



Autour de quelques équations fonctionnelles analytiques

Fabienne Naegele

► To cite this version:

Fabienne Naegele. Autour de quelques équations fonctionnelles analytiques. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1995. Français. NNT: . tel-00005056

HAL Id: tel-00005056

<https://theses.hal.science/tel-00005056>

Submitted on 24 Feb 2004

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
Présentée par

Fabienne NAEGELE

pour obtenir le grade de DOCTEUR
de l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

(Arrêté ministériel du 30 mars 1992)

Spécialité: **Mathématiques Appliquées**

Autour de quelques équations fonctionnelles analytiques

Date de soutenance : 15 décembre 1995

Composition du jury :	Président :	Werner BALSER
	Rapporteurs :	Anne DUVAL
		Manuel BRONSTEIN
	Examineurs :	Jean-Pierre RAMIS
		Jean DELLA-DORA

Thèse préparée au sein du **L**aboratoire de **M**odélisation et **C**alcul de Grenoble.

Je tiens ici à remercier très chaleureusement toutes les personnes que j'ai rencontrées au cours des dernières années et qui m'ont soutenue, tant par l'intérêt qu'elles ont manifesté pour mon travail que par leur gentillesse. Je pense notamment à tous les Grenoblois et Strasbourgeois, ainsi qu'à tous les membres du groupe CATHODE.

J'aimerais exprimer plus particulièrement ma gratitude à Jean DELLA-DORA, mon directeur de thèse, ainsi qu'à Jean-Pierre RAMIS, lequel a accepté de co-diriger cette thèse, pour m'avoir proposé un sujet aussi intéressant que varié et pour m'avoir fait confiance. Qu'ils en soient vivement remerciés.

Je suis particulièrement fière de pouvoir remercier Anne DUVAL. Elle a accepté avec gentillesse et sans aucune hésitation de faire un rapport sur ma thèse.

Il en va de même pour Manuel BRONSTEIN. Bien que sollicité de toutes parts, il a toujours accepté de répondre à mes questions et c'est un honneur pour moi qu'il ait fait un rapport sur mon travail.

Quant à Werner BALSER, je n'aurais osé espérer qu'il fasse partie de mon jury. J'en suis particulièrement heureuse.

J'ai également le plaisir de remercier Françoise JUNG. Par ses conseils et sa disponibilité, elle a su me soutenir quand j'en avais besoin.

Mes pensées vont à Frédéric FAUVET qui a montré un intérêt manifeste pour mon travail de recherche et qui a eu la patience de m'éclaircir de nombreux points mathématiques.

Ma gratitude va également à Claire DICRESCENZO, pour sa gentillesse et sa patience. Sans elle, je connaîtrais encore moins bien A^\natural .

Mes plus vifs remerciements vont à Jean THOMANN. Je suis fière de travailler avec lui. Il a toujours montré une patience et une bonne humeur hors du commun.

Table des matières

Introduction	7
 Chapitre 1	
THÉORÈMES D'INDICES POUR LES ÉQUATIONS q -DIFFÉRENCES-DIFFÉRENTIELLES	13
1. Introduction	13
2. Opérateurs à indice	14
3. Espaces Gevrey généralisés	15
4. Théorèmes d'indices et de comparaison	17
4.1. Opérateurs étudiés	17
4.2. Espaces de Banach	18
4.3. Les (ρ, r, μ, ν) -opérateurs	20
4.4. Théorèmes d'indices	22
4.5. Théorème d'indice dans $\mathcal{C}[[x]]$	26
4.6. Théorème d'indice dans $\mathcal{C}\{x\}$	27
4.7. Théorème de comparaison	27
5. Interprétation des théorèmes d'indices	31
5.1. Polygone de Newton	31
5.2. Traduction des théorèmes d'indices en fonction du polygone de Newton	33
6. Equations à coefficients polynomiaux	34
6.1. Dualité topologique	34
6.2. Etude à l'infini	37
6.3. Théorèmes d'indices	38
6.4. Estimations de croissance	39
 Chapitre 2	
THÉORÈMES D'INDICES POUR LES ÉQUATIONS AUX DIFFÉRENCES À COEFFICIENTS POLYNOMIAUX EN q^x	45
1. Introduction	45
2. Préliminaires	46
2.1. La fonction Gamma q -analogue	46
2.2. La q -intégrale	48
3. Transformée q-Mellin formelle	48
3.1. Transformée q -Mellin formelle d'un opérateur	48
3.2. Séries de q -factorielles	49
4. Théorèmes d'indices dans $\tilde{\mathcal{C}}_q[[t]]$	53
4.1. Indice dans $\tilde{\mathcal{C}}_q[[t]]$	53
4.2. Indices dans les espaces q -Gevrey	55
5. Solutions séries de q-factorielles formelles	58
5.1. Polygone de Newton	58

5.2. Caractérisation q -Gevrey des séries de q -factorielles solutions	59
6. Théorèmes d'indices (séries de q-factorielles)	60
6.1. Indices formels	60
6.2. Indices q -Gevrey	63
7. Transformée de q-Laplace formelle	64
7.1. Définitions	64
7.2. Solutions formelles	66

Chapitre 3

UN ALGORITHME DE RESOMMATION DE SÉRIES FORMELLES SOLUTIONS D'ÉQUATIONS DIFFÉRENTIELLES ORDINAIRES LINÉAIRES	69
-------------------------------------------------------------------------------------------------------------	----

1. Introduction	69
2. Résultats théoriques	70
2.1. Notations	70
2.2. La k -sommabilité et la multisommabilité	71
2.3. Méthode des transformées de Laplace itérées (W. Balser)	74
2.4. Polynômes d'Ore	77
3. Outils formels	78
3.1. Définition du concept de série formelle	79
3.2. Passage d'une équation aux différences à une équation différentielle puis à un système différentiel	79
3.3. Scindage d'une série formelle	81
3.4. Transformée de Borel formelle	83
3.5. Transformées de Borel itérées	84
3.6. L'exemple de Ramis-Sibuya	86
4. Prolongement analytique	87
4.1. Transformées de Laplace itérées	87
4.2. Localisation des singularités	88
4.3. Prolongement analytique	88
4.4. Logiciel Compas	89
5. Implantation	90
5.1. Primitives	90
5.2. Contenu algorithmique	97
6. Résultats	102
6.1. Equation d'Euler	102
6.2. Exemple de 2-sommabilité	104
6.3. Exemple de Ramis-Sibuya	105
7. Conclusion	106

Bibliographie	111
----------------------	------------

Annexes	115
A.1. Convergence des séries de factorielles q -analogues	117
A.2. Fichiers "source" de l'algorithme de Balser	121

Introduction

Le sujet de cette thèse concerne l'étude d'équations fonctionnelles analytiques. Plus précisément nous nous intéressons d'une part aux équations linéaires algébriques q -différences-différentielles pour lesquelles nous établissons des théorèmes d'indices ainsi que des estimations de croissance des solutions entières de telles équations. Nous nous attachons d'autre part à l'étude des séries de factorielles q -analogues solutions d'équations aux différences à coefficients polynomiaux en q^x dont nous donnons des estimées q -Gevrey. Enfin nous poursuivons le travail de stage de DEA sur l'implantation d'un algorithme de multisommation des séries formelles solutions d'équations différentielles linéaires algébriques. Les annexes comprennent d'une part quelques résultats sur la convergence des séries de factorielles q -analogues et d'autre part les fichiers "source" de l'algorithme de Balser.

Nous nous proposons à présent de détailler les principaux résultats obtenus.

Le premier chapitre de ce rapport traite des opérateurs mélangeant opérateurs différentiels et opérateurs aux q -différences de la forme

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(\frac{d}{dx}\right)^i \sigma_q^j$$

où les coefficients $a_{i,j}(x)$ sont des séries formelles; σ_q est l'opérateur qui à la série formelle $\sum_{n \geq 0} a_n x^n$ associe la série formelle $\sum_{n \geq 0} a_n q^n x^n$ q étant un complexe de module $0 < |q| < 1$. De tels opérateurs ont été étudiés auparavant notamment par W. Hahn [14] qui a cherché des solutions analytiques de $Ly = 0$ lorsque les coefficients $a_{i,j}(x)$ sont constants.

En considérant les travaux menés par J.-P. Ramis et B. Malgrange concernant les équations différentielles ordinaires [27][29][9] et par J.-P. Bézivin sur les équations aux q -différences [6] nous sommes amenés à définir pour $s, s' \in \mathbb{R}$ l'espace $\mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$) des séries formelles $\sum_{n \geq 0} a_n x^n$ dont les coefficients ont une majoration du type

$$\exists A > 0, \exists C > 0, \forall n \in \mathbb{N}, |a_n| < C|q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} A^n$$

(resp. $\forall A > 0, \exists C > 0, \forall n \in \mathbb{N}, |a_n| < C|q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} A^n$).

La méthode de filtration par des espaces de Banach [29] et la recherche des opérateurs prépondérants au sens de la compacité [29][9][6] se transposent naturellement dans notre cas. Nous établissons ainsi le théorème :

Théorème 4.4.8 : Soit $L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(\frac{d}{dx}\right)^i \sigma_q^j$, avec $a_{i,j}(x) = \sum_{k \geq 0} \alpha_{i,j,k} x^k \in \mathcal{C}\{x\}$ et q un complexe, $0 < |q| < 1$. Soient s et $s' \in \mathbb{R}$ tels que $s > 0$ et $s' \in \mathbb{R}$, ou $s = 0$ et $s' \geq 0$. Soient $p_s^0(L) = \inf_{\alpha_{i,j,k} \neq 0} (j + (k-i)s)$ et $M(s) = \{(i, j, k) | \alpha_{i,j,k} \neq 0 \text{ et } j + (k-i)s = p_s^0(L)\}$. Soient $v_{s,s'}(L) = \inf_{(i,j,k) \in M(s)} ((k-i)s' - i)$ et $N(s, s') = \{(i, j, k) \in M(s) | (k-i)s' - i = v_{s,s'}(L)\}$.

- Si $s > 0$ et $s' \in \mathbb{R}$ ou si $s = 0$ et $s' \geq 0$ alors L est un opérateur à indice dans $\mathcal{C}[[x]]_{q,s,s'}$, d'indice égal à :

$$\chi(L, \mathcal{C}[[x]]_{q,s,s'}) = -\inf\{k - i / \exists j, (i, j, k) \in N(s, s')\}$$

- Si $s > 0$ et $s' \in \mathbb{R}$ ou si $s = 0$ et $s' > 0$ alors L est un opérateur à indice dans $\mathcal{C}[[x]]_{(q,s,s')}$, d'indice égal à :

$$\chi(L, \mathcal{C}[[x]]_{(q,s,s')}) = -\sup\{k - i / \exists j, (i, j, k) \in N(s, s')\}$$

Ces indices peuvent se lire directement sur un polygone de Newton associé à l'opérateur, dont les pentes correspondent d'une part aux réels s pour lesquels il existe (i_1, j_1, k_1) et (i_2, j_2, k_2) dans $M(s)$ avec $k_1 - i_1 \neq k_2 - i_2$ (nous dirons qu'alors s est une valeur exceptionnelle) d'autre part aux réels s' associés à une valeur exceptionnelle s tels qu'il existe (i_1, j_1, k_1) et (i_2, j_2, k_2) dans $N(s, s')$ avec $k_1 - i_1 \neq k_2 - i_2$ (s' est dit exceptionnel relativement à s).

Si les séries $a_{i,j}(x)$ sont convergentes nous établissons le théorème suivant :

Théorème 4.7.9 : Soit $L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) (\frac{d}{dx})^i \sigma_q^j$, avec $a_{i,j}(x) \in \mathcal{C}\{x\}$. Soient $\hat{f} \in \mathcal{C}[[x]]$ et $g \in \mathcal{C}\{x\}$ tels que $L(\hat{f}) = g$. Alors :

- Soit $\hat{f} \in \mathcal{C}\{x\}$,
- Soit il existe un unique réel $s' > 0$ tel que $\hat{f} \in \mathcal{C}[[x]]_{q,0,s'}$ et $\hat{f} \notin \mathcal{C}[[x]]_{(q,0,s')}$; on dit que \hat{f} est Gevrey d'ordre exact s' ,
- Soit il existe un unique réel $s > 0$ et un unique réel $s' \in \mathbb{R}$ tels que $\hat{f} \in \mathcal{C}[[x]]_{q,s,s'}$ et $\hat{f} \notin \mathcal{C}[[x]]_{(q,s,s')}$; on dit alors que \hat{f} est q -Gevrey d'ordres exacts s et s' .

Les théorèmes d'indices pour les opérateurs à coefficients polynomiaux s'étendent par dualité et par passage à l'infini.

Théorèmes 6.3.2 et 6.4.2 : Soit L un opérateur à coefficients polynomiaux

$$L = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \alpha_{i,j,k} x^k (\frac{d}{dx})^i \sigma_q^j$$

avec $0 < |q| < 1$. Soient $f \in A(\mathcal{C})$ une fonction entière et g un polynôme tels que $L(f) = g$. Alors :

- Soit f est un polynôme,
- Soit il existe un unique réel $s' < 0$ tel que $f \in \mathcal{C}[[x]]_{q,0,s'}$ et $f \notin \mathcal{C}[[x]]_{(q,0,s')}$. De plus, il existe $C, \alpha > 0$ tels que $|f(x)| < C \exp(\alpha r^{-1/s'})$ ($r = |x|$),
- Soit il existe un unique réel $s < 0$ et un unique réel $s' \in \mathbb{R}$ tels que $f \in \mathcal{C}[[x]]_{q,s,s'}$ et $f \notin \mathcal{C}[[x]]_{(q,s,s')}$. Il existe alors $C, \alpha > 0$ tels que $|f(x)| < C \exp(\frac{k_1}{2} \ln^2(r) + k_1 s' \ln(r) \ln |\ln(r)| + \alpha \ln(r))$ ($r = |x|$).

Dans le second chapitre nous étudions les équations aux différences dont les coefficients sont des polynômes en q^x . Ces équations apparaissent dans le travail de W. Hahn [14].

Considérons la transformée intégrale formelle $f(x) = \int_C t^{x-1} F(t) dt$. Formellement $\tau f(x) = f(x+1) = \int_C t^{x-1} \times t F(t) dt$, $xf(x) = \int_C xt^{x-1} F(t) dt = \int_C t^{x-1} (-t \frac{d}{dt}) F(t) dt$ et $q^x f(x) = \int_C t^{x-1} \times \sigma_p F(t) dt$.

A. Barkatou et A. Duval [5] ont utilisé l'isomorphisme $\mathcal{C}[\tau, x] \rightarrow \mathcal{C}[t, -t \frac{d}{dt}]$ défini par $\tau \mapsto t$ et $x \mapsto -t \frac{d}{dt}$ et les théorèmes d'indices établis dans le cas différentiel pour obtenir des théorèmes d'indices pour des opérateurs de $\mathcal{C}[\tau, x]$ agissant sur les séries formelles de factorielles.

Nous considérons la transformée de Mellin q -analogue c'est-à-dire l'isomorphisme $\mathcal{C}[\tau, q^x] \rightarrow \mathcal{C}[t, \sigma_p]$ défini par $\tau \mapsto t$ et $q^x \mapsto \sigma_p$. Le bon espace des séries formelles stables par l'action de $\mathcal{C}[\tau, q^x]$ est celui des séries q -factorielles généralisées $\sum_{n \geq 0} \frac{\Gamma_q(x)}{\Gamma_q(x+\rho+n+1)}$ où $\rho \in \mathcal{C}$ (Γ_q est la fonction Gamma q -analogue). La transformée q -Mellin formelle que nous avons été amenés à définir pour une telle série est :

Définition 3.2.4 : La transformée q -Mellin formelle d'une série formelle de q -factorielles généralisées est la série formelle

$$M_q(\hat{f}_q)(t) = \sum_{n \geq 0} \frac{a_n}{\Gamma_q(n+1+\rho)} \frac{(tq; q)_\infty}{(tq^{n+1+\rho}; q)_\infty} = \frac{(tq; q)_\infty}{(tq^{\rho+1}; q)_\infty} \sum_{n \geq 0} \frac{a_n}{\Gamma_q(n+1+\rho)} (tq^{\rho+1}; q)_n$$

où $(tq^{\rho+1}; q)_n = (1 - tq^{\rho+1})(1 - tq^{\rho+2}) \dots (1 - tq^{\rho+n})$ et $(a; q)_\infty = (1 - a)(1 - aq)(1 - aq^2) \dots$ pour $|q| < 1$.

Nous vérifions que

Proposition 3.2.6 : Si \hat{f}_q est une série formelle de q -factorielles généralisées alors pour tout opérateur $\Delta_q \in \mathcal{C}[q^x, \tau]$, $M_q(\Delta_q(\hat{f}_q)) = M_q(\Delta_q)M_q(\hat{f}_q)$.

Cette propriété de commutativité nous permet de nous ramener à des équations polynomiales aux p -différences agissant sur les séries formelles en $(tq; q)_n$ notées $\tilde{\mathcal{C}}_q[[t]]$ pour lesquels nous établissons des théorèmes d'indices par l'isomorphisme $\tilde{\mathcal{C}}_q[[t]] \rightarrow \mathcal{C}[[v]]$ qui à $(tq; q)_n$ associe v^n et par l'isomorphisme $[t, \sigma_p] \rightarrow [v, v^{-1}, \sigma_p]$ défini par $t \mapsto p(1-v)\sigma_p$ et $\sigma_p \mapsto \sigma_p + \frac{1}{v} - \frac{1}{v}\sigma_p$.

Nous pouvons également définir un polygone de Newton associé à un opérateur de $\mathcal{C}[\tau, q^x]$. Nous dirons que la série de q -factorielles $\sum_{n \geq 1} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)}$ est q -Gevrey d'ordre $s \in \mathbb{R}$ s'il existe $C, A > 0$ tels que $\forall n \in \mathbb{N}$, $|a_{n+1}| < C \Gamma_q(n+1) |q|^{-\frac{sn(n+1)}{2}} A^n$.

Théorème 5.2.3 : Soient $\Delta_q \in \mathcal{C}[q^x, \tau]$ et \hat{g}_q une série de q -factorielles q -Gevrey d'ordre 0. Soit $\hat{f}_q = \sum_{n \geq 1} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)}$ solution de $\Delta_q \hat{f}_q = \hat{g}_q$. Alors \hat{f}_q est q -Gevrey d'ordre 0 ou il existe un unique réel $s > 0$ tel que \hat{f}_q est q -Gevrey d'ordre s optimal (s est l'inverse de l'une des pentes du polygone de Newton associé à Δ_q).

Les théorèmes d'indices que nous obtenons sont des q -analogues de ceux établis par A. Barkatou et A. Duval.

Le troisième chapitre décrit un algorithme de resommation des séries formelles solutions d'équations différentielles linéaires ordinaires au voisinage d'une singularité irrégulière Γ supposée à l'origine. Nous transformons la méthode des transformées de Borel-Laplace itérées proposée par W. Balser en un algorithme formel-numérique de calcul effectif de la somme de séries multisommables.

L'algorithme repose sur les trois théorèmes suivants.

Théorème 2.3.3 (W. Balser) : Soit \hat{f} une série (k_1, \dots, k_r) -sommable. Soit d une direction non singulière. Soient $\kappa_1, \dots, \kappa_r > 0$ définis par $1/\kappa_1 = 1/k_1$ et $1/\kappa_j = 1/k_j - 1/k_{j-1}$, $j = 2, \dots, r$. La somme de \hat{f} dans la direction d est égale à

$$L_{\kappa_1} \cdot_d \dots L_{\kappa_r} \cdot_d S \hat{B}_{\kappa_r} \dots \hat{B}_{\kappa_1} \hat{f}$$

où \cdot_d est le prolongement analytique le long de d , S est la somme de la série formelle convergente, \hat{B}_κ est la transformée de Borel formelle d'ordre κ ($\hat{B}_\kappa(x^\lambda)(\xi) = \frac{\xi^{\lambda-\kappa}}{\Gamma(\frac{\lambda}{\kappa})}$) et L_κ est la transformée de Laplace d'ordre κ ($L_\kappa(f)(x) = \int_d f(t) \exp(-(\frac{t}{x})^\kappa) d(t^\kappa)$).

Définition 2.3.4 : Soient $\kappa_1, \dots, \kappa_r > 0$. Soient $k_1 > \dots > k_r$ définis successivement par $1/k_1 = 1/\kappa_1$ et

$$\frac{1}{k_l} = \frac{1}{\kappa_l} + \frac{1}{k_{l-1}}, \quad l = 2, \dots, r$$

Une série formelle est $(\kappa_1, \dots, \kappa_r) - iL$ sommable si elle est (k_1, \dots, k_r) -sommable.

Théorème 3.3.1 : Supposons que $\hat{f}(x)$ est $(\kappa_1, \dots, \kappa_r) - iL$ sommable. Soient $\alpha \in \mathbb{N}^*$ et $\hat{f}_0(x), \dots, \hat{f}_{\alpha-1}(x)$ les α sous-séries.

Alors chaque sous-série $\hat{f}_q(x)$ est $(\kappa_1/\alpha, \dots, \kappa_r/\alpha) - iL$ sommable. De plus, si $d \in \mathbb{R}$ est tel que $d, d + \theta, \dots, d + (\alpha - 1)\theta$ sont des directions non singulières pour $\hat{f}(x)$ ($\theta = 2\pi/\alpha$), alors αd est une direction non singulière pour les sous-séries. Dans un secteur bissecté par d , la somme $f(x)$ de $\hat{f}(x)$ est égale à

$$f(x) = (a_0 + f_0(x^\alpha)) + x(a_1 + f_1(x^\alpha)) + \dots + x^{\alpha-1}(a_{\alpha-1} + f_{\alpha-1}(x^\alpha))$$

Théorème 3.4.1 : Soient $\kappa_1, \dots, \kappa_r > 0$ avec $\kappa_1 = 1/\beta_1$, $\beta_1 \in \mathbb{N}$.

Si $\hat{g}(x)$ est $(\kappa_1, \dots, \kappa_r) - iL$ sommable dans la direction d alors la transformée de Borel formelle $\hat{B}_{1/\beta_1}(\hat{g})$ est $(\kappa_2, \dots, \kappa_r) - iL$ sommable dans la direction d .

Nous détaillons l'algorithme pour une série $\hat{f}(x)$ $(\kappa_1, \kappa_2) - iL$ sommable Γ avec $\kappa_1 = \frac{\alpha_1}{\beta_1}$ et $\frac{\kappa_2}{\alpha_1} = \frac{\alpha_2}{\beta_2}$. Les directions d de resommation autorisées par l'algorithme sont les directions non

singulières de $\hat{f}(x)$ telles qu'en faisant $\alpha_1\alpha_2$ tours autour de l'origine on ne retombe pas sur une direction singulière c'est-à-dire : $d \neq d_i + \frac{2\pi}{\alpha_1\alpha_2}l$ pour tout $l \in \mathbb{Z}$ et pour toute direction singulière d_i .

Pour calculer la somme de $\hat{f}(x)$ en un point ou plus généralement le long d'un chemin nous scindons la série formelle en α_1 sous-séries dont nous calculons les transformées de Borel formelle d'ordre $\frac{1}{\beta_1}$. Ces transformées de Borel $\hat{\phi}_0(x), \dots, \hat{\phi}_{\alpha_1-1}(x)$ sont $(\frac{\kappa_2}{\alpha_1}) - iL$ sommables. Nous avons à calculer le prolongement de leur somme le long d'un chemin. Près de l'origine en général singularité irrégulière nous calculons la somme en recommençant le processus précédent. Nous scindons en α_2 sous-séries chaque transformée de Borel et prenons les transformées de Borel d'ordre $\frac{1}{\beta_2}$ lesquelles sont convergentes. Nous calculons leur prolongement analytique le long d'un chemin et par une formule de quadrature nous obtenons le prolongement des sommes de $\hat{\phi}_0(x), \dots, \hat{\phi}_{\alpha_1-1}(x)$ au voisinage de l'origine. Le prolongement sur le chemin peut ensuite se faire par une méthode de Runge-Kutta. Nous pouvons alors calculer la somme de la série initiale.

La participation au projet européen CATHODE (Computer Algebra Tools for Handling Ordinary Differential Equations) nous a permis de dégager les primitives informatiques nécessaires à la réalisation de cet algorithme en les intégrant aux autres primitives définies dans ce groupe de travail en particulier par M. Bronstein et par C. Dacresenzo et F. Jung.

Chapitre 1

Théorèmes d'indices pour les équations q -différences-différentielles

1. Introduction

Les équations fonctionnelles q -différences-différentielles *mélangeant* opérateurs différentiels et opérateurs aux q -différences ont déjà intéressé quelques auteursΓnotamment T. Kato et J.-B. McLeod [5] qui se sont penchés sur une équation fonctionnelle de la forme $y'(x) = ay(\lambda x) + by(x)$ provenant d'un problème industriel ($a \in \mathbb{C}, b \in \mathbb{R}$ et $\lambda \in \mathbb{R}^+$); K. Mahler [7] a quant à lui considéré les équations du type $\frac{f(z+w)-f(z)}{w} = f(qz)$ où $w \neq 0$ et $0 < q < 1$ sont deux constantes réelles. Enfin W. Hahn [4] s'est intéressé aux solutions analytiques d'une équation fonctionnelle à coefficients constants de la forme $\sum_{j=0}^r \sum_{k=0}^s a_{j,k} f^{(j)}(zq^k) = 0$ où q est un réel ($0 < q < 1$); il se ramène par des transformations intégrales à l'étude d'une équation aux différences à coefficients polynomiaux.

En faitΓ nous n'aborderons pas le problème de la même façon que W. Hahn. Il se trouve en effet que les études analogues menées par J.-P. Ramis et B. Malgrange sur les équations différentielles ordinaires [91018]Γet par J.-P. Bézivin [1] sur les équations aux q -différences (et en réalité sur des équations fonctionnelles plus générales) se transposent assez naturellement dans notre cas.

Les opérateurs que nous examinerons dans cet article seront de la forme :

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(\frac{d}{dx}\right)^i \sigma_q^j \quad (1)$$

où les $a_{i,j}(x)$ sont des séries formellesΓet où σ_q est l'opérateur qui à une série formelle $\sum_{n \geq 0} a_n x^n$ associe la série formelle $\sum_{n \geq 0} a_n q^n x^n$. Nous nous limitons à q complexeΓ $0 < |q| < 1$ (le cas $|q| > 1$ s'y ramène aisément). En multipliant par une puissance convenable de x Γnous nous ramènerons en fait à des opérateurs du type

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(x \frac{d}{dx}\right)^i \sigma_q^j \quad (2)$$

où les $a_{i,j}(x)$ sont des séries formelles; cette écriture est intéressante dans la mesure où les opérateurs $x \frac{d}{dx}$ et σ_q commutent. Nous établirons les théorèmes d'indices pour ces derniers opérateursΓla généralisation aux opérateurs de la forme (1) étant immédiate.

Nous ferons dans un premier temps des rappels sur la théorie des opérateurs à indice en insistant sur les théorèmes essentiels utilisés tout au long de notre étude. Puis nous définirons la

notion de divers espaces Gevrey généralisés et nous y établirons des théorèmes d'indices et de comparaison dont on donnera une interprétation en termes de polygones de Newton. Comme dans [11] l'étude des indices pour les opérateurs à coefficients polynomiaux se fait par dualité et par passage à l'infini.

2. Opérateurs à indice

Nous ne rappelons ici que les résultats essentiels auxquels nous ferons référence dans cet article. Pour davantage de compléments se référer par exemple à [310].

Définition 2.1 : Soient E et F deux espaces vectoriels. On dit qu'une application linéaire $u : E \rightarrow F$ est à indice si son noyau et son conoyau sont de dimension finie. L'indice de u est alors par définition : $\chi(u) = \dim \text{Ker}(u) - \dim \text{Coker}(u)$.

Proposition 2.2 : Soient $u : E \rightarrow F$ et $v : F \rightarrow G$ deux opérateurs à indice. Alors $v \circ u : E \rightarrow G$ est un opérateur à indice, d'indice égal à : $\chi(v \circ u) = \chi(v) + \chi(u)$.

Théorème 2.3 : Soient E et F deux espaces de Banach et $u : E \rightarrow F$ linéaire continue.

Pour que u soit à indice, il faut et il suffit que son transposé u^* soit à indice. Si c'est le cas, alors $\chi(u^*) = -\chi(u)$. Ce théorème reste vrai si E et F sont tous deux du type FS ou DFS.

Théorème 2.4 : Soient E et F deux espaces de Banach et $u : E \rightarrow F$ linéaire continue. On suppose que u est à indice. Soit $K : E \rightarrow F$ un opérateur linéaire continu et compact. Alors $u + K$ est un opérateur à indice et $\chi(u + K) = \chi(u)$.

Remarque : Ce théorème est la clé de notre étude. Le but de tout notre travail est d'éliminer certains termes de l'opérateur étudié par des arguments de *compacité*. On ne retiendra que les termes de *poids* le plus important.

Lemme 2.5 : Soient E_1 et F_1 deux espaces de Banach ou Fréchet ou DFS (en tous les cas de même type). Soient E_2 et F_2 donnés avec les mêmes conditions. Soit le diagramme commutatif d'applications linéaires continues :

$$\begin{array}{ccc} E_1 & \xrightarrow{u_1} & F_1 \\ v \downarrow & & \downarrow w \\ E_2 & \xrightarrow{u_2} & F_2 \end{array}$$

On suppose que u_1 et u_2 sont à indice, que v est injective et que w est injective et d'image dense. Alors on a une suite exacte

$$0 \rightarrow \text{Ker } \bar{u}_2 \rightarrow \frac{E_2}{E_1} \rightarrow \frac{F_2}{F_1} \rightarrow 0 \quad \text{et} \quad \dim \text{Ker}(\bar{u}_2) = \chi(u_2) - \chi(u_1).$$

3. Espaces Gevrey généralisés

En considérant l'opérateur $D = x^2 \frac{d}{dx} + 1$ et en résolvant l'équation $Df = x$ on obtient comme solution formelle la série : $\hat{f}(x) = \sum_{n \geq 0} (-1)^n n! x^{n+1}$. Cette série diverge mais on contrôle la croissance de ses coefficients.

Rappelons l'un des résultats fondamentaux concernant les solutions formelles des équations différentielles ordinaires linéaires analytiques [9][10] :

Théorème 3.1 : Soit $D \in \mathcal{C}\{x\}[\frac{d}{dx}]$. Soient $\hat{f}(x) = \sum_{n \geq 0} a_n x^n$ et $g \in \mathcal{C}\{x\}$ tels que $D\hat{f} = g$. Alors $\hat{f} \in \mathcal{C}\{x\}$ ou il existe un unique réel $s > 0$ tel que $\hat{f} \in \mathcal{C}[[x]]_s$ et $\hat{f} \notin \mathcal{C}[[x]]_{(s)}$. De plus, s est rationnel et est l'inverse de l'une des pentes strictement positives du polygone de Newton associé à l'opérateur D .

Dans cet énoncé $\mathcal{C}[[x]]_s = \{\hat{f}(x) = \sum_{n \geq 0} a_n x^n / \exists A > 0, \exists C > 0, \forall n, |a_n| < C(n!)^s A^n\}$ et $\mathcal{C}[[x]]_{(s)} = \{\hat{f}(x) = \sum_{n \geq 0} a_n x^n / \forall A > 0, \exists C_A > 0, \forall n, |a_n| < C_A(n!)^s A^n\}$ sont respectivement l'espace des séries Gevrey-Roumieu d'ordre s et l'espace des séries Gevrey-Beurling d'ordre s .

On rappelle que $\mathcal{C}\{x\}$ désigne l'espace des séries ayant un rayon de convergence non nul.

Remarque : L'ordre s est décalé par rapport à celui utilisé par J.-P. Ramis.

Dans le cas des équations aux q -différences J.-P. Bézivin a montré que la croissance à l'infini des coefficients des séries formelles solutions de telles équations est également contrôlée [1] :

Théorème 3.2 : Soit $L = \sum_{i=0}^t P_i(x) \psi(xq^{k_i})$ où les $P_i(x)$ sont des séries convergentes à l'origine, où q est une complexe ($0 < |q| < 1$), et où $k_0 < k_1 < \dots < k_t$ sont des réels. Soient $\hat{\psi} \in \mathcal{C}[[x]]$ et $\theta \in \mathcal{C}\{x\}$ tels que $L\hat{\psi} = \theta$. Alors $\hat{\psi} \in \mathcal{C}\{x\}$ ou il existe un unique réel positif s tel que $\hat{\psi} \in \mathcal{C}[[x]]_{q,s}$ et $\hat{\psi} \notin \mathcal{C}[[x]]_{q,(s)}$. De plus, s fait partie d'un ensemble fini que l'on peut déterminer à partir du q -polygone de Newton associé à L .

Dans ce cas $\mathcal{C}[[x]]_{q,s} = \{\hat{\psi} = \sum_{n \geq 0} a_n x^n / \sum_{n \geq 0} |q^{\frac{sn(n+1)}{2}} a_n| x^n \in \mathcal{C}\{x\}\}$ est l'espace q -Gevrey d'ordre s ; $\mathcal{C}[[x]]_{q,(s)} = \{\hat{\psi} = \sum_{n \geq 0} a_n x^n / \sum_{n \geq 0} |q^{\frac{sn(n+1)}{2}} a_n| x^n \in A(\mathcal{C})\}$ est l'espace q -Gevrey-Beurling d'ordre s ($A(\mathcal{C})$ désigne les séries entières).

Nous verrons que si une série formelle $\hat{f}(x)$ est solution d'une équation du type

$$L\hat{f}(x) = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(x \frac{d}{dx}\right)^i \hat{f}(q^j x) = g(x)$$

ou plus généralement du type

$$L\hat{f}(x) = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(\frac{d}{dx}\right)^i \hat{f}(q^j x) = g(x)$$

où les coefficients et le second membre sont des séries convergentes. Alors cette série a un comportement asymptotique que l'on peut contrôler. On a donc besoin d'introduire de nouveaux espaces que l'on définit par analogie aux espaces précédents.

On rappelle que $q \in \mathbb{C}$, $0 < |q| < 1$.

Définition 3.3 : Soit $\hat{f}(x) = \sum_{n \geq 0} a_n x^n$ une série formelle. Soient $s, s' \in \mathbb{R}$ et $A > 0$. S'il existe $C > 0$ tel que, pour tout $n \in \mathbb{N}$

$$|a_n| < C |q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} A^n$$

on dira que \hat{f} est q -Gevrey d'ordres s et s' , et d'ordre précisé (s, s', A) .

Notation : $\mathcal{C}[[x]]_{q,s,s',A}$.

On note :

$$\mathcal{C}[[x]]_{q,s,s'} = \bigcup_{A>0} \mathcal{C}[[x]]_{q,s,s',A}$$

$$\mathcal{C}[[x]]_{(q,s,s')} = \bigcap_{A>0} \mathcal{C}[[x]]_{q,s,s',A}$$

Propriété 3.4 : Soient $s, s' \in \mathbb{R}$.

Si $\hat{f} \in \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$), alors \hat{f}' et $\sigma_q \hat{f} \in \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$).

Propriété 3.5 : Soient $s, s' \in \mathbb{R}$.

Si \hat{f} et $\hat{g} \in \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$), alors $\hat{f} + \hat{g} \in \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$).

Remarque : L'espace $\mathcal{C}[[x]]_{q,s,0}$ est ce que J.-P. Bézivin note $\mathcal{C}[[x]]_{q,s}$ l'espace des séries q -Gevrey; $\mathcal{C}[[x]]_{(q,s,0)}$ est l'espace des séries q -Gevrey-Beurling noté par J.-P. Bézivin $\mathcal{C}[[x]]_{(q,s)}$. Enfin l'espace $\mathcal{C}[[x]]_{q,0,s'}$ est à rapprocher de l'espace $\mathcal{C}[[x]]_{s'}$ défini par J.-P. Ramis (attention au décalage des indices).

Propriétés 3.6 : Soient s_0, s, s_1 et $s'_0, s', s'_1 \in \mathbb{R}$.

Si $s'_0 \leq s'_1$ alors $\mathcal{C}[[x]]_{q,s,s'_0} \subset \mathcal{C}[[x]]_{q,s,s'_1}$ et $\mathcal{C}[[x]]_{(q,s,s'_0)} \subset \mathcal{C}[[x]]_{(q,s,s'_1)}$.

Si $s_0 \leq s_1$ alors $\mathcal{C}[[x]]_{q,s_0,s'} \subset \mathcal{C}[[x]]_{q,s_1,s'}$ et $\mathcal{C}[[x]]_{(q,s_0,s')} \subset \mathcal{C}[[x]]_{(q,s_1,s')}$.

Propriétés 3.7 : Soient $s'_0 \leq s'_1$ et $s_0 \leq s_1$. Alors :

$$\begin{aligned} \mathcal{C}[[x]]_{q,s_0,s'_0} &\subset \mathcal{C}[[x]]_{q,s_0,s'_1} \subset \mathcal{C}[[x]]_{q,s_1,s'_1} \\ \mathcal{C}[[x]]_{q,s_0,s'_0} &\subset \mathcal{C}[[x]]_{q,s_1,s'_0} \subset \mathcal{C}[[x]]_{q,s_1,s'_1} \end{aligned}$$

(de même pour les espaces Beurling).

Propriété 3.8 : $\mathcal{C}[[x]]_{(q,s,s')} \subset \mathcal{C}[[x]]_{q,s,s'}$

Propriétés 3.9 : Si $s_0 < s_1$ alors $\mathcal{C}[[x]]_{q,s_0,s'} \subset \mathcal{C}[[x]]_{(q,s_1,s')}$
 Si $s'_0 < s'_1$ alors $\mathcal{C}[[x]]_{q,s,s'_0} \subset \mathcal{C}[[x]]_{(q,s,s'_1)}$
 Si $s_0 < s_1$ alors $\forall s'_0, \forall s'_1$ $\mathcal{C}[[x]]_{q,s_0,s'_0} \subset \mathcal{C}[[x]]_{(q,s_1,s'_1)}$

Toutes ces propriétés s'établissent simplement; les détails sont laissés au lecteur.

4. Théorèmes d'indices et de comparaison

Dans tout ce chapitre nous supposons que s est positif ou nul et que q est de module $0 < |q| < 1$.

