for extraction". Proportions can be given as:</p> <ol style="list-style-type: none"> 1. A percentage of what is present (e.g. 10%) *you must include the percentage sign for this <p>or</p> <ol style="list-style-type: none"> 2. A set mass relative to the full system (100). If this amount is larger than what is present the full amount of the present phase will be extracted. <p>or</p> <ol style="list-style-type: none"> 3. As functions of the form described in "Extraction expressions" below. <p>*Can use "any_phase" to set generic arguments for all phases that are present but don't already have a phase extraction proportion set i.e. pre existing definitions take precedence.</p> | | |

| | | |
|---|-----------------------------------|--|
| Extraction expressions | Numeric / percentage / expression | |
| <p>Extraction expressions can be evaluated for conditions or proportions consisting of x_i, y_i, real numbers, Arithmetic Operators, Logical Operators, {} and (). where {} separate terms for calculation order and () are reserved for solution model names.</p> <p>The following functions can be called by placing the function name before {} with function terms inside the brackets separated by commas (,) or semi-colons (;).</p> <ul style="list-style-type: none"> • retain{amount ; unit ; phase} <ul style="list-style-type: none"> ○ Retention extracts all but a set amount of a phase from the reactive subsystem where unit can be "mass", "wt%" or "vol%". In phase proportion boxes can omit "phase" to extract current phase. ○ e.g. retain{2;vol%;Melt} will extract melt from the reactive subsystem until 2 vol% of melt remains, this is useful for approximating melt retention on grain boundaries. This can also be written as retain{2;vol%} ○ *Warning: when utilising multiple phase extraction definitions ensure that the retention definition is last in order to retain to the finalised bulk. • delta{phase ; x_a ; y_a ; unit} <ul style="list-style-type: none"> ○ Delta calculates the incremental difference of a phase between the current point (point b) and a previous point (point a) such that: delta=phase_mode_b - phase_mode_a. | | |

*point a may be described as "prev_ext_X" where X is the name of any phase or "prev_ext" to use the name of the extracting phase as default

- phase = phase for extraction ; x_a and y_a describe the position of the previous calculated point and unit is one of "mass" or "wt%".

*phase may be described with "+" separating individual phase terms e.g. to calculate deltas for the alumina silicates phase would be "ky+and+sill"

- e.g. delta{Melt ; x_i-1 ; y_i ; mass}*0.5 to extract half of the delta mass for Melt when paths progress with increasing x_i values.

- ph{phase;unit;x_i;y_i}

- returns the requested variable (unit) for the given phase in the reactive subsystem. Where unit can be any column name in calc_phases and x_i and y_i are the current point by default, if ph{} is only given two arguments the current point will be evaluated by default.

Modelling Options

Modelling Data

Modelling Data

Meemum version

Perple_X Option File

Thermodynamic Data File

Solution Models File

Solution models

| | | |
|--|--------|----------------------|
| Meemum version {meemum_path} | String | Example = meemum.exe |
| The name of the meemum executable to be used in ~Rcrust/data for phase stability calculations. | | |

| | | |
|---|---------|------------------------------|
| Perple_X Option File {perplex_option_file} | Strings | Example = perplex_option.dat |
| The perplex option file present in ~Rcrust/data which controls extra settings for phase stability calculations. | | |

| | | |
|--|--------|-----------------------|
| Thermodynamic Data File {thermodynamic_data_file} | String | Example = hp11ver.dat |
| The thermodynamic data file present in ~Rcrust/data to be used for phase stability calculations. | | |

| | | |
|---|--------|----------------------------------|
| Solution Models File {solution_models_file} | String | Example = solution_model_673.dat |
| The solution model file present in ~Rcrust/data from which solution models can be chosen. | | |

| | | |
|---|-------------------------|--|
| Solution models {use_sol_models} | Comma-separated-strings | |
| The solution models to use in phase stability calculations sourced from the Solution Models File. (see Perple_X http://www.perplex.ethz.ch/Perple_X Solution Model Glossary) | | |

Additional optional parameters

Additional optional parameters

Saturated components

Saturated phase components

Independent potential/fugacity/activity

Exclude phases

| | | |
|--|--------|--|
| Saturated components {saturated_components} | String | |
| Set saturated components | | |

| | | |
|--|--------|--|
| Saturated phase components {saturated_phase_components} | String | |
| Set saturated phase components | | |

| | | |
|--|--------|--|
| Independent potential/fugacity/activity {independent_potential_fugacity_activity} | String | |
| Set independent potentials fugacities or activities | | |

| | | |
|---|--------|--|
| Exclude phases {exclude_phases} | String | |
| Set phases to be excluded from consideration during Gibbs energy minimisation | | |

Extra Settings

Extra Settings

When calculation is complete:

| | | |
|--|--------|--|
| When calculation is complete: {end_of_calc} | String | Options: Return to interface Logout Shutdown Default = Return to interface |
| Action to be performed when a calculation is complete. | | |

Outputs

| | | |
|--|-------------------------|------------------------------|
| Phase Aliases { phase_aliases} | Comma separated strings | e.g. TiBio(HP)=Bt,Gt(WPH)=Gt |
| A list of aliases to use for renaming phases of the form TiBio(HP)=Bt,Gt(WPH)=Gt,etc. To hide a phase from plotting use the alias "hide" e.g. TiBio(HP)=hide. | | |

Select Output = Data File

View the output data as a single filtered table

Projects Directory D:\Rcrust\Projects Working File Example_multi Save Load Run Clear Console

File saved to D:\Rcrust\Projects\Example_multi\Inputs\Example_multi.txt

Input Parameters Phase Manipulations Modelling Options **Outputs**

Phase Aliases

Select Output: Data File

View: Data

Select Columns: All

Select System/Phase: All

Select Points: (1,1)

File type: csv

Save To File

Compilation data file

| ID | Phase | y_i | x_j | Pressure(kbar) | Temperature(C) | mass | wt% | vol% | mol% | |
|----|-------------------|---------------|-----|----------------|----------------|------|--------|-------|-------|-------|
| 1 | Melt_rs_1_1 | Melt_rs | 1 | 1 | 12 | 810 | 0.8488 | 1 | 1.361 | 1.15 |
| 2 | Bio_rs_1_1 | Bio_rs | 1 | 1 | 12 | 810 | 14.35 | 16.91 | 17.44 | 5.235 |
| 3 | Kf_rs_1_1 | Kf_rs | 1 | 1 | 12 | 810 | 14.57 | 17.17 | 20.13 | 8.575 |
| 4 | Pl_rs_1_1 | Pl_rs | 1 | 1 | 12 | 810 | 3.47 | 4.069 | 4.67 | 2.087 |
| 5 | Cpu_rs_1_1 | Cpu_rs | 1 | 1 | 12 | 810 | 7.24 | 8.532 | 7.531 | 5.203 |
| 6 | Ilm_rs_1_1 | Ilm_rs | 1 | 1 | 12 | 810 | 7.41 | 8.732 | 5.284 | 7.74 |
| 7 | Ky_rs_1_1 | Ky_rs | 1 | 1 | 12 | 810 | 17.27 | 20.35 | 16.88 | 17.18 |
| 8 | Qz_rs_1_1 | Qz_rs | 1 | 1 | 12 | 810 | 19.7 | 23.22 | 26.7 | 52.63 |
| 9 | Bulk_rs_1_1 | Bulk_rs | 1 | 1 | 12 | 810 | 84.86 | 100 | 100 | 100 |
| 10 | Melt_es_1_1 | Melt_es | 1 | 1 | 12 | 810 | 15.14 | 0 | 0 | 0 |
| 11 | Bulk_es_1_1 | Bulk_es | 1 | 1 | 12 | 810 | 15.14 | 0 | 0 | 0 |
| 12 | Melt_es_cumul_1_1 | Melt_es_cumul | 1 | 1 | 12 | 810 | 15.14 | 0 | 0 | 0 |
| 13 | Bulk_es_cumul_1_1 | Bulk_es_cumul | 1 | 1 | 12 | 810 | 15.14 | 0 | 0 | 0 |

Select Output = Grid

View the output data by selecting a single variable to express on a grid of x_i and y_i coordinates.
Contour plotting can be achieved by selecting View>Plot.

Projects Directory D:\Rcrust\Projects Working File Example_multi Save Load Run Clear Console

File saved to D:\Rcrust\Projects\Example_multi\Inputs\Example_multi.txt

Input Parameters Phase Manipulations Modelling Options **Outputs**

Phase Aliases

Select Output: Grid

Data File

Grid

Phase Abundance Along Path: PAM

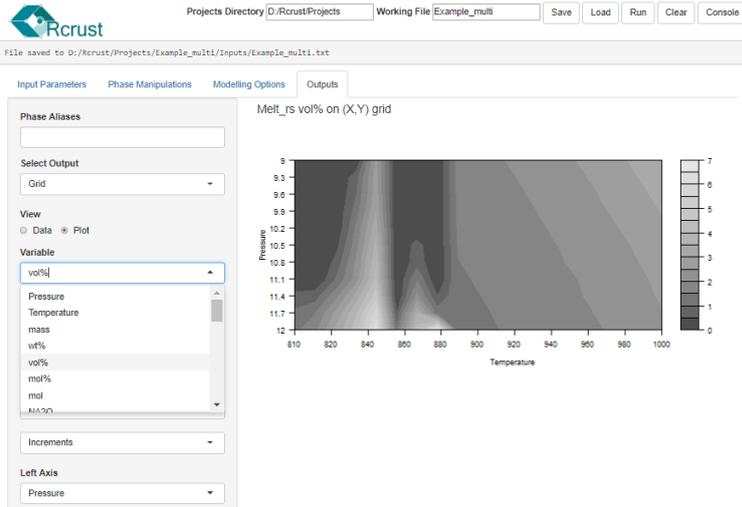
vol%

Melt_rs

Labelled axes: bottom left

Melt_rs vol% on (X,Y) grid

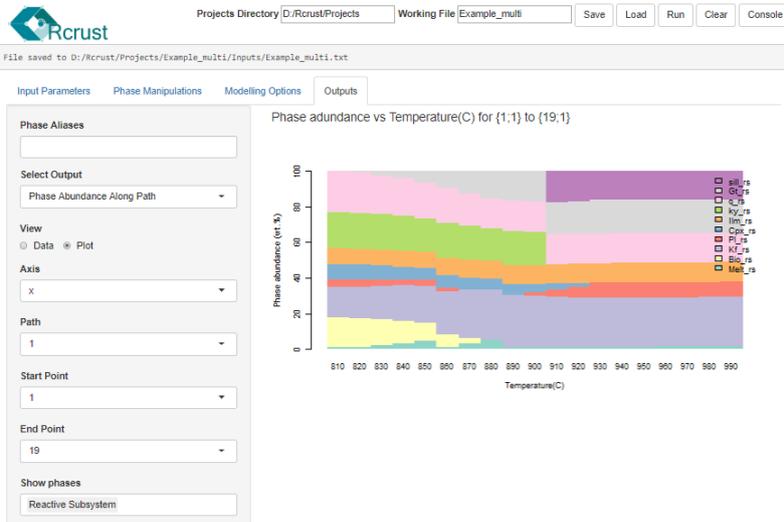
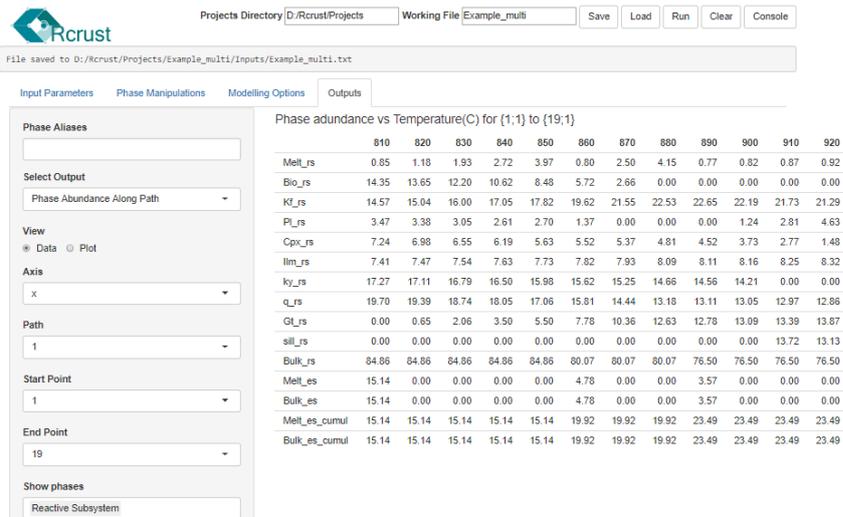
| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 0.00 | 0.00 | 0.00 | 0.00 | 2.63 | 0.00 | 0.00 | 0.00 | 1.76 | 1.86 | 1.97 | 2.08 | 2.21 | 2.38 |
| 2 | 0.00 | 0.00 | 0.00 | 0.78 | 2.81 | 0.00 | 0.00 | 0.00 | 1.71 | 1.81 | 1.91 | 2.02 | 2.14 | 2.28 |
| 3 | 0.00 | 0.00 | 0.00 | 0.96 | 2.83 | 0.00 | 0.00 | 0.00 | 1.66 | 1.73 | 1.86 | 1.96 | 2.08 | 2.21 |
| 4 | 0.00 | 0.00 | 0.00 | 1.20 | 3.01 | 0.00 | 0.00 | 0.00 | 1.60 | 1.70 | 1.80 | 1.91 | 2.02 | 2.14 |
| 5 | 0.00 | 0.00 | 0.00 | 1.57 | 3.29 | 0.00 | 0.17 | 0.00 | 1.57 | 1.66 | 1.73 | 1.88 | 1.96 | 2.08 |
| 6 | 0.00 | 0.00 | 0.31 | 1.78 | 3.70 | 0.00 | 0.70 | 0.00 | 1.53 | 1.61 | 1.68 | 1.78 | 1.90 | 2.02 |
| 7 | 0.00 | 0.00 | 0.61 | 2.06 | 3.91 | 0.00 | 0.79 | 0.00 | 1.49 | 1.59 | 1.66 | 1.75 | 1.88 | 1.96 |
| 8 | 0.00 | 0.00 | 1.00 | 2.25 | 4.20 | 0.00 | 1.48 | 0.00 | 1.45 | 1.53 | 1.61 | 1.70 | 1.80 | 1.90 |
| 9 | 0.69 | 0.78 | 1.81 | 3.14 | 4.81 | 0.00 | 2.17 | 0.53 | 1.41 | 1.49 | 1.57 | 1.66 | 1.75 | 1.85 |
| 10 | 1.11 | 1.46 | 2.60 | 3.95 | 5.75 | 0.60 | 3.51 | 1.36 | 1.38 | 1.45 | 1.53 | 1.63 | 1.72 | 1.80 |
| 11 | 1.36 | 1.88 | 3.07 | 4.30 | 6.24 | 1.36 | 4.22 | 6.97 | 1.36 | 1.45 | 1.50 | 1.57 | 1.68 | 1.77 |



Select Output = Phase Abundance Along Path

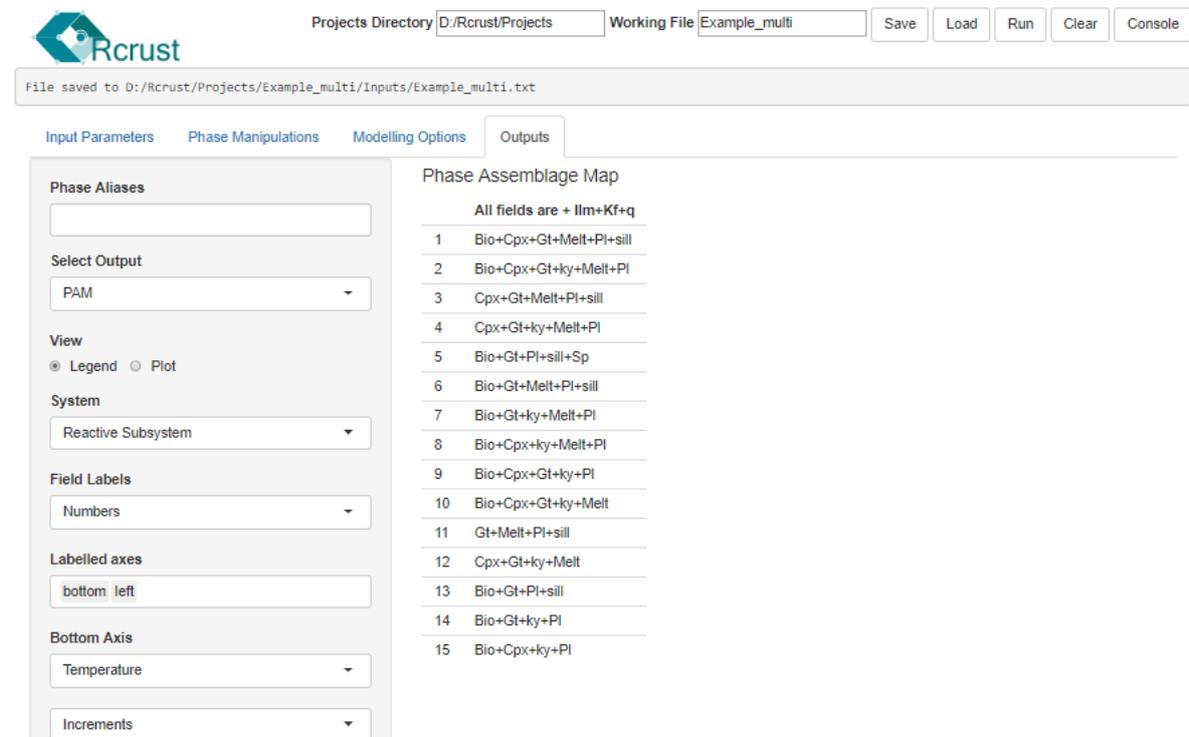
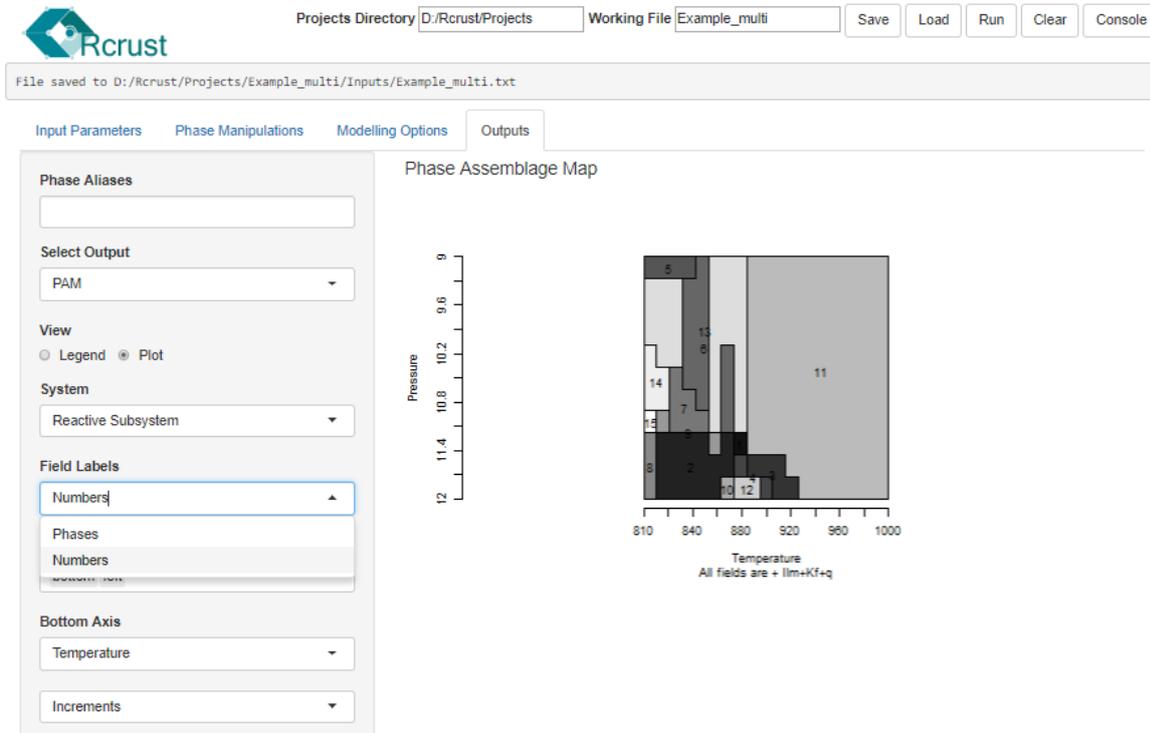
View phase abundance data by weight along a path in the x or y direction.

Phase abundance graphs can be viewed as 100% stacked column graphs by selecting View>Plot



Select Output = PAM

Create a phase assemblage map by merging fields in x-y space with identical assemblages.



File Management

Functions for file management. See Rcrust File management for more details

| | | |
|---|--------|--|
| Projects Directory {projects_directory} | String | |
| Absolute location of the projects directory within the Rcrust folder. | | |

| | | |
|--|--------|--|
| Working File {working_file} | String | |
| Name of the current file under operation. Each file has its own folder within the projects directory containing results of simulations (#.RData file), "Inputs" and "Outputs". | | |

| | | |
|---|---------------|--|
| Save {on_save} | Function call | |
| Saves the current GUI inputs and workspace to the working_file. | | |

| | | |
|--|---------------|--|
| Load {on_load} | Function call | |
| Loads the previously saved working_file into the GUI inputs and workspace. | | |

| | | |
|--|---------------|--|
| Run {manual_load} | Function call | |
| Saves the current GUI inputs and workspace to the working_file. Then launches the Rcrust calculation | | |

| | | |
|--|---------------|--|
| Clear {on_clear} | Function call | |
| Clears the current GUI inputs and workspace. | | |

| | | |
|--|---------------|--|
| Console {stopApp} | Function call | |
| Closes the current GUI to allow interaction with the R console. To relaunch the Rcrust GUI type 'runApp()' then press [Enter]. | | |

Perple_x options

Options parsed to wrapper calculation set in init_meem.r

| | | |
|---|---------|--------------|
| Number of chemical components {number_components} | Integer | Default = 15 |
| The number of chemical components to build the major elements from. | | |

| | | |
|---|---------|--|
| Unit for bulk composition definition | Integer | 0 = molar % 1 = weight % Default = 1 |
| The unit proportion to use for bulk composition definition. | | |

Advanced user options

Static variable options accessible through main.r

| | | |
|---|------------------|----------------------------|
| Calculation mode { calc_mode } | Character vector | normal Default = normal |
| Advanced setting toggling the calculation mode. | | |

| | | |
|---|---------|--|
| Reaction buffering { reaction_buffering } | Boolean | T (TRUE) F (FALSE) Default = FALSE |
| Allows reaction buffering (threshold buffering) whereby phase extractions set on conditions are postponed by the number of reaction buffer steps to ensure continued exceedance of the threshold. | | |

Useful functions in the R Console

`c` Continue steps in a browser (if accessed by clicking “Console” in the Rcrust GUI then closes the browser and returns to the Rcrust GUI)

`ls()` List all objects in the current environment

`ls(envir=.GlobalEnv)` List all objects in the global environment

`Q()` Quits the current session

`[Ctrl]+[w]` Toggles buffering of outputs

`Rcrust ()` Manually launches the Rcrust GUI

To access the R console out of a browser click anywhere in the R console window and press [esc]. This will close the Rcrust GUI which has current control over the console. To relaunch the Rcrust GUI at any time simply type `runApp()` and press [enter].

Rcrust variables

| | |
|--|------|
| <code>PT [[bulk]] [[step]] \$press\$temp</code> | list |
| List of pressure and temperature conditions for each step in each bulk | |

| | |
|---|------|
| <code>crust [[bulk]] [[step]] [phase, detail]</code> | list |
| The full system (FS). Contains details of the reactive subsystem (RS) at each step along with cumulative extract (ES) and addition (AS) subsystems. Phases in crust are reported as cumulative weighted averages. | |

| | |
|--|--------|
| <code>c0 [detail]</code> | vector |
| Bulk composition passed between points | |

| | |
|--------------------------|------------------|
| <code>workingfile</code> | Character vector |
| The current Working File | |

| | |
|---|------------------|
| <code>work_dir</code> | Character vector |
| The current Working Directory. This is the location of the folder containing the Working File | |

Running Rcrust

| | | |
|--|---------------|--|
| Relaunch GUI {.First()} or {Rcrust()} | Function call | |
| Relaunches the Rcrust GUI from the R console | | |

| | | |
|---|---------------|--|
| Manually initiate calculation {manual_load} | Function call | |
| Sends the current working file to be calculated | | |

Rcrust Outputs

| | | |
|---|---------------|--|
| Data file {data_file()} | Function call | |
| Compiles calculation results into a table | | |

| | | |
|--------------------------------------|---------------|--|
| Write data file {write_data_file} | Function call | |
| Writes compilation table to file | | |

| | | |
|--|---------------|--|
| Grid data {grid_data()} | Function call | |
| Compiles an X Y grid of the values of a given variable | | |

| | | |
|--------------------------------------|---------------|--|
| Write grid file {write_grid_file} | Function call | |
| Writes X Y grid to file | | |

R Syntax

| Arithmetic Operators | | | | | | | | | | | | | | | | | | |
|---|-----------------------------|--|----------|-------------|---|----------|---|-------------|---|----------------|---|----------|---------|----------------|--------|-----------------------------|---------|-----------------------------|
| <table border="1"> <thead> <tr> <th>Operator</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>+</td> <td>addition</td> </tr> <tr> <td>-</td> <td>subtraction</td> </tr> <tr> <td>*</td> <td>multiplication</td> </tr> <tr> <td>/</td> <td>division</td> </tr> <tr> <td>^ or **</td> <td>exponentiation</td> </tr> <tr> <td>x %% y</td> <td>modulus (x mod y) 5%%2 is 1</td> </tr> <tr> <td>x %/% y</td> <td>integer division 5%/%2 is 2</td> </tr> </tbody> </table> | | | Operator | Description | + | addition | - | subtraction | * | multiplication | / | division | ^ or ** | exponentiation | x %% y | modulus (x mod y) 5%%2 is 1 | x %/% y | integer division 5%/%2 is 2 |
| Operator | Description | | | | | | | | | | | | | | | | | |
| + | addition | | | | | | | | | | | | | | | | | |
| - | subtraction | | | | | | | | | | | | | | | | | |
| * | multiplication | | | | | | | | | | | | | | | | | |
| / | division | | | | | | | | | | | | | | | | | |
| ^ or ** | exponentiation | | | | | | | | | | | | | | | | | |
| x %% y | modulus (x mod y) 5%%2 is 1 | | | | | | | | | | | | | | | | | |
| x %/% y | integer division 5%/%2 is 2 | | | | | | | | | | | | | | | | | |

| Logical Operators | | | | | | | | | | | | | | | | | | | | | | |
|--|--------------------------|--|----------|-------------|---|-----------|----|-----------------------|---|--------------|----|--------------------------|----|------------------|----|--------------|----|-------|-------|--------|-------|---------|
| <table border="1"> <thead> <tr> <th>Operator</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><</td> <td>less than</td> </tr> <tr> <td><=</td> <td>less than or equal to</td> </tr> <tr> <td>></td> <td>greater than</td> </tr> <tr> <td>>=</td> <td>greater than or equal to</td> </tr> <tr> <td>==</td> <td>exactly equal to</td> </tr> <tr> <td>!=</td> <td>not equal to</td> </tr> <tr> <td>!x</td> <td>Not x</td> </tr> <tr> <td>x y</td> <td>x OR y</td> </tr> <tr> <td>x & y</td> <td>x AND y</td> </tr> </tbody> </table> | | | Operator | Description | < | less than | <= | less than or equal to | > | greater than | >= | greater than or equal to | == | exactly equal to | != | not equal to | !x | Not x | x y | x OR y | x & y | x AND y |
| Operator | Description | | | | | | | | | | | | | | | | | | | | | |
| < | less than | | | | | | | | | | | | | | | | | | | | | |
| <= | less than or equal to | | | | | | | | | | | | | | | | | | | | | |
| > | greater than | | | | | | | | | | | | | | | | | | | | | |
| >= | greater than or equal to | | | | | | | | | | | | | | | | | | | | | |
| == | exactly equal to | | | | | | | | | | | | | | | | | | | | | |
| != | not equal to | | | | | | | | | | | | | | | | | | | | | |
| !x | Not x | | | | | | | | | | | | | | | | | | | | | |
| x y | x OR y | | | | | | | | | | | | | | | | | | | | | |
| x & y | x AND y | | | | | | | | | | | | | | | | | | | | | |

Development

Developers of new features should know a few things on the structure of the code. When developing custom functions please stick to these conventions.

The following files are required; they must all be in the same directory (these are contained within the Rcrust folder which should simply be copied to the desired location):

- 1) From **Perple_X** suite (in the folder called "data"):
 - a. The various datafiles you wish to use, these include: thermodynamic datafiles and solution model files, typically *hp04ver.dat* and *solution_model.dat* as well as the Perple_X option file, *perplex_option.dat*.
 - b. Meemum.exe
 - c. The rest of Perple_X (vertex, build, werami, etc.) are not required.

- 2) From **Rcrust** (in the folder called "code"):
 - a. *ui.r* and *server.r*, these build the Rcrust Graphical User Interface (GUI)
 - b. *main.r*, this houses the main calls to run Rcrust
 - c. *meemum_connect.r*, the functions for calling and interpreting outputs from meemum
 - d. Various *init_xxx* files, used to transform user input in data structures that Rcrust can understand.
 - i. *init_bulk.r* sets the bulk composition(s) of the system
 - ii. *init_pt.r* sets the P - T conditions
 - iii. *init_ph_add.r* sets the phases to add
 - iv. *init_ph_extr.r* sets the phases to extract
 - v. *init_meem.r* writes user inputs into a meemum build file
 - vi. *init_dependence* determines the calculation order of points
 - vii. *init_wrapper* the phase stability calculator from Perple_X
 - e. *parse_meem*, a temporary Perple_X build file created to pass data into the wrapper
 - f. *run.Rcrust.r*, the Rcrust calculation loop to be called for each point
Technically, each function works `c0` which tracks the bulk compositional changes invoked by phase manipulations. *main.r* loops through each point, calculating and modifying the phases according to the chosen definitions and eventually stores the final product in a list called *crust*, whose structure is `crust[[y_i]][[x_i]]`. So, for instance the SiO₂ content in the melt of point $y_i=4$ and $x_i=2$ is

```
crust[[4]][[2]][ "melt (HP) ", "SiO2" ]
```
 - g. Various *xxx.dll* files which contain compiled libraries needed to perform calculations within R

Troubleshooting

A list of known errors that are unavoidable or are still to be fixed.

