



HAL
open science

Simultaneous Rational Function Reconstruction and applications to Algebraic Coding Theory

Ilaria Zappatore

► **To cite this version:**

Ilaria Zappatore. Simultaneous Rational Function Reconstruction and applications to Algebraic Coding Theory. Symbolic Computation [cs.SC]. Université de Montpellier, 2020. English. NNT: . tel-03013914v1

HAL Id: tel-03013914

<https://theses.hal.science/tel-03013914v1>

Submitted on 19 Nov 2020 (v1), last revised 29 Mar 2021 (v2)

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.

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Informatique

École doctorale : I2S - Information, Structures, Systèmes

Unité de recherche : LIRMM - Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

Reconstruction Rationnelle Simultanée et applications à la Théorie des Codes Correcteurs d'Erreurs

Simultaneous Rational Function Reconstruction
and applications to Algebraic Coding Theory

Présentée par Ilaria Zappatore

Le 16 octobre 2020

Sous la direction de Laurent Imbert

Devant le jury composé de

Gilles VILLARD

Daniel AUGOT

Clément PERNET

Magali BARDET

Elisa GORLA

Laurent IMBERT

Eleonora GUERRINI

Romain LEBRETON

Directeur de Recherche CNRS, LIP, Lyon

Directeur de Recherche, Inria Saclay

Maître de Conférences, Université Grenoble Alpes, LJK, Grenoble

Maître de Conférences, Université de Rouen, LITIS, Rouen

Professeuse, Université de Neuchâtel (Switzerland)

Directeur de Recherche CNRS, LIRMM, Montpellier

Maître de Conférences, Université de Montpellier, LIRMM, Montpellier

Maître de Conférences, Université de Montpellier, LIRMM, Montpellier

Président

Rapporteur

Rapporteur

Examinatrice

Examinatrice

Directeur

Co-encadrante

Co-encadrant



UNIVERSITÉ
DE MONTPELLIER

Résumé

Cette thèse étudie un problème de calcul formel qui a des applications et conséquences importantes sur la théorie des codes correcteurs algébriques : la *reconstruction rationnelle simultanée* (RRS). En effet, une analyse rigoureuse de ce problème amène à des résultats intéressants dans ce deux domaines scientifiques.

Plus précisément, la reconstruction simultanée de fractions rationnelles est le problème de la reconstruction d'un vecteur de fractions rationnelles ayant le même dénominateur étant donné ses évaluations (ou plus généralement étant donné ses restes modulo de polynômes différents). La particularité de ce problème consiste dans le fait que la contrainte du dénominateur commun réduit le nombre de points d'évaluation qui assurent l'existence d'une solution, au prix d'une éventuelle perte d'unicité. Une des principales contributions de ce travail consiste à prouver que l'unicité est garantie pour *quasiment tous les instances* de ce problème.

Ce résultat a été obtenu par l'élaboration des résultats et techniques précédents dérivées des applications du problème RRS, depuis la résolution de systèmes linéaires polynomiaux jusqu'au décodage de codes Reed-Solomon entrelacés.

Dans ce travail, nous avons aussi étudié et présenté une autre application du problème RRS, concernant le problème de la construction d'algorithmes *tolérants aux fautes* : des algorithmes *résistants* aux erreurs de calcul. Ces algorithmes sont construits en introduisant une redondance et en utilisant des outils de codes correcteurs d'erreurs pour détecter et éventuellement corriger les erreurs qui se produisent pendant les calculs. Dans ce contexte d'application, nous améliorons une technique existante de tolérance aux fautes pour la résolution de systèmes linéaires polynomiaux par interpolation-évaluation, avec une attention particulière au problème RRS correspondant.

Abstract

This dissertation deals with a *Computer Algebra* problem which has significant consequences in *Algebraic Coding Theory* and *Error Correcting Codes: the simultaneous rational function reconstruction*. Indeed, an accurate analysis of this problem leads to interesting results in both these scientific domains.

More precisely, the simultaneous rational function reconstruction is the problem of reconstructing a vector of rational functions with the same denominator given its evaluations (or more generally given its remainders modulo different polynomials). The peculiarity of this problem consists in the fact that the common denominator constraint reduces the number of evaluation points needed to guarantee the existence of a solution, possibly losing the uniqueness. One of the main contribution of this work consists in the proof that uniqueness is guaranteed for *almost all* instances of this problem.

This result was obtained by elaborating some other contributions and techniques derived by the applications of SRFR, from the polynomial linear system solving to the decoding of Interleaved Reed-Solomon codes.

In this work, we will also study and present another application of the SRFR problem, concerning the problem of constructing *fault-tolerant* algorithms: algorithms *resilient* to computational errors. These algorithms are constructed by introducing redundancy and using error correcting codes tools to detect and possibly correct errors which occur during computations. In this application context, we improve an existing fault-tolerant technique for polynomial linear system solving by interpolation-evaluation, by focusing on the SRFR problem related to it.

Contents

Remerciements	9
Résumé de la thèse	11
Introduction	19
1 Simultaneous Rational Function Reconstruction	29
1.1 Rational Function Reconstruction	29
1.1.1 RFR by the Extended Euclidean Algorithm	32
1.2 Simultaneous Rational Function Reconstruction	33
1.3 The $\mathbb{K}[x]$ -module of solutions of SRFR	35
1.3.1 Row degrees and reduced basis	38
1.3.2 Solutions of SRFR and Relation Module	42
1.4 Application to Polynomial Linear System Solving	45
1.5 A short summary of the chapter	48
2 Application of SRFR to Coding Theory	51
2.1 Basics of Coding Theory	52
2.1.1 Channel Model	52
2.1.2 Block codes	54
2.1.3 Basic decoding principles	57
2.2 Reed-Solomon Codes	60
2.2.1 Decoding RS codes	62
2.3 Interleaved Reed-Solomon code	69
2.3.1 Decoding IRS codes	70
2.4 A short summary of the chapter	78
3 Generic Uniqueness of SRFR	81
3.1 Generic Row Degrees of the Relation Module	82
3.1.1 Monomial orders on modules	83
3.1.2 Row degrees of a relation module as row rank profile	86
3.1.3 Constraints on linearly independent monomial families	88

3.1.4	Generic row degrees	93
3.2	Generic row degrees of the SRFR Relation Module	95
3.3	Conclusions and open problems	98
4	Simultaneous Cauchy interpolation with errors	101
4.1	ABFT for Polynomial Linear System Solving by Evaluation-Interpolation . .	103
4.1.1	Simultaneous Cauchy interpolation with errors	104
4.1.2	Polynomial Linear Solving with Errors	111
4.2	Early termination techniques	115
4.2.1	Previous results	117
4.2.2	Our contribution	123
4.3	Conclusion and open problems	129
	Concluding Remarks	137
	List of Acronyms	141
	List of Algorithms	142
	List of Figures	143

Remerciements

Le moment d'écrire les remerciements est crucial : c'est là que l'on réalise que ce long voyage touche à sa fin. C'est donc avec une grande émotion (assez étrange pour "le roc" que je suis) que je commence à écrire ce *premier chapitre* de cette thèse, qui représente pourtant la *conclusion* de ce *chapitre* de ma vie.

J'ai décidé d'écrire en plusieurs langues, en me basant sur les personnes auxquelles ces remerciements se réfèrent. D'ailleurs grâce à cette expérience qui m'a "forcé" à apprendre le français : je me souviens encore de la panique que j'ai ressentie au début lorsque les gens me parlaient.

Je tiens tout d'abord à remercier Daniel Augot et Clément Pernet d'avoir accepté de rapporter cette thèse, sous le soleil brûlant de juillet. Un grand merci pour le temps que vous avez consacré à ce travail. Je tiens aussi à remercier les autres membres du jury : Magali Bardet, Elisa Gorla, Gilles Villard.

Merci à Laurent pour avoir accepté d'être mon directeur de thèse et pour m'avoir toujours soutenue.

Un grosso ringraziamento a Eleonora. Non dimenticherò mai le tue parole di sostegno sin dagli inizi di questa avventura. Sei stata tu a spronarmi a darle inizio, quando tutte le mie insicurezze stavano prendendo il sopravvento portandomi a fare delle scelte lavorative di cui sicuramente in seguito mi sarei pentita. Sei stata la mia roccia umanamente e scientificamente e senza il tuo supporto, i tuoi dolcetti, le nostre risate e i nostri lunghi discorsi esistenziali, questo percorso non sarebbe stato lo stesso

Je remercie Romain de m'avoir fait découvrir le calcul formel, un nouveau domaine pour moi. Je te remercie pour tout ce que tu m'a appris, pour ton immense patience et pour ta disponibilité surtout en période de stress et de difficultés.

Merci à tous mes collègues d'équipe. Merci Fabien, pour nos discours, qui ont été très importants pour moi et que je chérirai toujours. Merci à Bruno toujours disponible à avoir des petites conversations, surtout les soirs quand désormais le laboratoire était désert. Merci à "chef" Pascal, pour son support et soutien. Merci à Armelle, ma copine de bureau et de thèse dans l'équipe.

Merci Nico pour ton aide pour toutes les formalités administratives. Merci pour ta disponibilité, ton aide a été très précieuse.

Merci à Cyril, pour tes délicieuses pâtisseries et biscuits.

Un ringraziamento speciale al mio compagno in questa avventura, Emanuele. Compagno di dottorato, di uscite, risate, di giochi di società, la mia valvola di sfogo nei momenti di crisi esistenziale. Dovrebbero farti una statua solo per avermi sopportato in questi tre anni.

Grazie a Ceci, fedele compagna di chiaccherate e yoga mattutino.

Merci à tous mes amis de Montpellier : Francesco et Chloé, Marcello et Ada, Bastien et Kathrine, Mathieu et Sara, Safa, Linh. Sans vous, cette expérience n'aurait pas été la même.

Grazie alla mia FAMIGLIA : Mamma, Papá, Riccardo, Anna, Tonino, Maristella, Vale e Francesco e ai miei nipotini Fede, Lollo, Tato. Grazie per il vostro continuo supporto e perché mi sopportate pazientemente, nonostante la distanza che ci separa.

Infine concludo ringraziando Francesco, *“If the sun refused to shine I would still be loving you. When mountains crumble to the sea, there will still be you and me...”* (Thank you - Led Zeppelin).

Résumé de la thèse

Cette thèse a pour objectif d'étudier un problème de calcul formel ainsi que son application à la théorie des codes correcteurs algébriques.

Le calcul formel est le domaine scientifique qui s'intéresse à l'analyse et au développement d'algorithmes pour la résolution de problèmes mathématiques en représentations finie et exacte.

D'autre part, la théorie du codage trouve son origine dans les publications pionnières de Shannon [Sha48] et Hamming [Ham50]. Shannon a formalisé le concept de *communication fiable* sur un *canal bruité*. Il a également déterminé une limite inférieure sur la quantité de redondance qui doit être rajoutée aux informations transmises pour assurer une transmission *presque sans erreur* (ce résultat est connu comme le *deuxième théorème de Shannon*). Cependant, la preuve de ce théorème n'est pas constructive et on ne sait pas comment toujours construire des codes correcteurs d'erreurs qui atteignent réellement la limite de Shannon. Par ailleurs, Hamming a proposé l'une des premières classes de *codes correcteurs d'erreurs* et il a introduit la notion de *distance de Hamming*, une métrique qui mesure la distance entre deux mots de code en comptant le nombre de positions dans lesquelles ils sont différents.

La théorie des codes correcteurs algébriques est un sous-domaine de la théorie du codage, dans lequel toutes les propriétés des codes sont exprimées en termes algébriques. Elle utilise également des techniques algébriques classiques et modernes pour la conception de codes correcteurs d'erreurs.

Les deux disciplines du calcul formel et de la théorie des codes correcteurs algébriques peuvent être combinées : les problèmes algébriques liés aux codes correcteurs d'erreurs peuvent être résolus efficacement par des algorithmes et outils de calcul formel. Un exemple classique qui met en évidence cette interaction concerne une classe célèbre et très utilisée en pratique de codes correcteurs d'erreurs algébriques : les codes de Reed-Solomon (RS) [RS60]. Les codes RS ont plusieurs propriétés remarquables qui les rendent largement utilisés dans des applications pratiques. Par exemple, ils sont *maximum distance séparable* (MDS), c'est-à-dire qu'ils atteignent la borne de Singleton. D'un point de vue algébrique, les codes RS peuvent être considérés comme les évaluations de polynômes de degrés bornés. Le décodage d'un mot de code RS consiste à récupérer un polynôme de degré borné étant donné ses évaluations potentiellement erronées (*problème d'interpolation avec erreurs*, IaE). Les deux techniques classiques de décodage des codes RS, l'une basée sur l'interpolation [BW86] et l'autre basée

sur le syndrome [Ber68], ramènent le problème de décodage à un problème classique de calcul formel : *la reconstruction rationnelle* (RR). Cela conduit à la construction des décodeurs efficaces à *distance minimale limitée* qui diffèrent fondamentalement par l’algorithme choisi pour effectuer la RR correspondante ([BW86, Gao03, SKHN75, Ber68]).

Dans cette thèse, nous étudions la *reconstruction rationnelle simultanée* (RRS) en analysant surtout la condition sur ses paramètres qui garantit l’unicité de sa solution ainsi que son impact sur les problèmes connexes de la théorie des codes correcteurs algébriques.

La RRS désigne le problème de la reconstruction d’un vecteur de fractions rationnelles ayant le même dénominateur, c’est-à-dire $\mathbf{v}/d = (v_1/d, \dots, v_n/d)$, étant donné $u_i = v_i/d \pmod{a_i}$ et les bornes sur les degrés $\deg(v_i) < N_i$, $\deg(d) < D$. La RRS généralise le problème d’interpolation en considérant $a_1 = \dots = a_n = \prod(x - \alpha_j)$ pour des points d’évaluation α_j distincts. En effet dans ce cas les équations modulaires $u_i = v_i/d \pmod{a_i}$ deviennent des équations sur les évaluations $u_i(\alpha_j) = v_i(\alpha_j)/d(\alpha_j)$. Par soucis de simplicité, nous nous concentrons dans cette partie sur la version d’interpolation de la RRS et nous supposons que $N_1 = \dots = N_n = N$. Cependant, nous remarquons que dans ce travail, nous étudions également le cas général.

Dans ce travail, nous étudions la RRS en nous concentrant sur le nombre de points d’évaluation qui assurent l’unicité de la solution du problème. Par *unicité*, nous entendons que tous les vecteurs des fractions rationnelles correspondants à des solutions de la RRS sont égaux, ou en d’autres termes, que chaque solution (\mathbf{v}, d) est un multiple polynomial d’une *solution minimale*. Afin de déterminer les solutions de la RRS, comme pour le problème RR classique, nous nous concentrons sur le *problème linéaire plus faible* de la reconstruction de (\mathbf{v}, d) qui satisfait $v_i(\alpha_j) = u_i(\alpha_j)d(\alpha_j)$ et tel que $\deg(v_i) < N$ et $\deg(d) < D$.

Pour trouver les solutions de la RRS, nous pouvons appliquer la RR classique à chaque composante et avec $L \geq L_{RR} = N + D - 1$ points d’évaluation, nous avons unicité de la RRS.

La particularité de la RRS réside dans le fait que le dénominateur commun réduit le nombre d’inconnues du système linéaire homogène liées à ce problème, en diminuant le nombre de points d’évaluation $L_{RRS} = N + (D - 1)/n$ qui assurent l’existence d’une solution non-triviale.

Cependant, il y a des instances \mathbf{u} de la RRS pour lesquels ce nombre de points d’évaluation n’est pas suffisant pour garantir l’unicité (comme montre l’exemple 1.2.1). Dans cette thèse, nous étudions des instances qui conduisent à l’unicité, en supposant que le nombre de points d’évaluation soit $L = L_{RRS}$. Les deux travaux précédents qui motivent notre analyse dans ce sens proviennent de deux applications de RRS : la résolution de systèmes linéaires polynomiaux et le décodage de codes de Reed-Solomon entrelacés (RSE).

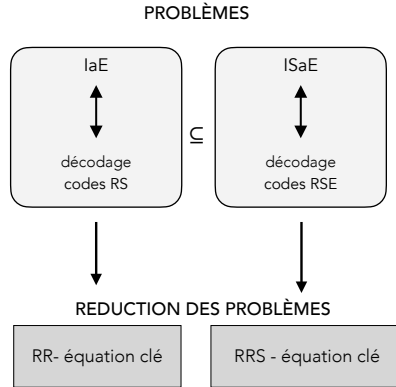
La solution d’un système carré et non singulier d’équations linéaires à coefficients polynomiaux (SLP), est un vecteur de fractions rationnelles ayant le même dénominateur, selon la règle de Cramer. Certaines des techniques de résolution des SLP, par exemple l’évaluation-

interpolation, peuvent utiliser RRS pour abaisser le nombre de points d'évaluation. Dans ce cadre, la réduction du nombre d'évaluations a un impact direct sur la complexité des algorithmes de résolution qui dépend de ce nombre. Une contribution importante dans ce contexte d'application provient de [OS07]. Dans cet article, les auteurs ont prouvé qu'avec L_{RRS} points d'évaluation et sous certaines hypothèses spécifiques de degrés, RRS admet une solution unique.

Un autre résultat important provient de la théorie des codes algébrique et il est lié aux codes RSE. En général, l'*entrelacement* est une construction qui peut être appliquée à différents codes correcteurs d'erreurs et qui est utilisée pour la correction des *erreurs par paquets*, c'est-à-dire des erreurs étendues à des symboles consécutifs. Nous considérons ici cette construction appliquée aux codes RS. Plus précisément, un code l -RSE est une somme directe de l codes RS avec les mêmes points d'évaluation. Par conséquent, ils peuvent être considérés comme des évaluations de vecteurs de polynômes de degrés bornés. Dans ce contexte particulière, une erreur par paquets est une erreur qui corrompt une évaluation dans tous les l mots de code. D'un point de vue algébrique, le décodage d'un mot de code RSE consiste à la reconstruction d'un vecteur de polynômes étant données ses évaluations, dont certaines sont erronées (*interpolation simultanée avec erreurs*, ISaE).

Le problème du décodage des codes RSE a été l'objet de beaucoup d'attention en ces derniers temps ([BKY03, BMS04, SSB07, SSB09, SSB10, PR17]. En effet, les codes RSE sont intéressants car ils peuvent être décodés au-delà de la moitié de la distance minimale. C'est pourquoi tous les décodeurs proposés dans ces articles sont des décodeurs à *distance limitée partiels*, car ils peuvent échouer pour quelques erreurs spécifiques. En outre, le *rayon de décodage* de ces décodeurs atteint la *limite de Shannon* en supposant que le paramètre d'entrelacement l tend vers l'infini.

La technique pour le décodage des codes RSE basée sur l'interpolation ([BKY03, BMS04]) réduit le problème de décodage à une RRS. En effet, dans [BKY03] il a été prouvé qu'avec le nombre d'évaluations dérivées de la contrainte du dénominateur commun, pour tous les vecteurs de polynômes \mathbf{v} et pour quasiment toutes les erreurs \mathbf{e} , le problème RRS appliqué aux instances \mathbf{u} tel que $\mathbf{u}(\alpha_j) = \mathbf{v}(\alpha_j) + \mathbf{e}_j$ admet une solution unique. Ce résultat, même s'il s'agit d'un scénario différent avec des erreurs, motive notre étude sur les cas conduisant à l'unicité du problème général de la RRS.



Dans ce document, nous étudions et présentons aussi une autre application de la RRS, concernant la construction d’algorithmes *tolérants aux fautes*. Dans ce cadre, nous voyons comment les outils des codes correcteurs d’erreurs sont utilisés pour une application qui va au-delà du scénario de communication classique.

Les technologies de calcul à haute performance (supercalculateurs) contiennent des milliers de nœuds de calcul mis en réseau (calcul parallèle) pour fournir de très hautes performances. Plus le nombre de composants du système augmente, plus le problème de corriger des erreurs introduites par ces nœuds devient important. Par exemple, les supercalculateurs modernes commettent environ 3,5 fautes par jour [DGP⁺19, LC18]. Par conséquent, sans un changement drastique au niveau algorithmique, un tel taux de fautes empêchera certainement les supercalculateurs de progresser. C’est pourquoi de nombreuses techniques et algorithmes *tolérants aux fautes* ont été proposés pour détecter et corriger ces erreurs.

Ces fautes peuvent être traitées par des techniques de *contrôle et de redémarrage*, consistant à enregistrer périodiquement des données sur des dispositifs de stockage ([BD93]). Cependant, cette approche pourrait s’avérer coûteuse en termes de ressources, car elle pourrait nécessiter un stockage externe ou de la bande passante sur le réseau. L’autre grande approche pour tolérer les fautes est logicielle (*technique de construction des algorithmes tolérants aux fautes* (TCATF) [HA84]). Cette technique exploite les outils des codes correcteurs d’erreurs algébriques en rajoutant une redondance aux entrées du problème afin de détecter et éventuellement de corriger les erreurs de calcul survenues dans des environnements distribués parallèles.

Les techniques TCATF [HA84] se caractérisent par le *codage* des entrées de l’algorithme, la *ré-conception* de l’algorithme pour qu’il puisse fonctionner sur les données codées et la *répartition* de certaines étapes de calcul entre des nœuds (*parallélisation*). Dans ce cadre, les erreurs sont introduites par les nœuds et le modèle d’erreur dépend fortement du schéma de parallélisation.

Dans cette thèse, nous étudions une technique TCATF pour la résolution des systèmes linéaires polynômiaux par évaluation-interpolation [BK14, Per14, KPSW17]. Considérons un

SLP carré, non singulier $A(x)\mathbf{y}(x) = \mathbf{b}(x)$ et prenons $\mathbf{v}(x)/d(x)$ sa solution où $\text{pgcd}(\text{pgcd}_i(v_i), d) = 1$ and d est unitaire. En général, la technique d'évaluation-interpolation pour la résolution des SLP consiste à *évaluer* $A(x)$ et $\mathbf{b}(x)$ en certains points d'évaluation distincts $\{\alpha_1, \dots, \alpha_L\}$, à la *résolution ponctuelle* des systèmes évalués $\mathbf{y}(\alpha_j) = A(\alpha_j)^{-1}\mathbf{b}(\alpha_j)$ et à l'*interpolation* et reconstruction de $\mathbf{y}(x) = \mathbf{v}(x)/d(x)$ étant données ses évaluations.

L'idée principale de la technique TCATF appliquée à l'évaluation-interpolation est de modifier cette méthode par l'introduction de la redondance, en considérant plus de points d'évaluation par rapport au nombre nécessaire pour le cas général. Dans ce scénario, l'étape d'*évaluation* de la technique classique d'évaluation-interpolation est réalisée par différents nœuds (*parallélisation*). Ces nœuds introduisent éventuellement des erreurs et calculent $\mathbf{y}(\alpha_j) \neq \mathbf{v}(\alpha_j)/d(\alpha_j)$.

Le problème de résoudre un *système linéaire polynomial avec erreurs* (SLPaE) [BK14, Per14, KPSW17] est alors le problème de la récupération du vecteur des fractions rationnelles \mathbf{v}/d , qui est une solution du SLP, étant données ses évaluations où certaines pourraient être erronées. Dans ce cas également, la technique de résolution de ce problème peut être considérée comme une RRS.

Nos contributions

Nos contributions concernent un résultat d'unicité sur la RRS [GLZ20b] et d'autres résultats [GLZ19, GLZ20a] sur le SLPaE et son problème plus général de l'*interpolation de Cauchy simultanée avec erreurs* (ICSaE).

Unicité générique de RRS. Une première contribution nouvelle développée dans cette thèse concerne le problème général de la RRS.

Problème 2. *Reconstruction Rationnelle Simultanée*

Entrée : $a_1, \dots, a_n \in \mathbb{K}[x], \mathbf{u} = (u_1, \dots, u_n) \in \mathbb{K}[x]^n$, où $\deg(u_i) < \deg(a_i)$ et $1 \leq N_i \leq \deg(a_i), 1 \leq D \leq \min_{1 \leq i \leq n} \{\deg(a_i)\}$

Sortie : $(\mathbf{v}, d) = (v_1, \dots, v_n, d) \in \mathbb{K}[x]^{n+1}$ tel que

$$[v_i = du_i \bmod a_i]_{1 \leq i \leq n}, \quad \deg(v_i) < N_i, \quad \deg(d) < D. \quad (1.5)$$

Dans le Chapitre 3 nous prouvons (Théorème 3.2.1) que si

$$\sum_{i=1}^n \deg(a_i) = \sum_{i=1}^n N_i + D - 1,$$

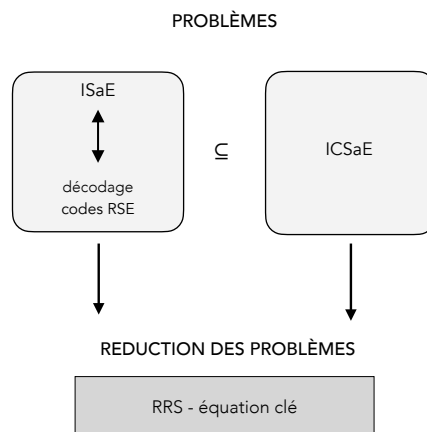
pour *quasiment tous les instances* \mathbf{u} , la RRS admet une solution unique. Notre approche pour

prouver le Théorème 3.2.1 consiste à étudier les *degrés en ligne* d'un $\mathbb{K}[x]$ -module particulier, le *module de relation* lié à une *matrice spécifique*. En effet, les solutions de RRS sont des éléments de ce module de relation avec *degrés en ligne décalés*, où les décalages sont nécessaires pour intégrer les contraintes de degré. Dans le cas d'unicité, il n'y a qu'un seul élément d'une base du module de relation avec des degrés en ligne négatifs. Cela représente un outil central pour vérifier l'unicité de la RRS. En effet, dans le Théorème 3.2.1 nous prouvons que pour quasiment toutes les instances du problème (*instances génériques*) il n'y a qu'un seul générateur avec des degrés en ligne décalés négatifs.

Avant de prouver ce théorème, nous prouvons un résultat sur le *degré en ligne générique* des modules de relation liés à des matrices générales (Section 3.1, Corollaire 3.1.2). Des travaux précédents ont étudié les degrés en ligne génériques de différents $\mathbb{K}[x]$ -modules : *e.g.* les modules des récurrences vectorielles d'une suite de matrices scalaires [Vil97] ou pour le noyau d'une matrice polynomiale de dimensions spécifiques [JV05]. Les degrés génériques apparaissent également comme des dimensions de blocs d'une forme de Hessenberg décalée [PS07]. Dans tous ces cas, aucun décalage n'est pris en compte. Nous prouvons notre résultat pour tous décalages et pour tous modules de relation en reformulant et en adaptant certaines des techniques introduites dans les articles mentionnés ci-dessus.

Interpolation de Cauchy Simultanée avec Erreurs. La *interpolation de Cauchy simultanée avec erreurs* (ICSaE) consiste dans la reconstruction d'un vecteur de fractions rationnelles $v(x)/d(x)$ de degrés bornés, étant données ses évaluations dont certaines pourraient être erronées. Il s'agit d'un problème plus général que la résolution du SLPaE défini ci-dessus.

Dans [GLZ19], à la suite de [Per14], nous observons que ce problème est l'*extension rationnelle* du problème de décodage des codes RSE. En effet, dans ce cas nous voulons récupérer un vecteur de *fractions rationnelles* au lieu d'un vecteur de *polynômes*.



Ce lien nous permet d'étendre la même technique *basée sur l'interpolation* [BKY03] à ce cas rationnel, en réduisant ICSaE à une RRS appliqué aux instances \mathbf{u} de la forme $\mathbf{u}(\alpha_j) =$

$\mathbf{v}(\alpha_j)/d(\alpha_j) + \mathbf{e}_j$. Les résultats précédents sur ce sujet [BK14, KPSW17], plus liés au cas spécifique du SLPaE, ont montré qu'avec

$$L_{BK} = N + D - 1 + 2\tau \quad (1)$$

où $\tau \geq |E| := |\{j \mid \mathbf{e}_j \neq \mathbf{0}\}|$ est une borne sur le nombre d'erreurs, alors nous avons l'unicité de la RRS. Ce nombre de points coïncide avec L_{RR} , obtenu en appliquant le RR avec contraintes de degrés $N + \tau$, $D + \tau$ composant par composant. Dans [GLZ19], en soulignant la similarité entre ce problème et le décodage des codes RSE, nous proposons l'Algorithme 6 pour la résolution de ICSaE qui réduit ce nombre de points à

$$L_{GLZ1} = N + D - 1 + \tau + \lceil \tau/n \rceil. \quad (2)$$

Cependant, comme pour le cas des codes RSE, puisque nous sommes en dessous du nombre de points d'évaluation qui garantit l'unicité de la RRS, cet algorithme pourrait échouer pour quelques erreurs spécifiques.

Résolution d'un Système Linéaire Polynomial avec Erreurs. Le ICSaE est un cas général de SLPaE, dans lequel on veut récupérer un vecteur de fractions rationnelles ayant le même dénominateur, qui est une solution d'un SLP $A(x)\mathbf{y}(x) = \mathbf{b}(x)$. Pour cette raison, dans ce cas, nous pouvons rajouter le degré de la matrice de coefficients A et du vecteur \mathbf{b} comme entrées supplémentaires. Ainsi, de la même manière que ICSaE, SLPaE peut également être réduit à une reconstruction rationnelle simultanée appliqué à des instances de la forme \mathbf{u} où $\mathbf{u}(\alpha_j) = \mathbf{v}(\alpha_j)/d(\alpha_j) + \mathbf{e}_j$ et $\mathbf{v}(\alpha_j)/d(\alpha_j)$ est une solution d'un SLP $A(x)\mathbf{y}(x) = \mathbf{b}(x)$. Par conséquent, nous pouvons obtenir les mêmes résultats sur le nombre de points d'évaluation d'avant.

D'ailleurs dans [KPSW17], Kaltofen *et al.* ont montré qu'avec

$$L \geq \min\{L_{BK}, L_{KPSW}\} \quad (3)$$

points d'évaluation, où $L_{BK} = N + D - 1 + 2\tau$ (comme dans (1)) et

$$L_{KPSW} = \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + 2\tau \quad (4)$$

la RRS correspondant a une solution unique.

Dans cette thèse, nous présentons un résultat, issu d'un travail en cours [GLZ20a] qui réduit ce nombre à

$$L \geq \min\{L_{GLZ1}, L_{GLZ2}\} \quad (5)$$

points d'évaluation, où $L_{GLZ1} = N + D - 1 + \tau + \lceil \tau/n \rceil$ (comme dans (6)) et

$$L_{GLZ2} = \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + \tau + \lceil \tau/n \rceil. \quad (6)$$

Cependant, même dans ce cas, notre algorithme (une version étendue de l'Algorithme 6) peut échouer pour quelques erreurs spécifiques.

Techniques de terminaison anticipée. Nous concluons en présentant la dernière contribution de cette thèse, qui fait partie d'un travail en cours (certains résultats se trouvent dans [GLZ20a]).

Considérez le problème SLPaE. Nous observons que tous les nombres de points d'évaluation introduits jusqu'à présent $L_{BK}, L_{KPSW}, L_{GLZ1}, L_{GLZ2}$ dépendent fortement des bornes N (respectivement D) de degrés du numérateur (dénominateur) de la solution qu'on cherche à reconstruire et de la borne sur le nombre d'erreurs τ . Ainsi, une surestimation des degrés réels et du nombre réel d'erreurs (que nous ne connaissons pas a priori) pourrait augmenter considérablement le nombre de points d'évaluation par rapport au nombre dont nous avons réellement besoin. Une stratégie classique [KPSW17] pour surmonter cette limite consiste à effectuer une *technique de terminaison anticipée* qui, à partir d'un nombre de points d'évaluation *petit*, incrémente itérativement ce nombre jusqu'à ce qu'une certaine valeur *minimale* soit atteinte. Nous soulignons que cette technique vise à diminuer éventuellement le nombre de points d'évaluation, afin d'accélérer les calculs.

Dans cette thèse, nous présentons aussi un *algorithme de terminaison anticipée* (Algorithme 11) qui *réduit* le nombre d'évaluations par rapport aux résultats précédents [KPSW17].

Introduction

This dissertation deals with a *Computer Algebra* problem especially focusing on its application in *Algebraic Coding Theory*.

Computer algebra refers to the scientific domain aiming to the analysis and development of algorithms for solving mathematical problems and manipulating algebraic tools. It involves computations in algebraic structures, such as groups and fields, polynomial rings, rational function fields etc.

Coding theory has its origin in the pioneering publications by Shannon [Sha48] and Hamming [Ham50]. Shannon formalized the concept of *reliable communication* over a *noisy* channel. He also determined a limit on the amount of redundancy which should be added to the transmitted information to provide a nearly error-free transmission (this results is known as the *Noisy Channel Coding Theorem*). However, the proof of the Noisy Channel Coding Theorem is non-constructive and it is not clear how to construct error-correcting codes which actually achieve the Shannon limit. Hamming proposed one of the first class of *error correcting codes* and introduced the *Hamming distance* notion, a metric which measures the distance between codewords by counting the number of positions in which they disagree.

Algebraic Coding Theory is a subfield of coding theory, where all the properties of codes are expressed in algebraic terms. It also employs classical and modern algebraic techniques for the design of error correcting codes.

The two disciplines of computer algebra and algebraic coding theory can be combined: algebraic problems related to error correcting codes can be efficiently solved by computer algebra algorithms. A classical example which highlights this interaction involves a famous and widespread class of algebraic error correcting code: the Reed-Solomon (RS) codes [RS60]. RS codes have several remarkable properties which make them widely used in practical applications. For instance they are *maximum distance separable* (MDS), *i.e.* they attain the Singleton bound [Sin64] on the minimum distance with equality. From an algebraic point of view, RS codes can be seen as the evaluations of polynomials of bounded degrees. The decoding of a RS codeword consists in the recovering of a bounded degree polynomial given its evaluations, some of which being erroneous (*the interpolation with errors* problem). Both classical decoding techniques for RS codes, the *interpolation-based* [BW86] and the *syndrome-based* [Ber68], can be seen as a classical computer algebra problem: the *rational function reconstruction* (RFR). This leads to the construction of efficient *bounded minimum distance*

(BMD) decoders which basically differ in the algorithm chosen to solve the corresponding RFR ([BW86, Gao03, SKHN75, Ber68]).

In this thesis we study a computer algebra problem, *i.e.* the *Simultaneous Rational Function Reconstruction* (SRFR), focusing on the conditions on its parameters in order to guarantee the uniqueness of its solution. We also analyze its impact on the related algebraic coding theory problems.

SRFR refers to the problem of recovering a *vector of rational functions* with the *same denominator*, *i.e.* $\mathbf{v}/d = (v_1/d, \dots, v_n/d)$, given $u_i = v_i/d \bmod a_i$ and the degree bounds $\deg(v_i) < N_i$, $\deg(d) < D$. SRFR generalizes the *interpolation problem* when the a_i 's satisfy $a_1 = \dots = a_n = \prod (x - \alpha_j)$ for distinct α_j , since in this case the modular equations $u_i = v_i/d \bmod a_i$ become equations on the evaluations $u_i(\alpha_j) = v_i(\alpha_j)/d(\alpha_j)$. For the sake of simplicity in this part we focus on the interpolation version of SRFR and we assume that $N_1 = \dots = N_n = N$. Besides, we remark that in this work we study also the general case.

In this thesis we study SRFR focusing on the number of evaluation points needed for the *uniqueness* of this problem solution. By *uniqueness* we mean that all the vectors of rational function which are solutions of SRFR are equal, or in other terms, that any solution (\mathbf{v}, d) is a polynomial multiple of a *minimal one*. In order to determine SRFR solutions, as for the classical RFR problem, we focus on the *weaker* linear problem of recovering (\mathbf{v}, d) which satisfies $v_i(\alpha_j) = u_i(\alpha_j)d(\alpha_j)$ and such that $\deg(v_i) < N$ and $\deg(d) < D$.

In order to solve SRFR we can apply the classical RFR componentwise. With $L \geq L_{RFR} = N + D - 1$ evaluation points, we have the uniqueness of SRFR.

Besides, the peculiarity of SRFR lies in the fact that the common denominator feature reduces the number of unknowns of the homogeneous linear system related to this linear problem, decreasing the number of evaluation points, *i.e.* $L_{SRFR} = N + (D - 1)/n$, needed to ensure the *existence* of a nontrivial solution.

However, there are instances \mathbf{u} of SRFR for which this number of evaluation points is not sufficient to guarantee uniqueness (as shown in Example 1.2.1). In this dissertation we study instances which lead to uniqueness, assuming that the number of evaluation points is $L = L_{SRFR}$. The two contributions which motivates our analysis in this direction derived from two applications of SRFR: the *polynomial linear system solving* and the *decoding of interleaved Reed-Solomon* (IRS) codes.

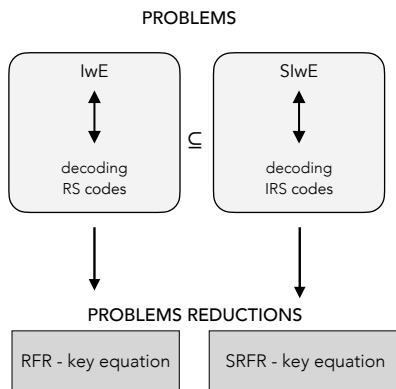
The solution of a square, nonsingular system of linear equations with polynomial coefficients, *i.e.* *Polynomial Linear System* (PLS) is a vector of rational functions with the same denominator, by the Cramer's Rule. Some of the techniques for PLS solving, *e.g.* the *evaluation-interpolation*, are based on an SRFR. In this framework, the reduction of the number of evaluations has a direct impact on the complexity of the resolution algorithms which depends on this number. A significant contribution in this application context derives from [OS07]. In this article, the authors proved that with the number of evaluation points derived

from the common denominator constraint and under some *specific assumptions on the degree bounds*, SRFR admits a unique solution.

Another important result comes from algebraic coding theory. It is related to IRS codes. In general *interleaving* is an encoding construction which can be applied to different error correcting codes, used in setting of *burst errors*, *i.e.* errors extended to consecutive symbols. Here we consider this construction applied to RS codes. More specifically, an l -IRS code is a direct sum of l RS codes with the same evaluation points. Therefore, they can be seen as evaluations of *vector* of polynomials of bounded degrees. In this specific setting, a burst error is an error which corrupts the same evaluation point position in all the l codewords. From an algebraic point of view, decoding an IRS codeword is the problem of recovering a vector of polynomials given its evaluations, some of which erroneous (*Simultaneous Rational Function with Errors*).

In recent years, the decoding problem of IRS codes has achieved a lot of attention, *e.g.* [BKY03, BMS04, SSB07, SSB09, SSB10, PR17]. Indeed, IRS codes are interesting because they can be decoded beyond half of the minimum distance. For this reason all the decoders proposed in these articles are *partial* BD decoders, in the sense they can fail for few error patterns. Furthermore, the decoding radius of these decoders reaches the Shannon's limit assuming that the interleaving parameter l tends to infinity.

The *interpolation-based* technique for IRS codes decoding ([BKY03, BMS04]) reduces the decoding problem to an SRFR applied to instances \mathbf{u} such that $\mathbf{u}(\alpha_j) = \mathbf{v}(\alpha_j) + \mathbf{e}_j$ where \mathbf{e}_j is the error vector. Indeed, [BKY03] proved that with the number of evaluations derived from the common denominator constraint, for all vectors of polynomials \mathbf{v} and for almost all errors \mathbf{e} , the SRFR problem applied to instances \mathbf{u} such that $\mathbf{u}(\alpha_j) = \mathbf{v}(\alpha_j) + \mathbf{e}_j$ admits a unique solution. This result, even if in a different scenario with errors, motivates our study about instances leading to uniqueness of the general SRFR problem.



In this document we also study and present another application of SRFR, concerning the construction of *fault-tolerant algorithms*. In this framework, we see how the error correcting

codes tools are used for an application setting which goes beyond the classical communication scenario.

With the advent of High Performance Technologies (supercomputers), composed of thousands of computing nodes networked together to provide very high performances, the problem of correcting faults introduced by these nodes becomes increasingly relevant. These faults can be handled by *checkpoint-restart* techniques, consisting in periodically saving data onto storage devices ([BD93]). However this approach could be expensive in terms of resources since it could require external storage like memories or network bandwidth. An alternative approach consists in the application of the *algorithm-based fault tolerance* (ABFT) technique [HA84]. This technique exploits the algebraic tools of error correcting codes adding redundancy to the problem's inputs in order to detect and possibly correct computational errors occurred in parallel-distributed environments.

ABFT techniques [HA84] are characterized by the *encoding* of inputs of the algorithm, the *redesign* of the algorithm to ensure that it can operate on the encoding data and the *distribution* of some computation steps among computational nodes. In this setting, errors are introduced by nodes and the error model strongly depends on the parallelization scheme.

In this thesis we study an ABFT technique for PLS solving by evaluation-interpolation [BK14, Per14, KPSW17]. Consider a square, non singular PLS $A(x)\mathbf{y}(x) = \mathbf{b}(x)$ and let $\mathbf{v}(x)/d(x)$ be its solution, where $\gcd(\gcd_i(v_i), d) = 1$ and d is monic. The evaluation-interpolation technique for PLS solving consists in the *evaluation* of $A(x)$ and $\mathbf{b}(x)$ at some distinct evaluation points $\{\alpha_1, \dots, \alpha_L\}$, the *pointwise resolution* of the evaluated systems $\mathbf{y}(\alpha_j) = A(\alpha_j)^{-1}\mathbf{b}(\alpha_j)$ and the *interpolation* and reconstruction of $\mathbf{y}(x) = \mathbf{v}(x)/d(x)$ given its evaluations.

The main idea of the ABFT technique applied to the evaluation-interpolation is to modify this method by the introduction of redundancy, considering more evaluation points compared to the number we need in the general case. In this scenario the *evaluation step* of the classic evaluation-interpolation technique is performed by different nodes (*parallelization*). These nodes may possibly introduce some errors and compute $\mathbf{y}(\alpha_j) \neq \mathbf{v}(\alpha_j)/d(\alpha_j)$.

The *polynomial linear system with errors* [BK14, Per14, KPSW17] is then the problem of recovering the vector of rational function \mathbf{v}/d , which is a solution of our PLS, given its evaluations where some could be erroneous. Also in this case the resolution technique of this problem can be seen as an SRFR.

Our contributions

Our contributions concern a uniqueness result on SRFR [GLZ20b] and some other results [GLZ19, GLZ20a] about the PLSwE and the more general problem of *simultaneous Cauchy interpolation with errors* (SCIwE).

On the generic uniqueness of SRFR. A first new contribution developed in this thesis is about the general SRFR problem.

Problem 2. *Simultaneous Rational Function Reconstruction*

Input: $a_1, \dots, a_n \in \mathbb{K}[x], \mathbf{u} = (u_1, \dots, u_n) \in \mathbb{K}[x]^n$, where $\deg(u_i) < \deg(a_i)$
and $1 \leq N_i \leq \deg(a_i), 1 \leq D \leq \min_{1 \leq i \leq n} \{\deg(a_i)\}$

Output: $(\mathbf{v}, d) = (v_1, \dots, v_n, d) \in \mathbb{K}[x]^{n+1}$ such that

$$[v_i = du_i \bmod a_i]_{1 \leq i \leq n}, \quad \deg(v_i) < N_i, \quad \deg(d) < D. \quad (1.5)$$

In Chapter 3 we prove (Theorem 3.2.1) that if

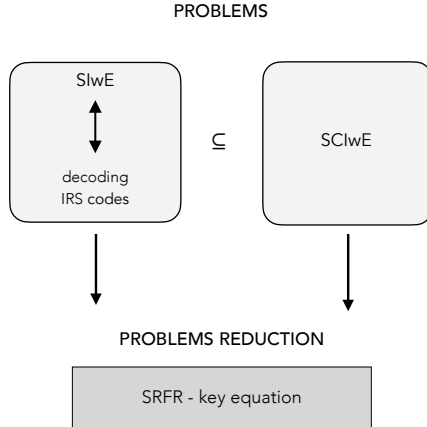
$$\sum_{i=1}^n \deg(a_i) = \sum_{i=1}^n N_i + D - 1,$$

then for almost all instances \mathbf{u} , SRFR admits a unique solution. Our approach to prove Theorem 3.2.1 consists in the study of the *row degrees* of a particular $\mathbb{K}[x]$ -module, the *relation module* related to a *specific matrix*. Indeed, we observe that solutions of SRFR are elements of this relation module with *negative shifted row degrees*, where the shifts are necessary to integrate the degree constraints. In the uniqueness case there is only one element of a basis of the relation module with negative row degrees. This represents a useful tool to check the uniqueness of SRFR. Indeed, in Theorem 3.2.1 we prove that for almost all instances (*generic instances*) there is only one generator with negative shifted row degree.

Before proving this theorem, we prove a general result about *generic row degrees* of relation modules related to general matrices (Corollary 3.1.2). Previous works studied generic row degrees of different $\mathbb{K}[x]$ -modules: *e.g.* the module of generating polynomials of a scalar matrix sequence [Vil97] or for the kernel of a polynomial matrix of specific dimensions [JV05]. The generic degrees also appear as dimensions of blocks of a shifted Hessenberg form [PS07]. However, the link with the degrees of a module is unclear. In all these cases any shift is considered. We prove our result for any shift and for any relation module also by re-laborating and adapting some of the techniques introduced in the articles mentioned above.

Simultaneous Cauchy interpolation with errors. The *simultaneous Cauchy interpolation with errors* (SCIwE) is the problem of recovering a vector of rational functions $\mathbf{v}(x)/d(x)$ of bounded degrees, given its evaluations where some could be erroneous. It is a more general problem than the PLSwE defined above.

In [GLZ19], following [Per14], we observe that this problem is the *rational extension* of the problem of decoding IRS codes. Indeed, in this case we want to recover a vector of *rational functions* instead of a vector of *polynomials*.



This link allows us to extend the same *interpolation-based* technique [BKY03] to this rational case, reducing the problem to an SRFR applied to instances \mathbf{u} of the form $\mathbf{u}(\alpha_j) = \mathbf{v}(\alpha_j)/d(\alpha_j) + \mathbf{e}_j$. Previous results about this topic [BK14, KPSW17], more related to the specific case of PLSwE, showed that with

$$L_{BK} = N + D - 1 + 2\tau \quad (7)$$

where $\tau \geq |E| := |\{j \mid \mathbf{e}_j \neq \mathbf{0}\}|$ is a bound on the number of errors, then we have uniqueness of SRFR. This number of points coincides with L_{RFR} , obtained by applying RFR component-wise. In [GLZ19], by stressing the similarity between this problem and the decoding of IRS codes we propose Algorithm 6 for SCIwE solving which reduces this number of points to

$$L_{GLZ1} = N + D - 1 + \tau + \lceil \tau/n \rceil. \quad (8)$$

However, as for the IRS case, since we are below the number which guarantees uniqueness of SRFR, this algorithm could fail for a few error patterns.

Polynomial Linear System Solving with Errors. The SCIwE is a general case of PLSwE, in which we want to recover a vector of rational functions with the same denominator, which is a solution of a PLS $A(x)\mathbf{y}(x) = \mathbf{b}(x)$. For this reason, in this case we can add the degrees of the coefficient matrix A and of the vector \mathbf{b} as additional inputs. So, in the same way as SCIwE also PLSwE can be reduced to an SRFR applied to instances of the form \mathbf{u} where $\mathbf{u}(\alpha_j) = \mathbf{v}(\alpha_j)/d(\alpha_j) + \mathbf{e}_j$ and $\mathbf{v}(\alpha_j)/d(\alpha_j)$ is a solution of a PLS $A(x)\mathbf{y}(x) = \mathbf{b}(x)$. Therefore, we have the same previous results about the number of evaluation points.

Besides in [KPSW17], Kaltofen *et al.* showed that with

$$L \geq \min\{L_{BK}, L_{KPSW}\} \quad (9)$$

evaluation points, where $L_{BK} = N + D - 1 + 2\tau$ (as in (7)) and

$$L_{KPSW} = \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + 2\tau, \quad (10)$$

then the corresponding SRFR has a unique solution.

In this thesis we present a result, derived from a work in progress [GLZ20a] which reduces this number to

$$L \geq \min\{L_{GLZ1}, L_{GLZ2}\} \quad (11)$$

evaluation points, where $L_{GLZ1} = N + D - 1 + \tau + \lceil \tau/n \rceil$ (as in (8)) and

$$L_{KPSW} = \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + \tau + \lceil \tau/n \rceil. \quad (12)$$

However, also in this case our algorithm (an extended version of Algorithm 6) can fail for a few fractions of errors.

Early termination techniques. We conclude by presenting the last contribution of this work, which is still a work in progress (some results are in [GLZ20a]).

Consider the PLSwE problem. We observe that all the number of evaluation points introduced so far $L_{BK}, L_{KPSW}, L_{GLZ1}, L_{GLZ2}$ strongly depends on the bounds on the degrees of the numerators and the denominator of our vector solution and on the bound on the number of errors τ . So, an overestimation of the real degrees and of the real number of errors (which we do not know a priori) could significantly increase the number of evaluation points compared to the number we really need. A classical strategy [KPSW17] to overcome this limit consists in performing an *early termination technique* which, starting from a *minimal value* of evaluation points, iteratively increment this number until a result is found. We point out that this technique aims at possibly decreasing the number of evaluation points, in order to speed up the computations.

In this thesis we also present an *early termination algorithm* (Algorithm 11) which *decrease* the number of evaluations to compared to previous results in [KPSW17].

Outline of the thesis

This dissertation is structured as follows.

- We start the first chapter with the introduction of the *rational function reconstruction* (Section 1.1) problem with a focus on the conditions of parameters which ensure uniqueness. We then generalize this problem to its vector version and we then present the main problem of this work: the *simultaneous rational function reconstruction* (Section 1.2). Then, in Section 1.3 we introduce some algebraic tools as $\mathbb{K}[x]$ -modules, *shifted row degrees*, *relation modules*, *reduced bases*, etc. which we use to check the uniqueness of SRFR. Finally in Section 1.4 we see how the *polynomial linear system solving* can be reduced to the SRFR problem.
- Chapter 2 is devoted to the introduction of basic *coding theory* notions (Section 2.1), especially focusing on the famous and widespread family of Reed-Solomon codes. In particular, we introduce some *decoding techniques* for RS codes and we see how these techniques reduce the decoding problem to a rational function reconstruction. Finally we move on to interleaved RS codes (Section 2.3), which are obtained by applying the *interleaving* technique on RS codes. This technique is applied in *burst* error settings. We then introduce a technique for IRS decoding and we see that this technique also consists in an SRFR.
- Chapter 3 is devoted to the presentation of our article [GLZ20b]. In detail, we present our result about the generic uniqueness of SRFR, under the assumption on parameters derived from the common denominator constraint. This chapter is divided in two parts. In the first section we present some general results about shifted row degrees of general relation modules. In the second part we transpose all these results in the specific case of SRFR.
- In Chapter 4 we focus on an *algorithm-based fault tolerant technique* (Section 4.1) for polynomial linear system solving by evaluation interpolation. Indeed, we see how this technique leads to the problem of recovering the solution of a PLS given its evaluations where some could be erroneous, which is the *polynomial linear system solving with errors* (PLSwE). This problem is an application of a more general problem, *i.e.* the *simultaneous Cauchy interpolation with errors* (SCIwE), which is a natural extension of SRFR to an error context.

SCIwE can be seen as an extension of the problem of decoding IRS codes to rational functions and we can reduce this problem to an SRFR. In this chapter we present our results of [GLZ19] and new ones, which come from a work in progress [GLZ20a]. We compare them to the current state of the art about this topic [BK14, Per14, KPSW17]. Finally in Section 4.2 we present a new *early termination algorithm* which possibly reduces the number of evaluation points needed to guarantee uniqueness of SRFR and

so to uniquely recover the solution of SCIwE.

A short summary of the main results of this document and further research tracks completes this document.