4.1. Opérateurs étudiés

Soit un opérateur mélange d'opérateurs différentiels et aux q -différences de la forme :

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(\frac{d}{dx}\right)^i \sigma_q^j \quad (1)$$

où les $a_{i,j}(x)$ sont des séries formelles et avec $0 < |q| < 1$. En multipliant par l'opérateur x^I (qui est à indice d'indice $-I$ dans tous les espaces considérés) on se ramène à l'étude d'un opérateur de la forme :

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(x \frac{d}{dx}\right)^i \sigma_q^j \quad (2)$$

Propriété 4.1.1 : Soient $s, s' \in \mathbb{R}$ avec $s \geq 0$.

Si \hat{f} et $\hat{g} \in \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$), alors $\hat{f} \times \hat{g} \in \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$).

Démonstration :

Soient $\hat{f} = \sum_{n \geq 0} a_n x^n$ et $\hat{g} = \sum_{n \geq 0} b_n x^n$ deux séries formelles. Le coefficient générique de la série produit est égal à : $c_n = \sum_{k=0}^n a_k b_{n-k}$.

Si $|a_k| < C|q|^{-\frac{sk(k+1)}{2}} (k!)^{s'} A^k$ et si $|b_{n-k}| < D|q|^{-\frac{s(n-k)(n-k+1)}{2}} (n-k)!^{s'} B^{n-k}$ alors :

$$|c_n| < CD|q|^{-\frac{sn(n+1)}{2}} \sum_{k=0}^n |q|^{-sk(k-n)} (k!)^{s'} (n-k)!^{s'} A^k B^{n-k}$$

On majore $|q|^{-sk(k-n)}$ par 1 ($|q| < 1$ et $s \geq 0$). Etudions le terme $u_k = k!(n-k)!$ pour $k = 0 \dots n$. On vérifie facilement que :

- Si $0 \leq k \leq \lfloor \frac{n-1}{2} \rfloor$: $u_{\lfloor \frac{n-1}{2} \rfloor} \leq u_k \leq u_0 = n!$

- Si $[\frac{n-1}{2}] \leq k \leq n : u_{[\frac{n-1}{2}]} \leq u_k \leq u_n = n!$

Deux cas se présentent :

1. Si $s' \geq 0$ alors $u_k^{s'} \leq (n!)^{s'}$ et $|\sum_{k=0}^n a_k b_{n-k}| \leq CD|q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} (A+B)^n$.
2. Si $s' < 0$ alors $u_k^{s'} \leq u_{[\frac{n-1}{2}]}^{s'}$ et $|\sum_{k=0}^n a_k b_{n-k}| \leq CD|q|^{-\frac{sn(n+1)}{2}} u_{[\frac{n-1}{2}]}^{s'} (A+B)^n$. Comme $n!$ est équivalent à $n^n e^{-n} \sqrt{2\pi n} \Gamma$ il existe une constante \tilde{C} telle que le second membre est majoré par : $\tilde{C}|q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} n^{\frac{s'}{2}} (2^{-s'} (A+B))^n$. Comme $s' < 0$ le terme $n^{\frac{s'}{2}}$ est majoré par 1.

Définitions 4.1.2 : On note $\mathbf{L}_{q,s,s'}$ (resp. $\mathbf{L}_{(q,s,s')}$) l'algèbre des opérateurs de cette forme avec $a_{i,j}(x) \in \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$). Si $a_{i,j}(x) \in \mathcal{C}[[x]]$, l'algèbre est notée $\hat{\mathbf{L}}$. Enfin, si les coefficients sont des séries convergentes, la notation correspondante est \mathbf{L} .

Propriétés 4.1.3 : Soient $0 \leq s_0 \leq s_1$ et $s'_0 \leq s'_1$. Tout opérateur $L \in \mathbf{L}_{q,s_0,s'_0}$ opère dans $\mathcal{C}[[x]]_{q,s_1,s'_1}$. Tout opérateur $L \in \mathbf{L}_{(q,s_0,s'_0)}$ opère dans $\mathcal{C}[[x]]_{(q,s_1,s'_1)}$ et dans $\mathcal{C}[[x]]_{(q,s_1,s'_1)}$.

Propriétés 4.1.4 : Si $0 \leq s_0 < s_1$ alors $L \in \mathbf{L}_{q,s_0,s'}$ opère sur $\mathcal{C}[[x]]_{(q,s_1,s')}$.
Si $s'_0 < s'_1$ alors $L \in \mathbf{L}_{q,s,s'_0}$ opère sur $\mathcal{C}[[x]]_{(q,s,s'_1)}$.

Toutes ces propriétés découlent de celles établies en 3.

4.2. Espaces de Banach

Pour le calcul des indices éventuels nous allons nous ramener à l'étude des opérateurs dans des espaces de Banach afin de pouvoir appliquer des théorèmes connus.

Soient s et $s' \in \mathbb{R}$ $\rho > 0$ et $\mu \in \mathbb{R}$.

On définit l'application $\psi_{s,s',\rho,\mu} : \mathcal{C}[[x]] \longrightarrow \mathcal{C}[[x]]$ avec

$$\sum_{n \geq 0} a_n x^n \longmapsto \sum_{n \geq 0} b_n x^n$$

$$b_n = |q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} \rho^{-n} (1+n)^{-\mu} a_n.$$

On identifie l'espace des suites $\mathcal{C}^{\mathbb{N}}$ à l'espace des séries formelles [8].

On note $\ell_{s,s',\rho,\mu}^1 = \psi_{s,s',\rho,\mu}(\ell^1(\mathcal{C}))$. On définit sur $\ell_{s,s',\rho,\mu}^1$ une norme. Pour $b = \{b_n\} \in \ell_{s,s',\rho,\mu}^1$

$$\|b\|_{s,s',\rho,\mu} = \sum_{n=0}^{\infty} \frac{|b_n|}{|q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} \rho^{-n} (1+n)^{-\mu}} < \infty$$

Muni de cette norme $\ell_{s,s',\rho,\mu}^1$ est un espace de Banach.

Lemme 4.2.1 : Soient s, s' et $\mu \in \mathbb{R}$.

Si $\rho_1 > \rho_2 > 0$, l'injection naturelle $\ell_{s,s',\rho_1,\mu}^1 \xrightarrow{v} \ell_{s,s',\rho_2,\mu}^1$ est \mathcal{C} -linéaire continue nucléaire (donc compacte) et d'image dense.

Démonstration : On pose $c_n = (\frac{\rho_2}{\rho_1})^n$ et on note f_n la forme linéaire sur $\ell_{s,s',\rho_1,\mu}^1$ définie par

$$f_n(b) = \frac{b_n}{|q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} \rho_1^{-n} (1+n)^{-\mu}}$$

Soit y_n la suite : $y_k^n = 0$ pour $k \neq n$ et $y_n^n = |q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} \rho_2^{-n} (1+n)^{-\mu}$. Alors $\sup_{n \geq 0} \|f_n\| < \infty$ et $\sup_{n \geq 0} \|y_n\|_{s,s',\rho_2,\mu} < \infty$ et $\sum_{n \geq 0} |c_n| < \infty$. De plus $\forall b \in \ell_{s,s',\rho_1,\mu}^1$, $v(b) = \sum_{n \geq 0} c_n f_n(b) y_n$ donc v est nucléaire [212].

L'application v est d'image dense. Soit $b \in \ell_{s,s',\rho_2,\mu}^1$ définie par la suite $\{a_n\} \in \ell^1(\mathcal{C})$. Soit V un voisinage de b dans $\ell_{s,s',\rho_2,\mu}^1$ contenant la boule ouverte $B_0(b, r)$ ($r > 0$). On cherche dans cette boule un élément y appartenant à $\ell_{s,s',\rho_1,\mu}^1$.

Soit $N \geq 1$ tel que $\sum_{n \geq N} |a_n| < r$. Pour $n = 0 \dots N-1$ on pose $x_n = (\frac{\rho_1}{\rho_2})^n a_n$. Pour $n \geq N$ on pose $x_n = a_n$. Soit $y = \{y_n\}$ la suite définie par $y_n = |q|^{-\frac{sn(n+1)}{2}} (n!)^{s'} \rho_1^{-n} (1+n)^{-\mu} x_n$. La suite y appartient à $\ell_{s,s',\rho_1,\mu}^1$ et $\|b - y\|_{s,s',\rho_2,\mu} < r$.

Remarque : En fait la propriété qui nous intéressera sera le caractère compact de cette application ce qui est moins fort que le caractère nucléaire; cela sera suffisant pour utiliser des arguments de perturbations compactes.

Lemme 4.2.2 : Soient s et $s' \in \mathbb{R}$. Soit $\rho > 0$.

Si $\mu_1 > \mu_2$, l'injection naturelle $\ell_{s,s',\rho,\mu_1}^1 \xrightarrow{v} \ell_{s,s',\rho,\mu_2}^1$ est \mathcal{C} -linéaire continue et compacte.

Démonstration : Soit le diagramme commutatif :

$$\begin{array}{ccccc} & \ell_{s,s',\rho,\mu_1}^1 & \xrightarrow{v} & \ell_{s,s',\rho,\mu_2}^1 & \\ \psi_{s,s',\rho,\mu_1} & \uparrow & & \uparrow & \psi_{s,s',\rho,\mu_2} \\ & \ell^1(\mathcal{C}) & \xrightarrow{u} & \ell^1(\mathcal{C}) & \end{array}$$

On a $u(\{a_n\}) = (1+n)^{\mu_2-\mu_1} a_n$. En appliquant le lemme 1.3.1 p.15 [10] on sait que u est compacte. Donc v est aussi compacte.

Lemme 4.2.3 : Soient s et $s' \in \mathbb{R}$. Soient μ et $\bar{\mu} \in \mathbb{R}$.

Si $\rho_1 > \rho_2 > 0$, l'injection naturelle $\ell_{s,s',\rho_1,\mu}^1 \xrightarrow{v} \ell_{s,s',\rho_2,\bar{\mu}}^1$ est \mathcal{C} -linéaire continue compacte et d'image dense.

Démonstration :

Ce lemme est une généralisation du lemme 4.2.1.

Soit $b_n = |q|^{-\frac{sn(n+1)}{2}}(n!)^{s'}\rho_1^{-n}(1+n)^{-\mu}a_n$ avec $\sum_{n \geq 0} |a_n| < \infty$. Alors

$$b_n = |q|^{-\frac{sn(n+1)}{2}}(n!)^{s'}\rho_2^{-n}(1+n)^{-\bar{\mu}}[(\frac{\rho_2}{\rho_1})^n(1+n)^{\bar{\mu}-\mu}a_n].$$

Comme $\rho_2 < \rho_1 \Gamma \sum_{n \geq 0} |(\frac{\rho_2}{\rho_1})^n(1+n)^{\bar{\mu}-\mu}a_n| < \infty \Gamma \ell_{s,s',\rho_1,\mu}^1 \subset \ell_{s,s',\rho_2,\bar{\mu}}^1$.

L'application v est nucléaire donc compacte. On pose $c_n = (\frac{\rho_2}{\rho_1})^n(1+n)^{\bar{\mu}-\mu} : \sum_{n \geq 0} c_n < \infty$.

On pose $f_n(b) = \frac{b_n}{|q|^{-\frac{sn(n+1)}{2}}(n!)^{s'}\rho_1^{-n}(1+n)^{-\mu}} : \|f_n\| \leq 1$.

Soit $y_n = \{0, \dots, 0, |q|^{-\frac{sn(n+1)}{2}}(n!)^{s'}\rho_2^{-n}(1+n)^{-\bar{\mu}}, 0, \dots\} \Gamma$ suite dont le seul terme non nul est à la n -ième place : $\|y_n\|_{s,s',\rho_2,\bar{\mu}} = 1$ et $\forall b \in \ell_{s,s',\rho_1,\mu}^1, v(b) = \sum_{n \geq 0} c_n f_n(b) y_n$.

L'application v est d'image dense. Soit $b \in \ell_{s,s',\rho_2,\bar{\mu}}^1$ définie par la suite $\{a_n\}$. Soient $r > 0$ et $B_0(b, r)$ une boule ouverte dans $\ell_{s,s',\rho_2,\bar{\mu}}^1$. Soit $\alpha > 0 : \exists N \geq 1, \forall n \geq N, \sum_{n \geq N} |a_n| < r$ et $0 < \alpha \leq (\frac{\rho_1}{\rho_2})^n(1+n)^{\mu-\bar{\mu}}$. Pour $n = 0, \dots, N-1$ On pose $x_n = (\frac{\rho_1}{\rho_2})^n(1+n)^{\mu-\bar{\mu}}a_n$. Pour $n \geq N \Gamma$ on pose $x_n = \alpha a_n$. Soit $y = \{|q|^{-\frac{sn(n+1)}{2}}(n!)^{s'}\rho_1^{-n}(1+n)^{-\mu}x_n\}$. On vérifie que $y \in \ell_{s,s',\rho_1,\mu}^1$ et que $y \in B_0(b, r)$.

Lemme 4.2.4 : Soient $s, s', \mu \in \mathbb{R}$ fixés : $\mathcal{C}[[x]]_{q,s,s'} = \lim_{\rho > 0} \ell_{s,s',\rho,\mu}^1$ et $\mathcal{C}[[x]]_{(q,s,s')} = \lim_{\rho > 0} \ell_{s,s',\rho,\mu}^1$. On munit $\mathcal{C}[[x]]_{q,s,s'}$ de la topologie limite inductive et $\mathcal{C}[[x]]_{(q,s,s')}$ de la topologie limite projective.

4.3. Les (ρ, r, μ, ν) -opérateurs

Lemme 4.3.1 : $x : \ell_{s,s',\rho,\mu}^1 \longrightarrow \ell_{s,s',\frac{\rho}{|q|^s},\mu+s'}^1$.

Démonstration : Considérons le diagramme commutatif :

$$\begin{array}{ccc} \ell_{s,s',\rho,\mu}^1 & \xrightarrow{x} & \mathcal{C}^{\mathbb{N}} \\ \psi_{s,s',\rho,\mu} \uparrow & & \uparrow \psi_{s,s',\rho,\mu} \\ \ell^1(\mathcal{C}) & \xrightarrow{u} & \mathcal{C}^{\mathbb{N}} \end{array}$$

Si $\{a_n\} \in \ell^1(\mathcal{C})$ et si $\{d_n\} = u(\{a_n\})$ alors $d_n = |q|^{sn}n^{-s'}\rho(\frac{n}{n+1})^{-\mu}a_{n-1}$ donc $u(\ell^1(\mathcal{C})) \subset \ell_{0,0,|q|^{-s},s'}^1 \Gamma$ puis $x(\ell_{s,s',\rho,\mu}^1) \subset \ell_{s,s',\rho|q|^s,\mu+s'-1}^1$.

Lemme 4.3.2 : $\frac{d}{dx} : \ell_{s,s',\rho,\mu}^1 \longrightarrow \ell_{s,s',\rho|q|^s,\mu-s'-1}^1$.

Démonstration : Le principe est le même :

$$\begin{array}{ccc} \ell_{s,s',\rho,\mu}^1 & \xrightarrow{\frac{d}{dx}} & \mathcal{C}^{\mathbb{N}} \\ \psi_{s,s',\rho,\mu} \uparrow & & \uparrow \psi_{s,s',\rho,\mu} \\ \ell^1(\mathcal{C}) & \xrightarrow{v} & \mathcal{C}^{\mathbb{N}} \end{array}$$

On établit de même que $v(\ell^1(\mathcal{C})) \subset \ell_{0,0,|q|^s,-s'-1}^1 \Gamma$ puis que $\frac{d}{dx}(\ell_{s,s',\rho,\mu}^1) \subset \ell_{s,s',\rho|q|^s,\mu-s'-1}^1$.

Corollaire 4.3.3 : $x \frac{d}{dx} : \ell_{s,s',\rho,\mu}^1 \longrightarrow \ell_{s,s',\rho,\mu-1}^1$.

Lemme 4.3.4 : $\sigma_q : \ell_{s,s',\rho,\mu}^1 \longrightarrow \ell_{s,s',\rho|q|^{-1},\mu}^1$.

Proposition 4.3.5 : Soient $i, j, k \in \mathbb{N}$.

L'opérateur $x^k (x \frac{d}{dx})^i \sigma_q^j$ applique $\ell_{s,s',\rho,\mu}^1$ dans $\ell_{s,s',\rho|q|^{-(j+ks)},\mu-i+ks'}^1$.

Démonstration : Cette proposition résulte des lemmes précédents.

Définition 4.3.6 : Soient $s, s', \rho, r, \mu, \nu \in \mathbb{R}$ avec $\rho, r > 0$. Soit L un endomorphisme de $\mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$). Nous dirons que L est un (ρ, r, μ, ν) -opérateur s'il existe des applications linéaires continues $L_{s,s',\rho,r,\mu,\nu} : \ell_{s,s',\rho,\mu}^1 \longrightarrow \ell_{s,s',\rho r,\mu+\nu}^1$ pour $0 < \rho < \rho_0$ convenable (resp. $\rho > \rho_0$ convenable), rendant commutatifs les diagrammes

$$\begin{array}{ccc} \ell_{s,s',\rho,\mu}^1 & \xrightarrow{L_{s,s',\rho,r,\mu,\nu}} & \ell_{s,s',\rho r,\mu+\nu}^1 \\ \downarrow & & \downarrow \\ \mathcal{C}[[x]]_{q,s,s'} & \xrightarrow{L} & \mathcal{C}[[x]]_{q,s,s'} \end{array}$$

(resp. $\mathcal{C}[[x]]_{(q,s,s')}$).

Remarque : Si L est un (ρ, r, μ, ν) -opérateur alors L est un endomorphisme continu de $\mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$).

Proposition 4.3.7 : Soient $s, s', r, \mu, \nu \in \mathbb{R}$ avec $r > 0$. Soit $L : \mathcal{C}[[x]]_{q,s,s'} \longrightarrow \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$) un (ρ, r, μ, ν) -opérateur. On suppose que chacun des $L_{s,s',\rho,r,\mu,\nu}$ est à indice, d'indice indépendant de ρ pour $0 < \rho < \rho_0$ (resp. $\rho > \rho_0$) pour ρ_0 convenable. Alors L est à indice et $\chi(L) = \chi(L_{s,s',\rho,r,\mu,\nu})$.

Démonstration : Cette proposition est une conséquence directe du lemme 1.3.8 [10] (voir également [6]) et des lemmes établis en 4.2.

Proposition 4.3.8 : Soient $s, s', r_1, r_2, \mu \in \mathbb{R}$ avec $0 < r_1 < r_2$. Soient $\nu, \bar{\nu} \in \mathbb{R}$. Soit $L \in \mathbf{L}_{q,s,s'}$ (resp. $\mathbf{L}_{(q,s,s')}$) un (ρ, r_1, μ, ν) -opérateur tel que $\chi(L_{s,s',\rho,r_1,\mu,\nu})$ soit indépendant de ρ pour $0 < \rho < \rho_0$ (resp. $\rho > \rho_0$), ρ_0 convenable. Soit $K : \mathcal{C}[[x]]_{q,s,s'} \longrightarrow \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$) un $(\rho, r_2, \mu, \bar{\nu})$ -opérateur. Alors $L + K$ est un opérateur à indice, d'indice $\chi(L + K) = \chi(L)$.

Démonstration : Raisonnons sur les espaces $\mathcal{C}[[x]]_{q,s,s'}$. Considérons le diagramme suivant :

$$\begin{array}{ccccc} \ell_{s,s',\rho,\mu}^1 & \xrightarrow{K_{s,s',\rho,r_2,\mu,\bar{\nu}}} & \ell_{s,s',\rho r_2,\mu+\bar{\nu}}^1 & \xrightarrow{v} & \ell_{s,s',\rho r_1,\mu+\nu}^1 \\ \downarrow & & \downarrow & & \downarrow \\ \mathcal{C}[[x]]_{q,s,s'} & \xrightarrow{K} & \mathcal{C}[[x]]_{q,s,s'} & \xrightarrow{\text{identité}} & \mathcal{C}[[x]]_{q,s,s'} \end{array}$$

où v est l'injection naturelle Γ qui est compacte. Comme

$$\begin{array}{ccc} \ell_{s,s',\rho,\mu}^1 & \xrightarrow{L_{s,s',\rho,r_1,\mu,\nu}} & \ell_{s,s',\rho r_1,\mu+\nu}^1 \\ \downarrow & & \downarrow \\ \mathcal{C}[[x]]_{q,s,s'} & \xrightarrow{L} & \mathcal{C}[[x]]_{q,s,s'} \end{array}$$

alors

$$\begin{array}{ccc} \ell_{s,s',\rho,\mu}^1 & \xrightarrow{v \circ K_{s,s',\rho,r_2,\mu,\bar{\nu}} + L_{s,s',\rho,r_1,\mu,\nu}} & \ell_{s,s',\rho r_1,\mu+\nu}^1 \\ \downarrow & & \downarrow \\ \mathcal{C}[[x]]_{q,s,s'} & \xrightarrow{L+K} & \mathcal{C}[[x]]_{q,s,s'} \end{array}$$

L'opérateur $v \circ K_{s,s',\rho,r_2,\mu,\bar{\nu}}$ est un opérateur linéaire continu compact et $L_{s,s',\rho,r_1,\mu,\nu}$ est à indice. Alors $v \circ K_{s,s',\rho,r_2,\mu,\bar{\nu}} + L_{s,s',\rho,r_1,\mu,\nu}$ est un opérateur à indice (théorème 2.4) et :

$$\chi(v \circ K_{s,s',\rho,r_2,\mu,\bar{\nu}} + L_{s,s',\rho,r_1,\mu,\nu}) = \chi(L_{s,s',\rho,r_1,\mu,\nu})$$

qui est indépendant de ρ pour $0 < \rho < \rho_0$. Donc $L + K$ est un (ρ, r_1, μ, ν) -opérateur vérifiant les conditions de la proposition 4.3.7 Γ donc $L + K$ est indice et $\chi(L + K) = \chi(L_{s,s',\rho,r_1,\mu,\nu}) = \chi(L)$.

Remarque : On n'utilise que la compacité de l'injection naturelle $\ell_{s,s',\rho r_2,\mu+\bar{\nu}}^1 \longrightarrow \ell_{s,s',\rho r_1,\mu+\nu}^1$. D'autre part Γ les réels ν et $\bar{\nu}$ sont quelconques; cela signifie que pour étudier un (ρ, r_1, μ, ν_1) -opérateur par rapport à un (ρ, r_2, μ, ν_2) -opérateur Γ on compare d'abord r_1 à r_2 . Si $r_1 = r_2$ Γ on utilise la proposition suivante :

Proposition 4.3.9 : Soient $s, s', r, \mu, \nu_1, \nu_2 \in \mathbb{R}$ avec $0 < r$ et $\nu_1 < \nu_2$. Soit $L \in \mathbf{L}_{q,s,s'}$ (resp. $\mathbf{L}_{(q,s,s')}$) un (ρ, r, μ, ν_1) -opérateur, tel que $\chi(L_{s,s',\rho,r,\mu,\nu_1})$ soit indépendant de ρ pour $0 < \rho < \rho_0$ (resp. $\rho > \rho_0$), ρ_0 convenable. Soit $K : \mathcal{C}[[x]]_{q,s,s'} \longrightarrow \mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$) un (ρ, r, μ, ν_2) -opérateur. Alors $L + K$ est un opérateur à indice, d'indice $\chi(L + K) = \chi(L)$.

Démonstration : On montre comme précédemment que $L + K$ est un (ρ, r, μ, ν_1) -opérateur et on utilise la compacité de l'injection naturelle $\ell_{s,s',\rho r,\mu+\nu_2}^1 \longrightarrow \ell_{s,s',\rho r,\mu+\nu_1}^1$.

4.4. Théorèmes d'indices

Dans tout ce paragraphe Γ nous considérons l'opérateur :

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(x \frac{d}{dx}\right)^i \sigma_q^j$$

avec $a_{i,j}(x) = \sum_{k \geq 0} \alpha_{i,j,k} x^k \in \mathcal{C}\{x\}$. Soient s et $s' \in \mathbb{R}$ tels que $s > 0$ et s' quelconque Γ ou $s = 0$ et $s' \geq 0$.

Proposition 4.4.1 : L'opérateur L opère de $\ell_{s,s',\rho,\mu}^1$ dans $\ell_{s,s',\rho|q|^{-p_s^0(L)},\mu+v_{s,s'}(L)}^1$ avec

$$p_s^0(L) = \inf_{\alpha_{i,j,k} \neq 0} (j + ks) \text{ et } v_{s,s'}(L) = \inf_{\alpha_{i,j,k} \neq 0 \text{ et } j+ks=p_s^0(L)} (ks' - i).$$

En d'autres termes, L est un $(\rho, |q|^{-p_s^0(L)}, \mu, v_{s,s'}(L))$ -opérateur.

Démonstration : On vérifie que pour $s > 0$ $\Gamma p_s^0(L)$ est bien défini et l'ensemble des triplets (i, j, k) satisfaisant $j + ks = p_s^0(L)$ est fini donc $v_{s,s'}(L)$ est aussi défini. Si $s = 0$ $\Gamma p_s^0(L)$ est défini par un indice j_0 (le plus petit). On impose alors $s' \geq 0$ pour minorer les quantités $ks' - i$ et donc pour pouvoir définir $v_{s,s'}(L)$.

On va maintenant éliminer les termes qui par argument de compacité n'interviennent pas dans le calcul de l'indice (s'il existe).

Soit $M(s) = \{(i, j, k) / \alpha_{i,j,k} \neq 0 \text{ et } j + ks = p_s^0(L)\}$.

Soient $N(s, s') = \{(i, j, k) \in M(s) / ks' - i = v_{s,s'}(L)\}$ et $\tilde{L} = \sum_{(i,j,k) \in N(s,s')} \alpha_{i,j,k} x^k (x \frac{d}{dx})^i \sigma_q^j$.

Propriété 4.4.2 : L'opérateur \tilde{L} est un $(\rho, |q|^{-p_s^0(L)}, \mu, v_{s,s'}(L))$ -opérateur.

On va montrer que l'étude de L se ramène à l'étude de \tilde{L} :

Proposition 4.4.3 : Si $\tilde{L}_{s,s',\rho,|q|^{-p_s^0(L)},\mu,v_{s,s'}(L)}$ est à indice, indépendant de ρ pour $0 < \rho < \rho_0$ (resp. $\rho > \rho_0$) pour ρ_0 convenable, alors L est à indice dans $\mathcal{C}[[x]]_{q,s,s'}$ (resp. dans $\mathcal{C}[[x]]_{(q,s,s')}$) et $\chi(L) = \chi(\tilde{L})$.

Démonstration : Soient $L_1 = \sum_{\{(i,j,k) \in M(s), ks' - i > v_{s,s'}(L)\}} \alpha_{i,j,k} x^k (x \frac{d}{dx})^i \sigma_q^j$ et

$L_2 = \sum_{(i,j,k) \notin M(s)} \alpha_{i,j,k} x^k (x \frac{d}{dx})^i \sigma_q^j$.

Il est clair que L_1 est un $(\rho, |q|^{-p_s^0(L)}, \mu, \nu_1)$ -opérateur et que L_2 est un (ρ, r_2, μ, ν_2) -opérateur avec $\nu_1 > v_{s,s'}(L)$ et $r_2 > |q|^{-p_s^0(L)}$. Alors $\tilde{L} + L_1$ est un $(\rho, |q|^{-p_s^0(L)}, \mu, v_{s,s'}(L))$ -opérateur et $\chi(\tilde{L} + L_1) = \chi(\tilde{L})$. On vérifie ensuite que $\tilde{L} + L_1 + L_2$ est un $(\rho, |q|^{-p_s^0(L)}, \mu, v_{s,s'}(L))$ -opérateur et que $\chi(\tilde{L} + L_1 + L_2) = \chi(\tilde{L} + L_1) = \chi(\tilde{L})$. Or $\tilde{L} + L_1 + L_2 = L$ d'où $\chi(L) = \chi(\tilde{L})$.

Etude de \tilde{L} :

On rappelle que $\tilde{L} = \sum_{(i,j,k) \in N(s,s')} \alpha_{i,j,k} x^k (x \frac{d}{dx})^i \sigma_q^j$. On définit les deux séries formelles :

$$\eta_s(x) = \sum_{n \geq 0} q^{\frac{sn(n+1)}{2}} x^n \text{ et } \eta_s^*(x) = \sum_{n \geq 0} q^{-\frac{sn(n+1)}{2}} x^n$$

et le produit de Hadamard de 2 séries formelles

$$(\sum_{n \geq 0} a_n x^n) \square (\sum_{n \geq 0} b_n x^n) = \sum_{n \geq 0} a_n b_n x^n.$$

On définit le nouvel opérateur : $\tilde{\tilde{L}}(f) = \eta_s \square \tilde{L}(\eta_s^* \square f)$. L'étude de \tilde{L} se ramène à l'étude de $\tilde{\tilde{L}}$. Il est facile de voir que :

Propriétés 4.4.4 :

- $\tilde{L} : \ell_{s,s',\rho,\mu}^1 \longrightarrow \ell_{s,s',\rho|q|^{-p_s^0(L)},\mu+v_{s,s'}(L)}^1$ **ssi** $\tilde{\tilde{L}} : \ell_{0,s',\rho,\mu}^1 \longrightarrow \ell_{0,s',\rho|q|^{-p_s^0(L)},\mu+v_{s,s'}(L)}^1$
- $\tilde{L} : \mathcal{C}[[x]]_{q,s,s'} \longrightarrow \mathcal{C}[[x]]_{q,s,s'}$ **ssi** $\tilde{\tilde{L}} : \mathcal{C}[[x]]_{q,0,s'} \longrightarrow \mathcal{C}[[x]]_{q,0,s'}$
- $\tilde{L} : \mathcal{C}[[x]]_{(q,s,s')} \longrightarrow \mathcal{C}[[x]]_{(q,s,s')}$ **ssi** $\tilde{\tilde{L}} : \mathcal{C}[[x]]_{(q,0,s')} \longrightarrow \mathcal{C}[[x]]_{(q,0,s')}$

Soit l'application $\eta : \begin{array}{ccc} \ell_{0,s',\rho,\mu}^1 & \longrightarrow & \ell_{s,s',\rho,\mu}^1 \\ f & \longmapsto & \eta_s^* \square f \end{array}$

C'est un isomorphisme et $\eta^{-1} : \begin{array}{ccc} \ell_{s,s',\rho,\mu}^1 & \longrightarrow & \ell_{0,s',\rho,\mu}^1 \\ f & \longmapsto & \eta_s \square f \end{array}$

Donc η est à indice nul. Or $\tilde{\tilde{L}} = \eta^{-1} \circ \tilde{L} \circ \eta$ c'est-à-dire que si \tilde{L} est à indice nul alors $\tilde{\tilde{L}}$ l'est aussi (et réciproquement).

Propriété 4.4.5 : $\tilde{L} : \ell_{s,s',\rho,\mu}^1 \longrightarrow \ell_{s,s',\rho|q|^{-p_s^0(L)},\mu+v_{s,s'}(L)}^1$ **est à indice nul si et seulement si**
 $\tilde{\tilde{L}} : \ell_{0,s',\rho,\mu}^1 \longrightarrow \ell_{0,s',\rho|q|^{-p_s^0(L)},\mu+v_{s,s'}(L)}^1$ **est à indice nul; dans ce cas, $\chi(\tilde{L}) = \chi(\tilde{\tilde{L}})$ (énoncé similaire dans les cas $\mathcal{C}[[x]]_{q,s,s'}$ et $\mathcal{C}[[x]]_{(q,s,s')}$).**

Etude de $\tilde{\tilde{L}}$:

On montre sans difficulté la propriété :

Propriété 4.4.6 :

$$\tilde{\tilde{L}} = \left[\sum_{(i,j,k) \in N(s,s')} \alpha_{i,j,k} q^{\frac{sk(k+1)}{2}} x^k \left(x \frac{d}{dx} \right)^i \right] \sigma_q^{p_s^0(L)}$$

L'opérateur $\sigma_q^{p_s^0(L)}$ est un isomorphisme de $\ell_{0,s',\rho,\mu}^1$ dans $\ell_{0,s',\rho|q|^{-p_s^0(L)},\mu}^1$ (resp. un automorphisme de $\mathcal{C}[[x]]_{q,0,s'}$ et de $\mathcal{C}[[x]]_{(q,0,s')}$) donc est à indice nul dans ces espaces. Soit $T_L = \sum_{(i,j,k) \in N(s,s')} \alpha_{i,j,k} q^{\frac{sk(k+1)}{2}} x^k \left(x \frac{d}{dx} \right)^i : \tilde{\tilde{L}} = T_L \circ \sigma_q^{p_s^0(L)}$.

Etude de l'opérateur différentiel T_L :

On s'est ramené à l'étude d'une équation différentielle ordinaire cas traité par J.-P. Ramis [910]. Soient $\mathfrak{S} = \{i = 0, \dots, I / \exists j, \exists k, (i, j, k) \in N(s, s')\}$ et $K_i = \{k \geq 0 / \exists j, (i, j, k) \in N(s, s')\}$ pour un $i \in \mathfrak{S}$. Soient $i \in \mathfrak{S}$ et $k \in K_i$; on note $j_{(i,k)}$ l'unique indice j tel que $(i, j, k) \in N(s, s')$. Alors :

$$T_L = \sum_{i \in \mathfrak{S}} \left(\sum_{k \in K_i} \alpha_{i,j_{(i,k)},k} q^{\frac{sk(k+1)}{2}} x^k \right) \left(x \frac{d}{dx} \right)^i.$$

- Premier cas : $s > 0$ et s' quelconque.

Alors $M(s)$ est fini ainsi que $N(s, s')$ et T_L est à coefficients polynomiaux. J.-P Ramis a montré que dans ce cas pour tout $s' \in \mathbb{R}$ T_L est un opérateur à indice. En appliquant ses résultats l'indice de T_L est donné par :

$$\begin{aligned}\chi(T_L, \mathcal{C}[[x]]_{q,s,s'}) &= -k_{i_1}(s') = -\inf\{k/\exists i, \exists j, (i,j,k) \in N(s,s')\} \\ \chi(T_L, \mathcal{C}[[x]]_{(q,s,s')}) &= -k_{i_2}(s') = -\sup\{k/\exists i, \exists j, (i,j,k) \in N(s,s')\}\end{aligned}$$

- Second cas : $s = 0$ et $s' > 0$.

Si $(i,j,k) \in N(s,s')$ alors $j = j_0 = \inf\{j/\exists i, \exists k, \alpha_{i,j,k} \neq 0\}$ et pour un $i \in \mathfrak{I}$ il existe un unique indice k tel que $(i,j_0,k) \in N(s,s')$. L'opérateur T_L est à coefficients polynomiaux et on a de même :

$$\begin{aligned}\chi(T_L, \mathcal{C}[[x]]_{q,s,s'}) &= -\inf\{k/\exists i, \exists j, (i,j,k) \in N(s,s')\} \\ \chi(T_L, \mathcal{C}[[x]]_{(q,s,s')}) &= -\sup\{k/\exists i, \exists j, (i,j,k) \in N(s,s')\}\end{aligned}$$

- Troisième cas : $s = 0$ et $s' = 0$.

Si $(i,j,k) \in N(s,s')$ alors $j = j_0 = \inf\{j/\exists i, \exists k, \alpha_{i,j,k} \neq 0\}$ et $i = i_0 = \sup\{i/\exists k, \alpha_{i,j_0,k} \neq 0\}$. Alors $T_L = a_{i_0,j_0}(x)(x \frac{d}{dx})^{i_0}$ avec $a_{i_0,j_0}(x) \in \mathcal{C}\{x\}$. On sait alors que T_L est à indice dans $\mathcal{C}[[x]]_{q,0,0} = \mathcal{C}\{x\}$ [8] et $\chi(T_L, \mathcal{C}[[x]]_{q,0,0}) = -v(a_{i_0,j_0}) = -\inf\{k/\exists i, \exists j, (i,j,k) \in N(s,s')\}$.

On obtient ainsi le théorème fondamental :

Théorème 4.4.7 : Soit $L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x)(x \frac{d}{dx})^i \sigma_q^j$, avec $a_{i,j}(x) = \sum_{k \geq 0} \alpha_{i,j,k} x^k \in \mathcal{C}\{x\}$ et q complexe, $0 < |q| < 1$. Soient s et $s' \in \mathbb{R}$ ($s > 0$ et $s' \in \mathbb{R}$, ou $s = 0$ et $s' \geq 0$).

Soient $p_s^0(L) = \inf_{\alpha_{i,j,k} \neq 0} (j + ks)$ et $M(s) = \{(i,j,k) / \alpha_{i,j,k} \neq 0 \text{ et } j + ks = p_s^0(L)\}$.

Soient $v_{s,s'}(L) = \inf_{(i,j,k) \in M(s)} (ks' - i)$ et $N(s,s') = \{(i,j,k) \in M(s) / ks' - i = v_{s,s'}(L)\}$.

- Si $s > 0$ et $s' \in \mathbb{R}$ ou si $s = 0$ et $s' \geq 0$ alors L est un opérateur à indice dans $\mathcal{C}[[x]]_{q,s,s'}$, d'indice égal à :

$$\chi(L, \mathcal{C}[[x]]_{q,s,s'}) = -\inf\{k/\exists i, \exists j, (i,j,k) \in N(s,s')\}$$

- Si $s > 0$ et $s' \in \mathbb{R}$ ou si $s = 0$ et $s' > 0$ alors L est un opérateur à indice dans $\mathcal{C}[[x]]_{(q,s,s')}$, d'indice égal à :

$$\chi(L, \mathcal{C}[[x]]_{(q,s,s')}) = -\sup\{k/\exists i, \exists j, (i,j,k) \in N(s,s')\}$$

Plus généralement :

Théorème 4.4.8 : Soit $L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x)(\frac{d}{dx})^i \sigma_q^j$, avec $a_{i,j}(x) = \sum_{k \geq 0} \alpha_{i,j,k} x^k \in \mathcal{C}\{x\}$ et q un complexe, $0 < |q| < 1$. Soient s et $s' \in \mathbb{R}$ ($s > 0$ et $s' \in \mathbb{R}$, ou $s = 0$ et $s' \geq 0$).

Soient $p_s^0(L) = \inf_{\alpha_{i,j,k} \neq 0} (j + (k-i)s)$ et $M(s) = \{(i,j,k) / \alpha_{i,j,k} \neq 0 \text{ et } j + (k-i)s = p_s^0(L)\}$.