Bulk_ss system properties

Warning: some bulk system properties are reported as molar properties but perplex considers the bulk system to be one mol thus all molar properties need to be adjusted accordingly

Molar phase proportions

Only weight definitions of bulk and phases is currently possible, read.meemum cannot read molar phase proportions. If molar proportions for bulk are entered, then bulk is molar but individual phases are weights thus phase extractions crash.

Buffered Output

The R console by default returns a buffered output which forces the console to only refresh when flush.console() is called. To disable the buffering and view run data live deselect from R toolbar Misc/Buffered Output.

External Sources

Perple_X

<http://www.perplex.ethz.ch/>

Perple_X Solution Model Glossary

http://www.perplex.ethz.ch/PerpleX_solution_model_glossary.html

THERMOCALC's list of mineral abbreviations

<http://www.metamorph.geo.uni-mainz.de/thermocalc/documentation/abbreviations/index.html>

REFERENCES

Mayne, M. J., Moyer, J. F., Stevens, G., & Kaislaniemi, L. (2016). Rcrust: a tool for calculating path-dependent open system processes and application to melt loss. *Journal of Metamorphic Geology*, **34**(7), 663-682

Connolly, J.A.D., 2005. Computation of phase equilibria by linear programming: a tool for geodynamic modeling and its application to subduction zone decarbonation. *Earth and Planetary Science Letters*, **236**, 524–541.

Connolly, J.A.D., 2009. The geodynamic equation of state: what and how. *Geochemistry, Geophysics, Geosystems*, **10**, 1–19.

Connolly, J.A.D. & Kerrick, D.M., 1987. An algorithm and computer program for calculating composition phase diagrams. *Calphad*, **11**, 1–55.

ADDENDUM C: Presentation of the programming code for Rcrust

Included within the thesis files for evaluation is the programming code of the Rcrust program. This code is heavily commented throughout and is divided into a number of separate files for convenience. Each file is preceded by a brief description of its overall purpose and a summary of the tasks it performs.

File structure

| Purpose | File name | Page |
|--|-------------------|------|
| Initialise P-T conditions | init_pt.r | 5 |
| Initialise bulk compositions | init_bulk.r | 7 |
| Initialise phase additions | init_ph_add.r | 10 |
| Initialise phase extractions | init_ph_extr.r | 11 |
| Initialise dependence structure | init_dependence.r | 11 |
| Initialise meemum | init_meem.r | 14 |
| Rcrust launcher | main.r | 16 |
| Rcrust calculation loop | run.Rcrust.r | 21 |
| Wrapper for phase stability calculations | init_wrapper.r | 32 |
| Meemum connect | meemum_connect.r | 34 |
| Server | server.r | 36 |
| User interface | ui.r | 80 |

Brief description of code

Rcrust operates by receiving user input through a graphical user interface (GUI built using server.r and ui.r) which writes parameters for modelling into a text file (the file is stored in the projects directory within the workingfile in the inputs folder). The text file itself can be altered and additional parameters accessed. This file is then loaded into the current environment and program initialisation begins. A modelling space is set up consisting of a number of points which record information including temperature, pressure and bulk composition of each point (init_pt.r and init_bulk.r). Phase manipulations are optionally set on points to be analysed during the calculation of phase stabilities for each point (init_ph_add.r and init_ph_extr.r) which can alter the chemical composition of points. Input parameters are examined for possible dependence relations and these dependence relations are solved to determine the calculation order of points (init_dependence.r). A dummy “meemum build file” is created to parse modelling settings to “meemum” from Perple_X

(init_meem.r). In sequential order Rcrust runs through each point (main.r): a calculation loop (run.Rcrust.r) is called which adds chemical contributions to the reactive bulk composition (c0) then calls a wrapper function (wrapper.r) that interrogates meemum with the pressure, temperature and bulk composition of the reactive bulk (meemum_connect.r). Meemum then uses Gibbs free energy minimisation to predict the most stable phase assemblage for the system and returns results to Rcrust. Phase extractions are then performed on this assemblage and the new altered bulk composition is optionally recalculated for its new phase assemblage. This final data is stored at its point coordinates within the modelling space (crust) and the next point in the calculation order is calculated. Since points are calculated in sequential order determined by dependence relations, the reactive bulk composition of each subsystem can be passed between points thus inheriting compositional changes. Once all points are calculated focus is returned to the graphical user interface where outputs can be filtered, saved and graphed.

Methodology and syntax utilised in programming code

Tags

Tags to be used within code

| | |
|-----------|---|
| # fix-tag | requires urgent implementation critical to stability of program |
| # mod-tag | non-urgent addition that would improve program |

Variables

| | |
|-----------------------------|--|
| #variable-def | definition of a variable created and the syntax to be used within the variable structure |
| crust[[y_i]][[x_i]] | two dimensional array holding outputs for each point |
| input_pt[[y_i]][[x_i]] | two dimensional array holding pressure and temperature conditions of each point |
| input_bulk[[y_i]][[x_i]] | two dimensional array holding bulk composition and relative mass for each point |
| input_ph_add[[y_i]][[x_i]] | two dimensional array holding phase addition definitions for each point |
| input_ph_extr[[y_i]][[x_i]] | two dimensional array holding phase extraction definitions for each point |

| | |
|----------------------------|--|
| dependencies[[y_i]][[x_i]] | two dimensional array holding references for each dependent point (evaluates functions of the form $rs\{x_{i-1},y_i\}+5-as\{x_{n-2},y_i^2\}$) |
| available components | chemical components available for component transformations, set in thermodynamic data file |
| major_elements/comps | the current chemical components to use |
| c0 | point specific bulk composition and relative mass (to be passed between points) |
| use_sol_models | the current solution models to use |
| calc_choice | selector for calculation method |
| comp_mat | matrix recording the status of calculation of all points |
| project_directory | folder storing all projects |
| working_file | the current project |
| tuples | vector notation for identifying points in x,y space |
| x_i | the current increment in the x direction |
| x_n | the maximum increment in the x direction |
| y_i | the current increment in the y direction |
| y_n | the maximum increment in the y direction |

Functions

#function-def definition of a function and the syntax to be used within the function calls

wrapper(comps,c0,press,temp,calc_choice="read.meemum")

run.Rcrust(comps,c0,press,temp,ph_extr_pnt,cumul_extract_pnt=NULL,ph_add_pnt,cumul_add_pnt=NULL)

run.meemum(meemum.path="",build.file="",meemum.order="",press=press*1000,temp=temp+273.15,bulk="",components="",pt_comp=pt_comp)

get_val(y_i,x_i,phase,variable,crust=crust,oxy_num=24,site_occup="biotite",select=1)

eval_expr(expr,calc_phases=calc_phases,crust=crust)

Evaluate expression given calc_phases and crust, “()” are for solution models, “{}” are for function terms and bodmas. Evaluate any “{” and any word before it up until “}”

Evaluates the following subfunctions

ph{phase;unit;x_i;y_i}

delta{ph;x_#;y_#;unit}

retain{amount;unit;ph}

Calculate mass of retention phases to extract: extract till retention amount of retention phases is left, where ph can be omitted to take on the current ph

return{phase;amount}

where amount can be % or mass

Triggers

| | |
|---------------------|---|
| exit_calc | leave calculation |
| reequilibrate_steps | calculate point again after phase manipulations |
| pass | tag points to be left out |
| silent_calc | limit the messages to console |

System labels

| | |
|----------|----------------------|
| rs | reactive subsystem |
| is | isolated subsystem |
| es | extract subsystem |
| es_cumul | cumulative subsystem |

Advanced functionalities not yet worked into GUI

| | |
|----------------|--|
| scale_bulk_x_i | scale a composition incrementally across x-i |
|----------------|--|

Initialise P-T conditions (init_pt.r)

Creates a two dimensional data structure that will record pressure and temperature conditions of the form:

```
input_pt[[y_i]][[x_i]]$Pressure,$Temperature
```

Reads either pressure and temperature definitions defined in the graphical user interface or imported from a file of the form:

```
pt_definitions<-list("{x_a,y_a}_{x_b,y_b}"=c("pressure_#", "temperature_#"),
"{x_a,y_a}_{x_b,y_b}"=c("pressure_#", "temperature_#"))
```

Populates pressure and temperature conditions for each point or assigns dependence

Performs error validation ensuring that all points have an assigned pressure and temperature conditions

```
#####
#
# Initialize PT conditions
#
#####
cat("Initializing PT conditions...\n")
# fix-tag: error handling
input_pt<-rep(list(rep(list(c(Pressure=NULL, Temperature=NULL)), x_n)), y_n)
#####
#
# PT definition from configuration file
#
#####
if(pt_def=="input") {
cat("Calculating PT conditions from inputs...\n")
pnts<-unlist(strsplit(names(pt_definitions), "_"))
for(h in 1:length(pt_definitions)) {
a<-unlist(strsplit(gsub("\\{", "", gsub("\\}", "", pnts[h*2-1]), split=";"))
b<-unlist(strsplit(gsub("\\{", "", gsub("\\}", "", pnts[h*2]), split=";"))
for(x_i in a[1]:b[1]) {
for(y_i in a[2]:b[2]) {
press<-eval(parse(text=paste0(pt_definitions[[h]][1])))
temp<-eval(parse(text=paste0(pt_definitions[[h]][2])))
input_pt[[y_i]][[x_i]]<-matrix(c(press, temp), nrow=1, dimnames =
list("PT", c("Pressure", "Temperature")))
}
}
}
}
#####
#
# PT definition from input file
#
#####
if(pt_def=="file") {
# mod-tag: cleanup below
cat("Reading PT conditions from file...\n")
PT_file<-paste(work_dir, "/Projects/", workingfile, "/Inputs/PT.txt", sep="")
PT0<-read.table(PT_file, sep="\t")
if(any(is.na(PT0[1, 1])) {
warning("Error at init_pt.r\nError in PT file, NA found in column names,
probably a tab left at the end of the first row, will try to rectify and continue")
# mod-tag: Surely there's a better way to make a matrix numeric?
make_numeric_mat<-function(x) {
y<-matrix(as.numeric(as.vector(as.matrix(x))), nrow(x), )
rownames(y)<-rownames(x)
}
}
```

```

colnames(y)<-colnames(x)
return(y)
}
err<-length(PT0[1,])
mat<-PT0[-1,-1]
colnames(mat)<-as.character(unlist(PT0[1,1:(err-1)]))
rownames(mat)<-1:(nrow(PT0)-1)
PT0<-make_numeric_mat(mat)
}
for(i in 1:nrow(PT0)){
PT[[PT0[i,1]][[PT0[i,2]]]$press<-PT0[i,3]
PT[[PT0[i,1]][[PT0[i,2]]]$temp<-PT0[i,4]
}
}
#####
#
# Error validation
#
#####
for(y_i in 1:y_n){
  for(x_i in 1:x_n){
    if(is.null(input_pt[[y_i]][[x_i]])){cat("Error: No pt defined for x_i =",x_i,"
y_i =",y_i," \n");stop()}
  }
}
cat("Done with PT conditions\n")
cat(".....\n")

```

Initialise bulk compositions (init_bulk.r)

Creates a two dimensional data structure that will record bulk chemical conditions of the form:

```
input_bulk[[y_i]][[x_i]]$major_elements,$mass
```

where major_elements is a list of chemical components as wt.% oxides

and mass is a numeric value against which to scale the reactive subsystem

Reads either bulk compositional definitions defined in the graphical user interface or imported from a file of the form:

```
bulk_definitions<-list("{x_a,y_a}_{x_b,y_b}"=c(major_elements_wt.%, "mass"),
"{x_a,y_a}_{x_b,y_b}"=c(major_elements_wt.%, "mass"))
```

where major_elements_wt.% is a comma separated list of real numbers as weight percentage of each chemical component or terms of the form rs{x_i;y_i} to assign dependence relations.

Possible dependents = rs,fs,es,as non-dependents = c0,x_i,y_i,x_n,y_n

If the only one term is given e.g. rs{x_i;y_i} then this will be copied for all chemical components

Populates bulk composition conditions for each point or assigns dependence

Performs error validation ensuring that all points have an assigned bulk composition condition

```
#####
#
# Initialize bulk composition data (only major elements for now)
#
#####
cat("Initializing bulk composition...\n")
#fix-tag: error handling
# Error handling
if(exists("bulk_def")){
if(bulk_def=="input"){
cat("Bulk composition defined from inputs...\n")
}else{
if(bulk_def=="file"){
cat("Bulk composition defined from file...\n")
}else{
cat("Error ! No valid option specified for major composition...\n")
stop()
}
}
}else{
cat("Error ! bulk_def not specified...\n")
stop()
}
# Create data structure
cat("P-T-X space under investigation with x =",x_n,"and y =",y_n,"\n")
input_bulk<-rep(list(rep(list(NULL),x_n)),y_n)
#####
#
# Bulk definition from input
#
#####
## Majors
```

```

if(bulk_def=="input"){
cat("Creating bulk compositions from definitions in configuration file\n")
#fix-tag should set Mass and volume somewhere but can only normalise once
expressions have been evaluated before parsing to Rcrust()
pnts<-unlist(strsplit(names(bulk_definitions),"_"))
for(h in 1:(length(bulk_definitions))){
a<-unlist(strsplit(gsub("\\{","",gsub("\\}","",pnts[h*2-1])),split=";"))
b<-unlist(strsplit(gsub("\\{","",gsub("\\}","",pnts[h*2])),split=";"))
for(x_i in a[1]:b[1]){
for(y_i in a[2]:b[2]){
if(length(bulk_definitions[[h]])==1){
input_bulk[[y_i]][[x_i]]<-rep(bulk_definitions[[h]],length(major_elements)+1)
}else{
input_bulk[[y_i]][[x_i]]<-c(bulk_definitions[[h]])
}
}

if(!length(c(major_elements,"mass"))==length(input_bulk[[y_i]][[x_i]])){cat(paste0(
"Error number of entries in input_bulk (",length(input_bulk[[y_i]][[x_i]]),") is
not equal to major elements + 1 (",length(c(major_elements,"mass")),") for x_i =
",x_i,"; y_i = ",y_i,"\\n"));stop()}
names(input_bulk[[y_i]][[x_i]])<-c(major_elements,"mass")
#Apply Scaling on x_i - Note if assigning dependence do not leave gaps while
scaling
#mod-tag: work this into GUI
scale_bulk_x_i<-FALSE
if(scale_bulk_x_i){
if(x_i>1){
leftmost<-0
for(i in 2:(x_i-1)){if(is.null(input_bulk[[y_i]][[i]])){leftmost<-i
break
}}
incr<-1
for(i in leftmost:(x_i-1)){
#scale proportionate value to distance
b_prop<-incr/(length(leftmost:(x_i-1))+1)
input_bulk[[y_i]][[i]]<-
as.numeric(input_bulk[[y_i]][[x_i]])*b_prop+as.numeric(input_bulk[[y_i]][[leftmost-
1]])*(1-b_prop)
names(input_bulk[[y_i]][[i]])<-c(major_elements,"mass")
class(input_bulk[[y_i]][[i]])<-"character"
incr<-incr+1
}
}
}
}
}
}
}
#####
#
# Bulk definition from file
#
#####
## Majors
if(bulk_def=="file"){
cat("Bulk composition for majors defined from bulk file:",bulk_file,"\\n")
bulk_file_loc<-paste0(projects_directory,"/",working_file,"/Inputs/",bulk_file)
if(file.exists(bulk_file_loc)){
table_in<-as.matrix(read.table(bulk_file_loc,sep="\\t"))
table_out<-table_in[-1,,drop=FALSE]
colnames(table_out)<-table_in[1,]
major_elements<-setdiff(table_in[1,],c("from","to","mass"))
for(h in 1:nrow(table_out)){
a<-unlist(strsplit(table_out[h,1],split=";"))
b<-unlist(strsplit(table_out[h,2],split=";"))
for(x_i in a[1]:b[1]){
for(y_i in a[2]:b[2]){
input_bulk[[y_i]][[x_i]]<-table_out[h,c(-1,-2)]
}
}
}
}
}
}
}

```

```

        names(input_bulk[[y_i]][[x_i]])<-c(major_elements,"mass")
    }
}
}
}else{
    cat("Error during bulk initialisation\nBulk file: ",bulk_file_loc," not found")
    stop()
}
}
#####
#
# Error validation
#
#####
for(y_i in 1:y_n){
    for(x_i in 1:x_n){
        if(is.null(input_bulk[[y_i]][[x_i]])){cat("Error: No bulk defined for x_i
=",x_i," y_i =",y_i," \n");stop()}
    }
}
cat("Done with bulk composition preparation\n")
cat(".....\n")

```

Initialise phase additions (init_ph_add.r)

Creates a two dimensional data structure that will record phase addition conditions of the form:

```
input_ph_add[[y_i]][[x_i]]
```

Reads either phase addition definitions defined in the graphical user interface or imported from a file of the form:

```
ph_add_definitions<-  
list("{x_a,y_a}_{x_b,y_b}"=c(major_element_proportions,"mass"),  
"{x_a,y_a}_{x_b,y_b}"=c(major_elements,"mass"))
```

where major_element_proportions is a comma separated list of real numbers for each phase

```
#####  
#  
# Initialize phase addition options  
#  
#####  
#handles definitions of the form: ph_add_definitions<-  
list("{1,1}_{10,1}"=c(2.95,0.618,15.3,7.11,5.17,0.15,0.183,6.44,63.5,5), "{1,1}_{10,  
1}"=c(2.95,0.618,15.3,7.11,5.17,0.15,0.183,6.44,63.5,2))  
if(ph_add){  
  cat("Setting phase addition options...\n")  
  # fix-tag: error handling  
  # Create data structure  
  input_ph_add<-rep(list(rep(list(NULL),x_n)),y_n)  
  #####  
  #  
  # Phase addition from input definitions  
  #  
  #####  
  cat("Setting phase additions from definitions in configuration file\n")  
  pnts<-unlist(strsplit(names(ph_add_definitions),"_"))  
  for(h in 1:length(ph_add_definitions)){  
    a<-unlist(strsplit(gsub("\\{"," ",gsub("\\}", " ",pnts[h*2-1])),split=";"))  
    b<-unlist(strsplit(gsub("\\{"," ",gsub("\\}", " ",pnts[h*2])),split=";"))  
    if(length(ph_add_definitions[[h]]==length(c(major_elements,"mass"))){  
      names(ph_add_definitions[[h]])<-c(major_elements,"mass")  
      for(x_i in a[1]:b[1]){  
        for(y_i in a[2]:b[2]){  
          if(is.null(input_ph_add[[y_i]][[x_i]])){  
            input_ph_add[[y_i]][[x_i]]<-list(ph_add_definitions[[h]])  
          }else{  
            input_ph_add[[y_i]][[x_i]][length(input_ph_add[[y_i]][[x_i]])+1]<-  
ph_add_definitions[[h]]  
          }  
        }  
      }  
    }  
  }  
}else{  
  cat("No phase addition.\n")  
}  
cat("Done with phase addition options\n")  
cat(".....\n")
```

Initialise phase extractions (init_ph_extr.r)

Creates a two dimensional data structure that will record phase extraction conditions of the form:

```
input_ph_extr[[y_i]][[x_i]]
```

Reads either phase addition definitions defined in the graphical user interface or imported from a file of the form:

```
ph_extr_definitions<-  
list("{x_a,y_a}_{x_b,y_b}"=c(condition="expression",Ph_#1,Ph_#2,...,Ph_#x),  
"{x_a,y_a}_{x_b,y_b}"=c(condition="expression",Ph_#1,Ph_#2,...,Ph_#x))
```

where ph_# denotes a number of different phases and their proportion of phase proportion statement by which they should be extracted.

```
#####  
#  
# Initialize phase extraction options  
#  
#####  
#handles definitions of the form: ph_extr_definitions<-  
list("{1,1}_{6,1}"=c(condition="melt(HP)>=8wt.%",P1="20%",ky="3", "melt(HP)"="rt(4,w  
t.%)"), "{4,1}_{9,1}"=c(condition="melt(HP)>=8wt.%",P1="20%",ky="3", "melt(HP)"="rt(4  
,wt.%)"))  
if(ph_extr){  
cat("Setting phase extraction options...\n")  
# fix-tag: error handling  
# Create data structure  
input_ph_extr<-rep(list(rep(list(NULL),x_n)),y_n)  
#####  
#  
# Phase extraction from input definitions  
#  
#####  
cat("Setting phase extractions from definitions in configuration file\n")  
pnts<-unlist(strsplit(names(ph_extr_definitions),"_"))  
for(h in 1:length(ph_extr_definitions)){  
a<-unlist(strsplit(gsub("\\{"," ",gsub("\\}", " ",pnts[h*2-1])),split=";"))  
b<-unlist(strsplit(gsub("\\{"," ",gsub("\\}", " ",pnts[h*2])),split=";"))  
for(x_i in a[1]:b[1]){  
for(y_i in a[2]:b[2]){  
if(is.null(input_ph_extr[[y_i]][[x_i]])){  
input_ph_extr[[y_i]][[x_i]]<-list(ph_extr_definitions[[h]])  
}else{  
input_ph_extr[[y_i]][[x_i]][length(input_ph_extr[[y_i]][[x_i]])+1]<-  
ph_extr_definitions[[h]]  
}  
}  
}  
}  
}else{  
cat("No phase extraction.\n")  
}  
cat("Done with phase extraction options\n")  
cat(".....\n")
```

Initialise dependence structure (init_dependence.r)

Creates a two dimensional data structure that will record dependence relations of the form: dependencies [[y_i]][[x_i]]

Searches through points and for each chemical component in “major_elements” evaluates the expression for dependence

Merges common dependence and determines tiers for hierarchy of dependence

Determines calculation order from tiers

Performs error validation to ensure all dependences have been resolved to a determinable element

```
#####  
#  
# Determine the dependence relations between points  
#  
#####  
# Bulk Composition *For now dependence is only on composition  
#possible handled variables dependents = rs,fs,es,as non-dependents =  
c0,x_i,y_i,x_n,y_n  
#handles functions of the form"rs{x_i-1,y_i}+5-as{x_n-2,y_i*2}"  
#fixtag error handling  
dependencies<-rep(list(rep(list(),x_n)),y_n)  
#for each point  
for(y_i in 1:y_n){  
  for(x_i in 1:x_n){  
    dependence<-NULL  
    # for each major element  
    for(maj_i in 1:length(major_elements)){  
      #pull out tuples that major element for that point is dependent on  
      #fix-tag: look at main.r for bulk comp to allow multiple rs calls in one tuple  
      n<-  
      gsub("rs","",gsub("es","",gsub("as","",gsub("fs","",input_bulk[[y_i]][[x_i]][maj_i]  
      )))  
      left_brac<-gregexpr("\\{", n)[[1]]  
      right_brac<-gregexpr("\\}", n)[[1]]  
      tuples<-NULL  
      for(tup_i in 1:length(left_brac)){  
        tuples<-c(tuples,substr(n, left_brac[tup_i], right_brac[tup_i]))  
      }  
      if(tuples==""){dependence="base"}else{  
        split_tuples<-strsplit(gsub("\\{","",gsub("\\}", "", tuples)),split=";")  
        for(split_i in 1:length(split_tuples)){  
          dependence<-  
          c(dependence,paste(eval(parse(text=split_tuples[[split_i]][1])),eval(parse(text=split  
          it_tuples[[split_i]][2])),sep=";")  
        }  
      }  
    }  
  }  
  #Keep common dependence  
  dependencies[[y_i]][[x_i]]<-union(dependence,dependence)  
}  
#####  
# Determine tiers  
#####  
tiers<-rep(list(rep(list(NULL),x_n)),y_n)  
eval_dep<-dependencies  
for(y_i in 1:y_n){  
  for(x_i in 1:x_n){
```

```

tier_counter<-0
while(is.null(tiers[[y_i]][[x_i]])){
if(eval_dep[[y_i]][[x_i]]=="base"){
  tiers[[y_i]][[x_i]]<-1+tier_counter
}else{
tier_counter<-tier_counter+1
#Evaluate dependencies
new_dep<-NULL
for(dep_i in 1:length(eval_dep[[y_i]][[x_i]])){
  dep_pos<-strsplit(eval_dep[[y_i]][[x_i]][dep_i],split=";")[[1]]
  new_dep<-
c(new_dep,dependencies[[as.numeric(dep_pos[2])]][[as.numeric(dep_pos[1])]])
}
#keep common dependence
eval_dep[[y_i]][[x_i]]<-union(new_dep,new_dep)
}
}
}
#####
# Determine calculation order
#####
tiers_union<-union(unlist(tiers),unlist(tiers))
calc_order<-rep(list(rep(list(rep(list(),1)),1)),length(tiers_union))
group_num<-0
for(i in tiers_union){
  group_num<-group_num+1
  first_in_group<-TRUE
for(y_i in 1:y_n){
  for(x_i in 1:x_n){
if(tiers[[y_i]][[x_i]]==i){
if(first_in_group){
  calc_num<-0
}
calc_num<-calc_num+1
first_in_group<-FALSE
calc_order[[group_num]][[calc_num]]<-list(x_i=x_i,y_i=y_i)
}
}
}
}
#####
#
# Error validation
#
#####
for(y_i in 1:y_n){
  for(x_i in 1:x_n){
    if(is.null(dependencies[[y_i]][[x_i]])){cat("Error: No dependence defined for
x_i =",x_i," y_i =",y_i," \n");stop()}
  }
}
if(!length(unlist(calc_order))/2==x_n*y_n){cat("Error: Calculation order could not
be determined\n");stop()}
cat("Done with dependence determination\n")
cat(".....\n")

```

```

0 unused place holder, post 06
",n_comp_trans," number component transformations\n      ",
number_components," number of components in the data base\n",
paste(transformations,collapse="\n"),"\n",
molar_vs_wt," component amounts, 0 - molar, 1 weight
0 unused place holder, post 06
0 unused place holder, post 06
0 unused place holder, post 05
5 ifug EoS for saturated phase
2 gridded minimization dimension (1 or 2)
0 special dependencies: 0 - P and T independent, 1 - P(T), 2 - T(P)
0.00000      0.00000      0.00000      0.00000      0.00000      Geothermal
gradient polynomial coeffs.

begin thermodynamic component list
",paste(comp_mat,collapse=""),
"end thermodynamic component list

begin saturated component list
",paste(unlist(strsplit(saturated_components,split=",")),collapse="\n"),"
end saturated component list

begin saturated phase component list
",paste(unlist(strsplit(saturated_phase_components,split=",")),collapse="\n"),"
end saturated phase component list

begin independent potential/fugacity/activity list
",paste(unlist(strsplit(independent_potential_fugacity_activity,split=",")),collaps
e="\n"),"
end independent potential list

begin excluded phase list
",paste(unlist(strsplit(exclude_phases,split=",")),collapse="\n"),"
end excluded phase list

begin solution phase list
",paste(use_sol_models,collapse="\n"),
"\nend solution phase list

0.0000      0.0000      0.00000000 0.0000      0.0000      max p, t, xco2, u1, u2
0.0000      0.0000      0.00000000 0.0000      0.0000      min p, t, xco2, u1, u2
0.0000      0.0000      0.00000000 0.0000      0.0000      unused place holder post
06

2 1 4 5 3 indices of 1st & 2nd independent & sectioning variables
",sep="")
write(dummy,file=paste0(data_directory,"/parse_meem.dat"))
cat("Created meemum build file as
",paste0(data_directory,"/parse_meem.dat"),"\n",sep="")
cat(".....\n")

```

Rcrust launcher (main.r)

Initialises packages and functions to be used on runtime

Applies additional settings accessible to advanced users (including functionalities not yet worked into GUI)

Sets data structures

Evaluates dependent values to populate all fields of required parameters

Sets up calculation loop to sequentially call “run.Rcrust” saving outputs in “crust” and re-evaluate dependences by tier levels

Saves image of R environment on completion of calculation loop

```
#####  
## Rcrust (main.r)  
#####  
# Function for shortcut  
.First<-function(){  
first_load<<-FALSE  
  #Load dependencies  
  library(utils)  
  if(!require(shiny)){  
    install.packages("shiny")  
  }  
  library(shiny,quietly=TRUE)  
  if(!require(raster)){  
    install.packages("raster")  
  }  
  if(!require(rgeos)){  
    install.packages("rgeos")  
  }  
  if(!require(grDevices)){  
    install.packages("grDevices")  
  }  
  #Launch GUI  
  Rcrust<<-function(){  
    #Load dependencies  
    library(utils)  
    if(!require(shiny)){  
      install.packages("shiny")  
    }  
    library(shiny)  
    if(!require(raster)){  
      install.packages("raster")  
    }  
    if(!require(rgeos)){  
      install.packages("rgeos")  
    }  
    if(!require(grDevices)){  
      install.packages("grDevices")  
    }  
    #If working directory is x\Projects\y then set to x\code  
    if(length(grep("Rcrust/Projects/",getwd()))==1){  
      setwd(paste0(strsplit(getwd(),split="Projects")[[1]][1],"code"))  
    }  
    runApp()  
  }  
  #Launch without GUI  
  manual_load<<-  
function(working_file,projects_directory=paste0(substr(getwd(),1,nchar(getwd())-  
3),"Projects")){  
  source(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))  
  source("main.r")  
}  
#If working directory is x\Projects\y then set to x\code  
if(length(grep("Rcrust/Projects/",getwd()))==1){  
  setwd(paste0(strsplit(getwd(),split="Projects")[[1]][1],"code"))  
}
```

```

first_load<<-TRUE
}
runApp()
}
#Launch without GUI
manual_load<<-
function(working_file,projects_directory=paste0(substring(getwd(),1,nchar(getwd())-
3),"Projects")){

source(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
  source("main.r")
}
#####
#
# Additional Settings (Advanced users)
#
#####
#####
#Errors
run_errors<<-NULL
exit_calc<<-FALSE
pause_on_error<<-FALSE
# Calculation settings
request_readline<<-FALSE
stop_errors<<-NULL
#Determine calculation mode #normal,parallel (still to come)
calc_mode<<-"normal"
silent_calc<<-TRUE
cumulate_on_rs<<-TRUE
# PT settings
pt_def<<-"input"
PT_restrictions<<-c("Pressure >_0","Temperature >_0")
#PT_restrictions<-
c("Pressure > 2.5","Pressure <_20","Temperature >_600","Temperature <_1100")
#Reaction buffering
reaction_buffering<<-FALSE
reaction_buffer_steps<<-1
start_in_reaction<<-FALSE
merge_duplicates<<-FALSE
# Sec to clock function
  sec_to_clock<<-function(total_sec){
    hr<-floor(total_sec/3600)
    mn<-floor((total_sec-hr*3600)/60)
    if(mn<10){mn_0<-"0"}else{mn_0<-NULL}
    sec<-total_sec-hr*3600-mn*60
    if(sec<10){sec_0<-"0"}else{sec_0<-NULL}
    clock<-paste(hr,":",mn_0,mn,":",sec_0,sec,sep="")
    return(clock)
  }
#####
#
# Set defaults
#
#####
#####
if(class(try(reequilibrate_steps,silent=TRUE))== "try-error"){
reequilibrate_steps<<-TRUE
}
if(class(try(calc_choice,silent=TRUE))== "try-error"){
calc_choice<<-"read.meemum" #options = lars.wrap,read.meemum
}
if(class(try(data_directory,silent=TRUE))== "try-error"){
data_directory<<-gsub("/code","/data",getwd())
}
#####
#
# Initialize datastructures
#