Publications

International conferences with proceedings

- **Polynomial Linear System Solving with Errors by Simultaneous Polynomial Reconstruction of Interleaved Reed-Solomon codes.** E. Guerrini, R. Lebreton, I. Zappatore. *In Proceedings of ISIT'19*, pages 1542-1546. IEEE, 2019.
- **On the Uniqueness of Simultaneous Rational Function Reconstruction.** E. Guerrini, R. Lebreton, I. Zappatore. *In Proceedings of ISSAC'20*. ACM, 2020

Preprints, work in progress

- **Enhancing Simultaneous Rational Function Recovery: adaptive error correction capability and new bounds for applications.** E. Guerrini, R. Lebreton, I. Zappatore. arXiv:2003.01793

CHAPTER 1

Simultaneous Rational Function Reconstruction

Contents

1.1 Rational Function Reconstruction	29
1.1.1 RFR by the Extended Euclidean Algorithm	32
1.2 Simultaneous Rational Function Reconstruction	33
1.3 The $\mathbb{K}[x]$-module of solutions of SRFR	35
1.3.1 Row degrees and reduced basis	38
1.3.2 Solutions of SRFR and Relation Module	42
1.4 Application to Polynomial Linear System Solving	45
1.5 A short summary of the chapter	48

Rational function reconstruction is a classic computer algebra problem, whose main aim is to reconstruct a rational function given its remainder modulo a polynomial. This problem can be straightforwardly generalized to its vector version, whose goal is to reconstruct a vector of rational functions. In this thesis we especially focus on the particular case of the vector rational function reconstruction in which all the rational functions share the same denominator: the *simultaneous rational function reconstruction* problem.

In this chapter, we present the current state of the art of these problems. All the results of this chapter have been reinterpreted with a focus on the condition on the parameters which guarantees the uniqueness of the solution. Indeed, we explain how the common denominator assumption impacts that condition, with interesting consequences especially from an application point of view.


1.1 Rational Function Reconstruction

In this section we define the *rational function reconstruction*, *i.e.* the problem of reconstructing a rational function whose numerator and denominator degrees are bounded, given its remainder modulo a polynomial. We also introduce the notion of *uniqueness* of the solution, which is crucial for this work, explaining the condition that guarantees such a property.

Finally, in the uniqueness case we explain how this problem can be solved using the *Extended Euclidean Algorithm*.

Definition 1.1.1 (Rational function reconstruction). Let \mathbb{K} be a field, $a, u \in \mathbb{K}[x]$ such that $\deg(u) < \deg(a)$ and $1 \leq N, D \leq \deg(a)$. The rational function reconstruction (RFR) is the problem of finding a couple of polynomials $(v, d) \in \mathbb{K}[x]^2$ such that,

$$\gcd(d, a) = 1 \text{ and } \frac{v}{d} = u \pmod{a}, \quad \deg(v) < N, \quad \deg(d) < D. \quad (1.1)$$

Remark 1.1.1. To be rigorous, we observe that in (1.1) we consider the unique monic gcd of $d(x)$ and $a(x)$. 

This problem is also known as:

- *Padé approximation*: if $a = x^L$,
- *Cauchy interpolation*: if $a = \prod_{i=1}^L (x - \alpha_i)$, where the $\{\alpha_1, \dots, \alpha_L\}$ are $L \geq 1$ pairwise distinct elements of the field \mathbb{K} . Note that in this case, by the Chinese Remainder Theorem, the equation (1.1) becomes an equation on the *evaluations*

$$d(\alpha_i) \neq 0 \text{ and } \frac{v(\alpha_i)}{d(\alpha_i)} = u(\alpha_i), \quad \deg(v) < N, \quad \deg(d) < D. \quad (1.2)$$

for any $1 \leq i \leq L$.

- *Rational Hermite interpolation*: if $a = \prod_{i=1}^L (x - \alpha_i)^{e_i}$, where the $\{\alpha_1, \dots, \alpha_L\}$ are pairwise distinct elements of the field \mathbb{K} and the e_i are nonnegative integers, called *multiplicities*.


Let us assume that

$$\deg(a) = N + D - 1. \quad (1.3)$$

In this case, equation (1.1) may have no solution, as shown in the next example.

Example 1.1.1. Let $a = x^3$, $u = x^2 - 1$ and $N = D = 2$. We want to reconstruct (v, d) which satisfies (1.1). Let $d = d_1x + d_0$. Since d has to be a unit modulo a , the constant term $d_0 \neq 0$. Hence

$$v = (d_1x + d_0)(x^2 - 1) = d_0x^2 - d_1x - d_0 \pmod{x^3}$$

and $\deg(v) = 2$ which contradicts the degree constraint $\deg(v) < 2 = N$. Hence in this case, equation (1.1) does not admit a nonzero solution. 

Nevertheless, under assumption (1.3), if a solution exists, it is *unique*, *i.e.* the corresponding rational function is unique. In detail, if there exist (v_1, d_1) and (v_2, d_2) , both solutions of (1.1) then


$$\frac{v_1}{d_1} = \frac{v_2}{d_2} \pmod{a} \implies a \mid (v_1d_2 - v_2d_1).$$

Since the degree of the polynomial $v_1d_2 - v_2d_1$ is strictly smaller than the degree of a , it is the zero polynomial. Therefore $\frac{v_1}{d_1} = \frac{v_2}{d_2}$.

Remark 1.1.2. We observe that:

1. if $\deg(a) \geq N+D-1$ we still have the uniqueness of the rational solution. Indeed, if there exist (v_1, d_1) and (v_2, d_2) solutions of (1.1) then the degree of $v_1d_2 - v_2d_1$ is smaller than the degree of a , which implies the equality of the two corresponding rational functions;
2. if $\deg(a) < N+D-1$, the uniqueness is not always guaranteed, as shown in the following example.




Example 1.1.2. Let $\mathbb{K} = \mathbb{F}_{11}$, $a = x^4 + 3x^3 + 5x^2 + 8x + 1$, $u = 6x^3 + 7x^2 + 4x + 2$ and $N = 3$, $D = 3$. Note that $\deg(a) = 4 < N + D - 1 = 5$. Both $(v_1, d_1) = (4x + 7, 7x^2 + x + 1)$ and $(v_2, d_2) = (8x^2 + 4x + 1, 9x)$ are solutions of (1.1) and their corresponding rational functions are different. 

We now observe that if $(v, d) \in \mathbb{K}[x]^2$ is a solution of the RFR problem (1.1), it is also a solution of the *weaker* problem,

$$v = du \pmod{a}, \quad \deg(v) < N, \quad \deg(d) < D. \quad (1.4)$$

Unlike (1.1), if (1.3) holds, the weaker problem (1.4) always admits a nonzero solution $(v, d) \in \mathbb{K}[x]^2$. Indeed, a solution of (1.4) belongs to the right kernel of the homogeneous linear system associated to the equation (1.4). This homogeneous linear system has $\deg(a)$ equations and $N + D$ unknowns, which are the coefficients of the polynomials of the solution that we want to recover. Since $\deg(a) = N + D - 1$, by the *Rank-Nullity Theorem*, the dimension of the right kernel is at least 1. Hence it is nontrivial, meaning that (1.4) always admits a nonzero solution (v, d) . Furthermore, the uniqueness of the rational solution is still guaranteed: let (v_1, d_1) and (v_2, d_2) be solutions of (1.4), then a divides $v_1 - d_1u$ and $v_2 - d_2u$ and so it divides $d_2(v_1 - d_1u) - d_1(v_2 - d_2u) = v_1d_2 - v_2d_1$. With the same argument as before, we can prove that the polynomial $v_1d_2 - v_2d_1$ is zero. This implies that any solution of (1.4) is a polynomial multiple of a *minimal* solution and we will see later (Theorem 1.1.1) how to compute such a solution. Therefore, the solution space is a subset of a free $\mathbb{K}[x]$ -module of rank 1 (see Definition 1.3.1).

Remark 1.1.3. If $\deg(a) < N + D - 1$ the uniqueness of the rational solution is not anymore guaranteed. Indeed by the Rank-Nullity Theorem, the dimension of the right kernel is greater than 2, meaning that there could exist two linearly independent solutions of the problem (1.4). 

The weaker problem (1.4) is easier to study for its linearity. For this reason, we refer to RFR as this version of the problem,

Problem 1. *Rational Function Reconstruction*

Input: $a, u \in \mathbb{K}[x]$, with $\deg(u) < \deg(a)$ and $1 \leq N, D \leq \deg(a)$

Output: $(v, d) \in \mathbb{K}[x]^2$ such that $v = du \pmod{a}$, $\deg(v) < N$, $\deg(d) < D$.

1.1.1 RFR by the Extended Euclidean Algorithm

From now on we suppose that (1.3) holds, *i.e.* RFR always admits a unique solution. A classic approach to solve the RFR problem consists in using the *Extended Euclidean Algorithm* (EEA) (Algorithm 1). Recall that, given two polynomials $f, g \in \mathbb{K}[x]$, EEA returns a $\gcd(f, g)$ and the *Bézout coefficients* for f, g , *i.e.* the polynomials $s, t \in \mathbb{K}[x]$ such that $sf + tg = \gcd(f, g)$.

Algorithm 1: Extended Euclidean Algorithm, $\text{EEA}(f, g)$

Input : $f, g \in \mathbb{K}[x]$

Output: $\gcd(f, g) = r$ and the Bézout coefficients $s, t \in \mathbb{K}[x]$ for f, g .

```

1  $r_0 \leftarrow f; s_0 \leftarrow 1; t_0 \leftarrow 0; r_1 \leftarrow g; s_1 \leftarrow 0; t_1 \leftarrow 1; i \leftarrow 1;$ 
2 while  $r_i \neq 0$  do
3    $q_i \leftarrow r_{i-1} \text{ quo } r_i;$ 
4    $r_{i+1} \leftarrow r_{i-1} - q_i r_i;$ 
5    $s_{i+1} \leftarrow s_{i-1} - q_i s_i;$ 
6    $t_{i+1} \leftarrow t_{i-1} - q_i t_i;$ 
7    $i \rightarrow i + 1;$ 
8  $r \leftarrow r_{i-1}; s \leftarrow s_{i-1}; t \leftarrow t_{i-1};$ 
9 return  $r, s, t$ 

```

An important property of this algorithm is that the i -th step results s_i, t_i, r_i satisfy $s_i f + t_i g = r_i$. This is useful for many computer algebra applications as RFR.

We can now underline the link between the rational function reconstruction and EEA observing that the congruence of (1.4) implies the existence of a polynomial e such that $v = du + ea$. In detail, the following theorem shows that some intermediate results of the Extended Euclidean Algorithm are solutions of RFR.

Theorem 1.1.1. *Let a, u, N, D be the inputs of the RFR problem. Let r_j, s_j, t_j be the output of the j -th step of EEA with input a, u , where j is the smallest integer such that $\deg(r_j) < N$. Then, (r_j, t_j) is a solution of RFR and if $\gcd(r_j, t_j) = 1$ it is also a solution of (1.1). Moreover (r_j, t_j) is the minimal solution, *i.e.* any other solution is a polynomial multiple of this one.*

Proof. We first observe that $r_j = s_j a + t_j u = t_j u \pmod{a}$. By [GG13, Lemma 3.15(b)], by the minimality of j and since $\deg(a) = N + D - 1$ (equation (1.3)), we can conclude that $\deg(t_j) = \deg(a) - \deg(r_{j-1}) \leq N + D - 1 - N = D - 1$ and so (r_j, v_j) is a solution of RFR. Moreover, since $\gcd(r_j, t_j) = \gcd(a, t_j)$, if r_j and t_j are coprime then t_j is invertible modulo a and so (r_j, t_j) is a solution of (1.1).

We now prove the minimality of such a solution. Let (v, d) be a solution of RFR, then there exists $e \in \mathbb{K}[x]$ such that $v = du + ea$. On the other hand, also (r_j, t_j) is a solution of RFR and $r_j = s_j a + t_j u$. So, since $\deg(a) = N + D - 1$ (1.3), then

$$0 = vt_j - r_j d = (du + ea)t_j - (s_j a + t_j u)d = eat_j - s_j ad.$$

Now, t_j divides $s_j d$ and since $\gcd(t_j, s_j) = 1$, then t_j divides d . So, there exists $p \in \mathbb{K}[x]$ such that $d = pt_j$.

Recall that $vt_j = r_j d$ and so by replacing d we get $vt_j = r_j pt_j$ and we can conclude that $v = pr_j$. \square

In conclusion, we can derive the following Algorithm based on EEA to solve RFR using $\mathcal{O}(\deg(a)^2)$ arithmetic operations in the field \mathbb{K} ([GG13, Theorem 3.16]). There are some other strategies based on the *half-gcd* computation, which improve the complexity of the EEA to $\mathcal{O}(M(\deg(a)) \log(\deg(a)))$ arithmetic operations (see [BCG⁺17, Section 6.3]), where $M(t)$ is the classic polynomial multiplication complexity ([BCG⁺17, Section 2.7]).

Algorithm 2: Rational function reconstruction by EEA, $\text{RFR}_{\text{EEA}}(a, u, N)$

Input : a, u, N instances of RFR (Problem 1)

Output: (v, d) the minimal degree solution of RFR

```

1  $r_0 \leftarrow a; s_0 \leftarrow 1; t_0 \leftarrow 0; r_1 \leftarrow u; s_1 \leftarrow 0; t_1 \leftarrow 1; i \leftarrow 1;$ 
2 while  $\deg(r_i) \geq N$  do
3    $q_i \leftarrow r_{i-1} \text{ quo } r_i;$ 
4    $r_{i+1} \leftarrow r_{i-1} - q_i r_i;$ 
5    $s_{i+1} \leftarrow s_{i-1} - q_i s_i;$ 
6    $t_{i+1} \leftarrow t_{i-1} - q_i t_i;$ 
7    $i \rightarrow i + 1;$ 
8 return  $r_i, t_i$ 

```

1.2 Simultaneous Rational Function Reconstruction

The rational function reconstruction problem can be straightforwardly generalized to its vector version, in which we want to reconstruct a vector of rational functions. In this section we introduce the *simultaneous rational function reconstruction*, *i.e.* the problem of reconstructing a *vector* of rational functions with the same denominator, given their remainders

modulo some polynomials. We explain how the common denominator feature can be used to reduce the number of unknowns of the related linear system and how this effect the uniqueness of the reconstructed solution.

Formally, given $a_1, \dots, a_n \in \mathbb{K}[x]$, $\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{K}[x]^{1 \times n}$ where $\deg(u_i) < \deg(a_i)$ and the degree constraints $1 \leq N_i, D_i \leq \deg(a_i)$, the *vector rational function reconstruction* (VRFR) is the problem of reconstructing (v_i, d_i) for $1 \leq i \leq n$ such that

$$v_i = d_i u_i \bmod a_i, \quad \deg(v_i) < N_i, \quad \deg(d_i) < D.$$

In order to solve this problem we can apply RFR componentwise and as we saw in the previous section, if $\deg(a_i) = N_i + D_i - 1$ we can uniquely reconstruct the solution.

In this thesis, we focus on the following specific case:

Problem 2. *Simultaneous Rational Function Reconstruction*

Input: $a_1, \dots, a_n \in \mathbb{K}[x]$, $\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{K}[x]^{1 \times n}$, where $\deg(u_i) < \deg(a_i)$ and $1 \leq N_i \leq \deg(a_i)$, $1 \leq D \leq \min_{1 \leq i \leq n} \{\deg(a_i)\}$

Output: $(\mathbf{v}, d) = (v_1 \dots, v_n, d) \in \mathbb{K}[x]^{1 \times (n+1)}$ such that

$$[v_i = d u_i \bmod a_i]_{1 \leq i \leq n}, \quad \deg(v_i) < N_i, \quad \deg(d) < D. \quad (1.5)$$

Since solutions of SRFR are solutions of VRFR, then SRFR has a unique solution (if it exists) whenever

$$\deg(a_i) = N_i + D - 1, \quad \text{for any } 1 \leq i \leq n. \quad (1.6)$$


We observe that in this case, the unknowns of the homogeneous linear system related to (1.5) are the coefficients of any polynomial v_i and of the polynomial d . So, the number of unknowns is $\sum_{i=1}^n N_i + D$ while the number of equations is $\sum_{i=1}^n \deg(a_i)$. If

$$\sum_{i=1}^n \deg(a_i) = \sum_{i=1}^n N_i + D - 1 \quad (1.7)$$

SRFR always admits a nonzero solution by the Rank-Nullity Theorem. However, the uniqueness is not always guaranteed as shown in the following example.

Example 1.2.1. Let $\mathbb{K} = \mathbb{F}_{17}$, $n = 2$, $N = 4$, $D = 5$, $a_1 = a_2 = \prod_{i=1}^6 (x - 3^i) = x^6 + 13x^5 + x^4 + 15x^3 + 11x^2 + 9x + 5$, where $\deg(a) = N + \frac{D-1}{n} = 6$ and $\mathbf{u} = (11x^5 + 3x^4 + 3x^3 + 4x^2 + 9x + 5, 4x^4 + 14x^3 + 8x^2 + 7x + 14)$. Then, both

$$\begin{aligned} (\mathbf{v}_1, d_1) &= (v_{1,1}, v_{1,2}, d) = (13x^3 + 2x^2 + 14x + 12, 6x^3 + 3x^2 + 11x + 12, x^3 + 5x^2 + 8x + 9) \\ (\mathbf{v}_2, d_2) &= (v_{2,1}, v_{2,2}, d) = (16x^3 + 12x^2 + 4x + 12, 10x^2 + 4x + 6, 8x^4 + 4x^2 + 11x + 1) \end{aligned}$$

are solutions of SRFR. 

Remark 1.2.1. We now remark that SRFR decreases the number of equations of the corresponding homogeneous linear system up to a factor 2, compared to VRFR or RFR. Indeed, consider for simplicity $N_1 = \dots = N_n = N$ and $a_1 = \dots = a_n = a$ with $\deg(a) = L$. Recall that if we solve SRFR by applying RFR componentwise (1.3), if

$$L = N + D - 1 =: L_{RFR}$$

we have the uniqueness. On the other hand by (1.7), if

$$L = N + (D - 1)/n =: L_{SRFR}$$

then SRFR admits a nontrivial solution, but the uniqueness is not anymore guaranteed. We claim that


$$L_{SRFR}/L_{RFR} \geq 1/2 \tag{1.8}$$

First we observe that

$$L_{SRFR}/L_{RFR} \geq 1/2 \iff (n - 2)(D - 1) \leq nN. \tag{1.9}$$

Then, by our assumptions $D - 1 < L_{SRFR}$ (see Problem 2). So, $n(D - 1) < nL_{SRFR} = nN + (D - 1)$ and since

$$(n - 1)(D - 1) < nN$$

we can deduce that $(n - 2)(D - 1) \leq nN$ and so (1.8) holds. 

Let \mathbf{u} be an instance of the SRFR problem. We denote by $\mathcal{S}_{\mathbf{u}}$ the set of solutions. We also denote by s the rank of the $\mathbb{K}[x]$ -module spanned by the solutions in $\mathcal{S}_{\mathbf{u}}$. In other terms, all the solutions can be written as a linear combination of s vectors of polynomials with polynomial coefficients. Note that the case $s = 1$ corresponds to the uniqueness case.

In the following section we will see that there is a specific way to find these solutions: they are generated by *some* rows of a *shifted row-reduced* basis (Definition 1.3.4) of a particular $\mathbb{K}[x]$ -module, *i.e.* the *relation module* (see Subsection 1.3.2).

1.3 The $\mathbb{K}[x]$ -module of solutions of SRFR

In this section we explain the link between solutions of SRFR and the *relation module*. For this purpose, we introduce useful definitions and recall some results about specific bases of free $\mathbb{K}[x]$ -modules, the *shifted reduced basis*. We refer to [Nei16] for all notions and historical references.

Preliminaries about $\mathbb{K}[x]$ -modules. We start by briefly recalling the definition of a module [DF04, Section 10.1], which is the generalization to rings of the notion of vector space.

Let R be a commutative ring with identity.

Definition 1.3.1 (R -module). An R -module is a set \mathcal{N} together with

1. a binary operation $+$ on \mathcal{N} , under which \mathcal{N} is an abelian group, and
2. a map $R \times \mathcal{N} \longrightarrow \mathcal{N}$ denoted by rn , for all $r \in R$ and for all $n \in \mathcal{N}$ which satisfies
 - (a) $(r + s)n = rn + sn$, for all $r, s \in R, n \in \mathcal{N}$,
 - (b) $(rs)n = r(sn)$, for all $r, s \in R, n \in \mathcal{N}$,
 - (c) $r(m + n) = rm + rn$, for all $r \in R, m, n \in \mathcal{N}$,
 - (d) if 1 is the multiplicative identity in R , $1n = n$ for all $n \in \mathcal{N}$.

An \mathcal{R} -module \mathcal{N} is *free* if it admits a *basis*, *i.e.* a set of linearly independent generators of \mathcal{N} . The *rank* of \mathcal{N} is the cardinality of such a basis. If b_1, \dots, b_r is a basis of \mathcal{N} , we denote $\mathcal{N} := \langle b_1, \dots, b_r \rangle$.

Definition 1.3.2 (R -submodule). Let \mathcal{N} be an R -module. An R -submodule of \mathcal{N} is a subgroup \mathcal{M} of \mathcal{N} such that $rm \in \mathcal{M}$ for all $r \in R$ and $m \in \mathcal{M}$.

Contrary to vector spaces over a field, not every module has a basis. Besides, modules over *Principal Ideal Domains* (PID) have some important properties, as stated by the following two results.

Lemma 1.3.1. *Let \mathcal{R} be a PID, let \mathcal{N} be a free \mathcal{R} -module of finite rank ν and let \mathcal{M} be a submodule of \mathcal{N} . Then,*

1. \mathcal{M} is free of rank $\mu \leq \nu$,
2. *there exists a basis y_1, \dots, y_ν of \mathcal{N} so that $a_1 y_1, \dots, a_\mu y_\mu$ is a basis of \mathcal{M} , where a_1, \dots, a_μ are nonzero elements of \mathcal{R} with the divisibility relations*

$$a_1 | a_2 | \dots | a_\mu.$$

Proof. For the proof of this result we refer the reader to [DF04, Theorem 4, Section 12.1]. \square

Theorem 1.3.2 (Invariant Factor Form [DF04, Theorem 5, Section 12.1]). *Let \mathcal{R} be a PID and \mathcal{N} be a finitely generated \mathcal{R} -module. Then \mathcal{N} is isomorphic to the direct sum of finitely many cyclic modules. More precisely,*

$$\mathcal{N} \simeq \mathcal{R}^\rho \oplus \mathcal{R}/(a_1) \oplus \dots \oplus \mathcal{R}/(a_\mu)$$

for some integer $\rho \geq 0$ and nonzero elements $a_1, \dots, a_\mu \in \mathcal{R}$ which are not units in \mathcal{R} and which satisfy the divisibility relations

$$a_1 | a_2 | \dots | a_\mu.$$

Proof. Let ν be the rank of \mathcal{N} and x_1, \dots, x_ν be its basis. Consider the \mathcal{R} -module \mathcal{R}^ν of rank ν and basis b_1, \dots, b_ν and define the homomorphism $\pi : \mathcal{R}^\nu \rightarrow \mathcal{N}$ such that $b_i \mapsto x_i$ for all i . This homomorphism is surjective since x_1, \dots, x_ν is a basis of \mathcal{N} . So, by the *First Isomorphism Theorem* for modules [DF04, Theorem 4, Section 10.2], we have that $\mathcal{R}^\nu / \ker(\pi) \simeq \mathcal{N}$. Now, by Lemma 1.3.1, there exists a basis y_1, \dots, y_ν such that $a_1 y_1, \dots, a_\mu y_\mu$ is a basis of $\ker(\pi)$, for some nonzero elements $a_1, \dots, a_\mu \in \mathcal{R}$ with $a_1 | \dots | a_\mu$. Therefore,

$$\mathcal{N} \simeq \mathcal{R}^\nu / \ker(\pi) = (Ry_1 \oplus \dots \oplus Ry_\nu) / (Ra_1 y_1 \oplus \dots \oplus Ra_\mu y_\mu).$$

Now consider the map

$$\begin{aligned} \theta : Ry_1 \oplus \dots \oplus Ry_\nu &\longrightarrow \mathcal{R}/(a_1) \oplus \dots \oplus \mathcal{R}/(a_\mu) \oplus \mathcal{R}^{\nu-\mu} \\ (\alpha_1 y_1, \dots, \alpha_\nu y_\nu) &\longmapsto (\alpha_1 \bmod (a_1), \dots, \alpha_\mu \bmod (a_\mu), \alpha_{\mu+1}, \dots, \alpha_\nu). \end{aligned}$$

This is a surjective \mathcal{R} -module homomorphism. Note that $\ker(\theta) = Ra_1 y_1 \oplus \dots \oplus Ra_\mu y_\mu$ and so, by the First Isomorphism Theorem applied to θ we get

$$\mathcal{N} \simeq \mathcal{R}/(a_1) \oplus \dots \oplus \mathcal{R}/(a_\mu) \oplus \mathcal{R}^{\nu-\mu}.$$

Finally we remark that if a is a unit in \mathcal{R} , then the quotient $\mathcal{R}/(a) = 0$ and so in the direct sum we may remove any of the a_i which are units. \square

Remark 1.3.1 (Uniqueness of the Invariant Factor Form). By the divisibility condition, the decomposition of Theorem 1.3.2 is *unique* (see [DF04, Section 12.1]), *i.e.* if we have

$$\mathcal{N} \simeq \mathcal{R}^{\rho'} \oplus \mathcal{R}/(b_1) \oplus \dots \oplus \mathcal{R}/(b_{\mu'})$$

for some integer $\rho' \geq 0$ and nonzero $b_1, \dots, b_{\mu'}$ which are not units of \mathcal{R} and with $b_1 | \dots | b_{\mu'}$, then $\rho' = \rho$, $\mu = \mu'$ and $(a_i) = (b_i)$ for all i .

Moreover, the elements a_1, \dots, a_μ , defined up to multiplication by units in \mathcal{R} , are called *invariants* of the \mathcal{R} -module \mathcal{N} . \spadesuit

In this thesis we focus on modules over the polynomial ring $\mathcal{R} = \mathbb{K}[x]$. Note that, by the previous lemma, if \mathcal{M} is a $\mathbb{K}[x]$ -submodule of $\mathbb{K}[x]^\nu := \mathbb{K}[x]^{1 \times \nu}$, it is free of rank $\mu \leq \nu$. Any basis P of \mathcal{M} can be represented by a $\mu \times \nu$ matrix of polynomials over $\mathbb{K}[x]$ whose rows $P_{j,*}$ are the elements of the basis, hence $P \in \mathbb{K}[x]^{\mu \times \nu}$. We observe that, in this notation, \mathcal{M} coincides with the *row space* of P , *i.e.* $\mathcal{M} = \mathbb{K}[x]^{1 \times \mu} P = \{\lambda P \mid \lambda \in \mathbb{K}[x]^{1 \times \mu}\}$.

As it turns out, basis of $\mathbb{K}[x]$ -modules are related through *unimodular* transformations.

Recall that a square matrix $U \in \mathbb{K}[x]^{\mu \times \mu}$ is unimodular if its determinant is a nonzero element of the field \mathbb{K} .

Lemma 1.3.3. *Let P, R be two bases of \mathcal{M} , a $\mathbb{K}[x]$ -submodule of $\mathbb{K}[x]^\nu$ of rank $\mu \leq \nu$. Then P and R are unimodularly equivalents, i.e. there exist $U, V \in \mathbb{K}[x]^{\mu \times \mu}$ unimodular matrices such that $P = UR$ and $R = VP$.*

Proof. We observe that each row of P belongs to the row space of R and *vice versa*. So there exist $U, V \in \mathbb{K}[x]^{\mu \times \mu}$ such that $P = UR$ and $R = VP$. Now, $P = UVP$ which implies that $(I_\mu - UV)P = \mathbf{0}$, where I_μ is the identity matrix. Since by definition the rows of P are linearly independent, $I_\mu - UV = \mathbf{0}$ and so $\det(U)\det(V) = 1$. We can then conclude that both U, V are unimodular. \square

1.3.1 Row degrees and reduced basis

We start by defining the *shifted row degree* which extends the notion of degrees for polynomial row vectors.

Definition 1.3.3 (Shifted row degree). Let $\mathbf{p} = (p_1, \dots, p_\nu) \in \mathbb{K}[x]^{1 \times \nu}$ be a row vector, $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$ a *shift* and $r_i := \deg(p_i) + s_i$ for $1 \leq i \leq \nu$. The \mathbf{s} -row degree of \mathbf{p} is

$$\text{rdeg}_{\mathbf{s}}(\mathbf{p}) = \max_{1 \leq i \leq \nu} (r_i).$$

We denote $\mathbf{p} = ([r_1]_{s_1}, \dots, [r_\nu]_{s_\nu})$ to stress the shifted degrees of \mathbf{p} .

If $\mathbf{s} = (0, \dots, 0)$ is the *uniform shift*, the shifted row degree is simply called *row degree*.

This definition of shifted row degree is equivalent, up to change of sign, to the notion of *defect* introduced in [BL94, Definition 3.1] which is sometimes used in the literature ([OS07]).

Remark 1.3.2. The shifted row degree is particularly useful for the degree constraints representation. For instance, consider a row vector $\mathbf{p} = (p_1, \dots, p_\nu)$ and the degree constraints $\mathbf{n} = (n_1, \dots, n_\nu) \in \mathbb{Z}_{>0}^\nu$, then


$$\deg(p_i) < n_i, \quad 1 \leq i \leq \nu \iff \text{rdeg}_{-\mathbf{n}}(\mathbf{p}) < 0. \quad (1.10)$$

💡

We can easily extend Definition 1.3.3 to polynomial matrices: let $P \in \mathbb{K}[x]^{\mu \times \nu}$ and $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$, then the \mathbf{s} -row degrees of P are $\text{rdeg}_{\mathbf{s}}(P) = (r_1, \dots, r_\mu)$ where $r_i := \text{rdeg}_{\mathbf{s}}(P_{i,*})$.

Example 1.3.1. Let $P = \begin{pmatrix} x^2 + 1 & x \\ 2x^3 + x^2 & 3x^2 + 1 \end{pmatrix} \in \mathbb{F}_7[x]^{2 \times 2}$ and $\mathbf{s} = (1, 0)$. Using the notation of the Definition 1.3.3, we get

$$P = \begin{pmatrix} [3]_1 & [1]_0 \\ [4]_1 & [2]_0 \end{pmatrix}$$

Note that in this case $\text{rdeg}_{\mathbf{s}}(P) = (3, 4)$ 

In this work we are interested in some particular bases of $\mathbb{K}[x]$ -modules with minimal shifted row degrees.

Definition 1.3.4 (Shifted row reduced form). Let $\mu \leq \nu$, $P \in \mathbb{K}[x]^{\mu \times \nu}$ be a full rank polynomial matrix and $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$ a shift. Then P is \mathbf{s} -row reduced if $\text{rdeg}_{\mathbf{s}}(P) \leq \text{rdeg}_{\mathbf{s}}(UP)$ for all unimodular matrices $U \in \mathbb{K}[x]^{\mu \times \mu}$. In this last inequality the row degrees are sorted in non-decreasing order and then lexicographically compared.

Again, if we consider the uniform shift we can say that the polynomial matrix is simply *row reduced*.

Note that since all the bases are unimodularly equivalents, then the shifted row reduced bases have minimal shifted row degrees. Furthermore, this notion is invariant under permutations of the rows and we can conclude that all shifted row reduced basis have the same row degree up to permutation.

Given $\mathbf{t} = (t_1, \dots, t_\nu) \in \mathbb{Z}^\nu$, we denote by $X^{\mathbf{t}}$ the diagonal matrix whose entries are the monomials $x^{t_1}, \dots, x^{t_\nu}$.

Definition 1.3.5 (Shifted Leading Matrix). Let $\mu \leq \nu$, $P \in \mathbb{K}[x]^{\mu \times \nu}$ a full rank polynomial matrix and $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$ a shift. The \mathbf{s} -leading matrix of P , $LM_{\mathbf{s}}(P)$, is the matrix in $\mathbb{K}[x]^{\mu \times \nu}$ whose entries are the coefficients of degree zero of $X^{-\text{rdeg}_{\mathbf{s}}(P)} P X^{\mathbf{s}}$.

The following lemma gives us a more practical method to verify if a basis is \mathbf{s} -row reduced.

Lemma 1.3.4 ([Nei16, Theorem 1.11]). *Let $\mu \leq \nu$, $P \in \mathbb{K}[x]^{\mu \times \nu}$ be a full rank polynomial matrix and $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$ a shift. Then P is \mathbf{s} -row reduced if and only if the \mathbf{s} -leading matrix of P $LM_{\mathbf{s}}(P)$ has full rank.*

Example 1.3.2. Let $\mathbb{K} = \mathbb{F}_7$, $\mathbf{s} = (0, 2, 1)$ and


$$P = \begin{pmatrix} 2x^4 + 6x^3 + x^2 + 3x + 1 & 5x + 1 & 6x^2 + 3x \\ 2x^3 + 6x^2 + 3 & 6x + 2 & 5x + 1 \\ 4x^5 + 5x^4 + 2x^3 + 3x^2 & 2x^3 + 3x^2 + 3x + 5 & 4x^4 + 2x^3 + x + 1 \end{pmatrix}$$

Using the notation of Definition 1.3.3,

$$P = \begin{pmatrix} [4]_0 & [3]_2 & [3]_1 \\ [3]_0 & [3]_2 & [2]_1 \\ [5]_0 & [5]_2 & [5]_1 \end{pmatrix}$$

The \mathbf{s} -leading matrix of P is

$$LM_{\mathbf{s}}(P) = \begin{pmatrix} 2 & 0 & 0 \\ 2 & 6 & 0 \\ 4 & 2 & 4 \end{pmatrix}$$

which is full rank, hence the basis P is \mathbf{s} -row reduced. 

We can now introduce an important property of shifted row reduced bases which is central for this work.

Proposition 1.3.5 (Predictable degree property). *Fix a shift $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$. A polynomial matrix $P \in \mathbb{K}[x]^{\mu \times \nu}$ is \mathbf{s} -row reduced if and only if for all $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_\mu) \in \mathbb{K}[x]^{1 \times \mu}$,*

$$rdeg_{\mathbf{s}}(\boldsymbol{\lambda}P) = \max_{1 \leq i \leq \mu} \{\deg(\lambda_i) + rdeg_{\mathbf{s}}(P_{i,*})\} = rdeg_{\mathbf{d}}(\boldsymbol{\lambda})$$

where $\mathbf{d} = rdeg_{\mathbf{s}}(P)$.

The proof of this classic proposition can be found in [Nei16, Theorem 1.11]. Note that if $P \in \mathbb{K}[x]^{\mu \times \nu}$ is an \mathbf{s} -row reduced basis of a $\mathbb{K}[x]$ -module \mathcal{M} , the latter proposition implies that any nonzero element $\mathbf{p} \in \mathcal{M}$ has $rdeg_{\mathbf{s}}(\mathbf{p})$ greater than $\min_{1 \leq i \leq \mu} \{rdeg_{\mathbf{s}}(P)\}$.

Remark 1.3.3. Consider a shift $\mathbf{s} \in \mathbb{Z}^\nu$, $t \in \mathbb{Z}$ and the \mathbb{K} -vector space¹ $\mathcal{M}_{<t} := \{\mathbf{p} \in \mathcal{M} \mid rdeg_{\mathbf{s}}(\mathbf{p}) < t\}$. Let R be an \mathbf{s} -row reduced basis of \mathcal{M} with $rdeg_{\mathbf{s}}(R) = (r_1, \dots, r_\mu)$. Note that by Proposition 1.3.5, elements $\mathbf{p} \in \mathcal{M}_{<t}$ are the linear combination (with polynomial coefficients) of the rows $R_{i,*}$ with $rdeg_{\mathbf{s}}(R_{i,*}) = r_i < t$ and so,

$$\dim(\mathcal{M}_{<t}) = \sum_{r_i < t} (t - r_i). \quad (1.11)$$

We now consider the case of the Remark 1.3.2 in which we want to recover an element \mathbf{p} of a $\mathbb{K}[x]$ -module whose degrees are bounded by $\mathbf{n} = (n_1, \dots, n_\nu)$. Then we can consider the shift $\mathbf{s} = -\mathbf{n}$. By the previous considerations $t = 0$ and we get that

$$\dim(\mathcal{M}_{<0}) = \sum_{r_i < 0} (-r_i) \quad (1.12)$$

where (r_1, \dots, r_μ) are the row degrees of any $-\mathbf{n}$ -row reduced basis of \mathcal{M} . 

As already remarked, shifted reduced basis are invariant under permutation of the rows. Nevertheless, in this thesis, we need to define shifted row degrees *uniquely* and not just up to a permutation. For this reason we introduce shifted *ordered weak Popov bases* which are special reduced bases, based on the notion of *pivot*.

1. Notice that to lighten the notations we omit the shift dependency of the \mathbb{K} -vector space $\mathcal{M}_{<t}$.

Definition 1.3.6 (Pivot). Let $\mathbf{p} \in \mathbb{K}[x]^{1 \times \nu}$ and $\mathbf{s} \in \mathbb{Z}^\nu$ be a shift. The \mathbf{s} -pivot index of \mathbf{p} is $\max\{1 \leq i \leq \nu \mid \text{rdeg}_{\mathbf{s}}(\mathbf{p}) = \deg(p_i) + s_i\}$. The corresponding p_i is the \mathbf{s} -pivot entry and $\deg(p_i)$ is the \mathbf{s} -pivot degree of \mathbf{p} .

We can naturally extend the notion of pivot to polynomial matrices. Indeed, consider $\mu \leq \nu$, $P \in \mathbb{K}[x]^{\mu \times \nu}$ a full rank polynomial matrix and $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$ a shift. Then the \mathbf{s} -pivot indices are $\mathbf{j} = (j_1, \dots, j_\mu)$ and the \mathbf{s} -pivot degrees are $\mathbf{d} = (d_1, \dots, d_\mu)$ where j_i and d_i are respectively the \mathbf{s} -pivot index and the \mathbf{s} -pivot degree of $P_{i,*}$, for $1 \leq i \leq \mu$.

We are now ready to introduce the following notion.

Definition 1.3.7 ((Ordered) weak Popov form). Let $\mu \leq \nu$, $P \in \mathbb{K}[x]^{\mu \times \nu}$ be a full rank polynomial matrix and $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$ a shift. The polynomial matrix P is in

- \mathbf{s} -weak Popov form if the \mathbf{s} -pivot indices are pairwise distinct,
- \mathbf{s} -ordered weak (or quasi) Popov form if the sequence of \mathbf{s} -pivot indices is strictly increasing.

Note that if $\mu = \nu$ and P is in \mathbf{s} -ordered weak Popov form, the pivot indices are $(1, \dots, \mu)$. The polynomial matrix of Example 1.3.2 is in \mathbf{s} -ordered weak Popov form. We observe that bases in \mathbf{s} -weak Popov form (and in particular in ordered form) are also \mathbf{s} -row reduced. First, any matrix in \mathbf{s} -weak Popov form can be transformed into an ordered one by a row permutation. Moreover, the leading matrix of an ordered weak Popov basis is a lower triangular matrix with nonzero entries on the diagonal (e.g. Example 1.3.2). Hence it is full rank.

Ordered weak Popov bases have a strong degree minimality property that is particularly useful for this work, stated by the following lemma.

Lemma 1.3.6 ([Nei16, Lemma 1.17]). Let $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$ be a shift, $\mu \leq \nu$ and $P \in \mathbb{K}[x]^{\mu \times \nu}$ a full rank polynomial matrix in \mathbf{s} -weak Popov form with \mathbf{s} -pivot degrees $\mathbf{d} = (d_1, \dots, d_\mu)$. Let $\mathbf{p} \in \mathbb{K}[x]^{1 \times \nu}$ be a nonzero vector in the row space of P with \mathbf{s} -pivot index i . Then, the \mathbf{s} -pivot degree of \mathbf{p} is at least d_i .

Note that if P is a basis of \mathcal{M} , a $\mathbb{K}[x]$ -submodule of $\mathbb{K}[x]^\nu$ of rank ν , and $\mathbf{p} \in \mathcal{M}$ with \mathbf{s} -pivot index i , then by the previous lemma $\deg(p_i) \geq d_i$, and so $\text{rdeg}_{\mathbf{s}}(\mathbf{p}) \geq \text{rdeg}_{\mathbf{s}}(P_{i,*})$. Therefore, these bases provide more information about the row degree of elements of the corresponding $\mathbb{K}[x]$ -module than row-reduced ones.

We are now ready for the following result, which states the *uniqueness* of the \mathbf{s} -row degrees of ordered weak Popov bases.

Proposition 1.3.7. Let $\mathbf{s} = (s_1, \dots, s_\nu) \in \mathbb{Z}^\nu$ be a shift, $\mu \leq \nu$ and $P, R \in \mathbb{K}[x]^{\mu \times \nu}$ two full rank, unimodularly equivalent matrices in \mathbf{s} -ordered weak Popov form. Then P and R have the same \mathbf{s} -row degrees.

Proof. We denote $\text{rdeg}_s(P) := (\pi_1, \dots, \pi_\mu)$ and $\text{rdeg}_s(Q) := (\rho_1, \dots, \rho_\mu)$. Note that any row $P_{i,*}$ of P belongs to the row space of R , for $1 \leq i \leq \mu$. By Lemma 1.3.6, the pivot degree of $P_{i,*}$, i.e. $\pi_i - s_i$, is greater than the pivot degree of $R_{i,*}$, i.e. $\rho_i - s_i$, so $\text{rdeg}_s(P) \geq \text{rdeg}_s(R)$. On the other hand, any row $R_{i,*}$ of R belongs to the row space of P , hence the equality follows. \square

Therefore, we can conclude that for ordered weak Popov bases of a $\mathbb{K}[x]$ -modules, the tuples of shifted row degrees and pivot degrees are uniquely defined.

The shifted ordered weak Popov form is a weaker version of the shifted Popov form. The shifted Popov form is a shifted ordered weak Popov form with some additional constraints on the pivot entry (that must be monic) and on the degrees of elements of the columns containing the pivot entry. This notion was firstly introduced in [Pop72] for the uniform shift, in control theory. Contrary to shifted ordered weak Popov form, the shifted Popov form is canonical: given a $\mathbb{K}[x]$ -module there exists only one basis in shifted Popov form. However since in this thesis we are only interested in the uniqueness of the row and pivot degrees, we can simply focus on shifted ordered weak Popov forms.

The classic algorithm [MS03] for computing reduced bases, compute a basis in ordered weak Popov form.

1.3.2 Solutions of SRFR and Relation Module

Let $m \geq n \geq 0$ and $M \in \mathbb{K}[x]^{m \times n}$. We consider a $\mathbb{K}[x]$ -submodule \mathcal{M} of $\mathbb{K}[x]^n$ of rank n . We define the $\mathbb{K}[x]$ -module homomorphism,

$$\begin{aligned} \hat{\varphi}_M : \mathbb{K}[x]^m &\longrightarrow \mathbb{K}[x]^n / \mathcal{M} \\ \mathbf{p} &\longmapsto \mathbf{p}M \end{aligned}$$

We denote

$$\mathcal{A}_{\mathcal{M}, M} := \ker(\hat{\varphi}_M) = \{\mathbf{p} \in \mathbb{K}[x]^m \mid \mathbf{p}M = \mathbf{0} \text{ mod } \mathcal{M}\}, \quad (1.13)$$

the *relation module* whose elements are relations between rows of the matrix M .

By the First Isomorphism Theorem of modules [DF04, Theorem 4, Section 10.2], we get the injection,

$$\begin{aligned} \varphi_M : \mathbb{K}[x]^m / \mathcal{A}_{\mathcal{M}, M} &\hookrightarrow \mathbb{K}[x]^n / \mathcal{M} \\ \mathbf{p} &\longmapsto \mathbf{p}M \end{aligned} \quad (1.14)$$

In order to lighten the notations, we denote by $\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_m$ the *canonical basis* of $\mathbb{K}[x]^m$, by $\boldsymbol{\varepsilon}'_1, \dots, \boldsymbol{\varepsilon}'_n$ the *canonical basis* of $\mathbb{K}[x]^n$. Moreover, let $\mathbf{e}_i = \boldsymbol{\varepsilon}_i \text{ mod } \mathcal{A}_{\mathcal{M}, M}$ for $1 \leq i \leq m$.

Remark 1.3.4. By Theorem 1.3.2,

$$\mathcal{K} := \mathbb{K}[x]^n / \mathcal{M} \simeq \mathbb{K}[x] / (a_1(x)) \oplus \dots \oplus \mathbb{K}[x] / (a_n(x))$$

for nonzero polynomials $a_i(x) \in \mathbb{K}[x]$, $\deg(a_i(x)) \neq 0$ and such that $a_n \mid \dots \mid a_1$. Recall that such polynomials are the invariants of the module. For this reason, from now on we assume $\mathcal{M} := \langle a_i(x)\mathbf{e}'_i \rangle_{1 \leq i \leq n}$. We also denote $L_i := \deg(a_i)$ for $1 \leq i \leq n$ and observe that $L_1 \geq \dots \geq L_n$.

💡

Let $\mathbf{s} \in \mathbb{Z}^m$ be a shift, $M \in \mathbb{K}[x]^{m \times n}$, where $m \geq n \geq 0$ and suppose that all the invariants are equals $a = a_1 = \dots = a_n$. According to the form of the invariants, an \mathbf{s} -reduced basis of the relation module $\mathcal{A}_{\mathcal{M}, M}$ is called:

1. \mathbf{s} -minimal approximant basis of order L for M , if $a = x^L$;
2. \mathbf{s} -minimal interpolant basis of order L for M , if $a = \prod_{i=1}^L (x - \alpha_i)$, where $\{\alpha_1, \dots, \alpha_L\}$ are pairwise distinct elements of \mathbb{K} .

Recall that the row degrees of ordered Weak Popov bases are uniquely defined (Lemma 1.3.6). Hence it is convenient to give the following,

Definition 1.3.8 (Row and pivot degrees of a relation module). Let $\mathbf{s} \in \mathbb{Z}^m$ be a shift and P be any \mathbf{s} -ordered weak Popov basis of $\mathcal{A}_{\mathcal{M}, M}$. We shortly call $\boldsymbol{\rho} := \text{rdeg}_{\mathbf{s}}(P)$ and $\boldsymbol{\delta} = \boldsymbol{\rho} - \mathbf{s}$ respectively the \mathbf{s} -row degrees and \mathbf{s} -pivot degrees of $\mathcal{A}_{\mathcal{M}, M}$.

Sometimes, we also denote them $\boldsymbol{\rho}_M$ and $\boldsymbol{\delta}_M$ to stress their matrix dependency.

We now recall the SRFR problem (Problem 2). Fix an instance of SRFR, *i.e.* a_1, \dots, a_n and $\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{K}[x]^n$, where $\deg(u_i) < \deg(a_i)$ and the degree constraints N_i, D . We denote by $\mathcal{S}_{\mathbf{u}}$ the set of the corresponding solutions $(\mathbf{v}, d) \in \mathbb{K}[x]^{1 \times (n+1)}$ such that $v_i = du_i \pmod{a_i}$, $\deg(v_i) < N_i$ and $\deg(d) < D$.

Let $\mathcal{M} = \langle a_i(x)\mathbf{e}'_i \rangle$. The following lemma exploits the link between solutions of SRFR and elements of the relation module related to a specific matrix. In this case the shift is determined by the degree constraints.

Lemma 1.3.8. *Given the shift $\mathbf{s} = (-N_1, \dots, -N_n, -D) \in \mathbb{Z}^{n+1}$, then $(\mathbf{v}, d) \in \mathcal{S}_{\mathbf{u}}$ if and only if $(\mathbf{v}, d) \in \mathcal{A}_{\mathcal{M}, R_{\mathbf{u}}}$ with $\text{rdeg}_{\mathbf{s}}((\mathbf{v}, d)) < 0$ where*

$$R_{\mathbf{u}} := \begin{bmatrix} I_n \\ -\mathbf{u} \end{bmatrix} \in \mathbb{K}[x]^{(n+1) \times n} \quad (1.15)$$

Proof. Observe that $(\mathbf{v}, d) \in \mathcal{S}_{\mathbf{u}}$ if and only if $\mathbf{v} - d\mathbf{u} = (\mathbf{v}, d)R_{\mathbf{u}} = 0 \pmod{\mathcal{M}}$, *i.e.* $(\mathbf{v}, d) \in \mathcal{A}_{\mathcal{M}, R_{\mathbf{u}}}$, and $\text{rdeg}_{\mathbf{s}}((\mathbf{v}, d)) = \max\{\deg(v_1) - N_1, \dots, \deg(v_n) - N_n, \deg(d) - D\} < 0$ (see Remark 1.3.2). \square

By Remark 1.3.3, we can conclude that

$$\dim(\mathcal{S}_{\mathbf{u}}) = - \sum_{\rho_{\mathbf{u},i} < 0} \rho_{\mathbf{u},i} \quad (1.16)$$

where $\rho_{\mathbf{u}}$ are the \mathbf{s} -row degree of the relation module $\mathcal{A}_{\mathcal{M},R_{\mathbf{u}}}$.

Notice that in the uniqueness case $\dim(\mathcal{S}_{\mathbf{u}}) = 1$ there is only one element of an \mathbf{s} -ordered weak Popov basis of $\mathcal{A}_{\mathcal{M},R_{\mathbf{u}}}$ with negative \mathbf{s} -row degree.

In Chapter 3 (see Theorem 3.2.1) we will prove that for almost all instances \mathbf{u} of SRFR the \mathbf{s} -row degrees of $\mathcal{A}_{\mathcal{M},R_{\mathbf{u}}}$ are of the form $(0, \dots, 0, -1)$. Indeed, this means that there is only one generator of the solution space and we have the uniqueness of the solution ([BL97]).

Previous works about SRFR. In [OS07], the authors studied the SRFR problem (Problem 2) in the specific case of $a = a_1 = \dots = a_n$ and $N = N_1 = \dots = N_n$. They proved the following.

Theorem 1.3.9 ([OS07, Theorem 4.2]). *Let $a \in \mathbb{K}[x]$, $\mathbf{u} \in \mathbb{K}[x]^n$ and $1 \leq N, D \leq \deg(a)$. Let k be minimal such that $\deg(a) \geq N + (D - 1)/k$. Then the rank s of the $\mathbb{K}[x]$ -module spanned by the solutions of SRFR with instance \mathbf{u} , $a_1 = \dots = a_n = a$ and degree constraints $N_1 = \dots = N_n = N$ and D , is $s \leq k$.*

In other terms, the rank s of the $\mathbb{K}[x]$ -module of solutions is bounded. Note that, if $k = 1$, the solution is always unique since $s = 1$. This matches the uniqueness condition $\deg(a) \geq N + D - 1$ of the classic rational function reconstruction (Section 1.1). On the other hand, if $k = n$ and $\deg(a) \geq N + (D - 1)/n$ then $s \leq n$, which is always true. Therefore in this case this theorem does not provide any new information about the solution space. Besides, this was the starting point of this work, since it represents a connection between the classic bound on the $\deg(a)$ which guarantees the existence (classic RFR) and the *ideal* one, *i.e.* $\deg(a) = N + (D - 1)/n$ (see (1.7)) which exploits the common denominator constraint, assuring the existence of a nontrivial solution.

In the same paper, the authors also proposed an algorithm that allows to find all the rows of an $(-N, \dots, -N, -D)$ -row reduced basis with negative row degrees, *i.e.* a basis of the solution space, in

$$\mathcal{O}(nk^{\omega-1}B(\deg(a))) \quad (1.17)$$

where

- $2 \leq \omega \leq 3$ is the exponent of the matrix multiplication,
- $B(t) := \mathcal{O}(M(t) \log t)$, where $M(t)$ is the classic polynomial multiplication arithmetic complexity (see for instance [GG13]),
- k minimal such that $\deg(a) \geq N + (D - 1)/k$ (as in Theorem 1.3.9).

This complexity was then generalized in [RS16] where it was proposed an algorithm for the general case of different moduli (different invariants a_i) whose complexity is

$$\mathcal{O}(n^{\omega-1}B(L)\log(L/n)^2) \quad (1.18)$$

where $L := \max_{1 \leq i \leq n} \{\deg(a_i)\}$.

Both algorithms of [OS07] and [RS16] are strongly based on minimal approximant bases computation.

Notice that if we consider $k = n$ and $a_1 = \dots = a_n = a$, the complexity (1.17) of the algorithm of [OS07] becomes $\mathcal{O}(n^\omega B(\deg(a)))$, which is bigger than $\mathcal{O}(n^{\omega-1}B(\deg(a))\log(\deg(a)/n)^2)$ (see equation (1.18)). Therefore, in this case (which corresponds to $\deg(a) \geq N + (D-1)/n$) the complexity of the algorithm proposed in [RS16] is better than the complexity of the algorithm proposed in [OS07].