Soient $v_{s,s'}(L) = \inf_{(i,j,k) \in M(s)} ((k-i)s' - i)$ et $N(s,s') = \{(i,j,k) \in M(s) / (k-i)s' - i = v_{s,s'}(L)\}$.

- Si $s > 0$ et $s' \in \mathbb{R}$ ou si $s = 0$ et $s' \geq 0$ alors L est un opérateur à indice dans $\mathcal{C}[[x]]_{q,s,s'}$, d'indice égal à :

$$\chi(L, \mathcal{C}[[x]]_{q,s,s'}) = -\inf\{k-i/\exists j, (i,j,k) \in N(s,s')\}$$

- Si $s > 0$ et $s' \in \mathbb{R}$ ou si $s = 0$ et $s' > 0$ alors L est un opérateur à indice dans $\mathcal{C}[[x]]_{(q,s,s')}$, d'indice égal à :

$$\chi(L, \mathcal{C}[[x]]_{(q,s,s')}) = -\sup\{k - i / \exists j, (i, j, k) \in N(s, s')\}$$

4.5. Théorème d'indice dans $\mathcal{C}[[x]]$ ($s = +\infty$)

On se propose d'étudier l'existence éventuelle d'un indice dans l'espace des séries formelles $\mathcal{C}[[x]]$. La démarche est la même que pour les opérateurs aux q -différences [1] ou que pour les opérateurs différentiels [8]. Soit L un opérateur de la forme :

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(x \frac{d}{dx}\right)^i \sigma_q^j$$

où les $a_{i,j}$ sont des séries formelles et $0 < |q| < 1$. On pose $a_{i,j}(x) = \sum_{k \geq 0} \alpha_{i,j,k} x^k$. On suppose qu'il existe deux indices i et j tels que les séries formelles $a_{i,j}(x)$ et $a_{i,J}(x)$ ne soient pas identiquement nulles. On note $v(a_{i,j})$ le plus petit entier k pour lequel $\alpha_{i,j,k}$ soit non nul.

Théorème 4.5.1 : L'opérateur $L : \mathcal{C}[[x]] \longrightarrow \mathcal{C}[[x]]$ est un opérateur à indice, d'indice égal à :

$$\chi(L, \mathcal{C}[[x]]) = \sup(-v(a_{i,j})) = -\inf(v(a_{i,j})).$$

Plus généralement, L de la forme (1) est à indice dans $\mathcal{C}[[x]]$, d'indice égal à $-\inf(v(a_{i,j}) - i)$.

Démonstration : On pose $m(L) = \inf(v(a_{i,j}))$.

Soit $\Gamma = \{(i, j), i = 0 \dots I, j = 0 \dots J / v(a_{i,j}) = m(L)\}$. Si $(i, j) \notin \Gamma$ alors $v(a_{i,j}) > m(L)$.

Soient $\theta(x) = \sum_{p \geq 0} b_p x^p$ et $f(x) = \sum_{n \geq 0} f_n x^n$ deux séries formelles vérifiant $L f = \theta$.

En écrivant alors formellement cette relation et après identification des deux membres on obtient pour p assez grand une relation du type :

$$U(p) \times f_{p-m(L)} = S_p(f_0, \dots, f_{p-m(L)-1}, b_p)$$

où $S_p(f_0, \dots, f_{p-m(L)-1}, b_p)$ est une expression algébrique en les variables $f_0, \dots, f_{p-m(L)-1}, b_p$ et où l'on a posé :

$$U(p) = \sum_{(i,j) \in \Gamma} \alpha_{i,j,m(L)} q^{j(p-m(L))} (p-m(L))^i.$$

Soit j_0 le plus petit des indices j pour lesquels il existe i tel que $(i, j) \in \Gamma$. Soit $i_0 = \sup\{i / (i, j_0) \in \Gamma\}$.

Comme $0 < |q| < 1$ et $\alpha_{i_0,j_0,m(L)} \neq 0$ $\Gamma U(p)$ est non nul à partir d'un certain rang $P > m(L)$. Pour θ telle que $v(\theta) \geq P$ il existe donc une unique série formelle f telle que $L f = \theta$ et $v(f) \geq P - m(L)$ c'est-à-dire que $L : x^{P-m(L)} \mathcal{C}[[x]] \longrightarrow x^P \mathcal{C}[[x]]$ est un isomorphisme donc est à indice quel est nul. On en déduit alors le résultat par des arguments classiques (proposition 1.3 [8]).

4.6. Théorème d'indice dans $\mathcal{C}\{x\}$

On note $\mathcal{C}\{x\}$ l'ensemble des séries convergentes à l'origine. On suppose que les $a_{i,j}$ sont dans $\mathcal{C}\{x\}$. Alors $L : \mathcal{C}\{x\} \longrightarrow \mathcal{C}\{x\}$. On va étudier l'indice de L d'une autre façon qu'en 4.4.

Soit j_0 le plus petit des indices tels qu'il existe un i avec a_{i,j_0} non identiquement nul. Soit i_0 le plus grand des i tels que a_{i,j_0} ne soit pas identiquement nul.

Soit $r > 0$. On note $\Delta_r = \{x \mid |x| \leq r\}$ et $B^i(\Delta_r)$ les fonctions de classe C^i sur Δ_r et holomorphes sur l'ouvert $\dot{\Delta}_r$. On munit $B^i(\Delta_r)$ de la norme :

$$\|f\|_{r,i} = \sum_{l=0}^i \sup_{x \in \Delta_r} |f^{(l)}(x)|$$

qui en fait un espace de Banach.

Pour établir le théorème d'indice dans $\mathcal{C}\{x\}$ on établit le théorème suivant puis on utilise l'argument classique de limite inductive.

Théorème 4.6.1 : Soit $r > 0$ assez petit pour que les séries $a_{i,j}(x)$ soient toutes dans $B^1(\Delta_r)$ et que $a_{i_0,j_0}(x)$ ne s'annule pas sur Δ_r ailleurs qu'à l'origine. Alors $L : B^{i_0}(\Delta_r) \longrightarrow B^0(\Delta_{|q|^{-j_0}r})$ est à indice, d'indice égal à : $-v(a_{i_0,j_0})$.

Démonstration : Soit $f \in B^{i_0}(\Delta_r)$. Pour (i,j) tel qu'il existe k avec $\alpha_{i,j,k} \neq 0$: $\sigma_q^j f(x) \in B^{i_0}(\Delta_{|q|^{-j}r})$. Si $j = j_0$ alors $i \leq i_0$ donc $(\frac{d}{dx})^i \sigma_q^{j_0} f \in B^{i_0-i}(\Delta_{|q|^{-j_0}r}) \subset B^0(\Delta_{|q|^{-j_0}r})$. Si $j > j_0$ alors $|q|^{-j}r > |q|^{-j_0}r$ et donc $(\frac{d}{dx})^i \sigma_q^{j_0} f \in B^0(\Delta_{|q|^{-j_0}r})$.

On pose $L = \tilde{L} + L_1 + L_2$ avec $\tilde{L} = a_{i_0,j_0}(x)(x\frac{d}{dx})^{i_0} \sigma_q^{j_0} \Gamma L_1 = \sum_{j=j_0+1}^J \sum_{i=0}^I a_{i,j}(x)(x\frac{d}{dx})^i \sigma_q^j$ et $L_2 = (\sum_{i=0}^{i_0-1} a_{i,j_0}(x)(x\frac{d}{dx})^i) \sigma_q^{j_0}$. L'opérateur \tilde{L} est d'indice $-v(a_{i_0,j_0})$ ΓL_1 est compacte (théorème d'Ascoli-Arzelà) et L_2 est également compacte (cf par exemple [8]). On applique alors le théorème 2.4.

On utilise ensuite le fait que $\mathcal{C}\{x\} = \lim_{r \rightarrow 0} B^{i_0}(\Delta_r)$ et que l'on travaille avec des applications linéaires continues $L_r : B^{i_0}(\Delta_r) \longrightarrow B^0(\Delta_{|q|^{-j_0}r})$ d'où le :

Théorème 4.6.2 : L'opérateur $L : \mathcal{C}\{x\} \longrightarrow \mathcal{C}\{x\}$ est à indice, d'indice égal à :

$$\chi(L, \mathcal{C}\{x\}) = -v(a_{i_0,j_0})$$

4.7. Théorèmes de comparaison

Rappelons quelques notations. Soit L un opérateur de la forme :

$$L = \sum_{i=0}^I \sum_{j=0}^J \sum_{k \geq 0} \alpha_{i,j,k} x^k (x\frac{d}{dx})^i \sigma_q^j \quad (2)$$

Soient s et $s' \in \mathbb{R}$. On note : $p_s^0(L) = \inf_{\alpha_{i,j,k} \neq 0} (j + ks)$ (si cette quantité existe) et $M(s) = \{(i, j, k) | \alpha_{i,j,k} \neq 0 \text{ et } j + ks = p_s^0(L)\}$. Enfin si elle existe on pose $v_{s,s'}(L) = \inf_{(i,j,k) \in M(s)} (ks' - i)$ et $N(s, s') = \{(i, j, k) \in M(s) | ks' - i = v_{s,s'}(L)\}$.

Définition 4.7.1 : On dit que $s \in \mathbb{R}$ est une valeur exceptionnelle pour L s'il existe deux triplets (i_1, j_1, k_1) et (i_2, j_2, k_2) dans $M(s)$ tels que $k_1 \neq k_2$.

Remarque : Si s est une valeur exceptionnelle s'écrit sous la forme $s = \frac{j_2 - j_1}{v(a_{i_1, j_1}) - v(a_{i_2, j_2})}$: il y en a un nombre fini.

Définition 4.7.2 : Soit $s \in \mathbb{R}$. Le réel s' est une valeur exceptionnelle relativement à s pour l'opérateur L s'il existe deux triplets (i_1, j_1, k_1) et (i_2, j_2, k_2) dans $N(s, s')$ tels que $k_1 \neq k_2$.

Remarque : Pour s fixé il y a un nombre fini de valeurs exceptionnelles relativement à s .

On note le cas échéant : $\chi_{q,s,s'} = \chi(L, \mathcal{C}[[x]]_{q,s,s'})$ et $\chi_{(q,s,s')} = \chi(L, \mathcal{C}[[x]]_{(q,s,s')})$.

Proposition 4.7.3 : Soient $s > 0$ et $s'_0 < s'_1 \in \mathbb{R}$ (resp. $s = 0$ et $0 \leq s'_0 < s'_1$). On suppose que pour tout $s' \in]s'_0, s'_1[$, s' n'est pas exceptionnelle relativement à s . Alors :

$$\begin{aligned} \chi_{q,s,s'} &= \chi_{(q,s,s')} = \chi \quad \forall s' \in]s'_0, s'_1[\\ \chi_{q,s,s'_0} &= \chi_{(q,s,s'_0)} = \chi \end{aligned}$$

où χ est une constante.

Démonstration : Soit $s' \in]s'_0, s'_1[$. Par hypothèse s' n'est pas exceptionnelle donc si (i_1, j_1, k_1) et (i_2, j_2, k_2) sont dans $N(s, s')$ alors $k_1 = k_2$. On note $k(s')$ cet élément $i(s') = s'k(s') - v_{s,s'}(L)$ et $j(s') = p_s^0(L) - sk(s')$. Alors $\chi_{q,s,s'} = \chi_{(q,s,s')} = -k(s')$. On montre ensuite que le triplet $(i(s'), j(s'), k(s'))$ est localement constant sur le connexe $]s'_0, s'_1[$ donc constant sur $]s'_0, s'_1[$. Notons le désormais (i_0, j_0, k_0) . Pour $s' \in]s'_0, s'_1[$: $\chi_{q,s,s'} = \chi_{(q,s,s')} = -k_0 = \chi$. On vérifie que $(i_0, j_0, k_0) \in N(s, s'_0) \cap N(s, s'_1)$.

Soit $(i_1, j_1, k_1) \in N(s, s'_0)$ avec $k_1 = \inf\{k/\exists i, \exists j, (i, j, k) \in N(s, s'_0)\} : k_1 \leq k_0$. Si $k_1 < k_0$: $s'_0 = \frac{i_0 - i_1}{k_0 - k_1}$ et comme $s'_0 < s'_1$ on en déduit que $k_1 s'_1 - i_1 < k_0 s'_1 - i_0$ ce qui est en contradiction avec le fait que $(i_0, j_0, k_0) \in N(s, s'_1)$.

L'autre égalité se démontre de la même manière.

Proposition 4.7.4 : Soit $s > 0$. Alors il existe s'_0 et $s'_1 \in \mathbb{R}$ tels que :

$$\begin{aligned} \forall s' < s'_0, \quad \chi_{q,s,s'} &= \chi_{(q,s,s')} = \sup\{k/\exists i, \exists j, (i, j, k) \in M(s)\} \\ \forall s' > s'_1, \quad \chi_{q,s,s'} &= \chi_{(q,s,s')} = \inf\{k/\exists i, \exists j, (i, j, k) \in M(s)\} \end{aligned}$$

Démonstration :

Soient $k^+ = \sup\{k/\exists i, \exists j, (i, j, k) \in M(s)\}$ et $k^- = \inf\{k/\exists i, \exists j, (i, j, k) \in M(s)\}$. Soit Ω l'ensemble des valeurs s' exceptionnelles relativement à s . Si $\Omega = \emptyset$ alors $\chi_{q,s,s'} = \chi_{(q,s,s')} =$

constante pour tout $s' \in \mathbb{R}$. Si $\Omega \neq \emptyset$ soit $s'_0 = \inf \Omega$ et $s'_1 = \sup \Omega$. D'après la proposition 4.7.3 $\chi_{q,s,s'} = \chi_{(q,s,s')} = \text{constante}$ pour tout $s' < s'_0$ et $\chi_{q,s,s'} = \chi_{(q,s,s')} = \text{constante}$ pour tout $s' > s'_1$. Dans les deux cas on calcule les constantes pour $s' \rightarrow -\infty$ et $s' \rightarrow +\infty$. Si $s' \rightarrow -\infty$ soit $(i_{-\infty}, j_{-\infty}, k_{-\infty}) \in N(s, s') : \chi_{q,s,s'} = \chi_{(q,s,s')} = -k_{-\infty}$. Puisque $(i_{-\infty}, j_{-\infty}, k_{-\infty}) \in N(s, s') \cap \Gamma$ on montre qu'alors $k_{-\infty} \geq k^+$ et donc $k_{-\infty} = k^+$. De même $k_{+\infty} = k^-$.

Proposition 4.7.5 : Soit $[s_0, s_1] \subset [0, +\infty[$ tel que pour tout $s \in]s_0, s_1[$, s n'est pas exceptionnelle. Alors : $\forall s \in]s_0, s_1[$, $\forall s' \in \mathbb{R}$, $\chi(L, \mathcal{C}[[x]]_{q,s,s'}) = \chi(L, \mathcal{C}[[x]]_{(q,s,s')}) = \chi$ et il existe s'_0 assez grand et s'_1 assez petit tels que $\chi(L, \mathcal{C}[[x]]_{q,s_0,s'_0}) = \chi(L, \mathcal{C}[[x]]_{(q,s_1,s'_1)}) = \chi$.

Démonstration : Soit $s \in]s_0, s_1[$: s n'est pas exceptionnelle donc si (i_1, j_1, k_1) et (i_2, j_2, k_2) appartiennent à $M(s) \cap \Gamma$ alors $k_1 = k_2$. On note $k(s)$ cet élément et $j(s)$ l'unique indice j tel qu'il existe i avec $(i, j, k(s)) \in M(s)$ ($j(s)$ et $k(s)$ sont indépendants de s'). On montre alors que $(j(s), k(s))$ est localement constant sur le connexe $]s_0, s_1[\cap \Gamma$ donc constant sur $]s_0, s_1[$: $\chi_{q,s,s'} = \chi_{(q,s,s')} = -k_0$ pour tout $s \in]s_0, s_1[$ et pour tout $s' \in \mathbb{R}$. Soit i_0 tel que $(i_0, j_0, k_0) \in M(s)$ pour $s \in]s_0, s_1[$: $(i_0, j_0, k_0) \in M(s_0) \cap M(s_1)$. Soit $(i^+, j^+, k^+) \in M(s_1)$ avec $k^+ = \sup\{k/\exists i, \exists j, (i, j, k) \in M(s_1)\} : \chi_{(q,s_1,s'_1)} = -k^+$ pour s'_1 tendant vers $-\infty$. Soit $(i^-, j^-, k^-) \in M(s_0)$ avec $k^- = \inf\{k/\exists i, \exists j, (i, j, k) \in M(s_0)\} : \chi_{q,s_0,s'_0} = -k^-$ pour s'_0 tendant vers $+\infty$ (proposition 4.7.4). Alors $k^- \leq k_0 \leq k^+$. Si $k^- < k_0$ alors $s_0 = \frac{j^- - j_0}{k_0 - k^-} < s_1 : j^- + k^- s_1 < j_0 + k_0 s_1$ ce qui est en contradiction avec $(i_0, j_0, k_0) \in M(s_1)$. Si $k^+ > k_0$ alors $s_1 = \frac{j^+ - j_0}{k_0 - k^+} > s_0 : j^+ + k^+ s_0 < j_0 + k_0 s_0$ ce qui est en contradiction avec $(i_0, j_0, k_0) \in M(s_0)$.

Proposition 4.7.6 : Soient $s > 0$ et $s'_0 < s'_1 \in \mathbb{R}$ (resp. $s = 0$ et $0 \leq s'_0 < s'_1$) tels que pour tout $s' \in]s'_0, s'_1[$, s' n'est pas exceptionnelle relativement à s . Alors

$$\dim \text{Ker} \left(L, \frac{\mathcal{C}[[x]]_{(q,s,s'_1)}}{\mathcal{C}[[x]]_{q,s,s'_0}} \right) = 0$$

Démonstration : On sait que $\chi(L, \mathcal{C}[[x]]_{(q,s,s'_1)}) = \chi(L, \mathcal{C}[[x]]_{q,s,s'_0})$ (proposition 4.7.3). Considérons le diagramme commutatif :

$$\begin{array}{ccc} \mathcal{C}[[x]]_{q,s,s'_0} & \longrightarrow & \mathcal{C}[[x]]_{q,s,s'_1} \\ \downarrow & & \downarrow \\ \mathcal{C}[[x]]_{(q,s,s'_1)} & \longrightarrow & \mathcal{C}[[x]]_{(q,s,s'_1)} \end{array}$$

L'injection naturelle $\mathcal{C}[[x]]_{q,s,s'_0} \longrightarrow \mathcal{C}[[x]]_{(q,s,s'_1)}$ est d'image dense. En appliquant le lemme 2.5 on obtient le résultat souhaité.

Proposition 4.7.7 : Soit $[s_0, s_1] \subset [0, +\infty[$ tel que pour tout $s \in]s_0, s_1[$, s n'est pas exceptionnelle. Alors il existe s'_0 et $s'_1 \in \mathbb{R}$ tels que $\forall s' < s'_0$, $\forall s'' > s'_1$,

$$\dim \text{Ker} \left(L, \frac{\mathcal{C}[[x]]_{(q,s_1,s'_1)}}{\mathcal{C}[[x]]_{q,s_0,s'_0}} \right) = 0$$

Démonstration : Le principe est le même en utilisant le résultat de la proposition 4.7.5.

Proposition 4.7.8 : Il existe $s > 0$ tel que pour tout $s' \in \mathbb{R}$:

$$\dim \text{Ker} \left(L, \frac{\mathcal{C}[[x]]}{\mathcal{C}[[x]]_{q,s,s'}} \right) = 0$$

Démonstration : L'opérateur L est à indice dans $\mathcal{C}[[x]]$ et $\chi(L, \mathcal{C}[[x]]) = -\inf\{v(a_{i,j})\} = -k_0$ (théorème 4.5.1). Soient i_0 et j_0 tels que $k_0 = v(a_{i_0,j_0})$. Si $s \rightarrow +\infty$ alors pour tout (i, j, k) tel que $\alpha_{i,j,k} \neq 0$ $k_0 \leq k$ et $(i_0, j_0, k_0) \in M(s)$ et s n'est pas exceptionnelle. Donc (i_0, j_0, k_0) est l'unique élément de $M(s)$: $\chi_{q,s,s'} = -k_0$ pour tout $s' \in \mathbb{R}$.

Théorème 4.7.9 : Soit $L \in \mathbf{L}$. Soient $s_1 > s_2 > \dots > s_l$ les éventuelles valeurs exceptionnelles ≥ 0 pour L . Pour chacune d'elles, s_p , soient $s'_{p,1} > s'_{p,2} > \dots > s'_{p,l_p}$ les (éventuelles) valeurs exceptionnelles relativement à s_p (si $s_p = 0$ on se limite aux s' -valeurs positives ou nulles). Soient $\hat{f} \in \mathcal{C}[[x]]$ et $g \in \mathcal{C}\{x\}$ tels que $L(\hat{f}) = g$. Alors :

- Soit $\hat{f} \in \mathcal{C}\{x\}$,
- Soit il existe un unique réel $s' > 0$ tel que $\hat{f} \in \mathcal{C}[[x]]_{q,0,s'}$ et $\hat{f} \notin \mathcal{C}[[x]]_{(q,0,s')}$: on dit que \hat{f} est Gevrey d'ordre exact s' ,
- Soit il existe un unique réel $s > 0$ et un unique réel $s' \in \mathbb{R}$ tels que $\hat{f} \in \mathcal{C}[[x]]_{q,s,s'}$ et $\hat{f} \notin \mathcal{C}[[x]]_{(q,s,s')}$: on dit alors que \hat{f} est q -Gevrey d'ordres exacts s et s' .

Démonstration : On utilise toutes les propositions précédentes. Ainsi Γ comme $\hat{f} \in \mathcal{C}[[x]]$ on en déduit que $\hat{f} \in \mathcal{C}[[x]]_{q,s_1,s'_{p,1}}$. Si $\hat{f} \notin \mathcal{C}[[x]]_{(q,s_1,s'_{p,1})}$ on s'arrête. Sinon $\hat{f} \in \mathcal{C}[[x]]_{q,s_1,s'_{1,2}}$. Si $\hat{f} \notin \mathcal{C}[[x]]_{(q,s_1,s'_{1,2})}$ on s'arrête sinon on recommence. L'ensemble des valeurs exceptionnelles relativement à s_1 est ainsi parcouru. Si on ne s'est pas arrêté à la dernière valeur s'_{1,l_1} alors $\hat{f} \in \mathcal{C}[[x]]_{(q,s_1,s'_{1,l_1})}$ donc $\hat{f} \in \mathcal{C}[[x]]_{q,s_2,s'_{2,1}}$. On recommence.

Si on a parcouru toutes les valeurs exceptionnelles strictement positives sans s'arrêter : $\hat{f} \in \mathcal{C}[[x]]_{q,0,s'_1}$. On parcourt suivant le même procédé les éventuelles s' -valeurs exceptionnelles positives relativement à 0. Si on ne s'est toujours pas arrêté alors $\hat{f} \in \mathcal{C}[[x]]_{q,0,0} = \mathcal{C}\{x\}$.

Unicité de s et s' : Supposons que $\hat{f} \in \mathcal{C}[[x]]_{q,s,s'}$ et que $\hat{f} \notin \mathcal{C}[[x]]_{(q,s,s')}$ pour $s > 0$ et $s' \in \mathbb{R}$ (troisième cas).

- Si $\bar{s} > s$ et \bar{s}' quelconque : $\hat{f} \in \mathcal{C}[[x]]_{q,s,s'} \subset \mathcal{C}[[x]]_{(q,\bar{s},\bar{s}')}.$ On ne peut donc pas avoir $\hat{f} \in \mathcal{C}[[x]]_{q,\bar{s},\bar{s}'}$ et $\hat{f} \notin \mathcal{C}[[x]]_{(q,\bar{s},\bar{s}')}.$
- Si $\bar{s} < s$ et \bar{s}' quelconque : Si $\hat{f} \in \mathcal{C}[[x]]_{q,\bar{s},\bar{s}'}$ et $\hat{f} \notin \mathcal{C}[[x]]_{(q,\bar{s},\bar{s}')}.$ On aurait $\hat{f} \in \mathcal{C}[[x]]_{(q,s,s')}$

La démarche est la même pour montrer l'unicité de s' dans le deuxième cas. On laisse au lecteur le soin de généraliser ce théorème pour L de la forme (1).

5. Interprétation des théorèmes d'indices

5.1. Polygone de Newton

On va généraliser les polygones de Newton définis par J.-P. Ramis [10Π1] aux opérateurs du type :

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(x \frac{d}{dx}\right)^i \sigma_q^j$$

(ou plus généralement de la forme (1)) Γ avec $a_{i,j}(x) = \sum_{k \geq k_0} \alpha_{i,j,k} x^k \in \mathbb{C}\{x\}[x^{-1}]$ ($k_0 \in \mathbb{Z}$).

On définit dans \mathbb{R}^2 les quadrants $\tau_1 = \{(u, v) \in \mathbb{R}^2 / u, v \geq 0\}$ $\tau_2 = \{(u, v) \in \mathbb{R}^2 / u \leq 0, v \geq 0\}$ $\tau_3 = \{(u, v) \in \mathbb{R}^2 / u, v \leq 0\}$ et $\tau_4 = \{(u, v) \in \mathbb{R}^2 / u \geq 0, v \leq 0\}$.

Pour $(a, b) \in \mathbb{R}^2$ et $r = 1 \dots 4$ on pose $\tau_r(a, b) = (a, b) + \tau_r \Gamma$ $M_r^q(L) = \cup \tau_r(j, k) \Gamma$ union étendue aux couples (j, k) pour lesquels il existe i tel que $\alpha_{i,j,k} \neq 0$ (pour L de la forme (1)) $M_r^q(L) = \cup \tau_r(j, k - i)$. Soit $P_r^q(L)$ l'enveloppe convexe dans \mathbb{R}^2 de $M_r(L)$. Soit $N_l^q(L) = P_1^q(L) \cap P_2^q(L) \Gamma$ soit $N_u^q(L) = P_3^q(L) \cap P_4^q(L)$ et $N_q(L) = N_l^q(L) \cap N_u^q(L)$.

Définition 5.1.1 : On appelle $N_l^q(L)$ le q -polygone de Newton inférieur associé à L et $N_u^q(L)$ le q -polygone de Newton supérieur associé à L ; $N_q(L)$ est le q -polygone de Newton associé à L .

Remarque : $N_q(L)$ est le polygone de Newton de l'opérateur aux q -différences

$$\sum_{j=0}^J \sum_{\{k/\exists i, \alpha_{i,j,k} \neq 0\}} x^k \sigma_q^j$$

(ou plus généralement de : $\sum_j \sum_i \sum_k x^{k-i} \sigma_q^j$).

Remarque : Si un $a_{i,j}(x) \notin \mathbb{C}[x, x^{-1}]$ alors $N_q(L) = N_l^q(L)$. Si tous les $a_{i,j}(x) \in \mathbb{C}[x, x^{-1}] \Gamma$ $N_q(L)$ est l'enveloppe convexe des points $\{(j, k) / \exists i, \alpha_{i,j,k} \neq 0\}$.

On définit un *sens de parcours* des bords de ce q -polygone (pour plus de détails on se reportera à [11]).

Cas $|q| > 1$: Si $\alpha \in \mathbb{R}^*$ et si l'intersection de $N_l^q(L)$ avec la droite d'appui de pente α est réduite à un point unique on note $(j(\alpha), k(\alpha))$ ce point. Si au contraire α est l'une des pentes $\neq 0$ de $N_q(L) \Gamma$ l'intersection est un segment de la forme $[(j_1(\alpha), k_1(\alpha)), (j_2(\alpha), k_2(\alpha))]$ avec $j_1(\alpha) < j_2(\alpha)$. Le segment horizontal de $N_l^q(L)$ est noté $[(j_1(0_l), k_1(0_l)), (j_2(0_l), k_2(0_l))]$ avec $j_1(0_l) < j_2(0_l)$ et $k_1(0_l) = k_2(0_l)$. Si $N_l^q(L)$ n'a pas de pente nulle on note $(j(0_l), k(0_l))$ le point *le plus bas* de $N_l^q(L)$. On note $(j_1(\infty), k_1(\infty))$ le point situé sur le côté vertical droit de $N_l^q(L)$ d'ordonnée la plus petite. On note $(j_2(-\infty), k_2(-\infty))$ le point situé sur le côté vertical

gauche de $N_l^q(L)$ d'ordonnée la plus petite.

Si tous les $a_{i,j}(x)$ sont dans $\mathcal{C}[x, x^{-1}]$: Si $\alpha \in \mathbb{R}^*$ et si l'intersection de $N_u^q(L)$ est réduite à un unique point Γ soit $(j(\alpha), k(\alpha))$ ce point. Si $\alpha \in \mathbb{R}^*$ est l'une des pentes de $N_u^q(L)$ Γ on note $[(j_1(\alpha), k_1(\alpha)), (j_2(\alpha), k_2(\alpha))]$ le segment Γ avec $j_1(\alpha) > j_2(\alpha)$. Le segment horizontal de $N_u^q(L)$ est noté $[(j_1(0_u), k_1(0_u)), (j_2(0_u), k_2(0_u))]$ avec $j_1(0_u) > j_2(0_u)$. S'il n'existe pas Γ on note $(j(0_u), k(0_u))$ le point *le plus haut* de $N_u^q(L)$. On note $(j_2(\infty), k_2(\infty))$ le point situé sur le côté vertical droit de $N_u^q(L)$ d'ordonnée la plus grande. Soit $(j_1(-\infty), k_1(-\infty))$ le point situé sur le côté vertical gauche de $N_u^q(L)$ d'ordonnée la plus grande.

Cas $|q| < 1$: Si $\alpha \in \mathbb{R}^*$ et si l'intersection de $N_l^q(L)$ avec la droite d'appui de pente $-\alpha$ est réduite à un point unique Γ on note $(j(\alpha), k(\alpha))$ ce point. Si au contraire $-\alpha$ est une des pentes $\neq 0$ de $N_l^q(L)$ Γ l'intersection est un segment de la forme $[(j_1(\alpha), k_1(\alpha)), (j_2(\alpha), k_2(\alpha))]$ avec $j_1(\alpha) > j_2(\alpha)$. Le segment horizontal de $N_l^q(L)$ est noté $[(j_1(0_l), k_1(0_l)), (j_2(0_l), k_2(0_l))]$ avec $j_1(0_l) > j_2(0_l)$ et $k_1(0_l) = k_2(0_l)$. Si $N_l^q(L)$ n'a pas de pente nulle Γ on note $(j(0_l), k(0_l))$ le point *le plus bas* de $N_l^q(L)$. On note $(j_1(\infty), k_1(\infty))$ le point situé sur le côté vertical gauche de $N_l^q(L)$ d'ordonnée la plus petite. On note $(j_2(-\infty), k_2(-\infty))$ le point situé sur le côté vertical droit de $N_l^q(L)$ d'ordonnée la plus petite.

Si tous les $a_{i,j}(x)$ sont dans $\mathcal{C}[x, x^{-1}]$: Si $\alpha \in \mathbb{R}^*$ et si l'intersection de $N_u^q(L)$ avec la droite de contact de pente $-\alpha$ est réduite à un unique point Γ soit $(j(\alpha), k(\alpha))$ ce point. Si $-\alpha \in \mathbb{R}^*$ est l'une des pentes de $N_u^q(L)$ Γ on note $[(j_1(\alpha), k_1(\alpha)), (j_2(\alpha), k_2(\alpha))]$ le segment Γ avec $j_1(\alpha) < j_2(\alpha)$. Le segment horizontal de $N_u^q(L)$ est noté $[(j_1(0_u), k_1(0_u)), (j_2(0_u), k_2(0_u))]$ avec $j_1(0_u) < j_2(0_u)$. S'il n'existe pas Γ on note $(j(0_u), k(0_u))$ le point *le plus haut* de $N_u^q(L)$. On note $(j_2(\infty), k_2(\infty))$ le point situé sur le côté vertical gauche de $N_u^q(L)$ d'ordonnée la plus grande. Soit $(j_1(-\infty), k_1(-\infty))$ le point situé sur le côté vertical droit de $N_u^q(L)$ d'ordonnée la plus grande.

Pour chaque segment (ou demi-droite) non horizontal de la frontière de $N_l^q(L)$ Γ repéré par $[(j_1(\alpha), k_1(\alpha)), (j_2(\alpha), k_2(\alpha))]$ avec $\alpha \in \mathbb{R}^* \cup \{\infty\} \cup \{-\infty\}$ Γ on construit un d -polygone de la façon suivante. A chaque point (j, k) de ce segment Γ on reporte sur l'orthogonale au segment passant par (j, k) et vers l'extérieur du q -polygone l'ensemble des indices i pour lesquels $\alpha_{i,j,k} \neq 0$; on définit en fait localement un nouveau repère orthonormé. L'axe des ordonnées *locales* Γ porté par le segment considéré Γ est orienté et gradué de telle sorte que les ordonnées *locales* soient identiques aux ordonnées *globales*; l'axe des abscisses *locales* i est tel que le demi-plan *local* $\{i > 0\}$ ne rencontre aucun point du q -polygone de Newton. Soit $P_{d,\alpha}^+(L)$ l'enveloppe convexe de la réunion des quadrants $\{(u, v) / 0 \leq u \leq i, v \geq k\}$ et soit $P_{d,\alpha}^-(L)$ celle des quadrants $\{(u, v) / 0 \leq u \leq i, v \leq k\}$ Γ les réunions étant faites sur l'ensemble $\{(i, k) / \exists j, \alpha_{i,j,k} \neq 0 \text{ et } (j, k) \in [(j_1(\alpha), k_1(\alpha)), (j_2(\alpha), k_2(\alpha))]\}$.

Définition 5.1.2 : On appelle d -polygone associé à la valeur α ($\alpha \in \mathbb{R}^* \cup \{\infty\}$) le convexe : $P_{d,\alpha}^+(L) \cap P_{d,\alpha}^-(L)$.

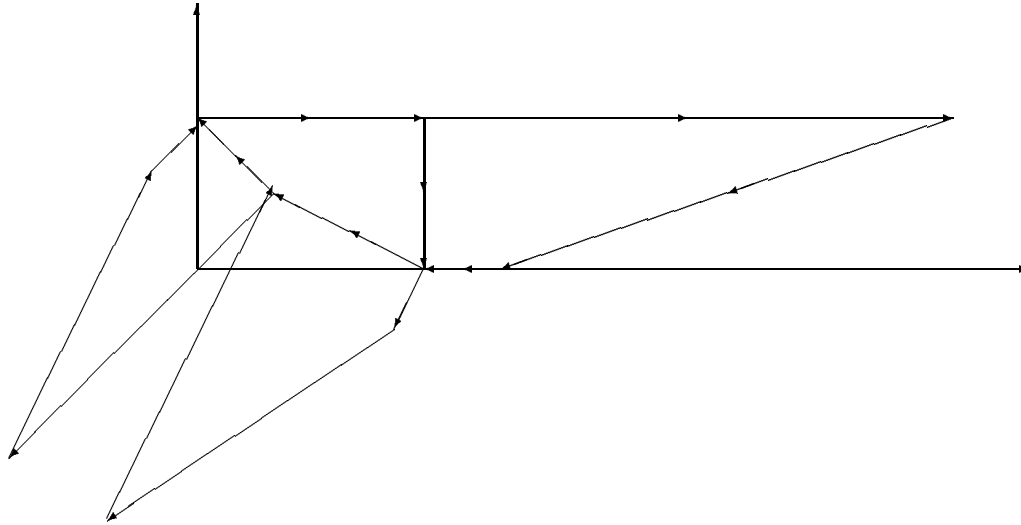
On définit également un sens de parcours sur chaque d -polygone [10] Γ dont les pentes sont définies dans le repère *local*. Si α' est une pente strictement positive du d -polygone associé à α Γ on note le segment $[(i_1(\alpha, \alpha'), k_1(\alpha, \alpha')), (i_2(\alpha, \alpha'), k_2(\alpha, \alpha'))]$ avec $k_1(\alpha, \alpha') - k_2(\alpha, \alpha')$ du

même signe que $k_1(\alpha) - k_2(\alpha)$. Si α' est une pente < 0 on note de la même façon le segment. Le côté vertical du d -polygone est noté $[(i_1(\alpha, \alpha'), k_1(\alpha, \alpha')), (i_2(\alpha, \alpha'), k_2(\alpha, \alpha'))]$ avec $i_1(\alpha, \infty) = i_2(\alpha, \infty)$ et $k_1(\alpha, \alpha') - k_2(\alpha, \alpha')$ du même signe que $k_1(\alpha) - k_2(\alpha)$. Le segment horizontal inférieur est repéré par : $i_1(\alpha, 0_+) = 0 < i_2(\alpha, 0_+)$ et $k_1(\alpha, 0_+) = k_2(\alpha, 0_+) = k_1(\alpha)$. Le segment horizontal supérieur est repéré par : $i_1(\alpha, 0_-) > i_2(\alpha, 0_-) = 0$ et $k_1(\alpha, 0_-) = k_2(\alpha, 0_-) = k_2(\alpha)$. Enfin si l'intersection du d -polygone avec la droite de contact de pente α' est réduite à un unique point on le note $(i(\alpha, \alpha'), k(\alpha, \alpha'))$.

Remarque : Le d -polygone d'un opérateur du type (1) se construit suivant le même procédé; il suffit de remplacer partout k par $k - i$.

Exemple : Soit $L = (x \frac{d}{dx}) \sigma_q^3 + x^2 (x \frac{d}{dx}) + x (x \frac{d}{dx})^3 \sigma_q + x (x \frac{d}{dx})^5 \sigma_q + x^2 (x \frac{d}{dx})^7 \sigma_q^3$.

Les flèches indiquent l'origine et l'extrémité de chaque segment pour $0 < |q| < 1$.



5.2. Traduction des théorèmes d'indices en fonction du polygone de Newton

Soient $s \in \mathbb{R} \cup \{+\infty\}$ et $s' \in \mathbb{R}$. Si $s = +\infty$ on pose $\alpha = 0_l$. Si $s = 0$ on pose $\alpha = \infty$. Sinon on pose $\alpha = \frac{1}{s}$. Si $s' = 0$ on pose $\alpha' = \infty$. Sinon on pose $\alpha' = \frac{1}{s'}$.

Soit $L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) (x \frac{d}{dx})^i \sigma_q^j$ à coefficients dans $\mathcal{C}\{x\}$.