```

```
#####
cat("\n")
source("run.Rcrust.r")      # The function doing the job
chk_valid<-try(
{source("init_bulk.r")      # Bulk composition
source("init_wrapper.r")    # Initialise wrapper function
source("meemum_connect.r")  # Initialise run and read meemum functions
source("init_meem.r")       # Create meemum build file
source("init_pt.r")         # PT conditions
source("init_ph_add.r")     #phase addition
source("init_ph_extr.r")    #phase extraction
source("init_dependence.r") # Determine dependence relations
}
)
#Input validation
input_valid<-TRUE
if(class(chk_valid)=="try-error"){
  cat("Initiation failed\n")
  input_valid<-FALSE
}
if(input_valid){
if(request_readline){
cat("Initiation succesful:\n  Please read the above lines and make sure this is
what you wanted.\n")
rd<-readline(prompt="Choose \"n\" to abort or press [enter] to continue\n")
}else{
cat("Initiation succesful:\n  Computation beginning\n")
rd<-"pass"}
if(!rd=="n"){
  #Normal calc by tiers
  #get values and pass them to run.Rcrust
  #Create calculation structures
  #variable-def: crust[[y_i]][[x_i]]
  crust<-rep(list(rep(list(NULL),x_n)),y_n)
  cumul_extract<-rep(list(rep(list(NULL),x_n)),y_n)
  cumul_add<-rep(list(rep(list(NULL),x_n)),y_n)
  calculation_matrix<-matrix(0,y_n,x_n)
  strt<<-proc.time()
  # Refresh input values for groups (j) and their members (i)
  for(j in 1:length(calc_order)){
    for(i in 1:length(calc_order[[j]])){
      #get and assign variables for each point in this calc group
      x_i<-calc_order[[j]][[i]]$x_i
      y_i<-calc_order[[j]][[i]]$y_i
      #Get c0
      c0<-input_bulk[[y_i]][[x_i]]
      #replace rs,es,as,fs tuples with values
      for(k in 1:length(c0)){
        c0_k<-c0[k]
        left_brac<-gregexpr("\\{", c0_k)[[1]]
        right_brac<-gregexpr("\\}", c0_k)[[1]]
        if(left_brac[1]>0){
          subsystem<-NULL
          for(kk in 1:length(left_brac)){
            subsystem<-c(subsystem,substr(c0[k],(left_brac[kk]-2),left_brac[kk]-1))
          }
          tuples<-NULL
          for(tup_i in 1:length(left_brac)){
            tuples<-c(tuples,substr(c0_k, left_brac[tup_i], right_brac[tup_i]))
          }
          split_tuples<-strsplit(gsub("\\{","",gsub("\\}", "",tuples)),split=";")
          #Grab dependent point if cumulating extracts - caution this only works if
          dependent entirely on one point
          if(k==1&ph_extr&cumulate_on_rs){dep_pnt<-
c(eval(parse(text=split_tuples[[1]][1])),eval(parse(text=split_tuples[[1]][2])))
          crust_rows<-rownames(crust[[dep_pnt[2]][[dep_pnt[1]])]

```

```

    parse_cumul_extract<-
crust[[dep_pnt[2]][[dep_pnt[1]][[crust_rows[grep("_es_cumul",crust_rows)],,drop=FA
LSE]
    if(!length(parse_cumul_extract)==0){cumul_extract[[y_i]][[x_i]]<-
parse_cumul_extract}
    }
    #Grab dependent point if cumulating additions - caution this only works if
dependent entirely on one point
    if(k==1&ph_add&cumulate_on_rs){dep_pnt<-
c(eval(parse(text=split_tuples[[1]][1]),eval(parse(text=split_tuples[[1]][2])))
crust_rows<-rownames(crust[[dep_pnt[2]][[dep_pnt[1]]])
    parse_cumul_addition<-
crust[[dep_pnt[2]][[dep_pnt[1]][[crust_rows[grep("_as_cumul",crust_rows)],,drop=FA
LSE]
    if(!length(parse_cumul_addition)==0){cumul_add[[y_i]][[x_i]]<-
parse_cumul_addition}
    }
    #substitute in value
    c0_k<-gsub("rs","",gsub("es","",gsub("as","",gsub("fs","",c0_k)))
    c0_k<-gsub("\\{","",gsub("\\}","",c0_k))
    for(split_i in 1:length(split_tuples)){
        outval<-
try(crust[[eval(parse(text=split_tuples[[split_i]][2])]][[eval(parse(text=split_tu
ples[[split_i][1])]][paste0("Bulk_",subsystem[split_i],names(c0_k)],silent=TRUE)
        if(class(outval)=="try-error"){
            outval<-0
        }
        if(is.null(outval)){
            c0_k_try<-0}else{
            c0_k_try<-
try(sub(paste0(unlist(split_tuples[split_i]),collapse=";"),outval,c0_k))
        }
        if(class(c0_k_try)!="try-error"){c0_k<-c0_k_try}
    }
    }
    c0[k]<-eval(parse(text=c0_k))
}
# Make numeric
nam<-names(c0)
c0<-as.numeric(c0)
names(c0)<-nam
# Save to input_bulk
input_bulk[[y_i]][[x_i]]<-c0
}
if(calc_mode=="normal"){
#run Rcrust in singular for each point in the calc group
for(i in 1:length(calc_order[[j]])){
    x_i<-calc_order[[j]][[i]]$x_i
    y_i<-calc_order[[j]][[i]]$y_i
    if(silent_calc){
        #Calculate times
        pull_time<-proc.time()-strt
        total_sec<-round(pull_time[3])
        run_time<-sec_to_clock(total_sec)
        cat("Computing Point","x_i=",x_i," ; y_i=",y_i,"... Simulation
",round(sum(calculation_matrix!=0)/(nrow(calculation_matrix)*ncol(calculation_matri
x))*100,2),"% complete\n")
        cat("Total run time:",run_time,"\n")
        flush.console()
    }
    #Check if point should be calculated
    #variable-def: 0=remaining,1=calculated,2=aborted
    PT_split<-strsplit(PT_restrictions,split="_")
    for(i in 1:length(PT_restrictions)){
        chk<-
paste(input_pt[[y_i]][[x_i]],PT_split[[i]][1],PT_split[[i]][2],PT_split[[i]][3],c
ollapse="")
        if(!eval(parse(text=chk))){calculation_matrix[y_i,x_i]<-2

```


Rcrust calculation loop (run.Rcrust.r)

Loads functions for:

merging of phase lines (summing extensive properties and averaging intensive properties)

evaluating a Boolean condition that handles missing values as false

evaluating expressions parsed with “{}”. This allows custom functions to be built and called from phase expressions

Runs until told to exit

Adds phases from the isolated subsystem into the reactive subsystem if applicable

Calculates stable phases of the system given the pressure, temperature and bulk composition of the system by calling a wrapper function that communicates with meemum from Perple_X

Renames duplicate phases and splits feldspar phases into Pl or Kf based on CaO/K2O ratio

Extracts phases from reactive subsystem into extract subsystem where applicable

Creates extract subsystem and cumulative extract subsystem

Compiles full system to be saved as a point in “crust”

Flushes response to console

```
#####  
#  
# Main Rcrust Loop (run.Rcrust.r)  
#  
#####  
#function-def:run.Rcrust<-  
function(comps,c0,press,temp,ph_extr_pnt,cumul_extract_pnt=NULL,ph_add_pnt,cumul_ad  
d_pnt=NULL)  
run.Rcrust<-  
function(comps,c0,press,temp,ph_extr_pnt,cumul_extract_pnt=NULL,ph_add_pnt,cumul_ad  
d_pnt=NULL, ... ) {  
#####  
#  
# Ancillary function to merge several lines (by mass)  
#  
#####  
.wtd.add<-function(thelines,prop="mass",avname="Averaged") {  
  if(nrow(thelines)==1) {  
    foo<-thelines  
  }else{  
    # Two sets of cols  
    if(exists_and_true(calc_mol)){  
      extensive.cn<-c("wt%","vol%","mol%","mol") # Extensive properties (mass  
dependant) -- add the others if required  
    }else{  
      extensive.cn<-c("wt%")  
    }  
  }  
  
  intensive.cn<-setdiff(colnames(thelines),c(prop,extensive.cn))
```

```

foo<-matrix(rep(0,length(colnames(thelines))),nrow=1)
colnames(foo)<-colnames(thelines)
# Intensive
foo[,intensive.cn]<-
thelines[,prop]*%thelines[,intensive.cn,drop=F]/sum(thelines[,prop])
# Extensive
foo[,prop]<-sum(thelines[,prop])
if(!length(intersect(extensive.cn,thelines))==0){
  foo[,extensive.cn]<-colSums(thelines[,extensive.cn])
}
rownames(foo)<-avname
return(foo)
}
#exists_and_true
exists_and_true<-function(x){
chk<-try(x,silent=TRUE)
if(class(chk)=="try-error"){return(FALSE)}else{
return(x)
}
}
#####
#
# Expression evaluator
#
#####
#function-def: eval_expr<-function(expr,calc_phases=calc_phases,crust=crust)
#Evaluate expression a given calc_phases and crust
#() are for solution models, {} are for function terms and bodmas
#evaluate any { and word before it up until }
eval_expr<-function(expr,calc_phases=calc_phases,crust=crust){
#wrap outside for evaluation
a<-paste0("{",expr,"}")
while(!unlist(gregexpr("[{}]",a))[1]==-1){
letters<-gregexpr("[a-z]",a)
left_bracs<-unlist(gregexpr("[{]",a))
names(left_bracs)<-rep("left",length(left_bracs))
right_bracs<-unlist(gregexpr("[}]",a))
names(right_bracs)<-rep("right",length(right_bracs))
bracs<-sort(c(left_bracs,right_bracs))
#Find inner brackets
i<-1
while(!(names(bracs)[i]=="left"&names(bracs)[i+1]=="right")){
i<-i+1
}
#evaluate inner brackets
#If require function call find name
j<-1
while((bracs[i]-j)%in%letters[[1]]){
j<-j+1
}
#if j==1 dont require function call, else j+1 is first letter of function
name
if(!j==1){
funct_name<-substr(a,bracs[i]-j+1,bracs[i]-1)
#apply function
if(funct_name=="ph"){
#arguments of the form ph{phase;unit;x_i;y_i} where unit can be any
column name in calc_phases and x_i and y_i are the current point by default
ph_args<-strsplit(substr(a,bracs[i]+1,bracs[i+1]-1),";,")[[1]]
if(length(ph_args)==2){
#current_variable
chk_var<-
try(eval(parse(text="calc_phases[ph_args[1],ph_args[2]]")),silent=TRUE)
if(class(chk_var)=="try-error"){
out<-0
}else{
out<-chk_var
}
}
}
}
}

```

```

    }else{
      #past_variable
      #default look in rs
      if(length(grep("_rs", ph_args[1]))==0&length(grep("_es",
ph_args[1]))==0){
        ph_args[1]<-paste0(ph_args[1],"_rs")
      }
      chk_var<-
try(eval(parse(text="crust[[eval(parse(text=ph_args[[4]])]][[eval(parse(text=ph_ar
gs[[3]])]][ph_args[1],ph_args[2]]")]),silent=TRUE)
      if(class(chk_var=="try-error"){
        out<-0
      }else{
        out<-chk_var
      }
    }
  }
  if(funcnt_name=="delta"){
    skip_delta<-FALSE
    #arguments of the form delta{ph;x_#;y_#;unit} where unit can be wt% or
mass
    delta_args<-strsplit(substr(a,bracs[i]+1,bracs[i+1]-1),"[;]")[[1]]
    delta_phs<-delta_args[1]
    if(substr(delta_args[2],1,8)=="prev_ext"){
      find_ph<-substring(delta_args[2],10)
      pnt<-"find"
      l<-1
      while(pnt=="find"){
        #find earliest of phase extraction or phase absent point, if neither show
warning and take pnt=1
        #find previous increase in _es_cumul
        chk_pnt<-try(crust[[y_i]][[x_i-
1]][paste0(find_ph,"_es"),"mass"],silent=TRUE)
        if(class(chk_pnt=="try-error"){
          chk_pnt<-try(crust[[y_i]][[x_i-
1]][paste0(find_ph,"_rs"),"mass"],silent=TRUE)
        }
        if(class(chk_pnt=="try-error"){
          pnt<-x_i-1
          cat(paste0("Delta calculated to previous phase absent point at x_i =
",pnt,"\n"))
        }
        }else{
          pnt<-x_i-1
          cat(paste0("Delta calculated to previous extract at x_i = ",pnt,"\n"))
        }
        }
        if((x_i-1)==1&pnt=="find"){
          pnt<-x_i-1
          cat(paste0("Delta calculated to first point at x_i = ",pnt,"\n"))
        }
        }
        l<-l+1
      }
      delta_index_x<-pnt
    }else{
      delta_index_x<-eval(parse(text=delta_args[2]))
      if(!(delta_index_x<=x_n&delta_index_x>=1)){cat("Warning delta_index_x not
<x_n and >=1, skipping this delta calculation\n");skip_delta<-TRUE}
      delta_index_y<-eval(parse(text=delta_args[3]))
      if(!(delta_index_y<=y_n&delta_index_y>=1)){cat("Warning delta_index_y not
<y_n and >=1, skipping this delta calculation\n");skip_delta<-TRUE}
      delta_unit<-delta_args[4]
      if(!(delta_unit=="mass"|delta_unit=="wt%")){
        cat("Error delta only currently accepted as mass or wt%\n")
        stop()}
      if(!skip_delta){
        current_mode<-0
        previous_mode<-0
        #Check for plus sign (e.g. aluminosilicate given as sill+ky+and)
        delta_phases<-strsplit(delta_args[1],"+",fixed=TRUE)[[1]]

```

```

    for(each_ph in delta_phases){
      #current_mode
      chk_var<-
try(eval(parse(text="calc_phases[each_ph,delta_unit]")), silent=TRUE)
      if(!class(chk_var)=="try-error"){
        current_mode<-current_mode+chk_var
      }
      #previous_mode
      delta_phase_rs<-paste(each_ph, "_rs", sep="")
      chk_var<-
try(eval(parse(text="crust[[delta_index_y]][[delta_index_x]][delta_phase_rs,delta_u
nit]")), silent=TRUE)
      if(!class(chk_var)=="try-error"){
        previous_mode<-previous_mode+chk_var
      }
    }
    #delta
    if(current_mode>previous_mode){
      out<-current_mode-previous_mode}else{
      out<-0
    }
  }else{out<-0}
}
if(funcnt_name=="retain"){
#c# If Retention Mode - Calculate mass of retention phases to extract
#c# Extract till retention amount of retention phases is left
#c# arguments of the form retain{amount;unit;ph} where ph can be omitted
to take on the current ph
retain_args<-strsplit(substr(a,bracs[i]+1,bracs[i+1]-1),";,")[[1]]
ret<-as.numeric(retain_args[1])
retention_unit<-retain_args[2]
if(!is.na(retain_args[3])){ph<-retain_args[3]}
bulk_no<-which(rownames(calc_phases)=="Bulk_rs")
ph_no<-which(rownames(calc_phases)==ph)
if(retention_unit=="mass"){
  system_less_ret<-calc_phases[-c(bulk_no,ph_no),"mass"]
  if(calc_phases[ph,"mass"]<=ret){
    out<-0
  }else{
    out<-calc_phases[ph,"mass"]-ret
  }
}
if(retention_unit=="vol%"){
  system_less_ret<-calc_phases[-c(bulk_no,ph_no),"vol%"]
  if(calc_phases[ph,"vol%"]<=ret){
    out<-0
  }else{
    new_ret_vol<-(ret*(sum(system_less_ret)))/(100-ret)
    new_ret_mass<-new_ret_vol*calc_phases[ph,"mass"]/calc_phases[ph,"vol%"]
    out<-calc_phases[ph,"mass"]-new_ret_mass
  }
}
if(retention_unit=="wt%"){
  system_less_ret<-calc_phases[-c(bulk_no,ph_no),"wt%"]
  if(calc_phases[ph,"wt%"]<=ret){
    out<-0
  }else{
    new_ret_wt<-(ret*(sum(system_less_ret)))/(100-ret)
    new_ret_mass<-new_ret_wt*calc_phases[ph,"mass"]/calc_phases[ph,"wt%"]
    out<-calc_phases[ph,"mass"]-new_ret_mass
  }
}
}
if(funcnt_name=="return"){
#c# arguments of the form return{phase;amount} where amount can be % or
mass
return_args<-strsplit(substr(a,bracs[i]+1,bracs[i+1]-1),";,")[[1]]
#check if phase is in extract cumul subsystem

```

```

        chk_var<-
try(eval(parse(text=paste0("cumul_extract_pnt[\"",return_args[1],\"_es_cumul\",\"\mas
s\"]\"))),silent=TRUE)
    if(class(chk_var)!="try-error"){
    if(!is.null(chk_var)){
percentage<-FALSE
        #if percentage tag to calculate
    if(length(grep("%",return_args[2]))!=0){
percentage<-TRUE
return_args[2]<-gsub("%","",return_args[2])
    }
    #evaluate for number
    chk_num<-try(eval(parse(text=return_args[2])),silent=TRUE)
    if(class(chk_num)=="try-error"){
cat("Error phase addition could not evaluate isolated function
correctly")
    stop()
    }
    #calculate
    if(percentage){
take<-chk_num/100*chk_var
leave<-(100-chk_num)/100*chk_var
    }else{
take<-chk_num
leave<-as.numeric(chk_var)-chk_num
    }
    if(leave<0){
take<-chk_var
leave<-0}
    #transfer
    cumul_extract_pnt[paste0(return_args[1],\"_es_cumul\"),\"mass\"]<<-leave
    #leave calc
    a<-
paste(c(cumul_extract_pnt[paste0(return_args[1],\"_es_cumul\"),comps],take),collapse=
\", \")
        break
    }else{
cat(paste0("Phase ",return_args[1],\" not found in extract cumul\n\"))
a<-paste(rep(0,length(comps)+1),collapse=\", \")
break
out<-\"\"
    }
    }else{
cat(paste0("Phase ",return_args[1],\" not found in extract cumul\n\"))
a<-paste(rep(0,length(comps)+1),collapse=\", \")
break
out<-\"\"
    }
    }
    #replace inner brackets with function output
a<-paste(substr(a,1,bracs[i]-
j),out,substr(a,bracs[i+1]+1,nchar(a)),sep=\", \")
    }else{
    #evaluate inner brackets for bodmas output
out<-eval(parse(text=substr(a,bracs[i]+1,bracs[i+1]-1)))
a<-paste(substr(a,1,bracs[i]-
j),out,substr(a,bracs[i+1]+1,nchar(a)),sep=\", \")
    }
    }
    return(a)
    }
calc<-TRUE
pass<-1
extract<-NULL
addition_subsystem<-NULL
while(calc){
    # Phase addition
    if(ph_add){

```

```

if(pass==1){
if(!is.null(ph_add_pnt)){
# mod-tag:DFM addition (Disequilibrium Fractional Melting calculation)
if(ph_add_pnt[[1]][[1]]=="dfm"){
#Disequilibrium Fractional Melting calculation
current_pnt<-run.Rcrust(comps=comps,c0=c0,
press=input_pt[[y_i]][[x_i]][1],
temp=input_pt[[y_i]][[x_i]][2],
ph_extr_pnt=NULL,
cumul_extract_pnt=NULL,
ph_add_pnt=NULL
)
next_pnt<-run.Rcrust(comps=comps,c0=c0,
press=input_pt[[y_i]][[x_i+1]][1],
temp=input_pt[[y_i]][[x_i+1]][2],
ph_extr_pnt=NULL,
cumul_extract_pnt=NULL,
ph_add_pnt=NULL
)
#Pl dfm calculation... X(Pl consum) = (Na(melt_2) X(melt_2) + Na(Kfs_2)
X(Kfs_2) - Na(Ms_1) X(Ms_1))/Na(Pl_2)
Na_melt_2<-try(next_pnt[paste0(melt.name,"_rs"),"NA2O"],silent=TRUE)
if(class(Na_melt_2=="try-error"){Na_melt_2<-0}
X_melt_2<-try(next_pnt[paste0(melt.name,"_rs"),"wt%"],silent=TRUE)
if(class(X_melt_2=="try-error"){X_melt_2<-0}
Na_Kf_2<-try(next_pnt["Kf_rs","NA2O"],silent=TRUE)
if(class(Na_Kf_2=="try-error"){Na_Kf_2<-0}
X_Kf_2<-try(next_pnt["Kf_rs","wt%"],silent=TRUE)
if(class(X_Kf_2=="try-error"){X_Kf_2<-0}
Na_Ms_1<-try(current_pnt["Mica(CH A)_rs","NA2O"],silent=TRUE)
if(class(Na_Ms_1=="try-error"){Na_Ms_1<-0}
X_Ms_1<-try(current_pnt["Mica(CH A)_rs","wt%"],silent=TRUE)
if(class(X_Ms_1=="try-error"){X_Ms_1<-0}
Na_Pl_2<-try(next_pnt["Pl_rs","NA2O"],silent=TRUE)
if(class(Na_Pl_2=="try-error"){cat("Error, Plagioclase not found in
second point")}else{
X_Pl_consm<-(Na_melt_2*X_melt_2+Na_Kf_2*X_Kf_2-Na_Ms_1*X_Ms_1)/Na_Pl_2
X_Pl_1<-try(current_pnt["Pl_rs","mass"],silent=TRUE)
if(class(X_Pl_1=="try-error"){cat("Error, Plagioclase not found in
first point")}else{
if(X_Pl_consm<X_Pl_1){
#Place unreacted Pl in addition subsystem
addition_subsystem<-current_pnt["Pl_rs",,drop=FALSE]
addition_subsystem["Pl_rs","mass"]<-
addition_subsystem["Pl_rs","mass"]-X_Pl_consm
rownames(addition_subsystem)<-"Pl_as"
#recalculate new c0 with only the reactive Pl amount
current_pnt["Pl_rs","mass"]<-X_Pl_consm
current_pnt["Bulk_rs","mass"]<-0
c0<- .wtd.add(current_pnt)[,names(c0)]
}
}
}
}
}
else{
#Standard addition
grab<-c(major_elements,"mass")
grab_phases<-NULL
for(add_def_no in 1:length(ph_add_pnt)){
#Evaluate if addition is required
statement<-ph_add_pnt[[add_def_no]][["condition"]]
#if statement is true
if(is.na(as.logical(eval_expr(statement,calc_phases,crust)))){cat(paste0("Error
condition for phase addition definition number ",add_def_no," as ",statement," does
not evaluate to logical output\n"));stop()}
if(as.logical(eval_expr(statement,calc_phases,crust))){
for(add_ph in 2:length(ph_add_pnt[[add_def_no]])){
#Evaluate expression unless numeric comma seperated vector

```



```

if(pass==1) {
  grab<-c(major_elements,"mass")
  grab_phases<-NULL
  for(ex_def_no in 1:length(ph_extr_pnt)){
    #Evaluate if extraction is required
    statement<-ph_extr_pnt[[ex_def_no]][["condition"]]
    #if statement is true
    if(is.na(as.logical(eval_expr(statement,calc_phases,crust)))){cat(paste0("Error
condition for phase extraction definition number ",ex_def_no," as ",statement,"
does not evaluate to logical output\n"));stop()}
    if(as.logical(eval_expr(statement,calc_phases,crust))){
      #must evaluate conditions in order then phases in order, retain phases must be
last, cannot have overlapping extract conditions that target the same phase
      extr_phases<-
names(ph_extr_pnt[[ex_def_no]][2:length(ph_extr_pnt[[ex_def_no]])]
      if(any(extr_phases=="any_phase")){
        #old_extr_phases<-extr_phases
        while(length(which(extr_phases=="any_phase"))!=0){
          rep_no<-which(extr_phases=="any_phase")[1]
          #Remove bulk
          cur_phases<-rownames(calc_phases)[-nrow(calc_phases)]
          #Remove phases with existing definitions
          existing_defs<-NULL
          for(i in 1:length(ph_extr_definitions)){
            existing_defs<-union(existing_defs,names(ph_extr_definitions[[i]]))
          }
          cur_phases<-setdiff(cur_phases,existing_defs)
          x<-as.list(extr_phases)
          x[[rep_no]]<-cur_phases
          extr_phases<-unlist(x)
        }
        extr_vals<-unlist(lapply(1:length(extr_phases),function(i){
          gsub("any_phase",extr_phases[i],ph_extr_pnt[[ex_def_no]][["any_phase"]])
        }))
        names(extr_vals)<-extr_phases
        ex_def<-names(ph_extr_pnt[[ex_def_no]][-1])
        ex_def<-ex_def[-which(ex_def=="any_phase")]
        extr_vals[ex_def]<-ph_extr_pnt[[ex_def_no]][ex_def]
      }else{
        extr_vals<-ph_extr_pnt[[ex_def_no]][extr_phases]
      }
      for(ph in extr_phases){
        #If extract phase is present
        if(length(which(rownames(calc_phases)==ph))>0){
          #grab phase details
          chk<-try(calc_phases[ph,grab,drop=FALSE],silent=TRUE)
          if(!class(chk=="try-error")){
            grab_phases<-rbind(grab_phases,calc_phases[ph,grab,drop=FALSE])
          }
          #Evaluate for extraction value unless ending in % sign
          a<-extr_vals[ph]

if(!substring(extr_vals[ph],nchar(extr_vals[ph]),nchar(extr_vals[ph]))=="%"){
  a<-eval_expr(a,calc_phases,crust)
  }
  #Evaluate unless extraction value is non numeric
  if(!is.na(suppressWarnings(as.numeric(gsub("%","",a))))){
    #If number then take smallest of number or mass present
    if(regexpr("%",a)[1]==-1){
      class(a)<- "numeric"
      if(a<calc_phases[ph,"mass"]){
        grab_phases[ph,"mass"]<-as.numeric(a)
        calc_phases[ph,"mass"]<-calc_phases[ph,"mass"]-
grab_phases[ph,"mass"]
      }else{
        calc_phases[ph,"mass"]<-0
      }
    }else{

```

```

        #If percentage then take as proportion
        grab_phases[ph,"mass"]<-
round(calc_phases[ph,"mass"]*as.numeric(gsub("%", "", a))/100,6)
        calc_phases[ph,"mass"]<-round(calc_phases[ph,"mass"],6)-
grab_phases[ph,"mass"]
    }
    }else{
        cat("Error in expression used for extraction\n")
        stop()
    }
}
}
}else{
    if(!silent_calc){cat("\nNo extraction at this point for this condition\n")}
}
}

#Check for negatives
if(any(calc_phases[, "mass"]<0)|any(grab_phases[, "mass"]<0)){
    cat("Error negative mass produced upon extraction\n")
}
#h# Create Extract
if(!sum(grab_phases[, "mass"]==0){
    #c# Place extracts into extract and calculate Bulk_es
    extr_bulk<-.wtd.add(grab_phases,prop="mass",avname="Bulk_es")
    rownames(grab_phases)<-paste0(rownames(grab_phases), "_es")
    extract<-rbind(grab_phases,extr_bulk)
    #c# Calculate new c0
    c0[c(major_elements, "mass")]<-c(.wtd.add(calc_phases[-
which(rownames(calc_phases)=="Bulk_rs"),c(major_elements, "mass")]))
    #Recalculate mass dependent properties in calc_phases if not re-equilibrating
    if(!reequilibrate_steps){
        calc_phases[, "mol"][-which(rownames(calc_phases)=="Bulk_rs")]<-
calc_phases[, "mass"][-
which(rownames(calc_phases)=="Bulk_rs")]/(calc_phases[, "wt%"][-
which(rownames(calc_phases)=="Bulk_rs")]/calc_phases[, "mol"][-
which(rownames(calc_phases)=="Bulk_rs")])
        calc_phases[, "mol%"][-which(rownames(calc_phases)=="Bulk_rs")]<-
calc_phases[, "mol"][-
which(rownames(calc_phases)=="Bulk_rs")]/sum(calc_phases[, "mol"][-
which(rownames(calc_phases)=="Bulk_rs")]) *100
        calc_phases[, "wt%"][-which(rownames(calc_phases)=="Bulk_rs")]<-
calc_phases[, "mass"][-
which(rownames(calc_phases)=="Bulk_rs")]/sum(calc_phases[, "mass"][-
which(rownames(calc_phases)=="Bulk_rs")]) *100
        vols<-calc_phases[, "mass"][-
which(rownames(calc_phases)=="Bulk_rs")]/calc_phases[, "Density(kg/m3)"][-
which(rownames(calc_phases)=="Bulk_rs")])
        calc_phases[, "vol%"][-which(rownames(calc_phases)=="Bulk_rs")]<-
vols/sum(vols)*100
        calc_phases["Bulk_rs",names(c0)]<-c0
        #mod-tag: check that changed all mass dependent properties
    }
}
}
#If(changed c0 comp) then trigger for recalculation
if(reequilibrate_steps){
    if(!all(calc_phases["Bulk_rs",names(c0)]==c0)){
        calc<-TRUE
    }
}
}
}
#end extraction
}
    pass<-pass+1
}
}
#end calculation