1.4 Application to Polynomial Linear System Solving

The simultaneous rational function reconstruction is applied for solving systems of linear equations with polynomial coefficients, called *polynomial linear systems* (PLS). Throughout this thesis we focus on nonsingular, square linear systems. Let

$$A(x)\mathbf{y}(x) = \mathbf{b}(x) \quad (1.19)$$

be a PLS where,

- $A(x)$ is an $n \times n$ nonsingular matrix, whose entries are polynomials in $\mathbb{K}[x]$,
- $\mathbf{b}(x)$ is a column vector of polynomials in $\mathbb{K}[x]$.

Lemma 1.4.1. *The PLS (1.19) admits only one solution which is a vector of rational functions with the same denominator, $\mathbf{y} = \frac{\mathbf{v}}{d} = (\frac{v_1}{d}, \dots, \frac{v_n}{d}) \in \mathbb{K}(x)^{n \times 1}$. Moreover $\deg(\mathbf{v}) \leq (n-1)\deg(A) + \deg(\mathbf{b})$ and $\deg(d) \leq n\deg(A)$, where $\deg(A) = \max_{1 \leq i, j \leq n} \{\deg(a_{ij})\}$.*

Proof. We denote by $A_{*,j}$ the j -th column of the matrix A , for $1 \leq j \leq n$. Note that we can write the system (1.19) as the linear combination $A_{*,1}y_1 + A_{*,2}y_2 + \dots + A_{*,n}y_n = \mathbf{b}$. Hence, by Cramer's Rule

$$y_i = \frac{\det(A_{*,1} \dots A_{*,i-1} \ \mathbf{b} \ A_{*,i+1} \dots A_{*,n})}{\det(A)} \in \mathbb{K}(x), \quad 1 \leq i \leq n.$$

and so follows the claim. □

Fix a PLS (1.19) and let $\mathbf{y} = \frac{\mathbf{v}}{d}$ be its solution. A classic approach to solve a PLS consists in the computation of $\mathbf{u} = A^{-1}\mathbf{b} \pmod{a}$, for a certain polynomial a , and then in SRFR² with

2. Usually for the application of SRFR to polynomial linear system solving it is assumed that all the polynomials a_i 's are equals.

instance \mathbf{u} . There are many techniques for the computation of \mathbf{u} , which differs according to the *decomposition* of the polynomial a : some are based on the *interpolation* (as *evaluation-interpolation*, that we will see in the following paragraph) [BCG⁺17, Section 5] and others on *p-adic methods* [Dix82, MC79, Sto03].

Uniqueness results. We now fix a polynomial linear system (1.19)

Lemma 1.4.2 ([OS07, Theorem 5.1]). *If $\deg(a) > \max\{N + \deg(A), D + \deg(\mathbf{b})\} - 1$, then SRFR with instance $\mathbf{u} = A^{-1}\mathbf{b} \bmod a$ and degree constraints N, D (see Problem 2) admits a unique solution. In other terms the rank of the $\mathbb{K}[x]$ -module generated by the solutions is $s = 1$.*

Proof. Let (\mathbf{v}_1, d_1) and (\mathbf{v}_2, d_2) be two solutions of SRFR, *i.e.*

$$A\mathbf{v}_i = d_i\mathbf{b} \bmod a, \quad \deg(\mathbf{v}_i) < N, \quad \deg(d_i) < D,$$

for $i = 1, 2$.

We observe that $\deg(A\mathbf{v}_i - d_i\mathbf{b}) \leq \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} - 1$ is smaller than the degree of a which implies that $A\mathbf{v}_i = d_i\mathbf{b}$ for $i = 1, 2$. Now, if we multiply $A\mathbf{v}_1 = d_1\mathbf{b}$ and $A\mathbf{v}_2 = d_2\mathbf{b}$ by d_2 and d_1 respectively and then we subtract them, we get $A(\mathbf{v}_1d_2 - \mathbf{v}_2d_1) = 0$. Since A is full rank then $\mathbf{v}_1d_2 - \mathbf{v}_2d_1 = \mathbf{0}$ and so $\frac{\mathbf{v}_1}{d_1} = \frac{\mathbf{v}_2}{d_2}$. \square

Recall that by Lemma 1.4.1, the degrees of the numerator and denominator of the solution of a PLS are bounded by $(n - 1)\deg(A) + \deg(\mathbf{b})$ and $n\deg(A)$. If we consider $N := (n - 1)\deg(A) + \deg(\mathbf{b}) + 1$ and $D := n\deg(A) + 1$, then by Lemma 1.4.2, if

$$\begin{aligned} \deg(a) > \max\{N + \deg(A), D + \deg(\mathbf{b})\} - 1 &= n\deg(A) + \deg(\mathbf{b}) \\ &= (N - 1 + \deg(A) - \deg(\mathbf{b})) + \deg(\mathbf{b}) \\ &= N - 1 + (D - 1)/n. \end{aligned}$$

we can uniquely reconstruct the solution of the PLS. Equivalently the rank of the $\mathbb{K}[x]$ -module generated by solutions is 1. The authors of [OS07] proved the same result in the particular case $N = D = n\deg(A) + 1$ and $\deg(A) = \deg(\mathbf{b})$. In this work we extend this result to a more general case (see Chapter 3). This proves that there are some cases, for which the condition $\deg(a) \geq N + (D - 1)/n$ guarantees uniqueness. This was the starting point of this work: indeed the existence of this specific case motivates us to investigate about the uniqueness of the solution of SRFR.

Evaluation-interpolation for PLS solving. Fix a PLS (1.19). If $a = \prod_{i=1}^L (x - \alpha_i)$, where α_i are pairwise distinct elements of the field \mathbb{K} , SRFR with instance $\mathbf{u} = A^{-1}\mathbf{b} \bmod a$

and with degree constraints N, D , is equivalent to the problem of finding (\mathbf{v}, d) such that

$$A(\alpha_i)\mathbf{v}(\alpha_i) = d(\alpha_i)\mathbf{b}(\alpha_i), \quad \deg(\mathbf{v}) < N, \deg(d) < D. \quad (1.20)$$

This is an evaluation-interpolation technique since we first evaluate the matrix A and the vector \mathbf{b} in the L evaluation points and then we interpolate the solution \mathbf{y} of the PLS given its evaluations $\mathbf{y}(\alpha_i) = \mathbf{u}(\alpha_i) = \frac{\mathbf{v}(\alpha_i)}{d(\alpha_i)} = A(\alpha_i)^{-1}\mathbf{b}(\alpha_i)$. Notice that we are assuming that for any $1 \leq i \leq L$, the corresponding evaluated matrix $A(\alpha_i)$ is still full rank.

Therefore we can conclude that, if

$$L \geq \min\{N + D - 1, \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}\} \quad (1.21)$$

we can uniquely reconstruct the solution of our problem. Notice that we take the minimum to reduce the computations.

Rank drop case. Fix a PLS (1.19). Consider L pairwise distinct evaluation points $\{\alpha_1, \dots, \alpha_L\}$. Following [BK14, KPSW17] we assume that there exists $1 \leq i \leq L$ such that $\det(A(\alpha_i)) = 0$. In other terms, we want to determine the number of evaluation points that we need to uniquely reconstruct the solution of the PLS by evaluation-interpolation, if we consider evaluation points which may drop the rank of the corresponding evaluated matrix.

For this purpose, consider

$$R = \{1 \leq i \leq L \mid \det(A(\alpha_i)) = 0\}$$

and assume to know a bound $r \geq |R|$.

In [BK14, KPSW17] authors proved that, in order to handle rank drops, it suffices to add this bound r to the number of points which guarantees the uniqueness.

Theorem 1.4.3. [BK14, KPSW17] *If $L \geq \min\{N + D - 1, \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}\} + r$, then SRFR (1.20) admits a unique solution.*

Proof. We divide the proof in two cases:

1. $N + D - 1 \leq \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}$,
2. $\max\{\deg(A) + N, \deg(\mathbf{b}) + D\} \leq N + D - 1$.

Let $(\mathbf{v}_1, d_1), (\mathbf{v}_2, d_2)$ solutions of (1.20).

1. For any $1 \leq i \leq L$,

$$\begin{aligned} A(\alpha_i)\mathbf{v}_1(\alpha_i) &= \mathbf{b}(\alpha_i)d_1(\alpha_i) \\ A(\alpha_i)\mathbf{v}_2(\alpha_i) &= \mathbf{b}(\alpha_i)d_2(\alpha_i) \end{aligned}$$

By multiplying the first equation by $d_2(\alpha_i)$ and the second by $d_1(\alpha_i)$ and then subtracting them we finally get,

$$A(\alpha_i)[\mathbf{v}_1(\alpha_i)d_2(\alpha_i) - \mathbf{v}_2(\alpha_i)d_1(\alpha_i)] = 0.$$

Now, for $i \notin R$, the matrix $A(\alpha_i)$ has linearly independent columns and so $\mathbf{v}_1(\alpha_i)d_2(\alpha_i) - \mathbf{v}_2(\alpha_i)d_1(\alpha_i) = 0$. Note that the vector of polynomials $\mathbf{v}_1(x)d_2(x) - \mathbf{v}_2(x)d_1(x)$ has $\deg(\mathbf{v}_1d_2 - \mathbf{v}_2d_1) < N + D$ and $L - |R| \geq L - r \geq N + D - 1$ roots, and so $\mathbf{v}_1(x)d_2(x) = \mathbf{v}_2(x)d_1(x)$.

2. We have that (\mathbf{v}_1, d_1) satisfies $A(\alpha_i)\mathbf{v}_1(\alpha_i) = \mathbf{b}(\alpha_i)d_1(\alpha_i)$ for any $1 \leq i \leq L$. Notice that the degree of $A(x)\mathbf{v}_1(x) - \mathbf{b}(x)d_1(x)$ is at most $\max\{\deg(A) + N, \deg(\mathbf{b}) + D\} - 1$ and the number of roots is $L - |R| \geq L - r \geq \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}$ hence $A(x)\mathbf{v}_1(x) - \mathbf{b}(x)d_1(x) = 0$. Therefore, we have also that $A(x)\mathbf{v}_2(x) - \mathbf{b}(x)d_2(x) = 0$. By multiplying $A(x)\mathbf{v}_1(x) - \mathbf{b}(x)d_1(x) = 0$ by $d_2(x)$ and $A(x)\mathbf{v}_2(x) - \mathbf{b}(x)d_2(x) = 0$ by $d_1(x)$ and then subtracting them we get $A(x)[\mathbf{v}_1(x)d_2(x) - \mathbf{v}_2(x)d_1(x)] = \mathbf{0}$. So, since $A(x)$ is nonsingular then $\frac{\mathbf{v}_1}{d_1} = \frac{\mathbf{v}_2}{d_2}$ and the claim follows. \square

1.5 A short summary of the chapter

We now briefly summarize the main notions and results of this chapter.

On the uniqueness of Simultaneous Rational Function Reconstruction. The *rational function reconstruction* (RFR) is the problem of reconstructing a rational function whose numerator and denominator degrees are bounded, given its remainder modulo a polynomial. We saw that under a certain assumption (1.3) on the parameters of the problem, the solution is *unique*, *i.e.* any solution is a polynomial multiple of a *minimal* one. In particular, the set of solutions of RFR is a subset of a $\mathbb{K}[x]$ -module of rank 1.

The *simultaneous rational function reconstruction* (SRFR) is the *vector* generalization of RFR, in which all the rational functions share the same denominator. In this case, we want to reconstruct a *vector* of rational functions with the same denominator, given its remainder modulo different polynomials. Therefore, under the same uniqueness condition of RFR, a solution of SRFR (if it exists) is also unique. Nevertheless, the common denominator constraint affects the condition on the parameters of the problem which guarantees the existence of a nontrivial solution, possibly losing its uniqueness (Example 1.2.1).

The $\mathbb{K}[x]$ -module of solutions of SRFR. In Section 1.3 we introduced some general notions about elements and bases of $\mathbb{K}[x]$ -modules: *shifted row degrees*, *shifted row reduced bases* and *ordered weak Popov bases*. Recall that a basis of a $\mathbb{K}[x]$ -module is shifted row reduced if it has minimal shifted row degrees. Moreover an ordered weak Popov basis is a specific shifted row reduced basis for which the shifted row degree is uniquely defined.

All these ingredients were essential to check the uniqueness of the SRFR: SRFR admits a unique solution if there is only one generator of an ordered weak Popov basis of a specific

module, *i.e.* the *relation module*, with negative row degree. Recall that the shifts are necessary to integrate the degree constraints.

Application of SRFR to Polynomial Linear System solving. In Section 1.4 we saw that solutions of polynomial linear systems are vectors of rational functions with the same denominator (Lemma 1.4.1). Indeed the classic evaluation interpolation approach to solve PLS basically consists in an SRFR. In this case, the common denominator assumption allows to reduce the number of evaluation points needed to *uniquely* reconstruct the solution of PLS, impacting the complexity of the chosen algorithm for the PLS solving.

CHAPTER 2

Application of SFRF to Coding Theory

Contents

2.1	Basics of Coding Theory	52
2.1.1	Channel Model	52
2.1.2	Block codes	54
2.1.3	Basic decoding principles	57
2.2	Reed-Solomon Codes	60
2.2.1	Decoding RS codes	62
2.3	Interleaved Reed-Solomon code	69
2.3.1	Decoding IRS codes	70
2.4	A short summary of the chapter	78

The simultaneous rational function reconstruction is also applied for the decoding of *interleaved Reed-Solomon codes*. This chapter introduces basic notions of coding theory (Section 2.1), especially focusing on linear block codes. It then presents a famous and widespread family of such codes: *Reed-Solomon codes* (Section 2.2). In Subsection 2.2.1 we explain how the decoding of this class of codes can be seen as the rational function reconstruction problem (Section 1.1).

Section 2.3 is devoted to the introduction of Interleaved Reed-Solomon codes. Interleaving is a technique that allows one to construct robust codes with interesting decoders [BKY03, BMS04, SSB07, SSB09, SSB10, PR17]. We then underline the link between the decoding of such codes and the simultaneous rational function reconstruction problem (Subsection 2.3.1).

So, by summing up, the goal of this chapter is to introduce the basic decoding techniques of Reed-Solomon and interleaving Reed-Solomon codes, which are reinterpreted with a focus on the main problem of this thesis: the simultaneous rational function reconstruction.

Notations. We start by making some clarifications about the notations we are going to use throughout this chapter. For any set A we denote by A^n the set of n -tuples $\mathbf{a} = (a_1, \dots, a_n)$ of elements of A .

If $A = \mathbb{F}_q$ or $A = \mathbb{F}_q[x]$, *i.e.* the finite field of order q or the ring of polynomials with coefficients in \mathbb{F}_q , we identify A^n with the row vector space $A^n = A^{1 \times n}$. In this case, as in the previous chapter, we use the lowercase bold notation for row vectors. Notice that in Section 2.3 we will use the same notation also for column vectors. Nevertheless, for the sake of clarity we will specify if they belong to A^n or to $A^{n \times 1}$.

2.1 Basics of Coding Theory

This section provides some preliminary notions related to coding theory. We refer to classic literature about this topic, *e.g.* [Ber15, Bla03, Rot06, HP03, PWBJ17] and the notes in [GRS19].

Coding theory is a discipline intersecting mathematics, computer science and engineering, concerned with the problem of a communication over a *noisy* channel that can introduce some errors, corrupting the transmitted message. It has many applications: from data transmission over the Internet, any kind of electronic communication device as cellular telephones, deep space communication and satellite broadcast, to compact disks and other physical media for which the data integrity is crucial. The main idea to recover a message after its transmission over an unreliable channel, is to add *redundancy*. In the classic communication scenario [Sha48], a k -symbol *message* (also called *information word*) is first *encoded* obtaining an n -symbol message, called *codeword*, transmitted over a noisy channel and finally *decoded* in order to recover the original message. Informally speaking, encoding is the process of adding redundancy and decoding is the process of removing errors.

2.1.1 Channel Model

The classic Shannon channel model is defined by the triple (Σ, Φ, Pr) , where

- Σ is the *input alphabet*,
- Φ is the *output alphabet*,
- the *conditional probability distribution* $Pr(\mathbf{y} \text{ received} | \mathbf{x} \text{ sent})$ is defined for every pair $(\mathbf{x}, \mathbf{y}) \in \Sigma^n \times \Phi^n$. Recall that Σ^n and Φ^n both denote the set of n -tuples in the alphabet Σ and Φ respectively.

Channels considered by Shannon are also *memoryless*, which means that the noise acts independently on each transmitted symbol.

In this work we focus on *discrete* channels: the input and the output alphabet are both finite. We also suppose that the channel is *additive*, *i.e.* $\Sigma = \Phi$ and Σ is a finite abelian group (indeed, for every positive integer q there is an abelian group of size q , *e.g.* the ring \mathbb{Z}_q of integers modulo q). The action of such a kind of channels can be described as adding

componentwise an *error word* $\mathbf{e} \in \Sigma^n$, e.g.

$$\mathbf{x} \in \Sigma^n \longrightarrow \boxed{\text{channel}} \longrightarrow \mathbf{y} = \mathbf{x} + \mathbf{e} \in \Sigma^n$$

\uparrow
 \mathbf{e}

The *error support* E is the set of nonzero positions of the *error vector* \mathbf{e} , i.e. $E := \{i \in \{1, \dots, n\} \mid e_i \neq 0\}$.

We now introduce two examples of channels frequently used in practical applications.

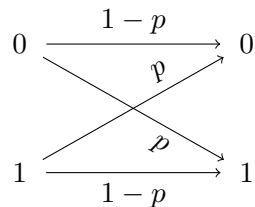
Example 2.1.1 (Binary Symmetric Channel). The Binary Symmetric Channel (BSC) has input and output alphabet equal to $\Sigma = \mathbb{F}_2$. Moreover, for any $(\mathbf{x}, \mathbf{y}) \in (\mathbb{F}_2)^{2n}$,

$$\Pr(\mathbf{y} \text{ received} \mid \mathbf{x} \text{ sent}) = \prod_{i=1}^n \Pr(y_i \text{ received} \mid x_i \text{ sent})$$

where for any $x, y \in \mathbb{F}_2$

$$\Pr(y \text{ received} \mid x \text{ sent}) = \begin{cases} 1 - p & \text{if } y = x \\ p & \text{if } y \neq x \end{cases}$$

The parameter $0 \leq p \leq 1$ is called *crossover probability* of the channel. Usually, BSC is represented by the following diagram,



We can assume that $p < 1/2$. Indeed, by Shannon Theorem (see for instance [HP03, Theorem 1.11.10]) if $p = 1/2$, the communication is not possible. Moreover, if one knows how to assure reliable communication over a BSC with crossing probability $p < 1/2$, he also knows how to handle the case $p > 1/2$. Indeed, after receiving a symbol $y \in \mathbb{F}_2$, the receiver could interpret a 0 with a 1 and *vice versa*, transforming the BSC channel with crossover probability p to an equivalent one with crossover probability $1 - p < 1/2$.

Example 2.1.2 (q -ary Symmetric Channel). The q -ary Symmetric Channel with *crossover probability* p is a generalization of BSC where the alphabet is $\Sigma = \mathbb{F}_q$, where $q \geq 2$. In this

case, given $x, y \in \mathbb{F}_q$, the conditional probability is

$$\Pr(y \text{ received} | x \text{ sent}) = \begin{cases} 1 - p & \text{if } y = x \\ p/(q - 1) & \text{if } y \neq x \end{cases}$$

In other terms, every symbol rests unchanged with probability $1 - p$ and it is switched to each of the $q - 1$ possible different symbols with probability $p/(q - 1)$.

This specific channel model usually applies to settings where aggregates of bits are sent and errors are assumed to be *bursty*. In particular, bursts errors are errors which are extended to consecutive bits of the received word. This kind of errors frequently appears in many communication and data storage channels: they could be caused by media defects or interference with contiguous symbols. For instance, for CDs they can be caused by scratches on the disk surface or by interferences from adjacent tracks.



2.1.2 Block codes

Coding theory mainly studies two classes of codes: finite and infinite length ones. Finite length codes, known as *block codes* were first studied by Golay [Gol49] and Hamming [Ham50]. Throughout this thesis we focus on the latter class of codes.

Definition 2.1.1 (Block code). Given an alphabet Σ , a (*block*) *code* \mathcal{C} of length n is a subset of Σ^n . We denote $q := |\Sigma|$. Elements $\mathbf{c} = (c_1, \dots, c_n) \in \mathcal{C}$ are called *codewords*. The *size* of the code is its cardinality $M := |\mathcal{C}|$ and its *dimension* (or *information length*) is $\log_q(M)$. Moreover, the *rate* of \mathcal{C} is $R := \log_q(M)/n$.

We refer to a code \mathcal{C} with length n and size M as an (n, M) -code.

Note that the rate of a code is the average amount of real information contained in each of the n symbols; so the higher the rate the lesser the redundancy of the code. In these terms, the rate measures the redundancy of a code.

Definition 2.1.2 (Encoder). Let \mathcal{C} be an (n, M) -code over Σ , where $q := |\Sigma|$. Assume that $M = q^k$. An *encoder* of \mathcal{C} is a one-to-one map

$$E : \Sigma^k \longrightarrow \Sigma^n$$

such that $\mathcal{C} = E(\Sigma^k)$. Given a codeword $\mathbf{c} \in \mathcal{C}$ there exists a unique $\mathbf{x} \in \Sigma^k$ such that $\mathbf{c} = E(\mathbf{x})$. This \mathbf{x} is called *message* or *information word*.

Besides the length and the dimension of a code, it is also important to define a metric which measures the distance between two codewords.

Definition 2.1.3 (Hamming distance and weight). The *Hamming distance* between $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$ in Σ^n is defined as the number of components in which \mathbf{x} and \mathbf{y} differ, *i.e.*

$$d(\mathbf{x}, \mathbf{y}) := |\{i \mid x_i \neq y_i\}|,$$

and the *Hamming weight* of \mathbf{x} is

$$w(\mathbf{x}) := |\{i \mid x_i \neq 0\}|.$$

The Hamming distance is a well defined metric on Σ^n (see [PWBJ17, Proposition 1.1.9]).

The *minimum distance* of an (n, M) -code \mathcal{C} is then the minimum Hamming distance between any two distinct codewords, *i.e.* $d := \min\{d(\mathbf{c}_1, \mathbf{c}_2) \mid \mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}, \mathbf{c}_1 \neq \mathbf{c}_2\}$. Length, size and distance are the *parameters* of a code. We can merge them into one expression and refer to \mathcal{C} as an (n, M, d) -code.

Let $\mathbf{x} \in \Sigma^n$ and $r \geq 0$. We denote by $\mathcal{B}^{(r)}(\mathbf{x}) := \{\mathbf{y} \in \Sigma^n \mid d(\mathbf{x}, \mathbf{y}) \leq r\}$, the *ball* of *radius* r centered at \mathbf{x} , *w.r.t* the Hamming metric.

We can now introduce the following important result.

Theorem 2.1.1 (Unique Decoding Capability). *Let \mathcal{C} be an (n, M, d) -code and $\tau_0 := \lfloor \frac{d-1}{2} \rfloor$. Then for any pair of distinct codewords $\mathbf{c}_1, \mathbf{c}_2$,*

$$\mathcal{B}^{(\tau_0)}(\mathbf{c}_1) \cap \mathcal{B}^{(\tau_0)}(\mathbf{c}_2) = \emptyset.$$

Proof. Fix $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}$ with $\mathbf{c}_1 \neq \mathbf{c}_2$ and suppose that $\mathbf{y} \in \mathcal{B}^{(\tau_0)}(\mathbf{c}_1) \cap \mathcal{B}^{(\tau_0)}(\mathbf{c}_2)$. Then $d(\mathbf{c}_i, \mathbf{y}) \leq \tau_0$ for $i = 1, 2$ and by the triangle inequality

$$d(\mathbf{c}_1, \mathbf{c}_2) \leq d(\mathbf{c}_1, \mathbf{y}) + d(\mathbf{c}_2, \mathbf{y}) \leq d - 1.$$

The minimum distance of the code is d , so $d(\mathbf{c}_1, \mathbf{c}_2) \geq d$. This implies that $\mathbf{c}_1 = \mathbf{c}_2$, which contradicts our hypothesis. \square

This theorem is crucial since it highlights the link between the minimum distance and the *error correction capability* of a code, which is the maximum number of errors that one can *uniquely* correct. Indeed, we assume to send a codeword \mathbf{c} of a certain (n, M, d) -code \mathcal{C} over a channel and to receive $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \Sigma^n$, where $\mathbf{e} \in \Sigma^n$ is the *error vector*. Note that the number of errors is the distance $d(\mathbf{y}, \mathbf{c})$ or the weight $w(\mathbf{e})$. Theorem 2.1.1 tells us that if the number of errors is less than τ_0 (*i.e.* $d(\mathbf{y}, \mathbf{c}) \leq \tau_0$), then \mathbf{y} can be corrected: it suffices to consider the codeword inside the ball $\mathcal{B}^{(\tau_0)}(\mathbf{y})$, which is then unique (see Subsection 2.1.3). Therefore the error correction capability of \mathcal{C} is τ_0 , which is strictly related to its minimum distance. In this sense the minimum distance allows one to quantify the error correction capability of a code.

For this reason, one of the main goal of coding theory is to construct for a given length and size a code with the largest possible minimum distance.

In the literature, there exist many bounds on the parameters of codes. One of the most famous is the *Singleton bound* [Sin64]: for any (n, M, d) -code over an alphabet Σ of size q then

$$d \leq n - (\log_q(M)) + 1. \quad (2.1)$$

Codes which attain the Singleton bound are called *Maximum Distance Separable* (MDS) codes. The name MDS comes from the fact that this code has the maximum possible distance between codewords.

Linear codes. One important class of algebraic codes, is the class of *linear codes*.

Definition 2.1.4 (Linear code). Let $\Sigma = \mathbb{F}_q$. A *linear code* \mathcal{C} of length n is an \mathbb{F}_q -vector subspace of $(\mathbb{F}_q)^n$. In this case the dimension of the code is its dimension as a vector space. If $k = \dim(\mathcal{C})$ we refer to \mathcal{C} as an $[n, k, d]$ -code.

Note that the size of an $[n, k, d]$ -linear code over \mathbb{F}_q is q^k and its rate is k/n .

We now briefly recall some basic notions of linear codes.

A *generator matrix* of a $[n, k, d]$ -code \mathcal{C} over \mathbb{F}_q is a $k \times n$ matrix G whose rows form a basis of the code. This matrix can be used to describe the *encoding* process. Indeed, we can consider the following map as an *encoder* (see Definition 2.1.2)

$$\begin{aligned} E : \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ \mathbf{x} &\longmapsto \mathbf{x}G \end{aligned}$$

In other terms, given a message \mathbf{x} in \mathbb{F}_q^k , the corresponding codeword \mathbf{c} is obtained by the multiplication $\mathbf{c} = \mathbf{x}G$.

Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, let $\mathbf{x} \cdot \mathbf{y} := \sum_{i=1}^n x_i y_i$ be the *inner product* and \mathcal{C} be an $[n, k, d]$ -linear code over \mathbb{F}_q . We define the dual code of \mathcal{C} as

$$\mathcal{C}^\perp := \{\mathbf{h} \in \mathbb{F}_q^n \mid \mathbf{h} \cdot \mathbf{c} = 0, \forall \mathbf{c} \in \mathcal{C}\}$$

The dual code \mathcal{C}^\perp is a linear code of length n and dimension $n - k$. A generator matrix H of the dual code is called *parity-check* matrix of \mathcal{C} . The minimum distance of this code is not always determined by its parameters (unless they are MDS). Note that for any $\mathbf{c} \in \mathcal{C}$, $H\mathbf{c}^T = \mathbf{0}$ and also $HG^T = \mathbf{0}$.

Let \mathcal{C} be an $[n, k, d]$ -code with parity check matrix H . Given $\mathbf{y} \in \mathbb{F}_q^n$, the vector $\mathbf{s} = H\mathbf{y}^T$ is a *syndrome* of \mathbf{y} . We remark that,

- the syndrome of a codeword is the zero vector;

- if $\mathbf{y} = \mathbf{c} + \mathbf{e}$ is the received word after the transmission over a channel where $\mathbf{c} \in \mathcal{C}$ and $\mathbf{e} \in \mathbb{F}_q^n$ is the *error vector*, the *syndrome* \mathbf{s} only depends on the error vector. Indeed,

$$\mathbf{s} = H\mathbf{y}^T = H\mathbf{c}^T + H\mathbf{e}^T = H\mathbf{e}^T.$$

For this reason, syndromes are used for the construction of decoders. For instance, in Subsection 2.2.1 we will see a syndrome-based decoding technique for RS codes.

2.1.3 Basic decoding principles

Let \mathcal{C} be an (n, M, d) -code over an alphabet Σ . Decoding is the process of reconstructing the sent codeword from the received word.

Definition 2.1.5 (Decoder). A *decoder* D for the code \mathcal{C} is a map

$$D : \Sigma^n \longrightarrow \Sigma^n \cup \{?\}$$

such that for any $\mathbf{c} \in \mathcal{C}$, $D(\mathbf{c}) = \mathbf{c}$.

If $M = q^k$ and $E : \Sigma^k \longrightarrow \Sigma^n$ is an encoder of \mathcal{C} (Definition 2.1.2), then the map $D : \Sigma^n \longrightarrow \Sigma^n \cup \{?\}$ such that $D(E(\mathbf{x})) = \mathbf{x}$ for any $\mathbf{x} \in \Sigma^k$ is called *decoder w.r.t the encoder E* .

Note that if $M = q^k$, E is an encoder of \mathcal{C} and D is a decoder *w.r.t* E , then the composition $D \circ E$ is a decoder of \mathcal{C} . The decoder could also give as outcome the symbol “?”, in this case we say that we have a *decoding failure*. This happens if it fails to find a codeword.

A decoder D is *complete* if it always returns a codeword.

Maximum-likelihood decoding. The *maximum-likelihood* (ML) decoding is a well studied decoding technique for block codes. In particular, for any $\mathbf{y} \in \Sigma^n$ the ML decoder D_{ML} is a complete decoder which gives as an outcome a codeword $\mathbf{c} = D_{ML}(\mathbf{y})$ that maximize the following probability, *i.e.*

$$Pr(\mathbf{y} \text{ received} | \mathbf{c} \text{ sent}).$$

If there exist two codewords for which the corresponding conditional probability is the same, the decoder can arbitrarily choose one of them (for example the first according to some ordering on the codewords of \mathcal{C}).

Nearest-codeword decoding. The *nearest-codeword* (NC) decoder (Figure 2.1) D_{NC} , maps any received word $\mathbf{y} \in \Sigma^n$ to $D_{NC}(\mathbf{y})$ which is a nearest codeword *w.r.t* the Hamming distance, *i.e.* $D_{NC}(\mathbf{y}) = \mathbf{c}$, where

$$\mathbf{c} \in \arg \min_{\mathbf{c} \in \mathcal{C}} \{d(\mathbf{y}, \mathbf{c})\} = \{\mathbf{c} \in \mathcal{C} \mid \forall \mathbf{c}' \in \mathcal{C}, d(\mathbf{c}, \mathbf{y}) \leq d(\mathbf{c}', \mathbf{y})\}.$$

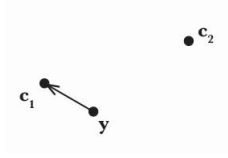


Figure 2.1: Nearest-codeword decoding

The following lemma shows the equivalence of the two decoders introduced so far for BSC.

Lemma 2.1.2 (Equivalence between ML and NC decoders for BSC). *In a BSC with crossover probability $p < 1/2$, the maximum-likelihood and the nearest-codeword decoder both coincide.*

Proof. Let $\mathbf{y} \in \Sigma^n$ and \mathbf{c} be a codeword, then

$$\Pr(\mathbf{y} \text{ received} | \mathbf{c} \text{ sent}) = p^{d(\mathbf{y}, \mathbf{c})} (1-p)^{n-d(\mathbf{y}, \mathbf{c})} = (1-p)^n \left(\frac{p}{1-p} \right)^{d(\mathbf{y}, \mathbf{c})}$$

We observe that since $p < 1/2$, then $\left(\frac{p}{1-p} \right) < 1$. Therefore, maximizing this probability is equivalent to minimizing the distance. \square

We can prove the same result for a q -ary symmetric channel with crossover probability $p < 1 - 1/q$.

The problem of NC decoding is known to be NP-complete [BMT78]. For this reason, it is customary to consider the following decoders.

Bounded minimum distance decoder. Let $\tau_0 = \lfloor \frac{d-1}{2} \rfloor$ be the error correction capability of the code \mathcal{C} . First notice that if \mathbf{y} is a received word such that $\mathcal{C} \cap \mathcal{B}^{(\tau_0)}(\mathbf{y}) \neq \emptyset$, then by Theorem 2.1.1, there is only one element in the ball, *i.e.* $|\mathcal{C} \cap \mathcal{B}^{(\tau_0)}(\mathbf{y})| = 1$. A *bounded minimum distance* (BMD) decoder (Figure 2.2) is a decoder D_{τ_0} such that for any received word $\mathbf{y} \in \Sigma^n$,

- if $\mathcal{C} \cap \mathcal{B}^{(\tau_0)}(\mathbf{y}) \neq \emptyset$, then $D_{\tau_0}(\mathbf{y}) = \mathbf{c}$, where c is the only codeword in the ball $\mathcal{B}^{(\tau_0)}(\mathbf{y})$,
- otherwise $D_{\tau_0}(\mathbf{y}) = ?$, meaning that it outputs a decoding failure.

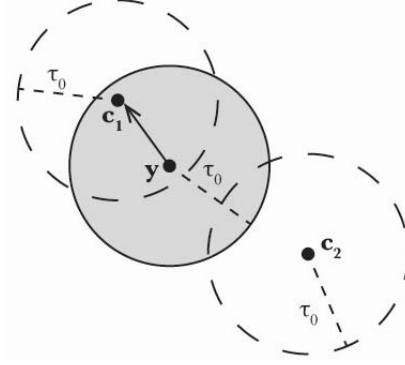


Figure 2.2: Bounded minimum distance decoding

Bounded distance decoder. A *bounded distance* (BD) decoder is a generalization of the BMD decoder for a general *decoding radius* $\tau \geq \tau_0$. We denote it D_τ . Note that if $\tau \geq \tau_0$, Theorem 2.1.1 does not hold anymore, meaning that if $\mathbf{y} \in \Sigma^n$ such that $\mathcal{C} \cap \mathcal{B}^{(\tau)}(\mathbf{y}) \neq \emptyset$ there could exist more than one codeword in the ball $\mathcal{B}^{(\tau)}(\mathbf{y})$ (see Figure 2.3).

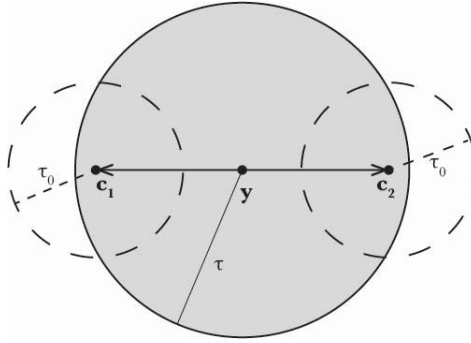


Figure 2.3: Existence of two codewords $\mathbf{c}_1, \mathbf{c}_2$ at the same distance $\leq \tau$ from the received word \mathbf{y} .

- Therefore, for any received word $\mathbf{y} \in \Sigma^n$,
- if $|\mathcal{C} \cap \mathcal{B}^{(\tau)}(\mathbf{y})| = 1$, then $D_\tau(\mathbf{y}) = \mathbf{c}$ where \mathbf{c} the only codeword in the ball $\mathcal{B}^{(\tau)}(\mathbf{y})$,
 - otherwise,
 - if $\mathcal{C} \cap \mathcal{B}^{(\tau)}(\mathbf{y}) = \emptyset$, then $D_\tau(\mathbf{y}) = ?$,
 - if $|\mathcal{C} \cap \mathcal{B}^{(\tau)}(\mathbf{y})| > 1$, then $D_\tau(\mathbf{y}) = ?$.

List decoding Another famous decoding technique, which can be seen as the generalization of the BD decoding is the *list decoding*, first introduced by [Eli57] and [Woz58]. Fix a decoding radius τ , a list decoder outputs for any $\mathbf{y} \in \Sigma^n$ the list of all codewords which belong to the ball $\mathcal{B}^{(\tau)}(\mathbf{y})$ or a decoding failure. However since list decoding is beyond the purposes of this work, we do not go further into details of this technique. We refer to [GRS19, Section 7.2] for more details about this topic.

2.2 Reed-Solomon Codes

Reed Solomon (RS) codes constitute a very popular family of linear codes thanks to their remarkable properties: they are MDS and their algebraic structure allows the construction of efficient BMD decoders. They were first introduced by I. S. Reed and G. Solomon in 1960 [RS60] and they are still very common in practical applications: *e.g.* storage devices (CD, DVD, Blu-Ray Disc), barcodes and QR-codes, wireless and mobile communications, satellite communications, digital television (DVB), etc...

Definition 2.2.1 (Reed-Solomon code). Let $k \leq n \leq q$ and $\{\alpha_1, \dots, \alpha_n\}$ be a set of pairwise distinct elements of the field \mathbb{F}_q , called *evaluation points*. A *Reed-Solomon code* is defined by

$$\mathcal{C}_{RS}(n, k) := \{(f(\alpha_1), \dots, f(\alpha_n)) \mid f \in \mathbb{F}_q[x], \deg(f) \leq k - 1\}$$

We now recall the classic communication scenario described at the beginning of the Section 2.1. We observe that we can interpret any message $\mathbf{m} = (m_0, \dots, m_{k-1}) \in \mathbb{F}_q^k$ as a polynomial f of degree $\deg(f) \leq k - 1$ whose coefficients are the components of the message, *i.e.* $f = \sum_{i=0}^{k-1} m_i x^i$. A codeword of $\mathcal{C}_{RS}(n, k)$ is then obtained by *evaluating* the corresponding polynomial at the n evaluation points $\{\alpha_1, \dots, \alpha_n\}$. Formally, we can define an *encoding map*

$$\begin{aligned} E_{RS} : \mathbb{F}_q[x]/x^k &\longrightarrow \mathbb{F}_q^n \\ f &\longmapsto (f(\alpha_1), \dots, f(\alpha_n)) \end{aligned}$$

The RS code $\mathcal{C}_{RS}(n, k)$ is a *linear code* of length n and dimension k . Indeed, the encoding function defined above is linear. It is also injective: if $f(\alpha_1) = \dots = f(\alpha_n) = 0$, then f would have more roots than its degree and so it is the zero polynomial. Therefore the dimension of $\mathcal{C}_{RS}(n, k)$ is k .

Theorem 2.2.1 (RS is MDS). *RS codes are MDS, i.e. they attain the Singleton bound (2.1).*

Proof. Since $\mathcal{C}_{RS}(n, k)$ is a linear code and satisfies the Singleton bound (2.1) it suffices to show that the minimum distance d of the code is $d \geq n - k + 1$. Let $\mathbf{c}_1 = (f(\alpha_1), \dots, f(\alpha_n))$ and $\mathbf{c}_2 = (g(\alpha_1), \dots, g(\alpha_n))$ be two distinct codewords of $\mathcal{C}_{RS}(n, k)$. Note that $d(\mathbf{c}_1, \mathbf{c}_2) = |\{1 \leq i \leq n \mid (f - g)(\alpha_i) \neq 0\}| = w(f - g)$. Now, since the polynomial $f - g$ has $\deg(f - g) \leq k - 1$ it has at most $k - 1$ roots, hence $d(\mathbf{c}_1, \mathbf{c}_2) \geq n - (k - 1)$. Therefore, $d \geq n - (k - 1)$ which implies the claim. \square

Generator and parity check matrix of an RS code. A generator matrix of $\mathcal{C}_{RS}(n, k)$ is

$$G = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_n^{k-1} \end{pmatrix} \quad (2.2)$$

Note that it is the transposed of the *Vandermonde* matrix $V_{n,k} := (v_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq k}} = (\alpha_i^{j-1})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq k}}$.

Before talking about the dual of an RS code, we need to introduce a generalization of this family of codes.

Definition 2.2.2 (Generalized RS codes.). Let $\{\alpha_1, \dots, \alpha_n\}$ be pairwise distinct elements of the field \mathbb{F}_q and v_1, \dots, v_n nonzero elements¹ of \mathbb{F}_q (called *column multipliers*). A *generalized RS code* (shortly GRS) is

$$\mathcal{C}_{GRS}(n, k) := \{(v_1 f(\alpha_1), \dots, v_n f(\alpha_n)) \mid f \in \mathbb{F}_q[x], \deg(f) \leq k\}.$$

We observe that since $v_i \neq 0$ for any $1 \leq i \leq n$, the map

$$\begin{aligned} \mathbb{F}_q^n &\longrightarrow \mathbb{F}_q^n \\ (y_1, \dots, y_n) &\longmapsto (v_1 y_1, \dots, v_n y_n) \end{aligned}$$

is a bijective, distance-preserving transformation, *i.e.* an *isometry*. This map transforms $\mathcal{C}_{RS}(n, k)$ into $\mathcal{C}_{GRS}(n, k)$ and so the two codes have the same dimension and minimum distance. Therefore GRS codes are also MDS.

A GRS code $\mathcal{C}_{GRS}(n, k)$ is also linear and a generator matrix is obtained by multiplying on the right $G = V_{n,k}^T$ (see equation (2.2)) by the diagonal matrix whose diagonal elements are the column multipliers of the code.

The dual of an $[n, k]$ -GRS code is an $[n, n - k]$ -GRS code (see for instance [Rot06, Proposition 5.2] or the notes [Hal12]). Moreover, a parity check matrix of $\mathcal{C}_{GRS}(n, k)$, or in other terms a generator matrix of its dual code is of the form

$$H_{GRS} = V_{n, n-k}^T \begin{pmatrix} v'_1 & 0 & \dots & 0 \\ 0 & v'_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & v'_n \end{pmatrix} \quad (2.3)$$

1. Note that v_1, \dots, v_n are not necessarily distinct.

where for any $1 \leq i \leq n$ the column multipliers are

$$v'_i := \frac{1}{v_i \prod_{\substack{1 \leq j \leq n \\ j \neq i}} (\alpha_i - \alpha_j)}.$$

We can conclude that the dual of an $[n, k]$ -RS code is an $[n, n - k]$ -GRS code with column multipliers $v'_i = \frac{1}{\prod_{\substack{1 \leq j \leq n \\ j \neq i}} (\alpha_i - \alpha_j)}$ and a parity check matrix is defined as in (2.3). Recall that $n \leq q$. In the special case $n = q$, it is possible to prove that the dual of an RS code is an RS code, *i.e.* $\mathcal{C}_{RS}(n, k)^\perp = \mathcal{C}_{RS}(n, n - k)$. Note that, the column multipliers of the parity check matrix of the code (see equation (2.3)) are all equal to 1.

2.2.1 Decoding RS codes

In this section we introduce some BMD decoders for RS codes. We also point out how the decoding of such a family of codes can be reduced to the rational function reconstruction (Definition 1.1.1).

Let $\mathcal{C}_{RS}(n, k)$ be an RS code over \mathbb{F}_q , with $k \leq n \leq q$ and evaluation points $\{\alpha_1, \dots, \alpha_n\}$. We want to construct a BMD decoder,

Problem 3. *BMD decoder for RS codes*

Input: $\mathcal{C}_{RS}(n, k)$ with $\{\alpha_1, \dots, \alpha_n\}$, $\mathbf{y} \in \mathbb{F}_q^n$ a received vector

Output: $f \in \mathbb{F}_q[x]$ with $\deg(f) \leq k - 1$ or “decoding failure”.

which for any received word \mathbf{y} returns

1. the only $f \in \mathbb{F}_q[x]$ with $\deg(f) \leq k - 1$ such that $(f(\alpha_1), \dots, f(\alpha_n)) \in \mathcal{B}^{\tau_0}(\mathbf{y})$, if $\mathcal{C}_{RS}(n, k) \cap \mathcal{B}^{\tau_0}(\mathbf{y}) \neq \emptyset$,
2. a decoding failure otherwise.

Note that this decoder (contrary to the general definition of decoders of Subsection 2.1.3), in the first case, returns exactly the polynomial corresponding to the message in \mathbb{F}_q^k instead of the codeword.

Interpolation with errors. First recall that given n pairwise distinct evaluation points $\{x_1, \dots, x_n\}$ in \mathbb{F}_q , where $n \leq q$, and given $y_1, \dots, y_n \in \mathbb{F}_q$, the *polynomial interpolation* (or simply interpolation) is the problem of reconstructing the unique polynomial $p \in \mathbb{F}_q[x]$ of degree at most $n - 1$ such that $p(x_i) = y_i$ for any $1 \leq i \leq n$. Informally speaking, it is the problem of recovering a polynomial from its evaluations. The *Lagrange interpolating polynomial* L is defined as

$$L(x) = \sum_{i=1}^n y_i l_i \prod_{j \neq i} (x - x_j) \quad (2.4)$$

where for any $1 \leq i \leq n$, $\ell_i = \prod_{j \neq i} \frac{1}{x_i - x_j}$.

We now consider a variation of the classic interpolation problem, in which we add some errors.

Definition 2.2.3 (Interpolation with errors). Fix some parameters n, τ, k, q , where $k, \tau \leq n \leq q$. Fix also $\{\alpha_1, \dots, \alpha_n\}$ pairwise distinct elements of \mathbb{F}_q . A satisfiable instance of the *interpolation with errors* (shortly IwE) problem is the vector $\mathbf{y} \in \mathbb{F}_q^n$ such that there exist

1. a polynomial $f \in \mathbb{F}_q[x]$, $\deg(f) \leq k - 1$,
2. an *error vector* $\mathbf{e} \in \mathbb{F}_q^n$ with error support $E := \{i \mid e_i \neq 0\}$, where $|E| \leq \tau$

which satisfy the following

$$\mathbf{y} = (f(\alpha_1), \dots, f(\alpha_n)) + \mathbf{e}. \quad (2.5)$$

IwE is the problem of finding a polynomial f as in (2.5) given an instance \mathbf{y} .

Note that (2.5) is equivalent to saying that $y_i = f(\alpha_i)$ if $i \notin E$ and $y_i \neq f(\alpha_i)$ otherwise. This clarifies the link with the interpolation problem. Therefore IwE is the problem of reconstructing a polynomial of bounded degree given its evaluations where some of them could be erroneous or corrupted. In Subsection 4.1.1 we will generalize the interpolation with errors problem to rational functions.

Remark 2.2.1. Note that \mathbf{y} is an instance of IwE with parameters $n, k, \tau_0, \{\alpha_1, \dots, \alpha_n\}$ if and only if $\mathcal{C}_{RS}(n, k) \cap \mathcal{B}^{(\tau_0)}(\mathbf{y}) \neq \emptyset$. Recall that by Theorem 2.1.1 if $\mathcal{C}_{RS}(n, k) \cap \mathcal{B}^{(\tau_0)}(\mathbf{y}) \neq \emptyset$, then $|\mathcal{C}_{RS}(n, k) \cap \mathcal{B}^{(\tau_0)}(\mathbf{y})| = 1$.

Therefore, we point out that IwE with instance \mathbf{y} coincides with decoding \mathbf{y} . Indeed, note that the main aim of both problems is to recover the polynomial f given its evaluations, where some are wrong. 💡

The state of the art of decoding techniques for constructing efficient BMD decoders can be classified according to two main paradigms: an *interpolation-based* and a *syndrome-based* approach. For both of them, we can reduce the decoding problem to the rational function reconstruction. More specifically, the interpolation-based technique can be reduced to a Cauchy interpolation while the syndrome-based one can be reduced to the Padé approximation (see Section 1.1).

For all the rest of this section we fix an RS code $\mathcal{C}_{RS}(n, k)$ over \mathbb{F}_q with evaluation points $\{\alpha_1, \dots, \alpha_n\}$.

We now introduce the following polynomials, which are relevant for both the decoding techniques that we will describe later. Let $\mathbf{y} = (f(\alpha_1), \dots, f(\alpha_n)) + \mathbf{e}$ be a received vector with error support $E = \{i \mid e_i \neq 0\}$. Let $\varepsilon = |E|$ be the number of errors.

Definition 2.2.4 (Error locator and error evaluator polynomial).

- The *error locator polynomial* Λ is a monic polynomial whose roots are the erroneous evaluations, *i.e.* $\Lambda = \prod_{i \in E} (x - \alpha_i)$. Note that its degree is exactly the number of errors.
- The *error evaluator polynomial* Γ is determined by the Lagrange interpolating polynomial (see (2.4)) of all the components of the error vector, *i.e.*

$$\Gamma = - \sum_{i=1}^n e_i \ell_i \prod_{j \neq i} (x - \alpha_j) = - \sum_{i \in E} e_i \ell_i \prod_{j \in E \setminus \{i\}} (x - \alpha_j)$$

where $\ell_i = \prod_{j \neq i} \frac{1}{\alpha_i - \alpha_j}$.

First, we observe that these two polynomials do not have any common root and so they are coprime. Moreover, they are related by the following relation

$$\Gamma(x) = - \sum_{i \in E} e_i \ell_i \frac{\Lambda(x)}{x - \alpha_i}.$$

We now denote $\mathcal{G} := \prod_{i=1}^n (x - \alpha_i)$ and we consider the Lagrange interpolating polynomial Y of the components of the received word, *i.e.* $Y = \sum_{i=1}^n y_i \ell_i \prod_{j \neq i} (x - \alpha_j)$, where $\ell_i = \prod_{j \neq i} \frac{1}{\alpha_i - \alpha_j}$ for all $1 \leq i \leq n$. The following result shows the link between all the polynomials introduced so far.

Lemma 2.2.2. $\Lambda(f - Y) = \Gamma \mathcal{G}$.

Proof. Since $\mathbf{y} = (f(\alpha_1), \dots, f(\alpha_n)) + \mathbf{e}$, then we have that

$$f - Y = - \sum_{i=1}^n e_i \ell_i \prod_{j \neq i} (x - \alpha_j) = - \sum_{i \in E} e_i \ell_i \frac{\mathcal{G}}{x - \alpha_i}$$

therefore

$$\Lambda(f - Y) = - \sum_{i \in E} e_i \ell_i \frac{\Lambda}{x - \alpha_i} \mathcal{G} = \Gamma \mathcal{G}.$$

□

Interpolation-based decoding technique. By the previous lemma we can deduce that

$$\Lambda f = \Lambda Y \pmod{\mathcal{G}} \tag{2.6}$$

or equivalently, that the following condition on the evaluations holds,

$$\Lambda(\alpha_i) f(\alpha_i) = \Lambda(\alpha_i) y_i, \text{ for all } 1 \leq i \leq n.$$

However the equation (2.6) is nonlinear. The classic technique for building a decoder consists in the study of the following linear equation

$$\varphi = \lambda Y \text{ mod } \mathcal{G}, \quad (2.7)$$

considering all the solutions $(\varphi, \lambda) \in \mathbb{F}_q[x]^2$ which satisfy the degree constraints

$$\deg(\varphi) \leq \varepsilon + k - 1, \deg(\lambda) \leq \varepsilon. \quad (2.8)$$

The equation (2.7) with degree constraints (2.8) is also called *key equation*. We observe that this is exactly the rational function reconstruction (Problem 1) with degree constraints $\varepsilon + k$, $\varepsilon + 1$ and instance Y . More specifically, since the polynomial \mathcal{G} is of the form $\prod_{i=1}^n (x - \alpha_i)$, this is the *weaker*² form of the Cauchy interpolation (see Section 1.4).

Note that $(\Lambda f, \Lambda)$ is a solution of this problem. Moreover, by the uniqueness results of Section 1.1 (see Remark 1.1.2) if

$$\deg(G) = n \geq (\varepsilon + k) + (\varepsilon + 1) - 1 = k + 2\varepsilon \iff \varepsilon \leq \frac{n - k}{2} \quad (2.9)$$

then RFR (Problem 1) with input \mathcal{G}, Y and the degree constraints $\varepsilon + k$, $\varepsilon + 1$ admits a unique solution: any solution is a polynomial multiple of a minimal one. In Lemma 2.2.3 we prove that this minimal solution is exactly $(\Lambda f, \Lambda)$.