Les valeurs exceptionnelles sont les réels s pour lesquels les droites d'appui de pente α (cas $|q| > 1$) ou de pente $-\alpha$ (cas $|q| < 1$) rencontrent le polygone de Newton suivant un segment ou une demi-droite. On convient que $s = +\infty$ est une valeur exceptionnelle si le segment horizontal inférieur n'est pas de longueur nulle. Pour une valeur exceptionnelle $s \neq +\infty$ on définit de même les valeurs exceptionnelles $\in]-\infty, +\infty[$ relativement à s . Sur l'exemple précédent les valeurs exceptionnelles sont $s = 2$ et $s = 1$; $s' = 4$ est la valeur exceptionnelle associée à

$s = 2$ et $s' = -4$ est la valeur exceptionnelle associée à $s = 1$.

Théorème 5.2.1 : On pose $\mathcal{C}[[x]]_{q,+\infty,s'} = \mathcal{C}[[x]] \quad \forall s' \in \mathbb{R}$. D'autre part, $\mathcal{C}\{x\} = \mathcal{C}[[x]]_{q,0,0}$.

1. L'opérateur L est d'indice fini de : $\mathcal{C}[[x]]_{q,s,s'} \longrightarrow \mathcal{C}[[x]]_{q,s,s'}$ pour $s \in]0, +\infty]$ et $s' \in \mathbb{R}$, et pour $s = 0$ et $s' \in [0, +\infty[$:

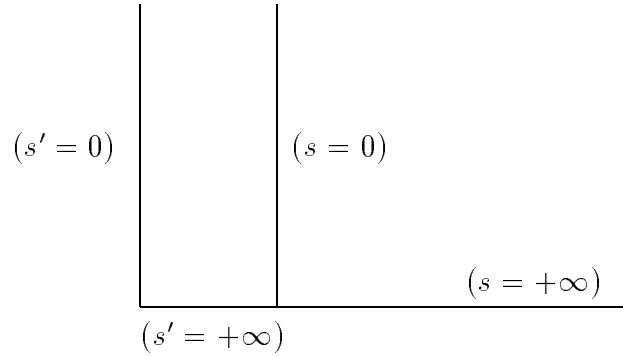
- Si s non exceptionnelle : $\chi_{q,s,s'} = -k(\alpha)$
- Si s exceptionnelle et s' non exceptionnelle : $\chi_{q,s,s'} = -k(\alpha, \alpha')$
- Si s exceptionnelle et s' exceptionnelle : $\chi_{q,s,s'} = -k_1(\alpha, \alpha')$

2. L'opérateur L est d'indice fini de : $\mathcal{C}[[x]]_{(q,s,s')} \longrightarrow \mathcal{C}[[x]]_{(q,s,s')}$ pour $s \in]0, +\infty[$ et $s' \in \mathbb{R}$, et pour $s = 0$ et $s' \in [0, +\infty[$:

- Si s non exceptionnelle : $\chi_{(q,s,s')} = -k(\alpha)$
- Si s exceptionnelle et s' non exceptionnelle : $\chi_{(q,s,s')} = -k(\alpha, \alpha')$
- Si s exceptionnelle et s' exceptionnelle : $\chi_{(q,s,s')} = -k_2(\alpha, \alpha')$

On laisse le soin au lecteur d'établir un théorème analogue pour L de la forme (1).

Théorème 5.2.2 : Soit $L \in \mathbf{L}$. Soit $g \in \mathcal{C}\{x\}$. Une condition nécessaire et suffisante pour que toutes les solutions de l'équation $L(\hat{f}) = g$ soient convergentes est que le polygone de Newton soit de la forme :



Démonstration : Il faut et il suffit qu'il n'y ait pas de valeurs exceptionnelles $s > 0$ ni de valeurs $s' > 0$ exceptionnelles relativement à $s = 0$.

6. Equations à coefficients polynomiaux

Nous considérons dans ce qui suit des opérateurs de $\mathcal{C}[x][x \frac{d}{dx}, \sigma_q]$ ($0 < |q| < 1$).

6.1. Dualité topologique

Si $\hat{f}(x) = \sum_{n \geq 0} a_n x^n \in \mathcal{C}[[x]]_{q,s,s'}$ et $\hat{g}(x) = \sum_{n \geq 1} b_n x^{-n} \in x^{-1}\mathcal{C}[[x^{-1}]]_{(q,s,s')}$ alors la série $\sum_{n \geq 0} a_n b_{n+1}$ converge. On pose $\langle \hat{f}, \hat{g} \rangle = \sum_{n=0}^{+\infty} a_n b_{n+1}$.

On désigne par $\Gamma^1(\mathcal{C}) = \mathcal{C} \cup \{\infty\}$ la sphère de Riemann. Nous rappelons le résultat suivant (voir [11]) dû à A. Grothendieck :

Proposition 6.1.1 :

(i) **L'accouplement**

$$\begin{aligned} \mathcal{C}\{x\} \times x^{-1}\mathcal{O}(\Gamma^1(\mathcal{C}) - \{0\}) &\rightarrow \mathcal{C} \\ (\hat{f}, \hat{g}) &\mapsto \langle \hat{f}, \hat{g} \rangle = \sum_{n=0}^{+\infty} a_n b_{n+1} \end{aligned}$$

($\hat{f}(x) = \sum_{n \geq 0} a_n x^n$ et $\hat{g}(x) = \sum_{n \geq 1} b_n x^{-n}$) met en dualité topologique l'espace $\mathcal{C}\{x\}$ muni de sa topologie DFN et l'espace $x^{-1}\mathcal{O}(\Gamma^1(\mathcal{C}) - \{0\})$ muni de sa topologie FN.

(ii) **L'accouplement**

$$\begin{aligned} \mathcal{C}[[x]] \times x^{-1}\mathcal{C}[[x^{-1}]] &\rightarrow \mathcal{C} \\ (\hat{f}, \hat{g}) &\mapsto \langle \hat{f}, \hat{g} \rangle \end{aligned}$$

met en dualité topologique l'espace FN $\mathcal{C}[[x]]$ et l'espace DFN $x^{-1}\mathcal{C}[[x^{-1}]]$.

L'isomorphisme

$$\begin{aligned} \Psi_{s,s'} : \mathcal{C}[[x]] &\rightarrow \mathcal{C}[[x]] \\ \sum_{n \geq 0} a_n x^n &\mapsto \sum_{n \geq 0} |q|^{\frac{sn(n+1)}{2}} (n!)^{-s'} a_n x^n \end{aligned}$$

induit des isomorphismes $\mathcal{C}[[x]]_{q,s,s'} \rightarrow \mathcal{C}\{x\}$ et $\mathcal{C}[[x]]_{(q,s,s')} \rightarrow A(\mathcal{C})$. On transporte sur $\mathcal{C}[[x]]_{q,s,s'}$ (resp. $\mathcal{C}[[x]]_{(q,s,s')}$) la topologie DFN (resp. FN) de $\mathcal{C}\{x\}$ (resp. $A(\mathcal{C})$).

L'isomorphisme

$$\begin{aligned} \Psi_{s,s'} : x^{-1}\mathcal{C}[[x^{-1}]] &\rightarrow x^{-1}\mathcal{C}[[x^{-1}]] \\ \sum_{n \geq 0} a_n x^{-n-1} &\mapsto \sum_{n \geq 0} |q|^{\frac{sn(n+1)}{2}} (n!)^{-s'} a_n x^{-n-1} \end{aligned}$$

induit des isomorphismes $x^{-1}\mathcal{C}[[x^{-1}]]_{q,s,s'} \rightarrow x^{-1}\mathcal{C}\{x^{-1}\}$ et $x^{-1}\mathcal{C}[[x^{-1}]]_{(q,s,s')} \rightarrow x^{-1}\mathcal{O}(\Gamma^1(\mathcal{C}) - \{0\})$. On transporte sur $x^{-1}\mathcal{C}[[x^{-1}]]_{q,s,s'}$ (resp. $x^{-1}\mathcal{C}[[x^{-1}]]_{(q,s,s')}$) la topologie DFN (resp. FN) de $x^{-1}\mathcal{C}\{x^{-1}\}$ (resp. $x^{-1}\mathcal{O}(\Gamma^1(\mathcal{C}) - \{0\})$).

Proposition 6.1.2 : Soient s et s' deux réels.

- (i) **Les espaces $\mathcal{C}[[x]]_{q,s,s'}$ et $x^{-1}\mathcal{C}[[x^{-1}]]_{(q,-s,-s')}$ sont en dualité topologique.**
- (ii) **Les espaces $\mathcal{C}[[x]]_{(q,s,s')}$ et $x^{-1}\mathcal{C}[[x^{-1}]]_{q,-s,-s'}$ sont en dualité topologique.**
- (iii) **Les espaces $\mathcal{C}[[x]]$ et $x^{-1}\mathcal{C}[[x^{-1}]]$ sont en dualité topologique.**

Le transposé de l'opérateur x considéré comme agissant sur $\mathcal{C}[[x]]$ est l'opérateur x^* agissant sur $x^{-1}\mathcal{C}[[x^{-1}]]$ défini par $x^*(\sum_{n \geq 0} a_n x^{-n-1}) = \sum_{n \geq 0} a_{n+1} x^{-n-1}$. Le transposé de l'opérateur $\frac{d}{dx}$ est

l'opérateur $-\frac{d}{dx}$. Le dual de l'opérateur $x\frac{d}{dx}$ est l'opérateur $-\frac{d}{dx}x^*$ qui coïncide avec l'opérateur $-\frac{d}{dx}x = -1 - x\frac{d}{dx}$. Le dual de σ_q est l'opérateur $p\sigma_p$.

$\mathcal{C}[[x]]$	$x^{-1}\mathcal{C}[[x^{-1}]]$
$\mathcal{C}[[x]]_{q,s,s'}$	$x^{-1}\mathcal{C}[[x^{-1}]]_{(q,-s,-s')}$
$\mathcal{C}[[x]]_{(q,s,s')}$	$x^{-1}\mathcal{C}[[x^{-1}]]_{q,-s,-s'}$
x	x^*
$\frac{d}{dx}$	$-\frac{d}{dx}$
$x\frac{d}{dx}$	$-1 - x\frac{d}{dx}$
σ_q	$p\sigma_p$

Remarque : Soit $\hat{f}(x) = \sum_{n \geq 0} a_n x^{-n-1} \in x^{-1}\mathcal{C}[[x^{-1}]]$. Alors $x\hat{f}(x) = \sum_{n \geq 0} a_n x^{-n} = a_0 + \sum_{n \geq 0} a_{n+1} x^{-n-1} = a_0 + x^* \hat{f}(x)$. L'opérateur $x^* : x^{-2}\mathcal{C}[[x^{-1}]] \rightarrow x^{-1}\mathcal{C}[[x^{-1}]]$ coïncide avec l'opérateur $x : x^{-2}\mathcal{C}[[x^{-1}]] \rightarrow x^{-1}\mathcal{C}[[x^{-1}]]$. On établit d'autre part que $\theta x^* = x^*(1 + \theta)$.

En utilisant le théorème 5.2.1 et le théorème 2.3 nous obtenons le résultat suivant :

Théorème 6.1.3 : Soit $L \in \mathcal{C}[x][x\frac{d}{dx}, \sigma_q]$ ($0 < |q| < 1$). Soit $L^* \in \mathcal{C}[x\frac{d}{dx}, \sigma_p][x^*] = \mathcal{C}[x^*][x\frac{d}{dx}, \sigma_p]$ son dual. On pose $x^{-1}\mathcal{C}[[x^{-1}]]_{(q,-\infty,s')} = x^{-1}\mathcal{C}[[x^{-1}]] \forall s' \in \mathbb{R}$.

(i) L'opérateur $L^* : x^{-1}\mathcal{C}[[x^{-1}]]_{q,s,s'} \rightarrow x^{-1}\mathcal{C}[[x^{-1}]]_{q,s,s'}$ est de Fredholm pour $s \in]-\infty, 0[, s' \in \mathbb{R}$ et pour $s = 0, s' \in]-\infty, 0[$, d'indice égal à $-\chi(L, \mathcal{C}[[x]]_{(q,-s,-s')})$.

(ii) L'opérateur $L^* : x^{-1}\mathcal{C}[[x^{-1}]]_{(q,s,s')} \rightarrow x^{-1}\mathcal{C}[[x^{-1}]]_{(q,s,s')}$ est de Fredholm pour $s \in [-\infty, 0[, s' \in \mathbb{R}$ et pour $s = 0, s' \in]-\infty, 0[$, d'indice égal à $-\chi(L, \mathcal{C}[[x]]_{q,-s,-s'})$.

Soit $L = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \alpha_{i,j,k} x^k (x\frac{d}{dx})^i \sigma_p^j$. Le transposé de L est alors l'opérateur $L^* = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \alpha_{i,j,k} p^j \sigma_p^j (-1 - x\frac{d}{dx})^i (x^*)^k = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \alpha_{i,j,k} p^{j(k+1)} (x^*)^k \sigma_p^j (-1 - k - x\frac{d}{dx})^i$. L'opérateur $L^* : x^{-K-1}\mathcal{C}[[x^{-1}]] \rightarrow x^{-1}\mathcal{C}[[x^{-1}]]$ est un opérateur de $\mathcal{C}[x][x\frac{d}{dx}, \sigma_p]$. Le p -polygone de Newton de L^* est par définition celui de $\sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \alpha_{i,j,k} p^{j(k+1)} x^k \sigma_p^j (-1 - k - x\frac{d}{dx})^i$ l'opérateur que l'on note encore L^* .

Propriété 6.1.4 : Le q -polygone de Newton de L coïncide avec le p -polygone de Newton de L^* .

Démonstration : On montre facilement que des opérateurs élémentaires de la forme $L_{a,b,c}^* = \sigma_p^b (\frac{d}{dx}x)^a x^c$ peuvent se mettre sous la forme : $L_{a,b,c}^* = \sum_{i=0}^a \beta_{i,b,c} x^c (x\frac{d}{dx})^i \sigma_p^b$ avec $\beta_{a,b,c} = p^{bc}$ non nul et $\beta_{i,b,c} = p^{bc} C_a^i (c+1)^{a-i}, i = 0 \dots a$. Donc si $L = \sum_{a=0}^A \sum_{b=0}^B \sum_{c \geq 0} \alpha_{a,b,c} x^c (x\frac{d}{dx})^a \sigma_p^b$ alors

$$L^* = \sum_{i=0}^A \sum_{b=0}^B b_{i,b}(x) (x\frac{d}{dx})^i \sigma_p^b$$

avec $b_{i,b}(x) = \sum_{c \geq 0} (\sum_{a=i}^A (-1)^a \alpha_{a,b,c} p^{bc} C_a^i (c+1)^{a-i}) x^c$. On a ajouté des termes mais ceux-ci ne jouent aucun rôle lors de la construction du polygone généralisé associé à L^* .

Théorème 6.1.5 : Soient $L \in \mathcal{C}[x][x \frac{d}{dx}, \sigma_q]$ ($0 < |q| < 1$) et $L^* \in \mathcal{C}[x^*][x \frac{d}{dx}, \sigma_p]$ son dual. Soit K le degré en x de L . Alors :

(i) L'opérateur $L^* : x^{-1-K} \mathcal{C}[[x^{-1}]]_{q,s,s'} \rightarrow x^{-1} \mathcal{C}[[x^{-1}]]_{q,s,s'}$ est de Fredholm pour $s \in]-\infty, 0[, s' \in \mathbb{R}$ et pour $s = 0, s' \in]-\infty, 0[$, d'indice égal à $-\chi(L, \mathcal{C}[[x]]_{(q,-s,-s')}) - K$.

(ii) L'opérateur $L^* : x^{-1-K} \mathcal{C}[[x^{-1}]]_{(q,s,s')} \rightarrow x^{-1} \mathcal{C}[[x^{-1}]]_{(q,s,s')}$ est de Fredholm pour $s \in]-\infty, 0[, s' \in \mathbb{R}$ et pour $s = 0, s' \in]-\infty, 0[$, d'indice égal à $-\chi(L, \mathcal{C}[[x]]_{q,-s,-s'}) - K$.

Démonstration : En considérant le diagramme commutatif

$$\begin{array}{ccccccc} 0 \rightarrow & x^{-1-K} \mathcal{C}[[x^{-1}]]_{q,s,s'} & \rightarrow & x^{-1} \mathcal{C}[[x^{-1}]]_{q,s,s'} & \rightarrow & x^{-1} \mathcal{C}[[x^{-1}]]_{q,s,s'} / x^{-1-K} \mathcal{C}[[x^{-1}]]_{q,s,s'} & \rightarrow 0 \\ & L^* \downarrow & & L^* \downarrow & & \downarrow & \\ 0 \rightarrow & x^{-1} \mathcal{C}[[x^{-1}]]_{q,s,s'} & \rightarrow & x^{-1} \mathcal{C}[[x^{-1}]]_{q,s,s'} & \rightarrow & \{0\} & \rightarrow 0 \end{array}$$

et le théorème 6.1.3 nous démontrons (i). L'assertion (ii) se démontre de la même façon.

6.2. Etude à l'infini

Soient $L \in \mathcal{C}[x][x \frac{d}{dx}, \sigma_q]$ ($0 < |q| < 1$) et L^* son dual. Nous allons étudier L^* à l'infini. Pour cela nous posons $z = \frac{1}{x}$. Soit $L_z^* \in \mathcal{C}[z^{-1}][z \frac{d}{dz}, \sigma_q]$ l'opérateur obtenu à partir de L^* en remplaçant formellement x^* par $\frac{1}{z} \Gamma x \frac{d}{dx}$ par $-z \frac{d}{dz}$ et σ_p par σ_q .

Propriété 6.2.1 : Le q -polygone de Newton de L_z^* et le p -polygone de Newton de L^* sont symétriques par rapport à l'axe des abscisses.

Soit $\phi : x^{-1} \mathcal{C}[[x^{-1}]] \rightarrow z \mathcal{C}[[z]]$ l'isomorphisme qui à x^{-n} associe z^n . Il induit des isomorphismes $\phi : x^{-1-K} \mathcal{C}[[x^{-1}]]_{q,s,s'} \rightarrow z^{K+1} \mathcal{C}[[z]]_{q,s,s'}$ et $\phi : x^{-1-K} \mathcal{C}[[x^{-1}]]_{(q,s,s')} \rightarrow z^{K+1} \mathcal{C}[[z]]_{(q,s,s')}$ ($K \in \mathbb{N}$). Alors $L_z^* \phi = \phi L^*$ et nous obtenons le théorème :

Théorème 6.2.2 : Soient $L \in \mathcal{C}[x][x \frac{d}{dx}, \sigma_q]$ ($0 < |q| < 1$) et $L_z^* \in \mathcal{C}[z^{-1}][z \frac{d}{dz}, \sigma_q]$ son dual à l'infini. Soit K le degré en x de L . Alors :

(i) L'opérateur $L_z^* : z^{1+K} \mathcal{C}[[z]]_{q,s,s'} \rightarrow z \mathcal{C}[[z]]_{q,s,s'}$ est de Fredholm pour $s \in]-\infty, 0[, s' \in \mathbb{R}$ et pour $s = 0, s' \in]-\infty, 0[$, d'indice égal à $-\chi(L, \mathcal{C}[[x]]_{(q,-s,-s')}) - K$.

(ii) L'opérateur $L_z^* : z^{1+K} \mathcal{C}[[z]]_{(q,s,s')} \rightarrow z \mathcal{C}[[z]]_{(q,s,s')}$ est de Fredholm pour $s \in]-\infty, 0[, s' \in \mathbb{R}$ et pour $s = 0, s' \in]-\infty, 0[$, d'indice égal à $-\chi(L, \mathcal{C}[[x]]_{q,-s,-s'}) - K$.

Si $L = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \alpha_{i,j,k} x^k (x \frac{d}{dx})^i \sigma_q^j$ alors $L_z^* = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \alpha_{i,j,k} p^{j(k+1)} z^{-k} (-k - 1 + z \frac{d}{dz})^i \sigma_q^j$ et l'opérateur $z^K L_z^* \in \mathcal{C}[z][z \frac{d}{dz}, \sigma_q]$. L'opérateur $z^K L_z^* : z^{K+1} \mathcal{C}[[z]]_{q,s,s'} \rightarrow z^{K+1} \mathcal{C}[[z]]_{q,s,s'}$ est de Fredholm pour $s \in]-\infty, 0[, s' \in \mathbb{R}$ et pour $s = 0, s' \in]-\infty, 0[$ d'indice égal à $-\chi(L, \mathcal{C}[[x]]_{(q,-s,-s')}) - K$. On montre aisément qu'alors l'opérateur $z^K L_z^* : \mathcal{C}[[z]]_{q,s,s'} \rightarrow \mathcal{C}[[z]]_{q,s,s'}$ est de Fredholm pour $s \in]-\infty, 0[, s' \in \mathbb{R}$ et pour $s = 0, s' \in]-\infty, 0[$ d'indice

égal à $-\chi(L, \mathcal{C}[[x]]_{(q, -s, -s')}) - K$. Le q -polygone de Newton de $z^K L_z^*$ est l'image du q -polygone de L_z^* par la translation $(u, v) \mapsto (u, v + K)$. Nous pouvons alors exprimer les indices de $\chi(L, \mathcal{C}[[x]]_{(q, -s, -s')})$ et $\chi(L, \mathcal{C}[[x]]_{(q, -s, -s')})$ en fonction du q -polygone de Newton de $z^K L_z^*$.

Soient $s \in [-\infty, 0]$ et $s' \in \mathbb{R}$. Si $s = -\infty$ on pose $\alpha = 0_u$. Si $s = 0$ on pose $\alpha = \infty$. Si $s \in]-\infty, 0[$ on pose $\alpha = \frac{1}{s}$. Si $s' = 0$ on pose $\alpha' = \infty$. Si $s' \in \mathbb{R}^*$ on pose $\alpha' = \frac{1}{s'}$.

Si $s = -\infty$ on pose $\mathcal{C}[[z]]_{(q, -\infty, s')} = \mathcal{C}[z] \forall s' \in \mathbb{R}$.

Théorème 6.2.3 : Soient $L \in \mathcal{C}[x][x^{\frac{d}{dx}}, \sigma_q]$ ($0 < |q| < 1$) et $L_z^* \in \mathcal{C}[z^{-1}][z^{\frac{d}{dz}}, \sigma_q]$ son dual à l'infini. Soit K le degré en x de L . Alors

(i) L'opérateur $z^K L_z^* : \mathcal{C}[[z]]_{(q, s, s')} \rightarrow \mathcal{C}[[z]]_{(q, s, s')}$ est de Fredholm pour $s \in]-\infty, 0[$, $s' \in \mathbb{R}$ et pour $s = 0$, $s' \in]-\infty, 0[$, d'indice égal à $-k(\alpha)$ si s est non exceptionnelle, $-k(\alpha, \alpha')$ si s est exceptionnelle et si s' est non exceptionnelle, $-k_1(\alpha, \alpha')$ si s est exceptionnelle et si s' est exceptionnelle.

(ii) L'opérateur $z^K L_z^* : \mathcal{C}[[z]]_{(q, s, s')} \rightarrow \mathcal{C}[[z]]_{(q, s, s')}$ est de Fredholm pour $s \in [-\infty, 0[$, $s' \in \mathbb{R}$ et pour $s = 0$, $s' \in]-\infty, 0[$, d'indice égal à $-k(\alpha)$ si s est non exceptionnelle, $-k(\alpha, \alpha')$ si s est exceptionnelle et si s' est non exceptionnelle, $-k_2(\alpha, \alpha')$ si s est exceptionnelle et si s' est exceptionnelle.

Nous avons ainsi obtenu des théorèmes d'indices pour les opérateurs de $\mathcal{C}[z][z^{\frac{d}{dz}}, \sigma_q]$ agissant sur $\mathcal{C}[[z]]_{(q, s, s')}$ (resp. $\mathcal{C}[[z]]_{(q, s, s')}$) avec $s \in]-\infty, 0[$, $s' \in \mathbb{R}$ ou $s = 0$, $s' \in]-\infty, 0[$ (resp. $s \in [-\infty, 0[$, $s' \in \mathbb{R}$ ou $s = 0$, $s' \in]-\infty, 0[$).

6.3. Théorèmes d'indices

En combinant le théorème 5.2.1 et le théorème 6.2.3 nous obtenons le résultat suivant :

Théorème 6.3.1 : Soit $L \in \mathcal{C}[x][x^{\frac{d}{dx}}, \sigma_q]$ ($0 < |q| < 1$). Soient $s \in \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$ et $s' \in \mathbb{R}$. Si $s = 0$, on pose $\alpha = \infty$. Si $s = -\infty$, on pose $\alpha = 0_u$. Si $s = +\infty$, on pose $\alpha = 0_l$. Si $s \in \mathbb{R}^*$, on pose $\alpha = \frac{1}{s}$. Si $s' = 0$, on pose $\alpha' = \infty$. Si $s' \in \mathbb{R}^*$, on pose $\alpha' = \frac{1}{s'}$. On pose $\mathcal{C}[[x]]_{(q, -\infty, s')} = \mathcal{C}[x]$ et $\mathcal{C}[[x]]_{(q, +\infty, s')} = \mathcal{C}[[x]] \forall s' \in \mathbb{R}$.

(i) L'opérateur $L : \mathcal{C}[[x]]_{(q, s, s')} \rightarrow \mathcal{C}[[x]]_{(q, s, s')}$ est de Fredholm pour $s \in]-\infty, +\infty[$ et $s' \in \mathbb{R}$, d'indice égal à $-k(\alpha)$ si s est non exceptionnelle, $-k(\alpha, \alpha')$ si s est exceptionnelle et si s' est non exceptionnelle, $-k_1(\alpha, \alpha')$ si s est exceptionnelle et si s' est exceptionnelle.

(ii) L'opérateur $L : \mathcal{C}[[x]]_{(q, s, s')} \rightarrow \mathcal{C}[[x]]_{(q, s, s')}$ est de Fredholm pour $s \in [-\infty, +\infty[$ et $s' \in \mathbb{R}$, d'indice égal à $-k(\alpha)$ si s est non exceptionnelle, $-k(\alpha, \alpha')$ si s est exceptionnelle et si s' est non exceptionnelle, $-k_2(\alpha, \alpha')$ si s est exceptionnelle et si s' est exceptionnelle.

De la même façon que nous avons démontré le théorème 4.7.9 On obtient le :

Théorème 6.3.2 : Soit L un opérateur à coefficients polynomiaux du type

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(\frac{d}{dx} \right)^i \sigma_q^j$$

avec $0 < |q| < 1$. Soient $f \in A(\mathbb{C})$ une fonction entière et $g \in \mathbb{C}[x]$ tels que $L(f) = g$. Alors :

- Soit $f \in \mathbb{C}[x]$,
- Soit il existe un unique réel $s' < 0$ tel que $f \in \mathbb{C}[[x]]_{q,0,s'}$ et $f \notin \mathbb{C}[[x]]_{(q,0,s')}$; $\frac{1}{s'}$ est alors l'une des pentes strictement négatives du d -polygone associé à la valeur exceptionnelle $s = 0$,
- Soit il existe un unique réel $s < 0$ et un unique réel $s' \in \mathbb{R}$ tels que $f \in \mathbb{C}[[x]]_{q,s,s'}$ et $f \notin \mathbb{C}[[x]]_{(q,s,s')}$; $\frac{1}{-s}$ est alors l'une des pentes strictement positives du q -polygone et $\frac{1}{s'}$ l'une des pentes du d -polygone $P_{d,\frac{1}{s}}(L)$.

6.4. Estimations de croissance

Nous allons à présent pouvoir donner des estimations de croissance pour les fonctions entières solutions d'équations à coefficients polynomiaux.

Proposition 6.4.1 : La fonction entière $f(x)$ a une croissance contrôlée par

$$|f(x)| < C \exp \left(\frac{k_1}{2} \ln^2(r) + a \ln(r) + k_1 s' \ln(r) \cdot \ln |\ln(r)| \right) \quad (r = |x|)$$

où $k_1 = \frac{k}{\ln|q|} > 0$, $a \in \mathbb{R}$ et $s' \in \mathbb{R}$, si et seulement si $f(x)$ admet pour développement asymptotique à l'origine une série q -Gevrey d'ordres $s = 1/k$ et s' .

Démonstration : Soit $\phi_n(r)$ la fonction définie sur $]0, +\infty[$ par

$$\begin{cases} \phi_n(r) &= \frac{k_1}{2} \ln^2(r) + a \ln(r) - n \ln(r) + b \ln(r) \cdot \ln |\ln(r)|, & r \neq 1 \\ \phi_n(1) &= 0 \end{cases}$$

où $n \in \mathbb{N}$, $k_1 = \frac{k}{\ln|q|} > 0$, $a \in \mathbb{R}$ et $b = k_1 s' \in \mathbb{R}$.

Nous allons chercher le minimum de $\phi_n(r)$ sur $]0, +\infty[$ puis en donner un équivalent quand n tend vers $+\infty$. La fonction $\phi_n(r)$ est continue sur $]0, +\infty[$. Pour $r \neq 1$

$$\phi'_n(r) = k_1 \frac{\ln(r)}{r} + \frac{a - n}{r} + \frac{b}{r} \ln |\ln(r)| + \frac{b}{r}$$

et on pose pour $f_n(r) = r\phi'_n(r) = k_1 \ln(r) + a - n + b + b \ln |\ln(r)|$. Pour $r \neq 1$

$$f'_n(r) = \frac{k_1}{r} + \frac{b}{r \ln(r)} = \frac{k_1 \ln(r) + b}{r \ln(r)}.$$

Premier cas : $s' = 0$. Le minimum de $\phi_n(r)$ est atteint pour $r = \exp(\frac{n-a}{k_1})$ et $\forall r > 0$, $\phi_n(r) \geq -\frac{(n-a)^2}{2k_1}$.

Deuxième cas : $s' > 0$.

	0	$\exp(-s')$	1	r_n	$+\infty$
$f'_n(r)$	+	0	-	+	+
$f_n(r)$	$-\infty$	\nearrow \searrow $-\infty$	$-\infty$	$-\infty$ \nearrow 0 \nearrow $+\infty$	$+\infty$
$\phi_n(r)$				\searrow \nearrow	

Pour n assez grand $f_n(\exp(-s')) \sim -n$ est strictement négatif et $\phi_n(r)$ admet sur $]0, +\infty[$ un unique minimum atteint pour $r = r_n > 1$. Etudions $f_n(r)$ sur $]1, +\infty[$. On pose $\alpha_n = \exp(\frac{n-a-b}{k_1})n^{-s'}k_1^{s'}$ et $u_n = \frac{r_n}{\alpha_n}$. Pour n assez grand

- $\frac{\alpha_n}{2} \in]1, +\infty[$ et $f_n(\frac{\alpha_n}{2}) \sim -k_1 \ln(2) < 0$
- $2\alpha_n \in]1, +\infty[$ et $f_n(2\alpha_n) \sim k_1 \ln(2) > 0$

donc $\frac{\alpha_n}{2} < r_n < 2\alpha_n$ et par suite $\frac{\ln(u_n)}{n} \rightarrow 0$. La condition $f_n(\alpha_n u_n) = 0$ entraîne $u_n \rightarrow 1$ donc

$$r_n \sim \exp\left(\frac{n-a-b}{k_1}\right) n^{-s'} k_1^{s'} \quad \text{quand } n \text{ tend vers } +\infty.$$

Un calcul simple mais laborieux permet alors d'obtenir l'estimation pour $n \rightarrow +\infty$:

$$\phi_n(r_n) + \frac{n(n+1)}{2k_1} - s' \ln(n!) - \left(\frac{1}{2k_1} + s' + \frac{a}{k_1} - s' \ln(k_1) \right) n \sim -\frac{k_1 s'^2}{2} \ln^2(n).$$

Troisième cas : $s' < 0$

	0	$r_{n,2}$	1	$r_{n,1}$	$\exp(-s')$	r_n	$+\infty$
$f'_n(r)$		+	+	-	-	0	+
$f_n(r)$	$-\infty$	$\nearrow \theta$	$+\infty$	$+\infty$	$\searrow \theta$	$\nearrow \theta$	$+\infty$
$\phi_n(r)$							

En raisonnant de la même manière que précédemment on montre que :

- L'unique racine $r_{n,2}$ de $f_n(r) = 0$ sur $]0, 1[$ tend vers 1 quand $n \rightarrow +\infty$ et $\phi_n(r_{n,2}) \rightarrow 0$.
- Pour n assez grand $f_n(\exp(-s')) < 0$. Soit r_n l'unique racine de $f_n(r) = 0$ sur $]\exp(-s'), +\infty[$. On montre que

$$r_n \sim \exp\left(\frac{n-a-b}{k_1}\right) n^{-s'} k_1^{s'}$$

et

$$\phi_n(r_n) + \frac{n(n+1)}{2k_1} - s' \ln(n!) - \left(\frac{1}{2k_1} + s' + \frac{a}{k_1} - s' \ln(k_1)\right) n \sim -\frac{k_1 s'^2}{2} \ln^2(n).$$

Démontrons à présent la proposition. Soit $f(x)$ une fonction entière telle que

$$|f(x)| < C \exp\left(\frac{k_1}{2} \ln^2(r) + a \ln(r) + k_1 s' \ln(r) \cdot \ln |\ln(r)|\right) \quad (r = |x|)$$

avec $k_1 = \frac{k}{\ln|q|} > 0$, $a \in \mathbb{R}$ et $s' \in \mathbb{R}$. Soit $\sum_{n \geq 0} a_n x^n$ son développement asymptotique à l'origine. Alors

$$|a_n| \leq \sup_{|x|=r} |f(x)| r^{-n} < C \exp\left(\frac{k_1}{2} \ln^2(r) + a \ln(r) + k_1 s' \ln(r) \cdot \ln |\ln(r)|\right) r^{-n}$$

pour tout $r > 0$ donc $|a_n| < C \exp(\phi_n(r_n))$.

- Si $s' = 0$ alors $|a_n| < C \exp(-\frac{(n-a)^2}{2k_1}) = \text{Constante} \times |q|^{-\frac{n(n+1)}{2k}} A^n$;
- Si $s' \neq 0$ alors

$$|a_n| < C |q|^{-\frac{n(n+1)}{2k}} (n!)^{s'} \left(\exp\left(\frac{1}{2k_1} + s' + \frac{a}{k_1} - s' \ln(k_1)\right)\right)^n \exp(\zeta(n))$$

où $\zeta(n) \sim -\frac{k_1 s'^2}{2} \ln^2(n) \rightarrow -\infty$ donc $e^{\zeta(n)}$ est borné et par suite $|a_n| < D |q|^{-\frac{n(n+1)}{2k}} (n!)^{s'} A^n$ (D constante).

Réciproquement supposons qu'une fonction entière $f(x)$ admette comme développement asymptotique $\sum_{n \geq 0} a_n x^n \in \mathcal{C}'[[x]]_{q,s,s'}$ c'est-à-dire qu'il existe C et $A > 0$ tels que

$$|a_n| < C|q|^{-\frac{n(n+1)}{2k}}(n!)^{s'}A^n,$$

avec $k = 1/s\Gamma k_1 = \frac{k}{\ln|q|} > 0$ et $s' \in \mathbb{R}$. Alors

$$|a_n x^n| < C \exp \left(-\frac{n(n+1)}{2k_1} + s' \ln(n!) + n \ln(A) + n \ln(r) \right).$$

Soit a tel que $\ln(A) = \frac{1}{2k_1} + s' + \frac{a}{k_1} - s' \ln(k_1)$. Alors $-\frac{n(n+1)}{2k_1} + s' \ln(n!) + n \ln(A) = \phi_n(r_n) + \zeta(n)$ avec $\zeta(n) \sim \frac{k_1 s'^2}{2} \ln^2(n)$.

Par conséquent

$$|a_n x^n| < C \exp(\phi_n(r_n) + \zeta(n) + n \ln(r))$$

et $\forall \rho > 0$

$$|a_n x^n| < C \exp \left(\frac{k_1}{2} \ln^2(\rho) + a \ln(\rho) + k_1 s' \ln(\rho) \cdot \ln |\ln(\rho)| \zeta(n) - n \ln(\rho) + n \ln(r) \right).$$

On pose $\rho = \lambda r$ ($\lambda > 0$).

- Si $s' = 0$: $\zeta(n) = 0$ et $|a_n x^n| < C \exp(\frac{k_1}{2} \ln^2(\lambda r) + a \ln(\lambda r)) \lambda^{-n}$ donc

$$|f(x)| < C_\lambda \exp \left(\frac{k_1}{2} \ln^2(\rho) + a \ln(\rho) \right) \quad \forall \lambda > 1$$

$$|f(x)| < C_\epsilon \exp \left(\frac{k_1}{2} \ln^2(r) + (a + \epsilon) \ln(r) \right) \quad \forall \epsilon > 0$$

- Si $s' \neq 0$: Soit $\delta_n = \exp(\zeta(n) - n \ln(\lambda))$. On montre que $\frac{\delta_{n+1}}{\delta_n} \sim \frac{1}{\lambda} \Gamma$ c'est-à-dire que la série $\sum_{n \geq 0} \delta_n$ converge pour tout $\lambda > 1$ et $\forall \lambda > 1$,

$$|f(x)| < C_\lambda \exp \left(\frac{k_1}{2} \ln^2(\rho) + k_1 s' \ln(\rho) \ln |\ln(\rho)| + a \ln(\rho) \right);$$

$\forall \epsilon > 0$,

$$|f(x)| < C_\epsilon \exp \left(\frac{k_1}{2} \ln^2(r) + k_1 s' (\epsilon + \ln(r)) \ln |\epsilon + \ln(r)| + (a + k_1 \epsilon) \ln(r) \right)$$

$$< C'_\epsilon \exp \left(\frac{k_1}{2} \ln^2(r) + k_1 s' \ln(r) \cdot \ln |\ln(r)| + (a + k_1 \epsilon + 1) \ln(r) \right).$$

De cette proposition et du théorème 6.3.2 découle le

Théorème 6.4.2 : Soit $0 < |q| < 1$ et L un opérateur à coefficients polynomiaux du type

$$L = \sum_{i=0}^I \sum_{j=0}^J a_{i,j}(x) \left(\frac{d}{dx} \right)^i \sigma_q^j$$

Soient $f \in A(\mathcal{C})$ une fonction entière et $g \in \mathcal{C}'[x]$ tels que $L(f) = g$. Alors :

- Soit $f \in \mathcal{C}'[x]$,

- Soit il existe un unique réel $s' < 0$ tel qu'il existe C et $\alpha > 0$,

$$|f(x)| < C \exp(\alpha r^{-1/s'}) \quad (r = |x|);$$

$\frac{1}{s'}$ est alors l'une des pentes strictement négatives du d -polygone associé à la valeur exceptionnelle $s = 0$,

- Soit il existe un unique réel $s < 0$ et un unique réel $s' \in \mathbb{R}$ tels qu'il existe C et α ,

$$|f(x)| < C \exp \left(\frac{k_1}{2} \ln^2(r) + k_1 s' \ln(r) \ln |\ln(r)| + \alpha \ln(r) \right) \quad (r = |x|, \quad k_1 = \frac{1}{s \ln |q|});$$

$\frac{1}{-s}$ est alors l'une des pentes strictement positives du q -polygone et $\frac{1}{s'}$ l'une des pentes du d -polygone $P_{d, \frac{1}{s}}(L)$.