```

```

#if successful calc phases
if(!exit_calc){
if(!is.na(match("Bulk_rs", rownames(calc_phases)))){
  #rename kf
  if(length(intersect(major_elements, c("CAO", "K2O")))==2){
  for(ph in which(rownames(calc_phases)=="Fsp")){
    if(calc_phases[ph, "K2O"]<=0){calc_phases[ph, "K2O"]<-0.0001}
    if(calc_phases[ph, "CAO"]/calc_phases[ph, "K2O"]>1){
      rownames(calc_phases)[ph]<-"P1"
    }else{
      rownames(calc_phases)[ph]<-"Kf"
    }
  }
}
#number duplicates
for(ph in rownames(calc_phases)[which(duplicated(rownames(calc_phases)))]){
  rownames(calc_phases)[which(rownames(calc_phases)==ph)[-1]]<-
paste0(ph, "_", 1:length(which(rownames(calc_phases)==ph)[-1]))
}
#add rs labels
rownames(calc_phases)[-match("Bulk_rs", rownames(calc_phases))]<-
paste0(rownames(calc_phases)[-match("Bulk_rs", rownames(calc_phases))], "_rs")
}
#Update combined extracts
combined_extracts<-NULL
#Expand extract
if(!is.null(extract)){
expanded<-matrix(0, nrow(extract), ncol(calc_phases))
rownames(expanded)<-rownames(extract)
colnames(expanded)<-colnames(calc_phases)
expanded[rownames(extract), colnames(extract)]<-extract
extract<-expanded
}
#Calculate new cumul_extract_pnt
if(is.null(cumul_extract_pnt)){
  new_cumul_extract<-extract
}else{
rownames(cumul_extract_pnt)<-gsub("_cumul", "", rownames(cumul_extract_pnt))
all_extr<-union(rownames(extract), rownames(cumul_extract_pnt))
new_cumul_extract<-matrix(0, length(all_extr), ncol(calc_phases))
rownames(new_cumul_extract)<-all_extr
for(nam_i in all_extr){
  #In extract and cumul - .wtd.add

if(length(which(rownames(extract)==nam_i))>0&length(which(rownames(cumul_extract_pnt)==nam_i))>0){
  new_cumul_extract[nam_i, ]<-
.wtd.add(rbind(cumul_extract_pnt[nam_i, ], extract[nam_i, ]))
}
#In extract- extract

if(length(which(rownames(extract)==nam_i))>0&!length(which(rownames(cumul_extract_pnt)==nam_i))>0){
  new_cumul_extract[nam_i, ]<-extract[nam_i, ]
}
#In cumul- cumul

if(!length(which(rownames(extract)==nam_i))>0&length(which(rownames(cumul_extract_pnt)==nam_i))>0){
  new_cumul_extract[nam_i, ]<-cumul_extract_pnt[nam_i, ]
}
}
#Calculate new Bulk_es_cumul
colnames(new_cumul_extract)<-colnames(cumul_extract_pnt)
new_cumul_extract<-rbind(new_cumul_extract[-
(which(rownames(new_cumul_extract=="Bulk_es")), , drop=FALSE], .wtd.add(new_cumul_ext
ract[-(which(rownames(new_cumul_extract=="Bulk_es")), , drop=FALSE], , "Bulk_es"))
}
}

```

```

if(!is.null(new_cumul_extract)){
rownames(new_cumul_extract)<-paste0(rownames(new_cumul_extract),"_cumul")
}
combined_extracts<-rbind(extract,new_cumul_extract)
#Update IS
#Calculate new cumul_add_pnt
if(is.null(cumul_add_pnt)){
new_cumul_add<-addition_subsystem
}else{
rownames(cumul_add_pnt)<-gsub("_cumul","",rownames(cumul_add_pnt))
all_add<-union(rownames(addition_subsystem),rownames(cumul_add_pnt))
new_cumul_add<-matrix(0,length(all_add),ncol(calc_phases))
rownames(new_cumul_add)<-all_add
for(nam_i in all_add){
#In add and cumul - .wtd.add

if(length(which(rownames(addition_subsystem)==nam_i))>0&length(which(rownames(cumul
_add_pnt)==nam_i))>0){
new_cumul_add[nam_i,]<-
.wtd.add(rbind(cumul_add_pnt[nam_i,],addition_subsystem[nam_i,]))
}
#In addition_subsystem - addition_subsystem

if(length(which(rownames(addition_subsystem)==nam_i))>0&!length(which(rownames(cumu
l_add_pnt)==nam_i))>0){
new_cumul_add[nam_i,]<-addition_subsystem[nam_i,]
}
#In cumul - cumul

if(!length(which(rownames(addition_subsystem)==nam_i))>0&length(which(rownames(cumu
l_add_pnt)==nam_i))>0){
new_cumul_add[nam_i,]<-cumul_add_pnt[nam_i,]
}
}
}
if(!is.null(new_cumul_add)){
rownames(new_cumul_add)<-paste0(rownames(new_cumul_add),"_cumul")
}
combined_add<-rbind(addition_subsystem,new_cumul_add)
#Compile Full System
full_system<-rbind(calc_phases,combined_extracts,combined_add)
}else{
# Return Blank Comp
full_system<-
matrix(0,2,length(c0)+1,dimnames=list(c("Error","Bulk_error"),c("wt%",names(c0))))
}
#Output results to console
if(silent_calc){
#flush.console()
}else{
cat("-----\n","Input:\n",paste0("x_i = ",x_i,"y_i =
",y_i,"P = ",press," kbar," ; T = ",temp," C\n"))
print(c0)
cat("-----\n","Output:\n")
print(full_system[,c(major_elements,"mass")])
flush.console()
}
return(full_system)
}

```

Wrapper for phase stability calculations (init_wrapper.r)

Contains functions for calculating the stable phases of a system given its pressure, temperature and bulk composition

Older versions called a compiled form of meemum from Perple_X to save calculation time however the documented experience of users prefers accessibility to multiple versions of Perple_X so that a wider choice of solution models and chemical systems is available.

Thus the dominant calculation method is now a direct call to meemum from Perple_X

Placing this call within a wrapper assists in handling error response and allows outputs to be formatted to be readable by Rcrust

Replaces placeholder composition in parse_meem.dat with composition of the given point to be calculated (c0)

Calls run.meemum to calculate results and read.meemum to format results

```
#####
#
# Wrapper.r containing calls to meemum from Perple_X
#
#####
#function-def:wrapper<-function(comps,c0,press,temp,calc_choice="read.meemum")
wrapper<-function(comps,c0,press,temp,calc_choice="read.meemum"){
#####
#
# Wrapper function, gives some exception handling
# So an error in the calc does not necessarily kill all the previous results !
#
#####
if(!(calc_choice=="read.meemum"|calc_choice=="lars.wrap")){
cat("\nError: ",calc_choice,"Is not an appropriate calculation choice
(calc_choice)\n")
stop()
}
if(calc_choice=="read.meemum"){
#control comps order
input_file<-
readLines(paste0(gsub("Projects","data",projects_directory),"/parse_meem.dat"))
comp_mat<-NULL
for(ox in major_elements){
comp_mat<-c(comp_mat,paste0(ox,"      1 ",c0[which(major_elements==ox)],"  0.00000
",if(molar_vs_wt==1){"weight amount"}else{"molar amount"},sep=""))
}
start_write<-grep("begin thermodynamic component list",input_file)+1
end_write<-grep("end thermodynamic component list",input_file)-1
input_file[start_write:end_write]<-comp_mat
write(input_file, file =
paste0(gsub("Projects","data",projects_directory),"/parse_meem.dat"))
# mod-tag: have to bring c0 into global, look for alternative
c0<<-c0
calc_out<-read.meemum(run.meemum(comps,c0,press,temp,meemum_path))
}
# mod-tag: Do we still need "lars.wrap"?
if(calc_choice=="lars.wrap"){
#mod-tag: check if need to normalise c0 or if perplex does this
#a (Options: P, T, bulk composition in given order)
a <- .Call("R_phaseq", as.numeric(press*1000), as.numeric(temp+273.15),
as.numeric(c0[comps]))
#'a' is a list, with following components:
```

```

# 1 - Return value (should be zero in case of successful minimization)
# 2 - Amounts of stable phases (wt% or mol, depending on PerpleX options)
# 3 - Names of the stable phases
# 4 - Compositions of stable phases (a[[4]][[i]][j] for phase a[[3]][[i]],
#     element comps[j] (wt% or mol, depending on PerpleX options)* PerpleX
options not editable
# 5 - Bulk properties
# 6 - System properties of stable phases (a[[6]][[i]][j] for phase a[[3]][[i]],
system property [j])
#Check for error
error<<-FALSE
chk_call<-try(length(a[[3]])==length(a[[4]]),silent=TRUE)
if(class(chk_call)=="try-error"){
error<<-TRUE
}
else{
if(!chk_call){
error<<-TRUE
}
}
out_colnames<<-
c("wt%", "vol%", comps, "mass", "V(J/bar)", "H(J)", "Gruneisen_T", "Ks(bar)", "Mu(bar)", "V0
(km/s)", "Vp(km/s)", "Vs(km/s)", "Vp/Vs", "Rho(kg/m3)", "Cp(J/K)", "alpha(1/K)", "beta(1/b
ar)", "S(J/K)", "N(g)", "Cp/Cv")
if(!error){
#compile phases
phases<-NULL
for(ii in 1:length(a[[3]])){
one<-matrix(a[[4]][[ii]],1,length(a[[4]][[ii]]))
prop_one<-matrix(a[[6]][[ii]],1,length(a[[6]][[ii]]))
phases<-rbind(phases,cbind(one,prop_one))
}
colnames(phases)<-
c(comps, "V(J/bar)", "H(J)", "Gruneisen_T", "Ks(bar)", "Mu(bar)", "V0(km/s)", "Vp(km/s)", "
Vs(km/s)", "Vp/Vs", "Rho(kg/m3)", "?", "Cp(J/K)", "alpha(1/K)", "beta(1/bar)", "S(J/K)", "?"
?", "N(g)", 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, "Cp/Cv")
# Create mass and wt columns
wt<-matrix(a[[2]],length(a[[3]]),1)
colnames(wt)<- "wt%"
mass<-matrix(a[[2]]/100*c0["mass"],length(a[[3]]),1)
colnames(mass)<- "mass"
# Create vol% columns
vol<-(wt/phases[, "Rho(kg/m3)", drop=FALSE])/(100/a[[5]][10])*100
colnames(vol)<- "vol%"
#Create Bulk
bulk_row<-matrix(c(100,100,c0[,a[[5]]),1)
rownames(bulk_row)<- "Bulk_rs"
# Compile calc_out
calc_out<-
cbind(wt,vol,phases[,names(c0[1:length(comps)])],mass,phases[, (length(comps)+1):ncol
(phases)])
rownames(calc_out)<-a[[3]]
calc_out<-rbind(calc_out,bulk_row)
calc_out<-calc_out[,out_colnames,drop=FALSE]
}else{
# Return Blank Comp
colms<-c("wt%",out_colnames)
calc_out<-matrix(0,2,length(out_colnames))
rownames(calc_out)<-c("Error","Bulk_error")
colnames(calc_out)<-out_colnames
}
}
return(calc_out)
}

```

Meemum connect (meemum_connect.r)

Run.meemum calls meemum.exe in the code folder of Rcrust with the current pressure, temperature and bulk composition of the system

Read.meemum formats the outputs from meemum.exe to be readable by Rcrust

Compile matrix of phases

Number duplicate phases

Merge data tables of parameters for each phase

Create Bulk_rs

```
#####
# Run meemum with parameters
#####
#function-def:run.meemum<-
function(meemum.path="",build.file="",meemum.order="",press=press*1000,temp=temp+27
3.15,bulk="",components="",pt_comp=pt_comp)
run.meemum<-function(comps=comps,c0=c0,press=press,temp=temp,meemum_path="") {
  parms<-
c(paste0(data_directory,"/parse_meem"), "n", c(temp+273.15,press*1000), "0", "0")
  res<-
system(shQuote(paste0(data_directory,"/",meemum_path)),invisible=T,input=parms,inte
rn=T)
return(res)
}
#####
# Parse meemum .prn file
#####
read.meemum<-function(meemum_in=meemum_in) {
#Grab data matrix after line given line text
data_matrix_from_line<-function(line_text) {
ln_given<-grep(line_text,meemum_in)
nm_phases<-which(nchar(meemum_in[-(1:ln_given)])==0)[1]-2
text_given<-meemum_in[(ln_given+1):(ln_given+nm_phases+1)]
text_altered<-trimws(text_given)
text_altered<-gsub(" %", "%", text_altered)
# mod+tag: remove system - fluid for now, may be useful in future to have access to
this data
text_altered<-gsub("System - fluid","System-fluid",text_altered)
text_altered<-gsub("Poisson ratio","Poisson_ratio",text_altered)
text_altered<-strsplit(text_altered,split="\s{1,}")
col_names<-c("Phase",text_altered[[1]])
mat_out<-matrix(unlist(text_altered)[-c(1:(length(col_names)-
1))],nm_phases,byrow=TRUE)
colnames(mat_out)<-col_names
#number duplicates
for(ph in mat_out[which(duplicated(mat_out[,1])),1]) {
  mat_out[which(mat_out[,1]==ph)[-1],1]<-
paste0(ph,"_",1:length(which(mat_out[,1]==ph)[-1]))
}
return(mat_out)
}
#merge data tables by common first field with rows in same order
merge_matrix<-function(a,b) {
full_rows<-union(a[,1],b[,1])
#add extra row of NA to any table that needs it
max_row<-max(nrow(a),nrow(b))
if(nrow(a)<max_row) {
a<-rbind(a,matrix(NA,max_row-nrow(a),ncol(a)))
}
}
```

```

if(nrow(b)<max_row){
b<-rbind(b,matrix(NA,max_row-nrow(b),ncol(b)))
}
#rename first row
a[,1]<-full_rows
#delete key row in second table
b<-b[,-1]
#merge
return(cbind(a,b))
}
#grab crust output by merging tables
#reorder major elements
ph_comp_mat<-data_matrix_from_line("Phase Compositions")
ph_comp_mat<-
ph_comp_mat[,c(setdiff(colnames(ph_comp_mat),major_elements),major_elements),drop=F
FALSE]
#merge info
merge_1<-merge_matrix(ph_comp_mat,data_matrix_from_line("Molar Properties and
Density"))
merge_2<-merge_matrix(merge_1,data_matrix_from_line("Seismic Properties"))
#make numeric and set phase column as row names
meemum_out<-merge_2[, -1]
meemum_out<-matrix(as.numeric(meemum_out),nrow(meemum_out))
rownames(meemum_out)<-merge_2[, 1]
colnames(meemum_out)<-colnames(merge_2)[-1]
# mod-tag: remove system - fluid properties for now, may be useful in future to
have access to this data
chk_call<-try(meemum_out["System-fluid",],silent=TRUE)
if(class(chk_call)!="try-error"){
meemum_out<-meemum_out[-grep("System-fluid",rownames(meemum_out)),]
}
#merge phases with the same name
if(merge_duplicates){
rownames(meemum_out)<-gsub("_[0-9]", "",rownames(meemum_out))
while(any(duplicated(rownames(meemum_out)))){
ph<-rownames(meemum_out)[which(duplicated(rownames(meemum_out)))[1]]
thelines<-meemum_out[which(rownames(meemum_out)==ph),]
prop<-"wt%"
new<- .wtd.add(thelines,prop,ph)
meemum_out<-rbind(new,meemum_out[-which(rownames(meemum_out)==ph),])
}
}
#Create Bulk_rs with system
new_names<-rownames(meemum_out)
new_names[which(new_names=="System")]<-"Bulk_rs"
rownames(meemum_out)<-new_names
meemum_out["Bulk_rs",comps]<-c0[comps]
#Set % to 100 for Bulk_rs
chk_call<-try(meemum_out["Bulk_rs","vol%"],silent=TRUE)
if(class(chk_call)!="try-error"){
meemum_out["Bulk_rs","wt%"]<-100
}
chk_call<-try(meemum_out["Bulk_rs","vol%"],silent=TRUE)
if(class(chk_call)!="try-error"){
meemum_out["Bulk_rs","vol%"]<-100
}
chk_call<-try(meemum_out["Bulk_rs","vol%"],silent=TRUE)
if(class(chk_call)!="try-error"){
meemum_out["Bulk_rs","mol%"]<-100
}
}
mass<-matrix(meemum_out[,"wt%"]/100*c0["mass"],nrow(meemum_out),1)
colnames(mass)<-"mass"
meemum_out<-cbind(mass,meemum_out)
return(meemum_out)
}

```

Server (server.r)

Define functions for manipulating data in Rcrust

Define functions to be used for formatting and plotting outputs

Set reactive values and build reactive stores for GUI

Define save function to write GUI inputs into text file

Define load function which reads text file input to populate GUI inputs

Define clear function to reset GUI inputs

Define console function to give access to the direct environment

Define run function to source Rcrust calculation loop with saved input variables

Dynamically create GUI inputs based on current selections

Dynamically create view outputs and filter data relative to the intended output

```
#####  
## Rcrust (server.r)  
#####  
##  
# Functions  
##  
#function-def:  
get_val(y_i,x_i,phase,variable,crust=crust,oxy_num=24,site_ocup="biotite",select=1)  
get_val<-  
function(y_i,x_i,phase,variable,crust=crust,oxy_num=24,site_ocup="biotite",select=1  
) {  
  if(variable=="Temperature"|variable=="Pressure") {  
    chk<-  
    try(input_pt[[y_i]][[x_i]][which(colnames(input_pt[[1]][[1]])==variable)],silent=TR  
UE)  
    if(class(chk=="try-error")) {val<-0} else {  
    val<-input_pt[[y_i]][[x_i]][which(colnames(input_pt[[1]][[1]])==variable)]}  
    return(val)  
  }  
  if(variable=="Min_formula") {  
    return(Min_formula(y_i,x_i,phase,oxy_num,site_ocup,crust=crust)[as.numeric(select)  
])  
  }  
  chk<-try(crust[[y_i]][[x_i]][phase,variable],silent=TRUE)  
  if(class(chk=="try-error")) {val<-0} else {  
  if(class(chk=="NULL")) {val<-NA} else {  
  val<-crust[[y_i]][[x_i]][phase,variable]  
  }  
  }  
  return(val)  
}  
#function-def:sub_brackets(x)  
#substitute brackets for obrac and cbrac  
sub_brackets<-function(x) {  
  return(gsub("\\(", "_obrac_", gsub("\\)", "_cbrac_", x)))  
}  
#function-def:ret_brackets(x)  
#return brackets from place holders obrac and cbrac  
ret_brackets<-function(x) {  
  return(gsub("_obrac_", "\\(", gsub("_cbrac_", "\\)", x)))  
}  
#function-def:.wtd.add(thelines,prop="mass",avname="Averaged")
```

```

.wtd.add<<-function(thelines,prop="mass",avname="Averaged"){
  if(nrow(thelines)==1){
    foo<-thelines
  }else{
    #fix-tag: is N(g) extensive or intensive
    extensive.cn<-c("wt%", "vol%", "mol%", "mol", "mass")
    extensive.cn<-intersect(extensive.cn,colnames(thelines))
    intensive.cn<-setdiff(colnames(thelines),c(prop,extensive.cn,"Phase"))
    foo<-matrix(rep(0,length(colnames(thelines))),nrow=1)
    colnames(foo)<-colnames(thelines)
    suppressWarnings(class(thelines)<-"numeric")
    # Intensive
    foo[,intensive.cn]<-
thelines[,prop]*%thelines[,intensive.cn,drop=F]/sum(thelines[,prop])
    # Extensive
    for(i in extensive.cn){
      foo[1,i]<-sum(thelines[,i])
    }
  }
  rownames(foo)<-avname
  return(foo)
}
#function-def:check_tuple(tuple)
check_tuple<-function(tuple=""){
  if(tuple==""){return(list("Valid tuple",""))}
  split_tuple<-unlist(strsplit(gsub("\\{"," ",gsub("\\}",",",tuple)),split=","))
  if(!length(split_tuple)==2){return(list("Error: Tuple is invalid",NULL))}
  for(k in 1:length(split_tuple)){
    if(is.na(suppressWarnings(as.numeric(split_tuple[k])))){return(list("Error:
Tuple argument is non-numeric",NULL))}
    if(as.numeric(split_tuple[k])<1){return(list("Error: Tuple argument is less
than 1",NULL))}
    if(!as.numeric(split_tuple[k])%1==0){return(list("Error: Tuple argument is not
a whole number",NULL))}
  }
  return(list("Valid tuple",paste0("{",paste(split_tuple,collapse=";"),"}"))
}
#function-def:flip_y(mat)
#matrix reflection in y direction
flip_y<-function(mat){
mat_out<-mat[nrow(mat):1,,drop=FALSE]
return(mat_out)
}
#function-def:flip_x(mat)
#matrix reflection in x direction
flip_x<-function(mat){
mat_out<-mat[,ncol(mat):1,drop=FALSE]
return(mat_out)
}
#function-def:rotate(x)
#Matrix rotation
rotate<-function(x)t(apply(x, 2, rev))
substrRight <- function(x, n){
  substr(x, nchar(x)-n+1, nchar(x))
}
#function-
def:phase_abundance(crust,axis,path=1,p_a=1,p_b=p_a,path_label="Point",input_pt=NULL)
phase_abundance<-
function(crust,axis,path=1,p_a=1,p_b=p_a,path_label="Point",input_pt=NULL){
  #Header
  outname<-switch(axis,
    x=paste0("Phase adundance vs ",path_label," for {",p_a,";",path,"} to
{",p_b,";",path,"}"),
    y=paste0("Phase adundance vs ",path_label," for {",path,";",p_a,"} to
{",path,";",p_b,"}"))
  #Abundance rows

```

```

switch(axis,
  x={pnt_y<-"path";pnt_x<-"p_i"},
  y={pnt_y<-"p_i";pnt_x<-"path"}
)
all_phases<-NULL
for(p_i in p_a:p_b){
  all_phases<-
union(all_phases,rownames(crust[[eval(parse(text=pnt_y))]][[eval(parse(text=pnt_x))
]]))
}
#reorder into rs,es,es_cumul,as
rs_phases<-all_phases[grepl(".*_rs",all_phases)]
rs_bulk<-grep("Bulk_rs",rs_phases)
es_phases<-all_phases[grepl(".*_es",all_phases)]
es_cumul<-grep(".*_es_cumul",es_phases)
as_phases<-all_phases[grepl(".*_as",all_phases)]
all_phases<-c(rs_phases[-rs_bulk],rs_phases[rs_bulk],es_phases[-
es_cumul],es_phases[es_cumul],as_phases)
abundance_rows<-NULL
for(ph in all_phases){
  abundace_phase<-NULL
  for(p_i in p_a:p_b){
    chk_phase<-
try(crust[[eval(parse(text=pnt_y))]][[eval(parse(text=pnt_x))]][ph,"mass"],silent=T
RUE)
    if(class(chk_phase)=="try-error"){
      abundace_phase<-c(abundace_phase,0)
    }else{
      abundace_phase<-
c(abundace_phase,crust[[eval(parse(text=pnt_y))]][[eval(parse(text=pnt_x))]][ph,"ma
ss"])
    }
  }
  abundance_rows<-rbind(abundance_rows,matrix(abundace_phase,1))
}
rownames(abundance_rows)<-all_phases
col_nms<-p_a:p_b
if(!path_label=="Point"){
  if(path_label=="Pressure(kbar)"|path_label=="Temperature(C)"){
    switch(path_label,"Pressure(kbar)"=slct<-1,"Temperature(C)"=slct<-2)
  }
  if(!is.null(input_pt)){
    col_nms<-NULL
    for(p_i in p_a:p_b){
      col_nms<-
c(col_nms,input_pt[[eval(parse(text=pnt_y))]][[eval(parse(text=pnt_x))]][slct])
    }
  }
}
colnames(abundance_rows)<-col_nms
return(list(outname,abundance_rows))
}
#function-
def:write_phase_abundance(data,working_file=working_file,projects_directory=project
s_directory,file_type)
write_phase_abundance<-
function(data,working_file=working_file,projects_directory=projects_directory,file_
type){
  outfile_path<-
paste0(projects_directory,"/",working_file,"/Outputs/",working_file,"
",data[[1]],file_type)
  if(file_type==" .txt"){
    write.table(data[[2]],outfile_path,sep="\t",quote=F,row.names = TRUE)
  }
  if(file_type==" .csv"){
    write_test<-try(write.csv(data[[2]],outfile_path,row.names = TRUE),silent=TRUE)
    if(class(write_test)=="try-error"){

```

```

    cat("Error cannot write to ",outfile_path,", please close all programs that
may be accessing the file then try again\n")
    return(paste0("Error could not save phase abundance file: ",outfile_path,",
file may be open in another program, please
close all programs that may be accessing the file then try again\n"))
}
}
cat("File written to ",data[[1]],"\n")
return(paste0("Phase abundance file saved to
",projects_directory,"/",working_file,"/Outputs/\n"))
}
#function-def:get_PAM_names(crust,PAM_system)
get_PAM_names<-function(crust,PAM_system){
y_n<-length(crust)
x_n<-length(crust[[1]])
#Group phases
PAM<-matrix(0,y_n,x_n)
for(x_i in 1:x_n){
for(y_i in 1:y_n){
#Grab phases from system under consideration
if(PAM_system=="Reactive Subsystem"){
grab<-
rownames(crust[[y_i]][[x_i]])[which(substrRight(rownames(crust[[y_i]][[x_i]]),3)==
_rs")]
}
if(PAM_system=="Extract Subsystem"){
grab<-
rownames(crust[[y_i]][[x_i]])[which(substrRight(rownames(crust[[y_i]][[x_i]]),3)==
_es")]
}
if(PAM_system=="Full System"){
grab<-rownames(crust[[y_i]][[x_i]])
}
if(length(grab)>0){
#Remove bulk arguments
grab<-grab[-which(grab=="Bulk_rs"|grab=="Bulk_es"|grab=="Bulk_es_cumul")]
if(!PAM_system=="Full System"){
#remove identifiers
for(i in 1:length(grab)){
grab[i]<-substr(grab[i],1,nchar(grab[i])-3)
}
}
#Reorder alphabetically
grab<-sort(grab)
}
collapsed<-paste(grab,collapse="+")
if(collapsed==""){collapsed<-"No Phases"}
PAM[y_i,x_i]<-collapsed
}
}
PAM_names<-setdiff(intersect(PAM,PAM),"0")
return(list(PAM_names,PAM))
}
#function-def:PAM_calc(crust,PAM_system,compile_PAM=FALSE,PAM_compilation=NULL)
PAM_calc<-function(crust,PAM_system,compile_PAM=FALSE,PAM_compilation=NULL){
PAM_data<-get_PAM_names(crust,PAM_system)
PAM<-PAM_data[[2]]
if(compile_PAM){
validate(need(file.exists(paste0(sub("/code","/Projects",getwd()),"/Compile/",PAM_c
ompilation," compilation legend.txt")),paste0(PAM_compilation," compilation
legend.txt not found in ",sub("/code","/Projects",getwd()),"/Compile/","\nPlease
compile legend first")))
compilation_names<-
read.table(paste0(sub("/code","/Projects",getwd()),"/Compile/",PAM_compilation,"
compilation legend.txt"))
class(compilation_names)<-"vector"
PAM_names<-as.character(compilation_names[[2]])}else{
PAM_names<-PAM_data[[1]]

```

```

}
# Shade by number of phases (variance)
no_phs<-NULL
for(i in 1:length(PAM_names)) {
  no_phs<-c(no_phs, length(strsplit(PAM_names, split="+", fixed=TRUE)[[i]]))
}
PAM_names<-PAM_names[rev(order(no_phs, PAM_names))]
#Populate grid
for(y_i in 1:length(crust)){
  for(x_i in 1:length(crust[[1]])){
    if(!PAM[y_i, x_i]==0) {
      PAM[y_i, x_i]<-which(PAM_names==PAM[y_i, x_i])
    }
  }
}
mode(PAM)<-"numeric"
# Pull common phases from legend
nam<-strsplit(PAM_names, split="+", fixed=TRUE)
inter<-nam[[1]]
if(length(PAM_names)>1) {
  for(i in 1:length(PAM_names)) {
    inter<-intersect(inter, nam[[i]])
  }
}
all_pres<-paste(inter, collapse="+")
internam<-list()
for(i in 1:length(PAM_names)) {
  internam[[i]]<-setdiff(nam[[i]], inter)
}
new_nam<-list()
for(i in 1:length(PAM_names)) {
  if(length(internam[[i]])>0) {
    new_nam[i]<-paste(internam[[i]], collapse="+")
  } else {
    new_nam[i]<-"-"
  }
}
PAM_names<-unlist(new_nam)
PAM_legend<-1:length(PAM_names)
names(PAM_legend)<-PAM_names
#if compiling save individual legend (phases relevant to this specific section)
compile_PAM_legend<-NULL
if(compile_PAM) {
  compile_PAM_legend<-PAM_legend
  PAM_legend<-PAM_legend[sort((setdiff(intersect(PAM, PAM), "0")))]
}

#Flip y-axis for matrix drawing
PAM<-flip_y(PAM)
library(raster)
library(rgeos)
x<-raster::raster(PAM)
pol <- raster::rasterToPolygons(x, dissolve=TRUE)
#create label ids
pol_ex<-extract(x, pol)
pol_id<-NULL
for(i in 1:length(pol_ex)) {
  pol_id<-c(pol_id, pol_ex[[i]][1])
}
return(list(PAM, PAM_legend, all_pres, pol, pol_id, compile_PAM_legend))
}
#function-
def:data_file(crust, x_n=length(crust[[1]]), y_n=length(crust), choose_columns=NULL, choose_rows=NULL, choose_points="All")
data_file<-
function(crust, x_n=length(crust[[1]]), y_n=length(crust), choose_columns=NULL, choose_rows=NULL, choose_points="All") {
  #####
  #Outputs select_data list
}