Note that this result is coherent with Theorem 2.1.1. Indeed, if $\varepsilon \leq \tau_0$, then $|\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^{(\tau_0)}(Y)| = 1$, meaning that we can uniquely decode the received word. Therefore, in this case the uniqueness of the solution of RFR is strictly related to unique decoding.

All the existing BMD-decoders basically differ in the algorithm chosen to solve RFR. The *Welch-Berlekamp* decoder [BW86] is based on the study of the homogeneous linear system related to the key equation (2.7), while the *Gao's* decoder [Gao03] (Algorithm 3) is based on the Extended Euclidean Algorithm³(see Algorithm 2).

Correctness of Algorithm 3. Recall that we are considering $\mathcal{C}_{RS}(n, k)$ with evaluation points $\{\alpha_1, \dots, \alpha_n\}$.

Lemma 2.2.3. *Let \mathbf{y} be a received word for which $\mathcal{C}_{RS}(n, k) \cap \mathcal{B}^{\tau_0}(\mathbf{y}) \neq \emptyset$. Let f the polynomial corresponding to the codeword in $\mathcal{C}_{RS}(n, k) \cap \mathcal{B}^{\tau_0}(\mathbf{y})$ and let $(\varphi, \lambda) = \mathbf{RFR}_{EEA}(G, Y, \tau_0 + k)$ be the minimal solution of RFR such that λ is monic.*

Then $(\varphi, \lambda) = (\Lambda f, \Lambda)$, where Λ is the error locator polynomial.

2. In the literature, the Cauchy interpolation usually refers to the rational function reconstruction as in the equation (1.2). Here we are considering its weaker linear version, dropping the gcd condition.

3. S. Gao in [Gao03] also proposed another version of the algorithm based on the half-gcd, which speeds up the computations improving the efficiency of the algorithm (see Subsection 1.1.1)

Algorithm 3: Interpolation-based BMD decoder based on EEA

Input : $\mathcal{C}_{RS}(n, k)$ with $\{\alpha_1, \dots, \alpha_n\}$, \mathbf{y} a received word

Output: $f \in \mathbb{F}_q[x]$ such that $d(\mathbf{y}, (f(\alpha_1), \dots, f(\alpha_n))) \leq \tau_0$, if
 $|\mathcal{C}_{RS}(n, k) \cap \mathcal{B}^{\tau_0}(\mathbf{y})| \neq 0$;
 otherwise a “decoding failure”.

- 1 Compute the Lagrange interpolating polynomial Y of the components of the received vector;
 - 2 $(\varphi, \lambda) = \text{RFR}_{\text{EEA}}(\mathcal{G}, Y, \tau_0 + k)$;
 - 3 perform the Euclidean division $\varphi = \lambda f + r$;
 - 4 **if** $r = 0$ **and** $\deg(f) \leq k - 1$ **then**
 - 5 **return** f
 - 6 **else**
 - 7 **return** “decoding failure”
-

Proof. Let $(\varphi, \lambda) = \text{RFR}_{\text{EEA}}(\mathcal{G}, Y, \tau_0 + k)$ with λ monic. Then, for any $1 \leq j \leq n$, $\varphi(\alpha_j) = y_j \lambda(\alpha_j)$. As already remarked, also $(\Lambda f, \Lambda)$ is a solution, *i.e.* for any $1 \leq j \leq n$, $\Lambda(\alpha_j) f(\alpha_j) = y_j \Lambda(\alpha_j)$ therefore, by multiplying the former equation by $\Lambda(\alpha_j)$ and the latter by $\lambda(\alpha_j)$ and then by subtracting them we get

$$\Lambda(\alpha_j)[\varphi(\alpha_j) - f(\alpha_j)\lambda(\alpha_j)] = 0 \quad (2.10)$$

Note that for $j \notin E$, since $\Lambda(\alpha_j) \neq 0$, then $\varphi(\alpha_j) - f(\alpha_j)\lambda(\alpha_j) = 0$. Therefore since the polynomial $\varphi - f\lambda$ has degree at most $\tau_0 + k - 1$ and $n - |E| \geq n - \tau_0 = \tau_0 + k$ roots it is the zero polynomial. Hence $\varphi = \lambda f$. Now, since for any $1 \leq j \leq n$, we have that $\lambda(\alpha_j) f(\alpha_j) = \varphi(\alpha_j) = y_j \lambda(\alpha_j)$, then all the erroneous evaluations α_j for $j \in E$ are roots of λ and so Λ divides λ . In conclusion, since (φ, λ) is the minimal solution with λ monic then $(\varphi, \lambda) = (\Lambda f, \Lambda)$. \square

Another useful result for the proof of the correctness of Algorithm 3 is the following.

Lemma 2.2.4. *Let $\mathbf{y} \in \mathbb{F}_q^n$ a received word, Y the Lagrange interpolating polynomial of the components of \mathbf{y} and $(\varphi, \lambda) = \text{RFR}_{\text{EEA}}(\mathcal{G}, Y, \tau_0 + k)$, where λ is monic. Then if $\lambda | \varphi$ and $\deg(f) \leq k - 1$, where $f := \varphi/\lambda$, then the Hamming distance satisfies $d((f(\alpha_1), \dots, f(\alpha_n)), \mathbf{y}) \leq \tau_0$.*

Proof. Since for any $1 \leq i \leq n$ we have that $\varphi(\alpha_i) = y_i \lambda(\alpha_i)$ and since $\varphi = \lambda f$, then $\lambda(\alpha_i)[f(\alpha_i) - y_i] = 0$. Therefore for $i \in \{1 \leq i \leq n \mid f(\alpha_i) \neq y_i\} = E$ then $\lambda(\alpha_i) = 0$. Note that $|E| \leq \deg(\lambda) \leq \tau_0$. \square

This proves the correctness of Algorithm 3 since, given a received word \mathbf{y} ,

- if $|\mathcal{C}_{RS}(n, k) \cap \mathcal{B}^{\tau_0}(\mathbf{y})| = 0$, by contrapposing Lemma 2.2.4, the solution $(\varphi, \lambda) = \text{RFR}_{\text{EEA}}(\mathcal{G}, Y, \tau_0 + k)$ either does not satisfy the divisibility criteria or leads to a poly-

nomial of degree greater than $k - 1$. In this case the algorithm outputs a decoding failure.

- if $|\mathcal{C}_{RS}(n, k) \cap \mathcal{B}^{\tau_0}(\mathbf{y})| = 1$, by Remark 2.2.1 \mathbf{y} is an instance of IwE and by Lemma 2.2.3 the solution $(\varphi, \lambda) = \text{RFR}_{\text{EEA}}(G, Y, \tau_0 + k)$ is a scalar multiple of $(\Lambda f, \Lambda)$, where f is the polynomial related to the codeword in the ball $\mathcal{B}^{\tau_0}(\mathbf{y})$. Hence (φ, λ) satisfies the divisibility and the degree condition of step 4 of the algorithm and so the decoder outputs the polynomial f .

Syndrome-based decoding technique. In this thesis we especially focus on the interpolation-based decoding technique for RS and later for interleaved RS codes. Nevertheless, for the sake of completeness, in this paragraph we also introduce the syndrome-based approach.

For the syndrome-based decoding technique it is important to suppose that the evaluation points of $C_{RS}(n, k)$ are all nonzero.

First we precise some notations and notions that we use here. The *reciprocal* polynomial of a polynomial p of degree t is the polynomial $x^t p(1/x)$. Since we often know bounds on the polynomial degrees instead of the real degrees, we consider a generalization of this notion.

Definition 2.2.5 (Reciprocal polynomial). Given a polynomial $p \in \mathbb{F}_q[x]$ with $\deg(p) \leq t$, the *reciprocal* of p for the degree t is the polynomial $\text{rev}_t(p) := x^t p(1/x)$.

In order to derive a syndrome-based key equation, given the received vector

$$\mathbf{y} = (f(\alpha_1), \dots, f(\alpha_n)) + \mathbf{e},$$

we

1. introduce the *syndrome polynomial* $S(x) = \sum_{l=0}^{n-k-1} s_l x^l$, a polynomial whose coefficients are the components of a syndrome;
2. prove that the syndrome polynomial can be seen as the truncated series expansion of the rational function $\frac{\text{rev}_{n-1}(Y)}{\text{rev}_n(\mathcal{G})}$ (Lemma 2.2.5);
3. use the previous result to link the error locator Λ , the error evaluator polynomial Γ and the syndrome polynomial S by the following

$$\text{rev}_\varepsilon(\Lambda)S = -\text{rev}_{\varepsilon-1}(\Gamma) \bmod x^{n-k} \quad (2.11)$$

Recall that we can easily compute a *syndrome*, i.e. $\mathbf{s} = H\mathbf{y}^T$, where H is a parity check matrix of our code $C_{RS}(n, k)$ (as in equation (2.3)). In particular, by expanding the matrix product we get for any $0 \leq l \leq n - k - 1$,

$$s_l = \sum_{j=1}^n y_j v_j' \alpha_j^l, \quad (2.12)$$

where for any j

$$v'_j = \frac{1}{\prod_{j \neq i} (\alpha_i - \alpha_j)} = \ell_j.$$

As previously remarked, syndromes only depend on the error vector, hence for any $0 \leq l \leq n - k - 1$

$$s_l = \sum_{j \in E} e_j \ell_j \alpha_j^l \quad (2.13)$$

Note that the syndrome polynomial $S(x)$, is of the form

$$S(x) = \sum_{l=0}^{n-k-1} s_l x^l = \sum_{l=0}^{n-k-1} x^l \sum_{j \in E} e_j \ell_j \alpha_j^l = \sum_{j \in E} e_j \ell_j \sum_{l=0}^{n-k-1} (x \alpha_j)^l \quad (2.14)$$

and since

$$\sum_{l=0}^{n-k-1} (x \alpha_j)^l = \frac{1 - x^{n-k} \alpha_j^{n-k}}{1 - x \alpha_j}$$

we finally get

$$S(x) = \sum_{j \in E} \frac{e_j \ell_j}{1 - x \alpha_j} \text{ mod } x^{n-k}. \quad (2.15)$$

We are now ready for the following result.

Lemma 2.2.5. $\frac{rev_{n-1}(Y)}{rev_n(\mathcal{G})} = S(x) \text{ mod } x^{n-k}$.

Proof. First observe that $rev_{n-1}(Y) = \sum_{i=1}^n y_i \ell_i \prod_{j \neq i} (1 - x \alpha_j)$ and $rev_n(\mathcal{G}) = \prod_{i=1}^n (1 - x \alpha_i)$. The claim follows by noticing that $\gcd(rev(\mathcal{G}), x) = 1$, $\frac{rev_{n-1}(Y)}{rev_n(\mathcal{G})} = \sum_{i=1}^n \frac{y_i \ell_i}{1 - x \alpha_i}$ and by (2.15). \square

If we now consider the reciprocal polynomials of the two members of $\Lambda Y = \Lambda f - \Gamma \mathcal{G}$, of Lemma 2.2.2, we get

$$rev_{\varepsilon+n-1}(\Lambda Y) = rev_{\varepsilon+k-1}(\Lambda f) x^{n-k} - rev_{\varepsilon+n-1}(\Gamma \mathcal{G}) = -rev_{\varepsilon+n-1}(\Gamma \mathcal{G}) \text{ mod } x^{n-k}.$$

Note that $rev_{\varepsilon+n-1}(\Lambda Y) = rev_{\varepsilon}(\Lambda) rev_{n-1}(Y)$ and $rev_{\varepsilon+n-1}(\Gamma \mathcal{G}) = rev_{\varepsilon-1}(\Gamma) rev_n(\mathcal{G})$ and by Lemma 2.2.5 we finally obtain

$$rev_{\varepsilon}(\Lambda) S = -rev_{\varepsilon-1}(\Gamma) \text{ mod } x^{n-k} \quad (2.16)$$

In this case, decoding is reduced to the problem of finding the polynomials (γ, λ) such that

$$\gcd(\gamma, \lambda) = 1, \quad \gamma = \lambda S \text{ mod } x^{n-k}, \quad \deg(\gamma) < \varepsilon, \quad \deg(\lambda) < \varepsilon + 1 \quad (2.17)$$

The congruence of this equation is the so-called syndrome-based key equation. Note that this is exactly the Padé approximation (see (1.1)) with degree constraints $\varepsilon, \varepsilon + 1$ and instance S .

By Remark 1.1.2 if

$$n - k \geq 2\varepsilon \iff \varepsilon \leq \frac{n - k}{2},$$

then we can uniquely reconstruct the rational solution. Even in this case all syndrome-based BMD decoders depend on the algorithm chosen for the Padé Approximation. For instance, we can construct a decoder based on the Extended Euclidean Algorithm [SKHN75] or on the Berlekamp-Massey algorithm [Ber68], which is the most commonly used in practice.

2.3 Interleaved Reed-Solomon code

Interleaving is an encoding technique used in the setting of burst errors. In this section we show how this technique applied to RS codes allows the construction of interesting decoders [KL97, KY98, Kra03, BKY03, BMS04, SSB07, SSB09, SSB10, PR17] which decode beyond the error correction capability of the code.

Definition 2.3.1 (Interleaved Reed-Solomon codes). Let $l \geq 1$ and $k_1, \dots, k_l \leq n \leq q$ and $\{\alpha_1, \dots, \alpha_n\}$ be pairwise distinct evaluation points in \mathbb{F}_q . The l -Interleaved Reed-Solomon (IRS) code is

$$\mathcal{C}_{IRS}(n, k_1, \dots, k_l) := \left\{ \left(\begin{array}{c} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_l \end{array} \right) \middle| \mathbf{c}_i \in \mathcal{C}_{RS}(n, k_i), 1 \leq i \leq l \right\} \in (\mathbb{F}_q)^{l \times n}$$

An *homogeneous* IRS code is an IRS code whose l constituent RS codes all have the same dimension $k_1 = \dots = k_l = k$. We denote it simply by $\mathcal{C}_{IRS}(n, k)$. In this work we focus on homogeneous IRS codes.

Remark 2.3.1. Note that by the definition of RS codes (Definition 2.2.1) we can see an IRS code $\mathcal{C}_{IRS}(n, k)$ as the evaluation of a (column) *vector* of polynomials of bounded degrees, *i.e.*

$$\mathcal{C}_{IRS}(n, k) = \left\{ (\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)) \mid \mathbf{f} \in \mathbb{F}_q[x]^{l \times 1}, \deg(\mathbf{f}) \leq k - 1 \right\}$$

Moreover for any codeword $C = (\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)) \in \mathcal{C}_{IRS}(n, k)$, we can interpret any component $\mathbf{f}(\alpha_i) \in \mathbb{F}_q^{l \times 1}$ as an element of \mathbb{F}_q^l . In this way, $\mathcal{C}_{IRS}(n, k) \subseteq \mathbb{F}_q^n$ and it is easy to see that it is an $[n, k, n - k + 1]$ -linear code over \mathbb{F}_q^l . The error correction capability of $\mathcal{C}_{IRS}(n, k)$ is then $\tau_0 = \lfloor \frac{n-k}{2} \rfloor$.

💡

The interpretation of any component of any codeword as a vector on an extension field is crucial for the error model comprehension.

2.3.1 Decoding IRS codes

Error model. We consider a *burst-error* channel as in [Kra03]. In this simplified model, the transmitted codeword is a matrix over a certain alphabet and the channel introduce some errors, called *phased burst* (simply bursts) that corrupt *columns* of the codeword. Therefore, in the specific case of IRS codes the action of these channels can be described as follows

$$C \in \mathcal{C}_{IRS}(n, k) \subseteq \mathbb{F}_q^{l \times n} \longrightarrow \boxed{\text{channel}} \longrightarrow Y = C + \Xi \in \mathbb{F}_q^{l \times n}$$

$$\begin{array}{c} \uparrow \\ \Xi \in \mathbb{F}_q^{l \times n} \\ \text{error matrix} \end{array}$$

where $E := \{i \in \{1, \dots, n\} \mid \Xi_{*,i} \neq \mathbf{0}\}$ is the error support.

In this case burst errors alter columns of the received matrix (see Figure 2.4).

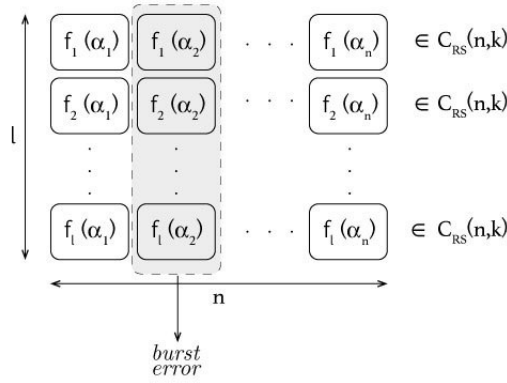


Figure 2.4: Received word of an IRS code under the burst error model.

Now, since $\mathbb{F}_q^l \simeq \mathbb{F}_{q^l}$, by using the field extension representation we can model this channel by a q^l -ary Symmetric Channel (see Example 2.1.2) over the alphabet \mathbb{F}_{q^l} .

Simultaneous interpolation with errors. Fix a received word $Y = (\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)) + \Xi$, where $\mathbf{f} \in \mathbb{F}_q[x]^{l \times 1}$, $\deg(\mathbf{f}) \leq k - 1$ and Ξ the error matrix with error support $E := \{i \in \{1, \dots, n\} \mid \Xi_{*,i} \neq \mathbf{0}\}$, $|E| \leq \tau$ for a given τ . We now observe that the problem of decoding Y coincides with the problem of reconstructing a *vector* of polynomials with bounded degree given its evaluations, where some of them could be erroneous. This is exactly the vector generalization of the interpolation with errors (Definition 2.2.3) which is called *simultaneous interpolation with errors*.⁴

4. In [BKY03], authors introduced the *simultaneous polynomial reconstruction*, a particular case of SIwE for which the number of errors is known, *i.e.* $|E| = \tau$.

Definition 2.3.2 (Simultaneous interpolation with errors). Fix some parameters l, n, τ, k, q , where $k, \tau \leq n \leq q$ and $l \geq 1$. Fix also $\{\alpha_1, \dots, \alpha_n\}$ pairwise distinct elements of \mathbb{F}_q . A satisfiable instance of the *simultaneous interpolation with errors* (shortly SIwE) problem is the matrix $Y \in \mathbb{F}_q^{l \times n}$ such that there exist

1. a vector of polynomials $\mathbf{f} \in \mathbb{F}_q[x]^{l \times 1}$, $\deg(\mathbf{f}) \leq k - 1$,
2. an error matrix $\Xi \in \mathbb{F}_q^{l \times n}$ with error support $E := \{i \mid \Xi_i \neq \mathbf{0}\}$, where $|E| \leq \tau$

which satisfy the following

$$Y = (\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)) + \Xi. \quad (2.18)$$

SIwE is the problem of finding a vector of polynomials \mathbf{f} as in (2.18) given an instance Y .

In Subsection 4.1.1 we will generalize this problem to vector of rational functions, the *simultaneous Cauchy interpolation with errors*.

Remark 2.3.2. Note that Y is an instance of SIwE with parameters $n, k, \tau, \{\alpha_1, \dots, \alpha_n\}$ if and only if $\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^\tau(Y) \neq \emptyset$. 💡

We now observe that, since any row of an IRS codeword is an RS codeword, we could construct a BMD decoder for IRS codes which decode any row of the received word separately as we saw in Subsection 2.2.1. Nevertheless, the interleaving technique applied to RS codes allows the construction of *partial* BD decoders [BKY03, SSB07, SSB09, SSB10, PR17], in the sense that they fail for a few error patterns of any weight beyond the error correction capability of the code τ_0 .

As for classic RS codes, we can distinguish two main approaches to construct such decoders: an interpolation-based and a syndrome-based one. In this section, we especially focus on the first one and on a decoder derived from it, since we generalize this approach for solving the simultaneous Cauchy interpolation with errors (Chapter 4).

By the way, it important to remark that the most efficient decoder for IRS is a syndrome-based one [SSB07, SSB09, SSB10]. It is based on a generalized version of the Berlekamp-Massey [Ber68, Mas69] algorithm for decoding RS codes and it has a similar arithmetic complexity [SSB09].

We now fix an IRS code $\mathcal{C}_{IRS}(n, k)$ over \mathbb{F}_q with evaluation points $\{\alpha_1, \dots, \alpha_n\}$ and a received word $Y = (\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)) + \Xi$, of error support $E := \{i \mid \Xi_{*,i} \neq \mathbf{0}\}$, $\varepsilon := |E|$. We also denote the received matrix by $Y = (y_{i,j})_{\substack{1 \leq i \leq l \\ 1 \leq j \leq n}}$.

For any $1 \leq i \leq l$, let Ψ_i be the Lagrange interpolating polynomial of the i -th row of Y , i.e. $\Psi_i = \sum_{j=1}^n Y_{i,j} \ell_j \prod_{k \neq j} (x - \alpha_k)$ where $\ell_j = \prod_{k \neq j} \frac{1}{\alpha_j - \alpha_k}$ for any $1 \leq j \leq n$. We denote $\Psi = (\Psi_1, \Psi_2, \dots, \Psi_l)^T \in \mathbb{F}_q^{l \times 1}$. Recall that $\mathcal{G} = \prod_{i=1}^n (x - \alpha_i)$.

Note that

$$\begin{aligned} \Lambda \mathbf{f} = \Lambda \Psi \pmod{\mathcal{G}} &\iff \Lambda(\alpha_j) \mathbf{f}(\alpha_j) = \Lambda(\alpha_j) Y_{*,j}, \text{ for any } 1 \leq j \leq n \\ &\iff \Lambda(\alpha_j) f_i(\alpha_j) = \Lambda(\alpha_j) y_{i,j}, \text{ for any } 1 \leq i \leq l, 1 \leq j \leq n \end{aligned}$$

Indeed, it suffices to observe that if $j \in E$ then $\Lambda(\alpha_j) = 0$, otherwise $y_{i,j} = f_i(\alpha_j)$ for any $1 \leq j \leq n$. Therefore, as for classic RS codes, since this equation is not linear in the unknowns \mathbf{f} and Λ , we study the linear problem

$$\varphi = \lambda \Psi \pmod{\mathcal{G}}, \quad \deg(\varphi) \leq \varepsilon + k - 1, \quad \deg(\lambda) \leq \varepsilon \quad (2.19)$$

or equivalently, for any $1 \leq i \leq l$ and $1 \leq j \leq n$,

$$\varphi_i(\alpha_j) = \lambda(\alpha_j) \Psi_i(\alpha_j), \quad \deg(\varphi_i) \leq \varepsilon + k - 1, \quad \deg(\lambda) \leq \varepsilon \quad (2.20)$$

Note that this problem coincides with the simultaneous rational function reconstruction⁵ (Problem 2) with input \mathcal{G}, Ψ and degree constraints $\varepsilon + k, \varepsilon + 1$.

Remark 2.3.3. We denote by $\mathcal{S}_{Y,\varepsilon,k} := \{(\varphi_1, \dots, \varphi_l, \lambda) \text{ satisfying (2.20)}\}$. Note that we write $\mathcal{S}_{Y,\varepsilon,k}$ to stress the dependency on the received matrix Y , on k and ε . If we consider the homogeneous linear system related to (2.20), we observe that the set of solutions $\mathcal{S}_{Y,\varepsilon,k}$ is the kernel of the matrix

$$M_{Y,\varepsilon,k} = \left(\begin{array}{cccc|c} V_{n,k+\varepsilon} & & & & -D_1 V_{n,\varepsilon+1} \\ & V_{n,k+\varepsilon} & & & -D_2 V_{n,\varepsilon+1} \\ & & \ddots & & \vdots \\ & & & V_{n,k+\varepsilon} & -D_l V_{n,\varepsilon+1} \end{array} \right) \quad (2.21)$$

where $V_{n,d}$ denotes the Vandermonde matrix whose entries are $(\alpha_i^{j-1})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq d}}$ and for any $1 \leq i \leq l$, D_i is the diagonal matrix whose elements on the diagonal are $y_{i,1}, \dots, y_{i,n}$. \spadesuit

We now recall (1.6), which derives from the fact that we can apply RFR to solve SRFR. We also recall that in this case $a_1 = \dots = a_l = \mathcal{G} = \prod_{i=1}^n (x - \alpha_i)$, $N_1 = \dots = N_l = N = \varepsilon + k$ and $D = \varepsilon + 1$. So we can conclude that if

$$\deg(\mathcal{G}) = n \geq (\varepsilon + k) + (\varepsilon + 1) - 1 = 2\varepsilon + k \iff \varepsilon \leq \frac{n - k}{2}$$

this problem admits a unique solution. In this case, we can prove with the same technique of Lemma 2.2.3 that all the solutions are polynomial multiples of a minimal one, which is exactly $(\Lambda \mathbf{f}, \Lambda)$. Indeed, recall that $\tau_0 = \frac{n-k}{2}$ is the error correction capability of the code, and by Theorem 2.1.1 then $|\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^{(\tau_0)}(Y)| = 1$. Therefore, also in this case, the uniqueness of the solution of SRFR is strictly related to the uniqueness of the decoding.

We now ask what happens if we consider the reduced number of points of (1.7) derived

5. Note that we can equivalently consider the SRFR problem for row or column vectors

from the common denominator property of SRFR. Specifically, if

$$\underbrace{\sum_{i=1}^l \deg(a_i)}_{ln} = \underbrace{\sum_{i=1}^l N_i}_{l(\varepsilon+k)} + \underbrace{D}_{\varepsilon+1} - 1 \iff n = (k + \varepsilon) + (\varepsilon + 1 - 1)/l$$

$$\iff \varepsilon = \frac{l(n - k)}{l + 1}.$$

In [BKY03, BMS04, SSB07, SSB09, SSB10], authors proved that we can construct some decoders, which can *uniquely* correct *almost all* errors up to the decoding radius

$$\tau_{IRS} = \left\lfloor \frac{l(n - k)}{l + 1} \right\rfloor. \quad (2.22)$$

Note that for $l \geq 1$ then $\tau_{IRS} \geq \tau_0$, meaning that these decoders could correct beyond the error correction capability of the code.

In what follows we “reinterpret” some results of [BKY03] thanks to our study of the generalized rational function case (see Subsection 4.1.1).

In [BKY03] authors first introduced a decoder which assumes to know exactly the number of errors that occurred during the transmission. Then, they generalized their results to the construction of a more general decoder which is suited to a channel’s model for which they estimated the expected number of errors. Here we define a more general decoder (Algorithm 4) fitting to the error model previously described.

An interpolation-based partial decoder. As previously observed, we can reduce the decoding problem of IRS to SRFR in the interpolation version (Problem 2) with input $G = \prod_{i=1}^n (x - \alpha_i), Y$ (where Y is the received word) and degree constraints $\tau_{IRS} + k, \tau_{IRS} + 1$, *i.e.*

$$\varphi(\alpha_j) = \lambda(\alpha_j)Y_{*,j}, \quad \deg(\varphi) \leq \tau_{IRS} + k - 1, \quad \deg(\lambda) \leq \tau_{IRS}. \quad (2.23)$$

However, since we are beyond the error correction capability of the code, the uniqueness of the solution to the decoding problem is not always guaranteed. In what follows we denote, as in Remark 2.3.3, by $S_{Y, \tau_{IRS}, k}$ the set of solutions of SRFR (2.23). Recall that this coincides with the kernel of $M_{Y, \tau_{IRS}, k}$ as in (2.21).

Note that to compute a basis of the $\mathbb{F}_q[x]$ -module of solutions (step 3 of Algorithm 4) we can use the algorithm of [RS16] (see Section 1.2), which computes all the rows of a shifted row reduced basis, (the shift is $\mathbf{s} = (-\tau_{IRS} - k, \dots, -\tau_{IRS} - k, -\tau_{IRS} - 1)$) with negative row degrees. As seen in Subsection 1.3.2 this is a basis of the $\mathbb{F}_q[x]$ -module generated by the solutions in $S_{Y, \tau_{IRS}, k}$.

Algorithm 4: Partial BD decoder for IRS codes

Input : $\mathcal{C}_{IRS}(n, k)$ with $\{\alpha_1, \dots, \alpha_n\}$, Y received word
Output: $\mathbf{f} \in \mathbb{F}_q[x]^{l \times 1}$, $\deg(\mathbf{f}) \leq k - 1$ or a “decoding failure”.

```

1 Let  $\mathcal{M}$  be  $\mathbb{F}_q[x]$ -module generated by solutions in  $\mathcal{S}_{Y, \tau_{IRS}, k}$ ;
2 if  $\text{rank}(\mathcal{M}) = 1$  then
3   | find  $(\varphi, \lambda)$  a generator of  $\mathcal{M}$  scaled to obtain  $\lambda$  monic;
4   | perform Euclidean division  $\varphi = \lambda \mathbf{f} + r$ ;
5   | if  $r = 0$  and  $\deg(\mathbf{f}) \leq k - 1$  then
6   |   | return  $\mathbf{f}$ 
7   | else
8   |   | return “decoding failure”
9 else
10  | return “decoding failure”

```

Correctness of Algorithm 4. We now introduce some lemmas useful to prove the correctness of the algorithm. In particular, the following lemma is an adaptation of Theorem 1 of [BKY03] for a more general result. The proof of this algorithm is new and it is a specific case of Theorem 4.1.2, in which we prove the same result for the reconstruction of a vector of rational functions instead of polynomials.

Lemma 2.3.1. Fix $E \subseteq \{1, \dots, n\}$, and assume that $\varepsilon := |E|$ satisfies $\varepsilon \leq \tau_{IRS}$. Moreover, fix $\mathbf{f} \in \mathbb{F}_q[x]^l$, $\deg(\mathbf{f}) \leq k - 1$.

Consider the random matrix $Y = (y_{i,j})_{\substack{1 \leq i \leq l \\ 1 \leq j \leq n}}$ such that,

- if $j \in E$, then $Y_{*,j}$ is a uniformly distributed element in $\mathbb{F}_q^{l \times 1}$,
- if $j \notin E$, then $Y_{*,j} = \mathbf{f}(\alpha_j)$.

Then $\mathcal{S}_{Y, \tau_{IRS}, k}$ is spanned by elements of the form $(x^i \Lambda \mathbf{f}, x^i \Lambda)$, i.e.

$$\mathcal{S}_{Y, \tau_{IRS}, k} = \langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle_{0 \leq i \leq \tau_{IRS} - \varepsilon - 1},$$

with probability at least $1 - \tau_{IRS}/q$.

Proof. First notice that, since $(\Lambda \mathbf{f}, \Lambda) \in \mathcal{S}_{Y, \tau_{IRS}, k}$ then

$$\langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle_{0 \leq i \leq \tau_{IRS} - \varepsilon - 1} \subseteq \ker(M_{Y, \tau_{IRS}, k}) = \mathcal{S}_{Y, \tau_{IRS}, k}. \quad (2.24)$$

In the first part of the proof we show the existence of a draw of columns of Y corresponding to the error positions, for which we have $\mathcal{S}_{Y, \tau_{IRS}, k} \subseteq \langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle_{0 \leq i \leq \tau_{IRS} - \varepsilon - 1}$ and so by (2.24), we have the equality.

Consider a partition of E , *i.e.* $E = \cup_{i=1}^l I_i$, such that for any $1 \leq i \leq l$, $|I_i| \leq \lceil \varepsilon/l \rceil$. Note that such a partition exists since $l \lceil \varepsilon/l \rceil \geq \varepsilon = |E|$. For any $j \in E$, we denote by i_j the unique index such that $j \in I_{i_j}$. Construct a matrix V , such that

- $V_{*,j} = \mathbf{f}(\alpha_j)$, if $j \notin E$,
- otherwise if $j \in E$, $V_{*,j}$ is chosen so that

$$\mathbf{f}(\alpha_j) - V_{*,j} = \boldsymbol{\epsilon}_{i_j}, \quad (2.25)$$

where $\boldsymbol{\epsilon}_i$ is a vector of $\mathbb{F}_q^{l \times 1}$, whose i -th entry is 1 and all the others are zero.

Now, consider $(\boldsymbol{\varphi}, \lambda) \in \mathcal{S}_{V, \tau_{IRS}, k}$. By multiplying (2.25) by $\lambda(\alpha_j)$ and since $(\boldsymbol{\varphi}, \lambda)$ belongs to the solution space $\mathcal{S}_{V, \tau_{IRS}, k}$ then,

$$\lambda(\alpha_j) \mathbf{f}(\alpha_j) - \underbrace{\lambda(\alpha_j) V_{*,j}}_{\boldsymbol{\varphi}(\alpha_j)} = \lambda(\alpha_j) \boldsymbol{\epsilon}_{i_j}.$$

Fix $1 \leq i \leq l$, observe that for any $j \notin I_i$ then,

- if $j \notin E$, then $V_{*,j} = \mathbf{f}(\alpha_j)$ and so we get $\lambda(\alpha_j) f_i(\alpha_j) - \varphi_i(\alpha_j) = 0$,
- if $j \in E \setminus I_i$, then by the choice of $V_{*,j}$ we have $\lambda(\alpha_j) f_i(\alpha_j) - \varphi_i(\alpha_j) = 0$,

hence in both cases, $\lambda(\alpha_j) f_i(\alpha_j) - \varphi_i(\alpha_j) = 0$. Note that the polynomial $\lambda f_i - \varphi_i$ has $n - |I_i| \geq n - \lceil \varepsilon/l \rceil$ roots. Since $\varepsilon \leq \tau_{IRS} \leq \frac{l(n-k)}{l+1}$, then $n - \lceil \varepsilon/l \rceil \geq n - \tau_{IRS}/l \geq \tau_{IRS} + k$. On the other hand, the degree of the polynomial $\lambda f - \varphi_i$ is at most $\tau_{IRS} + k - 1$, hence it is the zero polynomial. Hence $\boldsymbol{\varphi} - \lambda \mathbf{f} = \mathbf{0}$.

Now, since for any $1 \leq j \leq n$ $\lambda(\alpha_j) \mathbf{f}(\alpha_j) = \boldsymbol{\varphi}(\alpha_j) = \lambda(\alpha_j) V_{*,j}$, we have $\lambda(\alpha_j) [\mathbf{f}(\alpha_j) - V_{*,j}] = 0$. We remark that for $j \in E$, by construction $\mathbf{f}(\alpha_j) - V_{*,j} \neq \mathbf{0}$ and so $\Lambda = \prod_{j \in E} (x - \alpha_j)$ divides λ . This implies that there exists P such that $\lambda = P\Lambda$ and $\boldsymbol{\varphi} = \lambda \mathbf{f} = P\Lambda \mathbf{f}$.

Therefore, $\mathcal{S}_{V, \tau_{IRS}, k} \subseteq \langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle_{0 \leq i \leq \tau_{IRS} - \varepsilon - 1}$ and so we have the equality.

Given Y as in the assumption of this lemma, since $\langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle_{0 \leq i \leq \tau_{IRS} - \varepsilon - 1} \subseteq \mathcal{S}_{Y, \tau_{IRS}, k}$, then $\dim(\ker(M_{Y, \tau_{IRS}, k})) \geq \tau_{IRS} - \varepsilon$. By the Rank-Nullity Theorem

$$\text{rank}(M_{Y, \tau_{IRS}, k}) = l(\tau_{IRS} + k) + \tau_{IRS} + 1 - \dim(\ker(M_Y)) \leq l(\tau_{IRS} + k) + 1 + \varepsilon =: \rho.$$

Hence $\text{rank}(M_{Y, \tau_{IRS}, k}) \leq \rho$. On the other hand, as proved above, there exists a draw $V_{*,j}$ of $Y_{*,j}$ for $j \in E$, such that $\text{rank}(M_{V, \tau_{IRS}, k}) = \rho$. This means that there exists a nonzero ρ -minor in $M_{V, \tau_{IRS}, k}$. We consider this nonzero ρ -minor as a multivariate polynomial C whose indeterminates are $(y_{i,j})_{\substack{1 \leq i \leq n \\ j \in E}}$. We can remark that we showed the existence of a draw $V_{*,j}$ of $Y_{*,j}$, for $j \in E$ such that the corresponding $C(V_{*,j})$ is non zero. Hence the polynomial C is nonzero.

For any matrix Y such that $(Y_{*,j})_{j \in E}$ are not roots of C , then the solution space is $\mathcal{S}_{Y, \tau_{IRS}, k} = \langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle_{0 \leq i \leq \tau_{IRS} - \varepsilon - 1}$. Note that the total degree of the polynomial C is at

most τ_{IRS} , since only the last τ_{IRS} columns of the corresponding matrix $M_{Y, \tau_{IRS}, k}$ contain the variables $(y_{i,j})_{\substack{1 \leq j \leq n \\ j \in E}}$ (see (2.21)).

Finally by the Schwartz-Zippel Lemma, the polynomial C cannot be zero in more than τ_{IRS}/q fractions of its domain. Therefore we can conclude that the probability that $\mathcal{S}_{Y, \tau_{IRS}, k} \neq \langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle_{0 \leq i \leq \tau_{IRS} - \varepsilon - 1}$ is at most τ_{IRS}/q . \square

We observe that the assumption on the error distribution of Lemma 2.3.1 is *somewhat consistent* with the channel model: a q^l -ary Symmetric Channel over \mathbb{F}_{q^l} .

Remark 2.3.4. Note that if Y is a received word for which the solution space is $\mathcal{S}_{Y, \tau_{IRS}, k} = \langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle_{0 \leq i \leq \tau_{IRS} - \varepsilon - 1}$, then the $\mathbb{F}_q[x]$ -module generated by solutions of (2.23) has rank 1 and the generator of the basis (φ, λ) , scaled to have λ monic is exactly $(\Lambda \mathbf{f}, \Lambda)$. Therefore, the divisibility and the degree condition of the step 5 of Algorithm 4 are satisfied and it outputs \mathbf{f} . \spadesuit

Lemma 2.3.2. Let $Y \in \mathbb{F}_q^{l \times n}$ a received matrix and let $(\varphi, \lambda) \in \mathcal{S}_{Y, \tau_{IRS}, k}$ such that $\lambda | \varphi_i$ for any $1 \leq i \leq l$ and $\deg(\mathbf{f}) \leq k - 1$, where $\mathbf{f} := (f_1, \dots, f_l)^T$ and $f_i := \varphi_i / \lambda$. Then the Hamming distance⁶ satisfies $d((\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)), Y) \leq \tau_{IRS}$.

Proof. Indeed, since $\varphi = \lambda \mathbf{f}$, the $\lambda(\alpha_j)[\mathbf{f}(\alpha_j) - Y_{*,j}] = \mathbf{0}$ for any $1 \leq j \leq n$. Therefore, $|E| = d((\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)), Y) \leq \deg(\lambda) \leq \tau_{IRS}$. \square

Lemma 2.3.3. Let $Y \in \mathbb{F}_q^{l \times n}$ a received word such that $\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^{(\tau_{IRS})}(Y) \neq \emptyset$. Let \mathbf{f} be the vector of polynomials related to the codeword in the ball $\mathcal{B}^{(\tau_{IRS})}(Y)$ and Λ the error locator polynomial.

If the rank of the $\mathbb{F}_q[x]$ -module of solutions in $\mathcal{S}_{Y, \tau_{IRS}, k}$ is 1, then its generator (φ, λ) with λ monic coincides exactly with $(\Lambda \mathbf{f}, \Lambda)$.

Proof. Since (φ, λ) is a generator of the $\mathbb{F}_q[x]$ -module of solutions $\mathcal{S}_{Y, \tau_{IRS}, k}$ and since $(\Lambda \mathbf{f}, \Lambda) \in \mathcal{S}_{Y, \tau_{IRS}, k}$ there exists $P \in \mathbb{F}_q[x]$ such that $(\Lambda \mathbf{f}, \Lambda) = (P\varphi, P\lambda)$. Therefore, $\lambda | \Lambda$ and it is of the form $\lambda = \prod_{j \in E'} (x - \alpha_j)$, where $E' \subseteq E$. Moreover, $\varphi = \lambda \mathbf{f}$. As in the proof of Lemma 2.3.2, we have that $|E| = d((\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)), Y) \leq \deg(\lambda) = |E'|$ and so $E = E'$ and $P \in \mathbb{F}_q$. \square

Remark 2.3.5.

1. Let $Y \in \mathbb{F}_q^{l \times n}$ a received word such that $\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^{(\tau_{IRS})}(Y) \neq \emptyset$. We denote by \mathbf{f} the vector of polynomials corresponding to a codeword in $\mathcal{B}^{(\tau_{IRS})}(Y)$ and Λ the error locator polynomial. Lemma 2.3.3 basically tells us that if the rank of the $\mathbb{F}_q[x]$ -module of solutions in $\mathcal{S}_{Y, \tau_{IRS}, k}$ is 1 then $\mathcal{S}_{Y, \tau_{IRS}, k}$ is spanned by $\langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle$. This is a sort of “*vice versa*” of Remark 2.3.4. Informally speaking, this allows us to conclude that the

6. Recall that in Section 2.1 we defined the Hamming distance in the vector space \mathbb{F}_q^n , for a general finite field \mathbb{F}_q . By the extension field representation, we can consider codewords of $\mathcal{C}_{IRS}(n, k)$, i.e. $(\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)) \in \mathbb{F}_q^{l \times n}$, as elements in $\mathbb{F}_{q^l}^n$. Hence the Hamming distance in this case is defined in the vector space $\mathbb{F}_{q^l}^n$.

fact that the rank of the $\mathbb{F}_q[x]$ -module of solutions is 1 is equivalent to have a solution space of the form $\langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle$.

2. Note that by Lemma 2.3.3 we cannot have more than one codewords in the ball $\mathcal{B}^{(\tau_{IRS})}(Y)$ and rank of the $\mathbb{F}_q[x]$ -module equal to 1. This means that if the rank is 1, then $|\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^{(\tau_{IRS})}(Y)| = 1$.

💡

We now analyze the behavior of the decoder of Algorithm 4, in order to verify its correctness.

Remark 2.3.6 (Correctness of the partial BD decoder (Algorithm 4)). Let Y be a received word, \mathcal{M} the $\mathbb{F}_q[x]$ -module of solutions in $S_{Y, \tau_{IRS}, k}$ (as in step 1 of the algorithm) and $r = \text{rank}(\mathcal{M})$. We have the following possibilities,

- if $\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^{(\tau_{IRS})}(Y) = \emptyset$, the decoder outputs a decoding failure. Indeed, by Lemma 2.3.2, it cannot return a vector of polynomial $\mathbf{f} \in \mathbb{F}_q[x]^{l \times 1}$ of $\deg(\mathbf{f}) \leq k - 1$,
- if $|\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^{(\tau_{IRS})}(Y)| = 1$, then by Lemma 2.3.1, with probability at least $1 - \tau_{IRS}/q$ then $\mathcal{S}_{Y, \tau_{IRS}, k} = \langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle$ and so $\text{rank}(\mathcal{M}) = 1$. Note that in the equality case, the element of the basis computed at step 3 of Algorithm 4 is exactly $(\Lambda \mathbf{f}, \Lambda)$ and so by performing the division we get \mathbf{f} , which corresponds to the sent codeword.
- if $|\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^{(\tau_{IRS})}(Y)| > 1$, then by Remark 2.3.5, $\text{rank}(\mathcal{M}) > 1$ and so the decoder outputs a decoding failure.

💡

As previously remarked, this decoder is a *partial* BD decoder with decoding radius τ_{IRS} , indeed, if Y is such that $|\mathcal{C}_{IRS}(n, k) \cap \mathcal{B}^{(\tau_{IRS})}(Y)| = 1$ it could eventually fail, even if there is only one codeword in the ball.

Failure probability. We introduced a decoder which, given a received vector $Y = (\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_n)) + \Xi$, where $\varepsilon \leq \tau_{IRS}$, can recover the sent codeword \mathbf{f} with a certain probability. Precisely, by Lemma 2.3.1, the failure probability, that is the probability that the decoder does not recover this sent codeword, leading to a decoding failure message is at most τ_{IRS}/q . As previously observed, this result is just an adaptation of Theorem 1 of [BKY03], for a slightly different decoder.

In [BMS04] A. Brown *et al.* proved that the failure probability does not depend on the number of errors and it is $\mathcal{O}(1/q)$. More specifically they proved that this probability is at most

$$\frac{\exp(1/(q^{l-2}))}{q-1}.$$

On the other hand, in [SSB07, SSB09, SSB10] it was introduced a partial decoder based on the syndrome-based approach with decoding radius τ_{IRS} . They proved that the failure probability of this decoder is at most

$$\left(\frac{q^l - \frac{1}{q}}{q^l - 1}\right)^\varepsilon \frac{q^{-(l+1)(\tau_{IRS}-\varepsilon)}}{q-1} \quad (2.26)$$

In the same papers, the authors show that this failure probability is tight in practice.

2.4 A short summary of the chapter

Here we present a short summary of the main results and notions of the chapter.

Reed-Solomon codes. The Reed-Solomon (RS) code is a *linear* code which is determined by the evaluations of polynomials with bounded degrees. Decoding a Reed-Solomon codeword is then the problem of reconstructing a polynomial of bounded degree by its evaluations, where some are erroneous. We called this problem *interpolation with errors*.

We saw that there are basically two approaches to decode RS codes: an *interpolation-based* and a *syndrome-based*. Moreover, for both of them the decoding problem can be reduced to a *rational function reconstruction* (RFR) and so, all the existing RS decoders basically differ in the algorithm chosen to solve RFR.

In this coding theory scenario, the number of evaluation points which guarantees the uniqueness of RFR is related to the maximum number of errors that we can *uniquely* correct, *i.e.* the error correction capability of the RS code.

Interleaved Reed-Solomon codes. Interleaving is an encoding technique used in settings where errors are extended to consecutive symbols of the received word. This technique applied to RS codes allows the construction of *partial* decoders, which can correct almost all error patterns beyond the error correction capability of the code.

In detail, the interleaved RS (IRS) code is determined by evaluations of *vectors* of polynomials with bounded degrees.

Decoding an IRS codeword consists in reconstructing a vector of polynomials of bounded degrees given its evaluations, where some are erroneous (the *simultaneous interpolation with errors* problem). This is the vector extension of the interpolation with errors problem.

As for classic RS codes, there are two techniques for decoding IRS codes: an *interpolation-based* and a *syndrome-based* one. In this chapter we focus on the first one, since we will generalize this technique later, for the *simultaneous polynomial reconstruction with errors problem* (see Subsection 4.1.1).

The interpolation-based approach basically reduces the decoding problem to the *simultaneous rational function reconstruction* (SRFR) problem. In this case the common denominator of the vector of rational functions that we want to recover is the error locator polynomial.

In this coding theory scenario, we pointed out that *reducing* the number of evaluation points which guarantees the uniqueness of SRFR is equivalent to *increasing* the number of errors that we can *uniquely* correct. So, this reduction allows one to construct decoders which correct more errors.

As seen in the previous chapter, if we consider the number of evaluation points which guarantees the uniqueness of the classic RFR, we have also the uniqueness of SRFR. From the coding theory perspective, this means that if the number of errors is smaller than the error correction capability of the code, *i.e.* half of the minimum distance, we can uniquely decode any IRS codeword. Indeed, it suffices to separately decode any component of the IRS codeword, which is an RS codeword.

Besides, the interleaving construction allows to do better: if the number of errors is smaller than the number derived from the common denominator feature of SRFR, we have the uniqueness of SRFR for the corresponding instance $\mathbf{y} = \mathbf{f} + \mathbf{e}$, for all \mathbf{f} and for *almost all* (see Definition 3.1.1) errors \mathbf{e} . This means that we can construct decoders which can uniquely correct almost all error patterns beyond the error correction capability of the code.

This motivates our study about uniqueness of the general SRFR problem with this reduced number of evaluation points, as we will see in the following chapter.

CHAPTER 3

Generic Uniqueness of SRFR

Contents

3.1	Generic Row Degrees of the Relation Module	82
3.1.1	Monomial orders on modules	83
3.1.2	Row degrees of a relation module as row rank profile	86
3.1.3	Constraints on linearly independent monomial families	88
3.1.4	Generic row degrees	93
3.2	Generic row degrees of the SRFR Relation Module	95
3.3	Conclusions and open problems	98

The simultaneous rational function reconstruction (Problem 2), is the problem of reconstructing a vector of rational functions with the same denominator given their remainders modulo some polynomials.

We saw how the common denominator constraint affects the condition on the parameters of the problem which guarantees the existence of a nontrivial solution (see equation 1.7). For instance, in the interpolation case, in which we want to reconstruct a vector of polynomials given its evaluation, the common denominator constraint reduces the number of evaluation points needed to guarantee the *existence* of a nontrivial solution, possibly losing its uniqueness. We also saw that this reduction is significant for the applications of this problem:

- for the polynomial linear system solving (Section 1.4) by evaluation-interpolation, this means a reduction on the number of points needed to reconstruct the solution. This reduction has an impact on the complexity of the resolution algorithms which depend on this number of points;
- from a coding theory perspective, this reduction allows the construction of decoders for IRS (Section 2.3) which correct beyond the error correction capability of the code.

Moved by these considerations, in [GLZ20b] we proved that with this smaller number of evaluation points (or if equation (1.7) holds, in the general case) SRFR admits a unique solution for almost all instances of the problem (Theorem 3.2.1).

This chapter is devoted to the presentation of this result.

Recall that in Section 1.2 we saw that solutions of SRFR are elements of a specific relation module with negative shifted row degrees (see Lemma 1.3.8). The shift integrates the degree constraints. Therefore, in the uniqueness case there is only one element of an ordered weak Popov basis of this relation module with negative shifted row degrees. Hence the goal of this chapter is to prove that for almost all instances \mathbf{u} of SRFR the row degrees of the relation module are of the form $\boldsymbol{\rho} = (0, \dots, 0, -1)$ (Theorem 3.2.1).

In order to achieve it, in Section 3.1 we first prove some results about the row degrees of general relation modules, *i.e.* relation modules related to general matrices $M \in \mathbb{K}[x]^{m \times n}$. Then in Section 3.2 we transpose all these results to the relation module $\mathcal{A}_{\mathcal{M}, R_{\mathbf{u}}}$ related to solutions of SRFR in order to finally prove our uniqueness result (Theorem 3.2.1).

3.1 Generic Row Degrees of the Relation Module

In this section we derive the *generic* form of the row degrees of relation modules. For the sake of clarity, we start this section with a short summary of some notions and notations introduced in Section 1.3.

Short summary of previous notions. Let $M \in \mathbb{K}[x]^{m \times n}$, where $m \geq n$ and \mathcal{M} a $\mathbb{K}[x]$ -submodule of $\mathbb{K}[x]^n = \mathbb{K}[x]^{1 \times n}$ of rank n . The relation module $\mathcal{A}_{\mathcal{M}, M}$ is the kernel of the $\mathbb{K}[x]$ -module homomorphism $\hat{\varphi}_M : \mathbb{K}[x]^m \rightarrow \mathbb{K}[x]^n / \mathcal{M}$ such that $\hat{\varphi}(\mathbf{p}) = \mathbf{p}M$ for any $\mathbf{p} \in \mathbb{K}[x]^m$. We denote by $\varphi_M : \mathbb{K}[x]^m / \mathcal{A}_{\mathcal{M}, M} \hookrightarrow \mathbb{K}[x]^n / \mathcal{M}$ the corresponding injection.

To lighten the notations we also denote by $\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_m$ and $\boldsymbol{\varepsilon}'_1, \dots, \boldsymbol{\varepsilon}'_n$ the canonical bases of $\mathbb{K}[x]^m$ and $\mathbb{K}[x]^n$ respectively. Moreover, let $\mathcal{K} = \mathbb{K}[x]^n / \mathcal{M} \simeq \mathbb{K}[x]^n / \langle a_i(x)\boldsymbol{\varepsilon}'_i \rangle$ and $L_i = \deg(a_i)$, where $L_1 \geq L_2 \geq \dots \geq L_n$ (see Remark 1.3.4). We also consider $\mathbf{e}_i = \boldsymbol{\varepsilon}_i \bmod \mathcal{A}_{\mathcal{M}, M}$.

Given a shift $\mathbf{s} \in \mathbb{Z}^m$, we let $\boldsymbol{\rho}$ and $\boldsymbol{\delta}$ be the \mathbf{s} -row degrees and \mathbf{s} -pivot degrees of $\mathcal{A}_{\mathcal{M}, M}$ (see Definition 1.3.8). In particular, $\boldsymbol{\delta} = \boldsymbol{\rho} - \mathbf{s}$.