On retrouve l'estimation de $|f(x)|$ établie dans le cas différentiel dans [9].

Références bibliographiques

- [1] J.-P. Bézivin Γ *Sur les équations fonctionnelles aux q -différences*. Preprint Paris VI (1990).
- [2] R. Douady Γ *Produits tensoriels topologiques et espaces nucléaires*. Séminaire de géométrie analytique. Astérisque 16 (1974).
- [3] P. Grisvard Γ *Opérateurs à indice, Lemme de Compacité*. Séminaire Cartan-Schwartz Γ 16ème année Γ 1963-64 Γ n° 12.
- [4] W. Hahn Γ *Über die Funktional-differentialgleichung $f'(z) = f(qz)$ und verwandte Funktionalgleichungen*. Ann. Univ.Sc. Budapest Γ Eötvös Sect. Math. (1973) Γ p.3-21.
- [5] T. Kato Γ J.-B. McLeod Γ *The functional differential difference equation $y'(x) = ay(\lambda x) - by(x)$* . Bull. Amer. Math. Soc. Γ 77 (1971) Γ p. 891-937.
- [6] H. Komatsu Γ *On the index of differential operators*. J. Fac. Sci. Tokyo IA (1971) Γ p.379-398.
- [7] K. Mahler Γ *On a special functional equation*. J. London Math. Soc. 15 (1940) Γ 115-123. MR 2 Γ 133.
- [8] B. Malgrange Γ *Sur les points singuliers des équations différentielles*. L'Enseignement Mathématique Γ tome 20 Γ n° 1-2 (1974) Γ p. 147-176.
- [9] J.-P. Ramis Γ *Développement Gévrey*. Société Mathématique de France. Asterisque n° 59-60 (1978) Γ p.173-204.
- [10] J.-P. Ramis Γ *Théorèmes d'indices Gévrey pour les équations différentielles ordinaires*. Memoirs of the American Mathematical Society n° 296 Γ volume 48 (1984).
- [11] J.-P. Ramis Γ *About the growth of entire functions solutions of linear algebraic q -difference equations*. Annales de la Faculté des Sciences de Toulouse. Série 6 Γ Vol. II n° 1 Γ 1992.
- [12] K. Yosida Γ *Functional Analysis*. Berlin Γ Springer-Verlag (1971).

Chapitre 2

Théorèmes d'indices pour les équations aux différences à coefficients polynomiaux en q^x

1. Introduction

Le point de départ de ce travail consiste en l'étude de transformations canoniques permettant d'obtenir des estimations de croissance des solutions d'une classe d'opérateurs via les résultats obtenus pour les séries formelles solutions d'équations fonctionnelles q -différences-différentielles *mélangeant* opérateurs différentiels et opérateurs aux q -différences [6].

W. Hahn s'est intéressé au cas des équations q -différences-différentielles à coefficients constants de la forme

$$(1) \quad \sum_{j=0}^r \sum_{k=0}^s \alpha_{j,k} F^{(j)}(tq^k) = 0,$$

où q est un réel ($0 < q < 1$). Il en cherche des solutions analytiques [4]. Par la transformation intégrale

$$F(t) = \int_C \frac{t^x}{\Gamma(x+1)} f(x) dx, \quad t \neq 0$$

où il suppose le chemin d'intégration C et la fonction $f(x)$ tels que $F(t)$ soit analytique et

$$F^{(j)}(tq^k) = \int_C \frac{q^{kx} t^x}{\Gamma(x+1)} f(x+j) dx,$$

il se ramène à l'étude de $f(x)$ solution de l'équation aux différences à coefficients polynomiaux

$$(2) \quad \sum_{j=0}^r \left(\sum_{k=0}^s \alpha_{j,k} (q^x)^k \right) f(x+j) = 0.$$

Connaissant le comportement à l'infini de la fonction $f(x)$ (proposition de Poincaré-Perron) il obtient des solutions analytiques de (1).

On vérifie que les séries formelles $\sum_{n \geq 0} a_n t^n$ solutions d'équations de la forme (1) sont q -Gevrey d'ordre $s \in \mathbb{R}$ et $s' = -1$ [1] c'est-à-dire :

$$\exists C, \exists A > 0, \quad |a_n| < C |q|^{-\frac{sn(n+1)}{2}} \frac{1}{\Gamma(n+1)} A^n.$$

Il est donc naturel de ne pas tenir compte du facteur $\frac{1}{\Gamma(x+1)}$ dans le cas où les coefficients $\alpha_{j,k}$ ne sont plus constants. La transformation intégrale devient alors $F(t) = \int_C t^x f(x) dx$. Enfin on peut aussi bien étudier $f(-x)$ et donc considérer comme transformée

$$F(t) = \int_C t^{-x} f(x) dx \quad \text{ou encore} \quad f(x) = \int_C t^{x-1} F(t) dt.$$

Formellement cette dernière expression donne

$$\tau f(x) = f(x+1) = \int_C t^{x-1} \times t F(t) dt,$$

$$x f(x) = \int_C x t^{x-1} F(t) dt = [t^x F(t)] - \int_C t^x F'(t) dt = \int_C t^{x-1} \left(-t \frac{d}{dt}\right) F(t) dt$$

et

$$q^x f(x) = \int_C t^{x-1} \times \sigma_p F(t) dt.$$

On peut donc naturellement définir comme transformée q -Mellin formelle l'isomorphisme

$$\begin{array}{ccc} \mathcal{C}[\tau, x, q^x] & \longrightarrow & \mathcal{C}[t, -t \frac{d}{dt}, \sigma_p] \\ \tau & \mapsto & t \\ x & \mapsto & -t \frac{d}{dt} \\ q^x & \mapsto & \sigma_p \end{array}$$

En particulier A. Barkatou et A. Duval [1] considèrent la transformation de Mellin formelle

$$\begin{array}{ccc} \mathcal{C}[\tau, x] & \longrightarrow & \mathcal{C}[t, -t \frac{d}{dt}] \\ \tau & \mapsto & t \\ x & \mapsto & -t \frac{d}{dt} \end{array}$$

Ils étudient l'action des opérateurs de $\mathcal{C}[x, \tau]$ sur l'espace des séries de factorielles formelles. La transformation de Mellin formelle leur permet de se ramener aux séries formelles solutions d'équations différentielles et d'appliquer les résultats établis par B. Malgrange [5] et J.-P. Ramis [7]. Ils obtiennent alors des estimations Gevrey pour les séries de factorielles formelles initiales.

Dans cet article nous donnons un q -analogue de leurs résultats. Nous étudions les opérateurs de $\mathcal{C}[q^x, \tau]$ agissant sur l'espace des séries de q -factorielles formelles. La transformée de q -Mellin formelle permettra d'obtenir des estimations q -Gevrey à partir des résultats établis par J.-P. Bézivin [2] et J.-P. Ramis [8] dans le cadre des séries formelles solutions d'équations aux q -différences.

2. Préliminaires

Pour de plus amples détails on se reportera à [3].

2.1. La fonction Gamma q -analogue

Il s'agit ici de rappeler quelques propriétés fondamentales de la fonction q -Gamma. Nous introduirons ensuite quelques notations. Nous supposons dans tout cet article que q est un complexe non nul tel que $|q| \neq 1$. Soit $p = 1/q$. On définit

$$(a; q)_n = \begin{cases} 1 & \text{si } n = 0 \\ (1-a)(1-aq) \dots (1-aq^{n-1}) & \text{si } n = 1, 2, \dots \end{cases}$$

Pour $a \neq 0$ $\Gamma(a; q)_n = (a^{-1}; p)_n (-a)^n q^{\frac{n(n-1)}{2}}$.

Pour $a \in \mathcal{C}$ et $|q| < 1$ on définit $(a; q)_\infty = \prod_{k=0}^{\infty} (1 - aq^k)$.

Pour $a \in \mathcal{C}^*$ et $|q| > 1$ on définit $(a; q)_\infty = (a^{-1}; p)_\infty (-p)^{b-1} p^{\frac{(b-1)(b-2)}{2}}$ où $b = \frac{\log a}{\log q}$.

On vérifie que $(a; q)_n = \frac{(a; q)_\infty}{(aq^n; q)_\infty}$ pour $a \neq p^{n+k}$ pour tout $k \in \mathbb{N}$.

Avec ces notations la fonction Gamma q -analogue est définie comme suit :

- Si $0 < |q| < 1$ $\Gamma_q(x) = \frac{(q; q)_\infty}{(q^x; q)_\infty} (1 - q)^{1-x}$;
- Si $|q| > 1$ $\Gamma_q(x) = \frac{(p; p)_\infty}{(p^x; p)_\infty} (q - 1)^{1-x} q^{\frac{x(x-1)}{2}}$.

Elle est solution de l'équation fonctionnelle

$$\begin{cases} f(x+1) = \frac{1-q^x}{1-q} f(x) \\ f(1) = 1 \end{cases}$$

où nous prenons la valeur principale de q^x . La fonction $\Gamma_q(x)$ est un q -analogue de la fonction Γ dans la mesure où

$$\lim_{q \rightarrow 1} \Gamma_q(x) = \Gamma(x).$$

Elle admet des pôles (simples) en $x = -n \pm \frac{2\pi ik}{\log q}$ pour $k, n \in \mathbb{N}$. La fonction $\frac{1}{\Gamma_q(x)}$ est une fonction entière s'annulant en les points $x = -n \pm \frac{2\pi ik}{\log q}$ pour $k, n \in \mathbb{N}$.

Le coefficient q -binomial est défini par :

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \frac{\Gamma_q(\alpha + 1)}{\Gamma_q(\beta + 1) \Gamma_q(\alpha - \beta + 1)}$$

pour α et $\beta \in \mathcal{C}$. En particulier pour $n, k \in \mathbb{N}$ avec $k = 0, 1, \dots, n$:

$$\begin{bmatrix} n \\ k \end{bmatrix} = \frac{(q; q)_n}{(q; q)_k (q; q)_{n-k}}.$$

Pour $n, k \in \mathbb{N}$ nous avons

$$\begin{bmatrix} n+1 \\ k \end{bmatrix} = \begin{bmatrix} n \\ k \end{bmatrix} q^k + \begin{bmatrix} n \\ k-1 \end{bmatrix}$$

Définissons enfin pour $x \in \mathcal{C}$ et $n \in \mathbb{N}$

$$\langle q^x \rangle_n = \begin{cases} 1 & \text{si } n = 0 \\ \prod_{l=0}^{n-1} \frac{1-q^{x+l}}{1-q} & \text{pour } n = 1, 2, \dots \end{cases}$$

Si $x+n$ n'est pas un pôle pour la fonction $\Gamma_q(x)$ il est évident que $\langle q^x \rangle_n = \frac{\Gamma_q(x+n)}{\Gamma_q(x)}$.

Si $x+n$ est un pôle pour la fonction $\Gamma_q(x)$ on définit $\frac{\Gamma_q(x+n)}{\Gamma_q(x)}$ par $\langle q^x \rangle_n$.

2.2. La q -intégrale

Pour $|q| < 1$ la q -intégrale est définie par

$$\int_0^1 f(t) d_q t := (1 - q) \sum_{n=0}^{\infty} f(q^n) q^n,$$

$$\int_0^a f(t) d_q t := a(1 - q) \sum_{n=0}^{\infty} f(aq^n) q^n$$

et par

$$\int_0^{\infty} f(t) d_q t := (1 - q) \sum_{n=-\infty}^{\infty} f(q^n) q^n.$$

Ainsi la fonction q -beta définie par $B_q(x, y) = \frac{\Gamma_q(x)\Gamma_q(y)}{\Gamma_q(x+y)}$ s'exprime sous la forme d'une q -intégrale :

$$B_q(x, y) = \int_0^1 t^{x-1} \frac{(tq; q)_{\infty}}{(tq^y; q)_{\infty}} d_q t$$

pour $|q| < 1$ $Re(x \log q) < 0$ et $y \neq -n + \frac{2\pi ik}{\log q}$ $\Gamma n \in \mathbb{N}$ $k \in \mathbb{Z}$.

3. Transformée q -Mellin formelle

3.1. Transformée q -Mellin formelle d'un opérateur

On note $\mathcal{C}[q^x, \tau]$ l'algèbre non commutative des polynômes en q^x et τ ; la relation de commutation est : $\tau q^x = q^{x+1} \tau$. On note $\mathcal{C}[t, \sigma_p]$ l'algèbre non commutative des polynômes en t et σ_p ayant pour relation de commutation : $\sigma_p t = p t \sigma_p$.

Définition 3.1.1 : Soit $\Delta_q \in \mathcal{C}[q^x, \tau]$. La transformée q -Mellin formelle de l'opérateur Δ_q est l'opérateur $M_q \in \mathcal{C}[t, \sigma_p]$, image de Δ_q par l'isomorphisme d'algèbre IM_q qui à q^x fait correspondre σ_p et à τ l'opérateur t .

Première remarque : La transformation est canonique : $IM_q(\tau q^x) = IM_q(\tau) IM_q(q^x) = t \sigma_p = q \sigma_p t = q IM_q(q^x) IM_q(\tau) = IM_q(q^{x+1} \tau)$ et $q^{x+1} \tau = \tau q^x$.

Seconde remarque : On peut définir formellement la transformée de q -Mellin formelle par une transformée q -intégrale :

$$\hat{f}_q(x) := \int_0^{\infty} t^{x-1} \hat{F}_q(t) d_q t = (1 - q) \sum_{n=-\infty}^{\infty} q^{nx} \hat{F}_q(q^n).$$

L'action de q^x (resp. de τ) sur $\hat{f}_q(x)$ se traduit formellement par l'action de σ_p (resp. de t) sur

$\hat{F}_q(x) :$

$$\begin{aligned} q^x \hat{f}_q(x) &= (1-q) \sum_{n=-\infty}^{\infty} q^x q^{nx} \hat{F}_q(q^n) = (1-q) \sum_{n=-\infty}^{\infty} q^{x(n+1)} \hat{F}_q(q^n) \\ &= (1-q) \sum_{n=-\infty}^{\infty} q^{nx} (\sigma_p \hat{F}_q)(q^n) = \int_0^{\infty} t^{x-1} \sigma_p \hat{F}_q(t) d_q t, \\ \tau \hat{f}_q(x) &= (1-q) \sum_{n=-\infty}^{\infty} q^n q^{nx} \hat{F}_q(q^n) = (1-q) \sum_{n=-\infty}^{\infty} q^n q^{n(x-1)} q^n \hat{F}_q(q^n) \\ &= \int_0^{\infty} t^{x-1} t \hat{F}_q(t) d_q t. \end{aligned}$$

Soit $\Delta_q \in \mathcal{C}[q^x, \tau]$; Δ_q peut s'écrire sous l'une des deux formes

$$\Delta_q = \sum_{k=0}^r \sum_{j=0}^s \alpha_{k,j} \tau^k (q^x)^j$$

$$\Delta_q = \sum_{k=0}^r \sum_{j=0}^s \beta_{k,j} \tau^k < q^x >_j$$

Proposition 3.1.2 : La transformée q -Mellin formelle vérifie :

$$IM_q \left(\sum_{k=0}^r \sum_{j=0}^s \alpha_{k,j} \tau^k (q^x)^j \right) = \sum_{k=0}^r \sum_{j=0}^s \alpha_{k,j} t^k \sigma_p^j$$

$$IM_q \left(\sum_{k=0}^r \sum_{j=0}^s \beta_{k,j} \tau^k < q^x >_j \right) = \sum_{k=0}^r \sum_{j=0}^s \beta_{k,j} t^k < \sigma_p >_j$$

où $< \sigma_p >_j$ est l'identité si $j = 0$; pour $j \in \mathbb{N}^*$, $< \sigma_p >_j = \frac{1}{(1-q)^j} \prod_{l=0}^{j-1} (1 - q^l \sigma_p)$.

3.2. Séries de q -factorielles

Définition 3.2.1 : Les séries formelles de q -factorielles généralisées sont données par

$$\hat{f}_q(x) = \sum_{n \geq 0} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n+\rho+1)} = \frac{\Gamma_q(x)}{\Gamma_q(x+\rho)} \sum_{n \geq 0} \frac{a_n}{< q^{x+\rho} >_{n+1}}$$

où $\rho \in \mathcal{C}$. Le cas $\rho = 0$ correspond aux séries formelles de q -factorielles.

Propriété 3.2.2 : Les séries formelles de q -factorielles généralisées sont stables par l'action de $\mathcal{C}[q^x, \tau]$.

Pour établir cette propriété nous allons utiliser le lemme suivant :

Lemme 3.2.3 : Pour tout $l \in \mathbb{N}$ et pour tout $\beta \in \mathcal{C}$:

$$\frac{\Gamma_q(x+l)}{\Gamma_q(x)} = \sum_{i=0}^l \begin{bmatrix} l \\ i \end{bmatrix} q^{(\beta-i)(l-i)} \frac{\Gamma_q(\beta+1)}{\Gamma_q(\beta+1-i)} \frac{\Gamma_q(x+l-\beta)}{\Gamma_q(x-\beta+i)}$$

ou encore

$$< q^x >_l = \sum_{i=0}^l \begin{bmatrix} l \\ i \end{bmatrix} q^{(\beta-i)(l-i)} < q^{\beta+1-i} >_i < q^{x-\beta+i} >_{l-i}.$$

Ce lemme est un q -analogue de la relation

$$\frac{\Gamma(x+l)}{\Gamma(x)} = \sum_{i=0}^l \binom{l}{i} \beta(\beta-1) \dots (\beta-i+1) \frac{\Gamma(x+l-\beta)}{\Gamma(x-\beta+i)}.$$

Démonstration : Elle s'établit par récurrence sur $l \in \mathbb{N}$. Pour $l=0$ le lemme est trivialement vérifié. Supposons que nous l'ayons établi pour $l \in \mathbb{N}$. Soient x et $\beta \in \mathbb{C}$. Posons

$$\Gamma_q^{l+1} = \sum_{i=0}^{l+1} \left[\begin{matrix} l+1 \\ i \end{matrix} \right] q^{(\beta-i)(l+1-i)} \frac{\Gamma_q(\beta+1)}{\Gamma_q(\beta+1-i)} \frac{\Gamma_q(x+l+1-\beta)}{\Gamma_q(x-\beta+i)}.$$

Nous allons établir que $\Gamma_q^{l+1} = \frac{\Gamma_q(x+l+1)}{\Gamma_q(x)}$. Comme

$$\left[\begin{matrix} l+1 \\ i \end{matrix} \right] = \left[\begin{matrix} l \\ i \end{matrix} \right] q^i + \left[\begin{matrix} l \\ i-1 \end{matrix} \right],$$

$$\begin{aligned} \Gamma_q^{l+1} &= \sum_{i=0}^{l+1} \left[\begin{matrix} l \\ i \end{matrix} \right] q^{(\beta-i)(l-i)} q^\beta \frac{\Gamma_q(\beta+1)}{\Gamma_q(\beta+1-i)} \frac{1-q^{x+l-\beta}}{1-q} \frac{\Gamma_q(x+l-\beta)}{\Gamma_q(x-\beta+i)} \\ &\quad + \sum_{i=0}^{l+1} \left[\begin{matrix} l \\ i-1 \end{matrix} \right] q^{(\beta-i)(l+1-i)} \frac{\Gamma_q(\beta+1)}{\Gamma_q(\beta+1-i)} \frac{\Gamma_q(x+l+1-\beta)}{\Gamma_q(x-\beta+i)}. \end{aligned}$$

En utilisant le fait que $\left[\begin{matrix} l \\ l+1 \end{matrix} \right] = 0$ et l'hypothèse de récurrence la première somme est égale à $q^\beta \frac{1-q^{x+l-\beta}}{1-q} \frac{\Gamma_q(x+l)}{\Gamma_q(x)}$. Comme $\left[\begin{matrix} l \\ -1 \end{matrix} \right] = 0$ et en décalant l'indice de sommation le deuxième terme est égal à :

$$\sum_{i=0}^l \left[\begin{matrix} l \\ i \end{matrix} \right] q^{(\beta-i-1)(l-i)} \frac{\Gamma_q(\beta+1)}{\Gamma_q(\beta-i)} \frac{\Gamma_q(x+l+1-\beta)}{\Gamma_q(x-\beta+i+1)}$$

soit en appliquant l'hypothèse de récurrence (avec $\beta-1$) $\Gamma_q \frac{1-q^\beta}{1-q} \frac{\Gamma_q(x+l)}{\Gamma_q(x)}$, d'où le résultat.

Démonstration de la propriété 3.2.2 : Il suffit de montrer pour tout $\rho \in \mathbb{C}$ pour tous $k, j \in \mathbb{N}$ que $\tau^k < q^x >_j \frac{\Gamma_q(x)}{\Gamma_q(x+\rho)}$ est encore une série formelle de q -factorielles généralisées. Or

$$\begin{aligned} \tau^k < q^x >_j \frac{\Gamma_q(x)}{\Gamma_q(x+\rho)} &= \frac{\Gamma_q(x+j+k)}{\Gamma_q(x+k+\rho)} \\ &= \sum_{i=0}^{j+k} \left[\begin{matrix} j+k \\ i \end{matrix} \right] q^{(\beta-i)(j+k-i)} \frac{\Gamma_q(\beta+1)}{\Gamma_q(\beta+1-i)} \frac{\Gamma_q(x) \Gamma_q(x+j+k-\beta)}{\Gamma_q(x-\beta+i) \Gamma_q(x+k+\rho)} \end{aligned}$$

pour tout $\beta \in \mathbb{C}$. En particulier pour $\beta = j - \rho$:

$$\tau^k < q^x >_j \frac{\Gamma_q(x)}{\Gamma_q(x+\rho)} = \sum_{i=0}^{j+k} \left[\begin{matrix} j+k \\ i \end{matrix} \right] q^{(j-\rho-i)(j+k-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)} \times \frac{\Gamma_q(x)}{\Gamma_q(x-j+\rho+i)}.$$

Définition 3.2.4 : Avec les notations de la définition 3.2.1, la transformée q -Mellin formelle d'une série formelle de q -factorielles généralisées est la série formelle

$$M_q(\hat{f}_q)(t) = \sum_{n \geq 0} \frac{a_n}{\Gamma_q(n+1+\rho)} \frac{(tq; q)_\infty}{(tq^{n+1+\rho}; q)_\infty} = \frac{(tq; q)_\infty}{(tq^{\rho+1}; q)_\infty} \sum_{n \geq 0} \frac{a_n}{\Gamma_q(n+1+\rho)} (tq^{\rho+1}; q)_n$$

Remarque : La fonction q -beta $\Gamma B_q(x, y) = \frac{\Gamma_q(x)\Gamma_q(y)}{\Gamma_q(x+y)} \Gamma$ est égale à

$$B_q(x, y) = \int_0^1 t^{x-1} \frac{(tq; q)_\infty}{(tq^y; q)_\infty} d_q t$$

pour $|q| < 1$ $\Gamma Re(x \log q) > 0$ et $y \neq -n + \frac{2\pi ik}{\log q} \Gamma n \in \mathbb{N}$ et $k \in \mathbb{Z}$ [8] d'où

$$\frac{\Gamma_q(x)}{\Gamma_q(x+\rho)} = \int_0^1 t^{x-1} \times \frac{1}{\Gamma_q(\rho)} \frac{(tq; q)_\infty}{(tq^\rho; q)_\infty} d_q t.$$

Propriété 3.2.5 : Les opérateurs polynomiaux de $\mathcal{C}[t, \sigma_p]$ agissent sur les séries formelles en $\frac{(tq; q)_\infty}{(tq^{n+1+\rho}; q)_\infty} = u_{\rho+n+1}(t)$.

Démonstration : Il suffit de remarquer que pour tout $\rho \in \mathbb{C}$:

- $\sigma_p u_\rho(t) = \frac{(tq; q)_\infty}{(tq^{\rho-1}; q)_\infty} = \frac{1-t}{1-tq^{\rho-1}} \frac{(tq; q)_\infty}{(tq^\rho; q)_\infty} = q^{1-\rho} u_\rho(t) + (1 - q^{1-\rho}) u_{\rho-1}(t)$
- $tu_\rho(t) = q^{-\rho} (tq^\rho - 1 + 1) \frac{(tq; q)_\infty}{(tq^\rho; q)_\infty} = q^{-\rho} u_\rho - q^{-\rho} (1 - tq^\rho) \frac{(tq; q)_\infty}{(tq^\rho; q)_\infty} = q^{-\rho} u_\rho - q^{-\rho} u_{\rho+1}$

Proposition 3.2.6 : Si \hat{f}_q est une série formelle de q -factorielles généralisées alors pour tout opérateur $\Delta_q \in \mathcal{C}[q^x, \tau]$, $IM_q(\Delta_q(\hat{f}_q)) = IM_q(\Delta_q) IM_q(\hat{f}_q)$.

Démonstration : Par linéarité il suffit de vérifier que

$$IM_q(\tau^k < q^x >_j \left(\frac{\Gamma_q(x)}{\Gamma_q(x+\rho)} \right)) = t^k < \sigma_p >_j \left(\frac{u_\rho(t)}{\Gamma_q(\rho)} \right)$$

pour tout $\beta \in \mathbb{C}$. On a montré précédemment que :

$$\tau^k < q^x >_j \frac{\Gamma_q(x)}{\Gamma_q(x+\rho)} = \sum_{i=0}^{j+k} \left[\begin{matrix} j+k \\ i \end{matrix} \right] q^{(j-\rho-i)(j+k-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)} \times \frac{\Gamma_q(x)}{\Gamma_q(x-j+\rho+i)}$$

dont la transformée q -Mellin formelle est

$$\phi_{k,j}(t) = \sum_{i=0}^{j+k} \left[\begin{matrix} j+k \\ i \end{matrix} \right] q^{(j-\rho-i)(j+k-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)} \frac{1}{\Gamma_q(-j+\rho+i)} u_{-j+\rho+i}(t).$$

Pour montrer que $\phi_{k,j}(t) = t^k < \sigma_p >_j \left(\frac{u_\rho}{\Gamma_q(\rho)} \right) \Gamma$ nous allons procéder en deux étapes.

Première étape : Nous allons établir que $\phi_{0,j}(t) = < \sigma_p >_j \frac{1}{\Gamma_q(\rho)} u_\rho(t)$ par récurrence sur

$j \in \mathbb{N}$. Pour $j = 0$ $\Gamma \phi_{0,0}(t) = \frac{1}{\Gamma_q(\rho)} u_\rho(t)$.

Si $\phi_{0,j}(t) = < \sigma_p >_j \frac{1}{\Gamma_q(\rho)} u_\rho(t) : < \sigma_p >_{j+1} \frac{u_\rho(t)}{\Gamma_q(\rho)} = \frac{1-q^j \sigma_p}{1-q} \phi_{0,j}(t)$. Or

$$\phi_{0,j}(t) = \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} q^{(j-\rho-i)(j-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)\Gamma_q(-j+\rho+i)} u_{\rho-j+i}$$

et $\sigma_p u_\rho(t) = q^{1-\rho} u_\rho + (1-q^{1-\rho}) u_{\rho-1}$ pour tout ρ donc $\sigma_p \phi_{0,j}(t)$ est égal à

$$\begin{aligned} & \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} q^{(j-\rho-i)(j-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)\Gamma_q(-j+\rho+i)} q^{1+j-\rho-i} u_{\rho-j+i} \\ & + \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} q^{(j-\rho-i)(j-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)\Gamma_q(-j+\rho+i)} (1-q^{1+j-\rho-i}) u_{\rho-j+i-1} \end{aligned}$$

et $< \sigma_p >_{j+1} \frac{u_\rho}{\Gamma_q(\rho)}$ est égal à

$$\begin{aligned} & \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} q^{(j-\rho-i)(j-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)\Gamma_q(-j+\rho+i)} \frac{1}{1-q} \frac{1-q^{1+2j-\rho-i}}{1-q} u_{\rho-j+i} \\ & - \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} \frac{q^j}{1-q} q^{(j-\rho-i)(j-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)\Gamma_q(-j+\rho+i)} \frac{1-q^{1+j-\rho-i}}{1-q} u_{\rho-(j+1)+i}. \end{aligned}$$

En décalant dans le premier terme l'indice de sommation ($i' = i+1$) et en utilisant le fait que

$$\begin{bmatrix} j \\ -1 \end{bmatrix} = 0 \text{ et } \begin{bmatrix} j \\ j+1 \end{bmatrix} = 0 \Gamma < \sigma_p >_{j+1} \frac{u_\rho}{\Gamma_q(\rho)} \text{ est égal à}$$

$$\sum_{i=0}^{j+1} c_{i,j+1} q^{(j+1-\rho-i)(j-i)} \frac{\Gamma_q(j+1-\rho+1)}{\Gamma_q(j+1-\rho+1-i)\Gamma_q(-(j+1)+\rho+i)} \frac{1}{1-q} u_{\rho-(j+1)+i}$$

avec

$$\begin{aligned} c_{i,j+1} &= \begin{bmatrix} j \\ i-1 \end{bmatrix} \frac{1-q^{2j-\rho+2-i}}{1-q^{j-\rho+1}} - \begin{bmatrix} j \\ i \end{bmatrix} q^{2i-j-1+\rho} \frac{1-q^{j-\rho+1-i}}{1-q^{j-\rho+1}} \frac{1-q^{j-\rho+1-i}}{1-q^{-j+\rho+i-1}} \\ &= \begin{bmatrix} j \\ i-1 \end{bmatrix} \frac{1-q^{2j-\rho+2-i}}{1-q^{j-\rho+1}} - \begin{bmatrix} j \\ i \end{bmatrix} q^i \frac{1-q^{j-\rho+1-i}}{1-q^{j-\rho+1}} = \begin{bmatrix} j+1 \\ i \end{bmatrix}. \end{aligned}$$

Deuxième étape : On montre que $\phi_{k,j}(t) = t^k < \sigma_p >_j \frac{u_\rho}{\Gamma_q(\rho)}$ par récurrence sur $k \in \mathbb{N}$. Pour $k = 0$ c'est évident. Si $\phi_{k,j}(t) = t^k \phi_{0,j}(t)$:

$$t^{k+1} \phi_{0,j}(t) = t \phi_{k,j} = \sum_{i=0}^{j+k} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j-\rho-i)(j+k-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)\Gamma_q(-j+\rho+i)} t u_{-j+\rho+i}(t).$$

Or $t u_\rho = q^{-\rho} u_\rho - q^{-\rho} u_{\rho+1}$ donc $t^{k+1} \phi_{0,j}(t)$ est égal à

$$\sum_{i=0}^{j+k} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j-\rho-i)(j+k-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)\Gamma_q(-j+\rho+i)} q^{j-\rho-i} u_{-j+\rho+i}$$

$$- \sum_{i=0}^{j+k} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j-\rho-i)(j+k-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)\Gamma_q(-j+\rho+i)} q^{j-\rho-i} u_{-j+\rho+i+1}.$$

En décalant dans le deuxième terme l'indice ($i' = i + 1$) et en utilisant $\begin{bmatrix} j+k \\ -1 \end{bmatrix} = 0$ et

$$\begin{bmatrix} j+k \\ j+k+1 \end{bmatrix} = 0 \Gamma t^{k+1} \phi_{0,j}(t) \text{ est égal à}$$

$$\sum_{i=0}^{j+k+1} \left(\begin{bmatrix} j+k \\ i \end{bmatrix} - \begin{bmatrix} j+k \\ i-1 \end{bmatrix} q^{j+k-i+1} \right) q^{(j-\rho-i)(j+k+1-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)\Gamma_q(-j+\rho+i)} u_{-j+\rho+i}(t)$$

donc $t^{k+1} \phi_{0,j}(t) = \phi_{k+1,j}(t)$.

Remarque : Si $IM_q(\frac{\Gamma_q(x)}{\Gamma_q(x+\rho)}) = \phi_\rho(t)$ alors la commutativité se traduit par

$$t^k < \sigma_p >_j \phi_\rho(t) = \sum_{i=0}^{j+k} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j-\rho-i)(j+k-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)} \phi_{\rho-j+i}(t).$$

Pour $j = 0$ et $k = 1$: $t\phi_\rho = q^{-\rho}\phi_\rho + \frac{1-q^{-\rho}}{1-q}\phi_{\rho+1}$.

Pour $j = 1$ et $k = 0$: $\sigma_p\phi_\rho = -(1-q)q^{1-\rho}\phi_{\rho-1} + q^{1-\rho}\phi_\rho$.

En posant $\phi_\rho = \frac{u_\rho}{\Gamma_q(\rho)}\pi_\rho(t)\Gamma\pi_\rho(t)$ doit satisfaire $\pi_\rho(t) = \pi_{\rho+1}(t)$ et $\pi_\rho(pt) = \pi_\rho(t)$. Par analogie Γ A. Barkatou et A. Duval établissent que $\pi_\rho(t)$ est une fonction 1-périodique de la variable $\rho\Gamma$ indépendante de t .

4. Théorèmes d'indices dans $\tilde{\mathcal{C}}_q[[t]]$

Nous serons amenés à déterminer les indices pour des opérateurs $M_q \in \mathcal{C}[t, \sigma_p]$ agissant sur $\tilde{\mathcal{C}}_q[[t]]$ Γ espace des séries formelles de la forme $\hat{F}_q(t) = \sum_{n \geq 0} a_n(tq; q)_n$ avec $(tq; q)_0 = 1$ et $(tq; q)_n = (1-tq)(1-tq^2) \dots (1-tq^n)$ si $n = 1, 2, \dots$

4.1. Indice dans $\tilde{\mathcal{C}}_q[[t]]$

Nous allons calculer à la main cet indice. Soit $M_q = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} t^k < \sigma_p >_j \in \mathcal{C}[t, \sigma_p]$. Soient $\hat{F}_q(t) = \sum_{n \geq 0} a_n(tq; q)_n$ et $\hat{\theta}_q(t) = \sum_{n \geq 0} b_n(tq; q)_n \in \tilde{\mathcal{C}}_q[[t]]$ deux séries telles que $M_q \hat{F}_q = \hat{\theta}_q$. Nous avons montré que pour tout $\rho \in \mathbb{C}\Gamma$

$$t^k < \sigma_p >_j u_\rho = \sum_{i=0}^{j+k} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j-i-\rho)(j+k-i)} \frac{\Gamma_q(j-\rho+1)}{\Gamma_q(j-\rho+1-i)} \frac{\Gamma_q(\rho)}{\Gamma_q(\rho+i-j)} u_{\rho-j+i}.$$

En posant

$$\beta_{k,j,i,n} = \begin{bmatrix} j+k \\ i \end{bmatrix} q^{-(n+1)(j+k-i)} \frac{\Gamma_q(i-n)}{\Gamma_q(-n)} \frac{\Gamma_q(n+j-i+1)}{\Gamma_q(n+1)},$$

$M_q \hat{F}_q$ est égal à

$$\sum_{k=0}^K \sum_{j=0}^J \sum_{i=0}^{j+k} \sum_{n \geq 0} \alpha_{k,j} a_n \beta_{k,j,i,n-j+i} u_{-j+n+1+i} = \sum_{k=0}^K \sum_{j=0}^J \sum_{i=0}^{j+k} \sum_{n \geq -j+i} \alpha_{k,j} a_{n+j-i} \beta_{k,j,i,n} u_{n+1}.$$

Or $\frac{\Gamma_q(i-n)}{\Gamma_q(-n)} = \langle q^{-n} \rangle_i$ s'annule pour $i \neq 0$ et $n = 0 \dots i-1$ et $\frac{\Gamma_q(n+j-i+1)}{\Gamma_q(n+1)}$ s'annule pour $j-i \neq 0$ et $n \leq -1$ donc

$$M_q \hat{F}_q = \sum_{k=0}^K \sum_{j=0}^J \sum_{i=0}^{j+k} \sum_{n \geq i} \alpha_{k,j} a_{n+j-i} \beta_{k,j,i,n} (tq; q)_n.$$

L'égalité $M_q \hat{F}_q = \hat{\theta}_q$ se traduit par :

$$\forall n \geq 0, \quad \sum_{k=0}^K \sum_{j=0}^J \sum_{i=0}^{\min(j+k,n)} \alpha_{k,j} a_{n+j-i} \beta_{k,j,i,n} = b_n.$$

Posons $m(M_q) = \sup\{j = 0 \dots J \mid \exists k = 0 \dots K, \alpha_{k,j} \neq 0\}$. Soit $\Gamma = \{k = 0 \dots K \mid \alpha_{k,m(M_q)} \neq 0\}$. Si $k \notin \Gamma$ alors la condition $\alpha_{k,j} \neq 0$ entraîne $j < m(M_q)$. Par conséquent Γ pour n assez grand

$$U(n) a_{n+m(M_q)} = S_n(a_{n+m(M_q)-1}, \dots, a_{n-K}, b_n)$$

où $S_n(a_{n+m(M_q)-1}, \dots, a_{n-K}, b_n)$ est une expression algébrique en les $a_{n+m(M_q)-1}, \dots, a_{n-K}, b_n$ et avec

$$U(n) = \left(\sum_{k \in \Gamma} \alpha_{k,m(M_q)} q^{-(n+1)(m(M_q)+k)} \right) \frac{\Gamma_q(n+m(M_q)+1)}{\Gamma_q(n+1)}.$$

Soit $k_0 = \sup\{k \mid \alpha_{k,m(M_q)} \neq 0\} = \sup \Gamma$. Le coefficient $U(n)$ s'écrit

$$\alpha_{k_0,m(M_q)} q^{-(n+1)(m(M_q)+k_0)} \frac{\Gamma_q(n+m(M_q)+1)}{\Gamma_q(n+1)} \left(1 + \sum_{k \in \Gamma, k < k_0} \frac{\alpha_{k,m(M_q)}}{\alpha_{k_0,m(M_q)}} q^{-(n+1)(k-k_0)} \right).$$

Si $|q| < 1$ alors $\sum_{k \in \Gamma, k < k_0} \frac{\alpha_{k,m(M_q)}}{\alpha_{k_0,m(M_q)}} q^{-(n+1)(k-k_0)}$ tend vers 0 quand n tend vers $+\infty$ et $U(n)$ est non nul pour n assez grand soit $n \geq N$ pour N convenable.