```

```

#Settings for outputing data_file

# choose_columns options = from crust
("wt%",comps,"mass","V(J/bar)","H(J)","Gruneisen_T","Ks(bar)","Mu(bar)","V0(km/s)","
"Vp(km/s)","Vs(km/s)","Vp/Vs","Rho(kg/m3)","Cp(J/K)","alpha(1/K)","beta(1/bar)","S(
J/K)","N(g)","Cp/Cv")
#
# from other
("Phase","y_i","x_i","Pressure(kbar)","Temperature(C)")
#
# Brief
#
# All
#
# default = All
#choose_columns<-c("All")
#choose_columns<-c("Brief")

#choose_rows default = All
#choose_rows options = Reactive subsystem
#
# Extract subsystem
#
# Addition subsystem
#choose_rows<-c("All")
#choose_rows<-c("Bio(TCC)_rs","Bulk_rs")

#choose_points default = All

#####
#fix-tag: number duplicate names (feldspar, mica)
validate(need(!choose_points=="","To select points enter arguments seperated by
commas of the form {x_a;y_a} for single points or {x_a;y_a}:{x_b;y_b} for ranges
where a<=b<=n"))
if(choose_points=="All"){choose_points<-paste("{1;1}:{",x_n,";",y_n,"",sep="")}
choose_points<-unlist(strsplit(choose_points,split=","))
choose_points<-
unlist(strsplit(gsub("\\{","",gsub("\\}","",choose_points)),split=","))
choose_points<-strsplit(choose_points,split=":|;")
data_out<-NULL
for(i in 1:length(choose_points)){
x_a<-as.numeric(choose_points[[i]][1])
y_a<-as.numeric(choose_points[[i]][2])
if(length(choose_points[[i]])>2){
x_b<-as.numeric(choose_points[[i]][3])
y_b<-as.numeric(choose_points[[i]][4])
}else{
x_b<-as.numeric(choose_points[[i]][1])
y_b<-as.numeric(choose_points[[i]][2])
}
#error validation on selection (range must be possible i.e. b>=a,b<=n)
validate(need(all(x_a<=x_b,y_a<=y_b,x_b<=x_n,y_b<=y_n),"To select points enter
arguments seperated by commas of the form {x_a;y_a} for single points or
{x_a;y_a}:{x_b;y_b} for ranges where a<=b<=n"))
for(y_i in y_a:y_b){
for(x_i in x_a:x_b){
#fix-tag: create validation here for erroneous pixels
#error validation for point existence
if(is.null(crust[[y_i]][x_i])){
#fix-tag: only works if point 1;1 is populated - dont know how many columns
otherwise
new_pnt<-matrix(NA,1,(ncol(crust[[1]][[1]])+6))
new_pnt[1,1]<-paste("Blank",y_i,x_i,sep="_")
}else{
ID<-matrix(paste(rownames(crust[[y_i]][x_i]),y_i,x_i,sep="_"),ncol=1)
colnames(ID)<-"ID"
phase<-matrix(rownames(crust[[y_i]][x_i]),ncol=1)
colnames(phase)<-"Phase"
pnt<-matrix(c(y_i,x_i),nrow=nrow(crust[[y_i]][x_i]),ncol=2,byrow=TRUE)
colnames(pnt)<-c("y_i","x_i")
if(exists("input_pt")){
p_t<-
matrix(input_pt[[y_i]][x_i],nrow=nrow(crust[[y_i]][x_i]),ncol=2,byrow=TRUE)
}else{

```

```

    p_t<-matrix(0,nrow=nrow(crust[[y_i]][[x_i]]),ncol=2,byrow=TRUE)
  }
  colnames(p_t)<-c("Pressure(kbar)", "Temperature(C)")
  new_pnt<-cbind(ID,phase,pnt,p_t, signif(crust[[y_i]][[x_i]],4))
  rownames(new_pnt)<-ID
  }
  data_out<-rbind(data_out,new_pnt)
}
}
}
if(is.null(choose_columns)){
  choose_columns<-colnames(data_out)
}
if(any(choose_columns=="Brief")){
  choose_columns<-union(choose_columns[-
which(choose_columns=="Brief")],c("ID", "Phase", "y_i", "x_i", "Pressure(kbar)", "Temper
ature(C)", "wt%", comps, "mass"))
}
if(is.null(choose_rows)){
  select_rows<-1:nrow(data_out)
}else{
if(choose_rows=="All"){
select_rows<-1:nrow(data_out)}else{
  select_rows<-NULL
  if(!is.na(match("Reactive subsystem",choose_rows))){
    rs_rows<-grep("_rs",rownames(data_out))
    choose_rows<-choose_rows[-match("Reactive subsystem",choose_rows)]
    select_rows<-union(select_rows,rs_rows)
  }
  if(!is.na(match("Extract subsystem",choose_rows))){
    rs_rows<-grep("_es",rownames(data_out))
    choose_rows<-choose_rows[-match("Extract subsystem",choose_rows)]
    select_rows<-union(select_rows,rs_rows)
  }
  for(row_arg in choose_rows){
    chk_names<-NULL
    for(i in 1:length(rownames(data_out))){
      chk_names<-c(chk_names,paste(strsplit(rownames(data_out),"_")[[i]][c(-3,-
4)],collapse="_"))
    }
    select_rows<-union(select_rows,which(chk_names==row_arg))
  }
}
}
data_out<-data_out[sort(select_rows),choose_columns,drop=FALSE]
rownames(data_out)<-NULL
return(data_out)
}
#function-
def:write_data_file(data_out,working_file=working_file,projects_directory=projects_
directory,file_type=".csv")
write_data_file<-
function(data_out,working_file=working_file,projects_directory=projects_directory,f
ile_type=".csv"){
  #file_type          options = ".csv", ".txt"
  #                   default = ".csv"
outfile_path<-
paste0(projects_directory,"/",working_file,"/Outputs/",working_file,"_data_file",fi
le_type)
if(length(data_out)>0){
if(file_type=="txt"){
  write.table(data_out,outfile_path,sep="\t",quote=F,row.names = FALSE)
  cat("File written to ",outfile_path,"\n")
}
if(file_type=="csv"){
  write_test<-try( write.csv(data_out,outfile_path,row.names =
FALSE),silent=TRUE)
  if(class(write_test)=="try-error"){

```

```

    cat("Error cannot write to ",outfile_path,", please close all programs that
may be accessing the file then try again\n")
    return(paste0("Error could not save Data File: file may be open in another
program, please close all programs that may be accessing the file then try
again\n"))
}
cat("File written to ",outfile_path,"\n")
}
if(file_type==".ps"){
cat("Error cannot write Data File to .ps format\n")
return(paste0("Error cannot write Data File to .ps format\n"))
}
}
return(paste0("Data File saved to
",paste0(projects_directory,"/",working_file,"/Outputs/"),"\n"))
}
#function-
def:grid_data(Grid_variable,Grid_variable_phase="Bulk_rs",crust_in=crust_out(),input_
t_pt_in=input_pt,oxy_num=24,site_ocup="biotite",select=1)
grid_data<-
function(Grid_variable,Grid_variable_phase="Bulk_rs",crust_in=crust_out(),input_pt_
in=input_pt,oxy_num=24,site_ocup="biotite",select=1){
x_n<-length(crust_in[[1]])
y_n<-length(crust_in)
if(!is.null(Grid_variable)){
grid_out_mat<-matrix(0,y_n,x_n)
for(x_i in 1:x_n){
for(y_i in 1:y_n){
grid_out_mat[y_i,x_i]<-
get_val(y_i,x_i,Grid_variable_phase,Grid_variable,crust_in,oxy_num,site_ocup,select
)
}
}
#Flip matrix so origin is bottom left (psuedosection convention)
grid_out_mat<-flip_y(grid_out_mat)

if(Grid_variable=="y_i"|Grid_variable=="x_i"|Grid_variable=="Temperature"|Grid_vari
able=="Pressure"){
grid_out_title<-Grid_variable
}else{
grid_out_title<-paste(Grid_variable_phase,Grid_variable)
}
suppressWarnings(mode(grid_out_mat)<-"numeric")
return(list(grid_out_title,grid_out_mat))
}
}
#function-def:.First()
# Function for shortcut
.First<-function(){
library(utils)
if(!require(shiny)){
install.packages("shiny")}
library(shiny)
#Launch with GUI function
Rcrust<-function(){
#If working directory is x\Projects\y then set to x\code
if(length(grep("Rcrust/Projects/",getwd()))==1){
setwd(paste0(strsplit(getwd(),split="Projects")[[1]][1],"code"))
}
runApp()
}
#function-
def:manual_load(working_file,projects_directory=paste0(substring(getwd(),1,nchar(get
wd())-4),"Projects"))
#Launch without GUI function
manual_load<-
function(working_file,projects_directory=paste0(substring(getwd(),1,nchar(getwd())-
4),"Projects")){

```

```

source(paste0(projects_directory, "/", working_file, "/Inputs/", working_file, ".txt"))
  source("main.r")
}
#launch the GUI
#If working directory is x\Projects\y then set to x\code
if(length(grep("Rcrust/Projects/", getwd()))==1){
  setwd(paste0(strsplit(getwd(), split="Projects")[[1]][1], "code"))
}
runApp()
}
#function-def:Rcrust()
#Launch with GUI function
Rcrust<-function(){
  #If working directory is x\Projects\y then set to x\code
  if(length(grep("Rcrust/Projects/", getwd()))==1){
    setwd(paste0(strsplit(getwd(), split="Projects")[[1]][1], "code"))
  }
  runApp()
}
#Launch without GUI function
manual_load<-
function(working_file, projects_directory=paste0(substring(getwd(), 1, nchar(getwd())-
4), "Projects")){

source(paste0(projects_directory, "/", working_file, "/Inputs/", working_file, ".txt"))
  source("main.r")
}
#function-def:error_handling(working_file, projects_directory)
#error handling on button press
error_handling<-function(working_file, projects_directory){
  if(projects_directory==""){return("Error: no projects directory specified")}
  if(!dir.exists(projects_directory)){return("Error: projects directory does not
exist")}
  if(working_file==""){return("Error: no working file specified")}
  return("error handling passed")}
#function-def:bl(x)
#blank function : returns a value or quotes "" if blank
bl<-function(x){
  if(is.null(x)){return("\\""})else
  {if(x==""){return("\\""})else{
    return(x)
  }}}
#function-def:bl(str_to_phases)
#string to phases function : takes a string e.g. "melt(HP),Pl,q" and returns a
vector of the phases e.g. "melt(HP)", "Pl", "q"
str_to_phases<-function(x){
  x_list<-strsplit(x, ",")
  if(length(x_list)==0){
    x_out<-NULL
  }else{
    x_out<-x_list[[1]][1]
    if(length(x_list[[1]])>1){
      for(mi in 2:length(x_list[[1]])){
        x_out<-c(x_out, x_list[[1]][mi])
      }
    }
  }
  return(x_out)
}
#function-def:pasteq(x)
#pasteq fubction: collapse vector into comma separated terms with quotes around
them
pasteq<-function(x){
  if(length(x)==0){paste0("\\"", x, "\\"", collapse=",")}else{
    a<-""
    y<-NULL
    for(i in 1:length(x)){

```

```

    if(is.null(x[i])){y<-c(y, "")}else{
      if(x[i]==a){y<-c(y, "")}else{
        #if want to remove all quotes (dont know if I want this) then
gsub("'", "", x)
        #remove enclosing quotes if present
x_split<-strsplit(as.character(x[i]), split="")[[1]]
        if(x_split[1]=="\"")x_split[1]<-""
        if(x_split[length(x_split)]=="\"")x_split[length(x_split)]<-""
        x[i]<-paste(x_split, collapse="")

        y<-c(y, x[i])
        paste0("\"", y, "\"", collapse=", ")
      }
    }
  ##
  # Shiny Server
  ##
shinyServer(function(input, output, session) {
  #Stop app when browser closed
  session$onSessionEnded(stopApp)
  #Global Variables (all variables that must be kept between load events needs to
be in the vector keep_on_load)
  keep_on_load<-c("from_clear_button", "from_copy", "first_load")
  from_clear_button<-FALSE
  from_copy<-FALSE
  #Reactive variables (globally accessed)
  #Reactive stores
  pt_definitions_r<-reactiveValues(data = "")
  bulk_definitions_r<-reactiveValues(data = NULL)
  ph_add_definitions_r<-reactiveValues(data = NULL)
  ph_extr_definitions_r<-reactiveValues(data = NULL)
  solution_models_file_r<-reactiveValues(data = NULL)
  available_components_r<-reactiveValues()
  comp_transformations_r<-reactiveValues(data = NULL)
  current_components_r<-reactiveValues(data= NULL)
  major_elements_r<-reactiveValues(data= NULL)
  #mod-tag - this is a quick fix
  use_sol_models_r<-reactiveValues(data = NULL)
  #store_r contains output data from runs
  store_r<-
reactiveValues(crust_r=NULL, input_pt_r=NULL, input_bulk_r=NULL, major_elements_r=NULL
)
  #initialise a passing message for error checking and reporting
  reactive_message <- reactiveValues(data = NULL)
  load_pt_r<-reactiveValues(data = NULL)
  #save data reactive
  save_data_file <- reactive({
    #Grab working file and projects directory
    working_file<-input$working_file
    projects_directory<-input$projects_directory
    #error handling
    reactive_message$data <-error_handling(working_file, projects_directory)
    if(reactive_message$data=="error handling passed"){
      #if error handling is passed
      #Return success message
      return(paste0("File saved to
", projects_directory, "/", working_file, "/Inputs/", working_file, ".txt"))
    }else{
      #Return error message
      return(reactive_message$data)
    }
  })
  #Save function
  on_save <- reactive({
    #Grab working file and projects directory
    working_file<-input$working_file
    projects_directory<-input$projects_directory
    #Replace projects directory if from copy
    if(!exists("from_copy")){from_copy<-FALSE}

```

```

if(from_copy){
  projects_directory<-copy_directory
  from_copy<<-FALSE
}
#error handling
reactive_message$data <-error_handling(working_file,projects_directory)
if(reactive_message$data=="error handling passed"){
  #if error handling is passed
  w_file<-paste0("#####\n#\n#   Rcrust input
file\n#\n#####\n#\n# Location of project files\n",
                "working_file<-",pasteq(working_file),"\n",
                "projects_directory<-",pasteq(projects_directory ),"\n")

if(!input$x_n==""){if(is.na(suppressWarnings(as.numeric(input$x_n)))){return("Error
: X must be numeric")}}
  if(as.numeric(input$x_n)<1){return("Error: X must be greater than 1")}
  if(!as.numeric(input$x_n)%1==0){return("Error: X must be a whole number")}}

if(!input$y_n==""){if(is.na(suppressWarnings(as.numeric(input$y_n)))){return("Error
: Y must be numeric")}}
  if(as.numeric(input$y_n)<1){return("Error: Y must be greater than 1")}
  if(!as.numeric(input$y_n)%1==0){return("Error: Y must be a whole number")}}
  w_size<-paste0("#####\n#\n#   Size data\n#\n#####\n#\n#
number of points in x and y directions\n",
                "x_n<-",bl(input$x_n),"\n",
                "y_n<-",bl(input$y_n),"\n")
  if(input$n_pt_def=="load"){
    if(!all(pt_definitions_r$data=="")){
      list_pt<-NULL
      for(i in 1:length(pt_definitions_r$data)){
        list_pt<-
c(list_pt,paste0("\n",names(pt_definitions_r$data)[i],"\n=c(",pasteq(pt_definitions
_r$data[[i]]),"))")
      }
      pt_definitions<-paste0("list(",paste0(list_pt,collapse=","),")")
    }else{
      pt_definitions<-""
    }
    }else{
    if(!input$n_pt_def==""){
      #Error validation
      if(is.na(suppressWarnings(as.numeric(input$n_pt_def)))){return("Error:
Number of PT definitions must be numeric")}
      if(as.numeric(input$n_pt_def)<1){return("Error: Number of PT definitions
must be greater than 0")}
      if(!as.numeric(input$n_pt_def)%1==0){return("Error: Number of PT
definitions must be a whole number")}
      list_pt<-NULL
      for(i in 1:as.numeric(input$n_pt_def)){
        #check tuples
        from<-check_tuple(eval(parse(text=paste0("input$pt_from_",i))))
        if(!from[[1]]=="Valid tuple"){
          return(paste0("Error in PT Definition:      ",from[[1]]))
        }
        to<-check_tuple(eval(parse(text=paste0("input$pt_to_",i))))
        if(!to[[1]]=="Valid tuple"){
          return(paste0("Error in PT Definition:      ",to[[1]]))
        }
        list_pt<-
c(list_pt,paste0("\n",from[[2]],"_",to[[2]],"\n=c(",pasteq(eval(parse(text=paste0("
input$pressure_",i))))),",",pasteq(eval(parse(text=paste0("input$temperature_",i))))
,""))")
      }
      pt_definitions<-paste0("list(",paste0(list_pt,collapse=","),")")
    }else{
      pt_definitions<-""
    }
  }
}
}

```

```

w_pt<-paste0("#####\n#\n#   PT data\n#\n#####\n",
            "pt_def<-\input\"
            "pt_definitions<-\",bl(pt_definitions),\"\n"
)
if(input$n_comp_trans=="load"){
  if(!all(comp_transformations_r$data=="")){
    list_trans<-NULL
    for(i in 1:length(comp_transformations_r$data)){
      list_trans<-
c(list_trans,paste0("\",names(comp_transformations_r$data)[i],\"=c\",pasteq(comp_
transformations_r$data[[i]]),\""))
    }
    comp_transformations<-paste0("list(",paste0(list_trans,collapse=\",\"),\"")")
  }else{
    comp_transformations<-\"\"
  }
}else{
  if(!input$n_comp_trans==""){
    #Error validation

if(is.na(suppressWarnings(as.numeric(input$n_comp_trans)))){return("Error: Number
of Component transformations must be numeric")}
    if(as.numeric(input$n_comp_trans)<1){return("Error: Number of Component
transformations must be greater than 0")}
    if(!as.numeric(input$n_comp_trans)%%1==0){return("Error: Number of
Component transformations must be a whole number")}
    list_trans<-NULL
    for(i in 1:as.numeric(input$n_comp_trans)){
      list_trans<-
c(list_trans,paste0("\",eval(parse(text=paste0("input$old_comp_\",i))),\"_\",eval(par
se(text=paste0("input$new_comp_\",i))),\"=c\",pasteq(eval(parse(text=paste0("input$
comp_\",i))),\"\"))
    }
    comp_transformations<-paste0("list(",paste0(list_trans,collapse=\",\"),\"")")
  }else{
    comp_transformations<-\"\"
  }
}
}
if(input$n_bulk_def=="load"){
  if(!all(bulk_definitions_r$data=="")){
    list_bulk<-NULL
    for(i in 1:length(bulk_definitions_r$data)){
      list_bulk<-
c(list_bulk,paste0("\",names(bulk_definitions_r$data)[i],\"=c\",pasteq(bulk_defin
itions_r$data[[i]]),\"\"))
    }
    bulk_definitions<-paste0("list(",paste0(list_bulk,collapse=\",\"),\"")")
  }else{
    bulk_definitions<-\"\"
  }
}else{
  if(!input$n_bulk_def==""){
    #Error validation
    if(is.na(suppressWarnings(as.numeric(input$n_bulk_def)))){return("Error:
Number of Bulk definitions must be numeric")}
    if(as.numeric(input$n_bulk_def)<1){return("Error: Number of Bulk
definitions must be greater than 0")}
    if(!as.numeric(input$n_bulk_def)%%1==0){return("Error: Number of Bulk
definitions must be a whole number")}
    list_bulk<-NULL
    for(i in 1:as.numeric(input$n_bulk_def)){
      #check tuples
      from<-check_tuple(eval(parse(text=paste0("input$bulk_from_\",i))))
      if(!from[[1]]=="Valid tuple"){
        return(paste0("Error in Bulk Definition:      \",from[[1]]))
      }
      to<-check_tuple(eval(parse(text=paste0("input$bulk_to_\",i))))
      if(!to[[1]]=="Valid tuple"){

```

```

        return(paste0("Error in Bulk Definition:      ",to[[1]]))
    }
    bulk_eval<-
strsplit(eval(parse(text=paste0("input$bulk_",i))),split=",")[[1]]
    #place in quotes
    for(j in 1:length(bulk_eval)){
        bulk_eval[j]<-pasteq(bulk_eval[j])
    }
    list_bulk<-
c(list_bulk,paste0("\n",from[[2]],"_",to[[2]],"\n=c(",paste(bulk_eval,collapse=",",
,")))
    }
    bulk_definitions<-paste0("list(",paste0(list_bulk,collapse=",",",")")
    }else{
        bulk_definitions<-"
    }
    }
if(!is.null(input$major_elements)){
if(any(input$major_elements=="load")){
    major_elements<-major_elements_r$data
}else{
    major_elements<-unlist(strsplit(input$major_elements,split=","))
}
}else{
    major_elements<-"
}
if(input$bulk_def_file==FALSE){bulk_def<-"input"}else{bulk_def<-"file"}
w_bulk_composition<-paste0("#####\n#\n#      Bulk composition data
\n#\n#####\n#",
                                "comp_transformations<-
c(",bl(comp_transformations),")\n",
                                "bulk_def<-",pasteq(bulk_def),",
#input,file\n",
                                "major_elements<-c(",pasteq(major_elements),")\n",
                                "bulk_definitions<-c(",bl(bulk_definitions),")\n",
                                "bulk_file<-",pasteq(input$bulk_file),"\n"
)
if(input$ph_add){
    if(input$n_ph_add_def=="load"){
        if(!all(ph_add_definitions_r$data=="")){
            list_ph_add<-NULL
            for(i in 1:length(ph_add_definitions_r$data)){
                qt_names<-unlist(lapply(names(ph_add_definitions_r$data[[i]]),pasteq))
                qt_values<-unlist(lapply(ph_add_definitions_r$data[[i]],pasteq))
                list_ph_add<-
c(list_ph_add,paste0("\n",names(ph_add_definitions_r$data)[i],"\n=c(",paste(qt_name
s,qt_values,collapse=",",sep="="),","))
            }
            ph_add_definitions<-paste0("list(",paste0(list_ph_add,collapse=",",",")")
        }else{
            ph_add_definitions<-"
        }
    }else{
        if(!input$n_ph_add_def==""){
            #Error validation
}
if(is.na(suppressWarnings(as.numeric(input$n_ph_add_def)))){return("Error: Number
of Phase Addition definitions must be numeric")}
    if(as.numeric(input$n_ph_add_def)<1){return("Error: Number of Phase
Addition definitions must be greater than 0")}
    if(!as.numeric(input$n_ph_add_def)%1==0){return("Error: Number of
Phase Addition definitions must be a whole number")}
    list_ph_add<-NULL
    for(i in 1:as.numeric(input$n_ph_add_def)){
        ph_add_defs<-NULL
        #check tuples
        from<-check_tuple(eval(parse(text=paste0("input$ph_add_from_",i))))
        if(!from[[1]]=="Valid tuple"){

```

```

        return(paste0("Error in Phase Addition Definition:
",from[[1]]))
    }
    to<-check_tuple(eval(parse(text=paste0("input$ph_add_to_",i))))
    if(!to[[1]]=="Valid tuple"){
        return(paste0("Error in Phase Addition Definition:      ",to[[1]]))
    }
    ph_add_con<-
paste0("condition=",pasteq(eval(parse(text=paste0('input$ph_add_con_',i))))))
    phases<-
unlist(strsplit(eval(parse(text=paste0('input$ph_add_phs_',i))),split=","))
    for(j in 1:length(phases)){
        ph_add_defs<-
c(ph_add_defs,paste0(pasteq(phases[j]),"=",pasteq(eval(parse(text=paste0('input$ph_
add_phs_',i,'_',phases[j]))))))
    }
    list_ph_add<-
c(list_ph_add,paste0("\n",from[[2]],"_",to[[2]],"\n=c(",paste0(c(ph_add_con,ph_add_
defs),collapse=","),"))")
    }
    ph_add_definitions<-
paste0("list(",paste0(list_ph_add,collapse=","),"))")
    }else{
        ph_add_definitions<-" "
    }
    }
    }else{ph_add_definitions<-" "
w_phase_addition<-paste0("#####\n#\n#   Phase
addition\n#\n#####\n",
        "ph_add<-",input$ph_add,"\n",
        "ph_add_definitions<-
c(",bl(ph_add_definitions),")\n"
)
    if(input$ph_extr){
        if(input$n_ph_extr_def=="load"){
            if(!all(ph_extr_definitions_r$data=="")){
                list_ph_extr<-NULL
                for(i in 1:length(ph_extr_definitions_r$data)){
                    qt_names<-
unlist(lapply(names(ph_extr_definitions_r$data[[i]]),pasteq))
                    qt_values<-unlist(lapply(ph_extr_definitions_r$data[[i]],pasteq))
                    list_ph_extr<-
c(list_ph_extr,paste0("\n",names(ph_extr_definitions_r$data)[i],"\n=c(",paste(qt_na
mes,qt_values,collapse=","),sep=""),"))")
                }
                ph_extr_definitions<-
paste0("list(",paste0(list_ph_extr,collapse=","),"))")
            }else{
                ph_extr_definitions<-" "
            }
        }else{
            if(!input$n_ph_extr_def==""){
                #Error validation
            }
        }
    }
    if(is.na(suppressWarnings(as.numeric(input$n_ph_extr_def)))){return("Error: Number
of Phase Extraction definitions must be numeric")}
    if(as.numeric(input$n_ph_extr_def)<1){return("Error: Number of Phase
Extraction definitions must be greater than 0")}
    if(!as.numeric(input$n_ph_extr_def)%%1==0){return("Error: Number of
Phase Extraction definitions must be a whole number")}
    list_ph_extr<-NULL
    for(i in 1:as.numeric(input$n_ph_extr_def)){
        ph_extr_defs<-NULL
        #check tuples
        from<-check_tuple(eval(parse(text=paste0("input$ph_extr_from_",i))))
        if(!from[[1]]=="Valid tuple"){
            return(paste0("Error in Phase Extraction Definition:
",from[[1]]))
        }
    }

```

```

    }
    to<-check_tuple(eval(parse(text=paste0("input$ph_extr_to_",i))))
    if(!to[[1]]=="Valid tuple"){
      return(paste0("Error in Phase Extraction Definition:
",to[[1]]))
    }
    ph_extr_con<-
paste0("condition=",pasteq(eval(parse(text=paste0('input$ph_extr_con_',i))))))
    phases<-
unlist(strsplit(eval(parse(text=paste0('input$ph_extr_phs_',i))),split=","))
      for(j in 1:length(phases)){
        ph_extr_defs<-
c(ph_extr_defs,paste0(pasteq(phases[j]), "=",pasteq(eval(parse(text=paste0('input$ph
_extr_phs_',i,'_',sub_brackets(phases[j]))))))))
      }
      list_ph_extr<-
c(list_ph_extr,paste0("\n",from[[2]], "_",to[[2]], "\n=c(",paste0(c(ph_extr_con,ph_ex
tr_defs),collapse=",","")"))
    }
    ph_extr_definitions<-
paste0("list(",paste0(list_ph_extr,collapse=",","")")
      )else{
        ph_extr_definitions<-""
      }
    }
  }else{ph_extr_definitions<-""}
  w_phase_extraction<-paste0("#####\n#\n#   Phase
extraction\n#\n#####\n",
                             "ph_extr<-",input$ph_extr,"\n",
                             "reequilibrate_steps<-
",input$reequilibrate_steps,"\n",
                             "ph_extr_definitions<-
c(",bl(ph_extr_definitions),")\n"
  )
  if(input$solution_models_file=="load"){
    if(!exists("solution_models_file")){solution_models_file<-""}
  }else{
    solution_models_file<-input$solution_models_file
    use_sol_models_r$data<-input$use_sol_models
  }
  #error validation for meemum path
  if(input$meemum_path==""){return("Error: No meemum path defined")}

if(!file.exists(paste0(data_directory,"/",input$meemum_path))){return(paste0("Error
: Meemum not found at ",paste0(data_directory,"/",input$meemum_path))}
  w_modelling_options<-paste0("#####\n#\n#   Modelling
Options\n#\n#####\n",
                              "thermodynamic_data_file<-
",pasteq(input$thermodynamic_data_file),"\n",
                              "solution_models_file<-
",pasteq(solution_models_file),"\n",
                              "meemum_path<-",pasteq(input$meemum_path),"\n",
                              "perplex_option_file<-
",pasteq(input$perplex_option_file),"\n",
                              "use_sol_models<-
c(",pasteq(use_sol_models_r$data),")\n",
                              "saturated_components<-
",pasteq(input$saturated_components),"\n",
                              "saturated_phase_components<-
",pasteq(input$saturated_phase_components),"\n",
                              "independent_potential_fugacity_activity<-
",pasteq(input$independent_potential_fugacity_activity),"\n",
                              "exclude_phases<-
",pasteq(input$exclude_phases),"\n",
                              "end_of_calc<-",pasteq(input$end_of_calc),"\n")
  w_output_options<-paste0("#####\n#\n#   Output
Options\n#\n#####\n",

```

```

                                "phase_aliases<-
",pasteq(input$phase_aliases), "\n",
                                "PAM_compilation<-
",pasteq(input$PAM_compilation), "\n",
                                "compile_PAM<-",pasteq(input$compile_PAM), "\n")
  #Compile all tabs into a page
  thepage<-
c(w_file,w_size,w_pt,w_bulk_composition,w_phase_addition,w_phase_extraction,w_model
ling_options,w_output_options)
  # If directory doesnt exist, create it
  if(!dir.exists(paste0(projects_directory,"/",working_file))){
    dir.create(paste0(projects_directory,"/",working_file))
  }
  if(!dir.exists(paste0(projects_directory,"/",working_file,"/Inputs/"))){
    dir.create(paste0(projects_directory,"/",working_file,"/Inputs/"))
  }
  if(!dir.exists(paste0(projects_directory,"/",working_file,"/Outputs/"))){
    dir.create(paste0(projects_directory,"/",working_file,"/Outputs/"))
  }
  #Grab additional parameters if file already exists
  add_text<-NULL

if(file.exists(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,"
.txt"))){
  scanned<-
scan(file=paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"
),what="character",sep="\n",quiet=TRUE)
  break_line<-which(scanned=="# Additional Parameters")
  if(length(break_line)==1){
    add_text<-scanned[break_line:length(scanned)]
  }
}
#Save .txt file
write(c(thepage,add_text), file =
paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
#Save workspace
working_file<<-input$working_file

save.image(file=paste0(projects_directory,"/",working_file,"/",working_file,".RData
"))
  #Return success message
  return(paste0("File saved to
",projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
}else{
  #Return error message
  return(reactive_message$data)
}
})
#Load function
on_load <- reactive({
  projects_directory<-input$projects_directory
  working_file<-input$working_file
  #error handling
  reactive_message$data <-error_handling(working_file,projects_directory)
  if(reactive_message$data=="error handling passed"){
    #Remove current workspace but keeping marker: from clear button
    #mod-tag: should we really be clearing workspace? can't we just overwrite
values for load?
    rm(list=ls())
    rm(list=setdiff(ls(envir=.GlobalEnv),c("keep_on_load",keep_on_load)),envir =
.GlobalEnv)
    #clear reactive stores
    pt_definitions_r$data<-NULL
    bulk_definitions_r$data<-NULL
    ph_add_definitions_r$data<-NULL
    ph_extr_definitions_r$data<-NULL
    #Load .txt unless from clear button
    if(from_clear_button){