Sometimes, we write $\boldsymbol{\rho}_M$ and $\boldsymbol{\delta}_M$ to stress the matrix dependency.

At this point the reader may ask what we mean by the terminology “generic” which we frequently use in this work. We now give a formal definition.

Definition 3.1.1 (Generic property). A property \mathcal{P} is said to hold *generically*, or for *generic instances* or even for *almost all instances*, if there exists a nonzero polynomial such that the property holds for all instances for which the polynomial is not vanishing.

Remark 3.1.1. Lemma 2.3.1 can be reformulated by saying that for a fixed error support $E \subseteq \{1, \dots, n\}$, with $|E| \leq \tau_{IRS}$, for almost all error patterns in \mathbb{F}_q^l then $\mathcal{S}_{Y, \tau_{IRS}, k} = \langle x^i \Lambda f, x^i \Lambda \rangle_{0 \leq i \leq \tau_{IRS} - \varepsilon - 1}$. Indeed, in this case the nonzero polynomial is the ρ -minor of the matrix $M_{Y, \tau_{IRS}, k}$ (see the proof of Lemma 2.3.1 for more details). \spadesuit

3.1.1 Monomial orders on modules

Let \mathbb{K} be a field and $\mathbb{K}[\mathbf{x}] = \mathbb{K}[x_1, \dots, x_t]$ be a multivariate polynomial ring. In this subsection we introduce the notion of *monomial orders* on $\mathbb{K}[\mathbf{x}]$ -modules. We see a particular monomial order, the *term over position* (shortly *TOP*) in its general and *shifted* version (*s-TOP*). Finally, by transposing all these results to the univariate case, we underline the link between *s-pivots* (Definition 1.3.6) and leading terms *w.r.t s-TOP* order.

Monomial orders on $\mathbb{K}[\mathbf{x}]$. We now recall basic notions about monomial orders in $\mathbb{K}[\mathbf{x}]$. For more details, we refer the reader to [CLO07, Section 2.2].

Recall that a monomial in $\mathbb{K}[\mathbf{x}]$ is a product of the form $\mathbf{x}^\alpha := x_1^{\alpha_1} \cdots x_t^{\alpha_t}$, where $\alpha = (\alpha_1, \dots, \alpha_t) \in \mathbb{N}^t$.

Definition 3.1.2 (Monomial order). A monomial ordering \prec is any ordering relation on monomials in $\mathbb{K}[\mathbf{x}]$, where $\alpha = (\alpha_1, \dots, \alpha_t) \in \mathbb{N}^t$, such that,

- \prec is a *total ordering*, i.e. for any pair of monomials then we have only one of the following: $\mathbf{x}^\alpha \prec \mathbf{x}^\beta$, $\mathbf{x}^\alpha = \mathbf{x}^\beta$, $\mathbf{x}^\beta \prec \mathbf{x}^\alpha$.
- if $\mathbf{x}^\alpha \prec \mathbf{x}^\beta$, then $\mathbf{x}^\alpha \mathbf{x}^\gamma \prec \mathbf{x}^\beta \mathbf{x}^\gamma$, for any monomial \mathbf{x}^γ ,
- \prec is a *well ordering*, i.e. there are no infinite descending chains of monomials.

Among all the monomial orders we especially focus on the *lexicographic* order.

Example 3.1.1 (Lexicographic order). Let \mathbf{x}^α and \mathbf{x}^β be monomials in $\mathbb{K}[\mathbf{x}]$. Then $\mathbf{x}^\alpha \prec_{LEX} \mathbf{x}^\beta$, if in the difference $\alpha - \beta \in \mathbb{Z}^t$, the left-most nonzero entry is negative.

Here are some examples, for $t = 3$

1. $x_2^3 x_3^4 \prec_{LEX} x_1 x_2$
2. $x_1 x_2 \prec_{LEX} x_1 x_2^2 x_3^3$.



Remark 3.1.2. We observe that in the univariate case, the only monomial order is the natural degree order, $x^\alpha < x^\beta$ if $\alpha < \beta$.

Let $f = \sum_{\alpha} c_{\alpha} \mathbf{x}^{\alpha}$, be a polynomial in $\mathbb{K}[\mathbf{x}]$ and \prec be a monomial order. Let \mathbf{x}^{α} be the maximal monomial *w.r.t* \prec such that $c_{\alpha} \neq 0$. Then \mathbf{x}^{α} is called *leading monomial*, c_{α} is the *leading coefficient* and $c_{\alpha} \mathbf{x}^{\alpha}$ is the *leading term*. Formally, we denote them respectively $LM_{\prec}(f)$, $LC_{\prec}(f)$ and $LT_{\prec}(f)$.

Notice that, we can identify monomials \mathbf{x}^{α} in $\mathbb{K}[\mathbf{x}]$ with n -tuples $\alpha = (\alpha_1, \dots, \alpha_t)$ in \mathbb{N}^t . For this reason we can consider monomial orders on $\mathbb{K}[\mathbf{x}]$ as orders on \mathbb{N}^t .

Monomial orders on $\mathbb{K}[\mathbf{x}]$ -modules. We can naturally extend Definition 3.1.2 to the $\mathbb{K}[\mathbf{x}]$ -module $\mathbb{K}[\mathbf{x}]^t$ [CLO05, Chapter 2]. A monomial in $\mathbb{K}[\mathbf{x}]^t$ is an element \mathbf{m} of the form $\mathbf{x}^\alpha \boldsymbol{\varepsilon}_i$ for some i , $1 \leq i \leq t$. We denote by $\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_t$ the canonical basis of $\mathbb{K}[\mathbf{x}]^t$.

Definition 3.1.3 (Monomial order on a module). An ordering relation \prec on the monomials in $\mathbb{K}[\mathbf{x}]^t$ is a monomial ordering if,

- \prec is a total ordering,
- for any \mathbf{m}, \mathbf{n} monomials in $\mathbb{K}[\mathbf{x}]^t$, such that $\mathbf{m} \prec \mathbf{n}$ then $\mathbf{x}^\alpha \mathbf{m} \prec \mathbf{x}^\alpha \mathbf{n}$ for any \mathbf{x}^α monomial in $\mathbb{K}[\mathbf{x}]$,
- \prec is a well ordering.

Some of the common monomial orders on $\mathbb{K}[\mathbf{x}]^t$ derive from the extension of orders on $\mathbb{K}[\mathbf{x}]$. Basically, according to the chosen order on the canonical basis vectors, we can distinguish two main approaches to derive the monomial orders, which are described in the following definition. Note that here we suppose that $\boldsymbol{\varepsilon}_1 \prec \dots \prec \boldsymbol{\varepsilon}_t$.

Definition 3.1.4 (TOP and POT monomial orderings). Let \prec be a monomial order on $\mathbb{K}[\mathbf{x}]$. Then,

- $\mathbf{x}^\alpha \boldsymbol{\varepsilon}_i \prec_{TOP} \mathbf{x}^\beta \boldsymbol{\varepsilon}_j$ if $\mathbf{x}^\alpha \prec \mathbf{x}^\beta$, or if $\mathbf{x}^\alpha = \mathbf{x}^\beta$ and $i < j$;
- $\mathbf{x}^\alpha \boldsymbol{\varepsilon}_i \prec_{POT} \mathbf{x}^\beta \boldsymbol{\varepsilon}_j$ if $i < j$, or if $i = j$ and $\mathbf{x}^\alpha \prec \mathbf{x}^\beta$;

The terminologies TOP and POT derive from [AL94]. In detail TOP stands for “terms-over-position” and POT for “position-over-terms”. Indeed, the TOP order sorts monomial first by the monomial order on $\mathbb{K}[\mathbf{x}]$ and then by the position of the canonical basis elements. The POT sorts exactly in the opposite way.

We now introduce the *shifted* TOP order. In the literature shifted orders are also called *weighted* orders ([FF92, OF07]). Here the former terminology is preferred to the latter since it underlines the link with shifted row degrees (Definition 1.3.3).

Definition 3.1.5 (shifted TOP monomial orderings). Let \prec be a monomial order on $\mathbb{K}[\mathbf{x}]$ and $\mathbf{x}^{s_1}, \dots, \mathbf{x}^{s_t}$ be the *shifting* monomials in $\mathbb{K}[\mathbf{x}]$. Then $\mathbf{x}^\alpha \boldsymbol{\varepsilon}_i \prec_{s-TOP} \mathbf{x}^\beta \boldsymbol{\varepsilon}_j$ if $\mathbf{x}^\alpha \mathbf{x}^{s_i} \prec \mathbf{x}^\beta \mathbf{x}^{s_j}$, or if $\mathbf{x}^\alpha \mathbf{x}^{s_i} = \mathbf{x}^\beta \mathbf{x}^{s_j}$ and $i < j$.

Note that each shifting monomial is related to a shift $\mathbf{s} = (s_1, \dots, s_t) \in \mathbb{Z}^t$.

Example 3.1.2. Consider $t = 2$, the \prec_{LEX} -order on $\mathbb{K}[x_1, x_2]$ and shift $\mathbf{s} = (1, 0)$. We have,

1. $x_1 \boldsymbol{\varepsilon}_2 \prec_{TOP} x_1^2 \boldsymbol{\varepsilon}_1 \prec_{TOP} x_1^2 \boldsymbol{\varepsilon}_2 \prec_{TOP} x_1^2 x_2^3 \boldsymbol{\varepsilon}_2$,
2. $x_1 \boldsymbol{\varepsilon}_1 \prec_{POT} x_1^2 x_2^2 \boldsymbol{\varepsilon}_1 \prec_{POT} x_1 \boldsymbol{\varepsilon}_2 \prec_{POT} x_1 x_1^3 \boldsymbol{\varepsilon}_2$,
3. $x_1 \boldsymbol{\varepsilon}_1 \prec_{s-TOP} x_1^2 \boldsymbol{\varepsilon}_2 \prec_{s-TOP} x_1^2 x_2 \boldsymbol{\varepsilon}_1 \prec_{s-TOP} x_1^2 x_2 \boldsymbol{\varepsilon}_2$.



As for polynomials in $\mathbb{K}[\mathbf{x}]$, we can introduce the notions of leading coefficient, monomial and term. Let $\mathbf{f} \in \mathbb{K}[\mathbf{x}]^t$ and \prec a monomial order in $\mathbb{K}[\mathbf{x}]^t$. We can write $\mathbf{f} = \sum_{i=1}^t c_i \mathbf{m}_i$, where \mathbf{m}_i are monomials in $\mathbb{K}[\mathbf{x}]^t$. Let \mathbf{m}_i the maximal monomial *w.r.t* \prec such that $c_i \neq 0$. Then \mathbf{m}_i is the leading monomial, c_i is the leading coefficient and $c_i \mathbf{m}_i$ is the leading term, *i.e.* $LM_{\prec}(\mathbf{f}) = \mathbf{m}_i$, $LC_{\prec}(\mathbf{f}) = c_i$ and $LT_{\prec}(\mathbf{f}) = c_i \mathbf{m}_i$.

Given \mathcal{N} a submodule of $\mathbb{K}[\mathbf{x}]^t$ we can also define $LT(\mathcal{N})$, the monomial submodule $\{LT(\mathbf{f}) \mid \mathbf{f} \in \mathcal{N}\}$ of $\mathbb{K}[\mathbf{x}]^t$.

The univariate case. We now consider the $\mathbb{K}[x]$ -module $\mathbb{K}[x]^t$, then we have

- $x^\alpha \boldsymbol{\varepsilon}_i \prec_{TOP} x^\beta \boldsymbol{\varepsilon}_j$ if $(\alpha, i) \prec_{LEX} (\beta, j)$,
- $x^\alpha \boldsymbol{\varepsilon}_i \prec_{POT} x^\beta \boldsymbol{\varepsilon}_j$ if $(i, \alpha) \prec_{LEX} (j, \beta)$,
- given a shift $\mathbf{s} = (s_1, \dots, s_t) \in \mathbb{Z}^t$ then $x^\alpha \boldsymbol{\varepsilon}_i \prec_{\mathbf{s}-TOP} x^\beta \boldsymbol{\varepsilon}_j$ if $(\alpha + s_i, i) \prec_{LEX} (\beta + s_j, j)$.

We now focus on \prec_{TOP} and $\prec_{\mathbf{s}-TOP}$ orders. Notice that these orders sort monomials *w.r.t* to their row degree and shifted row degree respectively. Indeed, we can observe that given a monomial $x^\alpha \boldsymbol{\varepsilon}_i$ in the $\mathbb{K}[x]$ -module $\mathbb{K}[x]^t$ and a shift $\mathbf{s} \in \mathbb{Z}^n$, then $\text{rdeg}(x^\alpha \boldsymbol{\varepsilon}_i) = \alpha$ and $\text{rdeg}_{\mathbf{s}}(x^\alpha \boldsymbol{\varepsilon}_i) = \alpha + s_i$.

So, we can conclude that

- $x^\alpha \boldsymbol{\varepsilon}_i \prec_{TOP} x^\beta \boldsymbol{\varepsilon}_j$ if $\text{rdeg}(x^\alpha \boldsymbol{\varepsilon}_i) < \text{rdeg}(x^\beta \boldsymbol{\varepsilon}_j)$ or if $\text{rdeg}(x^\alpha \boldsymbol{\varepsilon}_i) = \text{rdeg}(x^\beta \boldsymbol{\varepsilon}_j)$ or $i < j$,
- $x^\alpha \boldsymbol{\varepsilon}_i \prec_{\mathbf{s}-TOP} x^\beta \boldsymbol{\varepsilon}_j$ if $\text{rdeg}_{\mathbf{s}}(x^\alpha \boldsymbol{\varepsilon}_i) < \text{rdeg}_{\mathbf{s}}(x^\beta \boldsymbol{\varepsilon}_j)$ and if $\text{rdeg}_{\mathbf{s}}(x^\alpha \boldsymbol{\varepsilon}_i) = \text{rdeg}_{\mathbf{s}}(x^\beta \boldsymbol{\varepsilon}_j)$ and $i < j$.

Example 3.1.3. Let $\mathbb{K} = \mathbb{F}_7$, $t = 3$ and $\mathbf{s} = (0, 1, 2)$. Then

Mon	$(1, 0, 0) \prec_{\mathbf{s}-TOP}$	$(x, 0, 0) \prec_{\mathbf{s}-TOP}$	$(0, 1, 0) \prec_{\mathbf{s}-TOP}$	$(x^2, 0, 0) \prec_{\mathbf{s}-TOP}$	$(0, x, 0)$
$\text{rdeg}_{\mathbf{s}}$	0	1		2	



We can now state the link between the $\prec_{\mathbf{s}-TOP}$ monomial order on $\mathbb{K}[x]^t$ and pivots (Definition 1.3.6).

Remark 3.1.3. Let $\mathbf{p} = (p_1, \dots, p_n) \in \mathbb{K}[x]^t$, $\mathbf{s} \in \mathbb{Z}^t$ be a shift and $LT(\mathbf{p}) = c_i x^\alpha \boldsymbol{\varepsilon}_i$. Then the \mathbf{s} -pivot index, entry and degree are respectively i , p_i and α .

Example 3.1.4. Let $\mathbb{K} = \mathbb{F}_7$, $t = 3$ and $\mathbf{s} = (0, 1, 2)$. Consider $\mathbf{p} = (x^2 + 3, 3x^3 + 5x + 2, 2x^2 + 1)$.

Mon	$(x^2, 0, 0) \prec_{\mathbf{s}-TOP}$	$(0, 3x^3, 0) \prec_{\mathbf{s}-TOP}$	$(0, 0, 2x^2)$
$\text{rdeg}_{\mathbf{s}}$	2	4	

Therefore $LT_{\prec_{\mathbf{s}-TOP}}(\mathbf{p}) = 2x^2 \boldsymbol{\varepsilon}_3$. Indeed, note that the \mathbf{s} -pivot index, entry and degree are respectively 3, $2x^2 + 1$ and 2.

The link between \mathbf{s} -pivots and the leading term *w.r.t* $\prec_{\mathbf{s}-TOP}$ is useful to prove the following result, which basically relates the existence of $\mathbf{p} \in \mathcal{A}_{\mathcal{M},M}$ with a certain leading term $LT(\mathbf{p}) = x^d \boldsymbol{\varepsilon}_i$ to a linearly dependency relation of monomials in the $\mathbb{K}[x]$ -module $\mathbb{K}[x]^m / \mathcal{A}_{\mathcal{M},M}$.

Proposition 3.1.1. *There exists $\mathbf{p} \in \mathcal{A}_{\mathcal{M},M}$ with \mathbf{s} -pivot index i and \mathbf{s} -pivot degree d if and only if $x^d \boldsymbol{\varepsilon}_i \in B_M^{\prec x^d \boldsymbol{\varepsilon}_i}$ where $B_M^{\prec x^d \boldsymbol{\varepsilon}_i} := \langle x^n \boldsymbol{\varepsilon}_j \mid x^n \boldsymbol{\varepsilon}_j \prec_{\mathbf{s}-TOP} x^d \boldsymbol{\varepsilon}_i \rangle$.*

Proof. Fix $i, d \in \mathbb{N}$ and let $\mathbf{p} \in \mathbb{K}[x]^m$ with \mathbf{s} -pivot index i and \mathbf{s} -pivot degree d . Then, $r := \text{rdeg}_{\mathbf{s}}(\mathbf{p}) = d + s_i$ and $\mathbf{p} = ([\leq r]_{s_1}, \dots, [\leq r]_{s_{i-1}}, [r]_{s_i}, [< r]_{s_{i+1}}, \dots, [< r]_{s_m})$. Therefore we can write $\mathbf{p} = cx^d \boldsymbol{\varepsilon}_i + \mathbf{p}'$ where $c \in \mathbb{K}^*$ and $\mathbf{p}' = ([\leq r]_{s_1}, \dots, [\leq r]_{s_{i-1}}, [< r]_{s_i}, [< r]_{s_{i+1}}, \dots, [< r]_{s_m})$. So,

$$\begin{aligned} \mathbf{p} \in \mathcal{A}_{\mathcal{M},M} \text{ has } \mathbf{s}\text{-pivot index } i \text{ and degree } d & \iff \\ x^d \boldsymbol{\varepsilon}_i = (-1/c) \mathbf{p}' \bmod \mathcal{A}_{\mathcal{M},M} & \iff \\ x^d \boldsymbol{\varepsilon}_i \in \left\langle x^n \boldsymbol{\varepsilon}_j \left| \begin{array}{l} n + s_j \leq d + s_i, \text{ for } 1 \leq j \leq i - 1 \\ n + s_j < d + s_i, \text{ for } i \leq j \leq m \end{array} \right. \right\rangle = B_M^{\prec x^d \boldsymbol{\varepsilon}_i}. & \end{aligned}$$

□

In other terms, this proposition tells us that the $\mathbb{K}[x]$ -module $\mathbb{K}[x]^m / \mathcal{A}_{\mathcal{M},M}$ is generated by monomials which do not belong to $LT(\mathcal{A}_{\mathcal{M},M})$. Proposition 3.1.1 is a specific case, adapted to the modules on which we are focusing $\mathcal{A}_{\mathcal{M},M}$ and $\mathbb{K}[x]^m$, of a more general result (see [Eis95, Theorem 15.3]).

In conclusion, the next result exploits the relation between the \mathbf{s} -pivot degrees of $\mathcal{A}_{\mathcal{M},M}$ and a set of generators of $\mathbb{K}[x]^m / \mathcal{A}_{\mathcal{M},M}$.

Theorem 3.1.2. *Let $\boldsymbol{\delta}$ be the \mathbf{s} -pivot degrees of the relation module $\mathcal{A}_{\mathcal{M},M}$. Then, for any $1 \leq j \leq m$,*

$$\delta_j = \min\{d \mid x^d \boldsymbol{\varepsilon}_j \in B_M^{\prec x^d \boldsymbol{\varepsilon}_j}\}.$$

Proof. Fix $1 \leq j \leq m$. During this proof we denote $\bar{\delta}_j := \min\{d \mid x^d \boldsymbol{\varepsilon}_j \in B_M^{\prec x^d \boldsymbol{\varepsilon}_j}\}$. We want to prove that $\delta_j = \bar{\delta}_j$. Recall that by Proposition 3.1.1, $x^{\delta_j} \boldsymbol{\varepsilon}_j \in B_M^{\prec x^{\delta_j} \boldsymbol{\varepsilon}_j}$. Hence, by the minimality of $\bar{\delta}_j$, $\delta_j \geq \bar{\delta}_j$. On the other hand, $x^{\bar{\delta}_j} \boldsymbol{\varepsilon}_j \in B_M^{\prec x^{\bar{\delta}_j} \boldsymbol{\varepsilon}_j}$ so by Proposition 3.1.1 there exists $\mathbf{p} \in \mathcal{A}_{\mathcal{M},M}$ of \mathbf{s} -pivot index j and degree $\bar{\delta}_j$. Finally, by Lemma 1.3.6 we can conclude that $\bar{\delta}_j \geq \delta_j$. □

3.1.2 Row degrees of a relation module as row rank profile

We now define the matrix O_M as the *ordered matrix w.r.t* $\prec_{\mathbf{s}-TOP}$ related to the $\mathbb{K}[x]$ -module homomorphism $\hat{\varphi}_M$. More specifically, we suppose that

- the rows of the matrix from top to bottom are indexed by the monomials of $\mathbb{K}[x]^m$ sorted increasingly *w.r.t* the $\prec_{\mathbf{s}-TOP}$,
- the rows are written *w.r.t* the basis $\{x^i \varepsilon'_j\}_{0 \leq i < L_j}$ of $\mathcal{K} = \mathbb{K}[x]^n / \mathcal{M}$ as a \mathbb{K} -vector space. Recall that the polynomials a_1, \dots, a_m are the invariants of the module \mathcal{M} and $L_i = \deg(a_i)$.

Therefore, this matrix has infinite number of rows and $\sum_{j=1}^n L_j$ columns. So, it has finite rank r , where $0 \leq r \leq \sum_{j=1}^n L_j$.

Example 3.1.5. Let $\mathbb{K} = \mathbb{F}_7$ and

$$M = \begin{pmatrix} x^2 + 1 & 0 \\ 0 & 2x \\ 0 & x + 4 \end{pmatrix}$$

so, $m = 3$ and $n = 2$. Let $a_1 = a_2 = x^3$ and $\mathbf{s} = (0, 1, 2)$.

$$O_M = \begin{pmatrix} \hat{\varphi}_M(\varepsilon_1) \\ \hat{\varphi}_M(x\varepsilon_1) \\ \hat{\varphi}_M(\varepsilon_2) \\ \hat{\varphi}_M(x^2\varepsilon_1) \\ \hat{\varphi}_M(x\varepsilon_2) \\ \hat{\varphi}_M(\varepsilon_3) \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 4 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Notice that any row of the matrix is written *w.r.t* the basis $\{\varepsilon'_1, x\varepsilon'_1, x^2\varepsilon'_1, \varepsilon'_2, x\varepsilon'_2, x^2\varepsilon'_2\}$ of the \mathbb{K} -vector space $\mathbb{K}[x]^2 / \langle a_1\varepsilon'_1, a_2\varepsilon'_2 \rangle$. This matrix has infinite rows and $L_1 + L_2 = \deg(a_1) + \deg(a_2) = 6$ columns.



In this subsection our goal is to relate the *row rank profile* ([Nei16, DPS15]) of O_M to the row degrees of the relation module.

First we give the following,

Definition 3.1.6 (Row rank profile). The row rank profile of a matrix $A \in \mathbb{K}^{\mu \times \nu}$ is the lexicographically smallest sub-sequence (i_1, \dots, i_s) of $(1, \dots, \mu)$ such that the rank of A is s and the rows $A_{i_1,*}, \dots, A_{i_s,*}$ are linearly independent.

In other terms it is the lexicographically smallest sequence of s (equal to the rank of A) indices of linearly independent rows of the matrix.

In our case, since the rows of O_M are indexed by monomials in $\mathbb{K}[x]^m$, we transpose the previous definition to monomials instead of indices. We denote Mon_r the set of sets of r

monomials of $\mathbb{K}[x]^m$, where r is the rank of the matrix O_M . In order to do so, we first need to define an “extension” of the lexicographical ordering on families of monomials of Mon_r .

Definition 3.1.7. Let $\mathcal{F} = \{x^{\alpha_i} \varepsilon_{\beta_i}\}_{1 \leq i \leq r}$ and $\mathcal{F}' = \{x^{\gamma_i} \varepsilon_{\lambda_i}\}_{1 \leq i \leq r}$ two families of monomials in Mon_r both sorted increasingly *w.r.t* $\prec_{\mathbf{s}\text{-TOP}}$ -order. Then, $\mathcal{F} \leq_{\text{LEX}} \mathcal{F}'$ if there exists $1 \leq t \leq r$ such that $x^{\alpha_i} \varepsilon_{\beta_i} = x^{\gamma_i} \varepsilon_{\lambda_i}$ for $i < t$ and $x^{\alpha_t} \varepsilon_{\beta_t} \prec_{\mathbf{s}\text{-TOP}} x^{\gamma_t} \varepsilon_{\lambda_t}$

We use this order to define the row rank profile of O_M .

Definition 3.1.8 (Row rank profile of the ordered matrix). For any matrix $M \in \mathbb{K}[x]^{m \times n}$, we define the *row rank profile* of O_M (shortly RRP_M) as the family of monomials of $\mathbb{K}[x]^m$ defined by $RRP_M := \min_{\leq_{\text{LEX}}} \mathcal{P}_M$ where

$$\mathcal{P}_M := \{\mathcal{F} \in \text{Mon}_r \mid \{mM\}_{m \in \mathcal{F}} \text{ are linearly independent in } \mathcal{K}\}. \quad (3.1)$$

We now introduce a particular family of monomials, that we frequently use in this work. Given $\mathbf{d} = (d_1, \dots, d_n)$ we denote $\mathcal{F}_{\mathbf{d}} := \{x^i \varepsilon_j\}_{i < d_j}$

This family allows us to finally relate the row rank profile of O_M to the pivot degrees of the relation module.

Proposition 3.1.3. *The row rank profile of the ordered matrix O_M is given by the \mathbf{s} -pivot degrees δ_M of the relation module $\mathcal{A}_{\mathcal{M}, M}$, i.e. $RRP_M = \mathcal{F}_{\delta_M}$.*

Proof. To lighten the notations we omit the matrix dependency of the row rank profile. We define $\delta'_j = \min \{\delta \mid x^\delta \varepsilon_j \notin RRP\}$ and $\delta' = (\delta'_1, \dots, \delta'_m)$. By the definition of row rank profile, we have that $x^{\delta'_j} \varepsilon_j \in B^{\prec x^{\delta'_j} \varepsilon_j}$ (otherwise we could create a smaller family of linearly independent monomials). Using Theorem 3.1.2, we deduce that $\delta'_j \geq \delta_j$. Therefore $\mathcal{F}_{\delta} \subset \mathcal{F}_{\delta'} \subset RRP$. Since the families of monomials \mathcal{F}_{δ} and RRP have the same cardinality r (i.e. the rank of O_M), they are equal and so $\mathcal{F}_{\delta} = RRP$. \square

3.1.3 Constraints on linearly independent monomial families

We now characterize families of monomials in Mon_r which are linearly independent in \mathcal{K} , or in other words which belong to \mathcal{P}_M .

Note that we can extend the n -tuple $\mathbf{L} = (L_1, \dots, L_n)$ of the degrees of the invariants of the module \mathcal{M} to an m -tuple, by adding some zeros, i.e. $L_{n+1} = \dots = L_m = 0$.

Theorem 3.1.4. *Let $\mathbf{d} = (d_1, \dots, d_m) \in \mathbb{N}^m$ such that $d_1 \geq d_2 \geq \dots \geq d_m$ and $\mathbf{L} = (L_1, \dots, L_m)$. Then,*

$$\exists M \in \mathbb{K}[x]^{m \times n} \text{ such that } \mathcal{F}_{\mathbf{d}} \in \mathcal{P}_M \iff \sum_{i=1}^l d_i \leq \sum_{i=1}^l L_i \text{ for all } 1 \leq l \leq m.$$

Remark 3.1.4. The non-increasing property of \mathbf{d} can be dropped. Consider $\mathbf{d} \in \mathbb{N}^m$ such that $d_1 \geq \dots \geq d_m$ and π any permutation of $\{1, \dots, m\}$. Denote $\mathbf{d}' = (d_{\pi(1)}, \dots, d_{\pi(m)})$. Then there exists $M \in \mathbb{K}[x]^{m \times n}$ such that $\mathcal{F}_{\mathbf{d}} \in \mathcal{P}_M$ if and only if there exists $M' \in \mathbb{K}[x]^{m \times n}$ such that $\mathcal{F}_{\mathbf{d}'} \in \mathcal{P}_{M'}$. Indeed, recall that $\mathcal{F}_{\mathbf{d}} = \{x^i \boldsymbol{\varepsilon}_j\}_{i < d_j}$ and so $\mathcal{F}_{\mathbf{d}'} = \{x^i \boldsymbol{\varepsilon}_{\pi(j)}\}_{i < d_{\pi(j)}}$. Therefore the permutation on \mathbf{d} leads to a permutation of the rows of M , not affecting the existence property. \spadesuit

Theorem 3.1.4 is an adaptation of [Vil97, Proposition 6.1] and its derivation [PS07, Theorem 3]. Even if the statements of these two papers are in a different but related context, their proof can be applied almost straightforwardly. We still provide the main steps of the proof, for the sake of clarity and also because we adapt it later in the proof of Theorem 3.2.1. Note also that we complete the “if” part of the proof because it was not detailed in earlier references. For this purpose, we introduce the following lemma and the corresponding corollary.

Lemma 3.1.5. *Let \mathcal{N} be a $\mathbb{K}[x]$ -submodule of $\mathcal{K} = \mathbb{K}[x]^n / \mathcal{M}$ of rank l . Then the dimension of \mathcal{N} as \mathbb{K} -vector space is at most $L_1 + \dots + L_l$.*

Proof. First, remark that if $\mathbf{q} = (q_1, \dots, q_n) \in \mathcal{N}$ has its first nonzero element at index j then since $a_n | a_{n-1} | \dots | a_1$ we have that $a_j \mathbf{q} = \mathbf{0}$. Now, since \mathcal{N} has rank l , we can consider the matrix B whose rows are the l elements of a basis of \mathcal{N} . We perform a transformation on the rows of B to obtain the *Hermite normal form* B' of B . Recall that the so-obtained B' is an upper triangular matrix and so the rows $B'_{i,*}$ have the first nonzero elements at distinct indices k_1, \dots, k_l . Therefore $a_{k_j} B'_{*,j} = \mathbf{0}$ and $\{x^i B'_{*,j}\}_{0 \leq i < L_{k_j}}$ is a generating set of \mathcal{N} . So the dimension of \mathcal{N} as a \mathbb{K} -vector space is $\dim(\mathcal{N}) \leq \sum_{\substack{1 \leq j \leq l \\ 1 \leq i < L_{k_j}}} 1 \leq L_{k_1} + \dots + L_{k_l} \leq L_1 + \dots + L_l$ since $L_1 \geq \dots \geq L_n$ and the k_j are pairwise distinct. \square

Corollary 3.1.1. *Let $l \geq 0$, $\mathbf{d} = (d_1, \dots, d_l) \in \mathbb{N}^l$ and $\mathbf{v}_1, \dots, \mathbf{v}_l \in \mathcal{K}$ such that $\{x^j \mathbf{v}_i\}_{\substack{0 \leq j < d_i \\ 1 \leq i \leq l}}$ are linearly independent in \mathcal{K} . Then $\sum_{i=1}^l d_i \leq \sum_{i=1}^l L_i$.*

Proof. We consider \mathcal{N} the $\mathbb{K}[x]$ -module spanned by $\{\mathbf{v}_1, \dots, \mathbf{v}_l\}$, and we observe that $d_1 + \dots + d_l \leq \dim(\mathcal{N}) \leq L_1 + \dots + L_l$ by Lemma 3.1.5. \square

Proof of Theorem 3.1.4. We observe that if $m > n$, we can write $\mathcal{K} = \mathbb{K}[x]^n / \langle a_i \boldsymbol{\varepsilon}_i' \rangle_{1 \leq i \leq n} = \mathbb{K}[x]^m / \langle a_i \boldsymbol{\varepsilon}_i \rangle_{1 \leq i \leq m}$ where $a_i = 1$ for $n+1 \leq i \leq m$. Hence, we can suppose *w.l.o.g.* that $m = n$.

\Rightarrow) By the hypothesis, there exists a matrix $M \in \mathbb{K}[x]^{m \times n}$ such that $\{x^i \boldsymbol{\varepsilon}_j M\}_{x^i \boldsymbol{\varepsilon}_j \in \mathcal{F}_{\mathbf{d}}} = \{x^i \mathbf{v}_j\}_{0 < i < d_j}$ are linearly independent in \mathcal{K} where $\mathbf{v}_j := \boldsymbol{\varepsilon}_j M$. Hence, for all $1 \leq l \leq m$, $\mathbf{v}_1, \dots, \mathbf{v}_l$ satisfy the conditions of the Corollary 3.1.1 and so $\sum_{i=1}^l d_i \leq \sum_{i=1}^l L_i$.

\Leftarrow) Note that for $1 \leq i \leq m$, $\{x^i \mathbf{u}_j\}_{i < L_j}$ are linearly independent in \mathcal{M} . Set $\mathbf{u}_i = \boldsymbol{\varepsilon}_i^T$. We now consider the matrix $K := [K_1 | \dots | K_m]$ where $K_j \in \mathbb{K}[x]^{m \times L_j}$ is in *Krylov* form, *i.e.*

$K_j = K(\mathbf{u}_j, L_j) := [\mathbf{u}_j | x\mathbf{u}_j | \dots | x^{L_j-1}\mathbf{u}_j]$. Note that K is full column rank by construction. Our goal is to find vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ such that $[K(\mathbf{v}_1, d_1) | \dots | K(\mathbf{v}_m, d_m)]$ is full column rank (see \tilde{K} later).

For this purpose, we first need to consider the matrix \overline{K} made of columns of K so that it remains full column rank. It is defined as $\overline{K} := [\overline{K}_1 | \dots | \overline{K}_m]$ where for $1 \leq j \leq m$, $\overline{K}_j \in \mathbb{K}[x]^{m \times d_j}$ are defined iteratively by

$$\overline{K}_j := [K(\mathbf{u}_j, \min(L_j, d_j)) | K(x^{s_1}\mathbf{u}_{j_1}, t_1) | \dots | K(x^{s_k}\mathbf{u}_{j_k}, t_k)]$$

and $K(x^{s_l}\mathbf{u}_{j_l}, t_l)$ derives from previously unused columns in K_{j_l} , which we add from left to right, *i.e.* the sequence of indices (j_l) is increasing. Since $\sum_{i=1}^j d_i \leq \sum_{i=1}^j L_i$, we only pick from previous blocks, *i.e.* $j_k < j$. Since we need to complete a block K_{i_l} before going to another one, we can observe that $s_l + t_l = L_l$ for $l < k$. The last block K_{i_k} is the only one that may not be completed, *i.e.* $s_k + t_k \leq L_k$. Conversely, $s_l = d_l$ for $l > 1$ because no columns have been picked yet from the blocks j_l , except maybe the first block j_1 where $s_1 \geq d_1$.

We want to transform \overline{K}_j into a Krylov matrix \tilde{K}_j , working block by block. First we extend $[K(\mathbf{u}_j, \min(L_j, d_j)) | 0 | \dots | 0]$ to the right to $K(\mathbf{u}_j, d_j)$. Then we extend all blocks $[0 | \dots | 0 | K(x^{s_l}\mathbf{u}_{j_l}, t_l) | 0 | \dots | 0]$ to the left and the right to $K(x^{s'_l}\mathbf{u}_{j_l}, d_l)$ where s'_l equals s_l minus the number of columns of the left extension. In this way, the extension matches the original matrix on its non-zero columns. Now we can define $\tilde{K} := [\tilde{K}_1 | \dots | \tilde{K}_m]$, where $\tilde{K}_j := K(\mathbf{v}_j, d_j)$ with $\mathbf{v}_j := \mathbf{u}_j + \sum_{l=1}^k x^{s'_l}\mathbf{u}_{j_l}$.

A crucial point of the proof is to show that $s'_k \geq 0$. But since the d_i are non-increasing, j_l are increasing and $j_k < j$. So we get $s_l \geq d_{j_l} \geq d_{j_k} \geq d_j$. As the number of columns of the left extension is at most d_j , we can conclude $s'_k \geq 0$.

In [Vil97] and [PS07] it is proved that there exist an upper triangular matrices T such that $\tilde{K} = \overline{K}T$. So we can conclude that \tilde{K} , which is in the desired block Krylov form, is full column rank as is \overline{K} , which concludes the proof. \square

We now illustrate with an example the construction of the proof of Theorem 3.1.4.

Example 3.1.6. Let $m = 4$, $n = 3$, $\mathbf{L} = (8, 4, 4)$ and $\mathbf{d} = (5, 5, 3, 3)$. We put $L_4 = 0$.

Note that,

$$\begin{aligned} 5 &\leq 8 \\ 5 + 5 &\leq 8 + 4 \\ 5 + 5 + 3 &\leq 8 + 4 + 4 \\ 5 + 5 + 3 + 3 &\leq 8 + 4 + 4 + 0 \end{aligned}$$

and

1. $K_1 = K(\mathbf{u}_1, 8) = [\mathbf{u}_1 | x\mathbf{u}_1 | x^2\mathbf{u}_1 | x^3\mathbf{u}_1 | x^4\mathbf{u}_1 | x^5\mathbf{u}_1 | x^6\mathbf{u}_1 | x^7\mathbf{u}_1]$,
2. $K_2 = K(\mathbf{u}_2, 4) = [\mathbf{u}_2 | x\mathbf{u}_2 | x^2\mathbf{u}_2 | x^3\mathbf{u}_2]$,

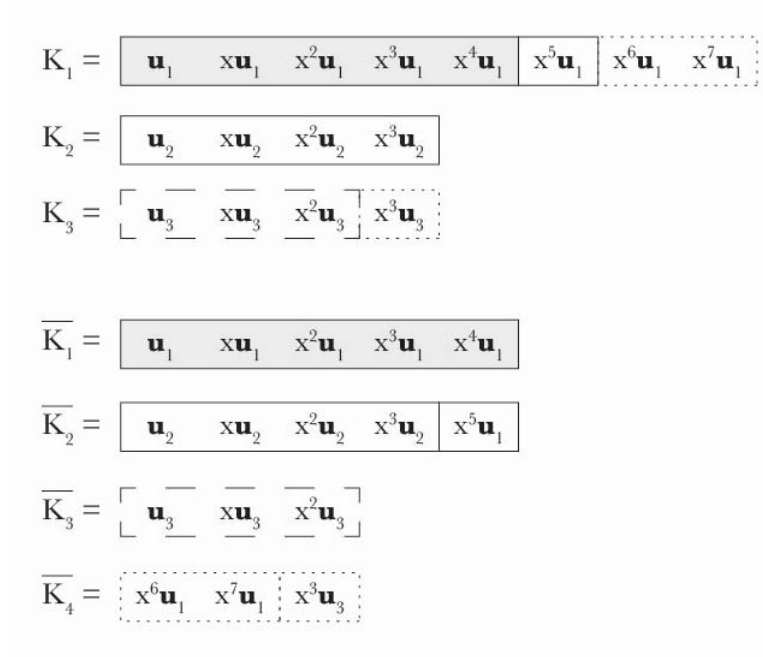


Figure 3.1: Construction of Example 3.1.6

3. $K_3 = K(\mathbf{u}_3, 4) = [\mathbf{u}_3 | x\mathbf{u}_3 | x^2\mathbf{u}_3 | x^3\mathbf{u}_3]$.

Now,

- since $\min(d_1, L_1) = d_1 = 5$, then $\overline{K}_1 = K(\mathbf{u}_1, d_1) = [\mathbf{u}_1 | x\mathbf{u}_1 | x^2\mathbf{u}_1 | x^3\mathbf{u}_1 | x^4\mathbf{u}_1]$,
- since $\min(d_2, L_2) = L_2 = 4$, then $\overline{K}_2 = [K(\mathbf{u}_2, L_2) | K(x^{d_1}\mathbf{u}_1, d_2 - d_1)] = [\mathbf{u}_2 | x\mathbf{u}_2 | x^2\mathbf{u}_2 | x^3\mathbf{u}_2 | x^5\mathbf{u}_1]$.
It picks its missing column from the first unused column of K_1 .
- since $\min(d_3, L_3) = d_3 = 3$, then $\overline{K}_3 = K(\mathbf{u}_3, d_3) = [\mathbf{u}_3 | x\mathbf{u}_3 | x^2\mathbf{u}_3]$,
- since $\min(d_4, L_4) = L_4 = 0$, then $\overline{K}_4 = [K(\mathbf{u}_4, L_4) = \emptyset | K(x^{d_1+1}\mathbf{u}_1, L_1 - (d_1 + 1)) | K(x^{d_3}\mathbf{u}_3, L_3 - d_3)] = [x^6\mathbf{u}_1 | x^7\mathbf{u}_1 | x^3\mathbf{u}_3]$. It picks its missing columns from the last 2 unused of K_1 and from K_3 .

Figure 3.1 illustrates the construction of $\overline{K}_1, \overline{K}_2$ and \overline{K}_3 . Finally in the proof, we extend \overline{K} to $\tilde{K} = [\tilde{K}_1 | \tilde{K}_2 | \tilde{K}_3 | \tilde{K}_4]$, where $\tilde{K}_i = K(\mathbf{v}_i, d_i)$. In detail,

- since $\overline{K}_1 = K(\mathbf{u}_1, d_1) = [\mathbf{u}_1 | x\mathbf{u}_1 | \dots | x^4\mathbf{u}_1]$ then $\mathbf{v}_1 = \mathbf{u}_1 = (1, 0, 0)^T$,
- we write

$$\overline{K}_2 = [\mathbf{u}_2 | x\mathbf{u}_2 | \dots | x^3\mathbf{u}_2 | 0] + [0 | 0 | 0 | 0 | x^5\mathbf{u}_1].$$

We now extend the first block to the right and the second to the left and we get,

$$\overline{K}_2 = [\mathbf{u}_2 | x\mathbf{u}_2 | x^2\mathbf{u}_2 | x^3\mathbf{u}_2 | \underbrace{0}_{x^4\mathbf{u}_2}] + [\underbrace{0}_{x\mathbf{u}_1} | \underbrace{0}_{x^2\mathbf{u}_1} | \underbrace{0}_{x^3\mathbf{u}_1} | \underbrace{0}_{x^4\mathbf{u}_1} | x^5\mathbf{u}_1].$$

In this way, by looking at the first column of both blocks we finally get

$$\mathbf{v}_2 = \mathbf{u}_2 + x\mathbf{u}_1 = (x, 1, 0)^T.$$

— Since $\overline{K}_3 = [\mathbf{u}_3 | x\mathbf{u}_3 | x^2\mathbf{u}_3]$, then $\mathbf{v}_3 = \mathbf{u}_3 = (0, 0, 1)^T$,

— $\overline{K}_4 = [x^6\mathbf{u}_1 | x^7\mathbf{u}_1 | x^3\mathbf{u}_3]$. Again we consider it as,

$$\overline{K}_4 = [x^6\mathbf{u}_1 | x^7\mathbf{u}_1 | 0] + [0 | 0 | x^3\mathbf{u}_3]$$

and we extend it, obtaining

$$\overline{K}_4 = [x^6\mathbf{u}_1 | x^7\mathbf{u}_1 | \underbrace{0}_{x^8\mathbf{u}_1}] + [\underbrace{0}_{x\mathbf{u}_3} | \underbrace{0}_{x^2\mathbf{u}_3} | x^3\mathbf{u}_3].$$

By looking at the first columns we get

$$\mathbf{v}_4 = x^6\mathbf{u}_1 + x\mathbf{u}_3 = (x^6, 0, x)^T.$$

Therefore the matrix of the Theorem 3.1.4 is

$$M = \begin{pmatrix} 1 & 0 & 0 \\ x & 1 & 0 \\ 0 & 0 & 1 \\ x^6 & 0 & x \end{pmatrix}$$



We can now state the principal constraint on the pivot degrees δ_M of the relation module $\mathcal{A}_{\mathcal{M}, M}$ where M varies in the set of matrices $\mathbb{K}[x]^{m \times n}$ such that the rank of O_M is fixed.

Theorem 3.1.6. *Recall that $\mathbf{L} = (L_1, \dots, L_m)$ are the degrees of the invariants of \mathcal{M} where $L_i = 0$ for $n + 1 \leq i \leq m$, and r is the rank of the ordered matrix O_M . Then $\mathcal{F}_{\mathbf{d}_r} \leq_{lex} \mathcal{F}_{\delta_M}$ where*

$$\mathcal{F}_{\mathbf{d}_r} = \min_{<_{LEX}} \left\{ \mathcal{F}_{\mathbf{d}} \in \text{Mon}_r \mid \forall 1 \leq l \leq m, \sum_{i=1}^l d_i \leq \sum_{i=1}^l L_i \right\} \quad (3.2)$$

Proof. We know from Proposition 3.1.3 that $RRP_M = \mathcal{F}_{\delta_M}$ so $\{x^i \varepsilon_j M\}_{i < \delta_j, M}$ are linearly independent in \mathcal{K} and $\sum_{i=1}^m \delta_{i, M} = r$. Using Theorem 3.1.4, we get that $\sum_{i=1}^l \delta_{i, M} \leq \sum_{i=1}^l L_i$ for all $1 \leq l \leq m$. This means that \mathcal{F}_{δ_M} belongs to the set whose minimum is $\mathcal{F}_{\mathbf{d}_r}$, which implies our result. \square

We observe that the rank r of the ordered matrix O_M satisfies the following $0 \leq r \leq \Sigma := \sum_{i=1}^m L_i$. Indeed, the dimension of \mathcal{K} as a \mathbb{K} -vector space is Σ .

3.1.4 Generic row degrees

The main aim of this subsection is to prove that the two families introduced so far are generically equal. Precisely we prove that for all matrices $M \in \mathbb{K}[x]^{m \times n}$ such that the rank r of the ordered matrix O_M is $r = \Sigma = \sum_{i=1}^m L_i$ (or equivalently such that the map φ_M is an isomorphism), then $\mathcal{F}_{\delta_M} = \mathcal{F}_{\mathbf{d}_\Sigma}$. This allows us to deduce that *generically* ρ_M , *i.e.* the \mathbf{s} -row degree of the relation module $\mathcal{A}_{\mathcal{M},M}$, coincides with $\mathbf{d}_\Sigma + \mathbf{s}$ (see (3.2)).

Corollary 3.1.2. *For almost all matrices $M \in \mathbb{K}[x]^{m \times n}$, with $m \geq n \geq 0$, $\delta_M = \mathbf{d}_\Sigma$.*

Proof. Our goal is to prove that there exists a non-zero polynomial C in the coefficients $m_{i,j,k}$ of the polynomial entries $m_{i,j}$ of M such that for any matrix with $C(m_{i,j,k}) \neq 0$, we have the equality $\delta_M = \mathbf{d}_\Sigma$.

Since $\sum_{i=1}^l \mathbf{d}_{\Sigma,i} \leq \sum_{i=1}^l L_i$ for all $1 \leq l \leq m$, we deduce from Theorem 3.1.4 that there exists $M \in \mathbb{K}[x]^{m \times n}$ such that $\{mM\}_{m \in \mathcal{F}_{\mathbf{d}_\Sigma}}$ are linearly independent. So the Σ -minor of the ordered matrix O_M corresponding to those rows is non-zero. We now consider this Σ -minor as a function C in the coefficients $m_{i,j,k}$ of the polynomial entries $m_{i,j}$ of M . Note that $C \in \mathbb{K}[m_{i,j,k}]$ since the entries of O_M are linear combinations of $m_{i,j,k}$. Indeed, we can write $m_{i,j} = \sum_{k=0}^{L_j-1} m_{i,j,k} x^k$ because the polynomials $m_{i,j}$ are only considered modulo a_j , and the coefficient of O_M (recall the definition of this matrix, subsection 3.1.2) *w.r.t* the row $x^u \varepsilon_i$ and the column $x^v \varepsilon'_j$ of O_M is $\sum_{k=0}^{L_j-1} m_{i,j,k} c_{j,k,u,v}$ where $c_{j,k,u,v} \in \mathbb{K}$ is the coefficient of $(x^{k+u} \bmod a_j)$ in x^v .

We saw that C admits a nonzero evaluation so it is a non-zero polynomial.

Now for any matrix M such that $C(m_{i,j,k}) \neq 0$, the vectors $\{mM\}_{m \in \mathcal{F}_{\mathbf{d}_\Sigma}}$ are linearly independent in \mathcal{K} , so the rank of O_M is equal to Σ . We have $RRP_M \leq_{\text{lex}} \mathcal{F}_{\mathbf{d}_\Sigma}$ because $\mathcal{F}_{\mathbf{d}_\Sigma} \in \mathcal{P}_M$ (see Definition 3.1.8). Theorem 3.1.6 gives the other inequality, so $\mathcal{F}_{\mathbf{d}_\Sigma} = RRP_M = \mathcal{F}_{\delta_M}$ and $\delta_M = \mathbf{d}_\Sigma$. \square

Hence, by summing up we have a characterization of the generic pivot degrees of relation modules.

Special cases. Under some assumptions on shifts and the degrees of the invariants, the generic row degrees of relation modules have a *simplified expression* which allows us to easily determine them.

Let $\bar{s} = \max_{1 \leq i \leq m} (s_i)$. Let us define p and u as the quotient and remainder of the Euclidean division of $\sum_{i=1}^m L_i + s_i$ by m , *i.e.* $\sum_{i=1}^m (L_i + s_i) = p \cdot m + u$. We prove that in this case the generic \mathbf{s} row degree $\rho_M = \mathbf{p}_\Sigma$ has this specific form

$$\mathbf{p} := \left(\underbrace{p+1, \dots, p+1}_{u \text{ times}}, \underbrace{p, \dots, p}_{m-u \text{ times}} \right). \quad (3.3)$$

We have this specific form if the following conditions on \mathbf{L} and \mathbf{s} hold:

$$p \geq \mathbf{s} \tag{3.4}$$

$$\forall 1 \leq l \leq m-1, \sum_{i=1}^l p_i \leq \sum_{i=1}^l (L_i + s_i) \tag{3.5}$$

Theorem 3.1.7. *Let \mathbf{p} as in (3.3), and let $\mathbf{L} = (L_1, \dots, L_m)$, where $L_1 \geq L_2 \geq \dots \geq L_m$ and such that (3.4) and (3.5) hold. Then, $\mathbf{p}_\Sigma := \mathbf{d}_\Sigma + \mathbf{s} = \mathbf{p}$.*

Proof. Recall that $\Sigma = \sum_{i=1}^n L_i$. Let $\overline{\mathcal{F}}$ be the first Σ monomials of $\mathbb{K}[x]^m$ sorted *w.r.t* $\prec_{\mathbf{s}\text{-TOP}}$ ordering. Let $\mathbf{p} = (p+1, \dots, p+1, p, \dots, p)$ be the candidate row degrees as in the theorem statement and $\mathbf{d} = \mathbf{p} - \mathbf{s}$ be the corresponding pivot degrees. Note that the assumption (3.4) implies that $\mathbf{d} \in \mathbb{N}^m$.

First we show that $p \geq \bar{s}$ implies $\overline{\mathcal{F}} = \mathcal{F}_{\mathbf{d}}$. In order to prove $\overline{\mathcal{F}} = \mathcal{F}_{\mathbf{d}}$, we need to show that $d_i = \min\{d \in \mathbb{N} \mid x^d \boldsymbol{\varepsilon}_i \notin \overline{\mathcal{F}}\}$. As already remarked, $d_i \in \mathbb{N}$. We need to study the row degrees of the first monomials to conclude.

Notice that the monomials of $\mathbb{K}[x]^m$ of \mathbf{s} -row degree r increasingly sorted *w.r.t* $\prec_{\mathbf{s}\text{-TOP}}$ are $x^{r-s_i} \boldsymbol{\varepsilon}_i$ for increasing $1 \leq i \leq m$ such that $s_i \leq r$. Now,

- if $r \geq \bar{s}$, there are exactly m of such monomials,
- otherwise, they are $\{x^i \boldsymbol{\varepsilon}_j\}_{i+s_j < \bar{s}}$ and their number is $\sum_{i=1}^m (\bar{s} - s_i)$.