Si $|q| > 1$ $U(n)$ s'écrit

$$\alpha_{k_1,m(M_q)} q^{-(n+1)(m(M_q)+k_1)} \frac{\Gamma_q(n+m(M_q)+1)}{\Gamma_q(n+1)} \left(1 + \sum_{k \in \Gamma, k > k_1} \frac{\alpha_{k,m(M_q)}}{\alpha_{k_1,m(M_q)}} q^{-(n+1)(k-k_1)} \right)$$

(avec $k_1 = \inf \Gamma$) donc est non nul pour n assez grand.

Pour tout $\hat{\theta}_q \in \tilde{\mathcal{C}}_q[[t]]$ tel que $\inf\{n \mid b_n \neq 0\} \geq N$ il existe donc une unique série $\hat{F}_q \in \tilde{\mathcal{C}}_q[[t]]$ telle que $M_q \hat{F}_q = \hat{\theta}_q$ et $\inf\{n \mid a_n \neq 0\} \geq N + m(M_q)$. En d'autres termes avec des notations

évidentes l'application $M_q : \tilde{\mathcal{C}}_{q,N+m(M_q)} \longrightarrow \tilde{\mathcal{C}}_{q,N}$ est un isomorphisme. Le diagramme

$$\begin{array}{ccccccc}
& & 0 & & 0 & & \\
& & \downarrow & & \downarrow & & \\
0 & \longrightarrow & \tilde{\mathcal{C}}_{q,N+m(M_q)} & \longrightarrow & \tilde{\mathcal{C}}_{q,N} & \longrightarrow & 0 \\
& & \downarrow & & \downarrow & & \\
0 & \longrightarrow & \tilde{\mathcal{C}}_q[[t]] & \longrightarrow & \tilde{\mathcal{C}}_q[[t]] & \longrightarrow & 0 \\
& & \downarrow & & \downarrow & & \\
0 & \longrightarrow & \tilde{\mathcal{C}}_q[[t]]/\tilde{\mathcal{C}}_{q,N+m(M_q)} & \longrightarrow & \tilde{\mathcal{C}}_q[[t]]/\tilde{\mathcal{C}}_{q,N} & \longrightarrow & 0 \\
& & \downarrow & & \downarrow & & \\
& & 0 & & 0 & &
\end{array}$$

est commutatif donc M_q est à indice dans $\tilde{\mathcal{C}}_q[[t]]$ l'égal à

$$\chi_\infty = m(M_q) = \sup\{j = 0 \dots J \mid \exists k = 0 \dots K, \alpha_{k,j} \neq 0\}.$$

4.2. Indices dans les espaces q -Gevrey

Définitions 4.2.1 : Une série $\hat{F}_q(t) = \sum_{n \geq 0} a_n(tq; q)_n$ est q -Gevrey d'ordre $s \in \mathbb{R}$ s'il existe C et $A > 0$ tels que

$$\forall n, \quad |a_n| < C q_0^{-\frac{sn(n+1)}{2}} A^n \quad (q_0 = \sup(|q|, |q|^{-1})).$$

L'ensemble des séries q -Gevrey d'ordre s est noté $\tilde{\mathcal{C}}_{q,s}[[t]]$.

La série est q -Gevrey-Beurling d'ordre s si pour tout $A > 0$ il existe $C_A > 0$ tel que

$$\forall n, \quad |a_n| < C_A q_0^{-\frac{sn(n+1)}{2}} A^n.$$

L'ensemble correspondant est noté $\tilde{\mathcal{C}}_{q,(s)}[[t]]$.

Soit $\mathcal{C}[[v]]$ l'espace des séries formelles entières en la variable v .

Considérons l'isomorphisme $\phi : \tilde{\mathcal{C}}_q[[t]] \rightarrow \mathcal{C}[[v]]$ qui à $(tq; q)_n$ associe v^n . Il induit des isomorphismes $\phi : \tilde{\mathcal{C}}_{q,s}[[t]] \rightarrow \mathcal{C}[[v]]_{p,s}$ et $\phi : \tilde{\mathcal{C}}_{q,(s)}[[t]] \rightarrow \mathcal{C}[[v]]_{p,(s)}$ où $\mathcal{C}[[v]]_{p,s}$ (resp. $\mathcal{C}[[v]]_{p,(s)}$) est l'espace p -Gevrey d'ordre s (resp. p -Gevrey-Beurling d'ordre s) [7].

On note $\sigma_{p,v}$ l'opérateur aux p -différences agissant sur les séries en v ($\sigma_{p,v}(g(v)) = g(pv)$).

Proposition 4.2.2 : Soit $M_q \in \mathcal{C}[t, \sigma_p]$. L'opérateur $M_v = \phi \circ M_q \circ \phi^{-1}$ est polynomial en v , v^{-1} et $\sigma_{p,v}$. De plus, $M_v : \mathcal{C}[[v]]_{p,s} \rightarrow \mathcal{C}[[v]]_{p,s}$ est de Fredholm, donc $M_q : \tilde{\mathcal{C}}_{q,s}[[t]] \rightarrow \tilde{\mathcal{C}}_{q,s}[[t]]$ est également de Fredholm, et les indices sont égaux.

Démonstration : En remarquant que $t = \phi^{-1}(p(1-v)\sigma_{p,v})\phi$ et d'autre part que $\sigma_p = \phi^{-1}(\sigma_{p,v} + \frac{1}{v} - \frac{1}{v}\sigma_{p,v})\phi$ nous établissons que $M_q = \sum \sum \alpha_{k,j} t^k \sigma_p^j = \phi^{-1} M_v \phi$ avec $M_v = \sum \sum \alpha_{k,j} p^k (\sigma_{p,v} - v\sigma_{p,v})^k (\sigma_{p,v} + \frac{1}{v} - \frac{1}{v}\sigma_{p,v})^j \in \mathcal{C}[v, v^{-1}][\sigma_{p,v}]$.

Soit J le degré en v^{-1} de l'opérateur M_v . L'opérateur $M = v^J M_v$ est un opérateur aux p -différences à coefficients polynomiaux. En utilisant les théorèmes d'indices établis pour des opérateurs aux q -différences à coefficients polynomiaux agissant dans les espaces p -Gevrey d'ordre $s \in \mathbb{R}$ [8] On montre aisément que $M : \mathcal{C}[[v]]_{p,s} \rightarrow v^J \mathcal{C}[[v]]_{p,s}$ est un opérateur de Fredholm d'indice χ_s égal à l'indice de $M : \mathcal{C}[[v]]_{p,s} \rightarrow \mathcal{C}[[v]]_{p,s}$ plus J . Par conséquent $M_v : \mathcal{C}[[v]]_{p,s} \rightarrow \mathcal{C}[[v]]_{p,s}$ et par suite $M_q : \tilde{\mathcal{C}}_{q,s}[[t]] \rightarrow \mathcal{C}_{q,s}[[t]]$ sont de Fredholm d'indice égal à χ_s . Cet indice est l'opposé de l'ordonnée de l'unique point (ou de l'origine du segment) intersection du polygone de Newton de M_v au sens p -différences et de la droite d'appui de pente $1/s$.

Nous allons définir un polygone de Newton pour M_q . Cela nous permettra de donner explicitement les indices en fonction de M_q sans avoir à passer par le polygone de Newton au sens p -différences de l'opérateur M_v et qui n'est qu'un intermédiaire.

Lemme 4.2.3 : Si $M_q = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} t^k \sigma_p^j$ alors

$$M_v = \sum_{i=0}^{J+K} \sum_{j=\max(0, i-K)}^J \beta_{i,j} v^{-j} \prod_{l=0}^{i-1} (1 - p^l v) \sigma_{p,v}^i$$

avec

$$\beta_{i,j} = \sum_{k=\max(0, i-j)}^{\min(K, i)} \alpha_{k,j} q^{k(j-1)} \begin{bmatrix} j \\ i-k \end{bmatrix} (-1)^{i-k} q^{\frac{(i-k)(i-k-1)}{2}}.$$

Démonstration : Nous savons que $M_v = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} p^k (\sigma_{p,v} - v \sigma_{p,v})^k (\sigma_{p,v} + \frac{1}{v} - \frac{1}{v} \sigma_{p,v})^j$. On montre aisément que $(\sigma_{p,v} - v \sigma_{p,v})^k = \prod_{l=0}^{k-1} (1 - p^l v) \sigma_{p,v}^k$ (si $k = 0$ $\prod_{l=0}^{k-1} (1 - p^l v)$ est égal à 1 par convention).

Montrons par récurrence sur j que $(\sigma_{p,v} + \frac{1}{v} - \frac{1}{v} \sigma_{p,v})^j = \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} v^{i-j} (v^{-1}; q)_i \sigma_{p,v}^i$. Si cette égalité est établie pour l'entier j alors :

$$\begin{aligned} (\sigma_{p,v} + \frac{1}{v} - \frac{1}{v} \sigma_{p,v})^{j+1} &= \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} v^{i-j} (v^{-1}; q)_i \sigma_{p,v}^i (\sigma_{p,v} + \frac{1}{v} - \frac{1}{v} \sigma_{p,v}) \\ &= \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} v^{i-j} (v^{-1}; q)_i (\sigma_{p,v}^{i+1} - \frac{q^i}{v} \sigma_{p,v}^{i+1}) + \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} v^{i-j} (v^{-1}; q)_i \frac{q^i}{v} \sigma_{p,v}^i \\ &= \sum_{i=1}^{j+1} \begin{bmatrix} j \\ i-1 \end{bmatrix} v^{i-j-1} (v^{-1}; q)_i \sigma_{p,v}^i + \sum_{i=0}^j \begin{bmatrix} j \\ i \end{bmatrix} v^{i-j-1} (v^{-1}; q)_i q^i \sigma_{p,v}^i \\ &= \sum_{i=0}^{j+1} \begin{bmatrix} j+1 \\ i \end{bmatrix} v^{i-(j+1)} (v^{-1}; q)_i \sigma_{p,v}^i. \end{aligned}$$

Le lemme se déduit de ces deux expressions.

Le polygone de Newton (au sens p -différences) de M_v fait intervenir les points $(i, i-j)$ et $(i, -j)$ pour tout couple (i, j) tel que $i = 0, \dots, J+K$, $j = \max(0, i-K), \dots, J$ et $\beta_{i,j} \neq 0$.

Soit j fixé tel que $\Gamma_j = \{k = 0, \dots, K \mid \alpha_{k,j} \neq 0\}$ est non vide. Soient $k_j^- = \min \Gamma_j$ et $k_j^+ = \max \Gamma_j$. Alors $\beta_{k_j^-,j}$ et $\beta_{k_j^+,j,j}$ sont non nuls : les points $(k_j^-, k_j^- - j)\Gamma(k_j^-, -j)\Gamma(k_j^+ + j, -j)$ et $(k_j^+ + j, k_j^+)$ interviennent dans la construction du polygone.

Soit $i = 0, \dots, J + K$ tel qu'il existe j avec $\beta_{i,j} \neq 0$ et $j \geq \max(0, i - K)$. Alors pour un tel $j \Gamma \Gamma_j$ est non vide. On détermine k_j^- et k_j^+ . On montre que : $k_j^- \leq i \leq k_j^+ + j$; les points $(i, i - j)$ et $(i, -j)$ se situent dans l'enveloppe convexe des points $(k_j^-, k_j^- - j)\Gamma(k_j^-, -j)\Gamma(k_j^+ + j, -j)$ et $(k_j^+ + j, k_j^+)$. On en déduit un polygone de Newton pour M_q :

Définition 4.2.4 : On considère l'opérateur $M_q = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} t^k \sigma_p^j$ aux p -différences considéré comme agissant sur $\tilde{\mathcal{C}}_q[[t]]$. Le polygone de Newton de M_q est l'enveloppe convexe dans \mathbb{R}^2 des points $(k_j^-, k_j^- - j)$, $(k_j^-, -j)$, $(k_j^+ + j, -j)$ et $(k_j^+ + j, k_j^+)$ tels que $j = 0 \dots J$, $\Gamma_j = \{k = 0, \dots, K \mid \alpha_{k,j} \neq 0\} \neq \emptyset$, $k_j^- = \min \Gamma_j$ et $k_j^+ = \max \Gamma_j$.

Dans le cas où $|p| > 1$ la seule partie du polygone qui nous intéresse est celle de droite. On peut donc donner dans ce cas une définition ne nécessitant pas la recherche de k_j^- et k_j^+ .

Définition 4.2.5 : Soit $M_q = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} t^k \sigma_p^j$ avec $|p| > 1$. Soient $\mathfrak{S}_2 = \{(u, v) \in \mathbb{R}^2 \mid u \leq 0, v \geq 0\}$ et $\mathfrak{S}_3 = \{(u, v) \in \mathbb{R}^2 \mid u, v \leq 0\}$. Pour $(a, b) \in \mathbb{R}^2$ et $r = 2, 3$ on pose $\mathfrak{S}_r(a, b) = (a, b) + \mathfrak{S}_r$, $M_2(M_q)$ l'union des quadrants $\mathfrak{S}_2(k + j, -j)$ et $M_3(M_q)$ l'union des quadrants $\mathfrak{S}_3(k + j, k)$ pour (k, j) tels que $\alpha_{k,j} \neq 0$.

Soient $P_l(M_q)$ (resp. $P_u(M_q)$) l'enveloppe convexe dans \mathbb{R}^2 de $M_2(M_q)$ (resp. $M_3(M_q)$) et $P(M_q) = P_l(M_q) \cap P_u(M_q)$. Par définition, $P_l(M_q)$ est le polygone de Newton inférieur de M_q , $P_u(M_q)$ est le polygone de Newton supérieur et $P(M_q)$ le polygone de Newton de M_q .

Les pentes strictement positives de $P_l(M_q)$ sont appelées les pentes irrégulières inférieures de $P(M_q)$ et les pentes strictement négatives de $P_u(M_q)$ sont appelées les pentes régulières supérieures de $P(M_q)$.

On définit les points $(absci(\alpha), ordo(\alpha))$ et un sens de parcours des arêtes du polygone Γ comme dans [8]. Rappelons le principe dans le cas $|q| < 1$:

Si $\alpha \in \mathbb{R}^*$ et si l'intersection de $P_l(M_q)$ avec sa droite de contact de pente α est réduite à un unique point Γ on note indifféremment $(absci_1(\alpha), ordo_1(\alpha))$ ou $(absci_2(\alpha), ordo_2(\alpha))$ les coordonnées de ce point. Sinon Γ l'intersection est un segment (α est dans ce cas une pente irrégulière inférieure de $P(M_q)$) et est notée $[(absci_1(\alpha), ordo_1(\alpha)), (absci_2(\alpha), ordo_2(\alpha))]$ avec $absci_1(\alpha) < absci_2(\alpha)$.

Si $\alpha \in \mathbb{R}^*$ et si l'intersection de $P_u(M_q)$ avec sa droite de contact de pente α est réduite à un unique point Γ on note indifféremment $(absci_1(\alpha), ordo_1(\alpha))$ ou $(absci_2(\alpha), ordo_2(\alpha))$ les coordonnées de ce point. Sinon Γ l'intersection est un segment (α est dans ce cas une pente régulière supérieure de $P(M_q)$) $\Gamma[(absci_1(\alpha), ordo_1(\alpha)), (absci_2(\alpha), ordo_2(\alpha))]$ avec $absci_2(\alpha) < absci_1(\alpha)$.

Le côté horizontal inférieur de $P(M_q)$ est repéré à droite par $(absci_2(0_l), ordo_2(0_l))$. Le côté horizontal supérieur de $P(M_q)$ est repéré à droite par $(absci_1(0_u), ordo_1(0_u))$. On repère par $[(absci_1(\infty), ordo_1(\infty)), (absci_2(\infty), ordo_2(\infty))]$ le côté vertical droit de $P(M_q)$ avec $absci_1(\infty) = absci_2(\infty)$ et $ordo_1(\infty) \leq ordo_2(\infty)$.

En utilisant les résultats de J.-P. Ramis [8] et de J.-P. Bézivin [6] nous établissons aisément les théorèmes suivants :

Théorème 4.2.6 : Soit $q \in \mathbb{R}$, $|q| \neq 0, 1$. Soit $s \in \mathbb{R} \cup \{\infty\} \cup \{-\infty\}$. Soit $\alpha \in \mathbb{R}^* \cup \{0_l, 0_u, \infty\}$ défini par $\frac{1}{s}$ si $s \in \mathbb{R}^*$, 0_l si $s = -\infty$ et par 0_u si $s = \infty$. Soit $M_q \in \mathcal{C}[t, \sigma_p]$. Alors :

- Les opérateurs $M_q : \tilde{\mathcal{C}}_{q,s}[[t]] \rightarrow \tilde{\mathcal{C}}_{q,s}[[t]]$ sont de Fredholm, d'indice égal à $-ordo_1(\alpha)$;
- Les opérateurs $M_q : \tilde{\mathcal{C}}_{q,(s)}[[t]] \rightarrow \tilde{\mathcal{C}}_{q,(s)}[[t]]$ sont de Fredholm, d'indice égal à $-ordo_2(\alpha)$.

Théorème 4.2.7 : Soient $M_q \in \mathcal{C}[t, \sigma_p]$ et $\gamma_q = \sum_{n \geq 0} b_n(tq; q)_n$ q -Gevrey d'ordre 0. On suppose que $\hat{F}_q(t) \in \tilde{\mathcal{C}}_q[[t]]$ est solution de $M_q \hat{F}_q = \gamma_q$.

Alors \hat{F}_q est q -Gevrey d'ordre 0 ou il existe un unique réel $s > 0$ tel que \hat{F}_q est q -Gevrey d'ordre s et $\hat{F}_q \notin \tilde{\mathcal{C}}_{q,(s)}[[t]]$. De plus, $s = 1/k$ où k est l'une des pentes irrégulières inférieures de $P(M_q)$.

Théorème 4.2.8 : Soient $M_q \in \mathcal{C}[t, \sigma_p]$ et $\gamma_q \in \mathcal{C}[t]$. On suppose que $\hat{F}_q(t) \in \tilde{\mathcal{C}}_{q,(0)}[[t]]$ est solution de $M_q \hat{F}_q = \gamma_q$. Alors \hat{F}_q est un polynôme ou il existe un unique réel $s < 0$ tel que $\hat{F}_q \in \tilde{\mathcal{C}}_{q,s}[[t]]$ et $\hat{F}_q \notin \tilde{\mathcal{C}}_{q,(s)}[[t]]$. De plus, $s = 1/k$ où k est l'une des pentes régulières supérieures de $P(M_q)$.

5. Solutions séries de q -factorielles formelles

5.1. Polygone de Newton

Nous allons définir un polygone de Newton pour les opérateurs aux différences à coefficients polynomiaux en q^x . Il se confond en fait avec celui que nous avons défini précédemment pour les opérateurs aux p -différences considérés comme agissant sur $\tilde{\mathcal{C}}_q[[t]]$.

Définition 5.1.1 : Soit $\Delta_q = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} \tau^k (q^x)^j$. Le polygone de Newton $P(\Delta_q)$ est par définition l'enveloppe convexe dans \mathbb{R}^2 des points $(k_j^-, k_j^- - j)$, $(k_j^-, -j)$, $(k_j^+ + j, -j)$ et $(k_j^+ + j, k_j^+)$ tels que $j = 0 \dots J$, $\Gamma_j = \{k = 0, \dots, K \mid \alpha_{k,j} \neq 0\} \neq \emptyset$, $k_j^- = \min \Gamma_j$ et $k_j^+ = \max \Gamma_j$.

Dans le cas $|q| < 1$ on peut donner une autre définition :

Définition 5.1.2 : Soit $\Delta_q = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} \tau^k (q^x)^j$ avec $|q| < 1$. Soient $\mathfrak{S}_2 = \{(u, v) \in \mathbb{R}^2 | u \leq 0, v \geq 0\}$ et $\mathfrak{S}_3 = \{(u, v) \in \mathbb{R}^2 | u, v \leq 0\}$. Pour $(a, b) \in \mathbb{R}^2$ et $r = 2, 3$ on pose $\mathfrak{S}_r(a, b) = (a, b) + \mathfrak{S}_r$, $M_2(\Delta_q)$ l'union des quadrants $\mathfrak{S}_2(k+j, -j)$ et $M_3(\Delta_q)$ l'union des quadrants $\mathfrak{S}_3(k+j, k)$ pour (k, j) tels que $\alpha_{k,j} \neq 0$.

Soient $P_l(\Delta_q)$ (resp. $P_u(\Delta_q)$) l'enveloppe convexe dans \mathbb{R}^2 de $M_2(\Delta_q)$ (resp. $M_3(\Delta_q)$) et $P(\Delta_q) = P_l(\Delta_q) \cap P_u(\Delta_q)$. Par définition, $P_l(\Delta_q)$ est le polygone de Newton inférieur de Δ_q , $P_u(\Delta_q)$ est le polygone de Newton supérieur et $P(\Delta_q)$ le polygone de Newton de Δ_q .

Les pentes strictement positives de $P_l(\Delta_q)$ sont les pentes irrégulières inférieures et les pentes strictement négatives de $P_u(\Delta_q)$ sont appelées les pentes régulières supérieures de $P(\Delta_q)$.

On définit comme précédemment les points $(absci(\alpha), ordo(\alpha))$.

5.2. Caractérisation q -Gevrey des séries de q -factorielles solutions

On rappelle que pour $x \in \mathbb{C}$ et $n \in \mathbb{N}^* \Gamma$

$$\frac{\Gamma_q(x)}{\Gamma_q(x+n)} = \frac{1-q}{1-q^{x+n-1}} \cdots \frac{1-q}{1-q^x}.$$

Définition 5.2.1 : La série formelle de q -factorielles $\hat{f}_q = \sum_{n \geq 1} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)}$ est q -Gevrey d'ordre $s \in \mathbb{R}$ s'il existe C et $A > 0$ tels que

$$\forall n \geq 0, \quad \left| \frac{a_{n+1}}{\Gamma_q(n+1)} \right| < C|q|^{-\frac{sn(n+1)}{2}} A^n.$$

On note $\hat{O}_{f,q}(s)$ l'ensemble de telles séries.

Nous rappelons la définition suivante :

Définition 5.2.2 : La série $\hat{F}_q(t) = \sum_{n \geq 0} a_n(tq; q)_n$ est q -Gevrey d'ordre $s \in \mathbb{R}$ s'il existe C et $A > 0$ tels que $\forall n \geq 0, \quad |a_{n+1}| < C|q|^{-\frac{sn(n+1)}{2}} A^n$.

Remarque : Si $\hat{f}_q = \sum_{n \geq 1} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)}$ est q -Gevrey d'ordre s alors sa transformée de q -Mellin formelle $IM_q(\hat{f}_q) = \sum_{n \geq 1} \frac{a_n}{\Gamma_q(n)} (tq; q)_{n-1}$ est q -Gevrey d'ordre s .

Théorème 5.2.3 : Soient $\Delta_q \in \mathbb{C}[q^x, \tau]$ et \hat{g}_q une série de q -factorielles q -Gevrey d'ordre 0. Soit $\hat{f}_q = \sum_{n \geq 1} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)}$ solution de $\Delta_q \hat{f}_q = \hat{g}_q$. Alors \hat{f}_q est q -Gevrey d'ordre 0 ou il existe un unique réel $s > 0$ tel que \hat{f}_q est q -Gevrey d'ordre s optimal.

Remarque : On dit qu'une série est q -Gevrey d'ordre s optimal s'il n'existe pas de réel $s' < s$ tel que cette série soit q -Gevrey d'ordre s' .

Démonstration : Soient $M_q = IM_q(\Delta_q) \Gamma \hat{F}_q = \sum_{n \geq 1} \frac{a_n}{\Gamma_q(n)} (tq; q)_{n-1}$ et $\hat{g}_q = IM_q(\hat{g}_q)$. D'après

la proposition 3.2.6 $\Gamma \hat{F}_q$ est solution de l'équation aux p -différences $M_q \hat{F}_q = \hat{\gamma}_q$ où $\hat{\gamma}_q$ est q -Gevrey d'ordre 0. On utilise alors le théorème 4.2.7.

6. Théorèmes d'indices (séries de q -factorielles)

6.1. Indices formels

Soit $\hat{K}_{f,q} = \{\hat{f}_q = \sum_{n \geq n_0} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \mid n_0 \in \mathbb{Z}\}$. Pour $\hat{f}_q \in \hat{K}_{f,q}$ on définit $val(\hat{f}_q) = \inf\{n \in \mathbb{Z} \mid a_n \neq 0\}$ avec la convention $val(0) = \infty$. Soit $\hat{O}_{f,q} = \{\hat{f}_q \in \hat{K}_{f,q} \mid val(\hat{f}_q) \geq 1\}$.

Lemme 6.1.1 : La suite d'espaces vectoriels

$$0 \longrightarrow \mathcal{C}[q^x] \longrightarrow \hat{K}_{f,q} \xrightarrow{\hat{M}_q} \tilde{\mathcal{C}}_q[[t]] \longrightarrow 0$$

est exacte.

Démonstration : Il suffit de remarquer d'une part que $\mathcal{C}[q^x]$ peut être considéré comme le sous-espace de $\hat{K}_{f,q}$ tel que

$$\mathcal{C}[q^x] = \left\{ \hat{f}_q = \sum_{n \geq n_0} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \mid n_0 \leq 0 \text{ et } a_n = 0 \ \forall n \geq 1 \right\}.$$

D'autre part soit $\hat{f}_q(x) = \sum_{n \geq n_0} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \in \hat{K}_{f,q} : \hat{M}_q(\hat{f}_q)(t) = \sum_{n \geq n_0} \frac{a_n}{\Gamma_q(n)} \frac{(tq;q)_\infty}{(tq^n;q)_\infty}$. Or $\frac{1}{\Gamma_q(n)} = 0$ pour $n = 0, -1, \dots$ donc

$$IM_q(\hat{f}_q)(t) = \sum_{n \geq \max(n_0 - 1, 0)} \frac{a_{n+1}}{\Gamma_q(n+1)} (tq;q)_n \in \tilde{\mathcal{C}}_q[[t]].$$

On en déduit que l'application $IM_q : \hat{K}_{f,q} \longrightarrow \tilde{\mathcal{C}}_q[[t]]$ est surjective.

Soit $\Delta_q \in \mathcal{C}[q^x, \tau]$; Δ_q opère sur $\mathcal{C}[q^x]$. En effet pour $k \in \mathbb{N}$ et $n \leq 0$

$$\tau^k < q^x >_j \left(\frac{\Gamma_q(x)}{\Gamma_q(x+n)} \right) = \sum_{i=0}^{j+k} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j-n-i)(j+k-i)} \frac{\Gamma_q(j-n+1)}{\Gamma_q(j-n+1-i)} \frac{\Gamma_q(x)}{\Gamma_q(x-j+n+i)}.$$

Or $\frac{\Gamma_q(j-n+1)}{\Gamma_q(j-n+1-i)} = \prod_{l=0}^{i-1} \frac{1-q^{j-n-l}}{1-q}$ s'annule pour $i \geq 1+j-n$ donc pour $n \leq 0$

$$\tau^k < q^x >_j \left(\frac{\Gamma_q(x)}{\Gamma_q(x+n)} \right) =$$

$$\sum_{i=0}^{\min(j+k, j-n)} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j-n-i)(j+k-i)} \frac{\Gamma_q(j-n+1)}{\Gamma_q(j-n+1-i)} \frac{\Gamma_q(x)}{\Gamma_q(x-j+n+i)} \in \mathcal{C}[q^x].$$

Il est évident que Δ_q opère sur $\hat{K}_{f,q}$. Le diagramme suivant est commutatif (avec $M_q = IM_q(\Delta_q)$):

$$\begin{array}{ccccc} & \hat{K}_{f,q} & \xrightarrow{IM_q} & \tilde{\mathcal{C}}_q[[t]] & \\ \Delta_q & \downarrow & & \downarrow & M_q \\ & \hat{K}_{f,q} & \xrightarrow{IM_q} & \tilde{\mathcal{C}}_q[[t]] & \end{array}$$

En général $\Gamma\Delta_q$ n'opère pas sur $\hat{O}_{f,q}$. Cependant nous pouvons établir le lemme suivant :

Lemme 6.1.2 : On note $\hat{O}_{f,q}(\Delta_q) = \{\hat{f}_q \in \hat{O}_{f,q} \mid \Delta_q(\hat{f}_q) \in \hat{O}_{f,q}\}$.

L'espace quotient $\hat{O}_{f,q}/\hat{O}_{f,q}(\Delta_q)$ est de dimension finie, et est isomorphe à

$$\tilde{\mathcal{C}}_q[[t]]/IM_q(\hat{O}_{f,q}(\Delta_q)).$$

Démonstration : Soit $\hat{f}_q(x) = \sum_{n \geq 1} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \in \hat{O}_{f,q}$.

Si $\Delta_q = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} \tau^k < q^x >_j$ (avec $\exists k, \alpha_{k,j} \neq 0$ et $\exists j, \alpha_{k,j} \neq 0$) alors $\Delta_q \hat{f}_q$ est égal à

$$\begin{aligned} & \sum_{k=0}^K \sum_{j=0}^J \sum_{n \geq 1} \sum_{i=0}^{j+k} \alpha_{k,j} a_n \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j-n-i)(j+k-i)} \frac{\Gamma_q(j-n+1)}{\Gamma_q(j-n+1-i)} \frac{\Gamma_q(x)}{\Gamma_q(x-j+n+i)} \\ &= \sum_{k=0}^K \sum_{j=0}^J \sum_{i=0}^{j+k} \sum_{n \geq 1-j+i} \alpha_{k,j} a_{n+j-i} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{-n(j+k-i)} \frac{\Gamma_q(-n+i+1)}{\Gamma_q(-n+1)} \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \end{aligned}$$

donc $\Delta_q(\hat{f}_q) \in \hat{O}_{f,q}$ s'écrit en annulant dans $\Delta_q(\hat{f}_q)$ les coefficients de $\frac{\Gamma_q(x)}{\Gamma_q(x+n)}$ pour $n = 1 - J, \dots, 0$: $S_i(a_1, \dots, a_i) = 0$ pour $i = 1, \dots, J$. Par conséquent $\hat{O}_{f,q}/\hat{O}_{f,q}(\Delta_q)$ est de dimension finie.

D'autre part les applications $IM_q : \hat{O}_{f,q} \longrightarrow \tilde{\mathcal{C}}_q[[t]]$ et $IM_q : \hat{O}_{f,q}(\Delta_q) \longrightarrow IM_q(\hat{O}_{f,q}(\Delta_q))$ sont bijectives donc $\bar{M}_q : \hat{O}_{f,q}/\hat{O}_{f,q}(\Delta_q) \longrightarrow \tilde{\mathcal{C}}_q[[t]]/IM_q(\hat{O}_{f,q}(\Delta_q))$ est un isomorphisme.

Proposition 6.1.3 : Soit $\Delta_q = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} \tau^k < q^x >_j$.

L'opérateur $\Delta_q : \hat{O}_{f,q}(\Delta_q) \longrightarrow \hat{O}_{f,q}$ est de Fredholm, d'indice :

$$\chi_{\infty}^+(\Delta_q) = \mu_q - d_q$$

avec $\mu_q = \sup\{j = 0 \dots J \mid \exists k = 0 \dots K, \alpha_{k,j} \neq 0\}$ et $d_q = \dim(\hat{O}_{f,q}/\hat{O}_{f,q}(\Delta_q))$.

Démonstration : Le diagramme suivant est commutatif :

$$\begin{array}{ccccccc} & & IM_q & & & & \\ 0 & \longrightarrow & \hat{O}_{f,q}(\Delta_q) & \longrightarrow & \tilde{\mathcal{C}}_q[[t]] & \longrightarrow & \tilde{\mathcal{C}}_q[[t]]/IM_q(\hat{O}_{f,q}(\Delta_q)) \longrightarrow 0 \\ & & \Delta_q \downarrow & & \downarrow M_q & & \downarrow \bar{M}_q \\ 0 & \longrightarrow & \hat{O}_{f,q} & \longrightarrow & \tilde{\mathcal{C}}_q[[t]] & \longrightarrow & \tilde{\mathcal{C}}_q[[t]]/IM_q(\hat{O}_{f,q}) \longrightarrow 0 \\ & & IM_q & & & & \end{array}$$

où $M_q = IM_q(\Delta_q) = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} t^k < \sigma_p >_j \Gamma$ d'indice μ_q ; $\hat{\mathcal{C}}_q[[t]]/IM_q(\hat{O}_{f,q}) = \{0\}$ donc \bar{M}_q est à indice Γ égal à $d_q \Gamma$ d'où le résultat.

Lemme 6.1.4 : Soit $\Delta_q = \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} \tau^k < q^x >_j$.

L'opérateur $\Delta_q : \hat{K}_{f,q}/\hat{O}_{f,q}(\Delta_q) \longrightarrow \hat{K}_{f,q}/\hat{O}_{f,q}$ est de Fredholm, d'indice : $-\mu_q + d_q$.

Démonstration : Soit $r \in \mathbb{N}$. On note $\hat{K}_{q,-r}$ les éléments de $\hat{K}_{f,q}$ de valuation supérieure ou égale à $-r$. Si $\hat{f}_q(x) = \sum_{n \geq -r} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \in \hat{K}_{q,-r}$ alors $\Delta_q \hat{f}_q(x)$ est égal à

$$\sum_{n \geq -r} a_n \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} \sum_{i=0}^{j+k} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j-n-i)(j+k-i)} \frac{\Gamma_q(j-n+1)}{\Gamma_q(j-n+1-i)} \frac{\Gamma_q(x)}{\Gamma_q(x-j+n+i)},$$

c'est-à-dire $\Gamma \Delta_q(\hat{K}_{q,-r}) \subset \hat{K}_{q,-r-\mu_q}$ et $\Delta_q : \hat{K}_{f,q}/\hat{K}_{q,-r} \longrightarrow \hat{K}_{f,q}/\hat{K}_{q,-r-\mu_q}$.

Pour $m \geq r$:

$$\begin{aligned} \Delta_q \left(\frac{\Gamma_q(x)}{\Gamma_q(x-m)} \right) &= \sum_{k=0}^K \sum_{j=0}^J \alpha_{k,j} \sum_{i=0}^{j+k} \begin{bmatrix} j+k \\ i \end{bmatrix} q^{(j+m-i)(j+k-i)} \frac{\Gamma_q(j+m+1)}{\Gamma_q(j+m+1-i)} \frac{\Gamma_q(x)}{\Gamma_q(x-j-m+i)} \\ &= \sum_{(k,j) | \alpha_{k,j} \neq 0} \alpha_{k,j} q^{(j+m)(j+k)} \frac{\Gamma_q(x)}{\Gamma_q(x-j-m)} + \text{termes de valuation} > -j-m \\ &= \left(\sum_{k \in \Gamma} \alpha_{k,\mu_q} q^{(\mu_q+m)(\mu_q+k)} \right) \frac{\Gamma_q(x)}{\Gamma_q(x-\mu_q-m)} + \text{termes de valuation} > -\mu_q-m. \end{aligned}$$

Or Γ avec $k_0 = \inf \Gamma$

$$\sum_{k \in \Gamma} \alpha_{k,\mu_q} q^{(\mu_q+m)(\mu_q+k)} = \alpha_{k_0,\mu_q} q^{(\mu_q+m)(\mu_q+k_0)} \left(1 + \sum_{k \in \Gamma, k > k_0} \frac{\alpha_{k,\mu_q}}{\alpha_{k_0,\mu_q}} q^{(\mu_q+m)(k-k_0)} \right)$$

ou Γ avec $k_1 = \sup \Gamma$

$$\sum_{k \in \Gamma} \alpha_{k,\mu_q} q^{(\mu_q+m)(\mu_q+k)} = \alpha_{k_1,\mu_q} q^{(\mu_q+m)(\mu_q+k_1)} \left(1 + \sum_{k \in \Gamma, k < k_1} \frac{\alpha_{k,\mu_q}}{\alpha_{k_1,\mu_q}} q^{(\mu_q+m)(k-k_1)} \right).$$

Par conséquent si r est assez grand $\sum_{k \in \Gamma} \alpha_{k,\mu_q} q^{(\mu_q+m)(\mu_q+k)}$ est non nul pour tout $m \geq r$ et on en déduit que l'application $\Delta_q : \hat{K}_{f,q}/\hat{K}_{q,-r} \longrightarrow \hat{K}_{f,q}/\hat{K}_{q,-r-\mu_q}$ est un isomorphisme.

Les diagrammes suivants sont commutatifs :

$$\begin{array}{ccccccc} 0 & \longrightarrow & \hat{K}_{q,-r} & \longrightarrow & \hat{K}_{f,q} & \longrightarrow & \hat{K}_{f,q}/\hat{K}_{q,-r} \longrightarrow 0 \\ & & \Delta_q \downarrow & & \downarrow \Delta_q & & \downarrow \wr \\ 0 & \longrightarrow & \hat{K}_{q,-r-\mu_q} & \longrightarrow & \hat{K}_{f,q} & \longrightarrow & \hat{K}_{f,q}/\hat{K}_{q,-r-\mu_q} \longrightarrow 0 \end{array}$$

$$\begin{array}{ccccccc} 0 \rightarrow & \hat{K}_{q,-r}/\hat{O}_{f,q}(\Delta_q) & \longrightarrow & \hat{K}_{f,q}/\hat{O}_{f,q}(\Delta_q) & \longrightarrow & (\hat{K}_{f,q}/\hat{K}_{q,-r})/\hat{O}_{f,q}(\Delta_q) & \rightarrow 0 \\ & \Delta_q \downarrow & & \downarrow \Delta_q & & \downarrow \wr & \\ 0 \rightarrow & \hat{K}_{q,-r-\mu_q}/\hat{O}_{f,q} & \longrightarrow & \hat{K}_{f,q}/\hat{O}_{f,q} & \longrightarrow & (\hat{K}_{f,q}/\hat{K}_{q,-r-\mu_q})/\hat{O}_{f,q} & \rightarrow 0 \end{array}$$

Les espaces $\hat{K}_{q,-r-\mu_q}/\hat{O}_{f,q}$ et $\hat{K}_{q,-r}/\hat{O}_{f,q}(\Delta_q)$ étant de dimension finie Γ respectivement égale à $r + \mu_q + 1$ et à $\dim(\hat{K}_{q,-r}/\hat{O}_{f,q}) + \dim(\hat{O}_{f,q}/\hat{O}_{f,q}(\Delta_q))$ il s'ensuit que $\Delta_q : \hat{K}_{f,q}/\hat{O}_{f,q}(\Delta_q) \longrightarrow \hat{K}_{f,q}/\hat{O}_{f,q}$ est à indice Γ égal à : $r + 1 + \dim(\hat{O}_{f,q}/\hat{O}_{f,q}(\Delta_q)) - r - \mu_q - 1$.