```

```

    projects_directory<-input$projects_directory
    working_file<-input$working_file
  }else{

source(paste0(input$projects_directory,"/",input$working_file,"/Inputs/",input$work
ing_file,".txt"))
  }
  #Values to load
  load_variables<-
c("x_n"="inp","y_n"="inp","n_pt_def"="reactive","n_comp_trans"="reactive","bulk_def
_file"="checkbox","major_elements"="reactive_select","n_bulk_def"="reactive","bulk_
_file"="inp","ph_add"="checkbox","n_ph_add_def"="reactive","ph_extr"="checkbox","ree
quilibrate_steps"="checkbox","n_ph_extr_def"="reactive","thermodynamic_data_file"="
inp","saturated_components"="inp","saturated_phase_components"="inp","independent_p
otential_fugacity_activity"="inp","exclude_phases"="inp","end_of_calc"="inp","solut
ion_models_file"="reactive","perplex_option_file"="inp","meemum_path"="inp","phase
aliases"="inp","PAM_compilation"="inp","compile_PAM"="checkbox")
  #load reactive stores
  if(exists("pt_definitions")){
    pt_definitions_r$data<-pt_definitions
  }
  if(exists("bulk_definitions")){
    bulk_definitions_r$data<-bulk_definitions
  }
  if(exists("comp_transformations")){
    comp_transformations_r$data<-comp_transformations
  }
  if(exists("ph_add_definitions")){
    ph_add_definitions_r$data<-ph_add_definitions
  }
  if(exists("ph_extr_definitions")){
    ph_extr_definitions_r$data<-ph_extr_definitions
  }
  if(exists("solution_models_file")){
    solution_models_file_r$data<-solution_models_file
  }
  if(exists("use_sol_models")){
    use_sol_models_r$data<-use_sol_models
  }
  if(exists("major_elements")){
    major_elements_r$data<-major_elements
  }
  #custom loads
  #bulk_def
  if(exists("bulk_def")){if(bulk_def=="input"){bulk_def_file<-
FALSE}else{bulk_def_file<-TRUE}}else{bulk_def_file<-FALSE}
  #get addition phases
  if(is.null(ph_add_definitions_r$data[[1]][1])){add_phases<-""}else{
    add_phases<-
paste(setdiff(names(ph_add_definitions_r$data[[1]]),"condition"),collapse=",")
  }
  #get extract phases
  if(is.null(ph_extr_definitions_r$data[[1]][1])){extr_phases<-""}else{
    extr_phases<-
paste(setdiff(names(ph_extr_definitions_r$data[[1]]),"condition"),collapse=",")
  }
  #load values
  for(i in 1:length(load_variables)){
    if(load_variables[i]=="reactive"){
      updateTextInput(session,names(load_variables)[i],value="load")
    }
    if(load_variables[i]=="reactive_select"){
      updateSelectizeInput(session,names(load_variables)[i],selected="load")
    }
    if(load_variables[i]=="inp"){
      if(exists(names(load_variables)[i])){

```

```

updateTextInput(session, names(load_variables)[i], value=eval(parse(text=names(load_variables)[i])))
    }else{
        updateTextInput(session, names(load_variables)[i], value="")
    }
}
#modtag - remove this
if(load_variables[i]=="def_count"){
    def_name<-
sub("n_", "", sub("_def", "_definitions", names(load_variables)[i]))
    if(exists(def_name)){
        if(is.null(eval(parse(text=def_name))[[1]][1])){val_out<-""}
        else{if(eval(parse(text=def_name))[[1]][1]==""){val_out<-""}
            else{val_out<-length(eval(parse(text=def_name)))}}else{val_out<-""}
        updateTextInput(session, names(load_variables)[i], value=val_out)
    }
    if(load_variables[i]=="checkbox"){
        if(exists(names(load_variables)[i])){
            updateCheckboxInput(session, names(load_variables)[i], value =
eval(parse(text=names(load_variables)[i])))
        }else{
            #False is default for all load checkboxes
            updateCheckboxInput(session, names(load_variables)[i], value = FALSE)
        }
    }
    if(load_variables[i]=="select"){
        if(exists(names(load_variables)[i])){
            updateSelectInput(session,
names(load_variables)[i], selected=eval(parse(text=names(load_variables)[i])))
        }else{
            updateSelectInput(session, names(load_variables)[i], selected="")
        }
    }
}
#Return success message
if(from_clear_button){
    from_clear_button<-FALSE
    return(paste0("Inputs and workspace cleared"))
}else{
    return(paste0("Loaded
", projects_directory, "/", working_file, "/Inputs/", working_file, ".txt"))
}
}else{
    #Return error message
    return(reactive_message$data)
}
})
#Toolbar buttons
#Load button
observeEvent(input$load, {
    #load the file from input$working_file
    working_file<-input$working_file
    projects_directory<-input$projects_directory
    #error handling
    reactive_message$data <-error_handling(working_file, projects_directory)
    if(reactive_message$data=="error handling passed"){
        #load

if(file.exists(paste0(projects_directory, "/", working_file, "/Inputs/", working_file, ".txt"))){
    reactive_message$data <- paste0(on_load())
}else{
    reactive_message$data <- paste0("No input file found at
", paste0(projects_directory, "/", working_file, "/Inputs/", working_file, ".txt"))
}
    #Load workspace if it exists (previous calculation results)

```

```

if(file.exists(paste0(projects_directory,"/",working_file,"/",working_file,".RData"
))){

load(paste0(projects_directory,"/",working_file,"/",working_file,".RData"),envir=.G
lobalEnv)
  }
  #Refresh reactive outputs if they exist
  if(exists("crust")){store_r$crust_r<-crust}
  if(exists("input_pt")){store_r$input_pt_r<-input_pt}
  if(exists("input_bulk")){store_r$input_bulk_r<-input_bulk}
  if(exists("major_elements")){store_r$major_elements_r<-major_elements}
}else{
  reactive_message$data
}
})
#Clear button
observeEvent(input$clear, {
  #delete selection inputs
  from_clear_button<<-TRUE
  reactive_message$data <- paste0(on_load())
})
#Console button
observeEvent(input$consoles, {
  cat("To regain access to the Rcrust GUI type \'c\' then press enter\n")
  browser()
})
#Run Button
observeEvent(input$run, {
  #Grab working file and projects directory
  working_file<-input$working_file
  projects_directory<-input$projects_directory
  #error handling
  reactive_message$data <-error_handling(working_file,projects_directory)
  if(reactive_message$data=="error handling passed"){
    #if error handling is passed
    #GUI level validation check that have at least minimum of P,T,and X
    definitions in GUI
    var_missing<-NULL
    chk_variables<-
c("input$x_n","input$y_n","input$n_pt_def","input$n_bulk_def","input$major_elements
")
    for(i in rev(chk_variables)){if(any(eval(parse(text=i))=="")){var_missing<-
i}}
    if(!is.null(var_missing)){reactive_message$data <- paste0("Error: Cannot run
calculation. Missing ",var_missing);return(reactive_message$data)}
    #Save
    reactive_message$data <- paste0(on_save())
    # source the saved variables into the workspace

source(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
  #Run
  source("main.r")
  reactive_message$data <- paste0("Calculation complete, Results saved to
",projects_directory,"/",working_file,"/",working_file,".RData\n Select outputs
throughout the 'Outputs' tab")
  #Save copy to directory
  if(FALSE){copy_directory<<- "H:/Rcrust/Projects"
  from_copy<<-TRUE
  reactive_message$data <- paste0(on_save())
}
  #End of Calculation
if(!exists("end_of_calc")){end_of_calc<- "Return to Interface"}
#Email report
if(FALSE){
  library(gmailr)
  mime() %>%
  to("mjmayne@outlook.com") %>%

```

```

    from("matt.mayne.1992@gmail.com") %>%
    html_body("Body text") -> html_msg
    html_msg %>%
    subject(paste0(working_file, " completed successfully after ", run_time, "with
finish option ", end_of_calc)) %>%

#attach_file(paste0(projects_directory, "/", working_file, "/", working_file, ".RData"))
-> file_attachment
  #attach_file(paste0(projects_directory, "/Yak_PT fh any_all/Yak_PT fh
any_all.RData")) -> file_attachment
  send_message()
}
}
if(end_of_calc=="Logout"){system('shutdown -l')}
if(end_of_calc=="Shutdown"){system('shutdown -s')}
#Refresh reactive outputs if they exist
if(exists("crust")){store_r$crust_r<-crust}
if(exists("input_pt")){store_r$input_pt_r<-input_pt}
if(exists("input_bulk")){store_r$input_bulk_r<-input_bulk}
if(exists("major_elements")){store_r$major_elements_r<-major_elements}
}
})
#Save button
observeEvent(input$save, {
  reactive_message$data <- paste0(on_save())
})
#Send error/success messages to GUI
output$print_message <- renderText({
  if (is.null(reactive_message$data)) return()
  paste0(reactive_message$data)
})
# Dynamically use number of PT definitions to create the correct number of
From, To, P, T inputs
output$pt <- renderUI({
  if(input$n_pt_def=="load"){
    if(all(pt_definitions_r$data=="") | is.null(pt_definitions_r$data)){
      def_num<=""
    }else{
      def_num<-length(pt_definitions_r$data)
    }
    updateTextInput(session, "n_pt_def", value=def_num)
  }else{
    def_num<-input$n_pt_def
  }
  if(!(is.null(def_num) | def_num=="" | def_num==0)){
    validate(if(is.na(suppressWarnings(as.numeric(def_num)))){"Error: Number of
PT definitions must be numeric"}else{NULL})
    validate(if(as.numeric(def_num)<1){"Error: Number of PT definitions must be
greater than 0"}else{NULL})
    validate(if(!as.numeric(def_num)%%1==0){"Error: Number of PT definitions must
be a whole number"}else{NULL})
    fixedRow(
      lapply(1:(as.numeric(def_num)*4), function(i) {
        a<-1; ii<-i
        while(ii>4){ii<-ii-4;a<-a+1}
        if(ii==1){chk_From<-
try(unlist(strsplit(names(pt_definitions_r$data)[a], split="_"))[1], silent=TRUE)
        if(class(chk_From)=="try-error"){chk_From<-NULL}

column(2, textInput(paste0('pt_from_', a), 'From', value=chk_From))}else{
        if(ii==2){chk_To<-
try(unlist(strsplit(names(pt_definitions_r$data)[a], split="_"))[2], silent=TRUE)
        if(class(chk_To)=="try-error"){chk_To<-NULL}
        column(2, textInput(paste0('pt_to_', a), 'To', value=chk_To))}else{
        if(ii==3){chk_pressure<-try(pt_definitions_r$data[[a]][1], silent=TRUE)
        if(class(chk_pressure)=="try-error"){chk_pressure<-NULL}
        column(4, textInput(paste0('pressure_', a), 'Pressure
(kbar)', value=chk_pressure))}else{
        if(ii==4){chk_temperature<-try(pt_definitions_r$data[[a]][2], silent=TRUE)

```

```

        if(class(chk_temperature=="try-error"){chk_temperature<-NULL}
        column(4,textInput(paste0('temperature_',a),'Temperature
(C)',value=chk_temperature))
    }
  }
  })
  #Import P-T definitions button
  observeEvent(input$import_pt, {
    #bugtag - try rather updating text to have only a local load
    #updateTextInput(session,input$n_ph_add_def,value="load")
    reactive_message$data <- paste0(on_save())
    #Import definitions from file (csv of definitions or txt as Rcrust input file)
    #read Rcrust Input file
    if(input$file_pt["type"]=="text/plain"){
      data_in<-readLines(paste(input$file_pt[4]))
      input_file<-
readLines(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"
))
      if(!length(grep("pt_definitions<-",data_in))==0){
        if(length(grep("pt_definitions<-",input_file))==0){
          input_file<-c(input_file,grep("pt_definitions<-",data_in))
        }else{
          input_file[grep("pt_definitions<-",input_file)]<-data_in[grep("pt_definitions<-
",data_in)]
        }
      }
      write(input_file, file =
paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
    }else{
      #read csv of definitions
      data_in<-as.matrix(read.csv(paste(input$file_pt[4])))
      lines_all<-NULL
      for(line_no in 1:nrow(data_in)){
        line_i<-
paste("\",{",data_in[line_no,1],";",data_in[line_no,2],"}_{" ,data_in[line_no,3],";",
data_in[line_no,4],"}\`=c(\`,`",data_in[line_no,5],"\`,`",data_in[line_no,6],"\`)" ,s
ep="")
        if(line_no==1){lines_all<-paste("pt_definitions<-list(",line_i,sep="")}else{
          lines_all<-paste(lines_all,line_i,sep=",")
        }
        if(line_no==nrow(data_in)){
          lines_all<-paste(lines_all,")",sep="")
        }
      }
      input_file<-
readLines(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"
))
      if(length(grep("pt_definitions<-",input_file))==0){
        input_file<-c(input_file,lines_all)
      }else{
        input_file[grep("pt_definitions<-",input_file)]<-lines_all
      }
      write(input_file, file =
paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
    }
    reactive_message$data <- paste0(on_load())
  })
  # Dyanmically use number of component transformations to create the correct
number of inputs
  output$trans <- renderUI({
    if(input$n_comp_trans=="load"){
if(all(comp_transformations_r$data=="")|is.null(comp_transformations_r$data)){
  def_num<-""
}else{
  def_num<-length(comp_transformations_r$data)
}
}
}

```

```

updateTextInput(session, "n_comp_trans", value=def_num)
}else{
  def_num<-input$n_comp_trans
}
if(!(is.null(def_num)|def_num==""|def_num==0)){
  validate(if(is.na(suppressWarnings(as.numeric(def_num)))){"Error: Number of
Component transformations must be numeric"}else{NULL})
  validate(if(as.numeric(def_num)<1){"Error: Number of Component
transformations must be greater than 0"}else{NULL})
  validate(if(!as.numeric(def_num)%%1==0){"Error: Number of Components
transformations must be a whole number"}else{NULL})
  fixedRow(
    lapply(1:(as.numeric(def_num)*3), function(i) {
      a<-1; ii<-i
      while(ii>3){ii<-ii-3;a<-a+1}
      if(ii==1){chk_old<-
try(unlist(strsplit(names(comp_transformations_r$data)[a],split="_"))[1],silent=TRU
E)
      if(class(chk_old)=="try-error"){chk_old<-NULL}
      column(2,textInput(paste0('old_comp_',a), 'Replace
component',value=chk_old))}else{
      #mod-tag: better to have a select input but for some reason it doesnt
work well with column()

#column(2,selectizeInput(paste0('old_comp_',a), 'Replace',c("",available_components_
r$data),selected=chk_old))}else{
      if(ii==2){chk_new<-
try(unlist(strsplit(names(comp_transformations_r$data)[a],split="_"))[2],silent=TRU
E)
      if(class(chk_new)=="try-error"){chk_new<-NULL}
      column(2,textInput(paste0('new_comp_',a), 'New
component',value=chk_new))}else{
      if(ii==3){
      comp_label<-paste(available_components_r$current,collapse = ",")
      chk_comp<-try(comp_transformations_r$data[[a]],silent=TRUE)
      if(class(chk_comp)=="try-error"){chk_comp<-NULL}

column(8,textInput(paste0('comp_',a),comp_label,value=paste(chk_comp,collapse=","))
)}else{
      })))
    )
  }
})
#Component transformation
observe({
  available_components<-NULL
  suppressWarnings(qq_try<-
try(scan(file=gsub("Rcrust/code",paste0("Rcrust/data/",input$thermodynamic_data_fil
e),getwd()),what="character",sep="\n",quiet=TRUE),silent=TRUE){
  if(class(qq_try)!="try-error"){
    qq<-qq_try
    start_foo<-grep("begin_components",qq)
    end_foo<-grep("end_components",qq)
    qq<-qq[(start_foo+1):(end_foo-1)]
    qq<-strsplit(qq," ")
    for(i in 1:length(qq)){
      available_components<-c(available_components,qq[[i]][1])
    }
  }
  available_components_r$current<-available_components
})
#Component transformation
observe({
  #use current component transformation inputs to update current components_r
and selection in major elements
  current_components<-available_components_r$current
  chk_trans<-FALSE
  if(!is.na(suppressWarnings(as.numeric(input$n_comp_trans)))){

```

```

    if(!as.numeric(input$n_comp_trans)<1){
      if(as.numeric(input$n_comp_trans)%1==0){
        chk_trans<-TRUE
      }
    }
  }}
  if(chk_trans){
    for(i in 1:input$n_comp_trans){
      pos<-
which(current_components==eval(parse(text=paste0("input$old_comp_",i))))
      current_components[pos]<-eval(parse(text=paste0("input$new_comp_",i)))
    }
  }
  current_components_r$data<-current_components
  })
  output$maj <- renderUI({
    #fix-tag: does not work if have multiple component transformations, scoping
means we come here after first transformation is complete because we alter "current
components"
    if(!input$n_comp_trans=="load"){
      if(any(input$major_elements=="load")){
        selectizeInput('major_elements', 'Major elements',
c(major_elements_r$data, setdiff(current_components_r$data, major_elements_r$data), "l
oad"), selected=major_elements_r$data, multiple=TRUE)
      }else{
        selectizeInput('major_elements', 'Major elements',
c(input$major_elements, setdiff(current_components_r$data, input$major_elements), "loa
d"), selected=input$major_elements, multiple=TRUE)
      }
    }
  })
  # Dynamically use number of bulk definitions to create the correct number of
From,To,bulk inputs
  output$bulk <- renderUI({
    if(input$n_bulk_def=="load"){
      if(all(bulk_definitions_r$data=="") | is.null(bulk_definitions_r$data)){
        def_num<-" "
      }else{
        def_num<-length(bulk_definitions_r$data)
      }
      updateTextInput(session, "n_bulk_def", value=def_num)
    }else{
      def_num<-input$n_bulk_def
    }
    if(!(is.null(def_num) | def_num==" " | def_num==0)){
      validate(if(is.na(suppressWarnings(as.numeric(def_num)))){"Error: Number of
Bulk definitions must be numeric"}else{NULL})
      validate(if(as.numeric(def_num)<1){"Error: Number of Bulk definitions must
be greater than 0"}else{NULL})
      validate(if(!as.numeric(def_num)%1==0){"Error: Number of Bulk definitions
must be a whole number"}else{NULL})
      fixedRow(
        lapply(1:(as.numeric(def_num)*3), function(i) {
          a<-1; ii<-i
          while(ii>3){ii<-ii-3;a<-a+1}
          if(ii==1){chk_From<-
try(unlist(strsplit(names(bulk_definitions_r$data)[a], split="_"))[1], silent=TRUE)
          if(class(chk_From=="try-error"){chk_From<-NULL}
          column(2, textInput(paste0('bulk_from_', a), 'From', value=chk_From))}else{
            if(ii==2){chk_To<-
try(unlist(strsplit(names(bulk_definitions_r$data)[a], split="_"))[2], silent=TRUE)
            if(class(chk_To=="try-error"){chk_To<-NULL}
            column(2, textInput(paste0('bulk_to_', a), 'To', value=chk_To))}else{
              if(ii==3){
                if(is.null(input$major_elements)){
                  bulk_label<-"Please choose major elements above"
                }else{
                  bulk_label<-paste(c(input$major_elements, "mass"), collapse =
", ")

```

```

    }
    chk_bulk<-try(bulk_definitions_r$data[[a]],silent=TRUE)
    if(class(chk_bulk)=="try-error"){chk_bulk<-NULL}

column(8,textInput(paste0('bulk_',a),bulk_label,value=paste(chk_bulk,collapse=","))
)else{
  }}}}
  )
}
})

#Import bulk definitions button
observeEvent(input$import_bulk, {
  #mod-tag: try rather updating text to have only a local load
#updateTextInput(session,input$n_ph_add_def,value="load")
  reactive_message$data <- paste0(on_save())
  #Import definitions from file (csv of definitions or txt as Rcrust input file)
  #read Rcrust Input file
  if(input$file_bulk["type"]=="text/plain"){
    data_in<-readLines(paste(input$file_bulk[4]))
    input_file<-
readLines(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"
))
    if(!length(grep("comp_transformations<-",data_in))==0){
      if(length(grep("comp_transformations<-",input_file))==0){
        input_file<-c(input_file,grep("comp_transformations<-",data_in))
      }else{
        input_file[grep("comp_transformations<-",input_file)]<-
data_in[grep("comp_transformations<-",data_in)]
      }
    }
    if(!length(grep("major_elements<-",data_in))==0){
      if(length(grep("major_elements<-",input_file))==0){
        input_file<-c(input_file,grep("major_elements<-",data_in))
      }else{
        input_file[grep("major_elements<-",input_file)]<-data_in[grep("major_elements<-",
data_in)]
      }
    }
    if(!length(grep("bulk_definitions<-",data_in))==0){
      if(length(grep("bulk_definitions<-",input_file))==0){
        input_file<-c(input_file,grep("bulk_definitions<-",data_in))
      }else{
        input_file[grep("bulk_definitions<-",input_file)]<-
data_in[grep("bulk_definitions<-",data_in)]
      }
    }
    write(input_file, file =
paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
  }else{
    #read csv of definitions
    data_in<-as.matrix(read.csv(paste(input$file_bulk[4])))
    lines_all<-NULL
    for(line_no in 1:nrow(data_in)){
      line_i<-
paste("\",{",data_in[line_no,1],";",data_in[line_no,2],"}_{",data_in[line_no,3],";",
data_in[line_no,4],"}\`=c(\``,paste(data_in[line_no,c(-1,-2,-3,-
4)],collapse="\`,`",sep="")`",sep="")
      if(line_no==1){lines_all<-paste("bulk_definitions<-c(list(",line_i,sep=""))}else{
        lines_all<-paste(lines_all,line_i,sep=","")
      }
      if(line_no==nrow(data_in)){
        lines_all<-paste(lines_all,")",sep="")
      }
    }
    input_file<-
readLines(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"
))
    if(length(grep("bulk_definitions<-",input_file))==0){

```

```

input_file<-c(input_file,lines_all)
}else{
input_file[grep("bulk_definitions<-",input_file)]<-lines_all
}
write(input_file, file =
paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
}
reactive_message$data <- paste0(on_load())
})
# Dynamically use number of Phase Addition definitions to create the correct
number of From,To,P,T inputs
output$ph_add <- renderUI({
  if(input$n_ph_add_def=="load"){
    if(all(ph_add_definitions_r$data=="")|is.null(ph_add_definitions_r$data)){
      def_num<-" "
    }else{
      def_num<-length(ph_add_definitions_r$data)
    }
    updateTextInput(session,"n_ph_add_def",value=def_num)
  }else{
    def_num<-input$n_ph_add_def
  }
  if(!(is.null(def_num)|def_num==""|def_num==0)){
    validate(if(is.na(suppressWarnings(as.numeric(def_num)))){"Error: Number of
Phase Addition definitions must be numeric"}else{NULL})
    validate(if(as.numeric(def_num)<1){"Error: Number of Phase Addition
definitions must be greater than 0"}else{NULL})
    validate(if(!as.numeric(def_num)%%1==0){"Error: Number of Phase Addition
definitions must be a whole number"}else{NULL})
    fixedRow(
      lapply(1:(as.numeric(def_num)*5), function(i) {
        a<-1; ii<-i
        while(ii>5){ii<-ii-5;a<-a+1}
        if(ii==1){chk_From<-
try(unlist(strsplit(names(ph_add_definitions_r$data)[a],split=" "))[1],silent=TRUE)
        if(class(chk_From=="try-error"){chk_From<-NULL}

column(2,textInput(paste0('ph_add_from_',a),'From',value=chk_From))}else{
        if(ii==2){chk_To<-
try(unlist(strsplit(names(ph_add_definitions_r$data)[a],split=" "))[2],silent=TRUE)
        if(class(chk_To=="try-error"){chk_To<-NULL}
        column(2,textInput(paste0('ph_add_to_',a),'To',value=chk_To))}else{
        if(ii==3){chk_Condition<-
try(ph_add_definitions_r$data[[a]][["condition"],silent=TRUE)
        if(class(chk_Condition=="try-error"){chk_Condition<-NULL}

column(3,textInput(paste0('ph_add_con_',a),'Condition',value=chk_Condition))}else{
        if(ii==4){
          chk_Phases<-try(paste(names(ph_add_definitions_r$data[[a]][-
1]),collapse=""),silent=TRUE)
          if(class(chk_Phases=="try-error"){chk_Phases<-NULL}

column(4,textInput(paste0('ph_add_phs_',a),'Phases',value=chk_Phases))
        }else{
        if(ii==5){
          column(12,uiOutput(paste0("ph_add_",a)))
        }
        }}
      })))
  )
}
})
#auto create phase addition inputs given number of phases
observe({

if(all(!input$n_ph_add_def=="load",!is.null(input$n_ph_add_def),!input$n_ph_add_def
=="",!input$n_ph_add_def==0)){

```

```

lapply(1:input$n_ph_add_def, function(i) {
eval(parse(text=paste0("output$ph_add_",i,"<-renderUI({
a<-i
chk_Phases<-try(eval(parse(text=paste0(\"input$ph_add_phs_\",a))),silent=TRUE)
if(!is.null(chk_Phases)){
if(!chk_Phases==""){
phases<-unlist(strsplit(chk_Phases,split=\",\"))
fixedRow(
lapply(1:(length(phases)), function(j) {
chk_in_ph<-
try(ph_add_definitions_r$data[[a]][phases[j]],silent=TRUE)
if(class(chk_in_ph)=="try-error"){chk_in_ph<-NULL}

column(3,textInput(paste0(\"ph_add_phs_\",a,\"_\",phases[j]),phases[j],value=chk_in_ph))
}))
})
}))

})
}
})

#Import addition definitions button
observeEvent(input$import_ph_add, {
#mod-tag: try rather updating text to have only a local load
#updateTextInput(session,input$n_ph_add_def,value="load")
reactive_message$data <- paste0(on_save())
#Import definitions from file (csv of definitions or txt as Rcrust input file)
#read Rcrust Input file
if(input$file_ph_add["type"]=="text/plain"){
data_in<-readLines(paste(input$file_ph_add[4]))
input_file<-
readLines(paste0(projects_directory, "/", working_file, "/Inputs/", working_file, ".txt"
))
if(!length(grep("ph_add<-",data_in))==0){
if(length(grep("ph_add<-",input_file))==0){
input_file<-c(input_file,grep("ph_add<-",data_in))
}else{
input_file[grep("ph_add<-",input_file)]<-data_in[grep("ph_add<-",data_in)]
}
}
if(!length(grep("ph_add_definitions<-",data_in))==0){
if(length(grep("ph_add_definitions<-",input_file))==0){
input_file<-c(input_file,grep("ph_add_definitions<-",data_in))
}else{
input_file[grep("ph_add_definitions<-",input_file)]<-
data_in[grep("ph_add_definitions<-",data_in)]
}
}
write(input_file, file =
paste0(projects_directory, "/", working_file, "/Inputs/", working_file, ".txt"))
}else{
#read csv of definitions
data_in<-as.matrix(read.csv(paste(input$file_ph_add[4])))
lines_all<-NULL
for(line_no in 1:nrow(data_in)){
line_i<-
paste("\{" ,data_in[line_no,1],";",data_in[line_no,2],"}_{" ,data_in[line_no,3],";",
data_in[line_no,4],"}\}"=c(condition="",data_in[line_no,5],"\",",data_in[line_no,6]
,")",sep="")
if(line_no==1){lines_all<-paste("ph_add_definitions<-
c(list(",line_i,sep=""))}else{
lines_all<-paste(lines_all,line_i,sep=",")
}
if(line_no==nrow(data_in)){
lines_all<-paste(lines_all,")",sep="")
}
}
}
}
}

```

```

input_file<-
readLines(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"
))
  if(length(grep("ph_add<-",input_file))==0){
input_file<-c(input_file,"ph_add<-TRUE")
}else{
input_file[grep("ph_add<-",input_file)]<-"ph_add<-TRUE"
}
if(length(grep("ph_add_definitions<-",input_file))==0){
input_file<-c(input_file,lines_all)
}else{
input_file[grep("ph_add_definitions<-",input_file)]<-lines_all
}
write(input_file, file =
paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
}
reactive_message$data <- paste0(on_load())
updateTextInput(session,input$file_ph_add,value="")
})
  #Import extraction definitions button
observeEvent(input$import_ph_extr, {
#mod-tag: try rather updating text to have only a local load
#updateTextInput(session,input$n_ph_extr_def,value="load")
reactive_message$data <- paste0(on_save())
#Import definitions from file (csv of definitions or txt as Rcrust input file)
#read Rcrust Input file
if(input$file_ph_extr["type"]=="text/plain"){
data_in<-readLines(paste(input$file_ph_extr[4]))
input_file<-
readLines(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"
))
  if(!length(grep("ph_extr<-",data_in))==0){
if(length(grep("ph_extr<-",input_file))==0){
input_file<-c(input_file,grep("ph_extr<-",data_in))
}else{
input_file[grep("ph_extr<-",input_file)]<-data_in[grep("ph_extr<-",data_in)]
}
}
if(!length(grep("reequilibrate_steps<-",data_in))==0){
if(length(grep("reequilibrate_steps<-",input_file))==0){
input_file<-c(input_file,grep("reequilibrate_steps<-",data_in))
}else{
input_file[grep("reequilibrate_steps<-",input_file)]<-
data_in[grep("reequilibrate_steps<-",data_in)]
}
}
if(!length(grep("ph_extr_definitions<-",data_in))==0){
if(length(grep("ph_extr_definitions<-",input_file))==0){
input_file<-c(input_file,grep("ph_extr_definitions<-",data_in))
}else{
input_file[grep("ph_extr_definitions<-",input_file)]<-
data_in[grep("ph_extr_definitions<-",data_in)]
}
}
write(input_file, file =
paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
}else{
#read csv of definitions
data_in<-as.matrix(read.csv(paste(input$file_ph_extr[4])))
lines_all<-NULL
for(line_no in 1:nrow(data_in)){
line_i<-
paste("\{"",data_in[line_no,1],";",data_in[line_no,2],"}_{",data_in[line_no,3],";",
data_in[line_no,4],"}\}"=c(condition="\\"",data_in[line_no,5],"\\"",data_in[line_no,6]
,")",sep="")
if(line_no==1){lines_all<-paste("ph_extr_definitions<-
c(list(",line_i,sep="") )else{
lines_all<-paste(lines_all,line_i,sep=",")
}
}
}