From this we can deduce that the row degree of the t -th smallest monomial is

$$\left\lfloor (t-1 - \sum_{i=1}^m (\bar{s} - s_i))/m \right\rfloor + \bar{s} = \left\lfloor (t-1 + \sum_{i=1}^m s_i)/m \right\rfloor$$

provided that $t \geq \sum_{i=1}^m (\bar{s} - s_i) + 1$. Hence the $(\Sigma+1)$ -th smallest monomial has \mathbf{s} -row degree p . More precisely, the $(\Sigma+1)$ -th smallest monomial is the $(u+1)$ -th monomial of row-degree r , so $\overline{\mathcal{F}}$ is equal to all monomials of row degree less than p and the first u monomials of row degree p . This proves $d_i = \min\{d \in \mathbb{N} \mid x^d \boldsymbol{\varepsilon}_i \notin \overline{\mathcal{F}}\}$ and $\overline{\mathcal{F}} = \overline{\mathcal{F}}_{\mathbf{d}}$.

Second we deduce from assumption 3.5 that for all $1 \leq l \leq m$, $\sum_{i=1}^l d_i = \sum_{i=1}^l (p_i - s_i) \leq \sum_{i=1}^l L_i$, so $\mathcal{F}_{\mathbf{d}_r} \leq_{\text{lex}} \mathcal{F}_{\mathbf{d}}$ by Theorem 3.1.6 and finally $\mathcal{F}_{\mathbf{d}_r} = \mathcal{F}_{\mathbf{d}}$ because $\overline{\mathcal{F}}$ is the smallest set of Σ monomials. \square

This specific form of row degree was already observed in different but related settings. To the best of our knowledge, it can be found in [Vil97, Proposition 6.1] for row degrees of minimal generating matrix polynomial and in [PS07, Corollary 1] for dimensions of blocks in a shifted *Hessenberg form* and in [JV05] for kernel basis were $m = 2n$. In all the three cases authors do not study the shifted case (the shifted Hessenberg form is not related to the shift that we defined).

Example 3.1.7. Let $m = n = 3$ and $\mathbf{s} = (0, 2, 4)$. So $\bar{s} = 4$ and $\sum(\bar{s} - s_i) = 6$.

1. Consider $\mathbf{L} = (6, 1, 0)$. Then $\sum_{i=1}^3 L_i + s_i = 13 = 4 \cdot m + 1$. Notice that (3.4) and (3.5) are both verified. Hence $\mathbf{p}_\Sigma = (5, 4, 4)$ and so $\mathbf{d}_\Sigma = (5, 2, 0)$. As we can see in the following table, $\overline{\mathcal{F}}_1$, *i.e.* the set of the first $\Sigma = \sum_{i=1}^m L_i = 7$ monomials of $\mathbb{K}[x]^3$, increasingly sorted *w.r.t* $\prec_{\mathbf{s}-TOP}$ coincides exactly with $\mathcal{F}_{\mathbf{d}_{\Sigma,1}} := \mathcal{F}_{\mathbf{d}_\Sigma} = \{\varepsilon_1, \dots, x^4\varepsilon_1, \varepsilon_2, x\varepsilon_2\}$.
2. Consider $\mathbf{L} = (3, 0, 0)$. Then $\sum_{i=1}^3 L_i + s_i = 9 = 3 \cdot m$. Notice that in this case $p = 3 \leq \bar{s} = 4$ and so (3.4) does not hold. By Corollary 3.1.2, $\mathbf{d}_\Sigma = (3, 0, 0)$ since $\sum_{i=1}^l \mathbf{d}_{\Sigma,i} \leq \sum_{i=1}^l L_i$ for any $1 \leq l \leq m$. So the row degrees are $\mathbf{p}_\Sigma = (3, 2, 4)$. The set $\overline{\mathcal{F}}_2$ of the first $\Sigma = 3$ monomials, increasingly sorted *w.r.t* $\prec_{\mathbf{s}-TOP}$ coincides with $\mathcal{F}_{\mathbf{d}_{\Sigma,2}} := \mathcal{F}_{\mathbf{d}_\Sigma}$.
3. Consider $\mathbf{L} = (3, 3, 1)$. Then $\sum_{i=1}^3 L_i + s_i = 13 = 4 \cdot m + 1$. Let $\mathbf{p} = (5, 4, 4)$. Note that $p = 4 \geq 4 = \bar{s} = 4$ but since $p_1 = 5 \geq L_1 + s_1 = 3$ then (3.5) does not hold. By Corollary 3.1.2, $\mathbf{d}_\Sigma = (3, 3, 1)$ since $\sum_{i=1}^l \mathbf{d}_{\Sigma,i} \leq \sum_{i=1}^l L_i$ for any $1 \leq l \leq m$. So the row degrees are $\mathbf{p}_\Sigma = (3, 5, 5)$. Notice that in this case the set $\overline{\mathcal{F}}_3$ of the first $\Sigma = 7$ monomials, increasingly sorted *w.r.t* $\prec_{\mathbf{s}-TOP}$ is different from $\mathcal{F}_{\mathbf{d}_{\Sigma,3}} := \mathcal{F}_{\mathbf{d}_\Sigma} = \{\varepsilon_1, \dots, x^2\varepsilon_1, \varepsilon_2, \dots, x^2\varepsilon_2, \varepsilon_3\}$.

<i>Mon</i>	ε_1	$x\varepsilon_1$	$x^2\varepsilon_1$	ε_2	$x^3\varepsilon_1$	$x\varepsilon_2$	$x^4\varepsilon_1$	$x^2\varepsilon_2$	ε_3
$\text{rdeg}_{\mathbf{s}}$	0	1	2		3		4		
$\mathcal{F}_{\mathbf{d}_{\Sigma,1}}$	•	•	•	•	•	•	•		
$\mathcal{F}_{\mathbf{d}_{\Sigma,2}}$	•	•	•						
$\mathcal{F}_{\mathbf{d}_{\Sigma,3}}$	•	•	•	•		•		•	•



3.2 Generic row degrees of the SRFR Relation Module

We can now transpose all these results to the specific case of SRFR in order to prove our main result (Theorem 3.2.1). First we recall SRFR (Problem 2). Let $a_1, \dots, a_n \in \mathbb{K}[x]$ with degrees $L_i := \deg(a_i)$ and $\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{K}[x]^n$ such that $\deg(u_i) < L_i$. Moreover let $1 \leq N_i \leq L_i$ for $1 \leq i \leq n$ and $1 \leq D \leq \min_{1 \leq i \leq n} \{\deg(a_i)\}$. The goal of SRFR is to reconstruct $(\mathbf{v}, d) \in \mathbb{K}[x]^{n+1}$ such that for any $1 \leq i \leq n$

$$v_i = du_i \text{ mod } a_i, \quad \deg(v_i) < N_i, \quad \deg(d) < D. \quad (3.6)$$

We consider $\mathcal{M} = \langle a_i(x)\varepsilon'_i \rangle_{1 \leq i \leq n}$ and $\mathcal{S}_{\mathbf{u}} := \{(\mathbf{v}, d) \mid \text{satisfying (3.6)}\}$. Recall that by Lemma 1.3.8

$$(\mathbf{v}, d) \in \mathcal{S}_{\mathbf{u}} \iff \begin{cases} (\mathbf{v}, d) \in \mathcal{A}_{\mathcal{M}, R_{\mathbf{u}}} \\ \text{rdeg}_{\mathbf{s}}((\mathbf{v}, d)) < 0 \end{cases} \quad (3.7)$$

where, the shift $\mathbf{s} = (-N_1, \dots, -N_n, -D)$ and

$$R_{\mathbf{u}} := \begin{bmatrix} I_n \\ -\mathbf{u} \end{bmatrix}$$

Therefore the $\mathbb{K}[x]$ -module of solutions of SRFR is generated by elements of the relation module $\mathcal{A}_{\mathcal{M}, R_{\mathbf{u}}}$ with negative shifted row degree.

We now denote by $\boldsymbol{\rho}_{\mathbf{u}} := \boldsymbol{\rho}_{R_{\mathbf{u}}}$ and $\boldsymbol{\delta}_{\mathbf{u}} := \boldsymbol{\delta}_{R_{\mathbf{u}}}$, respectively the \mathbf{s} -row degrees and the \mathbf{s} -pivot degrees of ordered weak Popov bases of the relation module $\mathcal{A}_{\mathcal{M}, R_{\mathbf{u}}}$. As already remarked in Section 1.2, by Remark 1.3.3,

$$\dim(\mathcal{S}_{\mathbf{u}}) = \sum_{\rho_{\mathbf{u},i} < 0} -\rho_{\mathbf{u},i} \quad (3.8)$$

We are now ready to introduce the main result of this section.

Theorem 3.2.1. *If $\sum_{i=1}^n L_i = \sum_{i=1}^n N_i + D - 1$ then for almost all instances \mathbf{u} , SRFR admits a unique solution.*

Moreover, if $\mathbb{K} = \mathbb{F}_q$ the proportion of instances leading to nonuniqueness is at most $(D - 1)/q$.

Proof. By the previous considerations it suffices to prove that for almost all $\mathbf{u} \in \mathbb{K}[x]^n$, then $\boldsymbol{\rho}_{\mathbf{u}} = (0, \dots, 0, -1)$.

We divide the proof in two steps:

1. first we observe that $\sum_{i=1}^n L_i + s_i = (\sum_{i=1}^n (L_i - N_i)) - D = -1$ and by performing the Euclidean division of $\sum_{i=1}^n L_i + s_i$ by $n + 1$ we get

$$\sum_{i=1}^n L_i + s_i = -1 \cdot (n + 1) + n.$$

In this first part we prove that the assumptions (3.4) and (3.5) of Theorem 3.1.7 are satisfied so that we can conclude that the generic row degree is of the specific form $\boldsymbol{\rho}_{\Sigma} = (0, \dots, 0, -1)$, where $\Sigma = \sum_{i=1}^n L_i$.

2. Then we prove that there exists $\mathbf{u} \in \mathbb{K}[x]^n$ such that the corresponding matrix $R_{\mathbf{u}}$ satisfies the generic condition of Corollary 3.1.2 and so $\boldsymbol{\rho}_{\mathbf{u}} = \boldsymbol{\rho}_{\Sigma}$.

1. Notice that $p = -1$ and $\bar{s} = \max_{1 \leq i \leq n+1} \{s_i\} \leq -1$. So (3.4) is verified. On the other hand also (3.5) holds since for any $1 \leq i \leq n + 1$, we have that $p_i \leq 0 \leq L_i + s_i$.

2. We now show that the construction of Theorem 3.1.4 provides a matrix of the form $R_{\mathbf{u}}$.

Notice that in this case, $(d_1, \dots, d_{n+1}) = (N_1, \dots, N_n, D - 1)$. By SRFR assumptions, for any $1 \leq i \leq n$, $d_i \leq L_i$ and so $\bar{K}_i = [K(\mathbf{u}_i, d_i)]$. Therefore, $\tilde{K}_i = \bar{K}_i$, for $1 \leq i \leq n$. The last matrix $\bar{K}_{n+1} = [K(x^{d_1} \mathbf{u}_1, t_1) | \dots | K(x^{d_n} \mathbf{u}_n, t_n)]$ where $d_j + t_j = L_j$ for any $1 \leq j \leq n$ (see Figure 3.2).

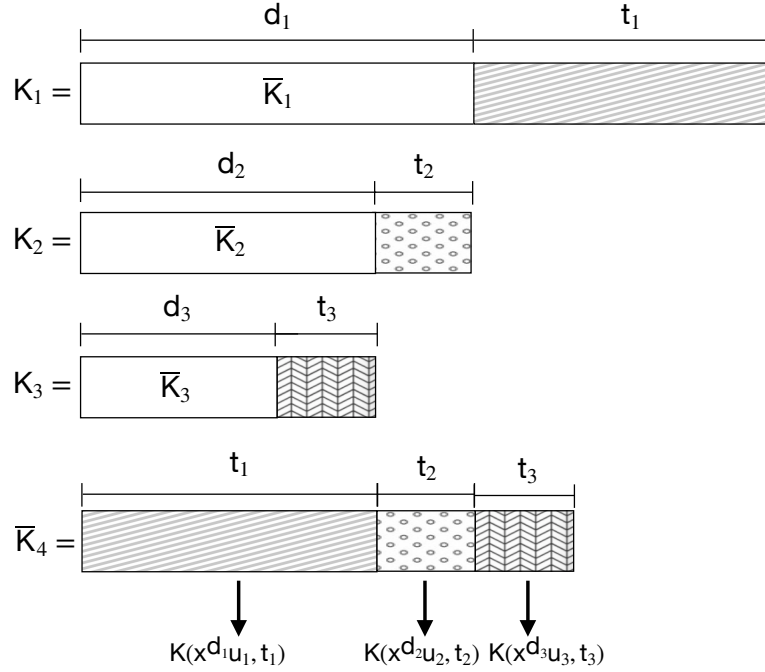


Figure 3.2: Construction of Krylov matrices of the proof of Theorem 3.2.1. In this case $n = 3$.

We then extend blocks of the matrix \bar{K}_{n+1} to get \tilde{K}_{n+1} (see Figure 3.3). Note that we have $\tilde{K}_{n+1} = [K(\mathbf{v}_{n+1}, d_{n+1})]$, where $\mathbf{v}_{n+1} = \sum_{j=1}^n x^{s'_j} \mathbf{u}_j$. We now need to show that $s'_j \geq 0$. We cannot use the same technique as the proof of Theorem 3.1.4 since the sequence of (d_i) is not increasing.

Fix j , recall that s'_j is equal to s_j which in this case is exactly d_j minus the number of columns added to extend the block to the left (see Figure 3.3). This number of columns is at most $d_{n+1} - t_j$. Therefore,

$$s'_j \geq d_j - (d_{n+1} - t_j) = d_j - (d_{n+1} - (L_j - d_j)) = L_j - d_{n+1} \geq 0$$

since $d_{n+1} = D - 1 \leq D \leq \min_{1 \leq i \leq n} \{L_i\}$.

So, with this construction we obtain $\mathbf{v}_i = \boldsymbol{\varepsilon}_i$ for $1 \leq i \leq n$ and $\mathbf{v}_{n+1} = \sum_{j=1}^n x^{s'_j} \mathbf{u}_j$. Note that the matrix whose rows are determined by these \mathbf{v}_i is of the form

$$M = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \dots & 1 \\ * & * & * & \dots & * \end{pmatrix}$$

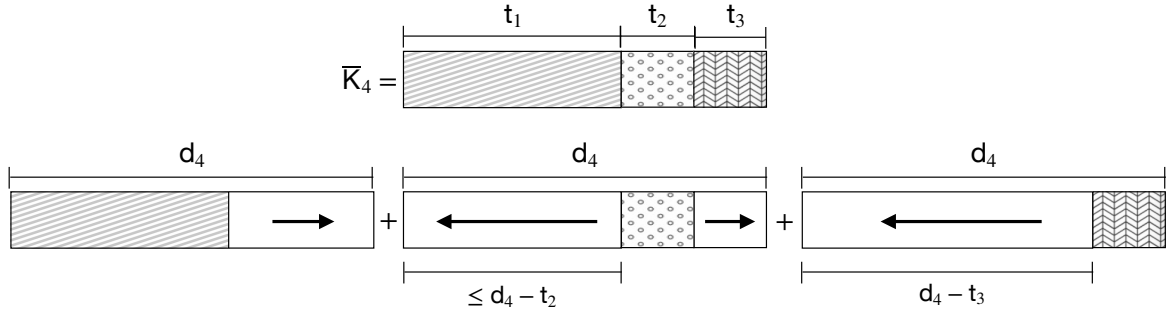


Figure 3.3: Construction of the last Krylov matrix of the proof of Theorem 3.2.1. In this case $n = 3$.

where $*$ represent polynomials in $\mathbb{K}[x]$. This shows that there exists \mathbf{u} such that the polynomial C (see proof of Corollary 3.1.2) does not vanish in $R_{\mathbf{u}}$. By the particular form of $R_{\mathbf{u}}$, we can consider C as a polynomial whose indeterminates are $u_{j,k}$, where for any $1 \leq j \leq n$, $u_j = \sum_{k=1}^{L_j-1} u_{j,k} x^k$. Recall that C is the Σ -minor of the ordered matrix $O_{R_{\mathbf{u}}}$ and note that all the coefficients of this matrix are scalar elements of the field, except for the $D - 1$ rows corresponding to $\{x^l \boldsymbol{\varepsilon}_{n+1}\}_{0 \leq l < D-1}$ which are linear combinations of $u_{j,k}$ (as remarked in the proof of Corollary 3.1.2). Therefore, the total degree of C is at most $D - 1$ and if $\mathbb{K} = \mathbb{F}_q$, by Schwartz-Zippel Lemma the proportion of instances leading to non-uniqueness among all possible instances is at most $(D - 1)/q$. \square

3.3 Conclusions and open problems

We now draw the conclusions on the main results of this section and we discuss about the related open problems.

In this thesis we focus on the simultaneous rational function reconstruction.

Problem 2. *Simultaneous Rational Function Reconstruction*

Input: $a_1, \dots, a_n \in \mathbb{K}[x], \mathbf{u} = (u_1, \dots, u_n) \in \mathbb{K}[x]^n$, where $\deg(u_i) < \deg(a_i)$
and $1 \leq N_i \leq \deg(a_i), 1 \leq D \leq \min_{1 \leq i \leq n} \{\deg(a_i)\}$

Output: $(\mathbf{v}, d) = (v_1, \dots, v_n, d) \in \mathbb{K}[x]^{n+1}$ such that

$$[v_i = du_i \bmod a_i]_{1 \leq i \leq n}, \quad \deg(v_i) < N_i, \quad \deg(d) < D. \quad (1.5)$$

In other terms, this is the problem of recovering several rational functions with the same denominator, given their remainders modulo different polynomials. As for the classical ratio-

nal function reconstruction problem (Problem 1), we drop the assumption on the invertibility of the denominator and we focus on the *weaker* equation (1.5). Indeed, this equation has the advantage to be linear and allows us to focus on the homogeneous linear system related to it, in order to deduce some useful information about the existence of a nontrivial solution and the uniqueness.

We observe that in the *interpolation case*, when $a_1 = \dots = a_n = \prod_{i=1}^L (x - \alpha_i)$, where $\{\alpha_1, \dots, \alpha_L\}$ are pairwise distinct elements of the field, then (1.5) becomes an equation on the evaluations

$$[v_i(\alpha_j) = d(\alpha_j)u_i(\alpha_j)]_{\substack{1 \leq i \leq n, \\ 1 \leq j \leq L}}, \quad \deg(v_i) < N_i, \quad \deg(d) < D.$$

So, in this case SRFR becomes the problem of recovering a vector of rational functions with the same denominator, given its evaluations.

In Chapter 1, we saw that in order to solve SRFR we could apply the rational function reconstruction componentwise and that we have uniqueness of SRFR whenever

$$\deg(a_i) = N_i + D - 1, \quad \text{for any } 1 \leq i \leq n \implies \sum_{i=1}^n \deg(a_i) = \sum_{i=1}^n (N_i + D - 1).$$

Alternatively, we could use the fact that all the rational functions that we want to recover share the same denominator to decrease the number of unknowns of the homogeneous linear system related to (1.5). As a matter of fact if

$$\sum_{i=1}^n \deg(a_i) = \sum_{i=1}^n N_i + D - 1 \tag{3.9}$$

SRFR admits a nontrivial solution. The main aim of this work is to study, under the assumption (3.9), instances leading to uniqueness. There were two results which motivate us to pursue our research in this direction: the former is related to the *computer algebra* application of SRFR to polynomial linear system solving (see Section 1.4) and the latter is related to the *coding theory* application of the decoding of interleaved Reed-Solomon codes (see Subsection 2.3.1).

- In [OS07] (see Lemma 1.4.2), Z. Olesh and A. Storjohann proved that in the case of the polynomial linear system solving, under some specific assumptions on the degree bounds $N_1 = \dots = N_n = N = D = n \deg(A) + 1$ and by considering $a_1 = \dots = a_n$, if $\deg(a) = N + (D - 1)/n$ (which coincides with (3.9)), then we have uniqueness of the corresponding SRFR.
- In Section 2.3 we saw how the interpolation-based decoding technique for interleaved Reed-Solomon codes can be reduced to SRFR.

More specifically, consider n -IRS code of length L and dimension N , L evaluation

points $\{\alpha_1, \dots, \alpha_L\}$ and a received vector $Y = (\mathbf{f}(\alpha_1), \dots, \mathbf{f}(\alpha_L)) + \Xi \in \mathbb{F}_q^{n \times L}$, where $\mathbf{f} \in \mathbb{F}_q[x]^{n \times 1}$ with $\deg(\mathbf{f}) < N$ (see Definition 2.3.1). Recall that Ξ is the error matrix with error support $E = \{j \mid \Xi_{*,j} \neq \mathbf{0}\}$.

In order to decode Y , we can solve SRFR in the interpolation version and find (φ, λ) such that for any $1 \leq j \leq L$,

$$\varphi(\alpha_j) = Y_{*,j}\lambda(\alpha_j), \quad \deg(\varphi) < \varepsilon + k, \quad \deg(\psi) < \varepsilon + 1$$

where $\varepsilon \geq |E|$, is a bound on the number of errors. Recall that $(\Lambda \mathbf{f}, \Lambda)$, where Λ is the error locator polynomial (see Definition 2.2.4), is a solution of this specific SRFR. In this context, uniqueness of SRFR means that we can uniquely recover such a solution and so the vector of polynomials related to the codeword of this IRS.

We now recall that Lemma 2.3.1 basically tells us that if

$$L = (N + \varepsilon) + (\varepsilon + 1 - 1)/n$$

(which coincides with (3.9) since $L = \deg(a) = \deg(\prod_{i=1}^L (x - \alpha_i))$) then for almost all error matrices Ξ with error support E , then SRFR admits a unique solution for the corresponding instance Y .

So, even if this result is true in an error scenario, we thought that it could suggest something about the uniqueness of SRFR also for the general case.

Therefore, the main result (Theorem 3.2.1) of this chapter states that under (3.9), for almost all instances \mathbf{u} , SRFR admits a unique solution. This represents a step towards the following conjecture.

Conjecture 3.3.1. If (3.9) is satisfied then for almost all $(\mathbf{v}, d) \in \mathbb{K}[x]^{n+1}$ with $\gcd(d, a_i) = 1$, for any $1 \leq i \leq n$, then SRFR with instance $\mathbf{u} = \frac{\mathbf{v}}{d}$ admits a unique solution.

We strongly believe in this conjecture, which is suggested by some tests that we implemented in **SageMath**. Indeed, we recall that here we dropped the gcd assumption, in order to focus on the linear problem (1.5). Therefore, informally speaking, instances \mathbf{u} of the SRFR problem in its linear version (Problem 2) may not derive from vector of rational functions. So, in order to formally prove the conjecture above, since we proved the existence for a generic instance, it would be sufficient to show the existence of an instance \mathbf{u} of the form $\frac{\mathbf{v}}{d} \bmod \mathbf{a}$. In the next chapter, we will see how this conjecture is crucial also in the error correction setting case.

CHAPTER 4

Simultaneous Cauchy interpolation with errors

Contents

4.1	ABFT for Polynomial Linear System Solving by Evaluation-Interpolation	103
4.1.1	Simultaneous Cauchy interpolation with errors	104
4.1.2	Polynomial Linear Solving with Errors	111
4.2	Early termination techniques	115
4.2.1	Previous results	117
4.2.2	Our contribution	123
4.3	Conclusion and open problems	129

The main aim of error correcting codes is to correct errors which can be introduced by noisy channels. Besides, they have others applications which do not involve communication over a channel: for instance they can be used to correct and detect computational errors.

High Performance Computing Technologies (supercomputers) contain thousands of computing nodes networked together (parallel computing) to provide very high performances and to complete heavy tasks. The more the number of system components grows, the more the failure of computing nodes becomes relevant. For instance, modern supercomputers commit approximately 3.5 faults per day [DGP⁺19, LC18]. Therefore, without a drastic change at the algorithmic level such a failure rate will certainly prevent supercomputers from progressing. For this reason, many *fault tolerant* techniques and algorithms have been proposed to detect and correct these faults.

Faults can be distinguished into hard and soft faults: hard faults, also called *fail-stop*, include hardware failures and cause an immediate interruption of processes. While soft faults are more *subtle* faults which basically do not lead to any interruption. Examples are bit flips and data corruption due to a number of possible causes, including occasional cosmic rays.

The hard faults can be handled by *checkpoint-restart* techniques whose main principle consists of periodically saving data onto a reliable storage device; the system can then recover from the most recent checkpoint whenever an hard fault occurs [BD93]. However, this tech-

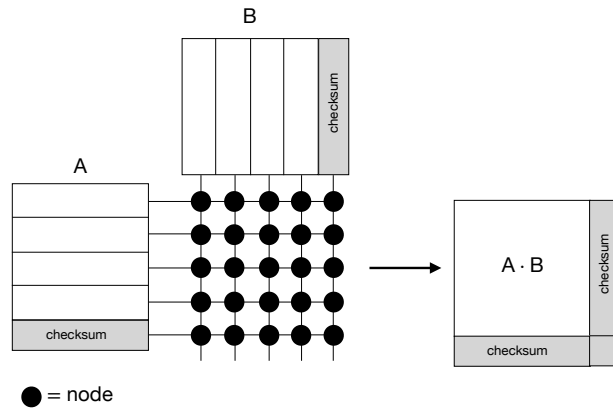


Figure 4.1: Matrix multiplication by ABFT method.

nique is expensive in terms of time and resources, since it requires a certain time to recover data and it could require external data storage and network bandwidth.

To overcome drawbacks of checkpoint-restart techniques, K-H. Huang and J.A. Abraham introduced the *algorithm-based fault tolerant* (ABFT) technique [HA84] which, exploiting algorithm’s characteristics, allows one to design a fault tolerant algorithm which can detect and/or correct faults. More specifically, this technique enriches the input of a given algorithm with redundancy (*e.g.* by using algebraic tools of error correcting codes) in order to correct computational errors which occur in parallel-distributed environments. It is characterized by

- the *encoding* of inputs of the algorithm,
- the *redesign* of the algorithm to operate on the encoding data,
- the *distribution* of some computation steps of the algorithm among some computational units.

For instance, consider the matrix multiplication of $A, B \in \mathbb{K}^{n \times n}$. The encoding of the two matrices is obtained by adding to A (respectively to B) a *checksum* row (column), whose components are determined by the sum of any element of the column (row) of A (B). Then, the multiplication row by column of A and B are performed by different nodes (parallelization) (see Figure 4.1). In this way, an error is detectable and correctable by checking if the sum of any row and column of the resulting matrix coincides with the corresponding checksum.

Notice that in this framework, the error model strongly depends on the chosen parallelization scheme: *e.g.* in the matrix multiplication example above, errors are introduced in the computation of any element of the product matrix.

Recently, many fault tolerant algorithms have been proposed for classic computer algebra problems: *e.g.* Chinese remaindering [BDFP15, GRS00, KPR⁺10], matrix multiplication and inversion [GLL⁺17, Pag13, Roc18], LU factorization [DHPR19].

In this chapter we focus on an algorithm-based fault tolerant technique for polynomial

linear system solving (Section 1.4) by evaluation-interpolation [BK14, KPSW17]. We describe our results of [GLZ19] and new contributions in progress [GLZ20a].

4.1 ABFT for Polynomial Linear System Solving by Evaluation-Interpolation

Consider a polynomial linear system (PLS) (see Section 1.4),

$$A(x)\mathbf{y}(x) = \mathbf{b}(x) \quad (4.1)$$

where $A \in \mathbb{F}_q[x]^{n \times n}$ is nonsingular and $\mathbf{b} \in \mathbb{F}_q[x]^{n \times 1}$. Recall that by Lemma 1.4.1, this system admits only one solution $\mathbf{y} = \frac{\mathbf{v}}{d} \in \mathbb{F}_q(x)$ which is a vector of rational functions with the same denominator. We assume that $\gcd(\gcd(v_1, \dots, v_n), d) = 1$ and that d is monic.

Our goal in this section is to introduce an algorithm-based fault tolerant technique for PLS solving by *evaluation-interpolation*.

Recall from Section 1.4 that the evaluation-interpolation technique for the PLS solving consists in:

1. (*evaluation*) given L distinct evaluation points $\{\alpha_1, \dots, \alpha_L\}$, evaluate the polynomial matrix A and the polynomial vector \mathbf{b} at these points.

Recall that in Section 1.4, we saw how to handle the rank drop case: it suffices to add $r \geq |R| = |\{j \mid \det(A(\alpha_j)) = 0\}|$ evaluation points to the number of points which allows one to uniquely recover the solution of the PLS. For simplicity, from now on, we assume that all the evaluation points do not cause rank drops of the corresponding evaluated matrices.

2. (*Pointwise resolution of the evaluated systems*) Compute for any $1 \leq j \leq L$, $\mathbf{y}(\alpha_j) = A(\alpha_j)^{-1}\mathbf{b}(\alpha_j) = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$.
3. (*Interpolation*) Reconstruct (\mathbf{v}, d) , given $\mathbf{y}(\alpha_j)$ for $1 \leq j \leq L$ and some bounds $N > \deg(\mathbf{v})$, $D > \deg(d)$. Or in other terms, perform SRF (Problem 2, in the interpolation form *i.e.* $a_1 = \dots = a_n = a = \prod_{j=1}^L (x - \alpha_j)$) with input $Y := (\mathbf{y}(\alpha_j))_{1 \leq j \leq L} \in \mathbb{F}_q^{L \times n}$, and N, D .

We also recall that the minimum number of evaluation points that we need to uniquely reconstruct the solution (see Section 1.4) is

$$L' := \min\{N + D - 1, \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}\} \quad (4.2)$$

As previously pointed out, ABFT techniques are characterized by the *encoding* of inputs, the *parallelization* of computations and the *redesign* of the algorithm so that it can manage

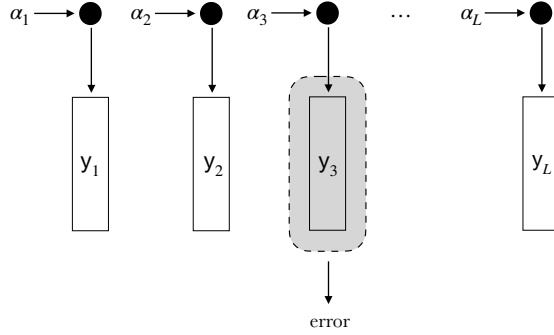


Figure 4.2: Parallelization in the evaluation step of PLS solving

the so obtained encoded data. Here, following this scheme we modify the classic evaluation-interpolation technique for PLS solving so that,

- (*Encoding*) inputs are encoded by considering *more* evaluation points than L' (adding redundancy),
- (*parallelization*) the evaluation step is performed by different nodes which can introduce some errors,
- (*redesign, interpolation with errors*) perform the interpolation step on the encoded data.

In what follows we formally detail this scheme.

Parallelization and error model. Fix L pairwise distinct evaluation points $\{\alpha_1, \dots, \alpha_L\}$. Assume that any node, which we can consider as a black box, computes $\mathbf{y}_j = A(\alpha_j)^{-1}\mathbf{b}(\alpha_j) \in \mathbb{F}_q^{n \times 1}$ for any $1 \leq j \leq L$ (Figure 4.2). These nodes could make some errors and compute $\mathbf{y}_j \neq A(\alpha_j)^{-1}\mathbf{b}(\alpha_j) = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$. Notice that, in this error model, the number of errors coincides with the number of nodes which compute an incorrect result. Hence, we assume that after the parallelization we get

$$Y = \left(\frac{\mathbf{v}(\alpha_1)}{d(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{d(\alpha_L)} \right) + \Xi \quad (4.3)$$

where $\Xi \in \mathbb{F}_q^{n \times L}$ is the error matrix, whose error support is $E := \{j \mid \Xi_{*,j} \neq \mathbf{0}\}$.

So, given the matrix Y , some bounds $N > \deg(\mathbf{v})$, $D > \deg(d)$ and $\deg(A)$, $\deg(\mathbf{b})$ our goal is to correct such errors and recover the solution (\mathbf{v}, d) of the PLS. We call this problem *polynomial linear system solving with errors* (PLSwE).

4.1.1 Simultaneous Cauchy interpolation with errors

Polynomial linear system solving with errors consists in the recovering of a vector of rational functions $\frac{\mathbf{v}}{d}$, which in this case is the solution of a PLS (4.1), given its evaluations (see (4.3)), where some are erroneous or corrupted. It is then an application of the following more general problem, that we introduced in [GLZ19].

Definition 4.1.1 (Simultaneous Cauchy interpolation with errors). Fix some parameters L, τ, N, D, q and $n \geq 1$, where

- L is the number of evaluation points $\{\alpha_1, \dots, \alpha_L\}$, $1 \leq L \leq q$,
- N, D are the degree bounds, $1 \leq N, D \leq L$,
- τ is the bound on the number of errors, $0 \leq \tau \leq L$.

A satisfiable instance of the *simultaneous Cauchy interpolation with errors* (shortly SCIwE) problem is a matrix $Y \in \mathbb{F}_q^{n \times L}$ such that there exist,

- a vector of rational functions $\frac{\mathbf{v}}{d} \in \mathbb{F}_q(x)^{n \times 1}$, where $\deg(\mathbf{v}) < N$, $\deg(d) < D$, $\gcd(\gcd_i(v_i), d) = 1$, d is monic and for any $1 \leq j \leq L$ then $d(\alpha_j) \neq 0$,
- an *error matrix* $\Xi \in \mathbb{F}_q^{n \times L}$ with error support $E := \{j \mid \Xi_{*,j} \neq \mathbf{0}\}$, where $|E| \leq \tau$,

which satisfy the following

$$Y = \left(\frac{\mathbf{v}(\alpha_1)}{d(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{d(\alpha_L)} \right) + \Xi. \quad (4.4)$$

SCIwE is the problem of finding a vector of rational functions $\frac{\mathbf{v}}{d}$ as in (4.4) given an instance Y .

We observe that this problem is the rational extension (see Figure 4.3) of the simultaneous interpolation with errors (SIwE, Definition 2.3.2) which is the problem of decoding an interleaved Reed-Solomon code. Indeed, here we want to reconstruct a vector of rational functions instead of a vector of polynomials. This link allows us to extend the same interpolation-based technique for decoding IRS codes to this rational case (as we proved in [GLZ19]).

In detail, from now on we fix an instance $Y = \left(\frac{\mathbf{v}(\alpha_1)}{d(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{d(\alpha_L)} \right) + \Xi$ of SCIwE with parameters L, τ, N, D, q and evaluation points $\{\alpha_1, \dots, \alpha_L\}$. For any $1 \leq j \leq L$, we denote $\mathbf{y}_j := Y_{*,j}$.

As for IRS and also RS codes, we can consider the *error locator polynomial* (see Definition 2.2.4),

$$\Lambda = \prod_{j \in E} (x - \alpha_j)$$

where $E = \{j \mid \Xi_{*,j} \neq \mathbf{0}\}$ is the error support. Therefore, we notice that for any $1 \leq j \leq L$,

$$\Lambda(\alpha_j)\mathbf{v}(\alpha_j) = \mathbf{y}_j d(\alpha_j)\Lambda(\alpha_j) \iff \Lambda(\alpha_j)[\mathbf{v}(\alpha_j) - \mathbf{y}_j d(\alpha_j)] = \mathbf{0}. \quad (4.5)$$

Indeed if $j \in E$ then since $\Lambda(\alpha_j) = 0$, then (4.5) becomes the identity $\mathbf{0} = \mathbf{0}$. On the other hand, if $j \notin E$ then $\mathbf{y}_j = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$ and so (4.5) becomes the identity $\mathbf{0} = \mathbf{0}$.

As IRS codes (and also for classic RS codes), we can linearize this equation by replacing the polynomial $\Lambda\mathbf{v}$ by φ and Λd by ψ , thus obtaining for any $1 \leq j \leq L$ the following *key equation*,

$$\varphi(\alpha_j) = \mathbf{y}_j \psi(\alpha_j), \quad \deg(\varphi) < N + \tau, \quad \deg(\psi) < D + \tau. \quad (4.6)$$

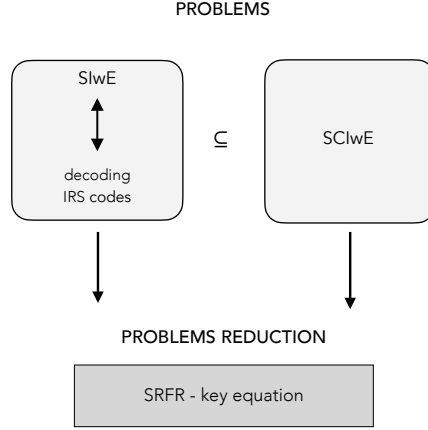


Figure 4.3: Scheme which illustrates the relation between SIwE and SCIwE.

So, we reduced the problem of reconstructing the vector of rational functions to the problem of finding the couple (φ, ψ) which satisfies the key equation (4.6), or equivalently SRFR (in the interpolation form) with input $Y, N + \tau, D + \tau$ (see Problem 2).

Remark 4.1.1. We denote by $\mathcal{S}_{Y,N,D,\tau} := \{(\varphi_1, \dots, \varphi_n, \psi) \text{ satisfying (4.6)}\}$. Again, we use this notation to stress the dependency on the instance Y and on the parameters τ, N, D . If we consider the homogeneous linear system related to (4.6), we observe that the set of solutions $\mathcal{S}_{Y,N,D,\tau}$ is the kernel of the matrix

$$M_{Y,N,D,\tau} = \left(\begin{array}{cccc|c} V_{L,N+\tau} & & & & -D_1 V_{L,D+\tau} \\ & V_{L,N+\tau} & & & -D_2 V_{L,D+\tau} \\ & & \ddots & & \vdots \\ & & & V_{L,N+\tau} & -D_n V_{L,D+\tau} \end{array} \right) \quad (4.7)$$

where $V_{L,N+\tau}$ and $V_{L,D+\tau}$ are Vandermonde matrices and for any $1 \leq i \leq n$ and D_i is the diagonal matrix whose elements on the diagonal are $y_{i,1}, \dots, y_{i,L}$. 💡

Previous results. We now observe that by (1.6) if

$$L \geq (N + \tau) + (D + \tau) - 1 = N + D + 2\tau - 1 \iff \tau \leq \frac{L - (N + D - 1)}{2} =: \tau'_0 \quad (4.8)$$

then SRFR admits a unique solution.

Note that in the polynomial case, *i.e.* $D = 1$, then $\tau'_0 = \frac{L-N}{2}$ is exactly the error correction capability τ_0 of an n -IRS code with length L and dimension N (see Definition 2.3.1).

Furthermore, we have the following result.

Theorem 4.1.1 ([BK14, Theorem 2.2]). *Consider $L \geq N + D + 2\tau - 1$ evaluation points $\{\alpha_1, \dots, \alpha_L\}$ and \mathcal{M} the $\mathbb{F}_q[x]$ -module of solutions in $\mathcal{S}_{Y,N,D,\tau}$. Then $\text{rank}(\mathcal{M}) = 1$ and given (φ, ψ) a generator of \mathcal{M} with ψ monic then $(\varphi, \psi) = (\Lambda \mathbf{v}, \Lambda d)$.*

Proof. First, notice that $\psi(x)$ is nonzero. Indeed, if $\psi = 0$, then by the key equation (4.6), for any $1 \leq j \leq L$, $\varphi(\alpha_j) = 0$. Now, notice that $\deg(\varphi) \leq N + \tau - 1$ and that the number of roots is $L \geq N + D + 2\tau - 1 > N + \tau - 1$. Hence, since the polynomial has more roots than its degree then $\varphi = \mathbf{0}$. Since (φ, ψ) is a generator of the $\mathbb{F}_q[x]$ -module, it cannot be equal to $(\mathbf{0}, 0)$.

As seen in Section 1.2, if $L \geq N + D + 2\tau - 1$ (see (1.6)), then $\varphi\psi' = \varphi'\psi$ for any $(\varphi', \psi') \in \mathcal{S}_{Y,\tau,N,D}$ and so $\text{rank}(\mathcal{M}) = 1$.

Now,

- since (φ, ψ) is a generator of \mathcal{M} and $(\Lambda \mathbf{v}, \Lambda d) \in \mathcal{S}_{Y,\tau,N,D}$, then there exists $R \in \mathbb{F}_q[x]$ such that $(\Lambda \mathbf{v}, \Lambda d) = (R\varphi, R\psi)$.
- on the other hand, since $\varphi\Lambda d = \psi\Lambda \mathbf{v}$ and both ψ and d are nonzero, then $\frac{\varphi}{\psi} = \frac{\mathbf{v}}{d}$. Moreover, $\gcd(\gcd_i(v_i), d) = 1$ and so there exists $P \in \mathbb{F}_q[x]$ such that $(\varphi, \psi) = (P\mathbf{v}, Pd)$.

Therefore, $\Lambda = PR$ and $P = \Lambda/R$ is of the form $P = \prod_{j \in E'} (x - \alpha_j)$, for $E' \subseteq E$. As in Lemma 2.3.3, we can observe that $|E| = |\{j \mid \Xi_{*,j} \neq 0\}| \leq \deg(P) = |E'|$ and so $E = E'$ and $P \in \mathbb{F}_q$. Since $\varphi = P\mathbf{v}$ and $\psi = Pd$ and ψ and d are both monic, then we can conclude that $P = 1$. \square

Remark 4.1.2. Notice that this lemma basically tells us that if the number of evaluation points is $L \geq N + D + 2\tau - 1$, then the rank of the module \mathcal{M} generated by solutions in $\mathcal{S}_{Y,N,D,\tau}$ is 1 and the solution space is exactly of the form

$$\mathcal{S}_{Y,N,D,\tau} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}} \quad (4.9)$$

where

$$\delta_{N,D,\tau} := \min\{N - \deg(\mathbf{v}), D - \deg(d)\} + (\tau - |E|). \quad (4.10)$$

Indeed, notice that

$$\begin{aligned} \deg(x^i \Lambda \mathbf{v}) &= i + |E| + \deg(\mathbf{v}) \leq N - 1 + \tau \\ \deg(x^i \Lambda d) &= i + |E| + \deg(d) \leq D - 1 + \tau \end{aligned}$$

and so,

$$i \leq \min\{N - \deg(\mathbf{v}), D - \deg(d)\} + (\tau - |E|) - 1 = \delta_{N,D,\tau} - 1.$$

\spadesuit

In [BK14, KPSW17]¹ the authors introduced Algorithm 5 which uniquely² recovers (\mathbf{v}, d) , a solution of SCIwE with instance Y and parameters τ, N, D, q, n and $\{\alpha_1, \dots, \alpha_L\}$. Note that this algorithm is correct: indeed, by Theorem 4.1.1, since $L = N + D + 2\tau - 1$, then (φ, ψ) computed at step 1 coincides with $(\Lambda \mathbf{v}, \Lambda d)$ and so the gcd computed at step 2 is exactly the error locator polynomial Λ .

Algorithm 5: Algorithm for SCIwE of [BK14]

Input : $Y = \left(\frac{\mathbf{v}(\alpha_1)}{d(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{d(\alpha_L)} \right) + \Xi$ an instance of SCIwE with parameters τ, N, D, q and $L := N + D + 2\tau - 1$ evaluation points $\{\alpha_1, \dots, \alpha_L\}$

Output: (\mathbf{v}, d)

- 1 compute $(\varphi_1, \dots, \varphi_n, \psi)$ a generator of the $\mathbb{F}_q[x]$ -module generated by solutions in $\mathcal{S}_{Y, N, D, \tau}$, scaled to obtain ψ monic;
 - 2 $\Lambda \leftarrow \text{gcd}(\varphi_1, \dots, \varphi_n, \psi)$;
 - 3 **return** $\left(\frac{\varphi}{\Lambda}, \frac{\psi}{\Lambda} \right)$
-

New results. In [GLZ19], motivated by the link between SCIwE and the decoding of IRS codes (SIwE, Definition 2.3.2 and Figure 4.3), we proposed³ Algorithm 6 that recovers (\mathbf{v}, d) a solution of our given instance Y , with $L \geq N + D - 1 + \tau + \lceil \frac{\tau}{n} \rceil =: L_{GLZ1}$ evaluation points. Notice that, for $n \geq 1$

$$L_{GLZ1} = N + D - 1 + \tau + \left\lceil \frac{\tau}{n} \right\rceil \leq N + D + 2\tau - 1$$

meaning that we reduce the number of evaluation points needed to reconstruct the solution of SCIwE. However, since we use fewer evaluation points than the number which guarantees to uniquely reconstruct (\mathbf{v}, d) , our algorithm can possibly fail.

Remark 4.1.3. We observe that

$$L \geq L_{GLZ1} = N + D - 1 + \tau + \left\lceil \frac{\tau}{n} \right\rceil \iff \tau \leq \frac{n(L - (N + D - 1))}{n + 1} =: \tau_{GLZ}.$$

Furthermore, since $n \geq 1$, note that

$$\tau'_0 = \frac{L - (N + D - 1)}{2} \leq \frac{n(L - (N + D - 1))}{n + 1} = \tau_{GLZ}$$

1. The authors introduced in [BK14, KPSW17] Algorithm 5 for the PLSwE problem. As previously remarked, PLSwE is a specific case of simultaneous Cauchy interpolation with errors (Definition 4.1.1) in which we want to recover a vector of rational functions which is a solution of a polynomial linear system.

2. The algorithm introduced in [BK14, KPSW17] is slightly different from Algorithm 5: it computes the column echelon form of a basis of the solution space $\mathcal{S}_{Y, N, D, \tau} = \ker(M_{Y, N, D, \tau})$ in order to find the minimal degree solution $(\Lambda \mathbf{v}, \Lambda d)$.

3. We remark that in [GLZ19] we considered a more specific case in which we assumed to know exactly the number of errors and the degree of the denominator. So in this thesis, we present a more general result.

Algorithm 6: A new algorithm for SCIwE

Input : $Y = \left(\frac{v(\alpha_1)}{d(\alpha_1)}, \dots, \frac{v(\alpha_L)}{d(\alpha_L)} \right) + \Xi$ an instance of SCIwE with parameters τ, N, D, q and $L = L_{GLZ1}$ evaluation points $\{\alpha_1, \dots, \alpha_L\}$

Output: (v, d) or “fail”

1 let \mathcal{M} be $\mathbb{F}_q[x]$ -module generated by solutions in $\mathcal{S}_{Y,N,D,\tau}$;

2 **if** $\text{rank}(\mathcal{M}) = 1$ **then**

3 find a generator (φ, ψ) of \mathcal{M} ;

4 $\Lambda \leftarrow \text{gcd}(\varphi, \psi)$;

5 **return** $\left(\frac{\varphi}{\Lambda}, \frac{\psi}{\Lambda} \right)$

6 **else**

7 **return** “fail”

meaning that our algorithm may correct more errors than Algorithm 5. However, as for interleaved Reed-Solomon codes (see Subsection 2.3.1), since we are beyond the number of errors that we can uniquely correct, the uniqueness of the solution recovery is not always guaranteed and the algorithm may fail.

Indeed, if $D = 1$, we remark that $\tau_{GLZ} = \tau_{IRS} = \frac{n(L-N)}{n+1}$ (see (2.22)), *i.e.* the decoding radius of the partial BD decoder (Algorithm 4) for the n -interleaved Reed Solomon code of length L and dimension N . In this sense, Algorithm 6 is a generalization of Algorithm 4 for IRS codes decoding.

💡

Correctness of Algorithm 6. The following result allows us to prove the correctness of our algorithm. It also determines a bound for the *failure probability*, *i.e.* the probability that the algorithm fails.

Theorem 4.1.2. *Let $\tau \geq 0$ and $n, N, D \geq 1$. Consider $L \geq L_{GLZ1}$ evaluation points $\{\alpha_1, \dots, \alpha_L\}$ and $E \subseteq \{1, \dots, L\}$, with $|E| \leq \tau$. Moreover, fix $\mathbf{v} \in \mathbb{F}_q(x)^{n \times 1}$ with $\text{gcd}(\text{gcd}_i(v_i), d) = 1$ such that $\text{deg}(\mathbf{v}) < N$ and $\text{deg}(d) < D$.*

Consider the random matrix $Y = (y_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq L}}$ such that,

- *if $j \in E$, $Y_{*,j}$ is a uniformly distributed element of $\mathbb{F}_q^{n \times 1}$,*
- *if $j \notin E$, $Y_{*,j} = \frac{v(\alpha_j)}{d(\alpha_j)}$.*

Then

$$\mathcal{S}_{Y,N,D,\tau} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}},$$

where $\delta_{N,D,\tau}$ is as in (4.10), with probability at least $1 - \frac{D+\tau}{q}$.

Proof. First notice that since $(\Lambda \mathbf{v}, \Lambda d) \in \mathcal{S}_{Y,\tau,N,D}$ then

$$\langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}} \subseteq \ker(M_{Y,N,D,\tau}) = \mathcal{S}_{Y,\tau,N,D}. \quad (4.11)$$

This proof has the same structure as the proof of Lemma 2.3.1. In detail, in the first part we show the existence of a draw of columns of Y corresponding to the error positions, *i.e.* $Y_{*,j}$ with $j \in E$, for which we have $\mathcal{S}_{Y,N,D,\tau} \subseteq \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}}$. So thanks to (4.11) we can derive the equality.

Consider a partition of E , *i.e.* $E = \cup_{i=1}^n I_i$, such that for any $1 \leq i \leq n$, $|I_i| \leq \lceil |E|/n \rceil$. Note that such a partition exists since $n \lceil |E|/n \rceil \geq |E|$. For any $j \in E$, denote by i_j the unique index such that $j \in I_{i_j}$. Construct a matrix V , such that

- $V_{*,j} = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$, if $j \notin E$,
- if $j \in E$, $V_{*,j} \in \mathbb{F}_q^{n \times 1}$ is chosen so that

$$\mathbf{v}(\alpha_j) - d(\alpha_j)V_{*,j} = \boldsymbol{\epsilon}_{i_j}, \quad (4.12)$$

where $\boldsymbol{\epsilon}_i$ is a vector of $\mathbb{F}_q^{n \times 1}$, whose i -th entry is 1 and all the others are zero.

Now, consider $(\boldsymbol{\varphi}, \boldsymbol{\psi}) \in \mathcal{S}_{V,\tau,N,D}$. By multiplying (4.12) by $\boldsymbol{\psi}(\alpha_j)$ and since $(\boldsymbol{\varphi}, \boldsymbol{\psi}) \in \mathcal{S}_{V,\tau,N,D}$ (see (4.6)) we get

$$\boldsymbol{\psi}(\alpha_j)\mathbf{v}(\alpha_j) - d(\alpha_j)\underbrace{\boldsymbol{\psi}(\alpha_j)V_{*,j}}_{\boldsymbol{\varphi}(\alpha_j)} = \boldsymbol{\psi}(\alpha_j)\boldsymbol{\epsilon}_{i_j}.$$

Fix $1 \leq i \leq n$, we claim that for any $j \notin I_i$ then $\boldsymbol{\psi}(\alpha_j)v_i(\alpha_j) - d(\alpha_j)\varphi_i(\alpha_j) = 0$.

Indeed,

- if $j \notin E$, then $V_{*,j} = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$ and so by replacing $V_{*,j}$ in (4.6), we get in particular $\boldsymbol{\psi}(\alpha_j)v_i(\alpha_j) - d(\alpha_j)\varphi_i(\alpha_j) = 0$,
- if $j \in E \setminus I_i$, by the choice of $V_{*,j}$, then $\boldsymbol{\psi}(\alpha_j)v_i(\alpha_j) - d(\alpha_j)\varphi_i(\alpha_j) = 0$.