Proposition 6.1.5 : Soit $\Delta_q \in \mathcal{C}[q^x, \tau]$.

- L'opérateur $\Delta_q : \hat{K}_{f,q} \longrightarrow \hat{K}_{f,q}$ est de Fredholm, d'indice 0;
- L'opérateur $\Delta_q : \mathcal{C}[q^x] \longrightarrow \mathcal{C}[q^x]$ est de Fredholm, d'indice $-\mu_q$.

Démonstration : En considérant le diagramme commutatif suivant

$$\begin{array}{ccccccc} 0 & \longrightarrow & \hat{O}_{f,q}(\Delta_q) & \longrightarrow & \hat{K}_{f,q} & \longrightarrow & \hat{K}_{f,q}/\hat{O}_{f,q}(\Delta_q) \longrightarrow 0 \\ & & \Delta_q \downarrow & & \Delta_q \downarrow & & \Delta_q \downarrow \\ 0 & \longrightarrow & \hat{O}_{f,q} & \longrightarrow & \hat{K}_{f,q} & \longrightarrow & \hat{K}_{f,q}/\hat{O}_{f,q} \longrightarrow 0 \end{array}$$

il s'ensuit que $\Delta_q : \hat{K}_{f,q} \longrightarrow \hat{K}_{f,q}$ est de Fredholm Γ d'indice 0. Le deuxième résultat de la proposition résulte du diagramme commutatif

$$\begin{array}{ccccccc} & & & IM_q & & & \\ 0 & \longrightarrow & \mathcal{C}[q^x] & \longrightarrow & \hat{K}_{f,q} & \longrightarrow & \tilde{\mathcal{C}}_q[[t]] \longrightarrow 0 \\ & & \Delta_q \downarrow & & \Delta_q \downarrow & & \downarrow \hat{M}_q(\Delta_q) \\ 0 & \longrightarrow & \mathcal{C}[q^x] & \longrightarrow & \hat{K}_{f,q} & \longrightarrow & \tilde{\mathcal{C}}_q[[t]] \longrightarrow 0 \\ & & & & IM_q & & \end{array}$$

6.2. Indices q -Gevrey

Pour $s \in \mathbb{R}\Gamma$ on note $\hat{O}_{f,q}(s)$ l'ensemble des séries $\hat{f}_q = \sum_{n \geq 1} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)}$ telles qu'il existe C et $A > 0$ tels que :

$$\forall n \geq 0, \quad |a_{n+1}| < C |p|^{\frac{sn(n+1)}{2}} |\Gamma_q(n+1)| A^n$$

et on note $\hat{K}_{f,q}(s) = \hat{O}_{f,q}(s) \oplus \mathcal{C}[q^x]$.

Tout opérateur $\Delta_q \in \mathcal{C}[q^x, \tau]$ opère sur $\hat{K}_{f,q}(s)$: pour k et $j \in \mathbb{N}\Gamma$

$$\begin{aligned} \tau^k < q^x >_j \hat{f}_q &= \sum_{n \geq 1} \sum_{i=0}^{j+k} a_n q^{(j-n-i)(j+k-i)} \frac{\Gamma_q(j-n+1)}{\Gamma_q(j-n+1-i)} \frac{\Gamma_q(x)}{\Gamma_q(x-j+n+i)} \\ &= \sum_{i=0}^{j+k} \sum_{n \geq 1-j+i} a_{n+j-i} q^{-n(j+k-i)} \frac{\Gamma_q(-n+1+i)}{\Gamma_q(-n+1)} \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \\ &= \sum_{n \geq 1-j} b_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \text{ avec } b_n = \sum_{i=0}^{\min(j+k, n-1+j)} a_{n+j-i} q^{-n(j+k-i)} \frac{\Gamma_q(-n+1+i)}{\Gamma_q(-n+1)}. \end{aligned}$$

Si $\hat{f}_q \in \hat{O}_{f,q}(s)$ alors pour $N \in \mathbb{N}$ assez grand $\sum_{n \geq N} b_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \in \hat{O}_{f,q}(s) : \sum_{n \geq 1} b_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \in \hat{O}_{f,q}(s)$. Donc $\tau^k < q^x >_j \hat{f}_q - \sum_{n \geq 1} b_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \in \mathcal{C}[q^x]$.

Soit $\hat{O}_{f,q}(s)(\Delta_q)$ le sous-espace de $\hat{O}_{f,q}(s)$ tel que $\Delta_q(\hat{O}_{f,q}(s)(\Delta_q)) \subset \hat{O}_{f,q}(s)$.

Théorème 6.2.1 : Soit $\Delta_q \in \mathcal{C}[q^x, \tau]$. Soient $s \in \mathbb{R}$ et $\alpha = 1/s$ pour $s \neq 0$, $\alpha = \infty$ pour $s = 0$.

- L'opérateur $\Delta_q : \hat{K}_{f,q}(s) \longrightarrow \hat{K}_{f,q}(s)$ est de Fredholm, d'indice égal à $\chi_{q,s}(\Delta_q) = -\mu_q - \text{ordo}_1(\alpha)$;
- L'opérateur $\Delta_q : \hat{O}_{f,q}(s)(\Delta_q) \longrightarrow \hat{O}_{f,q}(s)$ est de Fredholm, d'indice égal à $\chi_{q,s}^+(\Delta_q) = -\text{ordo}_1(\alpha) - d_q$.

Démonstration : Soit $\hat{f}_q = \sum_{n \geq n_0} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n)} \in \hat{K}_{f,q}(s)$ ($n_0 \in \mathbb{Z}$) : sa transformée de q -Mellin formelle $IM_q(\hat{f}_q) = \sum_{n \geq \max(n_0-1, 0)} \frac{a_{n+1}}{\Gamma_q(n+1)} (tq; q)_n$ appartient à $\tilde{\mathcal{C}}_{q,s}[[t]]$ et $IM_q(\hat{K}_{f,q}(s)) = \tilde{\mathcal{C}}_{q,s}[[t]]$. Le diagramme

$$\begin{array}{ccccccc}
0 & \longrightarrow & \mathcal{C}[q^x] & \longrightarrow & \hat{K}_{f,q}(s) & \xrightarrow{IM_q} & \tilde{\mathcal{C}}_{q,s}[[t]] \longrightarrow 0 \\
& & \Delta_q \downarrow & & \Delta_q \downarrow & & \downarrow \hat{M}_q(\Delta_q) \\
0 & \longrightarrow & \mathcal{C}[q^x] & \longrightarrow & \hat{K}_{f,q}(s) & \xrightarrow{IM_q} & \tilde{\mathcal{C}}_{q,s}[[t]] \longrightarrow 0
\end{array}$$

étant commutatif $\Gamma \Delta_q : \hat{K}_{f,q}(s) \longrightarrow \hat{K}_{f,q}(s)$ est de Fredholm d'indice égal à la somme de l'indice de $\Delta_q : \mathcal{C}[q^x] \longrightarrow \mathcal{C}[q^x]$ plus celui de $IM_q(\Delta_q) : \tilde{\mathcal{C}}_{q,s}[[t]] \longrightarrow \tilde{\mathcal{C}}_{q,s}[[t]]$ d'où le résultat.

L'indice de $\Delta_q : \hat{O}_{f,q}(s)(\Delta_q) \longrightarrow \hat{O}_{f,q}(s)$ s'établit comme dans le précédent paragraphe et est égal à $\chi_{q,s}^+(\Delta_q) = -\text{ordo}_1(\alpha) - d_s$ où $d_s = \dim \left(\hat{O}_{f,q}(s) / \hat{O}_{f,q}(s)(\Delta_q) \right) = d_q$.

7. Transformée de q -Laplace formelle

7.1. Définitions

Pour $Re(x) > 0$ et $n \in \mathbb{N}$

$$\frac{1}{x^{n+1}} = \int_0^{+\infty} e^{-tx} \frac{t^n}{\Gamma(n+1)} dt.$$

Plus généralement ΓA . Barkatou et A. Duval définissent une transformée de Laplace formelle :

$$\hat{f}(x) = \sum_{n \geq 0} a_n x^{-(\lambda+n+1)} \longrightarrow \hat{\phi}(z) = \sum_{n \geq 0} \frac{a_n}{\Gamma(\lambda+n+1)} z^{\lambda+n}.$$

Pour définir une transformée de q -Laplace formelle Γ nous allons donner un q -analogue de la relation $\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt$.

Pour $Re(x) > 0$ et $0 < q < 1$

$$\Gamma_q(x) = (q; q)_\infty (1-q)^{1-x} \sum_{n=0}^{\infty} \frac{q^{nx}}{(q; q)_n}.$$

Or $\frac{(q;q)_\infty}{(q;q)_n} = (q^{n+1}; q)_\infty$ donc pour $Re(x) > 0$

$$\begin{aligned}\Gamma_q(x) &= (1-q) \sum_{n=0}^{\infty} (1-q)^{-x} q^{n(x-1)} (q^{n+1}; q)_\infty q^n \\ &= \sum_{n=0}^{\infty} \frac{q^{n(x-1)}}{(1-q)^{x-1}} (q^{n+1}; q)_\infty q^n = \sum_{n=0}^{\infty} \left(\frac{q^n}{1-q}\right)^{x-1} \left(\frac{q^n}{1-q} q(1-q); q\right)_\infty q^n\end{aligned}$$

et

$$\Gamma_q(x) = \int_0^{\frac{1}{1-q}} t^{x-1} (tq(1-q); q)_\infty d_q t.$$

En particulier $\Gamma_q(n+1) = \int_0^{\frac{1}{1-q}} t^n (tq(1-q); q)_\infty d_q t$. Par le changement de variable $t \mapsto \xi q^x$:

$$\int_0^{\frac{1}{q^x(1-q)}} \xi^n Exp_q(-q\xi q^x) d_q t = q^{-x(n+1)} \Gamma_q(n+1)$$

où $Exp_q(x) = (-(1-q)x; q)_\infty$ pour $|q| < 1$.

Il est donc naturel de définir une transformation de q -Laplace formelle par

$$\mathbb{L}_q(\hat{F}_q)(x) = \int_0^{\frac{q^{-x}}{1-q}} \hat{F}_q(\xi) Exp_q(-q^{x+1}\xi) d_q t$$

et alors

$$q^{-x(n+1)} \Gamma_q(n+1) = \mathbb{L}(\xi^n)(x).$$

Définition 7.1.1 : Soit $\hat{f}_q(x) = \sum_{n \geq 0} a_n q^{-x(\lambda+n+1)}$. Sa transformée de q -Laplace formelle est la série

$$\mathbb{L}_q(\hat{f}_q)(z) = \sum_{n \geq 0} \frac{a_n}{\Gamma_q(\lambda + n + 1)} z^{\lambda+n}.$$

Remarque : Soit $\hat{f}_q(x) = \sum_{n \geq 0} a_n q^{-x(n+1)} \in q^{-x}\mathcal{C}[[q^{-x}]]$ et $\hat{F}_q(x) = \mathbb{L}_q \hat{f}_q = \sum_{n \geq 0} \frac{a_n}{\Gamma_q(n+1)} z^n \in \mathcal{C}[[z]]$.

Comme $\Gamma_q(n+1) = \frac{(q;q)_n}{(1-q)^n}$:

Si $|q| < 1$:

$$\Gamma_q(n+1) = \frac{(q;q)_\infty}{(q^{n+1}; q)_\infty} \frac{1}{(1-q)^n} \sim_{n \rightarrow +\infty} \frac{(q;q)_\infty}{(1-q)^n}.$$

Si $|q| > 1$:

$$\begin{aligned}\Gamma_q(n+1) &= (p;p)_n (-q)^n q^{\frac{n(n-1)}{2}} \frac{1}{(1-q)^n} = \frac{(p;p)_\infty}{(p^{n+1}; p)_\infty} (-1)^n q^{\frac{n(n+1)}{2}} \frac{1}{(1-q)^n} \\ &\sim_{n \rightarrow +\infty} \frac{(p;p)_\infty}{(1-q)^n} (-1)^n q^{\frac{n(n+1)}{2}}.\end{aligned}$$

Donc $\mathbb{L}_q : q^{-x}\mathcal{C}[[q^{-x}]]_{q,s} \longrightarrow \mathcal{C}[[z]]_{q,s}$ si $|q| < 1$ et $\mathbb{L}_q : q^{-x}\mathcal{C}[[q^{-x}]]_{q,s+1} \longrightarrow \mathcal{C}[[z]]_{q,s}$ si $|q| > 1$.

Définition 7.1.2 : Soit $\Delta_q \in \mathcal{C}[q^x, \tau]$. Sa transformée de q -Laplace formelle est l'opérateur $L_q \in \mathcal{C}[z^{-1}, \sigma_p, \sigma_q]$ image de Δ_q par la correspondance $q^x \mapsto \frac{1}{z} \frac{1-\sigma_q}{1-q}$, $\tau \mapsto p\sigma_p$.

Remarque : $\mathbb{I}_q(q^x \tau) = \frac{1}{z} \frac{1-\sigma_q}{1-q} p \sigma_p = \frac{p}{1-q} \frac{1}{z} (\sigma_p - 1) = p \frac{1}{z} \sigma_p \frac{1-\sigma_q}{1-q} = p^2 \sigma_p \frac{1-\sigma_q}{1-q} = \mathbb{I}_q(p \tau q^x)$ et $q^x \tau = p \tau q^x$.

Remarque : Définissons formellement la transformée de q -Laplace formelle par la q -intégrale ($|q| < 1$)

$$\hat{f}_q(x) = \int_0^{\frac{1}{q^x(1-q)}} \hat{F}_q(\xi) ((1-q)q^{x+1}\xi; q)_\infty d_q \xi = \sum_{l=0}^{\infty} q^{l-x} \hat{F}_q\left(\frac{q^{l-x}}{1-q}\right) (q^{l+1}; q)_\infty.$$

Alors

$$\tau \hat{f}_q(x) = \sum_{l=0}^{\infty} q^{l-x-1} \hat{F}_q\left(\frac{q^{l-x-1}}{1-q}\right) (q^{l+1}; q)_\infty = p \sum_{l=0}^{\infty} q^{l-x} \sigma_p(\hat{F}_q)\left(\frac{q^{l-x}}{1-q}\right) (q^{l+1}; q)_\infty = \mathbb{I}_q(p \sigma_p(\hat{F}_q))$$

et

$$\begin{aligned} \mathbb{I}_q\left(\frac{1}{\xi} \frac{1-\sigma_q}{1-q}(\hat{F}_q)\right) &= \sum_{l=0}^{\infty} (\hat{F}_q\left(\frac{q^{l-x}}{1-q}\right) - \hat{F}_q\left(\frac{q^{l+1-x}}{1-q}\right)) (q^{l+1}; q)_\infty \\ &= \sum_{l=0}^{\infty} \hat{F}_q\left(\frac{q^{l-x}}{1-q}\right) (q^{l+1}; q)_\infty - \sum_{l=1}^{\infty} \hat{F}_q\left(\frac{q^{l-x}}{1-q}\right) (q^l; q)_\infty \\ &= \hat{F}_q\left(\frac{q^{-x}}{1-q}\right) + \sum_{l=1}^{\infty} \hat{F}_q\left(\frac{q^{l-x}}{1-q}\right) (q^{l+1}; q)_\infty q^l = q^x \hat{f}_q(x). \end{aligned}$$

7.2. Solutions formelles

Proposition 7.2.1 : Pour tout opérateur $\Delta_q \in \mathcal{C}[q^x, \tau]$ et pour toute série formelle $\hat{f}_q(x) = \sum_{n \geq 0} a_n q^{-x(\lambda+n+1)}$, $\mathbb{I}_q(\Delta_q \hat{f}_q) = \mathbb{I}_q(\Delta_q) \mathbb{I}_q(\hat{f}_q)$.

Démonstration : Il suffit de montrer le résultat pour $\Delta_q = \tau^j (q^x)^i$ et $\hat{f}_q(x) = q^{-x(\lambda+1)}$ pour tous $i, j \in \mathbb{N}$ et $\lambda \in \mathbb{C}$.

Comme $\Delta_q \hat{f}_q = q^{-j(\lambda-i+1)} q^{-x(\lambda-i+1)} \Gamma \mathbb{I}_q(\Delta_q \hat{f}_q) = \frac{q^{-j(\lambda-i+1)}}{\Gamma_q(\lambda-i+1)} z^{\lambda-i}$. D'autre part $\Gamma \mathbb{I}_q(\hat{f}_q) = \frac{1}{\Gamma_q(\lambda+1)} z^\lambda$ et $\mathbb{I}_q(\Delta_q) = (p \sigma_p)^j \left(\frac{1}{z} \frac{1-\sigma_q}{1-q}\right)^i$. La proposition se montre alors aisément par récurrence sur i et j .

Théorème 7.2.2 : Soient $\Delta_q \in \mathcal{C}[q^x, \tau]$, $\hat{f}_q(x) = \sum_{n \geq 0} a_n q^{-x(n+1)}$ une série formelle en $\frac{1}{q^x}$ et $\hat{g}_q(x) = \sum_{n \geq 0} b_n q^{-x(n+1)}$ q -Gevrey d'ordre 1 si $|q| > 1$ (resp. q -Gevrey d'ordre 0 si $|q| < 1$). Alors si $|q| > 1$, \hat{f}_q est q -Gevrey d'ordre 1 ou il existe un unique $s > 0$ optimal tel que \hat{f}_q soit q -Gevrey d'ordre $s+1$ (resp. \hat{f}_q est q -Gevrey d'ordre 0 ou il existe un unique $s > 0$ optimal tel que \hat{f}_q soit q -Gevrey d'ordre s).

Démonstration : On vérifie d'une part que L_q agit sur $\mathcal{C}[[z]]$.

D'autre part si $\Delta_q = \sum_{j=0}^J \sum_{k=0}^K \alpha_{k,j} \tau^k (q^x)^j$ alors

$$\begin{aligned} L_q &= \sum_{j=0}^J \sum_{k=0}^K \alpha_{k,j} p^k \sigma_p^k \left(\frac{1}{z} \frac{1-\sigma_q}{1-q}\right)^j = \sum_{j=0}^J \sum_{k=0}^K \alpha_{k,j} p^k \frac{(p; p)_{j-1}}{(1-q)^j} q^{kj} z^{-j} \sigma_p^k (1-\sigma_q)^j \\ &= \sum_{j=0}^J \sum_{k=0}^K \sum_{i=0}^j \alpha_{k,j} p^k \frac{(p; p)_{j-1}}{(1-q)^j} q^{kj} \binom{j}{i} (-1)^i z^{-j} \sigma_q^{i-k} \end{aligned}$$

$$\begin{aligned}
&= \sum_{j=0}^J \sum_{k=0}^K \sum_{i=-k}^{j-k} \alpha_{k,j} p^k \frac{(p;p)_{j-1}}{(1-q)^j} q^{kj} \binom{j}{i+k} (-1)^{i+k} z^{-j} \sigma_q^i \\
&= \sum_{i=-K}^J \sum_{j=\max(0,i)}^J \beta_{i,j} z^{-j} \sigma_q^i
\end{aligned}$$

avec

$$\beta_{i,j} = \sum_{k=\max(0,-i)}^{\min(K,j-i)} \alpha_{k,j} p^k \frac{(p;p)_{j-1}}{(1-q)^j} q^{kj} \binom{j}{i+k} (-1)^{i+k}$$

et avec la convention $(p;p)_{j-1} = 1$ si $j = 0$.

On établit aisément que L_q est un opérateur de Fredholm dans les espaces $\mathcal{C}[[z]]_{q,s}$. Les indices s'expriment à l'aide d'un polygone de Newton au sens q -différences [7]; on étend la définition aux termes de la forme $z^k \sigma_q^i$ avec k et $i \in \mathbb{Z}$. En raisonnant comme en 4.2.4 on définit un nouveau polygone de Newton :

Définition 7.2.3 : Soit $\Delta_q \in \mathcal{C}[q^x, \tau]$. Son q -Laplace polygone de Newton est par définition l'enveloppe convexe des points $(-k_j^+, -j)$ et $(j - k_j^-, -j)$ tels que $j = 0 \dots J$, $\Gamma_j = \{k = 0, \dots, K \mid \alpha_{k,j} \neq 0\} \neq \emptyset$, $k_j^- = \min \Gamma_j$ et $k_j^+ = \max \Gamma_j$.

Le théorème 7.2.2 s'établit en considérant $L_q \Gamma \hat{F}_q$ et \hat{G}_q les transformées de q -Laplace formelles respectives de $\Delta_q \Gamma \hat{f}_q$ et \hat{g}_q .

Références bibliographiques

- [1] A. BarkatouΓA. DuvalΓ*Sur les séries formelles solutions d'équations aux différences polynomiales*ΓAnnal. Inst. FourierΓ44 (1994)Γ495-524.
- [2] J.-P. BézivinΓ*Sur les équations fonctionnelles aux q -différences*. Preprint Paris VI (1990).
- [3] G. GasperΓM. RahmanΓ*Basic Hypergeometric Series*. Encyclopedia of Mathematics and its ApplicationsΓG.C. Rota ed.ΓVol. 35 (1990).
- [4] W. HahnΓ*Über die Funktional-differentialgleichung $f'(z) = f(qz)$ und verwandte Funktionalgleichungen*. Ann. Univ.Sc. BudapestΓEötvös Sect. Math. (1973)Γp.3-21.
- [5] B. MalgrangeΓ*Sur les points singuliers des équations différentielles*. L'Enseignement MathématiqueΓtome 20Γn°1-2 (1974)Γp. 147-176.
- [6] F. NaegleΓ*Théorèmes d'indices pour les équations q -différences-différentielles*ΓC.R. Acad. Sci. ParisΓt. 317ΓSerie IIΓp. 579-582Π1993.
- [7] J.-P. RamisΓ*Théorèmes d'indices Gevrey pour les équations différentielles ordinaires*. Memoirs of the American Mathematical Society n°296Γvolume 48 (1984).
- [8] J.-P. RamisΓ*About the growth of entire functions solutions of linear algebraic q -difference equations*. Annales de la Faculté des Sciences de Toulouse. Série 6ΓVol. IIh°1Γ1992.

Chapitre 3

Un algorithme de resommation de séries formelles solutions d'équations différentielles ordinaires linéaires

1. Introduction

Les solutions fondamentales formelles au voisinage de l'origine (la singularité est supposée être en $x = 0$) d'équations différentielles linéaires homogènes à coefficients polynomiaux sont de la forme

$$\exp(Q(1/x^{\frac{1}{q}}))x^{\frac{\nu}{q}}\hat{f}(x^{\frac{1}{q}})$$

où Q est un polynôme $Q \in \mathbb{C}[x]$ ν est un nombre algébrique et

$$\hat{f}(x) = \hat{f}_0(x) + \hat{f}_1(x)\log(x) + \cdots + \hat{f}_m(x)\log^m(x)$$

($\hat{f}_0(x), \dots, \hat{f}_m(x)$ sont des séries formelles). Toutes les informations relatives à ces solutions (en particulier les coefficients des séries formelles $\hat{f}_0(x), \dots, \hat{f}_m(x)$) sont données par le logiciel DESIR [89]. Les séries formelles obtenues sont en général divergentes dans le cas d'une singularité irrégulière. Nous nous proposons de calculer leur somme.

La théorie de la k -sommabilité et de la multisommabilité développée par J. Ecalle [B. Malgrange et J.-P. Ramis [10][11][12]] a permis une première mise en oeuvre d'algorithmes de calcul de la somme de séries divergentes. Des méthodes spécifiques pour le calcul des sommes des séries k -sommables ont été élaborées [16][7] ainsi qu'un algorithme de multisommation utilisant l'accélératrice d'Ecalle [18]. Ce dernier procédé repose actuellement sur la possibilité d'obtenir une évaluation simple de cette accélératrice dans certains cas.

W. Balser préconise un procédé mathématique explicite pour obtenir la somme d'une série formelle multisommable dans le cas général [12][4]. Il s'agit d'une méthode d'itération de transformées de Borel puis de transformées de Laplace. Mon travail consiste en collaboration avec J. Thomann à transformer ce procédé en un algorithme formel-numérique de calcul effectif. Après une première tentative fructueuse de faisabilité (avec le processeur AXIOM et le langage FORTRAN) [9] sur l'exemple exposé ci-après il s'agissait d'élaborer un instrument de travail souple et général basé sur des spécifications informatiques simples et efficaces.

Pour ce faire la participation au projet européen CATHODE (Computer Algebra Tools for Handling Ordinary Differential Equations) nous a permis de dégager les primitives informatiques nécessaires à la réalisation de l'algorithme de Balser en les intégrant aux autres primitives définies dans ce groupe de travail.

Ainsi nous utilisons d'une part les polynômes d'Ore implantés par M. Bronstein pour représenter une série formelle soit par une équation aux différences dont est solution la suite des coefficients générant cette série soit par une équation différentielle admettant cette série comme solution. Nous avons d'autre part besoin des primitives graphiques élaborées par C. Dicrescenzo et F. Jung dans le logiciel *Compas* et par F. Jung dans sa thèse [14, 15] pour pouvoir visualiser à tout moment les résultats changer de chemin d'intégration etc.

L'un des exemples sur lequel nous allons illustrer notre travail est le suivant. Considérons l'équation de Ramis-Sibuya [13]

$$Dy = 4x + 2x^2 + 10x^3 - 3x^4$$

où D est l'opérateur différentiel

$$D = x^5(2-x)\frac{d^2}{dx^2} + x^2(4+5x^2-2x^3)\frac{d}{dx} + 2(2-x+x^2).$$

La série formelle $\hat{f}(x) = \sum_{n \geq 0} a_n x^n$ où

$$\begin{cases} a_{2p} = (2p)! \\ a_{2p+1} = -(2p+1)! + (-1)^p p! \end{cases}$$

est formellement solution de l'équation différentielle.

2. Résultats théoriques

2.1. Notations

Nous reprenons les notations utilisées par W. Balser [1].

Un secteur de la surface de Riemann du Logarithme est un ensemble de la forme

$$S = S(d, \alpha, \rho) = \{x = re^{i\phi} \mid 0 < r < \rho, \ d - \frac{\alpha}{2} < \phi < d + \frac{\alpha}{2}\}$$

où $d \in \mathbb{R}$, $\alpha > 0$, $\rho > 0$ ou $\rho = +\infty$. Un secteur fermé est un ensemble de la forme

$$\bar{S} = \bar{S}(d, \alpha, \rho) = \{x = re^{i\phi} \mid 0 < r \leq \rho, \ d - \frac{\alpha}{2} \leq \phi \leq d + \frac{\alpha}{2}\}$$

où $d \in \mathbb{R}$, $\alpha \geq 0$, $\rho > 0$.

L'algèbre différentielle $A_k(S)$ ($k > 0$) est l'ensemble des fonctions f analytiques dans le secteur S et admettant une série formelle $\hat{f} = \sum_{n \geq 0} a_n x^n$ comme développement asymptotique d'ordre k c'est-à-dire :

$$\forall \bar{S}_1 \subset S, \exists C, K > 0, \forall N \in \mathbb{N}, \forall x \in \bar{S}_1, |r_f(x, N)| \leq CK^N \Gamma\left(1 + \frac{N}{k}\right)$$

où $r_f(x, N) = x^{-N}(f(x) - \sum_{n=0}^{N-1} a_n x^n)$.
Notation : $f(x) \cong_k \hat{f}(x)$ dans S .

2.2. La k -sommabilité et la multisommabilité

Nous ne donnons ici que les notions qui seront utiles par la suite. Pour davantage de renseignements se reporter par exemple à [3101112].

Définition 2.2.1 : Soient $k > 0$, $d \in \mathbb{R}$ et $\hat{f} \in \mathcal{C}[[x]]$. La série formelle \hat{f} est k -sommable dans la direction d s'il existe un secteur $S = S(d, \alpha, \rho)$ d'ouverture $\alpha > \frac{\pi}{k}$ et une fonction $f \in A_k(S)$ tels que $f(x) \cong_k \hat{f}(x)$ dans S . La fonction f est unique et est appelée la somme de \hat{f} .

L'ensemble des séries k -sommables dans la direction d est noté $\mathcal{C}\{x\}_{k,d}$.

Définition 2.2.2 : Soient $k_1 > k_2 > \dots > k_r > 0$ et $d \in \mathbb{R}$ une direction issue de l'origine. La série formelle \hat{f} est (k_1, \dots, k_r) -sommable dans la direction d si elle s'écrit sous la forme $\hat{f} = \hat{f}_1 + \dots + \hat{f}_r$ avec $\hat{f}_l \in \mathcal{C}\{x\}_{k_l, d}$ pour $l = 1, \dots, r$. La somme de \hat{f} , définie sur un secteur bissecté par d et d'ouverture $> \frac{\pi}{k_1}$, est alors égale à $f_1 + \dots + f_r$ où f_l est la somme de \hat{f}_l dans la direction d ($l = 1, \dots, r$).

Remarque : Les séries $\hat{f}_l(x)$ peuvent être éventuellement ramifiées (séries en $x^{\frac{1}{m}} \Gamma m \in \mathbb{N}^*$). Une série formelle ramifiée $\hat{f}(x^{\frac{1}{m}}) \in \mathcal{C}[[x^{\frac{1}{m}}]]$ est dite (k_1, \dots, k_r) -sommable dans la direction d si la série $\hat{f}(x) \in \mathcal{C}[[x]]$ est (mk_1, \dots, mk_r) -sommable dans la direction $\frac{d}{m}$.

Si une série est (k_1, \dots, k_r) -sommable dans la direction d alors elle l'est dans les directions $d + 2k\pi \Gamma k \in \mathbb{Z}$.

Une série formelle est (k_1, \dots, k_r) -sommable si elle est (k_1, \dots, k_r) -sommable dans toutes les directions sauf en un nombre fini de directions (modulo 2π) appelées directions singulières.

Théorème 2.2.3 (J.P. Ramis) : Soit $\hat{f} \in \mathcal{C}[[x]]$ une série formelle solution de l'équation différentielle linéaire ordinaire à coefficients polynomiaux $D(\hat{f}) = 0$. On suppose que le polygône de Newton de l'opérateur D admet r pentes strictement positives $k_1 > k_2 > \dots > k_r > 0$. Alors \hat{f} est (k_1, \dots, k_r) -sommable. De plus, la somme de \hat{f} dans une direction non singulière d est solution de l'équation différentielle $D(f) = 0$.

Remarque : Si le polygone de Newton n'a qu'une pente strictement positive k alors $\hat{f}(x)$ est k -sommable.

Exemple de Ramis-Sibuya : On se ramène à une équation différentielle homogène en dérivant $1/(4x + 2x^2 + 10x^3 - 3x^4)Dy = 1$. La série formelle $\hat{f}(x)$ est alors solution de l'équation $D_0 y = 0$

où D_0 est l'opérateur

$$D_0 = (8x^6 + 18x^8 - 16x^9 + 3x^{10})\frac{d^3}{dx^3} + (-16x^3 + \cdots + 12x^9)\frac{d^2}{dx^2} \\ + (16x + \cdots + 6x^8)\frac{d}{dx} + (-16 + \cdots + 12x^5).$$

Le polygone de Newton associé à l'opérateur D_0 est représenté sur la figure 1.

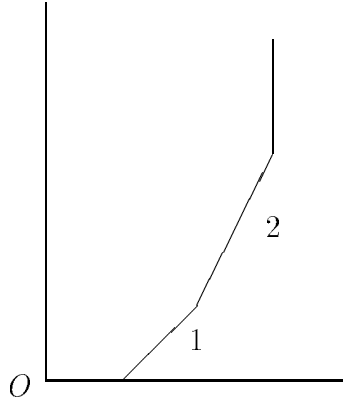


Figure 1 : Polygone de Newton associé à D_0

La série formelle $\hat{f}(x)$ est par conséquent $(2, 1)$ -sommable.

Remarque : Il n'est pas nécessaire de se ramener à une équation homogène pour connaître les pentes strictement positives du polygone de Newton et donc les réels définissant la multi-sommabilité. Ces pentes > 0 sont en effet les mêmes que celles du polygone de Newton associé à D car :

- Multiplier un opérateur par un polynôme de valuation $l \in \mathbb{N}$ a pour effet de translater le polygone de Newton : $(u, v) \in \mathbb{R}^2 \rightarrow (u, v + l)$;
- Diviser un opérateur par un polynôme de valuation $l \in \mathbb{N}$ a pour effet de translater le polygone de Newton : $(u, v) \in \mathbb{R}^2 \rightarrow (u, v - l)$;
- Dériver un opérateur $((a \frac{d^i}{dx^i} \rightarrow a'(x) \frac{d^i}{dx^i} + a(x) \frac{d^{i+1}}{dx^{i+1}})$ a pour effet de translater le polygone de Newton : $(u, v) \in \mathbb{R}^2 \rightarrow (u + 1, v - 1)$.

Lemme 2.2.4 : Soient $k > 0$ et $d \in \mathbb{R}$ une direction issue de l'origine. Soit $\hat{f}(x) = \sum_{n \geq 0} a_n x^n \in \mathcal{V}\{x\}_{k,d}$. On suppose que $a_0 = 0$. La série formelle $x^{-1}\hat{f}(x)$ est alors k -sommable dans la direction d .

Démonstration : Soient $S = S(d, \alpha, \rho)$ ($\alpha > \pi/k$) et $f \in A_k(S)$ tels que $f(x) \cong_k \hat{f}(x)$ dans S . Soit $\hat{g}(x) = x^{-1}\hat{f}(x) = \sum_{n \geq 0} a_{n+1} x^n$. On va montrer que $g(x) = x^{-1}f(x)$ admet $\hat{g}(x)$ comme développement asymptotique d'ordre k dans S .

Soit $\bar{S}_1 \subset S$: il existe $C, K > 0$ tels que pour tout $N \in \mathbb{N}$ et pour tout $x \in \bar{S}_1$ $|r_f(x, N)| \leq CK^N \Gamma(1 + \frac{N}{k})$. Or

$$r_g(x, N) = x^{-N} \left(g(x) - \sum_{n=0}^{N-1} a_{n+1} x^n \right) = r_f(x, N+1)$$

donc pour tout $\bar{S}_1 \subset S$ il existe $C, K > 0$ tels que pour tout $N \in \mathbb{N}$ et pour tout $x \in \bar{S}_1$:

$$|r_g(x, N)| \leq CK K^N \Gamma \left(1 + \frac{N}{k} \right) \frac{\Gamma(1 + \frac{N+1}{k})}{\Gamma(1 + \frac{N}{k})}$$

Or

$$\frac{\Gamma(1 + \frac{N+1}{k})}{\Gamma(1 + \frac{N}{k})} \sim \left(\frac{N}{k} \right)^{\frac{1}{k}}$$

et pour $N > 0$ $\Gamma(\frac{N}{k})^{\frac{1}{k}} \leq k^{-\frac{1}{k}} (\exp(\frac{1}{k}))^N$. Il existe donc deux constantes $\tilde{C}, \tilde{K} > 0$ (dépendantes de \bar{S}_1) telles que $\forall N \in \mathbb{N} \forall x \in \bar{S}_1$ $|r_g(x, N)| \leq \tilde{C} \tilde{K}^N \Gamma(1 + \frac{N}{k})$.

Lemme 2.2.5 : Soient $k > 0$ et $d \in \mathbb{R}$. Soit $\hat{f}(x) = \sum_{n \geq 0} a_n x^n \in \mathcal{C}\{x\}_{k,d}$. On suppose que $a_0 = a_1 = \dots = a_{q-1} = 0$ ($q \in \mathbb{N}^*$). La série formelle $x^{-q} \hat{f}(x)$ est alors k -sommable dans la direction d .

Démonstration : Pour $q = 1$ il s'agit du lemme précédent. Supposons que ce lemme soit vérifié jusqu'à l'entier q . Soit $\hat{f}(x) = \sum_{n \geq 0} a_n x^n \in \mathcal{C}\{x\}_{k,d}$ telle que $a_0 = a_1 = \dots = a_q = 0$. D'après l'hypothèse de récurrence la série formelle $\hat{g}(x) = x^{-q} \hat{f}(x) \in \mathcal{C}\{x\}_{k,d}$. Or le terme constant de $\hat{g}(x)$ est nul. En utilisant le lemme 2.2.4 la série $x^{-1} \hat{g}(x) = x^{-(q+1)} \hat{f}(x) \in \mathcal{C}\{x\}_{k,d}$.

Lemme 2.2.6 : Soient $k > 0$, $d \in \mathbb{R}$ et $q \in \mathbb{N}^*$. Soit $\hat{f}(x) = \sum_{n \geq 0} a_n x^{nq} \in \mathcal{C}\{x\}_{k,d}$. Alors la série $\hat{g}(x) = \sum_{n \geq 0} a_n x^n$ est $\frac{k}{q}$ -sommable dans la direction qd .

Démonstration : Soient $S = S(d, \alpha, \rho)$ ($\alpha > \pi/k$) et $f \in A_k(S)$ tels que $f(x) \cong_k \hat{f}(x)$ dans S . Considérons le secteur $\tilde{S} = \{x \in \mathcal{C} \mid x^{\frac{1}{q}} \in S\} = S(qd, q\alpha, \rho^q)$ d'ouverture $q\alpha > \frac{\pi}{k}$ ($\tilde{k} = \frac{k}{q}$). Soit $g(x) = f(x^{\frac{1}{q}})$: $g(x)$ est analytique dans \tilde{S} .