```

```

}
if(line_no==nrow(data_in)){
lines_all<-paste(lines_all,"")",sep=""
}
}
input_file<-
readLines(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"
))
input_file[grep("ph_extr_definitions",input_file)]<-lines_all
write(input_file, file =
paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
}
reactive_message$data <- paste0(on_load())
updateTextInput(session,input$file_ph_extr,value="")
})
# Dynamically use number of Phase Extraction definitions to create the correct
number of From,To,P,T inputs
output$ph_extr <- renderUI({
if(input$n_ph_extr_def=="load"){
if(all(ph_extr_definitions_r$data=="") | is.null(ph_extr_definitions_r$data)){
def_num<=""
}else{
def_num<-length(ph_extr_definitions_r$data)
}
updateTextInput(session,"n_ph_extr_def",value=def_num)
}else{
def_num<-input$n_ph_extr_def
}
if(!(is.null(def_num) | def_num=="" | def_num==0)){
validate(if(is.na(suppressWarnings(as.numeric(def_num)))){"Error: Number of
Phase Extraction definitions must be numeric"}else{NULL})
validate(if(as.numeric(def_num)<1){"Error: Number of Phase Extraction
definitions must be greater than 0"}else{NULL})
validate(if(!as.numeric(def_num)%%1==0){"Error: Number of Phase Extraction
definitions must be a whole number"}else{NULL})
fixedRow(
lapply(1:(as.numeric(def_num)*5), function(i) {
a<-1; ii<-i
while(ii>5){ii<-ii-5;a<-a+1}
if(ii==1){chk_From<-
try(unlist(strsplit(names(ph_extr_definitions_r$data)[a],split="_"))[1],silent=TRUE
)
if(class(chk_From)=="try-error"){chk_From<-NULL}
column(2,textInput(paste0('ph_extr_from_',a),'From',value=chk_From))}else{
if(ii==2){chk_To<-
try(unlist(strsplit(names(ph_extr_definitions_r$data)[a],split="_"))[2],silent=TRUE
)
if(class(chk_To)=="try-error"){chk_To<-NULL}
column(2,textInput(paste0('ph_extr_to_',a),'To',value=chk_To))}else{
if(ii==3){chk_Condition<-
try(ph_extr_definitions_r$data[[a]][["condition"],silent=TRUE)
if(class(chk_Condition)=="try-error"){chk_Condition<-NULL}
column(3,textInput(paste0('ph_extr_con_',a),'Condition',value=chk_Condition))}else{
if(ii==4){
chk_Phases<-try(paste(names(ph_extr_definitions_r$data[[a]][-
1]),collapse=""),silent=TRUE)
if(class(chk_Phases)=="try-error"){chk_Phases<-NULL}
column(4,textInput(paste0('ph_extr_phs_',a),'Phases',value=chk_Phases))
}else{
if(ii==5){
column(12,uiOutput(paste0("ph_extr_",a)))
}
}}}})
)
)

```

```

    }
  })
#auto create phase extraction inputs given number of phases
  observe({

if(all(!input$n_ph_extr_def=="load",!is.null(input$n_ph_extr_def),!input$n_ph_extr_def=="",!input$n_ph_extr_def==0)){
  lapply(1:input$n_ph_extr_def, function(i) {
    eval(parse(text=paste0("output$ph_extr_",i,"<-renderUI({
a<-i
chk_Phases<-try(eval(parse(text=paste0(\'input$ph_extr_phs_\',a))),silent=TRUE)
      if(!is.null(chk_Phases)){
        if(!chk_Phases==""){
          phases<-unlist(strsplit(chk_Phases,split=\\",\\"))
          fixedRow(
            lapply(1:length(phases), function(j) {
              chk_in_ph<-
try(ph_extr_definitions_r$data[[a]][phases[j]],silent=TRUE)
                if(class(chk_in_ph)=="try-error"){chk_in_ph<-NULL}

column(3,textInput(paste0(\'ph_extr_phs_\',a,\'_\',sub_brackets(phases[j])),phases[j],value=chk_in_ph))
              )))
            })
          })
        })
      })
    })
  })
# Dyanmically use solution model file to build phase models selection
  output$solution_models <- renderUI({
    if(input$solution_models_file=="load"){

if(all(solution_models_file_r$data=="")|is.null(solution_models_file_r$data)){
  solution_models_file<-" "
}else{
  solution_models_file<-solution_models_file_r$data
}
  updateTextInput(session,"solution_models_file",value=solution_models_file)
}
  if(!(is.null(input$solution_models_file)|input$solution_models_file=="")){
    data_directory<-gsub("/code","/data",getwd())
    if(!input$solution_models_file=="load"){
      validate(if(!dir.exists(data_directory)){return("Error: data directory does not exist")}else{NULL})
}

validate(if(!file.exists(paste0(data_directory,"/",input$solution_models_file))){return("Error: Solution model file does not exist")}else{NULL})
    selectizeInput('use_sol_models', 'Solution models',
solution_models_available_r(),selected=use_sol_models_r$data,multiple=TRUE)
  }
}
})
  # Dyanmically use abundance phases to select which phases to show
  output$select_abundance_phases <- renderUI({
    selectizeInput('show_abundance_phases', 'Show phases',
c(abundance_phases_available_r(),"Reactive Subsystem","Extract Subsystem","Cumulative Extract Subsystem","Full System"),multiple=TRUE,selected="Reactive Subsystem")
  })
  abundance_phases_available_r<- reactive({
    if(!is.null(input$axis)){
      rownames(switch(input$axis,
x=phase_abundance(store_r$crust_r,input$axis,as.numeric(input$path_y),input$start_x,input$end_x,input$path_label,input_pt = store_r$input_pt_r),
y=phase_abundance(store_r$crust_r,input$axis,as.numeric(input$path_x),input$start_y,input$end_y,input$path_label,input_pt = store_r$input_pt_r)

```

```

) [[2]])}
})
#Create Compilation button
observeEvent(input$create_compilation, {
#Compile PAM fields (corelation to other PAMs)
validate(need(!input$PAM_compilation=="", "Please provide a comma seperated list
of working files in PAM compilation"))
cat("Creating Compilation Legend for", input$PAM_compilation, "\n")
flush.console()
#mergers must contain the current file
mergers<-
sort(union(unlist(strsplit(input$PAM_compilation, ",")), input$working_file))

if(!dir.exists(paste0(input$projects_directory, "/Compile"))){dir.create(paste0(input$projects_directory, "/Compile"))}
compile_names<-NULL
for(i in mergers){

attach(paste0(input$projects_directory, "/", i, "/", i, ".RData"), warn.conflicts=FALSE)
pull_crust<-
get("crust", which(search()==paste0("file:", input$projects_directory, "/", i, "/", i, ".RData"))))

detach(pos=which(search()==paste0("file:", input$projects_directory, "/", i, "/", i, ".RData")))
compile_names<-
union(compile_names, get_PAM_names(neaten_crust(pull_crust, input$phase_aliases), input$PAM_system) [[1]])
}

write.table(compile_names, paste0(input$projects_directory, "/Compile/", input$PAM_compilation, " compilation legend.txt"), sep="\t", quote=F, col.names=FALSE)
cat("Compilation successfully created for", input$PAM_compilation, "\n")
flush.console()
})
#Save Data button
observeEvent(input$save_data, {
switch(input$output_type,
"Data File"=
reactive_message$data <-
write_data_file(data_file(crust_out(), x_n=length(crust_out() [[1]]), y_n=length(crust_out()), input$choose_columns, input$choose_rows, input$choose_points), input$working_file, input$projects_directory, input$file_type),
"Grid"=
if(TRUE){reactive_message$data <- "Saving"
reactive_message$data <- draw_Grid_r()},
"Phase Abundance Along Path"=
if(TRUE){reactive_message$data <- "Saving"
reactive_message$data <- draw_abundance_r()},
"PAM"=
if(TRUE){reactive_message$data <- "Saving"
reactive_message$data <- draw_PAM_r()}
)

})
#GCDkit button
observeEvent(input$send_gcdkit, {
#mod-tag: perform maintenance on this functionality
library(GCDkit)
setwd("D:\\Rcrust\\code")
Rcrust()
eqq<-
data_file(crust, x_n=length(crust[[1]]), y_n=length(crust), choose_columns=NULL, choose_rows=NULL, compile=c("y_i", "x_i"))
eqq_mat<-eqq[[1]]
bb<-apply(eqq_mat, FUN=as.numeric, MARGIN=1)
accessVar("bb", GUI=FALSE)
#needs data frame with samples as lines and info as columns

```

```

    #accessVar ("eqq", GUI=FALSE)
  })
  solution_models_available_r <- reactive({
    solution_models_available<-NULL
    data_directory<-gsub("/code", "/data", getwd())
    qq<-
scan(file=paste0(data_directory, "/", input$solution_models_file), what="character", se
p="\n", quiet=TRUE)
    foo<-grep("begin_model", qq)
    for(i in 1:length(foo)){
      if(i==length(foo)){
        toto<-substr(qq[foo[i]:length(qq)], 1, 1)
      }else{
        toto<-substr(qq[foo[i]:foo[i+1]], 1, 1)
      }
      modrow<-grep("[A-Za-z]", toto) [2]
      line<-qq[foo[i]+modrow-1]
      solution_models_available<-c(solution_models_available, strsplit(line, "
") [[1]] [1])
    }
    return(sort(solution_models_available))
  })
  grid_data_r <- reactive({
    if(input$Grid_variable=="Custom"){
      grid_in<-eval(parse(text=input$Custom_selection))
    }else{
      grid_in<-
grid_data(input$Grid_variable, input$Grid_variable_phase, crust_out(), input_pt)
    }
    #Remove values
    if(input$remove_values!=""){
      for(val in unlist(strsplit(input$remove_values, split=",|;"))){
        for(i in 1:length(grid_in[[2]])){
          if(!is.na(grid_in[[2]][i])){
            if(length(unlist(strsplit(val, split="-"))>1){
              if(grid_in[[2]][i]>=unlist(strsplit(val, split="-
")) [1]&grid_in[[2]][i]<=unlist(strsplit(val, split="-")) [2]){grid_in[[2]][i]<-NA
            }else{
              if(grid_in[[2]][i]==val){grid_in[[2]][i]<-NA}}}
          }
        }
      }
    }
    #apply matrix rotations
    if(input$rotation!=0){
      for(i in 1:input$rotation){
        grid_in[[2]]<-rotate(grid_in[[2]])
      }
    }
    #apply matrix reflections
    if(any(input$reflection=="Horizontal")){grid_in[[2]]<-flip_y(grid_in[[2]])}
    if(any(input$reflection=="Vertical")){grid_in[[2]]<-flip_x(grid_in[[2]])}
    validate(need(nrow(grid_in[[2]])>1, "Cannot plot a grid with only 1 row")
    validate(need(ncol(grid_in[[2]])>1, "Cannot plot a grid with only 1
column"))
    if(any(input$Grid_axes=="bottom")){bottom<-
create_Axes(input$Grid_bottom_axis, input$Grid_bottom_axis_grid_phase, input$Grid_bot
tom_axis_increments, "x", crust_out(), input_pt)}else{bottom<-NULL}
    if(any(input$Grid_axes=="left")){left<-
create_Axes(input$Grid_left_axis, input$Grid_left_axis_grid_phase, input$Grid_left_ax
is_increments, "y", crust_out(), input_pt)}else{left<-NULL}
    if(any(input$Grid_axes=="top")){top<-
create_Axes(input$Grid_top_axis, input$Grid_top_axis_grid_phase, input$Grid_top_axis_
increments, "x", crust_out(), input_pt, y_n)}else{top<-NULL}
    if(any(input$Grid_axes=="right")){right<-
create_Axes(input$Grid_right_axis, input$Grid_right_axis_grid_phase, input$Grid_right
_axis_increments, "y", crust_out(), input_pt, x_n)}else{right<-NULL}
    #mod-tag: look into rotations
    #Apply rotations
    #if(any(input$Grid_axes=="bottom")){colnames(grid_in[[2]])<-
paste0(create_Axes(input$Grid_bottom_axis, input$Grid_bottom_axis_grid_phase, length(

```

```

crust[[1]]+1,"x",crust_out(),input_pt)[[2]][-
(length(crust[[1]]+1)],"_{",1:length(crust[[1]]),";1}")}
#if(any(input$Grid_axes=="left")){rownames(grid_in[[2]])<-
paste0(create_Axes(input$Grid_left_axis,input$Grid_left_axis_grid_phase,length(crus
t)+1,"y",crust_out(),input_pt)[[2]][-
(length(crust)+1)],"_{1;",1:length(crust),"}")
return(list(grid_in[[1]],grid_in[[2]],bottom,left,top,right))
})
phase_abundance_r<-reactive({
if(!is.null(input$axis)){
switch(input$axis,

x=phase_abundance(store_r$crust_r,input$axis,as.numeric(input$path_y),input$start_x
,input$end_x,input$path_label,input_pt = store_r$input_pt_r),

y=phase_abundance(store_r$crust_r,input$axis,as.numeric(input$path_x),input$start_y
,input$end_y,input$path_label,input_pt = store_r$input_pt_r)
)
}
})

PAM_r<-reactive({

PAM_calc(crust_out(),input$PAM_system,input$compile_PAM,input$PAM_compilation)
})
neaten_crust<-function(crust_neat=NULL,phase_aliases=NULL){
#rename using aliases if given
if(!phase_aliases==""){
split_aliases<-strsplit(strsplit(phase_aliases,c(", "))[[1]],"=")
phase_aliases<-NULL
phase_aliases_names<-NULL
for(i in 1:length(split_aliases)){
phase_aliases<-c(phase_aliases,split_aliases[[i]][1])
phase_aliases_names<-c(phase_aliases_names,split_aliases[[i]][2])
}
names(phase_aliases)<-phase_aliases_names
#seperate merge commands
merge_aliases<-grep("&",phase_aliases,value=TRUE)
if(length(merge_aliases)!=0){
phase_aliases<-phase_aliases[-grep("&",phase_aliases)]
}
merge_aliases<-strsplit(merge_aliases,"&")
if(length(merge_aliases)==0){merge_aliases<-""}
if(length(phase_aliases)==0){phase_aliases<-""}
if(all(phase_aliases!="")|all(merge_aliases!="")){
for(x_i in 1:length(crust[[1]])){
for(y_i in 1:length(crust)){
delete<-NULL
grab<-rownames(crust_neat[[y_i]][[x_i]])
if(length(grab)>0){
grab_split<-strsplit(grab,"_")
grab_sys<-grab_name<-grab
for(ph in 1:length(grab)){
grab_name[ph]<-grab_split[[ph]][1]
#if first part of system is numeric extract up to end of _
if(!is.na(suppressWarnings(as.numeric(grab_split[[ph]][-1][1])))){
grab_sys[ph]<-paste(grab_split[[ph]][c(-1,-2)],collapse="_")
}else{
grab_sys[ph]<-paste(grab_split[[ph]][-1],collapse="_")
}
}
}
#rename using aliases
if(all(phase_aliases!="")){
for(ph in 1:length(grab)){
if(any(grab_name[ph]==phase_aliases)){
#Rename feldspars
if(names(which(grab_name[ph]==phase_aliases))=="Pl|Kf"){
if(length(intersect(major_elements,c("CAO","K2O")))==2){

```

```

# fix-tag: currently rename feldspars in two places, in order ot allow
phase extraction, simplify this
if(crust_neat[[y_i]][[x_i]][ph,"CAO"]>crust_neat[[y_i]][[x_i]][ph,"K2O"]){
  grab_name[ph]<-"Pl"
}else{grab_name[ph]<-"Kf"}
}
}else{
#Rename phases using aliases
grab_name[ph]<-names(which(grab_name[ph]==phase_aliases)[1])
}
#label unwanted phases (phases of zero mass or phases labelled as "hide")
if(crust_neat[[y_i]][[x_i]][ph,"mass"]==0|grab_name[ph]=="hide"){
  delete<-c(delete,ph)
}
}
}
#number duplicates within systems
renamed<-paste(grab_name,grab_sys,sep="_")
num<-1
while(any(duplicated(renamed))){
  renamed[which(duplicated(renamed))]<-
paste(grab_name[which(duplicated(renamed))],num,grab_sys[which(duplicated(renamed))
],sep="_")
  num<-num+1
}
rownames(crust_neat[[y_i]][[x_i]])<-renamed
}
#merge if required
if(all(merge_aliases!="")){
  systems<-c("rs","es","cumul")
  for(sys in systems){
    for(merge_try in 1:length(merge_aliases)){
      merge_nos<-NULL
      for(i in 1:length(merge_aliases[[merge_try]])){
if(any(paste(grab_name,grab_sys,sep="_")==paste(merge_aliases[[merge_try]][i],sys,s
ep="_"))){
      merge_nos<-union(merge_nos,which(grab_name==merge_aliases[[merge_try]][i]))
}
}
merger<-NULL
if(!is.null(merge_nos)){
  for(i in merge_nos){
    merger<-rbind(merger,crust_neat[[y_i]][[x_i]][i,,drop=FALSE])
  }
merged<-wtd.add(merger,avname=names(merge_aliases)[merge_try])
crust_neat[[y_i]][[x_i]][merge_nos[1],]<-merged
rownames(crust_neat[[y_i]][[x_i]][merge_nos[1]]<-
paste(names(merge_aliases)[merge_try],sys,sep="_")
  if(length(merge_nos)>1){
    delete<-c(delete,merge_nos[-1])
  }
}
}
}
}
#remove unwanted phases
if(!is.null(delete)){
  crust_neat[[y_i]][[x_i]]<-crust_neat[[y_i]][[x_i]][-delete,]
}
}
}
}
}
return(crust_neat)
}
crust_out<-reactive({

```

```

    return(neaten_crust(store_r$crust_r,input$phase_aliases))
  })
  create_Axes<-
function(axes_variable,axes_variable_phase,axes_increments,axes_direction,crust=NULL,
input_pt=NULL,axes_line=1){
  #Given axes_variable,axes_variable_phase,axes_increments,axes_direction create
axis values
  if(axes_increments=="Increments"){axes_increments<-11}else{axes_increments<-
as.numeric(axes_increments)}
  if(axes_direction=="y"){
    i_n<-length(crust)/(axes_increments-1)*(0:(axes_increments-1))+1
  }else{i_n<-length(crust[[1]])/(axes_increments-1)*(0:(axes_increments-1))+1}
  if(axes_variable=="y_i"|axes_variable=="x_i"){
    axis_values<-i_n[-length(i_n)]
  }else{
    axis_values<-NULL
    for(i in i_n[-length(i_n)]){
      if(axes_direction=="y"){
        if(axes_variable=="Temperature"|axes_variable=="Pressure"){
          axis_values<-c(axis_values,input_pt[[i]][[axes_line]][,axes_variable])
        }else{
          chk<-
try(crust[[i]][[axes_line]][axes_variable_phase,axes_variable],silent=TRUE)
          if(class(chk)=="try-error"){
            axis_values<-c(axis_values,0)}else{
            axis_values<-
c(axis_values,crust[[i]][[axes_line]][axes_variable_phase,axes_variable])
          }
        }else{
          if(axes_variable=="Temperature"|axes_variable=="Pressure"){
            axis_values<-c(axis_values,input_pt[[axes_line]][[i]][,axes_variable])
          }else{
            chk<-
try(crust[[axes_line]][[i]][axes_variable_phase,axes_variable],silent=TRUE)
            if(class(chk)=="try-error"){
              axis_values<-c(axis_values,0)}else{
              axis_values<-
c(axis_values,crust[[axes_line]][[i]][axes_variable_phase,axes_variable])
            }
          }
        }
      }
      axis_values<-
c(axis_values,axis_values[length(axis_values)]+axis_values[length(axis_values)]-
axis_values[length(axis_values)-1])

if(axes_variable=="y_i"|axes_variable=="x_i"|axes_variable=="Temperature"|axes_vari
able=="Pressure"){
  axis_title<-axes_variable
}else{
  axis_title<-paste(axes_variable_phase,axes_variable)
}
return(list(axis_title,signif(axis_values,digits=4)))
}
draw_Grid_r<-reactive({
  library(graphics)
  library(grDevices)
  validate(need(nrow(grid_data_r() [[2]])>1,"Cannot plot a grid with only 1
row"))
  validate(need(ncol(grid_data_r() [[2]])>1,"Cannot plot a grid with only 1
column"))
  bottom<-grid_data_r() [[3]]
  left<-grid_data_r() [[4]]
  top<-grid_data_r() [[5]]
  right<-grid_data_r() [[6]]
  action<- "View"

```

```

if(!is.null(reactive_message$data)) {if(reactive_message$data=="Saving") {action<-
"Save"}}
  if(action=="Save") {
    outfile_path<-
paste0(input$projects_directory,"/",input$working_file,"/Outputs/",input$working_file,
"_grid_",gsub("\\\\/", " per ",grid_data_r()[[1]]),input$file_type)
    if(input$file_type==".txt"){
      write.table(grid_data_r()[[2]],outfile_path,sep="\t",quote=F)
    }
    if(input$file_type==".csv"){
      write_test<-try(write.csv(grid_data_r()[[2]],outfile_path),silent=TRUE)
      if(class(write_test=="try-error")){
        cat("Error cannot write to ",outfile_path,", please close all programs that
may be accessing the file then try again\n")
        return(paste0("Error could not save Grid File: ",grid_data_r()[[1]],", file
may be open in another program, please close all programs that may be accessing the
file then try again\n"))
      }
    }
    if(input$file_type==".ps"){
      outfile_path<-gsub("%","%",outfile_path)
      postscript(file=outfile_path,onefile=TRUE,horizontal=TRUE)
    }
    if(action=="View"|(action=="Save"&input$file_type==".ps")){
      filled.contour(t(flip_y(grid_data_r()[[2]])),
      plot.axes={
        if(!is.null(bottom)){axis(1,(0:(length(bottom[[2]])-
1))/(length(bottom[[2]])-1),bottom[[2]})else{NULL};
        if(!is.null(left)){axis(2,(0:(length(left[[2]])-1))/(length(left[[2]])-
1),left[[2]})else{NULL};
        if(!is.null(top)){axis(3,(0:(length(top[[2]])-1))/(length(top[[2]])-
1),top[[2]})else{NULL};
        if(!is.null(right)){axis(4,(0:(length(right[[2]])-1))/(length(right[[2]])-
1),right[[2]})else{NULL}},
      color.palette=eval(parse(text=input$Grid_colours))
      #Colour options =
gray.colors,heat.colours,terrain.colours,rainbow,topo.colours
      if(!is.null(bottom)){mtext(bottom[[1]],1,line=3)}
      if(!is.null(left)){mtext(left[[1]],2,line=3)}
      if(!is.null(top)){mtext(top[[1]],3,line=3)}
      if(!is.null(right)){mtext(right[[1]],4,line=3)}
      if(action=="Save"){
        title(paste("Grid of ",gsub("\\\\/", " per ",grid_data_r()[[1]])," for
",input$working_file))
        dev.off()
      }
      detach("package:graphics")
      detach("package:grDevices")
      if(action=="Save"){
        cat("File written to ",outfile_path,"\n")
        return(paste0(paste("Grid of",gsub("\\\\/", " per ",grid_data_r()[[1]])),
saved to ",input$projects_directory,"/",input$working_file,"/Outputs/\n"))
      }
    }
  })
draw_abundance_r<-reactive({
  #Plot phase abundance versus path cumulative column graph
  library(graphics)
  library(RColorBrewer)
  library(grDevices)
  #All RColorBrewer palettes display.brewer.all()
  #Display a specific palette display.brewer.pal(12,"Set3")
  action<-"View"

if(!is.null(reactive_message$data)) {if(reactive_message$data=="Saving") {action<-
"Save"}}

```

```

        if(action=="Save"){
            outfile_path<-
paste0(input$projects_directory,"/",input$working_file,"/Outputs/",input$working_file,
"phase_abundance_",gsub("\\\\/", " per
",phase_abundance_r()[[1]]),input$file_type)
            if(input$file_type==".txt"){

write_phase_abundance(phase_abundance_r(),input$working_file,input$projects_directory,input$file_type)
            }
            if(input$file_type==".csv"){
                write_test<-
try(write_phase_abundance(phase_abundance_r(),input$working_file,input$projects_directory,input$file_type),silent=TRUE)
                if(class(write_test)=="try-error"){
                    cat("Error cannot write file please close all programs that may be
accessing the file then try again\n")
                    return(paste0("Error could not save Phase abundance File: file may be open
in another program, please close all programs that may be accessing the file then
try again\n"))
                }
            }
            if(input$file_type==".ps"){
                cat("Error Functionality not available yet\n")
                return(paste0("Error Functionality not available yet\n"))
                # mod-tag: look into saving phase abundance directly to .ps
                #outfile_path<-gsub("%","%",outfile_path)
                #postscript(file=outfile_path,onefile=TRUE,horizontal=TRUE)
            }
            if(action=="View"|(action=="Save"&input$file_type==".ps")){
                phase_abundance_data<-phase_abundance_r()[[2]]
                #subset phases
                if(!is.null(input$show_abundance_phases)){
                    phases_to_show<-input$show_abundance_phases
                    phases_present<-rownames(phase_abundance_data)
                    if(any(phases_to_show=="Reactive Subsystem")){
                        phases_to_show<-
union(phases_to_show,phases_present[setdiff(which(substrRight(phases_present,3)=="_
rs"),which(phases_present=="Bulk_rs"))])
                        phases_to_show<-phases_to_show[-grep("Reactive Subsystem",phases_to_show)]
                    }
                    if(any(phases_to_show=="Extract Subsystem")){
                        phases_to_show<-
union(phases_to_show,phases_present[setdiff(which(substrRight(phases_present,3)=="_
es"),which(phases_present=="Bulk_es"))])
                        phases_to_show<-phases_to_show[-grep("Extract Subsystem",phases_to_show)]
                    }
                    if(any(phases_to_show=="Cumulative Extract Subsystem")){
                        phases_to_show<-
union(phases_to_show,phases_present[setdiff(which(substrRight(phases_present,9)=="_
es_cumul"),which(phases_present=="Bulk_es_cumul"))])
                        phases_to_show<-phases_to_show[-grep("Cumulative Extract
Subsystem",phases_to_show)]
                    }
                    if(any(phases_to_show=="Full System")){
                        phases_to_show<-union(phases_to_show,phases_present)
                        phases_to_show<-phases_to_show[-grep("Full System",phases_to_show)]
                    }
                }
                phase_abundance_data<-
phase_abundance_data[match(phases_to_show,rownames(phase_abundance_data)),,drop=FALSE]
            }
            #Normalise to percentage
            phase_abundance_data<-apply(phase_abundance_data, 2,
function(x){x*100/sum(x,na.rm=T)})
            #Set colour palette
            cols<-brewer.pal(min(nrow(phase_abundance_data),12), "Set3")

```

```

#Plot

barplot(phase_abundance_data,space=0,col=cols,border=NA,legend.text=(input$legend!=
"None"), args.legend=list(x =input$legend,bty =
"n"),xlab=input$path_label,ylab="Phase abundance (wt.%)")
  if(action=="Save"){
    title(paste("Grid of ",gsub("\\\\/", " per ",grid_data_r()[[1]])," for
",input$working_file))
    dev.off()
  }
  detach("package:graphics")
  detach("package:grDevices")
  if(action=="Save"){
    cat("File written to ",outfile_path,"\n")
    return(paste0(paste("Grid of",gsub("\\\\/", " per ",grid_data_r()[[1]])),",
saved to ",input$projects_directory,"/",input$working_file,"/Outputs/\n"))
  }
})

draw_PAM_r<-reactive({
#Bind all_pres to PAM legend
PAM_legend<-c(PAM_r()[[3]],names(PAM_r()[[2]]))
names(PAM_legend)<-c("\All fields are +\\"",as.numeric(PAM_r()[[2]]))
compiled_legend<-c(PAM_r()[[3]],names(PAM_r()[[6]]))
names(compiled_legend)<-c("\All fields are +\\"",as.numeric(PAM_r()[[6]]))
action<-"View"

if(!is.null(reactive_message$data)){if(reactive_message$data=="Saving"){action<-
"Save"}}
  if(action=="Save"){
    outfile_path<-
paste0(input$projects_directory,"/",input$working_file,"/Outputs/",input$working_fi
le," PAM",input$file_type)
    outfile_legend_path<-
paste0(input$projects_directory,"/",input$working_file,"/Outputs/",input$working_fi
le," PAM legend",input$file_type)
    outfile_compile_legend_path<-
paste0(input$projects_directory,"/Compile/",paste(sort(union(unlist(strsplit(input$
PAM_compilation,"")),input$working_file)),collapse=",")," compile
legend",input$file_type)
    if(input$file_type=="txt"){
      write.table(PAM_r()[[1]],outfile_path,sep="\t",quote=F,row.names = TRUE)
      write.table(PAM_legend,outfile_legend_path,sep="\t",quote=F,row.names =
TRUE,col.names = FALSE)
    }
    if(input$compile_PAM){write.table(compiled_legend,outfile_compile_legend_path,sep="
\t",quote=F,row.names = TRUE,col.names = FALSE)}
  }
  if(input$file_type=="csv"){
    write_test<-try(write.csv(PAM_r()[[1]],outfile_path,row.names =
TRUE),silent=TRUE)
    if(class(write_test)=="try-error"){
      cat("Error cannot write to ",outfile_path,", please close all programs that
may be accessing the file then try again\n")
      return(paste0("Error could not save phase assemblage map: ",outfile_path,",
file may be open in another program, please close all programs that may be
accessing the file then try again\n"))
    }
    write_test<-try(write.csv(PAM_legend,outfile_legend_path,row.names =
TRUE,col.names = FALSE),silent=TRUE)
    if(class(write_test)=="try-error"){
      cat("Error cannot write to ",outfile_legend_path,", please close all programs
that may be accessing the file then try again\n")
      return(paste0("Error could not save phase assemblage map legend:
",outfile_legend_path,", file may be open in another program, please close all
programs that may be accessing the file then try again\n"))
    }
  }
}