So, $\boldsymbol{\psi}(\alpha_j)v_i(\alpha_j) - d(\alpha_j)\varphi_i(\alpha_j) = 0$. Note that $\deg(\boldsymbol{\psi}v_i - d\varphi_i) < N + D + \tau - 1$. On the other hand the number of roots of this polynomial is $L - |I_i| \geq L - \lceil |E|/n \rceil \geq L_{GLZ1} - \tau/n$ and since $L_{GLZ1} \geq N + D - 1 + \tau + \tau/n$ it is then $L - |I_i| \geq N + D + \tau - 1$. Therefore since this polynomial has more roots than its degree it is the zero polynomial. Hence $\boldsymbol{\psi}v - d\boldsymbol{\varphi} = \mathbf{0}$. Now, since $\gcd(\gcd_i(v_i), d) = 1$, there exists $R \in \mathbb{F}_q[x]$ such that $\boldsymbol{\varphi} = R\mathbf{v}$ and $\boldsymbol{\psi} = Rd$. Notice that for any $1 \leq j \leq L$ by (4.6) we get,

$$0 = \boldsymbol{\varphi}(\alpha_j) - \boldsymbol{\psi}(\alpha_j)V_{*,j} = R(\alpha_j)[\mathbf{v}(\alpha_j) - V_{*,j}d(\alpha_j)].$$

By construction, if $j \in E$, then $\mathbf{v}(\alpha_j) - V_{*,j}d(\alpha_j) \neq \mathbf{0}$ and so $R(\alpha_j) = 0$. Therefore, the error locator polynomial $\Lambda = \prod_{j \in E} (x - \alpha_j)$ divides R and so $(\boldsymbol{\varphi}, \boldsymbol{\psi}) \in \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}}$. Hence, $\mathcal{S}_{V,\tau,N,D} \subseteq \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}}$ and so the equality holds.

Therefore, we showed that $\mathcal{S}_{V,\tau,N,D} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}}$ for a draw of Y . Let us show that the equality holds for many draws. Given Y as in the assumption of this theorem, since

$\langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}} \subseteq \mathcal{S}_{Y,\tau,N,D} = \ker(M_{Y,N,D,\tau})$, then $\dim(\ker(M_{Y,N,D,\tau})) \geq \delta_{N,D,\tau}$. By

the Rank-Nullity Theorem

$$\text{rank}(M_{Y,N,D,\tau}) = n(N + \tau) + D + \tau - \dim(\ker(M_{Y,N,D,\tau})) \leq n(N + \tau) + D + \tau - \delta_{N,D,\tau} =: \rho.$$

On the other hand, as proved above, there exists a draw $V_{*,j}$ of $Y_{*,j}$, for $j \in E$, such that $\text{rank}(M_{V,N,D,\tau}) = \rho$. This means that there exists a nonzero ρ -minor in $M_{V,N,D,\tau}$. We consider the same nonzero ρ -minor in $M_{Y,N,D,\tau}$ as a multivariate polynomial C whose indeterminates are $(y_{i,j})_{\substack{1 \leq i \leq n \\ j \in E}}$. We remark that we showed the existence of a draw $V_{*,j}$ of $Y_{*,j}$, for $j \in E$, such that $C(V_{*,j})$ is non zero. Hence the polynomial C is nonzero. For any matrix Y such that $(Y_{*,j})_{j \in E}$ is not a root of C , then $\mathcal{S}_{Y,N,D,\tau} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}}$. Note that the total degree of the polynomial C is at most $D + \tau$, since only the last $D + \tau$ columns of the matrix $M_{Y,N,D,\tau}$ contains the variables $(y_{i,j})_{\substack{1 \leq i \leq n \\ j \in E}}$ (see (4.7)).

Finally by the Schwartz-Zippel Lemma, the polynomial C cannot be zero in more than $(D + \tau)/q$ fractions of its domain. Therefore, we can conclude that the probability that $\mathcal{S}_{Y,N,D,\tau} \neq \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}}$ is at most $(D + \tau)/q$. \square

Remark 4.1.4. We now recall that by Remark 4.1.2 we have that

$$\text{rank}(\mathcal{M}) = 1 \iff \mathcal{S}_{Y,\tau,N,D} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}}$$

Note that \Leftarrow is trivial, while the other implication derives from the proof of Theorem 4.1.1. This allows us to prove the correctness of our Algorithm 6. Indeed, if $\text{rank}(\mathcal{M}) = 1$ once found a generator (φ, ψ) of \mathcal{M} , by computing $\text{gcd}(\varphi, \psi)$ we then recover the error locator polynomial Λ , and by dividing (φ, ψ) by Λ we finally reconstruct the solution (\mathbf{v}, d) . \heartsuit

4.1.2 Polynomial Linear Solving with Errors

We shortly recall the main aim of this chapter: constructing an ABFT technique for PLS solving by evaluation interpolation.

We now consider a PLS $A(x)\mathbf{y}(x) = \mathbf{b}(x)$, where $\mathbf{y}(x) = \frac{\mathbf{v}(x)}{d(x)}$, $\text{gcd}(\text{gcd}_i(v_i), d) = 1$ and such that d is monic. Fix L evaluation points $\{\alpha_1, \dots, \alpha_L\}$. We assume that both the evaluation and the pointwise resolution of the evaluated systems steps (of the evaluation-interpolation algorithm for PLS solving) are parallelized. Therefore, our nodes compute

$$Y = \left(\frac{\mathbf{v}(\alpha_1)}{d(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{d(\alpha_L)} \right) + \Xi \quad (4.13)$$

with $E := \{j \mid \Xi_{*,j} \neq \mathbf{0}\}$.

The PLSwE is then the problem of recovering (\mathbf{v}, d) , given

- the matrix Y as in (4.13),
- the evaluation points $\{\alpha_1, \dots, \alpha_L\}$,

- the degree bounds $N > \deg(\mathbf{v})$, $D > \deg(d)$ and $\deg(A)$ and $\deg(\mathbf{b})$,
- an upper bound τ on the number of errors $|E|$ occurred at the parallelization step.

Notice that this is an application of the SCIwE problem introduced so far. Indeed, here we want to recover a vector of rational functions, which is a *solution of a PLS*, given its evaluations and some bounds on the degrees. For this reason we can add the degrees of A and b as additional inputs.

So, we can use the same technique for solving the SCIwE problem to solve PLSwE, *i.e.* the simultaneous polynomial reconstruction (4.6), to recover $(\Lambda \mathbf{v}, \Lambda d)$, where $\Lambda = \prod_{j \in E} (x - \alpha_j)$ is the error locator polynomial.

Previous results. In [KPSW17], E. Kaltofen *et al.* proved the following theorem.

Theorem 4.1.3. [KPSW17] *Consider $L \geq \min\{N + D - 1, \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}\} + 2\tau$ evaluation points $\{\alpha_1, \dots, \alpha_L\}$ and \mathcal{M} the $\mathbb{F}_q[x]$ -module of solutions in $\mathcal{S}_{Y,N,D,\tau}$. Then $\text{rank}(\mathcal{M}) = 1$ and given (φ, ψ) a generator of \mathcal{M} with ψ monic then $(\varphi, \psi) = (\Lambda \mathbf{v}, \Lambda d)$.*

Proof. If $N + D - 1 \leq \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}$ the claim follows by Theorem 4.1.1.

On the other hand, if $\max\{\deg(A) + N, \deg(\mathbf{b}) + D\} \leq N + D - 1$ denote $L_{KPSW} := \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + 2\tau$ and consider $(\varphi', \psi') \in \mathcal{S}_{Y,N,D,\tau}$. Then, by (4.6), for any $1 \leq j \leq L$,

$$\varphi'(\alpha_j) = \mathbf{y}_j \psi'(\alpha_j).$$

Now observe that for $j \notin E$, since $\mathbf{y}_j = A(\alpha_j)^{-1} \mathbf{b}(\alpha_j)$ then,

$$A(\alpha_j) \varphi'_j(\alpha_j) = \mathbf{b}(\alpha_j) \psi'(\alpha_j).$$

The vector of polynomials $A(x) \varphi'(x) - \mathbf{b}(x) \psi'(x)$ has degree $\deg(A(x) \varphi'(x) - \mathbf{b}(x) \psi'(x)) < \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + \tau$ and $L - |E| \geq L_{KPSW} - \tau \geq \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + \tau$ roots. Therefore,

$$A(x) \varphi'(x) - \mathbf{b}(x) \psi'(x) = 0. \quad (4.14)$$

We now consider (\mathbf{v}, d) , the solution of our given PLS. Then,

$$A(x) \mathbf{v}(x) = \mathbf{b}(x) d(x). \quad (4.15)$$

If we multiply the equation (4.14) by $d(x)$ and the equation (4.15) by $\psi(x)$ and we subtract them we get

$$A(x) [\varphi'(x) d(x) - \mathbf{v}(x) \psi'(x)] = 0$$

and since $A(x)$ is full rank, then $\varphi'(x) d(x) - \mathbf{v}(x) \psi'(x) = 0$. Therefore, the $\text{rank}(\mathcal{M}) = 1$ and the proof follows from the proof of Theorem 4.1.1. \square

Therefore, this means that if we consider $L := \min\{N + D - 1, \max\{N + \deg(A), D + \deg(\mathbf{b})\}\} + 2\tau$ evaluation points we can uniquely recover the solution (\mathbf{v}, d) . So we can adapt Algorithm 5 with this new number of evaluation points to get an algorithm for PLSwE. Indeed, notice that once again we have reduced the uniqueness of the recovering of our solution to the uniqueness of the solution of SRFR.

New results. In [GLZ19] we introduced an algorithm for SCIwE (Algorithm 6) which reduces the number of evaluation points *w.r.t* Theorem 4.1.1. As seen before, this algorithm is obtained by generalizing the interpolation-based decoding technique for interleaved Reed-Solomon codes (Subsection 2.3.1).

We now introduce a new result which allows us to construct an algorithm for PLSwE with a smaller number of evaluation points *w.r.t* Theorem 4.1.3 [GLZ20a].

Theorem 4.1.4. *Let $\tau \geq 0$ and $n, N, D \geq 1$. Consider*

$$L \geq \min\{N + D - 1, \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}\} + \tau + \left\lceil \frac{\tau}{n} \right\rceil =: L_{GLZ2}$$

evaluation points $\{\alpha_1, \dots, \alpha_L\}$ and $E \subseteq \{1, \dots, L\}$, with $|E| \leq \tau$. Moreover, fix a PLS $A(x)\mathbf{y}(x) = \mathbf{b}(x)$ and denote by $\frac{\mathbf{v}(x)}{d(x)}$ its solution, with $\gcd(\gcd_i(v_i), d) = 1$ and d monic. Let $\deg(\mathbf{v}) < N$ and $\deg(d) < D$.

Consider the random matrix Y such that,

- *if $j \in E$, $Y_{*,j}$ is a uniformly distributed element of $\mathbb{F}_q^{n \times 1}$,*
- *if $j \notin E$, $Y_{*,j} = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$,*

then the solution space of SRFR (4.6) is of the form

$$\mathcal{S}_{Y,N,D,\tau} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}},$$

where $\delta_{N,D,\tau}$ is as in (4.10), with probability at least $1 - \frac{D+\tau}{q}$.

Proof. Note that if $N + D - 1 \leq \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}$, the proof follows from Theorem 4.1.2. On the other hand, assume that $\max\{\deg(A) + N, \deg(\mathbf{b}) + D\} \leq N + D - 1$ and so consider $L \geq \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + \tau + \left\lceil \frac{\tau}{n} \right\rceil$ evaluation points.

This proof is similar to the proof of Theorem 4.1.2. In the first part we prove the existence of a draw of columns of Y corresponding to the error positions for which we have $\mathcal{S}_{Y,\tau,IRS,k} = \langle x^i \Lambda \mathbf{f}, x^i \Lambda \rangle_{0 \leq i < \delta_{N,D,\tau}}$.

Indeed, consider a partition of E , *i.e.* $E = \cup_{i=1}^n I_i$, such that for any $1 \leq i \leq n$, $|I_i| \leq \lceil |E|/n \rceil$. Note that such a partition exists since $n \lceil |E|/n \rceil \geq |E|$. For any $j \in E$, denote by i_j the unique index such that $j \in I_{i_j}$.

Construct a matrix V , such that

- *if $j \notin E$, then $V_{*,j} = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$,*

— if $j \in E$, then $V_{*,j}$ is chosen so that

$$\mathbf{v}(\alpha_j) - d(\alpha_j)V_{*,j} = -A(\alpha_j)^{-1}d(\alpha_j)\boldsymbol{\epsilon}_{i_j} \quad (4.16)$$

where $\boldsymbol{\epsilon}_i$ is a vector of $\mathbb{F}_q^{n \times 1}$, whose i -th entry is 1 and all the others are zero.

Now consider $(\boldsymbol{\varphi}, \psi) \in \mathcal{S}_{V,\tau,N,D}$ and since that by our assumptions $d(\alpha_j) \neq 0$, then

$$\begin{aligned} \mathbf{v}(\alpha_j) - d(\alpha_j)V_{*,j} &= -A(\alpha_j)^{-1}d(\alpha_j)\boldsymbol{\epsilon}_{i_j} && \iff \\ A(\alpha_j)d(\alpha_j)V_{*,j} - A(\alpha_j)\mathbf{v}(\alpha_j) &= d(\alpha_j)\boldsymbol{\epsilon}_{i_j} && \iff \\ A(\alpha_j)V_{*,j} - \underbrace{A(\alpha_j)\frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}}_{\mathbf{b}(\alpha_j)} &= \boldsymbol{\epsilon}_{i_j} \end{aligned} \quad (4.17)$$

for all $j \in E$. By multiplying (4.17) by $\psi(\alpha_j)$ we get

$$A(\alpha_j)V_{*,j}\psi(\alpha_j) - \mathbf{b}(\alpha_j)\psi(\alpha_j) = \psi(\alpha_j)\boldsymbol{\epsilon}_{i_j}$$

and since $(\boldsymbol{\varphi}, \psi) \in \mathcal{S}_{V,\tau,N,D}$, it satisfies $\boldsymbol{\varphi}(\alpha_j) = V_{*,j}\psi(\alpha_j)$ and so we have

$$A(\alpha_j)\boldsymbol{\varphi}(\alpha_j) - \mathbf{b}(\alpha_j)\psi(\alpha_j) = \psi(\alpha_j)\boldsymbol{\epsilon}_{i_j}.$$

We now denote

$$\mathbf{p} := A(x)\boldsymbol{\varphi}(x) - \psi(x)\mathbf{b}(x) \in \mathbb{F}_q[x]^{n \times 1}.$$

Fix $1 \leq i \leq n$, we claim that for any $j \notin I_i$ then $p_i(\alpha_j) = 0$, where p_i is the i -th component of \mathbf{p} .

Indeed,

- if $j \notin E$, then $V_{*,j} = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)} = A(\alpha_j)^{-1}\mathbf{b}(\alpha_j)$ and so since $\boldsymbol{\varphi}(\alpha_j) = V_{*,j}\psi(\alpha_j)$, we get $\mathbf{p}(\alpha_j) = A(\alpha_j)\boldsymbol{\varphi}(\alpha_j) - \psi(\alpha_j)\mathbf{b}(\alpha_j) = \mathbf{0}$. In particular $p_i(\alpha_j) = 0$.
- If $j \in E \setminus I_i$, by the choice of $V_{*,j}$, then $p_i(\alpha_j) = 0$.

Therefore $p_i(\alpha_j) = 0$. Note that $\deg(p_i(x)) < \max\{\deg(A) + N, \deg(\mathbf{b}) + D\}$. On the other hand the roots of this polynomial are

$$L - |I_i| \geq L - \lceil |E|/n \rceil \geq L_{GLZ2} - \tau/n = \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + \tau.$$

So we can conclude that $\mathbf{p}(x) = A(x)\boldsymbol{\varphi}(x) - \psi(x)\mathbf{b}(x) = \mathbf{0}$. Therefore, as in the proof of Theorem 4.1.3 we have that $\boldsymbol{\varphi}(x)d(x) - \psi(x)\mathbf{v}(x) = \mathbf{0}$. The rest of the proof coincides exactly with the proof of Theorem 4.1.2. \square

Therefore, we conclude by introducing Algorithm 7 for PLSwE, obtained by slightly modifying Algorithm 6 for the general SCIwE problem. The correctness of this algorithm can be proved in the same way as Remark 4.1.4.

Algorithm 7: A new algorithm for PLSwE

Input : $Y = \left(\frac{v(\alpha_1)}{d(\alpha_1)}, \dots, \frac{v(\alpha_L)}{d(\alpha_L)} \right) + \Xi$,
 $\tau \geq |E| = |\{j \mid \Xi_{*,j} \neq \mathbf{0}\}|$, $N > \deg(\mathbf{v})$, $D > \deg(d)$, $\deg(A)$, $\deg(\mathbf{b})$
 and
 $L = L_{GLZ2} = \min\{N + D - 1, \max\{N + \deg(A), D + \deg(\mathbf{b})\}\} + \tau + \lceil \tau/n \rceil$
 evaluation points $\{\alpha_1, \dots, \alpha_L\}$

Output: (\mathbf{v}, d) solution of (4.1) or “fail”.

- 1 Let \mathcal{M} be the $\mathbb{F}_q[x]$ -module generated by solutions in $\mathcal{S}_{Y,N,D,\tau}$;
- 2 **if** $\text{rank}(\mathcal{M}) = 1$ **then**
- 3 find (φ, ψ) a generator of \mathcal{M} ;
- 4 $\Lambda \leftarrow \text{gcd}(\varphi, \psi)$;
- 5 **return** $\left(\frac{\varphi}{\Lambda}, \frac{\psi}{\Lambda} \right)$
- 6 **else**
- 7 **return** “fail”

4.2 Early termination techniques

In the previous section, we introduced a method for solving the PLSwE problem which generalizes the interpolation-based decoding technique of interleaved Reed-Solomon codes. This technique basically reduces the problem to SRFR, and once again the goal is to determine the number of evaluation points needed for the uniqueness of SRFR solution.

We now briefly recall previous results. In [KPSW17], E. Kalfoten *et al.* proved that with

$$\bar{L} = \min\{L_{BK}, L_{KPSW}\} \quad (4.18)$$

evaluation points, where

- $L_{BK} := N + D - 1 + 2\tau$,
- $L_{KPSW} := \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + 2\tau$,

the corresponding SRFR admits a unique solution and so we can uniquely recover the solution of a PLS.

By Theorem 4.1.2 and Theorem 4.1.4, we can reduce this number to

$$\tilde{L} = \min\{L_{GLZ1}, L_{GLZ2}\}, \quad (4.19)$$

where

- $L_{GLZ1} := N + D - 1 + \tau + \lceil \frac{\tau}{n} \rceil$,
- $L_{GLZ2} := \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + \tau + \lceil \frac{\tau}{n} \rceil$.

However, since we are below the number of evaluation points which guarantees the uniqueness of SRFR (see Remark 4.1.3), Algorithm 7 can fail for at most $\frac{D+\tau}{q}$ fractions of possible errors.

Note that both \bar{L} and \tilde{L} strongly depends on the bounds N and D on the degrees of the solution that we want to recover and of the bound τ of the number on errors which occur in the parallelization step. Therefore, if we consider N, D much bigger than the real degrees of the solution and τ much bigger then the real number of errors, we can significantly increase the number of evaluation points compared to the number we really need.

We now provide an example, to better visualize the problem.

Example 4.2.1. Let \mathbb{F}_{37} and consider the PLS whose coefficient matrix is

$$A = \begin{pmatrix} 35x^3 + 4x^2 + 35x + 8 & 29x^5 + 35x^4 + 35x^3 + 5x + 9 \\ 12x^5 + 3x^4 + 3x^3 + 11x + 5 & 11x^7 + 9x^6 + 28x^4 + 27x^3 + 18x^2 + 36x + 10 \end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} 8x + 1 \\ 2x^2 + 5 \end{pmatrix}$$

Let

$$\frac{\mathbf{v}}{d} = \begin{pmatrix} \frac{10x^5 + 22x^4 + x^3 + 20x + 34}{x^5 + 10x^4 + 4x^3 + 28x^2 + 16x + 3} \\ \frac{16x^3 + 14x^2 + 9x + 3}{x^5 + 10x^4 + 4x^3 + 28x^2 + 16x + 3} \end{pmatrix}$$

be the solution with d monic that we want to recover. We do not know a priori the real degrees of \mathbf{v} and d , *i.e.* $\deg(\mathbf{v}) = 5$ and $\deg(d) = 5$. The same holds for the number of errors $|E|$ introduced by nodes. Assume for instance that it is $|E| = 2$.

Recall that by the Cramer's Rule (see Lemma 1.4.1), we can take as bounds for the degrees of \mathbf{v} and d

$$\begin{aligned} \deg(\mathbf{v}) &< N = (n - 1) \deg(A) + \deg(\mathbf{b}) + 1 = 10 \\ \deg(d) &< D = n \deg(A) + 1 = 15 \end{aligned}$$

and also consider $\tau = 4$. Then,


$$\begin{aligned} \bar{L} &= \min\{L_{BK}, L_{KPSW}\} = \min\{32, 25\} = 25 \\ \tilde{L} &= \min\{L_{GLZ1}, L_{GLZ2}\} = \min\{30, 23\} = 23. \end{aligned}$$

Besides, the real number of points that we need is

$$\begin{aligned} \bar{\mathcal{L}}_{\text{ideal}} &= \min\{\mathcal{L}_{BK}, \mathcal{L}_{KPSW}\} = 15 \\ \tilde{\mathcal{L}}_{\text{ideal}} &= \min\{\mathcal{L}_{GLZ1}, \mathcal{L}_{GLZ2}\} = 14 \end{aligned} \tag{4.20}$$

where

- $\mathcal{L}_{BK} := \deg(\mathbf{v}) + \deg(d) + 2|E| + 1$,
- $\mathcal{L}_{KPSW} := \max\{\deg(A) + \deg(\mathbf{v}), \deg(\mathbf{b}) + \deg(d)\} + 2|E| + 1$,
- $\mathcal{L}_{GLZ1} := \deg(\mathbf{v}) + \deg(d) + |E| + \lceil |E|/n \rceil + 1$,
- $\mathcal{L}_{KPSW} := \max\{\deg(A) + \deg(\mathbf{v}), \deg(\mathbf{b}) + \deg(d)\} + |E| + \lceil |E|/n \rceil + 1$.

Notice that we replace N, D, τ by $\deg(\mathbf{v}) + 1, \deg(d) + 1$ and $|E|$ respectively. 

Therefore, the discrepancy between bounds N, D and the real degrees $\deg(\mathbf{v}), \deg(d)$ and between the bound τ and the real number $|E|$ of errors implies an overestimation on the number of evaluation points needed for the computations.

A classical strategy to overcome this limit consists in performing an *early termination technique* [KPSW17] which is an *adaptive* strategy which, starting from a *minimal value* of evaluation points, iteratively increments this number until a nontrivial result is found. Note that this means that the *minimal number of evaluations* which guarantees the existence is reached and it is really important in this setting to *determine* this minimal number.

We point out that the main goal of this early termination technique is to possibly decrease the number of evaluation points needed for the reconstruction and not to decrease the complexity of the algorithm used for the resolution.

In the following paragraph we reinterpret and revisit the results of [KPSW17].

4.2.1 Previous results

In this subsection we consider a PLS $A(x)\mathbf{y}(x) = \mathbf{b}(x)$ and its solution $\frac{\mathbf{v}(x)}{d(x)}$, where $\gcd(\gcd_i(v_i), d) = 1$ and d is monic. Given $L \leq q$ evaluation points $\{\alpha_1, \dots, \alpha_L\}$, we receive the matrix

$$Y = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)} + \Xi, \text{ for any } 1 \leq j \leq L$$

computed at the parallelization step, where the error support is $E = \{j \mid \Xi_{*,j} \neq \mathbf{0}\}$. For any $1 \leq j \leq L$, we also denote $\mathbf{y}_j := Y_{*,j}$ and we consider the bounds $N > \deg(\mathbf{v}), D > \deg(d)$ and $\tau \geq |E|$.

In [KPSW17], E. Kaltofen *et al.* introduced Algorithm 8 which takes as inputs the additional parameters $\nu, \vartheta, \xi \geq 0$ allowing one to reduce the number of evaluation points needed for computations. Specifically, the main idea of the algorithm consists in the study of the solution space $\mathcal{S}_{Y, \nu, \vartheta, \xi}$ of the SRFR related to Y and with degree constraints determined by these new parameters $\nu + \xi$ and $\vartheta + \xi$. The algorithm declares a failure if these bounds are too small compared to the real degrees of the solution $(\Lambda \mathbf{v}, \Lambda d)$ that we are searching for, or in other terms $\deg(\mathbf{v}) + |E| > \nu + \xi$ or $\deg(d) + |E| > \vartheta + \xi$.

The correctness of this algorithm is based on the following theorem.

Theorem 4.2.1. *Let $\nu, \vartheta, \xi \geq 0$ and*

$$L \geq \min\{\mathfrak{L}_{BK}, \mathfrak{L}_{KPSW}\} =: \bar{\mathfrak{L}}$$

where

- $\mathfrak{L}_{BK} := \max\{\nu + \deg(d), \vartheta + \deg(\mathbf{v})\} + \xi + |E|$,
- $\mathfrak{L}_{KPSW} := \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\} + \xi + |E|$.

Algorithm 8: Algorithm which computes $\frac{v}{d}$ or determine if the degree bounds are too small [KPSW17]

Input : $Y = \left(\frac{v(\alpha_1)}{d(\alpha_1)}, \dots, \frac{v(\alpha_L)}{d(\alpha_L)} \right) + \Xi$,
 $\tau \geq |E| = |\{j \mid \Xi_{*,j} \neq \mathbf{0}\}|$, $N > \deg(\mathbf{v})$, $D > \deg(d)$, $\deg(A)$, $\deg(\mathbf{b})$,
 $\nu, \vartheta, \xi \geq 0$,
 $L = \min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \tau + \xi$,
the evaluation points $\{\alpha_1, \dots, \alpha_L\}$.

Output: (\mathbf{v}, d) or “the bounds ν, ϑ, ξ are too small”

1 Let \mathcal{M} be the $\mathbb{F}_q[x]$ -module generated by solutions in $\mathcal{S}_{Y,\nu,\vartheta,\xi}$;
2 **if** $\text{rank}(\mathcal{M}) = 0$ **then**
3 **return** “the bounds ν, ϑ, ξ are too small”;
4 **else**
5 find a generator (φ, ψ) of \mathcal{M} ;
6 $\Lambda \leftarrow \text{gcd}(\varphi, \psi)$;
7 **return** $\left(\frac{\varphi}{\Lambda}, \frac{\psi}{\Lambda} \right)$

Then

$$\mathcal{S}_{Y,\nu,\vartheta,\xi} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu,\vartheta,\xi}}$$

where

$$\delta_{\nu,\vartheta,\xi} := \min\{\nu - \deg(\mathbf{v}), \vartheta - \deg(d)\} + \xi - |E|.$$

By convention we set $\langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu,\vartheta,\xi}} = \{(\mathbf{0}, 0)\}$ if $\delta_{\nu,\vartheta,\xi} \leq 0$.

Before presenting the proof of this theorem we briefly explain the meaning of this result.

Remark 4.2.1. This result basically tells us that with $L \geq \bar{\mathfrak{L}}$ evaluation points, the solution space $\mathcal{S}_{Y,\nu,\vartheta,\xi}$ determined by solutions (φ, ψ) of the equation

$$\varphi(\alpha_j) = \mathbf{y}_j \psi(\alpha_j), \quad \deg(\varphi) < \nu + \xi, \quad \deg(\psi) < \vartheta + \xi$$

for any $1 \leq j \leq L$ (or equivalently of SRF in the interpolation form with inputs $Y, \nu + \xi, \vartheta + \xi$), is spanned by elements of the form $\langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle$. And so, if we consider ν, ϑ, ξ too small then $\mathcal{S}_{Y,\nu,\vartheta,\xi}$ is trivial. Formally, we observe that given $\nu, \vartheta, \xi \geq 0$,

$$\begin{cases} \deg(\mathbf{v}) + |E| < \nu + \xi \\ \deg(d) + |E| < \vartheta + \xi \end{cases} \iff \delta_{\nu,\vartheta,\xi} > 0 \iff \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu,\vartheta,\xi}} \neq \{(\mathbf{0}, 0)\}.$$

By Theorem 4.2.1,

$$\text{if } L \geq \bar{\mathfrak{L}} \text{ then } \mathcal{S}_{Y,\nu,\vartheta,\xi} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu,\vartheta,\xi}}$$

and so, we can conclude that if $L \geq \overline{\mathfrak{L}}$,

$$\delta_{\nu, \vartheta, \xi} > 0 \iff \mathcal{S}_{Y, \nu, \vartheta, \xi} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu, \vartheta, \xi}} \neq \{(\mathbf{0}, 0)\} \quad (4.21)$$

💡

Proof of Theorem 4.2.1. The outlines of this proof are the same as the proofs of Theorem 4.1.1 and Theorem 4.1.3. Besides, we remark that here we are considering ν, ϑ, ξ some *general* non-negative parameters while in Theorem 4.1.1 and Theorem 4.1.3 we considered the bounds N, D, τ on the degrees of \mathbf{v}, d and the number of errors $|E|$. Hence the solution space $\mathcal{S}_{Y, \nu, \vartheta, \xi}$ does not depend on the bounds N, D, τ , but it depends the parameters ν, ϑ, ξ . These parameters may be too small, meaning that $\mathcal{S}_{Y, \nu, \vartheta, \xi}$ could be trivial, since it may not contain the solution $(\Lambda \mathbf{v}, \Lambda d)$ that we are searching for.

So, the only difference compared to the proofs of the Theorem 4.1.1 and Theorem 4.1.3 we have to pay attention to, is how we prove that for any $(\varphi, \psi) \in \mathcal{S}_{Y, \nu, \vartheta, \xi}$

- $\varphi(x)\Lambda(x)d(x) - \psi(x)\Lambda(x)\mathbf{v}(x) = \mathbf{0}$, if $\mathfrak{L}_{BK} \leq \mathfrak{L}_{KPSW}$,
- $A(x)\varphi(x) - \mathbf{b}(x)\psi(x) = \mathbf{0}$, if $\mathfrak{L}_{KPSW} \leq \mathfrak{L}_{BK}$.

since it is here that we use the parameters ν, ϑ, ξ .

First, we assume that $\mathfrak{L}_{BK} \leq \mathfrak{L}_{KPSW}$. We consider $(\varphi, \psi) \in \mathcal{S}_{Y, \nu, \vartheta, \xi}$ and since $(\Lambda \mathbf{v}, \Lambda d) \in \mathcal{S}_{Y, \nu, \vartheta, \xi}$ we have that for any $1 \leq j \leq L$

$$\begin{aligned} \varphi(\alpha_j) &= \mathbf{y}_j \psi(\alpha_j) \\ \Lambda(\alpha_j) \mathbf{v}(\alpha_j) &= \mathbf{y}_j \Lambda(\alpha_j) d(\alpha_j) \end{aligned}$$

Fix $1 \leq j \leq L$ and by multiplying the former equation by $\Lambda(\alpha_j) d(\alpha_j)$ and the latter by $\psi(\alpha_j)$ and by subtracting them we finally get

$$\varphi(\alpha_j) \Lambda(\alpha_j) d(\alpha_j) - \psi(\alpha_j) \Lambda(\alpha_j) \mathbf{v}(\alpha_j) = 0.$$

Now we observe that $\deg(\varphi(x)\Lambda(x)d(x) - \psi(x)\Lambda(x)\mathbf{v}(x)) \leq \max\{\nu + \deg(d), \vartheta + \deg(\mathbf{v})\} + \xi + |E| - 1$ and so, since the number of evaluation points is $L \geq \mathfrak{L}_{BK} = \max\{\nu + \deg(d), \vartheta + \deg(\mathbf{v})\} + \xi + |E|$, we can conclude that

$$\varphi(x)\Lambda(x)d(x) - \psi(x)\Lambda(x)\mathbf{v}(x) = \mathbf{0}.$$

The rest of the proof is identical to the proof of Theorem 4.1.1. First we observe that $\psi(x) \neq 0$ and so by the previous equation we have that $\frac{\varphi(x)}{\psi(x)} = \frac{\mathbf{v}(x)}{d(x)}$. Then, the claim follows by remarking that the vector of rational functions $\frac{\mathbf{v}(x)}{d(x)}$ is $\gcd(\gcd_i(v_i), d) = 1$.

On the other hand, we now assume that $\mathfrak{L}_{KPSW} \leq \mathfrak{L}_{BK}$. In this case, we consider

$(\varphi, \psi) \in \mathcal{S}_{Y, \nu, \vartheta, \xi}$. Then, since for $j \notin E$ we have that $\mathbf{y}_j = A(\alpha_j)^{-1} \mathbf{b}(\alpha_j)$, then

$$A(\alpha_j) \varphi(\alpha_j) - \mathbf{b}(\alpha_j) \psi(\alpha_j) = \mathbf{0}$$

Now, observe that $\deg(A(x)\varphi(x) - \mathbf{b}(x)\psi(x)) \leq \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\} + \xi - 1$ and that the roots of this vector of polynomials are $L - |E| \geq \mathfrak{L}_{KPSW} - |E| = \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\} + \xi$ then

$$A(x)\varphi(x) - \mathbf{b}(x)\psi(x) = \mathbf{0}.$$

So, since $(\Lambda \mathbf{v}, \Lambda d) \in \mathcal{S}_{Y, \nu, \vartheta, \xi}$ then we also have that $A(x)\Lambda(x)\mathbf{v}(x) - \mathbf{b}(x)\Lambda(x)d(x) = \mathbf{0}$ and the rest of the proof coincides exactly with the proof of Theorem 4.1.3. \square

Correctness of Algorithm 8. Notice that, since by assumption $N - 1 \geq \deg(\mathbf{v})$, $D - 1 \geq \deg(d)$ and $\tau \geq |E|$ then

$$\begin{aligned} L &= \min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \tau + \xi \\ &\geq \min\{\max\{\deg(\mathbf{v}) + \vartheta, \deg(d) + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + |E| + \xi = \overline{\mathfrak{L}} \end{aligned}$$

and so by Theorem 4.2.1 and the Remark 4.2.1 (see (4.21)) we can deduce the correctness of Algorithm 8. Indeed, since $L \geq \overline{\mathfrak{L}}$,

- if $\mathcal{S}_{Y, \nu, \vartheta, \xi} = \ker(M_{Y, \nu, \vartheta, \xi})$ is trivial, then $\delta_{\nu, \vartheta, \xi} \leq 0$ and so $\deg(\mathbf{v}) + |E| \geq \nu + \xi$ or $\deg(d) + |E| \geq \vartheta + \xi$. In this case, the new parameters ν, ϑ, ξ are too small and so the algorithm outputs the failure message “the bounds ν, ϑ, ξ are too small”.
- otherwise, $\delta_{\nu, \vartheta, \xi} > 0$ and Algorithm 5 returns (\mathbf{v}, d) .

Based on these results, we can construct Algorithm 9 (which is a *revisited version*⁴ of Algorithm 2.2 of [KPSW17]) which dynamically increase the number of evaluation points and the parameters ν, δ and ξ , until it reaches the *minimal number of evaluation points* which allows us to find a solution

$$\overline{\mathfrak{L}}_{\text{ET}} = \min\{\mathcal{L}_{BK}, \mathcal{L}_{KPSW}\} \quad (4.22)$$

where

- $\mathcal{L}_{BK} = \max\{N + \deg(d), D + \deg(\mathbf{v})\} + \tau + |E|$,
- $\mathcal{L}_{KPSW} = \max\{\deg(A) + \deg(\mathbf{v}) + 1, \deg(\mathbf{b}) + \deg(d) + 1\} + \tau + |E|$.

4. In [KPSW17] it was introduced an early termination algorithm similar to Algorithm 9. The only difference consists in how the bound on the number of errors is handled. Indeed, in that article, authors estimated the error rate of the nodes and derived a bound on the number of errors which is close to the real one. Here, we introduce the parameter ξ , which, informally speaking, play the same role of the other two parameters ν, ϑ .

Algorithm 9: Early termination algorithm from [KPSW17]

Input : a stream of vectors (\mathbf{y}_j) , for $j = 1, \dots$, which is extensible in length on demand, where $\mathbf{y}_j = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)} + \mathbf{e}_j$
 τ an upper bound on the number of errors, $N > \deg(\mathbf{v})$, $D > \deg(d)$
 $\deg(A)$, $\deg(\mathbf{b})$

Output: (\mathbf{v}, d)

```

1  $\nu \leftarrow 0$ ;
2  $\vartheta \leftarrow 0$ ;
3  $\xi \leftarrow 0$ ;
4  $L \leftarrow \min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \tau + \xi$ ;
5 if Algorithm 8( $Y, \tau, \nu, \vartheta, L, \{\alpha_1, \dots, \alpha_L\}$ ) =  $(\mathbf{v}, d)$  then
6    $\lfloor$  return  $(\mathbf{v}, d)$ 
7 while true do
8    $L \leftarrow L + 1$ ;
9   require a new  $\mathbf{y}_L$ ;
10  foreach  $(\nu, \vartheta, \xi)$  with
       $\min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \tau + \xi = L$  do
11    if Algorithm 8( $Y, \tau, \nu, \vartheta, L, \{\alpha_1, \dots, \alpha_L\}$ ) =  $(\mathbf{v}, d)$  then
12       $\lfloor$  return  $(\mathbf{v}, d)$ 

```

Notice that we suppose that the of Algorithm 9 takes as input a stream of vectors (\mathbf{y}_j) which is extensible in length⁵. In other terms, we assume that any time that the number of evaluation points is incremented we can ask to a new node of the parallelization step to compute a new vector $\mathbf{y}_j = \mathbf{v}(\alpha_j)/d(\alpha_j) + \mathbf{e}_j$ related to a new evaluation point α_j .

We now illustrate this algorithm with an example.

Example 4.2.2. Let us consider the PLS of Example 4.2.1. Recall that $\deg(\mathbf{v}) = 5$, $\deg(d) = 5$, the number of errors is $|E| = 2$, $\deg(A) = 7$ and $\deg(\mathbf{b}) = 2$.

We take as degree bounds $N = 10$, $D = 15$ and $\tau = 4$. At the beginning $\nu = 0$, $\vartheta = 0$, $\xi = 0$ and so the initial number of evaluation points is

$$L = \min\{\max\{N + \vartheta - 1, D + \nu - 1\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \tau + \xi = 11.$$

We first notice that

$$\mathcal{S}_{Y, \nu, \vartheta, \tau} = \{\mathbf{0}, 0\}$$

5. Notice that in this case, since we consider as input a stream of vectors variable in length, the number of errors can vary. For this reason, we can take τ as a bound related to the bigger number of points $\bar{L} \geq \bar{\mathcal{L}}_{\text{ET}}$ (recall (4.18) and (4.22)). Or, following the same strategy of [KPSW17] we may derive a bound on the number of errors by estimating the error rate of the nodes. This is still a work in progress.

since

$$\begin{aligned} \deg(\Lambda \mathbf{v}) = 7 > \nu + \xi = 0 \\ \deg(\Lambda d) = 7 > \nu + \xi = 0 \end{aligned} \iff \delta_{\nu, \vartheta, \xi} < 0$$

and so, $\ker(M_{Y, \nu, \vartheta, \xi})$ is trivial and the number of points is incremented. We now observe that the number of evaluation points at which the algorithm is supposed to stop is

$$\overline{\mathcal{L}}_{\text{ET}} = \min\{\max\{N + \deg(d), D + \deg(\mathbf{v})\}, \max\{\deg(A) + \deg(\mathbf{v}) + 1, \deg(\mathbf{b}) + \deg(d) + 1\}\} + \tau + |E| = 19,$$

and we study the possible values of ν, ϑ, ξ for the number of evaluation points of the last two iterations of the algorithm

number of evaluation points	ν	ϑ	ξ
$L = 18$	7	0...12	0
	6	0...11	1
	5	0...10	2
	4	0...9	3
	3	0...8	4
	2	0...7	5
	1	0...6	6
	0	0...5	7
$L = 19$	8	0...13	0
	7	0...12	1
	6	0...11	2
	5	0...10	3
	4	0...9	4
	3	0...8	5
	2	0...7	6
	1	0...6	7
	0	0...5	8

Notice that if $L = 18$, then we have that $\nu + \xi \leq 7$ and $\vartheta + \xi \leq 12$ and so $\delta_{\nu, \vartheta, \xi} < 0$. Besides, for $L = 19$, then $\nu + \xi \leq 8$ and $\vartheta + \xi \leq 13$, and so

$$\begin{aligned} \deg(\Lambda \mathbf{v}) = 7 < \nu + \tau = 8 \\ \deg(\Lambda d) = 7 < \vartheta + \tau = 13 \end{aligned} \iff \delta_{\nu, \vartheta, \tau} \geq 0 \iff \mathcal{S}_{Y, \nu, \vartheta, \tau} \neq (\mathbf{0}, 0).$$



We now prove that Algorithm 4.2.3 terminates exactly when it reaches the following number of evaluation points

$$\overline{\mathcal{L}}_{\text{ET}} = \min\{\mathcal{L}_{BK}, \mathcal{L}_{KPSW}\}$$

where recall,

- $\mathcal{L}_{BK} = \max\{N + \deg(d), D + \deg(\mathbf{v})\} + \tau + |E|$,
- $\mathcal{L}_{KPSW} = \max\{\deg(A) + \deg(\mathbf{v}) + 1, \deg(\mathbf{b}) + \deg(\mathbf{d}) + 1\} + \tau + |E|$.

Proposition 4.2.2. *Let $L(\nu, \vartheta, \xi) = \min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \xi + \tau$ for the parameters ν, ϑ, ξ .*

Then Algorithm 9 terminates when $L(\nu, \vartheta, \xi) = \overline{\mathcal{L}}_{\text{ET}}$.

Proof. We need to prove the following two facts:

1. if $L(\nu, \vartheta, \xi) < \overline{\mathcal{L}}_{\text{ET}}$, for all ν, ϑ, ξ then $\deg(\mathbf{v}) + |E| \geq \nu + \xi$ or $\deg(d) + |E| \geq \vartheta + \xi$.
2. if $L(\nu, \vartheta, \xi) = \overline{\mathcal{L}}_{\text{ET}}$, then there exist ν, ϑ, ξ such that $\deg(\mathbf{v}) + |E| < \nu + \xi$ and $\deg(d) + |E| < \vartheta + \xi$.

1. We prove the first claim by contraposition. We assume that there exists ν, ϑ, ξ such that $\deg(\mathbf{v}) + |E| < \nu + \xi$ and $\deg(d) + |E| < \vartheta + \xi$. Then, since $\nu + \xi - 1 \geq \deg(\mathbf{v}) + |E|$ and $\vartheta + \xi - 1 \geq \deg(d) + |E|$, we have that

$$\begin{aligned} L(\nu, \vartheta, \xi) &= \min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \xi + \tau \\ &\geq \min\{\max\{N + \deg(d), D + \deg(\mathbf{v})\}, \max\{\deg(A) + \deg(\mathbf{v}) + 1, \deg(\mathbf{b}) + \deg(\mathbf{d}) + 1\}\} \\ &\quad + |E| + \tau = \overline{\mathcal{L}}_{\text{ET}}. \end{aligned}$$

2. If $L(\nu, \vartheta, \xi) = \overline{\mathcal{L}}_{\text{ET}}$, then the claim follows by taking $\nu = \deg(\mathbf{v}) + 1$, $\vartheta = \deg(d) + 1$ and $\xi = |E|$. \square

4.2.2 Our contribution

We now present an early termination technique which allows to further reduce the number of evaluation points. This is a new technique which is the result of a work in progress [GLZ20a]. It is based on the following theorem.

Theorem 4.2.3. *Let $\vartheta, \nu, \xi \geq 0$, $n \geq 1$, consider*

$$L \geq \min\{\mathfrak{L}_{GLZ1}, \mathfrak{L}_{GLZ2}\} =: \tilde{\mathfrak{L}}$$

evaluation points $\{\alpha_1, \dots, \alpha_L\}$, where

- $\mathfrak{L}_{GLZ1} := \max\{\nu + \deg(d), \vartheta + \deg(\mathbf{v})\} + \xi + \left\lceil \frac{|E|}{n} \right\rceil$,
- $\mathfrak{L}_{GLZ2} := \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\} + \xi + \left\lceil \frac{|E|}{n} \right\rceil$.

Also let $E \subseteq \{1, \dots, L\}$. Moreover, fix a PLS $A(x)\mathbf{y}(x) = \mathbf{b}(x)$ and denote by $\frac{\mathbf{v}(x)}{d(x)}$ its solution, with $\gcd(\gcd_i(v_i), d) = 1$ and d monic.

Consider the random matrix Y where we denote by $\mathbf{y}_j := Y_{,j}$ for any $1 \leq j \leq L$, such that*

- if $j \in E$, then \mathbf{y}_j is a uniformly distributed element of $\mathbb{F}_q^{n \times 1}$,
- if $j \notin E$, then $\mathbf{y}_j = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$,

then the solution space of SRFR with inputs Y, ν, ϑ, ξ is

$$\mathcal{S}_{Y, \nu, \vartheta, \tau} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu, \vartheta, \tau}}$$

where

$$\delta_{\nu, \vartheta, \tau} = \min\{\nu - \deg(\mathbf{v}), \vartheta - \deg(d)\} + (\xi - |E|),$$

with probability at least $1 - \frac{\vartheta + \xi}{q}$.

By convention if $\delta_{\nu, \vartheta, \tau} \leq 0$ we set $\langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i \leq \delta_{\nu, \vartheta, \tau}} = \{(\mathbf{0}, 0)\}$.

Proof. The structure of the proof is the same as the proofs of Theorem 4.1.2 and Theorem 4.1.4. We recall that in the proofs of the Theorem 4.1.2 and Theorem 4.1.4 are based on the following two steps:

1. first we need to prove that there exists a draw of \mathbf{y}_j for $j \in E$ for which the corresponding solution space $\mathcal{S}_{Y, \nu, \vartheta, \xi}$ (which in this case depends on ν, ϑ, ξ) is generated by elements of the form $\langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle$. More specifically recall that this inclusion \subseteq is always true and so we need the other one in order to prove the equality.
2. In the second part, we derive the *generic condition* (see Chapter 3) and the bound on fraction of errors for which the solution space is not of the form $\langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle$.

Since here we are considering some general parameters ν, ϑ, ξ instead of the bounds N, D, τ , the only difference between this proof and the previous ones consists in the first part.

We assume that $\mathcal{L}_{GLZ1} \leq \mathcal{L}_{GLZ2}$. We consider $E = \cup_{i=1}^n I_i$, such that for any $1 \leq i \leq n$, $|I_i| \leq \lceil |E|/n \rceil$. Construct a matrix V , such that

- $V_{*,j} = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$, if $j \notin E$,
- $V_{*,j} \in \mathbb{F}_q^{n \times 1}$ is chosen so that

$$\mathbf{v}(\alpha_j) - d(\alpha_j)V_{*,j} = \boldsymbol{\varepsilon}_{i_j} \tag{4.23}$$

where $\boldsymbol{\varepsilon}_{i_j}$ is a vector of the canonical basis of $\mathbb{F}_q^{n \times 1}$.

So, notice that we consider the same matrix as the proof of Theorem 4.1.2. We now consider $(\boldsymbol{\varphi}, \psi) \in \mathcal{S}_{V, \nu, \vartheta, \xi}$. By multiplying (4.23) by $\psi(\alpha_j)$ and since $\boldsymbol{\varphi}(\alpha_j) = V_{*,j}\psi(\alpha_j)$ for any $1 \leq j \leq L$, then

$$\psi(\alpha_j)\mathbf{v}(\alpha_j) - d(\alpha_j)\underbrace{\psi(\alpha_j)V_{*,j}}_{\boldsymbol{\varphi}(\alpha_j)} = \psi(\alpha_j)\boldsymbol{\varepsilon}_{i_j}.$$

Fix $1 \leq i \leq n$, then for any $j \notin I_i$ then (see the proof of Theorem 4.1.2)

$$\psi(\alpha_j)v_i(\alpha_j) - d(\alpha_j)\varphi_i(\alpha_j) = 0.$$

Now, notice that $\deg(\psi v_i - d\varphi_i) \leq \max\{\vartheta + \deg(\mathbf{v}), \deg(d) + \nu\} + \xi - 1$ and that the roots of this polynomial are

$$L - |I_i| \geq L - \lceil |E|/n \rceil \geq \mathfrak{L}_{GLZ1} - \lceil |E|/n \rceil = \max\{\nu + \deg(d), \vartheta + \deg(\mathbf{v})\} + \xi$$

and so it is the zero polynomial. Therefore $\psi \mathbf{v} - d\boldsymbol{\varphi} = \mathbf{0}$. The rest follows by observing that $\frac{\vartheta}{d}$ is such that $\gcd(\gcd_i(v_i), d) = 1$. So, we can conclude that $\mathcal{S}_{V,\nu,\vartheta,\xi} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu,\vartheta,\xi}}$.

We now assume that $\mathfrak{L}_{GLZ2} \leq \mathfrak{L}_{GLZ1}$. Again we consider a partition of E as before, and we construct a matrix V as the proof of Theorem 4.1.4 such that

- $V_{*,j} = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)}$, if $j \notin E$,
- $V_{*,j} \in \mathbb{F}_q^{n \times 1}$ is chosen so that

$$\mathbf{v}(\alpha_j) - d(\alpha_j)V_{*,j} = -A(\alpha_j)^{-1}d(\alpha_j)\boldsymbol{\varepsilon}_{i_j}. \quad (4.24)$$

Now, consider $(\boldsymbol{\varphi}, \psi) \in \mathcal{S}_{V,\nu,\vartheta,\xi}$ and since $d(\alpha_j) \neq 0$, then by performing the same algebraic operations as in (4.17) we get

$$A(\alpha_j)V_{*,j} - \mathbf{b}(\alpha_j) = \boldsymbol{\varepsilon}_{i_j}$$

and finally by multiplying all by $\psi(\alpha_j)$ and since $(\boldsymbol{\varphi}, \psi) \in \mathcal{S}_{V,\nu,\vartheta,\xi}$ then

$$A(\alpha_j)\boldsymbol{\varphi}(\alpha_j) - \mathbf{b}(\alpha_j)\psi(\alpha_j) = \psi(\alpha_j)\boldsymbol{\varepsilon}_{i_j}.$$

We now denote $\mathbf{p} := A(x)\boldsymbol{\varphi}(x) - \mathbf{b}(x)\psi(x)$ and by p_i its i -th component. Fix $1 \leq i \leq n$, then we observe that for any $j \notin I_i$ then $p_i(\alpha_j) = 0$ (by the same argument as in the proof of Theorem 4.1.4). Now, notice that $\deg(p_i) \leq \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\} + \xi - 1$ and that the number of roots is

$$L - |I_i| \geq L - \lceil |E|/n \rceil \geq \mathfrak{L}_{GLZ2} - \lceil |E|/n \rceil \geq \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\} + \xi$$

and so $p_i = 0$. Therefore, $\mathbf{p} = A(x)\boldsymbol{\varphi}(x) - \mathbf{b}(x)\psi(x) = \mathbf{0}$. The rest of the proof follows by observing that since $(\Lambda \mathbf{v}, \Lambda d) \in \mathcal{S}_{Y,\nu,\vartheta,\tau}$ then also $A(x)\mathbf{v}(x) - \mathbf{b}(x)d(x) = \mathbf{0}$ and that $\frac{\vartheta}{d}$ is such that $\gcd(\gcd_i(v_i), d) = 1$. \square

We now introduce Algorithm 10 which given some parameters $\nu, \vartheta, \xi \geq 0$ and using

$$L = \min\{\max\{\nu + D, \vartheta + N\}, \max\{\deg(A) + \nu + 1, \deg(\mathbf{b}) + \vartheta + 1\}\} + \xi + \left\lceil \frac{\tau}{n} \right\rceil \quad (4.25)$$

evaluation points, either recover the solution or determine if the parameters are too small, *i.e.* $\deg(\Lambda \mathbf{v}) > \nu + \xi$ or $\deg(\Lambda d) > \vartheta + \xi$.