```

```

        if(input$compile_PAM){write_test<-
try(write.csv(compiled_legend,outfile_compile_legend_path,row.names =
TRUE,col.names = FALSE),silent=TRUE)
    if(class(write_test)=="try-error"){
        cat("Error cannot write to ",outfile_compile_legend_path,", please close all
programs that may be accessing the file then try again\n")
        return(paste0("Error could not save phase assemblage map legend:
",outfile_compile_legend_path,", file may be open in another program, please close
all programs that may be accessing the file then try again\n"))
    }
}

    if(input$file_type=="ps"){
        library(grDevices)
        postscript(file=outfile_path,onefile=TRUE,horizontal=TRUE)
    }
}

    if(action=="View"|(action=="Save"&input$file_type=="ps")){
        library(graphics)
        par(mar=c(5,4,4,5)+.1)
        library(grDevices)
        library(raster)
        library(rgeos)
        max_pol<-max(length(PAM_r()[[2]]),length(PAM_r()[[6]])
raster::plot(PAM_r()[[4]],col=gray(1:max_pol/max_pol)[PAM_r()[[5]]])
        if(input$PAM_labels=="Numbers"){
            raster::text(gCentroid(PAM_r()[[4]]),byid = TRUE),labels=PAM_r()[[5]],col =
"Black")
        }else{
            if(input$compile_PAM){
                raster::text(gCentroid(PAM_r()[[4]]),byid =
TRUE),labels=compiled_legend[PAM_r()[[5]]],col = "Black")
            }else{
                raster::text(gCentroid(PAM_r()[[4]]),byid =
TRUE),labels=PAM_legend[PAM_r()[[5]]],col = "Black")
            }
        }

        # Create axes
        if(length(input$PAM_axes)>0){
            for(j in 1:length(input$PAM_axes)){
                #get variable and increment choice
                var_choice<-
eval(parse(text=paste("input$PAM_",input$PAM_axes[j],"_axis",sep="")))

                if(!(var_choice=="y_i"|var_choice=="x_i"|var_choice=="Temperature"|var_choice=="Pre
ssure")){
                    grid_phase_choice<-
eval(parse(text=paste("input$PAM_",input$PAM_axes[j],"_axis_grid_phase",sep="")))
                }
                increment_choice<-
eval(parse(text=paste("input$PAM_",input$PAM_axes[j],"_axis_increments",sep="")))
                if(increment_choice=="Increments"){increment_choice<-
11}else{increment_choice<-as.numeric(increment_choice)}
                if(input$PAM_axes[j]=="left"|input$PAM_axes[j]=="right"){
                    i_n<-length(crust)/(increment_choice-1)*(0:(increment_choice-1))+1
                }else{i_n<-length(crust[[1]])/(increment_choice-1)*(0:(increment_choice-1))+1}
                if(var_choice=="y_i"|var_choice=="x_i"){
                    axis_values<-i_n[-length(i_n)]
                }else{
                    axis_values<-NULL
                }
                for(i in i_n[-length(i_n)]){
                    if(input$PAM_axes[j]=="left"|input$PAM_axes[j]=="right"){
                        if(var_choice=="Temperature"|var_choice=="Pressure"){
                            axis_values<-c(axis_values,input_pt[[i]][[1]][,var_choice])
                        }else{
                            chk<-try(crust[[i]][[1]][grid_phase_choice,var_choice],silent=TRUE)
                            if(class(chk)=="try-error"){
                                axis_values<-c(axis_values,0)}else{

```

```

axis_values<-c(axis_values,crust[[i]][[1]][grid_phase_choice,var_choice])
}
}else{
if(var_choice=="Temperature"|var_choice=="Pressure"){
axis_values<-c(axis_values,input_pt[[1]][[i]][,var_choice])
}else{
chk<-try(crust[[1]][[i]][grid_phase_choice,var_choice],silent=TRUE)
if(class(chk=="try-error")){
axis_values<-c(axis_values,0)}else{
axis_values<-c(axis_values,crust[[1]][[i]][grid_phase_choice,var_choice])
}
}
}
}
}
axis_values<-
c(axis_values,axis_values[length(axis_values)]+axis_values[length(axis_values)]-
axis_values[length(axis_values)-1])
#round to 2 significant figures
axis_values<-signif(axis_values,digits=4)
side_no<-switch(input$PAM_axes[j],"bottom"=1,"left"=2,"top"=3,"right"=4)
axis(side_no,(0:(increment_choice-1))/(increment_choice-1),axis_values)

if(var_choice=="y_i"|var_choice=="x_i"|var_choice=="Temperature"|var_choice=="Pressure"){
mtext(var_choice, side=side_no, line=3)
}else{
mtext(paste(grid_phase_choice,var_choice), side=side_no, line=3)
}
}
}
if(!PAM_r()[[3]]==""){
title(paste("All fields are +",PAM_r()[[3]]))
}
#Add contour
if(!input$PAM_contour=="None"){
grid_out_mat<-matrix(0,y_n,x_n)
if(input$PAM_contour=="Pressure"|input$PAM_contour=="Temperature"){
for(x_i in 1:x_n){
for(y_i in 1:y_n){
grid_out_mat[y_i,x_i]<-input_pt[[y_i]][[x_i]][,input$PAM_contour]
}
}
}else{
for(x_i in 1:x_n){
for(y_i in 1:y_n){
chk<-
try(crust_out()[[y_i]][[x_i]][input$PAM_contour_grid_phase,input$PAM_contour],silent=TRUE)
if(class(chk=="try-error")){ grid_out_mat[y_i,x_i]<-0}else{
grid_out_mat[y_i,x_i]<-chk
}
}
}
}
x<-raster::raster(flip_y(grid_out_mat))
if(input$PAM_contour_increments=="Default
Increments"){raster::contour(x,add=TRUE,col="red")}else{

if(input$PAM_contour_increments=="In/Out"){raster::contour(x,add=TRUE,levels=0.0000
0000000000000000001,col="red")}else{

raster::contour(x,add=TRUE,nlevels=as.numeric(input$PAM_contour_increments),col="red")
}
}
}
if(action=="Save"){

```

```

title(paste("Phase Assemblage Map for",input$working_file),)
dev.off()
write.table(PAM_legend,outfile_legend_path,sep="\t",quote=F,row.names =
TRUE,col.names = FALSE)

if(input$compile_PAM){write.table(compiled_legend,outfile_compile_legend_path,sep="
\t",quote=F,row.names = TRUE,col.names = FALSE)
}
detach("package:graphics")
detach("package:rgeos")
detach("package:raster")
detach("package:sp")
detach("package:grDevices")
}
if(action=="Save"){
cat("File written to ",outfile_path,"\n")
cat("File written to ",outfile_legend_path,"\n")
return(paste0("Phase Assemblage Map saved to
",input$projects_directory,"/",input$working_file,"/Outputs/\n"))
}
})
output$output_header<-renderText(
switch(input$output_type,
"Data File"=if(!is.null(store_r$crust_r)){"Compilation data file"},
"Grid"=if(!is.null(grid_data_r())){paste0(grid_data_r()[[1]]," on
(X,Y) grid"}),
"Phase Abundance Along
Path"=if(!is.null(phase_abundance_r())){phase_abundance_r()[[1]]},
"PAM"=if(!is.null(store_r$crust_r)){"Phase Assemblage Map"}
))
# Dynamically create output for viewing
output$output_view <- renderUI({
if(!is.null(input$output_form)){
switch(input$output_form,
"Data"=tableOutput("table"),
"Legend"=tableOutput("table"),
switch(input$output_type,
"PAM"=plotOutput("plot"),
"Grid"=plotOutput("plot"),
"Phase Abundance Along Path"=plotOutput("plot")
)
)
}
})
output$table <- renderTable(
switch(input$output_type,
"Data
File"=data_file(crust_out(),x_n=length(crust_out()[[1]]),y_n=length(crust_out()),in
put$choose_columns,input$choose_rows,input$choose_points),
"Grid"=if(!is.null(grid_data_r())){grid_data_r()[[2]]},
"Phase Abundance Along Path"= phase_abundance_r()[[2]],

"PAM"=matrix(names(PAM_r()[[2]]),,1,byrow=FALSE,dimnames=list(PAM_r()[[2]],paste("A
ll fields are +",PAM_r()[[3]])))
),rownames=TRUE
)
)
output$plot <- renderPlot(
switch(input$output_type,
"PAM"=draw_PAM_r(),
"Grid"=draw_Grid_r(),
"Phase Abundance Along Path"=draw_abundance_r()
)
)
# Dynamically create output form selections
output$output_form_selection <- renderUI({
form_selection<-switch(input$output_type,
"Data File"="Data",
"PAM"=c("Legend","Plot"),

```

```

        c("Data", "Plot")
    )
    radioButtons('output_form', "View", form_selection, inline=TRUE)
  })
  # Dynamically create output selections
  output$output_selection <- renderUI({
    # Refresh reactive outputs if they exist
    if(exists("crust")){store_r$crust_r<-crust}
    if(exists("input_pt")){store_r$input_pt_r<-input_pt}
    if(exists("input_bulk")){store_r$input_bulk_r<-input_bulk}
    if(exists("major_elements")){store_r$major_elements_r<-major_elements}
    if(is.null(store_r$crust_r))
    return("To select outputs: first run calculation or load previously saved
calculation
        *(variables 'crust' and 'major_elements' must be present)")
    all_columns<-NULL
    all_phases<-NULL
    for(y_i in 1:length(crust_out())){
      for(x_i in 1:length(crust_out()[[1]])){
        all_columns<-union(all_columns, colnames(crust_out()[[y_i]][[x_i]]))
        all_phases<-union(all_phases, rownames(crust_out()[[y_i]][[x_i]]))
      }
    }
    #mod-tag: figure out better way of parsing multiple panels
    conditionalPanel("true",
      conditionalPanel("input.output_type == 'Data File'",
        selectizeInput('choose_columns', 'Select Columns',
          c("All"=="", "Brief", "ID", "Phase", "y_i", "x_i", "Pressure(kbar)", "Temperature(C)", "wt%",
            "vol%", major_elements, "mass", "V(J/bar)", "H(J)", "Gruneisen_T", "Ks(bar)", "Mu(bar)", "
            V0(km/s)", "Vp(km/s)", "Vs(km/s)", "Vp/Vs", "Rho(kg/m3)", "Cp(J/K)", "alpha(1/K)", "beta(1
            /bar)", "S(J/K)", "N(g)", "Cp/Cv"), multiple=TRUE),
        selectizeInput('choose_rows', 'Select System/Phase', c("All"=="", "Reactive
        subsystem", "Extract subsystem", all_phases), multiple=TRUE),
        textInput('choose_points', 'Select Points', "{1;1}")
      ),
      conditionalPanel("input.output_type == 'Grid'",
        selectInput('Grid_variable', 'Variable',
          c("Pressure", "Temperature", all_columns, "Custom"),
          selected="Pressure", selectize=TRUE),
        conditionalPanel(condition= "input.Grid_variable.indexOf('Custom') != -1",
          textInput('Custom_selection', "Custom Selection")
        ),
      conditionalPanel("[ 'Pressure', 'Temperature', 'Custom' ].indexOf(input.Grid_variable)
      == -1",
        selectInput('Grid_variable_phase', NULL, sort(all_phases),
          selected="Bulk_rs", selectize=TRUE),
        # mod-tag: allow increments setting
        #selectizeInput('Grid_variable_increments', NULL,
        c("Increments", 3:y_n), "Increments") ,
        selectInput('Grid_axes', 'Labelled axes', c("bottom", "left", "top", "right"),
          selected=c("bottom", "left"), selectize=TRUE, multiple=TRUE),
        conditionalPanel(condition=
          "input.Grid_axes.indexOf('bottom') != -1",
          selectInput('Grid_bottom_axis', 'Bottom Axis',
            c("x_i", "Pressure", "Temperature", colnames(crust[[1]][[1]])),
            selected="x_i", selectize=TRUE),
        conditionalPanel("[ 'x_i', 'Pressure', 'Temperature' ].indexOf(input.Grid_bottom_axis)
        == -1",
          selectInput('Grid_bottom_axis_grid_phase', NULL,
            sort(all_phases), selected="Bulk_rs", selectize=TRUE),
          selectizeInput('Grid_bottom_axis_increments', NULL,
            c("Increments", 3:x_n), "Increments")
        ),
        conditionalPanel(condition=
          "input.Grid_axes.indexOf('left') != -1",

```

```

        selectInput('Grid_left_axis', 'Left Axis',
c("y_i","Pressure","Temperature",colnames(crust[[1]][[1]])),
selected="y_i",selectize=TRUE),

conditionalPanel("[ 'y_i', 'Pressure', 'Temperature' ].indexOf(input.Grid_left_axis) ==
-1",
        selectInput('Grid_left_axis_grid_phase', NULL,
sort(all_phases), selected="Bulk_rs",selectize=TRUE)),
        selectizeInput('Grid_left_axis_increments', NULL,
c("Increments",3:y_n),"Increments")
    ),
        conditionalPanel(condition=
.Grid_axes.indexOf('top') != -1",
        selectInput('Grid_top_axis', 'Top Axis',
c("x_i","Pressure","Temperature",colnames(crust[[1]][[1]])),
selected="x_i",selectize=TRUE),

conditionalPanel("[ 'x_i', 'Pressure', 'Temperature' ].indexOf(input.Grid_top_axis) ==
-1",
        selectInput('Grid_top_axis_grid_phase', NULL,
sort(all_phases), selected="Bulk_rs",selectize=TRUE)),
        selectizeInput('Grid_top_axis_increments', NULL,
c("Increments",3:x_n),"Increments")
    ),
        conditionalPanel(condition=
.Grid_axes.indexOf('right') != -1",
        selectInput('Grid_right_axis', 'Right Axis',
c("y_i","Pressure","Temperature",colnames(crust[[1]][[1]])),
selected="y_i",selectize=TRUE),

conditionalPanel("[ 'y_i', 'Pressure', 'Temperature' ].indexOf(input.Grid_right_axis)
== -1",
        selectInput('Grid_right_axis_grid_phase', NULL,
sort(all_phases), selected="Bulk_rs",selectize=TRUE)),
        selectizeInput('Grid_right_axis_increments', NULL,
c("Increments",3:y_n),"Increments")
    ),
        textInput('remove_values',"Remove Values"),
        selectInput('Grid_colours', 'Colour Scheme',
c("gray.colors","heat.colors","terrain.colors","rainbow","topo.colors"),
selected="gray.colors",selectize=TRUE),
        radioButtons("rotation","Rotation",c("0"=0,"90"=1,"180"=2,"270"=3), selected
=0, inline = TRUE,width="200px"),
        checkboxGroupInput("reflection","Reflection",c("Horizontal","Vertical"))
    ),
        conditionalPanel("input.output_type == 'Phase Abundance Along Path'",
        selectInput('axis', 'Axis', c("x","y"),
selected="x",selectize=TRUE),
        conditionalPanel("input.axis == 'x'",
            selectInput('path_y', 'Path',
1:length(crust), selected=1,selectize=TRUE),
            selectInput('start_x', 'Start Point',
1:length(crust[[1]]), selected=1,selectize=TRUE),
            selectInput('end_x', 'End Point',
1:length(crust[[1]]), selected=1,selectize=TRUE)
        ),
        conditionalPanel("input.axis == 'y'",
            selectInput('path_x', 'Path',
1:length(crust[[1]]), selected=1,selectize=TRUE),
            selectInput('start_y', 'Start Point',
1:length(crust), selected=1,selectize=TRUE),
            selectInput('end_y', 'End Point',
1:length(crust), selected=1,selectize=TRUE)
        ),
        uiOutput("select_abundance_phases"),
        selectInput('path_label', 'Path Label',
c("Point","Pressure(kbar)","Temperature(C)"), selected="Point",selectize=TRUE),

```

```

        selectInput('legend', 'Legend', c("None","bottomright",
"bottom", "bottomleft", "left", "topleft", "top", "topright", "right","center"),
selected="topright",selectize=TRUE)
    ),
    conditionalPanel("input.output_type == 'PAM'",
        selectInput('PAM_system', 'System', c("Reactive
Subsystem","Extract Subsystem","Full System"), selected="Reactive
Subsystem",selectize=TRUE),
        selectInput('PAM_labels', 'Field Labels',
c("Phases","Numbers"), selected="Phases",selectize=TRUE),
        selectInput('PAM_axes', 'Labelled axes',
c("bottom","left","top","right"),
selected=c("bottom","left"),selectize=TRUE,multiple=TRUE),
        conditionalPanel(condition=
"input.PAM_axes.indexOf('bottom') != -1",
            selectInput('PAM_bottom_axis', 'Bottom Axis',
c("x_i","Pressure","Temperature",colnames(crust[[1]][[1]])),
selected="x_i",selectize=TRUE),

conditionalPanel("[ 'x_i','Pressure','Temperature'].indexOf(input.PAM_bottom_axis)
== -1",
            selectInput('PAM_bottom_axis_grid_phase', NULL,
sort(all_phases), selected="Bulk_rs",selectize=TRUE)),
            selectizeInput('PAM_bottom_axis_increments', NULL,
c("Increments",3:x_n),"Increments")
        ),
        conditionalPanel(condition= "input.PAM_axes.indexOf('left')
!= -1",
            selectInput('PAM_left_axis', 'Left Axis',
c("y_i","Pressure","Temperature",colnames(crust[[1]][[1]])),
selected="y_i",selectize=TRUE),

conditionalPanel("[ 'y_i','Pressure','Temperature'].indexOf(input.PAM_left_axis) ==
-1",
            selectInput('PAM_left_axis_grid_phase', NULL,
sort(all_phases), selected="Bulk_rs",selectize=TRUE)),
            selectizeInput('PAM_left_axis_increments', NULL,
c("Increments",3:y_n),"Increments")
        ),
        conditionalPanel(condition=
"input.PAM_axes.indexOf('top') != -1",
            selectInput('PAM_top_axis', 'Top Axis',
c("x_i","Pressure","Temperature",colnames(crust[[1]][[1]])),
selected="x_i",selectize=TRUE),

conditionalPanel("[ 'x_i','Pressure','Temperature'].indexOf(input.PAM_top_axis) == -
1",
            selectInput('PAM_top_axis_grid_phase', NULL,
sort(all_phases), selected="Bulk_rs",selectize=TRUE)),
            selectizeInput('PAM_top_axis_increments', NULL,
c("Increments",3:x_n),"Increments")
        ),
        conditionalPanel(condition=
"input.PAM_axes.indexOf('right') != -1",
            selectInput('PAM_right_axis', 'Right Axis',
c("y_i","Pressure","Temperature",colnames(crust[[1]][[1]])),
selected="y_i",selectize=TRUE),

conditionalPanel("[ 'y_i','Pressure','Temperature'].indexOf(input.PAM_right_axis) ==
-1",
            selectInput('PAM_right_axis_grid_phase', NULL,
sort(all_phases), selected="Bulk_rs",selectize=TRUE)),
            selectizeInput('PAM_right_axis_increments', NULL,
c("Increments",3:y_n),"Increments")
        ),
        selectInput('PAM_contour', 'Contour',
c("None","Pressure","Temperature",colnames(crust[[1]][[1]])),
selected="None",selectize=TRUE),

```

```

conditionalPanel("[ 'None', 'Pressure', 'Temperature' ].indexOf(input.PAM_contour) == -
1",
                selectInput('PAM_contour_grid_phase', NULL,
sort(all_phases), selected="Bulk_rs",selectize=TRUE)),
                selectizeInput('PAM_contour_increments', NULL, c("Default
Increments", "In/Out", "Set Levels", "Set Contours", 3:y_n), "Default Increments"),
                textInput('PAM_compilation', 'PAM Compilation'),
                actionButton("create_compilation", "Compile/Refresh
Legend"),
                checkboxInput("compile_PAM", "Apply compilation?", value =
FALSE)
        ),
        selectInput('file_type', 'File type', c(".csv", ".txt", ".ps"),
selected=".csv",selectize=TRUE),
        actionButton("save_data", "Save To File")
        # mod-tag: allow this functionality
        #actionButton("send_gcdkit", "Send To GCDkit")
    )
})
#if working_file is not blank on first opening, load file
observe({
  if(!exists("first_load")){first_load<-TRUE}
  if(first_load){
    projects_directory<-input$projects_directory
    working_file<-input$working_file
    #error handling
    reactive_message$data <-error_handling(working_file,projects_directory)
    if(reactive_message$data=="error handling passed"){
      #load

if(file.exists(paste0(projects_directory,"/",working_file,"/Inputs/",working_file,"
.txt"))){
  reactive_message$data <- paste0(on_load())
  }else{
    reactive_message$data <- paste0("No input file found at
",paste0(projects_directory,"/",working_file,"/Inputs/",working_file,".txt"))
  }
  #Load workspace if it exists (previous calculation results)

if(file.exists(paste0(projects_directory,"/",working_file,"/",working_file,".RData"
))){

load(paste0(projects_directory,"/",working_file,"/",working_file,".RData"),envir=.G
lobalEnv)
  }
  #Refresh reactive outputs if they exist
  if(exists("crust")){store_r$crust_r<-crust}
  if(exists("input_pt")){store_r$input_pt_r<-input_pt}
  if(exists("input_bulk")){store_r$input_bulk_r<-input_bulk}
  if(exists("major_elements")){store_r$major_elements_r<-major_elements}
}else{
  reactive_message$data
}
}
  first_load<-FALSE
})
})

```

User interface (ui.r)

Build the skeleton of the user interface with a top toolbar and multiple conditional panels contained in grouped tabs as follows:

- Toolbar
 - Projects directory
 - Working file
 - Save
 - Load
 - Run
 - Clear
 - Console
- Input parameters tab
 - Size
 - Pressure and temperature
 - Bulk composition
- Phase manipulations
 - Phase addition
 - Phase extraction
- Modelling options
 - Modelling data
 - Additional optional parameters
 - Extra Settings
- Outputs
 - Data file
 - Grid
 - Phase abundance
 - Phase assemblage maps

```
#####
## Rcrust (ui.r)
#####
#function-def: textInputRow(inputId, label, value = "")
textInputRow<-function (inputId, label, value = "")
{
  div(style="display:inline-block",
      tags$label(label, `for` = inputId),
      tags$input(id = inputId, type = "text", value = value,class="input-small"))
}
shinyUI(fixedPage(
# Add custom CSS & Javascript for Progress Indicator
  tagList(
    tags$head(
      tags$link(rel="stylesheet", type="text/css",href="style.css"),
      tags$script(type="text/javascript", src = "busy.js")
    )
  ),
  div(class = "busy",
      p("Busy.."),
      img(src="35.gif")
  ),
# Logo for Rcrust
  titlePanel(img(src="Rcrust_logo.png", align =
"left",width=150,height=60),windowTitle = "Rcrust"),
  fixedRow(textInputRow("projects_directory","Projects
Directory",value=paste0(substring(getwd(),1,nchar(getwd())-4),"Projects")),
```

```

        textInputRow("working_file", "Working File", working_file),

actionButton("save", "Save"), actionButton("load", "Load"), actionButton("run", "Run"), a
ctionButton("clear", "Clear"), actionButton("console", "Console"), align="right"),
    fixedRow(verbatimTextOutput("print_message")),
#Define Tabs
    tabsetPanel(id = "inTabset",
#Input Parameters UI Tab
tabPanel("Input Parameters",
    #Size Panel
wellPanel("Size", actionButton("size_panel_minimiser", label = "..."),
conditionalPanel("(input.size_panel_minimiser)%2 == 0",
fixedRow(
column(3, textInput("x_n", "X")),
column(3, textInput("y_n", "Y"))
)
)
),
    #PT Panel
wellPanel("Pressure and Temperature", actionButton("pt_panel_minimiser", label =
"..."), actionButton("pt_import_panel_minimiser", label = "<-"),
conditionalPanel("(input.pt_panel_minimiser)%2 == 0",
conditionalPanel("(input.pt_import_panel_minimiser)%2 == 1",
fileInput("file_pt", "Import P-T definitions"),
actionButton("import_pt", "Import")
),
        textInput("n_pt_def", "Number of PT definitions"),
    # Dynamic PT definition input boxes
uiOutput("pt")
)),
#Bulk Composition Panel
wellPanel("Bulk Composition", actionButton("bulk_comp_panel_minimiser", label =
"..."), actionButton("bulk_import_panel_minimiser", label = "<-"),
conditionalPanel("(input.bulk_comp_panel_minimiser)%2 == 0",
conditionalPanel("(input.bulk_import_panel_minimiser)%2 == 1",
fileInput("file_bulk", "Import bulk composition definitions"),
actionButton("import_bulk", "Import")
),
        textInput("n_comp_trans", "Number of Component Transformations"),
    # Dynamic transformation input boxes
uiOutput("trans"),
conditionalPanel("input.bulk_def_file == false",
uiOutput("maj"),
textInput("n_bulk_def", "Number of bulk definitions"),
    # Dynamic bulk definition input boxes
uiOutput("bulk")
),
conditionalPanel("input.bulk_def_file == true",
textInput("bulk_file", "Bulk file")
),
checkboxInput("bulk_def_file", "Import definitions from file", value = FALSE)
))
),
#End of Input Parameters UI Tab
#Phase Manipulations UI Tab
tabPanel("Phase Manipulations",
    #Phase Addition Panel
wellPanel("Phase Addition", actionButton("phase_addition_panel_minimiser", label =
"..."), actionButton("phase_addition_import_panel_minimiser", label = "<-"),
conditionalPanel("(input.phase_addition_panel_minimiser)%2 == 0",
conditionalPanel("(input.phase_addition_import_panel_minimiser)%2 == 1",
fileInput("file_ph_add", "Import phase addition definitions"),
actionButton("import_ph_add", "Import")
),
checkboxInput("ph_add", "Perform Phase Addition?", value = FALSE),
conditionalPanel("input.ph_add == true",
textInput("n_ph_add_def", "Number of addition definitions"),
    # Dynamic phase addition definition input boxes

```

```

        uiOutput("ph_add")
    )))
    #Phase Extraction Panel
    wellPanel("Phase Extraction",actionButton("phase_extraction_panel_minimiser",
label = "..."),actionButton("phase_extraction_import_panel_minimiser", label = "<-
"),
    conditionalPanel("(input.phase_extraction_panel_minimiser)%2 == 0",
    conditionalPanel("(input.phase_extraction_import_panel_minimiser)%2 == 1",
        fileInput("file_ph_extr", "Import phase extraction definitions"),
        actionButton("import_ph_extr","Import")
    ),
    ),
    checkboxInput("ph_extr", "Perform Phase Extraction?",value = FALSE),
    conditionalPanel("input.ph_extr == true",
        checkboxInput("reequilibrate_steps", "Re-equilibrate reactive subsystem after
phase extraction?",value = TRUE),
        textInput("n_ph_extr_def","Number of extraction definitions"),
        # Dynamic phase extraction input boxes
        uiOutput("ph_extr")
    )))
),
#End of Phase Manipulations UI Tab
#Modelling Options UI Tab
tabPanel("Modelling Options",
    #Modelling data Panel
    wellPanel("Modelling Data",actionButton("modelling_data_panel_minimiser", label =
"..."),
        conditionalPanel("(input.modelling_data_panel_minimiser)%2 == 0",
            textInput("meemum_path","Meemum version","meemum.exe"),
            textInput("perplex_option_file","Perple X Option File","perplex_option.dat"),
            textInput("thermodynamic_data_file","Thermodynamic Data File","hpl1ver.dat"),
            textInput("solution_models_file","Solution Models
File","solution_model_673.dat"),
            # Dynamic solution models input boxes
            uiOutput("solution_models")
        )),
    #Additional optional parameters Panel
    wellPanel("Additional optional
parameters",actionButton("additional_optional_parameters_panel_minimiser", label =
"..."),
        conditionalPanel("(input.additional_optional_parameters_panel_minimiser)%2 == 0",
            textInput("saturated_components","Saturated components"),
            textInput("saturated_phase_components","Saturated phase
components"),
            textInput("independent_potential_fugacity_activity","Independent
potential/fugacity/activity"),
            textInput("exclude_phases","Exclude phases")
        )),
    #Extra Settings Panel
    wellPanel("Extra Settings",actionButton("extra_settings_panel_minimiser", label =
"..."),
        conditionalPanel("(input.extra_settings_panel_minimiser)%2 == 0",
            selectInput("end_of_calc","When calculation is complete:",c("Return to
Interface","Return to
Interface","Logout","Logout","Shutdown","Shutdown"),selected="Return to
Interface",selectize=TRUE)
        ))
    ),
#End of Modelling Options UI Tab
#Outputs UI Tab
tabPanel("Outputs",
    sidebarLayout(
        sidebarPanel(
            textInput('phase_aliases',"Phase Aliases"),
            selectInput('output_type', 'Select Output', c("Data File","Grid","Phase
Abundance Along Path","PAM"), selected="Data File",selectize=TRUE),
            uiOutput("output_form_selection"),
            # Dynamic output selection boxes

```

```
      uiOutput("output_selection")
    ),
    mainPanel(
      h4(textOutput("output_header", container = span)),
      uiOutput("output_view")
    )
  )
),selected = "Input Parameters")
#End of tabs
)
#End of page
)
#End of Shiny
```

ADDENDUM D: Supplementary data for Research Paper 3

Included within the thesis files for evaluation is an excel document containing compositions of a natural s-type granite array and the modelled melt compositions produced by the study “A phase equilibrium investigation of selected source controls on the composition of melt batches generated by sequential melting of an average metapelite” developed as part of the thesis by M.J. Mayne.