Algorithm 10: Algorithm which determines if the bounds are too small

Input : $Y = \left(\frac{v(\alpha_1)}{d(\alpha_1)}, \dots, \frac{v(\alpha_L)}{d(\alpha_L)} \right) + \Xi$,
 $\tau \geq |E| = |\{j \mid \Xi_{*,j} \neq \mathbf{0}\}|$, $N > \deg(\mathbf{v})$, $D > \deg(d)$, $\deg(A)$, $\deg(\mathbf{b})$,
 $\nu, \vartheta, \xi \geq 0$,
 $L = \min\{\max\{N - 1 + \nu, D - 1 + \vartheta\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \xi + \lceil \frac{\tau}{n} \rceil$,
 $\{\alpha_1, \dots, \alpha_L\}$

Output: (\mathbf{v}, d) or “fail” or “the bounds ν, ϑ, ξ are too small”

- 1 Let \mathcal{M} be the $\mathbb{F}_q[x]$ -module generated by solutions in $\mathcal{S}_{Y,N,D,\tau}$;
- 2 **if** $\text{rank}(\mathcal{M}) = 0$ **then**
- 3 **return** “the bounds ν, ϑ, ξ are too small”;
- 4 **else**
- 5 **if** $\text{rank}(\mathcal{M}) = 1$ **then**
- 6 find (φ, ψ) a generator of \mathcal{M} ;
- 7 $\Lambda \leftarrow \text{gcd}(\varphi, \psi)$;
- 8 **return** $\left(\frac{\varphi}{\Lambda}, \frac{\psi}{\Lambda} \right)$
- 9 **else**
- 10 **return** “fail”

Correctness of Algorithm 10. Recall that,

$$\begin{aligned} \deg(\mathbf{v}) + |E| < \nu + \xi &\iff \delta_{\nu, \vartheta, \xi} > 0 \iff \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu, \vartheta, \xi}} \neq \{(\mathbf{0}, 0)\}. \\ \deg(d) + |E| < \vartheta + \xi & \end{aligned}$$

We now observe that since by assumption $N - 1 \geq \deg(\mathbf{v})$, $D - 1 \geq \deg(d)$ and $\tau \geq |E|$, then the number of evaluation points of Algorithm 10 is

$$\begin{aligned} L &= \min\{\max\{N - 1 + \nu, D - 1 + \vartheta\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \xi + \lceil \frac{\tau}{n} \rceil \\ &\geq \min\{\max\{\deg(\mathbf{v}) + \nu, \deg(d) + \vartheta\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \xi + \lceil \frac{|E|}{n} \rceil = \tilde{\mathfrak{L}} \end{aligned}$$

and so by Theorem 4.2.3, we have that since $L \geq \tilde{\mathfrak{L}}$,

$$\underbrace{\mathcal{S}_{Y, \nu, \vartheta, \tau} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu, \vartheta, \xi}}}_{\text{with probability at least } 1 - \frac{\vartheta + \xi}{q}} \begin{cases} = \{(\mathbf{0}, 0)\} & \iff \delta_{\nu, \vartheta, \xi} \leq 0 \\ \neq \{(\mathbf{0}, 0)\} & \iff \delta_{\nu, \vartheta, \xi} > 0 \end{cases}$$

This result allows us to deduce the following proposition, about our Algorithm 10.

Proposition 4.2.4.

— If Algorithm 10 outputs “the bounds ν, ϑ, ξ are too small” at step 3 then $\delta_{\nu, \vartheta, \xi}$ is always negative;

— Otherwise if Algorithm 10 returns (\mathbf{v}, d) , then $\delta_{\nu, \vartheta, \xi} > 0$ with probability at least $1 - \frac{\vartheta + \xi}{q}$.

Proof. First notice that if $\mathcal{S}_{Y, \nu, \vartheta, \xi} = \ker(M_{Y, \nu, \vartheta, \xi}) = \{(\mathbf{0}, 0)\}$, then since $\langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu, \vartheta, \xi}} \subseteq \mathcal{S}_{Y, \nu, \vartheta, \xi} = \{(\mathbf{0}, 0)\}$ we can conclude that $\delta_{\nu, \vartheta, \xi} \leq 0$.

The other claim follows from the previous remark: indeed if Algorithm 10 arrives at step 4, then $\mathcal{S}_{Y, \nu, \vartheta, \xi} \neq \{(\mathbf{0}, 0)\}$ and with probability at least $1 - \frac{\vartheta + \xi}{q}$, then $\mathcal{S}_{Y, \nu, \vartheta, \xi} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu, \vartheta, \xi}} \neq \{(\mathbf{0}, 0)\}$ which implies that $\delta_{\nu, \vartheta, \xi} > 0$. \square

So, by summing up,

— if $\mathcal{S}_{Y, \nu, \vartheta, \xi} = \ker(M_{Y, \nu, \vartheta, \xi}) = \{(\mathbf{0}, 0)\}$ then $\delta_{\nu, \vartheta, \xi} \leq 0$, which means that $\deg(\mathbf{v}) + |E| > \nu + \xi$ or $\deg(d) + |E| > \vartheta + \xi$ and Algorithm 10 outputs the message “the bounds ν, ϑ, ξ are too small”.

— Otherwise, if $\mathcal{S}_{Y, \nu, \vartheta, \xi} = \ker(M_{Y, \nu, \vartheta, \xi})$ is nontrivial, then

— with probability at least $1 - \frac{\vartheta + \xi}{q}$ the solution space is $\mathcal{S}_{Y, \nu, \vartheta, \xi} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu, \vartheta, \xi}} \neq \{(\mathbf{0}, 0)\}$, which implies that $\delta_{\nu, \vartheta, \xi} > 0$ and so by Remark 4.1.4, we have that $\text{rank}(\mathcal{M}) = 1$. Hence, Algorithm 10 returns (\mathbf{v}, d) .

— On the other hand, with probability at most $\frac{\vartheta + \xi}{q}$ the solution space is $\mathcal{S}_{Y, \nu, \vartheta, \xi} \neq \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{\nu, \vartheta, \xi}}$ and we may have that $\delta_{\nu, \vartheta, \xi} < 0$. In this case Algorithm 10 can either returns “fail” or another solution $(\mathbf{v}', d') \neq (\mathbf{v}, d)$.

Notice that this algorithm is deterministic, but since its input contains random errors, it may output an incorrect result with a certain probability. Indeed, we have seen that if $\mathcal{S}_{Y, \nu, \vartheta, \xi} = \ker(M_{Y, \nu, \vartheta, \xi})$ is nontrivial, with probability at most $\frac{\vartheta + \xi}{q}$, the algorithm can output an incorrect solution, *i.e.* it can either returns “fail” or $(\mathbf{v}', d') \neq (\mathbf{v}, d)$. For this reason we can compare this algorithm to a probabilistic Monte Carlo algorithm. Indeed, recall that a Monte Carlo algorithm is an algorithm which can produce an incorrect solution with a certain probability. For decision problems, a Monte Carlo algorithm is one-sided if the probability that it computes an incorrect solution is zero for at least one possible output that it produces [MR95]. In this case, this algorithm can be compared to a *one-sided* Monte Carlo algorithm for the decision problem “is the solution space $\mathcal{S}_{Y, \nu, \vartheta, \xi} = \ker(M_{Y, \nu, \vartheta, \xi})$ trivial?” Indeed, notice that it always returns the message “the bounds ν, ϑ, ξ are too small” if the kernel is trivial, which is the correct solution since in this case $\delta_{\nu, \vartheta, \xi} \leq 0$.

We conclude this subsection by introducing an early termination Algorithm 11 which terminates when it reaches the following number of evaluation points

$$\widetilde{\mathcal{L}}_{\text{ET}} = \min\{\mathcal{L}_{\text{GLZ1}}, \mathcal{L}_{\text{GLZ2}}\} \quad (4.26)$$

where

— $\mathcal{L}_{\text{GLZ1}} := \max\{N + \deg(d), D + \deg(\mathbf{v})\} + |E| + \lceil \frac{\tau}{n} \rceil$,

— $\mathcal{L}_{\text{GLZ2}} := \max\{\deg(A) + \deg(\mathbf{v}) + 1, \deg(\mathbf{b}) + \deg(d) + 1\} + |E| + \lceil \frac{\tau}{n} \rceil$.

This algorithm is obtained by slightly modifying Algorithm 9 and by adapting it to our number of evaluation points.

Algorithm 11: Adapted Early termination Algorithm

Input : a stream of vectors (\mathbf{y}_j) for $j = 1, \dots$ which is extensible in length on demand, where $\mathbf{y}_j = \frac{\mathbf{v}(\alpha_j)}{d(\alpha_j)} + \mathbf{e}_j$
 τ an upper bound on the number of errors, $N > \deg(\mathbf{v})$, $D > \deg(d)$,
 $\deg(A)$, $\deg(\mathbf{b})$
Output: (\mathbf{v}, d)

```

1  $\nu \leftarrow 0$ ;
2  $\vartheta \leftarrow 0$ ;
3  $\xi \leftarrow 0$ ;
4  $L \leftarrow \min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \xi + \lceil \frac{\tau}{n} \rceil$ ;
5 if Algorithm 10( $Y, \tau, \nu, \vartheta, L, \{\alpha_1, \dots, \alpha_L\}$ ) =  $(\mathbf{v}, d)$  then
6    $\lfloor$  return  $(\mathbf{v}, d)$ 
7 while true do
8    $L \leftarrow L + 1$ ;
9   require a new  $\mathbf{y}_L$ ;
10  foreach  $(\nu, \vartheta, \xi)$  with
     $\min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \xi + \lceil \frac{\tau}{n} \rceil = L$  do
11    if Algorithm 10( $Y, \tau, \nu, \vartheta, L, \{\alpha_1, \dots, \alpha_L\}$ ) =  $(\mathbf{v}, d)$  then
12       $\lfloor$  return  $(\mathbf{v}, d)$ 

```

As in Proposition 4.2.2 we now prove that this algorithm eventually terminates.

Proposition 4.2.5. *Let $\tilde{L}(\nu, \vartheta, \xi) = \min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \xi + \lceil \frac{\tau}{n} \rceil$ for the parameters ν, ϑ, ξ .*

Then Algorithm 11 terminates when $L(\nu, \vartheta, \xi) = \tilde{\mathcal{L}}_{\text{ET}}$ (see (4.26)).

Proof. As for Proposition 4.2.2, we need to prove the following two facts:

1. if $L(\nu, \vartheta, \xi) < \overline{\mathcal{L}}_{\text{ET}}$, then for all ν, ϑ, ξ , $\deg(\mathbf{v}) + |E| \geq \nu + \xi$ or $\deg(d) + |E| \geq \vartheta + \xi$.
2. if $L(\nu, \vartheta, \xi) = \overline{\mathcal{L}}_{\text{ET}}$, then there exist ν, ϑ, ξ such that $\deg(\mathbf{v}) + |E| < \nu + \xi$ and $\deg(d) + |E| < \vartheta + \xi$.

1. We prove the first claim by contraposition. We assume that there exists ν, ϑ, ξ such that $\deg(\mathbf{v}) + |E| < \nu + \xi$ and $\deg(d) + |E| < \vartheta + \xi$. Then, since $\nu + \xi - 1 \geq \deg(\mathbf{v}) + |E|$ and $\vartheta + \xi - 1 \geq \deg(d) + |E|$, we have that

$$\begin{aligned}
L(\nu, \vartheta, \xi) &= \min\{\max\{N - 1 + \vartheta, D - 1 + \nu\}, \max\{\deg(A) + \nu, \deg(\mathbf{b}) + \vartheta\}\} + \xi + \lceil \frac{\tau}{n} \rceil \\
&\geq \min\{\max\{N + \deg(d), D + \deg(\mathbf{v})\}, \max\{\deg(A) + \deg(\mathbf{v}) + 1, \deg(\mathbf{b}) + \deg(d) + 1\}\} \\
&\quad + |E| + \lceil \frac{\tau}{n} \rceil = \tilde{\mathcal{L}}_{\text{ET}}.
\end{aligned}$$

2. If $L(\nu, \vartheta, \xi) = \widetilde{\mathcal{L}}_{\text{ET}}$, then the claim follows by taking $\nu = \deg(\mathbf{v}) + 1$, $\vartheta = \deg(d) + 1$ and $\xi = |E|$. \square

4.3 Conclusion and open problems

We now shortly summarize the main results of this chapter and we present open problems and future perspectives related to them.

Algorithm-based fault tolerant technique for PLS solving by evaluation-interpolation.

In this chapter we studied an application of error correcting codes that goes beyond the classical communication scenario in which they are traditionally applied. Indeed, we used the algebraic tools of error correcting codes for constructing an algorithm-based fault tolerant technique for solving polynomial linear systems by evaluation-interpolation.

The ABFT is a technique which exploits the algorithm's features in order to design a fault tolerant algorithm. This technique is characterized by the following three steps: encoding, parallelization and the redesign of the algorithm so that it can work with the encoded data.

We applied this technique to the polynomial linear system solving by evaluation-interpolation. So, informally speaking, we modified the classic evaluation-interpolation algorithm for PLS solving by

1. adding redundancy by encoding inputs of the algorithm by considering more evaluation points than the classic algorithm,
2. parallelizing the evaluation step, which is performed by different nodes which can introduce some errors,
3. performing the interpolation step on the encoded data affected by some errors.

Note that in this model, the errors are introduced at the parallelization step. The third step is related to what we called *polynomial linear system solving with errors* (PLSwE), which is the problem of recovering the solution of a PLS (which is a vector of rational functions) given its evaluations where some are possibly erroneous.

Simultaneous Cauchy Interpolation with Errors as the decoding of Interleaved Rational Function Codes.

PLSwE is an application of a more general problem that we introduced in Subsection 4.1.1: the *simultaneous Cauchy interpolation with errors* (SCIwE). This is the problem of reconstructing a *vector of rational functions*, given its evaluations, where some are erroneous. It can be seen as the *rational extension* of the *simultaneous interpolation with errors* (SIwE) (see Definition 2.3.2), which is basically the problem of decoding interleaved RS codes. This connection allowed us to extend the interpolation-based decoding technique of IRS to this rational case.

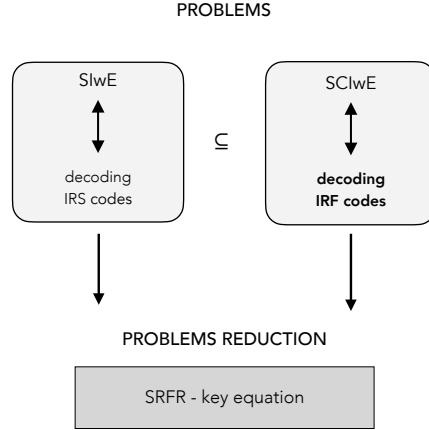


Figure 4.4: Scheme which illustrates the relation between SIwE, SCIwE and IRF codes

But there is more than that. Indeed, even in this case, SCIwE is related to a coding theory problem which is the decoding of particular interleaved *rational function codes*. These codes represent the rational generalization of IRS codes and they were first introduced in [Per14] (see Figure 4.4).

We now define *rational function codes*.

Definition 4.3.1 (Rational Function Codes). Let $N, D \leq L \leq q$ and $\{\alpha_1, \alpha_2, \dots, \alpha_L\}$ be pairwise distinct evaluation points in \mathbb{F}_q . The *rational function* (RF) code is

$$\mathcal{C}_{RF}(L, N, D) := \left\{ \left(\frac{v(\alpha_1)}{d(\alpha_1)}, \dots, \frac{v(\alpha_L)}{d(\alpha_L)} \right) \middle| \frac{v}{d} \in \mathbb{F}_q(x), \deg(v) < N, \deg(d) < D, d(\alpha_i) \neq 0 \right\} \in \mathbb{F}_q^L$$

Notice that this is a generalization of RS codes.

As for classic RS codes, we can define *interleaved rational function codes* as follows.

Definition 4.3.2 (Interleaved Rational Function Codes). Let $n \geq 1, N, D \leq L \leq q$ and $\{\alpha_1, \alpha_2, \dots, \alpha_L\}$ be pairwise distinct evaluation points in \mathbb{F}_q . An (*homogeneous*) *interleaved rational function* (IRF) code is

$$\mathcal{C}_{IRF}(L, N, D) := \left\{ \left(\begin{array}{c} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_n \end{array} \right) \middle| \mathbf{c}_i \in \mathcal{C}_{RF}(L, N, D), 1 \leq i \leq n \right\} \in (\mathbb{F}_q)^{n \times L}$$

Notice that we can see an IRF code as the evaluation of a vector of rational functions whose degrees of any numerator and denominator are bounded,

$$\mathcal{C}_{IRF}(L, N, D) := \left\{ \left(\frac{\mathbf{v}(\alpha_1)}{\mathbf{d}(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{\mathbf{d}(\alpha_L)} \right) \middle| \frac{\mathbf{v}}{\mathbf{d}} \in \mathbb{F}_q(x)^{n \times 1}, \deg(\mathbf{f}) < N, \deg(\mathbf{d}) < D, d_i(\alpha_j) \neq 0 \right\}.$$

This is the rational extension of IRS codes.

We now observe that the SCIwE problem can be seen as the problem of decoding IRF codes, in particular when the codeword that we want to recover is a vector of rational functions with the *same denominator*.

There are many open problems related to this rational code and to its interleaved version. First of all, notice that in this rational extension of RS codes we lose some important properties, like the *linearity* of the code. This prevents rational codes to have all the nice characteristics of linear codes (see Subsection 2.1.3). So we would like to study these codes more in depth, determine their properties and parameters (like their minimum distance) and understand other possible applications besides the one proposed in this chapter.

Following [Per14, Theorem 2.3.1] we can prove that the minimum distance of this code is $d \geq L - (N + D + 2)$.

More specifically, we remark that we provide a slightly different definition of RF codes compared to [Per14, Definition 2.3.1] since we consider denominators not vanishing on the fixed evaluation points. Indeed, we recall that this assumption on the denominator was crucial in the proof of Theorem 4.1.2. This difference makes it difficult to compute exactly the minimum distance of this code, showing the existence of two codewords at distance exactly $d = L - (N + D + 2)$.

Therefore, there are two different possible research tracks: either extend the proof of Theorem 4.1.2 in order to include the possibility for the denominator to vanish at evaluation points, or continue to investigate this slightly different rational function code (which is a subcode of the one defined in [Per14, Definition 2.3.1]) of Definition 4.3.1 and its properties.

Analysis of Algorithm 6 from a Coding Theory point of view. We recall that a satisfiable instance of SCIwE (Definition 4.1.1) with parameters L, τ, N, D, q and $\{\alpha_1, \dots, \alpha_n\}$ is a matrix

$$Y = \left(\frac{\mathbf{v}(\alpha_1)}{d(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{d(\alpha_L)} \right) + \Xi$$

where

- $\frac{\mathbf{v}}{d}$ is a vector of rational functions with $\gcd(\gcd_i(v_i), d) = 1$ and $\deg(\mathbf{v}) < N$, $\deg(d) < D$, d is monic and $d(\alpha_j) \neq 0$ for any $1 \leq j \leq L$,
- $\Xi \in \mathbb{F}_q^{n \times L}$ is an error matrix, with error support $E := \{j \mid \Xi_{*,j} \neq \mathbf{0}\}$ and $|E| \leq \tau$.

As previously said, this instance can be seen as a received word of a n -IRF code with length L and parameters N, D and so SCIwE can be seen as the decoding of IRF codewords.

We now briefly sum up the two main results of Subsection 4.1.1:

- By Theorem 4.1.1, if we consider

$$L \geq N + D + 2\tau - 1 = L_{BK}$$

evaluation points, or equivalently if

$$\tau \leq \frac{L - (N + D - 1)}{2} = \tau'_0$$

we have that

$$\mathcal{S}_{Y,N,D,\tau} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}}$$

where $\delta_{N,D,\tau} = \min\{N - \deg(\mathbf{v}), D - \deg(d)\} + (\tau + |E|)$. This result derives from [BK14].

— On the other hand in [GLZ19] (see Theorem 4.1.2) we proved that with

$$L = N + D - 1 + \tau + \left\lceil \frac{\tau}{n} \right\rceil = L_{GLZ1}$$

evaluation points, or equivalently if

$$\tau \leq \frac{n(L - (N + D - 1))}{n + 1} = \tau_{GLZ}$$

under some assumptions on the error distribution,

$$\mathcal{S}_{Y,N,D,\tau} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}} \text{ with probability } \geq 1 - \frac{D + \tau}{q}.$$

We now observe that the former result tells us that if the number of errors is smaller than τ'_0 then we can uniquely recover the vector of rational functions $\frac{\mathbf{v}}{d}$ corresponding to the codeword of an IRF. This seems to suggest that the *error correction capability* of IRF and also RF codes is exactly τ'_0 (see Theorem 2.1.1) and so that the minimum distance of this code is $L - (N - D + 2)$.

Besides, as for IRS codes, we can use the interleaving construction to construct a decoder with a bigger decoding radius. However, since we are beyond τ'_0 we may lose the guarantee of the uniqueness of the decoding.

This is exactly what we did in [GLZ19]. Indeed, thanks to the result of Theorem 4.1.2, we constructed Algorithm 6, which can be seen as a *partial BD decoder of specific IRF codewords*. This algorithm is a generalized version of the partial BD decoder of IRS codes (Algorithm 4) introduced in Subsection 2.3.1. We now analyze the behavior of Algorithm 6 as a decoder, as we did for the partial BD decoder of IRS codes in Remark 2.3.6. First recall that $\mathcal{B}^{\tau_{GLZ}}(Y)$ is the Hamming ball of radius τ_{GLZ} , centered in Y and that we are considering the Hamming distance in the vector space \mathbb{F}_q^L . We now observe that since in this case we are assuming that $|E| \leq \tau \leq \tau_{GLZ}$ then $\mathcal{C}_{IRF}(L, N, D) \cap \mathcal{B}^{\tau_{GLZ}}(Y) \neq \emptyset$ and so we have the following two possibilities,

— if $|\mathcal{C}_{IRF}(L, N, D) \cap \mathcal{B}^{\tau_{GLZ}}(Y)| = 1$, then with probability at least $1 - \frac{D + \tau}{q}$ the solution space is $\mathcal{S}_{Y,N,D,\tau} = \langle x^i \Lambda \mathbf{v}, x^i \Lambda d \rangle_{0 \leq i < \delta_{N,D,\tau}}$ and so $\text{rank}(\mathcal{M}) = 1$. In this case, the *gcd*

computed at step 4 of Algorithm 6 is exactly the error locator polynomial Λ and so by dividing the generator (φ, ψ) (computed at step 3) of \mathcal{M} by Λ we obtain (\mathbf{v}, d) .

- If $|\mathcal{C}_{IRF}(L, N, D) \cap \mathcal{B}^{(\tau_{GLZ})}(Y)| > 1$, then by Remark 4.1.4, $\text{rank}(\mathcal{M}) > 1$ and so the algorithm outputs a failure message.

In conclusion we observe that this decoder is partial since for Y such that $|\mathcal{C}_{IRF}(L, N, D) \cap \mathcal{B}^{(\tau_{GLZ})}(Y)| = 1$ it may possibly fail, even if there is only one codeword in the ball.

Failure Probability. In this chapter we saw how we can generalize the interpolation-based decoding technique for IRS to solve the SCIwE problem, or in other words, to decode IRF codes. We introduced Algorithm 6 which can be seen as a partial BD decoder of these codes, and we saw that under some assumptions on the error distribution, this algorithm could fail with probability which depends on the bound D on the degree of the denominator, on the bound on the number of errors τ and on the order of the field q . Specifically, this *failure probability* is at most $(D + \tau)/q$. Notice that this is a generalization of the failure probability of Algorithm 4 for IRS codes, *i.e.* τ/q (see Lemma 2.3.1, [BKY03]). However, in Subsection 2.3.1 we pointed out that in [BMS04], A. Brown *et al.* proved that the failure probability of the interpolation-based partial BD decoder for IRS codes, does not depend on the number of errors. Later, [SSB07, SSB09, SSB10] introduced another decoding algorithm based on the syndrome-based approach and estimated its failure probability (see (2.26)). The bound introduced in these articles is tight in practice.

In our case, indeed, experiments implemented in SageMath suggest that the failure probability of our algorithm does not depend on the number of errors, as for IRS codes. For this reason a possible future research track could be to better estimate the failure probability of our algorithm.

On the uniqueness of simultaneous rational function reconstruction related to SCIwE. In this chapter we saw how we can reduce the SCIwE problem to the simultaneous rational function reconstruction. Indeed, given an instance Y with parameters L, τ, N, D, q and evaluation points $\{\alpha_1, \dots, \alpha_L\}$ as above, we find a solution of SRFR related to Y by performing an SRFR in the interpolation form, with inputs $Y, N + \tau, D + \tau$ and the given evaluation points. As a matter of fact we want to recover (φ, ψ) such that

$$\varphi(\alpha_j) = Y_{*,j}\psi(\alpha_j), \quad \deg(\varphi) < N + \tau, \quad \deg(\psi) < D + \tau \quad (4.27)$$

for any $1 \leq j \leq L$, since our main goal is to recover the solution $(\Lambda \mathbf{v}, \Lambda d)$, where Λ is the error locator polynomial. We now observe that Theorem 4.1.2 basically tells us that, given $1 \leq N, \tau, D \leq L \leq q$ and an error support $E \subseteq \{1, \dots, L\}$ such that $|E| \leq \tau$, if $L \geq L_{GLZ1}$, then for all $(\mathbf{v}, d) \in \mathbb{K}[x]^{n+1}$ and for almost all error matrices Ξ of error support E , then the corresponding SRFR admits a unique solution on the instance $Y = \left(\frac{\mathbf{v}(\alpha_1)}{d(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{d(\alpha_L)} \right) + \Xi$.

Notice that if $\tau = 0$, *i.e.* in the case where there are no errors, we get $L \geq L_{GLZ1} = N + D - 1 + \tau + \lceil \frac{\tau}{l} \rceil = N + D - 1 = L_{BK}$ and so SRFR always admits a unique solution.

We now observe that if we consider the SRFR problem of (4.27) and the corresponding homogeneous linear system, then the number of equations is nL and the number of unknowns is $n(N + \tau) + D + \tau$. By the Rank-Nullity Theorem, if

$$nL = n(N + \tau) + D + \tau - 1 \iff L = N + \tau + (D + \tau - 1)/n$$

then there exists a nontrivial solution of (4.27).

Conjecture 4.3.1. Fix $1 \leq N, \tau, D \leq L \leq q$ and an error support $E \subseteq \{1, \dots, L\}$ such that $|E| \leq \tau$. If

$$L = N + \tau + (D + \tau - 1)/n \tag{4.28}$$

then for almost all (\mathbf{v}, d) and almost all error matrices Ξ of error support E , SRFR admits a unique solution on the instance $Y = \left(\frac{\mathbf{v}(\alpha_1)}{d(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{d(\alpha_L)} \right) + \Xi$.

We point out that we conjecture the uniqueness *for almost all* (\mathbf{v}, d) whereas Theorem 4.1.2 holds for all (\mathbf{v}, d) . This difference is due to our counterexample (Example 1.2.1) which shows that we cannot have uniqueness for all instances $\mathbf{u} = \frac{\mathbf{v}}{d}$ whenever $L = N + (D - 1)/n$.

We also remark that this latter number of evaluations matches the one in the Conjecture 3.3.1 in the case $\tau = 0$.

Our Theorem 3.2.1 is a step towards Conjecture 4.3.1. Indeed, since we proved uniqueness of SRFR under assumption (4.28) for generic instances \mathbf{u} , it remains to prove the existence of an instance of the form $\mathbf{u} = \left(\frac{\mathbf{v}(\alpha_1)}{d(\alpha_1)}, \dots, \frac{\mathbf{v}(\alpha_L)}{d(\alpha_L)} \right) + \Xi$ for any N, D, τ, E in order to prove the conjecture.

Polynomial Linear System Solving with Errors. SCIwE is the problem of recovering a vector of rational functions with the same denominator given its evaluations, some of which are possibly erroneous. The PLSwE is then a specific case of SCIwE in which we want to recover a vector of rational functions which is a solution of a polynomial linear system. Recall indeed, that the main aim of this chapter was to introduce an ABFT technique for PLS solving by evaluation-interpolation. In order to do so, we delegated the computation of the evaluated systems to some nodes which could eventually introduce some errors. Therefore, the PLSwE coincides exactly with the third step of our technique, in which we want to recover the solution of our PLS by its evaluations where some have been corrupted by the nodes computations.

We saw how we can apply the same technique for SCIwE also to this case and reduce the problem to SRFR in which we want to recover $(\Lambda \mathbf{v}, \Lambda d)$, where Λ is the error locator polynomial. Moreover, since we have considered a vector of rational functions which is a solution of a PLS we can also add some additional parameters which are the degree of the coefficient matrix and of the vector of the corresponding PLS. Also in this case, the main

goal was to determine the number of evaluation points which guarantees the uniqueness of the corresponding SRFR.

In [KPSW17], E. Kaltofen *et al.* proved that with

$$\bar{L} = \min\{L_{BK}, L_{KPSWE}\}, \quad (4.29)$$

where

- $L_{BK} = N + D - 1 + 2\tau$,
- $L_{KPSWE} = \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + 2\tau$,

we can uniquely recover $(\Lambda\mathbf{v}, \Lambda d)$ and then in particular (\mathbf{v}, d) .

Besides, in Theorem 4.1.4 we showed how we can decrease this number of points to

$$\tilde{L} = \min\{L_{GLZ1}, L_{GLZ2}\}, \quad (4.30)$$

where

- $L_{GLZ1} = N + D - 1 + \tau + \lceil \frac{\tau}{n} \rceil$,
- $L_{GLZ2} = \max\{\deg(A) + N, \deg(\mathbf{b}) + D\} + \tau + \lceil \frac{\tau}{n} \rceil$.

We observe that since we are below the number of evaluation points that guarantees to uniquely reconstruct the solution our Algorithm 7 (derived from Theorem 4.1.4), could eventually fail.

Early Termination. All the bounds on the number of evaluation points introduced so far, *i.e.* \bar{L} and \tilde{L} of (4.29), (4.30), strongly depends on the bounds on the degrees of the solution that we want to recover and on the bound on the number of errors. Therefore, if we consider N, D, τ much bigger than the real degrees of the solution or than the real number of errors, we could significantly overestimate the number of evaluation points, compared to the real number of points that we really need, *i.e.*

$$\bar{\mathcal{L}}_{\text{ideal}} = \min\{\mathcal{L}_{BK}, \mathcal{L}_{KPSW}\} \quad (4.31)$$

where

- $\mathcal{L}_{BK} = \deg(\mathbf{v}) + \deg(d) + 2|E| + 1$,
- $\mathcal{L}_{KPSW} = \max\{\deg(A) + \deg(\mathbf{v}), \deg(\mathbf{b}) + \deg(d)\} + 2|E| + 1$;

or

$$\tilde{\mathcal{L}}_{\text{ideal}} = \min\{\mathcal{L}_{GLZ1}, \mathcal{L}_{GLZ2}\} \quad (4.32)$$

where

- $\mathcal{L}_{GLZ1} = \deg(\mathbf{v}) + \deg(d) + |E| + \lceil \frac{|E|}{n} \rceil + 1$,

$$— \mathcal{L}_{GLZ2} = \max\{\deg(A) + \deg(\mathbf{v}), \deg(\mathbf{b}) + \deg(d)\} + |E| + \left\lceil \frac{|E|}{n} \right\rceil + 1.$$

In [KPSW17], E. Kaltofen *et al.* proposed an *early termination algorithm* (Algorithm 9), which starting from a minimal number of evaluation points, iteratively increases this number until a result is found. We remark that the main goal of this technique is to possibly decrease the number of evaluation points to speed up the computations. So, basically this algorithm terminates when a sort of stabilization is detected, that is when the minimum number which guarantees the existence of a solution is reached. More specifically, this number is

$$\overline{\mathcal{L}}_{\text{ET}} = \min\{\mathcal{L}_{BK}, \mathcal{L}_{KPSW}\} \quad (4.33)$$

where

$$\begin{aligned} — \mathcal{L}_{BK} &= \max\{N + \deg(d), D + \deg(\mathbf{v})\} + \tau + |E|, \\ — \mathcal{L}_{KPSW} &= \max\{\deg(A) + \deg(\mathbf{v}) + 1, \deg(\mathbf{b}) + \deg(d) + 1\} + \tau + |E|. \end{aligned}$$

Notice that \mathcal{L}_{KPSW} is really close to the ideal number \mathcal{L}_{KPSW} of (4.31).

In Section 4.2, we also presented our algorithm (Algorithm 11) which allowed us to further decrease the number of evaluation points compared to (4.33). Precisely, this algorithm terminates when the following number of evaluation points is reached,

$$\widetilde{\mathcal{L}}_{\text{ET}} = \min\{\mathcal{L}_{GLZ1}, \mathcal{L}_{GLZ2}\} \quad (4.34)$$

where

$$\begin{aligned} — \mathcal{L}_{GLZ1} &= \max\{N + \deg(d), D + \deg(\mathbf{v})\} + |E| + \left\lceil \frac{\tau}{n} \right\rceil, \\ — \mathcal{L}_{GLZ2} &= \max\{\deg(A) + \deg(\mathbf{v}) + 1, \deg(\mathbf{b}) + \deg(d) + 1\} + |E| + \left\lceil \frac{\tau}{n} \right\rceil. \end{aligned}$$

We observe that from a coding theory point of view, an overestimation on the bounds on the degrees of the solution that we want to recover, could significantly decrease the number of errors that we could correct. Indeed, in [KPR⁺10], authors already observed that even for classic RS codes, the overestimation of the bound on the degree of the polynomial related to an RS codeword instead of the real degree could significantly decrease the amount of errors that the decoder could correct. For this reason, [KPR⁺10, Per14] proposed a *parameter oblivious algorithm* which could correct up to

$$\frac{n - \deg(\mathbf{f}) - 1}{2} \geq \frac{n - k}{2}$$

errors, for a given $f \in \mathcal{C}_{RS}(n, k)$. So, a natural question related to this topic could concern the extension of this technique to other codes, as for instance IRS codes.

Concluding Remarks

As a conclusion we summarize the main contributions and the proposed new research perspectives related to them (which we already introduced in Section 3.3 and Section 4.3). We divide this presentation into two sections according to the purpose of these new research tracks and open problems.

Improving the previous results

Simultaneous Rational Function Reconstruction problem. One of the main contribution of this work concerns [GLZ20b] the general SRFR problem (see Chapter 3). Indeed, in Theorem 3.2.1 we proved that under the assumption on the degrees of the moduli derived from the common denominator constraint, SRFR admits a unique solution for almost all instances \mathbf{u} .

In Section 3.3 we noticed that this result represents a step towards Conjecture 3.3.1 according to which for almost all (\mathbf{v}, d) with d invertible, then SRFR with instance $\mathbf{u} = \frac{\mathbf{v}}{d}$ admits a unique solution.

Indeed, in our result we showed uniqueness for almost all instances \mathbf{u} which could not derive from a vector of rational functions. This is due to the fact that, following the classic RFR problem, we focused on the weaker linear problem of recovering (\mathbf{v}, d) such that $v_i = du_i \bmod a_i$ and with $\deg(v_i) < N_i$, $\deg(d) < D$, dropping the invertibility of d modulo a_i .

Nevertheless, since in Theorem 3.2.1 we proved uniqueness for generic instances \mathbf{u} , it would be sufficient to show the existence of an instance $\mathbf{u} = \mathbf{v}/d$ to prove Conjecture 3.3.1.

Failure probability of SCIwE and PLSwE algorithms. In Chapter 4 we introduced Algorithm 6 for SCIwE solving which generalizes the interpolation-based decoding technique of IRS codes. In Section 4.3, we saw how SCIwE can be seen as the decoding of IRF codes (Definition 4.3.2). We also reinterpreted Algorithm 6 as a partial BD decoder for these kind of codes. Indeed, under some assumptions on the error distribution, the Algorithm 6 may fail with a certain probability (Theorem 4.1.2). Specifically, the *failure probability* is at most $(D + \tau)/q$, where D and τ are respectively bounds on the degree of the common denominator of all rational functions and on the number of errors. We also remarked that this probability bound generalizes the failure probability bound of the partial BD decoder of

IRS codes (Lemma 2.3.1, [BKY03]), *i.e.* τ/q . This IRS decoding failure probability bound was then improved ([BMS04, SSB07, SSB09, SSB10]) by showing that the failure probability does not depend on the number of errors or on its bound.

In the case of Algorithm 6, experiments implemented in `SageMath` suggest that even in our rational generalization, the failure probability does not depend on the number of errors. Hence, it would be interesting to improve the bound on the failure probability, trying to find a more tight one.

Extending the previous results

Rational Function Codes. In Section 4.3 we introduced Rational Function codes (Definition 4.3.1) and their interleaved version (Definition 4.3.2). We saw the link between the decoding of IRF codes and the SCIwE problem. However, as previously underlined, there are many open problems related to these codes. First, RF codes are not *linear* and so they do not have all the useful properties of linear codes. Also, manipulating them could be more complicated since we cannot reduce all the related problems, *e.g.* the decoding, to linear problems.

A future and ambitious research track would be a better investigation and study of these codes, starting from the determination of their parameters. It would also be interesting to understand other possible application scenarios, different from the one proposed in this work.

Early termination techniques. In this thesis, by extending the results of [BK14, KPSW17] we proposed an *early termination technique* which leads to a possible reduction on the number of evaluation points needed to PLSwE solving. This strategy, starting from a small value of evaluation points, dynamically increase this number until a result is found. This means that the minimal number which guarantees to uniquely recover the solution is attained.

In [KPR⁺10], the authors observed that the problem of overestimating the degree bounds leading to more evaluation points compared to the needed one (for the RFR in this case), could affect also error correcting codes and in particular RS codes. Indeed, even for RS codes an overestimation of the bound k on the degree of the codeword polynomial, could significantly increase the number of evaluation points needed for the uniqueness of RFR (and so uniqueness of the decoding). Equivalently it can decrease the amount of errors that the decoder could uniquely correct. Indeed,

$$\tau = \frac{n - \deg(f) - 1}{2} \geq \frac{n - k}{2} = \tau_0$$

for $f \in \mathcal{C}_{RS}(n, k)$. For this reason, [KPR⁺10] proposed *parameter oblivious* algorithm which basically performs an early termination technique on the corresponding RFR problem related to the decoding.

So, in view of these results it would be interesting to extend this technique also to IRS codes or to other codes constructions [Jus06].

List of Acronyms

ABFT	Algorithm-Based Fault Tolerance Technique
BD	Bounded Distance (code)
BMD	Bounded Minimum Distance (decoder)
BSC	Binary Symmetric Channel
EEA	Extended Euclidean Algorithm
GRS	Generalized Reed-Solomon (code)
IRF	Interleaved Rational Function (code)
IRS	Interleaved Reed-Solomon (code)
IwE	Interpolation with Errors
MDS	Maximum Distance Separable (code)
ML	Maximum Likelihood (decoder)
NC	Nearest Codeword (decoder)
PID	Principal Ideal Domain
PLS	Polynomial Linear System
PLSwE	Polynomial Linear System with Errors
POT	Position Over Term (monomial ordering)
RF	Rational Function (code)
RRP	Row Rank Profile
RS	Reed-Solomon (code)
RFR	Rational Function Reconstruction
SIwE	Simultaneous Interpolation with Errors
SRFR	Simultaneous Rational Function Reconstruction
SCIwE	Simultaneous Cauchy Interpolation with Errors
TOP	Term Over Position (monomial ordering)
VRFR	Vector Rational Function Reconstruction

List of Algorithms

1	Extended Euclidean Algorithm, $\text{EEA}(f, g)$	32
2	Rational function reconstruction by EEA, $\text{RFR}_{\text{EEA}}(a, u, N)$	33
3	Interpolation-based BMD decoder based on EEA	66
4	Partial BD decoder for IRS codes	74
5	Algorithm for SCIwE of [BK14]	108
6	A new algorithm for SCIwE	109
7	A new algorithm for PLSwE	115
8	Algorithm which computes $\frac{v}{d}$ or determine if the degree bounds are too small [KPSW17]	118
9	Early termination algorithm from [KPSW17]	121
10	Algorithm which determines if the bounds are too small	126
11	Adapted Early termination Algorithm	128

List of Figures

2.1	Nearest-codeword decoding	58
2.2	Bounded minimum distance decoding	59
2.3	Existence of two codewords $\mathbf{c}_1, \mathbf{c}_2$ at the same distance $\leq \tau$ from the received word \mathbf{y}	59
2.4	Received word of an IRS code under the burst error model.	70
3.1	Construction of Example 3.1.6	91
3.2	Construction of Krylov matrices of the proof of Theorem 3.2.1. In this case $n = 3$	97
3.3	Construction of the last Krylov matrix of the proof of Theorem 3.2.1. In this case $n = 3$	98
4.1	Matrix multiplication by ABFT method.	102
4.2	Parallelization in the evaluation step of PLS solving	104
4.3	Scheme which illustrates the relation between SIwE and SCIwE.	106
4.4	Scheme which illustrates the relation between SIwE, SCIwE and IRF codes	130

Bibliography

- [AL94] W. Adams and P. Loustau. *An Introduction to Gröbner Bases*. Amer Mathematical Society, 7 1994.
- [BCG⁺17] A. Bostan, F. Chyzak, M. Giusti, R. Leducq, G. Lecerf, B. Salvy, and E. Schost. *Algorithmes Efficaces en Calcul Formel*. Frédéric Chyzak (auto-édit.), 2017.
- [BD93] A. Bode and M. Dal Cin. *Parallel Computer Architectures*. Springer-Verlag, 1993.
- [BDFP15] J. Böhm, W. Decker, C. Fieker, and G. Pfister. The use of bad primes in rational reconstruction. *Mathematics of Computation*, 84(286):3013 – 3027, 2015.
- [Ber68] E. R. Berlekamp. Nonbinary bch decoding. *IEEE Transactions on Information Theory*, 14(2):242–242, 1968.
- [Ber15] E. R. Berlekamp. *Algebraic Coding Theory - Revised Edition*. World Scientific Publishing Co., Inc., USA, 2015.
- [BK14] B. Boyer and E. Kaltofen. Numerical Linear System Solving with Parametric Entries by Error Correction. In *Proceedings of SNC'14*, pages 33 – 38. ACM, 2014.
- [BKY03] D. Bleichenbacher, A. Kiayias, and M. Yung. Decoding of interleaved Reed-Solomon codes over noisy data. In *Proceedings of ICALP'03*, pages 97 – 108, 2003.
- [BL94] B. Beckermann and G. Labahn. A Uniform Approach for the Fast Computation of Matrix-Type Padé Approximants. *SIAM J. Matrix Anal. Appl.*, 15(3):804 – 823, 1994.
- [BL97] B. Beckermann and G. Labahn. Recursiveness in matrix rational interpolation problems. *Journal of Computational and Applied Mathematics*, 77(1):5 – 34, 1997.
- [Bla03] R. E. Blahut. *Algebraic Codes for Data Transmission*. Cambridge University Press, 2003.
- [BMS04] A. Brown, L. Minder, and A. Shokrollahi. Probabilistic decoding of Interleaved RS-Codes on the Q-ary symmetric channel. In *Proceedings of ISIT'04*, pages 326 – 326. IEEE, 2004.

- [BMT78] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384 – 386, 1978.
- [BW86] E. R. Berlekamp and L. R. Welch. Error Correction of Algebraic Block Codes, U.S. Patent 4 633 470, Dec. 1986.
- [CLO05] D. A. Cox, J. Little, and D. O’Shea. *Using Algebraic Geometry*. Springer-Verlag New York, 3rd edition, 2005.
- [CLO07] D. A. Cox, J. Little, and D. O’Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer-Verlag New York, 3rd edition, 2007.
- [DF04] D. S. Dummit and R. M. Foote. *Abstract Algebra*. John Wiley & Sons, 2004.
- [DGP⁺19] S. Di, H. Guo, E. Pershey, M. Snir, and F. Cappello. Characterizing and Understanding HPC Job Failures Over The 2K-Day Life of IBM BlueGene/Q System. In *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 473 – 484, 2019.
- [DHPR19] J.-G. Dumas, J. van der Hoeven, C. Pernet, and D. S. Roche. LU Factorization with Errors. In *Proceedings of ISSAC’19*, pages 131 – 138. ACM, 2019.
- [Dix82] J. D. Dixon. Exact solution of linear equations using p -adic expansions. *Numerische Mathematik*, 40(1):137 – 141, 1982.
- [DPS15] J.-G. Dumas, C. Pernet, and Z. Sultan. Computing the Rank Profile Matrix. In *Proceedings of ISSAC’15*, pages 149 – 156. ACM, 2015.
- [Eis95] D. Eisenbud. *Commutative Algebra: with a View Toward Algebraic Geometry*. Springer-Verlag New York, 1995.
- [Eli57] P. Elias. List Decoding for Noisy Channels. Technical Report 335, Massachusetts Institute of Technology, Cambridge, 1957.
- [FF92] P. Fitzpatrick and J. Flynn. A Gröbner basis technique for Padé approximation. *Journal of Symbolic Computation*, 13(2):133 – 138, 1992.
- [Gao03] S. Gao. A New Algorithm for Decoding Reed-Solomon Codes. In *Communications, Information and Network Security*, pages 55 – 68. Springer US, 2003.
- [GG13] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3rd edition, 2013.

- [GLL⁺17] L. Gąsieniec, C. Levcopoulos, A. Lingas, R. Pagh, and T. Tokuyama. Efficiently Correcting Matrix Products. *Algorithmica*, 79(2):428 – 443, 2017.
- [GLZ19] E. Guerrini, R. Lebreton, and I. Zappatore. Polynomial Linear System Solving with Errors by Simultaneous Polynomial Reconstruction of Interleaved Reed-Solomon Codes. In *Proceedings of ISIT'19*, pages 1542 – 1546. IEEE, 2019.
- [GLZ20a] E. Guerrini, R. Lebreton, and I. Zappatore. Enhancing Simultaneous Rational Function Recovery: adaptive error correction capability and new bounds for applications. arXiv:2003.01793, 2020.
- [GLZ20b] E. Guerrini, R. Lebreton, and I. Zappatore. On the Uniqueness of Simultaneous Rational Function Reconstruction. In *Proceedings of ISSAC'20*. ACM, 2020.
- [Gol49] M. J. E. Golay. Notes on Digital Coding. In *Proceedings of the Institute of Radio Engineers*, page 657, 1949.
- [GRS00] O. Goldreich, D. Ron, and M. Sudan. Chinese remaindering with errors. *IEEE Transactions on Information Theory*, 46(4):1330 – 1338, 2000.
- [GRS19] V. Guruswami, A. Rudra, and M. Sudan. Essential Coding Theory, 2019.
- [HA84] K.-H. Huang and J. A. Abraham. Algorithm-Based Fault Tolerance for Matrix Operations. *IEEE Transactions on Computers*, 33(6):518 – 528, 1984.
- [Hal12] J. I. Hall. Notes in Coding Theory, 2012.
- [Ham50] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell Labs Technical Journal*, 29(2):147 – 160, 1950.
- [HP03] W.C. Huffman and V. Pless. *Fundamentals of error-correcting codes*. Cambridge Univ. Press, Cambridge, 2003.
- [Jus06] J. Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Trans. Inf. Theor.*, 18(5):652 – 656, 2006.
- [JV05] C.-P. Jeannerod and G. Villard. Essentially optimal computation of the inverse of generic polynomial matrices. *Journal of Complexity*, 21(1):72 – 86, 2005.
- [KL97] V. Y. Krachkovsky and Y. X. Lee. Decoding for iterative Reed-Solomon coding schemes. *IEEE Transactions on Magnetics*, 33(5):2740–2742, 1997.
- [KPR⁺10] M. Khonji, C. Pernet, J.-L. Roch, T. Roche, and T. Stalinski. Output-Sensitive Decoding for Redundant Residue Systems. In *Proceedings of ISSAC'10*, page 265 – 272. ACM, 2010.

- [KPSW17] E. Kaltofen, C. Pernet, A. Storjohann, and C. Waddell. Early Termination in Parametric Linear System Solving and Rational Function Vector Recovery with Error Correction. In *Proceedings of ISSAC'17*, pages 237 – 244. ACM, 2017.
- [Kra03] V. Y. Krachkovsky. Reed-Solomon codes for correcting phased error bursts. *IEEE Transactions on Information Theory*, 49(11):2975 – 2984, 2003.
- [KY98] V. Y. Krachkovsky and Yuan Xing Lee. Decoding of parallel Reed-Solomon codes with applications to product and concatenated codes. In *Proceedings of ISIT'98*. IEEE, 1998.
- [LC18] R.-T. Liu and Z.-N. Chen. A Large-Scale Study of Failures on Petascale Supercomputers. *Journal of Computer Science and Technology*, 33(1):24 – 41, 2018.
- [Mas69] J. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15(1):122 – 127, 1969.
- [MC79] R. T. Moenck and J. H. Carter. Approximate algorithms to derive exact solutions to systems of linear equations. In Edward W Ng, editor, *Symbolic and Algebraic Computation*, pages 65 – 73, Berlin, Heidelberg, 1979. Springer Berlin Heidelberg.
- [MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [MS03] T. Mulders and A. Storjohann. On lattice reduction for polynomial matrices. *Journal of Symbolic Computation*, 35(4):377 – 401, 2003.
- [Nei16] V. Neiger. *Bases of relations in one or several variables: fast algorithms and applications*. Ph.d. thesis, École Normale Supérieure de Lyon - University of Waterloo, 2016.
- [OF07] H. O’Keeffe and P. Fitzpatrick. Gröbner basis approach to list decoding of algebraic geometry codes. *Applicable Algebra in Engineering, Communication and Computing*, 18(5):445 – 466, 2007.
- [OS07] Z. Olesh and A. Storjohann. The Vector Rational Function Reconstruction problem. In *Proceedings of the Waterloo Workshop*, pages 137 – 149. World Scientific, 2007.
- [Pag13] R. Pagh. Compressed Matrix Multiplication. *ACM Trans. Comput. Theory*, 5(3), 2013.
- [Per14] C. Pernet. *High Performance and Reliable Algebraic Computing*. Habilitation à diriger des recherches, Université Joseph Fourier, Grenoble 1, 2014.
- [Pop72] V. M. Popov. Invariant Description of Linear, Time-Invariant Controllable Systems. *SIAM Journal on Control*, 10(2):252 – 264, 1972.

- [PR17] S. Puchinger and J. Rosenkilde. Decoding of interleaved Reed-Solomon codes using improved power decoding. In *Proceedings of ISIT'17*, pages 356 – 360. IEEE, 2017.
- [PS07] C. Pernet and A. Storjohann. Faster Algorithms for the Characteristic Polynomial. In *Proceedings of ISSAC'07*, pages 307 – 314, New York, NY, USA, 2007. ACM.
- [PWBJ17] R. Pellikaan, X.W. Wu, S. Bulygin, and R. Jurrius. *Codes, Cryptology and Curves with Computer Algebra*. Cambridge University Press, 1st edition, 2017.
- [Roc18] D. S. Roche. Error Correction in Fast Matrix Multiplication and Inverse. In *Proceedings of ISSAC'18*, page 343 – 350, New York, NY, USA, 2018. Association for Computing Machinery.
- [Rot06] R. Roth. *Introduction to Coding Theory*. Cambridge University Press, USA, 2006.
- [RS60] I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society of Industrial and Applied Mathematics*, 8(2):300 – 304, 1960.
- [RS16] J. Rosenkilde and A. Storjohann. Algorithms for Simultaneous Padé Approximations. In *Proceedings of ISSAC'16*, pages 405 – 412, New York, NY, USA, 2016. Association for Computing Machinery.
- [Sha48] C.E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379 – 423, 1948.
- [Sin64] R. Singleton. Maximum distance q-nary codes. *IEEE Transactions on Information Theory*, 10(2):116 – 118, 1964.
- [SKHN75] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. A method for solving key equation for decoding Goppa codes. *Information and Control*, 27(1):87 – 99, 1975.
- [SSB07] G. Schmidt, V. Sidorenko, and M. Bossert. Enhancing the Correcting Radius of Interleaved Reed-Solomon Decoding using Syndrome Extension Techniques. In *Proceedings of ISIT'07*, pages 1341 – 1345. IEEE, 2007.
- [SSB09] G. Schmidt, V. R. Sidorenko, and M. Bossert. Collaborative Decoding of Interleaved Reed-Solomon Codes and Concatenated Code Designs. *IEEE Transactions on Information Theory*, 55(7):2991 – 3012, 2009.
- [SSB10] G. Schmidt, V. R. Sidorenko, and M. Bossert. Syndrome Decoding of Reed-Solomon Codes Beyond Half the Minimum Distance Based on Shift-Register Synthesis. *IEEE Transactions on Information Theory*, 56(10):5245–5252, Oct 2010.
- [Sto03] A. Storjohann. High-order lifting and integrality certification. *Journal of Symbolic Computation*, 36(3):613 – 648, 2003.

- [Vil97] G. Villard. A study of Coppersmith's block Wiedemann algorithm using matrix polynomials. Rapport de recherche 975-I, IMAG-LMC, 1997.
- [Woz58] J. M. Wozencraft. List Decoding. Technical report, Research Laboratory of Electronics, MIT, 1958.