Soit $\tilde{S}_1 \subset \tilde{S}$. Soit $\bar{S}_1 = \{x \in \mathcal{C} \mid x^q \in \tilde{S}_1\}$. Il existe $C, K > 0$ tels que pour tout $N \in \mathbb{N}$ et pour tout $x \in \bar{S}_1$:

$$|r_g(x, N)| = |r_f(x^{\frac{1}{q}}, Nq)| \leq CK^{Nq} \Gamma \left(1 + \frac{Nq}{k} \right) = C(K^q)^N \Gamma \left(1 + \frac{N}{\tilde{k}} \right)$$

Lemme 2.2.7 : Soient $\alpha \in \mathbb{N}^*$ et $\hat{f}(x) = \sum_{n \geq 0} a_n x^n$ une série formelle k -sommable. Soient $q \in \mathbb{N}$ ($0 \leq q \leq \alpha - 1$) et $\hat{f}_q(x) = \sum_{n \geq 0} a_{\alpha n + q} x^n$. Soit $\theta = \frac{2\pi}{\alpha}$. Si $d, d + \theta, \dots, d + \theta(\alpha - 1)$ sont des directions non singulières pour $\hat{f}(x)$, alors les séries $\hat{f}_q(x)$ sont $\frac{k}{\alpha}$ -sommables dans la direction αd .

Démonstration : Soit $\hat{g}_q(x) = \sum_{l=0}^{\alpha-1} e^{-iq\theta l} \hat{f}(xe^{i\theta l}) = \sum_{n \geq 0} (\sum_{l=0}^{\alpha-1} e^{-i\theta l(n-q)}) a_n x^n$.

Soit $\alpha_n = \sum_{l=0}^{\alpha-1} e^{-i\theta l(n-q)} = \sum_{l=0}^{\alpha-1} (e^{-i\theta(n-q)})^l$. Si $e^{-i\theta(n-q)} = 1$ c'est-à-dire $n = q + m\alpha$ où $m \in \mathbb{N}$ alors $\alpha_n = \alpha$. Sinon $\alpha_n = \frac{1-e^{-i\theta(n-q)\alpha}}{1-e^{-i\theta(n-q)}} = 0$. Donc $\hat{g}_q(x) = \sum_{m \geq 0} \alpha a_{q+m\alpha} x^{q+m\alpha} = \alpha x^q \hat{f}_q(x^\alpha)$.

La série formelle $\hat{f}(x)$ est k -sommable dans la direction d . Etant également k -sommable dans la direction $d + \theta\Gamma$ la série $\hat{f}(xe^{i\theta})$ est également k -sommable dans la direction d . De même pour $l = 0, \dots, \alpha - 1$ les séries $\hat{f}(xe^{i\theta l})$ sont k -sommables dans la direction d . Donc $\hat{g}_q(x) \in \mathcal{C}\{x\}_{k,d}$. D'après le lemme 2.2.5 la série $x^{-q}\hat{g}(x)$ est k -sommable dans la direction d . Le lemme 2.2.6 permet alors de conclure.

2.3. Méthode des transformées de Laplace itérées (W. Balser)

Nous allons donner maintenant un procédé explicite qui va nous permettre de calculer la somme d'une série multisommable. Dans le cadre particulier de la k -sommabilité la définition 2.2.1 est équivalente à :

Définition 2.3.1 : Soient $k > 0$ et $d \in \mathbb{R}$. La série formelle $\hat{f}(x)$ est k -sommable dans la direction d si sa transformée de Borel formelle $\hat{\phi} = \hat{B}_k(\hat{f})$ d'ordre k (définie par $\hat{B}_k(x^\lambda)(t) = t^{\lambda-k}/\Gamma(\frac{\lambda}{k})$) est convergente et si sa somme se prolonge analytiquement en une fonction ϕ holomorphe et à croissance exponentielle d'ordre au plus k sur un secteur ouvert bissecté par d . Alors

$$f(x) = \int_d \phi(t) \exp(-(\frac{t}{x})^k) d(t^k)$$

est la somme de \hat{f} dans la direction d , au sens de la k -sommabilité.

Proposée par W. Balser [124] la méthode de calcul de la somme d'une série multisommable repose sur le principe des transformées de Laplace itérées (et de Borel itérées) et découle de la proposition suivante :

Proposition 2.3.2 : Soient $k_1 > \dots > k_r > 0$ ($r \geq 2$) et $d \in \mathbb{R}$. Soit $\hat{f}(x)$ une série formelle. Les conditions suivantes sont équivalentes :

- (i) La série formelle $\hat{f}(x)$ est (k_1, \dots, k_r) -sommable dans la direction d
- (ii) Soient $\tilde{k}_1, \dots, \tilde{k}_{r-1} > 0$ définis par $1/\tilde{k}_j = 1/k_{j+1} - 1/k_1$, $j = 1, \dots, r-1$. La série formelle $\hat{\phi} = \hat{B}_{k_1}(\hat{f})$ est $(\tilde{k}_1, \dots, \tilde{k}_{r-1})$ -sommable dans la direction d . De plus, la somme ϕ de $\hat{\phi}$ peut être prolongée analytiquement sur un petit secteur bissecté par d et de rayon infini sur lequel elle a une croissance exponentielle d'ordre au plus k_1 .

Par récurrence W. Balser en déduit le théorème :

Théorème 2.3.3 (W. Balser) : Soit \hat{f} une série (k_1, \dots, k_r) -sommable. Soit d une direction non singulière. Soient $\kappa_1, \dots, \kappa_r > 0$ définis par $1/\kappa_1 = 1/k_1$ et $1/\kappa_j =$

$1/k_j - 1/k_{j-1}$, $j = 2, \dots, r$. La somme de \hat{f} dans la direction d est égale à

$$L_{\kappa_1} \cdot_d \dots L_{\kappa_r} \cdot_d S \hat{B}_{\kappa_r} \dots \hat{B}_{\kappa_1} \hat{f}$$

où \cdot_d est le prolongement analytique le long de d , S est la somme de la série formelle convergente, \hat{B}_κ est la transformée de Borel formelle d'ordre κ ($\hat{B}_\kappa(x^\lambda)(t) = \frac{t^{\lambda-\kappa}}{\Gamma(\frac{\lambda}{\kappa})}$) et L_κ est la transformée de Laplace d'ordre κ ($L_\kappa(f)(x) = \int_d f(t) \exp(-(\frac{t}{x})^\kappa) d(t^\kappa)$).

Définition 2.3.4 : Soient $\kappa_1, \dots, \kappa_r > 0$. Soient $k_1 > \dots > k_r$ définis successivement par $1/k_1 = 1/\kappa_1$ et, si $r \geq 2$, par :

$$\frac{1}{k_l} = \frac{1}{\kappa_l} + \frac{1}{k_{l-1}}, \quad l = 2, \dots, r$$

Une série formelle est $(\kappa_1, \dots, \kappa_r) - iL$ sommable si elle est (k_1, \dots, k_r) -sommable.

La notation " iL " (pour "*iterated Laplace*") signifie que les réels $\kappa_1, \dots, \kappa_r$ sont les ordres des transformées de Laplace itérées.

Théorème 2.3.5 : Soient $\kappa_1, \dots, \kappa_r > 0$ ($r \geq 2$). Si $\hat{g}(x)$ est $(\kappa_1, \dots, \kappa_r) - iL$ sommable dans la direction $d \in \mathbb{R}$ alors la transformée de Borel formelle $\hat{B}_{\kappa_1}(\hat{g})$ est $(\kappa_2, \dots, \kappa_r) - iL$ sommable dans la direction d .

Démonstration : La série formelle $\hat{B}_{\kappa_1}(\hat{g})$ est $(\tilde{\kappa}_1, \dots, \tilde{\kappa}_{r-1})$ -sommable dans la direction d (proposition 2.3.2.) C'est-à-dire $(\tilde{\kappa}_1, \dots, \tilde{\kappa}_{r-1}) - iL$ sommable où $\tilde{\kappa}_j = \kappa_{j+1}$ ($j = 1, \dots, r-1$).

Pour calculer la somme d'une série formelle $\hat{f}(x)$ $(\kappa_1, \dots, \kappa_r) - iL$ sommable on procède donc de la façon suivante.

Posons $\hat{f}_1 = \hat{f}$ et $\hat{f}_{j+1} = \hat{B}_{\kappa_j}(\hat{f}_j)$ pour $j = 1, \dots, r-1$.

Chaque série formelle \hat{f}_{j+1} est $(\kappa_{j+1}, \dots, \kappa_r) - iL$ sommable (théorème 2.3.5). La série \hat{f}_r étant $(\kappa_r) - iL$ sommable c'est-à-dire κ_r -sommable sa transformée de Borel d'ordre κ_r $\hat{f}_{r+1} = \hat{B}_{\kappa_r}(\hat{f}_r)$ (dernier niveau) est convergente à l'origine et peut être prolongée analytiquement le long d'une direction non singulière d en une fonction holomorphe à croissance exponentielle d'ordre au plus κ_r sur un secteur bissecté par d (définition 2.3.1).

En itérant les transformées de Laplace

$$f_j(x) = L_{\kappa_j}(f_{j+1})(x) = \int_d f_{j+1}(t) \exp\left(-\left(\frac{t}{x}\right)^{\kappa_j}\right) d(t^{\kappa_j})$$

et en remontant dans les plans de Borel (c'est-à-dire en passant du niveau $j = r$ jusqu'au niveau $j = 1$) nous obtenons la somme $f_1(x) = f(x)$ de $\hat{f}(x)$ sur un secteur convenable bissecté par d .

L'algorithme formel calcule les transformées de Borel formelles d'ordre rationnel κ_j et donne une équation différentielle dont est solution chaque série formelle \hat{f}_j pour $j = 2, \dots, r+1$.

Le prolongement analytique de chaque fonction f_j est réalisé par des méthodes d'intégration numériques (méthode de Runge-Kutta par exemple) à partir de conditions initiales données par l'intégrale

$$\int_d f_{j+1}(t) \exp\left(-\left(\frac{t}{x}\right)^{\kappa_j}\right) d(t^{\kappa_j})$$

et calculées par des quadratures de Gauss-Laguerre (ce sont des conditions initiales numériques). La fonction f_{j+1} a été précédemment calculée par la même méthode l'exception faite du dernier niveau pour lequel les conditions initiales à partir desquelles est effectué le prolongement analytique se lisent directement à partir des coefficients de la série formelle convergente \hat{f}_{r+1} et sont donc exactes.

La figure 2 décrit le procédé des transformées de Laplace itérées (C.I.F.Γ resp. C.I.E.Γ signifie Conditions Initiales FlouesΓ resp. Conditions Initiales Exactes).

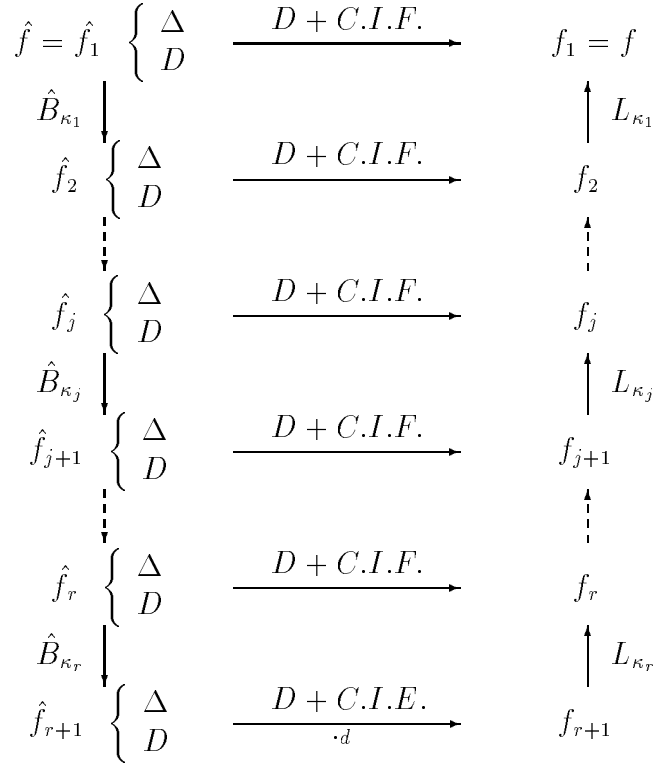


Figure 2 : Transformées de Laplace itérées

La précision doit être contrôlée à chaque niveau. En effetΓ les fonctions f_j sont solutions d'équations différentielles (les mêmes que celles dont sont solutions les séries formelles $\hat{f}_j(x)$)Γ déterminées *exactement* à partir d'algorithmes formels. Mais les conditions initiales sont calculées par des algorithmes numériques (sauf pour le dernier niveau).

2.4. Polynômes d'Ore

Nous avons besoin d'outils de bases pour implanter la partie formelle de l'algorithme de Balser afin de calculer des transformées de Borel de séries formelles des transformées de Mellin d'opérateurs aux différences pour obtenir les équations différentielles dont sont solutions les transformées de Borel à chaque niveau.

Plus généralement nous avons besoin d'outils permettant d'effectuer des opérations algébriques élémentaires des homographies des ramifications et des factorisations sur les opérateurs différentiels ou aux différences.

Les opérateurs différentiels et aux différences peuvent être interprétés comme des polynômes d'Ore non commutatifs. L'algèbre des polynômes d'Ore a été implantée en Axiom et $A^\#$ par M. Bronstein en Maple par T. Mulders pour un endomorphisme σ et une application $\delta : k \rightarrow k$ tels que $\delta(a + b) = \delta a + \delta b$ et $\delta(ab) = \sigma(a)\delta(b) + \delta(a)b$ pour tous $a, b \in k$ (k est un corps). Si x est une indéterminée sur k l'anneau des polynômes d'Ore est l'anneau noté $k[x; \sigma, \delta]$ des polynômes en x à coefficients dans k muni de l'addition usuelle; la multiplication est définie par $xa = \sigma(a)x + \delta a$ ($a \in k$) étendue aux monômes par

$$(ax^n)(bx^m) = (ax^{n-1})(\sigma(b)x^{m+1} + \delta bx^m) \quad (n > 0, \quad a, b \in k)$$

et aux polynômes par

$$\left(\sum_i a_i x^i\right)\left(\sum_j b_j x^j\right) = \sum_i \sum_j (ax^i)(b_j x^j).$$

Nous allons à présent voir comment on peut faire agir les polynômes d'Ore sur un corps k ou une extension de corps de k (pour de plus amples détails et pour les démonstrations on se reportera à [6]).

Soit V un espace vectoriel sur k .

Définition 2.4.1 : Une application $\theta : V \rightarrow V$ est dite k -pseudo linéaire relativement à σ et δ si : $\forall u, v \in V, a \in k$,

$$\begin{cases} \theta(u + v) = \theta u + \theta v \\ \theta(au) = \sigma(a)\theta u + \delta au \end{cases}$$

Lemme 2.4.2 : Toute application k -pseudo linéaire est $Const_{\sigma, \delta}(k)$ -linéaire, où $Const_{\sigma, \delta}(k) = \{a \in k \mid \sigma(a) = a, \delta a = 0\}$.

Soit θ une application k -pseudo linéaire. On peut faire agir les polynômes d'Ore sur V par l'application

$$\begin{aligned} *_{\theta} : k[x; \sigma, \delta] \times V &\rightarrow V \\ \left(\sum_{i=0}^n a_i x^i, u\right) &\mapsto \sum_{i=0}^n a_i \theta^i u \end{aligned}$$

Lemme 2.4.3 : Si K est une extension de corps de k compatible avec k , pour tout $c \in K$, l'application $\theta_c : K \rightarrow K$ définie par $\theta_c a = c\sigma(a) + \delta a$ est $Const_{\sigma, \delta}(K)$ -linéaire et

K -pseudo linéaire.

Si $c = 0$ $\Gamma\sigma = 1$ et si δ est un opérateur de dérivation $D\Gamma\theta_c a = Da$. Si $c = 1$ $\Gamma\delta = 0$ et si σ est l'opérateur de translation $\tau\Gamma\theta_c a = \tau a$.

Tout opérateur différentiel linéaire ordinaire dont les coefficients sont des fractions de polynômes en ξ et à coefficients dans un anneau R peut se mettre sous l'une des formes canoniques suivantes :

- $\sum_{i=0}^q a_i(\xi) \left(\frac{d}{d\xi}\right)^i$
- $\sum_{i=0}^q b_i(\xi) \theta^i$ où θ est l'opérateur d'Euler $\theta = \xi \frac{d}{d\xi}$
- $\sum_{i=0}^q \left(\frac{d}{d\xi}\right)^i \alpha_i(\xi)$
- $\sum_{i=0}^q \theta^i \beta_i(\xi)$

Les deux premiers opérateurs peuvent être considérés respectivement comme des éléments des anneaux d'Ore $k[x; 1, ']$ (resp. $k[x; 1, \theta]$) où $' = \frac{d}{d\xi}$. Les deux derniers opérateurs peuvent également être considérés comme des polynômes d'Ore mais dont les coefficients sont à droite.

Nous pouvons également définir des formes canoniques pour des opérateurs aux différences. Tout opérateur linéaire aux différences dont les coefficients sont des polynômes en j à coefficients dans un anneau R peut se mettre sous l'une des formes canoniques suivantes :

- $\sum_{i=0}^r \sum_{l=0}^q a_{i,l} j^l \tau^i$ où τ est l'opérateur de translation de pas $+1$
- $\sum_{i=0}^r \sum_{l=0}^q b_{i,l} j^l \delta^i$ où δ est l'opérateur $\tau - 1$
- $\sum_{i=0}^r \sum_{l=0}^q \alpha_{i,l} \tau^i j^l$
- $\sum_{i=0}^r \sum_{l=0}^q \beta_{i,l} \delta^i < j >_l$ où $< j >_l = j(j+1) \dots (j+l-1)$ $l > 0$ et $< j >_0 = 1$

La première forme est utilisée pour définir le τ -polygone de Newton la deuxième pour définir le δ -polygone de Newton [5]. La transformée de Mellin d'un opérateur écrit sous la troisième forme (resp. la quatrième forme) est l'opérateur différentiel $\sum_{i=0}^r \sum_{l=0}^q \alpha_{i,l} \xi^i \theta^l$ (resp. $\sum_{i=0}^r \sum_{l=0}^q (-1)^l \beta_{i,l} \xi^{i+l} \left(\frac{d}{d\xi}\right)^l$).

3. Outils formels

A chaque niveau de l'algorithme nous aurons besoin de scinder des séries formelles de connaître leur transformée de Borel formelle et de chercher des équations différentielles. Toutes ces opérations vont être effectuées par des algorithmes formels.

3.1. Définition du concept de série formelle

Une série formelle $\hat{f}(x) = \sum_{j \geq 0} a_j x^j$ est définie dès que les coefficients $(a_j)_{j \geq 0}$ le sont. Nous nous plaçons dans le cadre des équations aux différences finies; la suite des coefficients $(a_j)_{j \geq 0}$ est solution d'une équation aux différences finies à coefficients polynomiaux de la forme :

$$(\Delta) \begin{cases} P_0(j)a_j + \cdots + P_r(j)a_{j+r} = 0 & , \quad j \geq \mu \\ a_0, \dots, a_{m-1} & \text{donnés} \end{cases}$$

où $P_0(j), \dots, P_r(j)$ sont des polynômes en j à coefficients algébriques; $r \in \mathbb{N}$ est le degré de l'équation aux différences $\Gamma \mu \in \mathbb{N} \mid m = \mu + r$ et a_0, \dots, a_{m-1} sont les premiers termes algébriques de la suite (également appelés conditions initiales). Nous supposons de plus que $P_0(j)$ et $P_r(j)$ ne sont pas identiquement nuls.

Si $P_r(j)$ n'admet pas de zéro entier supérieur ou égal à μ chaque coefficient a_j peut être calculé exactement et l'équation (Δ) représente la série formelle $\hat{f}(x)$.

Première remarque : Cette définition nous permet de donner plus de conditions initiales que les r nécessaires. S'il existe $j_0 \in \mathbb{N}$ tel que $P_r(j_0) = 0$ le coefficient a_{j_0+r} ne peut pas être calculé à partir de l'équation aux différences. Il est donc intéressant de pouvoir donner davantage de premiers termes (les $j_0 + 1 + r$ premiers) afin de n'utiliser l'équation aux différences qu'à partir de $j \geq j_0 + 1$.

Seconde remarque : Une série formelle du type précédent ($\mu = m - r > 0$) peut aussi s'écrire

$$\begin{cases} 0.a_0 + \dots + 0.a_{j+\mu-1} + P_0(j+\mu)a_{j+\mu} + \cdots + P_r(j+\mu)a_{j+\mu+r} = 0, & j \geq 0 \\ a_0, \dots, a_{m-1} & \text{donnés} \end{cases}$$

Nous aurions donc également pu définir les séries formelles par des équations aux différences de la forme

$$(\Delta) \begin{cases} Q_0(j)a_j + \cdots + Q_r(j)a_{j+r} = 0 & , \quad j \geq 0 \\ a_0, \dots, a_{r-1} & \text{donnés} \end{cases}$$

Le coefficient $Q_0(j)$ peut dans ce cas être identiquement nul.

3.2. Passage d'une équation aux différences à une équation différentielle puis à un système différentiel

Pour calculer le prolongement analytique par une méthode de quadrature de type Runge-Kutta d'une fonction $f(x)$ le long d'une direction d nous avons besoin de connaître une équation différentielle dont est solution $f(x)$. Si $f(x)$ est la somme d'une série formelle multisummable $\hat{f}(x)$ dont nous savons qu'elle est solution formelle d'une équation différentielle alors $f(x)$ est également solution de cette équation. Il nous reste donc à déterminer une équation différentielle satisfaite par une série formelle définie par une équation aux différences.

Considérons une série formelle $\hat{f}(x)$ représentée par l'équation aux différences

$$(\Delta) \begin{cases} P_0(j)a_j + \dots + P_r(j)a_{j+r} = 0 & , \quad j \geq \mu \\ a_0, \dots, a_{m-1} & \text{donnés} \end{cases}$$

où $P_i(j) = \sum_{l=0}^p \alpha_{i,l} j^l$, $i = 0, \dots, r$; $\mu \in \mathbb{N}$ et $m = \mu + r$.

Soit D_t la transformée de Mellin formelle de l'opérateur aux différences :

$$D_t = \sum_{i=0}^r \sum_{l=0}^p \alpha_{i,l} \left(-t \frac{d}{dt}\right)^l t^i, \quad t = \frac{1}{x}$$

et $D_x = \sum_{i=0}^r \sum_{l=0}^p \alpha_{i,l} \left(x \frac{d}{dx}\right)^l x^{-i}$. On établit aisément que $D_x(\hat{f}) = \sum_{l=-r}^{\mu-1} p_l x^l$ Les coefficients p_l dépendants des conditions initiales a_0, \dots, a_{m-1} . En utilisant l'identité $\left(x \frac{d}{dx}\right)^l x^{-i} = x^{-i} (-i + x \frac{d}{dx})^l$ nous obtenons l'équation différentielle à coefficients polynomiaux suivante :

$$\sum_{i=0}^r \sum_{l=0}^p \alpha_{r-i,l} x^i \left(i - r + x \frac{d}{dx}\right)^l \hat{f} = \sum_{l=0}^{m-1} p_{l-r} x^l$$

La méthode de Runge-Kutta que nous utilisons pour effectuer le prolongement analytique intègre un système différentiel d'ordre 1 d'une variable réelle. Considérons l'équation différentielle

$$\sum_{l=0}^p P_l(x) y^{(l)}(x) = Q(x)$$

où $P_0(x), \dots, P_p(x), Q(x)$ sont des polynômes. Supposons que l'on veuille calculer $f(x)$ solution de cette équation différentielle le long de la courbe $\gamma(t)$.

Posons $y_1(x) = y(x), y_2(x) = y'(x), \dots, y_p(x) = y^{(p-1)}(x)$ (les dérivées étant prises par rapport à x) et $z_1(t) = y_1(\gamma(t)), \dots, z_p(t) = y_p(\gamma(t))$.

Pour $l = 1, \dots, p-1$

$$\frac{d}{dt}(z_l(t)) = \gamma'(t) z_{l+1}(t)$$

et

$$\frac{d}{dt}(z_p(t)) = \gamma'(t) \frac{Q(\gamma(t)) - \sum_{l=0}^{p-1} P_l(\gamma(t)) z_{l+1}(t)}{P_p(\gamma(t))}.$$

Le long de la courbe $\gamma(t)$ la fonction $f(x)$ vérifie le système différentiel d'ordre 1 en la variable t suivant :

$$\frac{d}{dt} \begin{pmatrix} z_1(t) \\ z_2(t) \\ \dots \\ \dots \\ z_{p-1}(t) \\ z_p(t) \end{pmatrix} = \gamma'(t) \cdot \left[\begin{pmatrix} 0 & 1 & 0 & 0 & \dots & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{P_0(\gamma(t))}{P_p(\gamma(t))} & \dots & \dots & \dots & \dots & -\frac{P_{p-1}(\gamma(t))}{P_p(\gamma(t))} \end{pmatrix} \begin{pmatrix} z_1(t) \\ z_2(t) \\ \dots \\ \dots \\ z_{p-1}(t) \\ z_p(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \dots \\ \dots \\ 0 \\ \frac{Q(\gamma(t))}{P_p(\gamma(t))} \end{pmatrix} \right]$$

3.3. Scindage d'une série formelle

La transformée de Borel formelle de niveau κ d'une série formelle $\hat{f}(x) = \sum_{j \geq 0} a_j x^j \Gamma$ représentée par l'équation aux différences

$$(\Delta) \begin{cases} P_0(j)a_j + \cdots + P_r(j)a_{j+r} = 0 & , \quad j \geq \mu \\ a_0, \dots, a_{m-1} & \text{donnés} \end{cases}$$

est définie par $\hat{\phi}(\xi) = \sum \frac{a_j}{\Gamma(j/\kappa)} x^{j-\kappa}$. La présence de la fonction Γ nous empêche de pouvoir définir la suite $(a_j/\Gamma(j/\kappa))$ par une équation aux différences polynomiale. Nous sommes ainsi amenés à scinder préalablement la série formelle $\hat{f}(x)$.

Soit $\alpha \in \mathbb{N}$, $\alpha \neq 0$. Soient $q \in \mathbb{N}$, $0 \leq q \leq \alpha - 1$ et $\hat{f}_q(x) = \sum_{j \geq 1} a_{j\alpha+q} x^j$ (q -ième sous-série associée à $\hat{f}(x)$) :

$$\hat{f}(x) = (a_0 + \hat{f}_0(x^\alpha)) + \cdots + x^{\alpha-1} (a_{\alpha-1} + \hat{f}_{\alpha-1}(x^\alpha))$$

Comme nous allons le voir, chaque sous-série $\hat{f}_q(x)$ peut être définie par une équation aux différences finies à coefficients polynomiaux (Δ_q) d'ordre $r_q \leq r$.

Si $\hat{f}(x)$ était représentée par une équation aux différences de la forme

$$Q_0(j)a_j + Q_1(j)a_{j+\alpha} + \cdots + Q_{\bar{r}}(j)a_{j+\bar{r}\alpha} = 0$$

alors chaque équation aux différences

$$Q_0(j\alpha + q)a_{j\alpha+q} + Q_1(j\alpha + q)a_{(j+1)\alpha+q} + \cdots + Q_{\bar{r}}(j\alpha + q)a_{(j+\bar{r})\alpha+q} = 0$$

définirait la q -ième sous-série (les conditions initiales étant calculées à partir des premiers termes de $\hat{f}(x)$ et de son équation aux différences associée). Il nous reste donc à donner un algorithme permettant de passer formellement d'une équation aux différences de la forme $P_0(j)a_j + \cdots + P_r(j)a_{j+r} = 0$ à une équation donnant *tous les α termes*.

Considérons le système (S_{init}) de $r(\alpha - 1) + 1$ équations :

$$((P_0(j+i)a_{j+i} + P_1(j+i)a_{j+i+1} + \cdots + P_r(j+i)a_{j+i+r} = 0)_{i=0, \dots, r(\alpha-1)})$$

Les inconnues de ce système sont les termes a_{j+k} tels que α ne divise pas k . C'est-à-dire les termes $a_{j+1}, \dots, a_{j+\alpha-1} \Gamma, a_{j+\alpha+1}, \dots, a_{j+2\alpha-1} \Gamma, \dots, a_{j+(r-1)\alpha+1} \Gamma, \dots, a_{j+r\alpha-1} \Gamma$. Il y en a $r(\alpha - 1)$.

Soit $(S) = (S_{init})$. Notons $card(S)$ le nombre d'équations de S et S_i la i -ième équation de (S) ($i = 1, \dots, card(S)$).

Algorithme :

$(S) := (S_{init});$

Tant qu'on n'a pas fini faire :

- Considérer la première équation S_1 du système (S) ;
- Chercher dans cette équation le plus petit indice k non multiple de α tel que le coefficient de a_{j+k} ne soit pas identiquement nul;
- Si un tel k n'existe pas On a fini (il n'y a plus d'inconnues).
Sinon :
 - Construire un nouveau système (S_{aux}) à $\text{card}(S) - 1$ équations en posant $(S_{aux})_{l-1} := C(1, k)S_l - C(l, k)S_1$ où $C(1, k)$ est le facteur de a_{j+k} dans S_1 et $C(l, k)$ celui dans S_l ($l = 2, \dots, \text{card}(S)$). On a éliminé l'inconnue a_{j+k} ;
 - $(S) := (S_{aux})$

L'équation S_1 ne comporte plus d'inconnues et est de la forme $Q_0(j)a_j + Q_1(j)a_{j+\alpha} + \dots + Q_{\bar{r}}(j)a_{j+\bar{r}\alpha} = 0$.

Première remarque : A chaque étape le nombre d'équations considérées ainsi que le nombre d'inconnues diminue d'une unité. L'algorithme se termine en au plus $r(\alpha - 1)$ étapes.

Seconde remarque : On définit le degré d'une équation par l'indice k le plus grand tel que le coefficient de a_{j+k} soit non nul. Initialement les équations de (S) sont de degré respectivement égal à $r, r+1, \dots, r\alpha$. On montre facilement par récurrence qu'à l'étape i ($i = 0, \dots, r(\alpha - 1) - 1$) les équations sont de degré $r+i, r+i+1, \dots, r\alpha$. L'équation S_1 n'est donc pas égale à $0 = 0$.

Les sous-séries obtenues sont également multisommables :

Théorème 3.3.1 : Supposons que $\hat{f}(x)$ est $(\kappa_1, \dots, \kappa_r) - iL$ sommable. Soient $\alpha \in \mathbb{N}^*$ et $\hat{f}_0(x), \dots, \hat{f}_{\alpha-1}(x)$ les α sous-séries.

Alors chaque sous-série $\hat{f}_q(x)$ est $(\kappa_1/\alpha, \dots, \kappa_r/\alpha) - iL$ sommable. De plus, si $d \in \mathbb{R}$ est tel que $d, d+\theta, \dots, d+(\alpha-1)\theta$ sont des directions non singulières pour $\hat{f}(x)$ ($\theta = 2\pi/\alpha$), alors αd est une direction non singulière pour les sous-séries. Dans un secteur bissecté par d , la somme $f(x)$ de $\hat{f}(x)$ est égale à

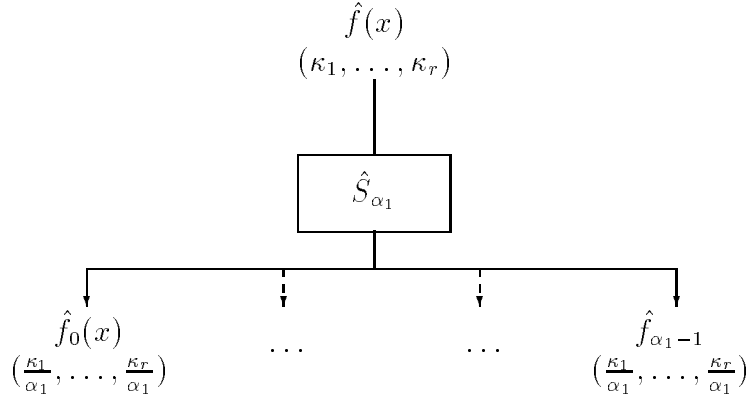
$$f(x) = (a_0 + f_0(x^\alpha)) + x(a_1 + f_1(x^\alpha)) + \dots + x^{\alpha-1}(a_{\alpha-1} + f_{\alpha-1}(x^\alpha))$$

Démonstration : On décompose la série $\hat{f}(x)$ en une somme de séries $\hat{f}_l(x)$ k_l -sommables ($k_1 = \kappa_1$ et $1/k_l = 1/\kappa_l + 1/k_{l-1}$ $l = 2, \dots, r$). La q -ième sous-série associée à $\hat{f}(x)$ est égale à la somme sur l des q -ièmes sous-séries associées à chaque $\hat{f}_l(x)$. On conclut en utilisant le lemme 2.2.7 pour chaque série $\hat{f}_l(x)$ (le lemme 2.2.7 est également vérifié si les séries $\hat{f}_l(x)$ sont ramifiées).

Supposons que les directions singulières de $\hat{f}(x)$ sont données par $d_i + 2\pi m$ $d_i \in [0, 2\pi[$ $i = 1, \dots, N$ et $m \in \mathbb{Z}$. Les directions singulières pour les sous-séries sont alors données par $\alpha d_i + 2\pi m$ ($i = 1, \dots, N$ et $m \in \mathbb{Z}$).

Nous n'appliquons donc pas l'algorithme de Balser sur la série formelle $\hat{f}(x)$ elle-même mais sur chacune des sous-séries. Si κ_1 est égal à $\frac{\alpha_1}{\beta_1}$ (α_1 et β_1 étant premiers entre eux) nous scindons $\hat{f}(x)$ en α_1 sous-séries. Chaque sous-série est alors $(\kappa_1/\alpha_1 = 1/\beta_1, \dots, \kappa_r/\alpha_1) - iL$ sommable.

Le schéma ci-dessous décrit le processus de scindage formel \hat{S}_{α_1} appliqué à une série formelle $\hat{f}(x)$ $(\kappa_1, \dots, \kappa_r) - iL$ sommable ($\alpha_1/\kappa_1 \in \mathbb{N}$).



3.4. Transformée de Borel formelle

Ainsi que nous l'avons vu précédemment nous n'avons besoin de connaître les transformées de Borel formelles que d'ordre $\frac{1}{\beta}$ où β est un entier strictement positif.

Soit $\hat{g}(x) = \sum_{j \geq 1} b_j x^j$ une série formelle représentée par une équation aux différences polynomiale (Δ) . La transformée de Borel formelle d'ordre $\frac{1}{\beta}$ de $\hat{g}(x)$ est la série formelle

$$\hat{B}_{1/\beta}(\hat{g})(x) = \sum_{j \geq 1} \frac{b_j}{\Gamma(j\beta)} x^{j-1/\beta} = x^{1-1/\beta} \sum_{j \geq 0} \frac{b_{j+1}}{(j\beta + \beta - 1)!} x^j$$

Il est aisé de voir que la série formelle $\hat{\phi}(x) = \sum_{j \geq 0} \frac{b_{j+1}}{(j\beta + \beta - 1)!} x^j$ (notée $\tilde{B}_{1/\beta}(\hat{g})$) peut être définie par une équation aux différences finies à coefficients polynomiaux $(\tilde{B}_{1/\beta}(\Delta))$.

Théorème 3.4.1 : Soient $\kappa_1, \dots, \kappa_r > 0$ avec $\kappa_1 = 1/\beta_1$, $\beta_1 \in \mathbb{N}$.

Si $\hat{g}(x)$ est $(\kappa_1, \dots, \kappa_r) - iL$ sommable dans la direction d alors la transformée de Borel formelle $\hat{B}_{1/\beta_1}(\hat{g})$ et $\tilde{B}_{1/\beta_1}(\hat{g})$ sont $(\kappa_2, \dots, \kappa_r) - iL$ sommables dans la direction d .

Démonstration : La série $\hat{B}_{1/\beta_1}(\hat{g})$ est $(\kappa_2, \dots, \kappa_r) - iL$ sommable dans la direction d (théorème 2.3.5). Par suite $\tilde{B}_{1/\beta_1}(\hat{g})$ l'est aussi.

Le schéma suivant décrit la transformation de Borel formelle appliquée à une série formelle $(\kappa_1, \dots, \kappa_r) - iL$ sommable ($\beta_1 \kappa_1 = 1$).

$$\begin{array}{ccc} \hat{g}(x) & & \hat{\phi}(x) \\ (\kappa_1, \dots, \kappa_r) & \xrightarrow{\quad \tilde{B}_{1/\beta_1} \quad} & (\kappa_2, \dots, \kappa_r) \end{array}$$

Remarque : De façon générale le degré des coefficients de l'équation aux différences définissant $(\tilde{B}_{1/\beta}(\Delta))$ augmente. En d'autres termes l'équation différentielle associée (cf 3.2.) est d'ordre plus élevé.

3.5. Transformées de Borel itérées

Soit $\hat{f}(x)$ une série formelle solution d'une équation différentielle à coefficients polynomiaux dont le polygone de Newton a deux pentes $k_1 > k_2 > 0$. Soient $\kappa_1 = k_1$ et $\kappa_2 = \frac{k_1 k_2}{k_1 - k_2}$. La série formelle $\hat{f}(x)$ est $(\kappa_1, \kappa_2) - iL$ sommable. Supposons que les directions singulières de $\hat{f}(x)$ soient données par $d_i + 2\pi l \Gamma d_i \in [0, 2\pi[\Gamma i = 1, \dots, N$ et $l \in \mathbb{Z}$.

Considérons les entiers strictement positifs $\alpha_1, \beta_1, \alpha_2, \beta_2$ (α_1 et β_1 sont premiers entre eux Γ α_2 et β_2 sont premiers entre eux) définis par $\kappa_1 = \frac{\alpha_1}{\beta_1}$ et $\frac{\kappa_2}{\alpha_1} = \frac{\alpha_2}{\beta_2}$. Soient $\theta_1 = \frac{2\pi}{\alpha_1}$ et $\theta_2 = \frac{2\pi}{\alpha_2}$.

Soit $d \in \mathbb{R}$ une direction issue de l'origine Γ telle que

$$\forall i = 1 \dots N, \quad \forall l \in \mathbb{Z}, \quad d \neq d_i + \frac{2\pi}{\alpha_1 \alpha_2} l.$$

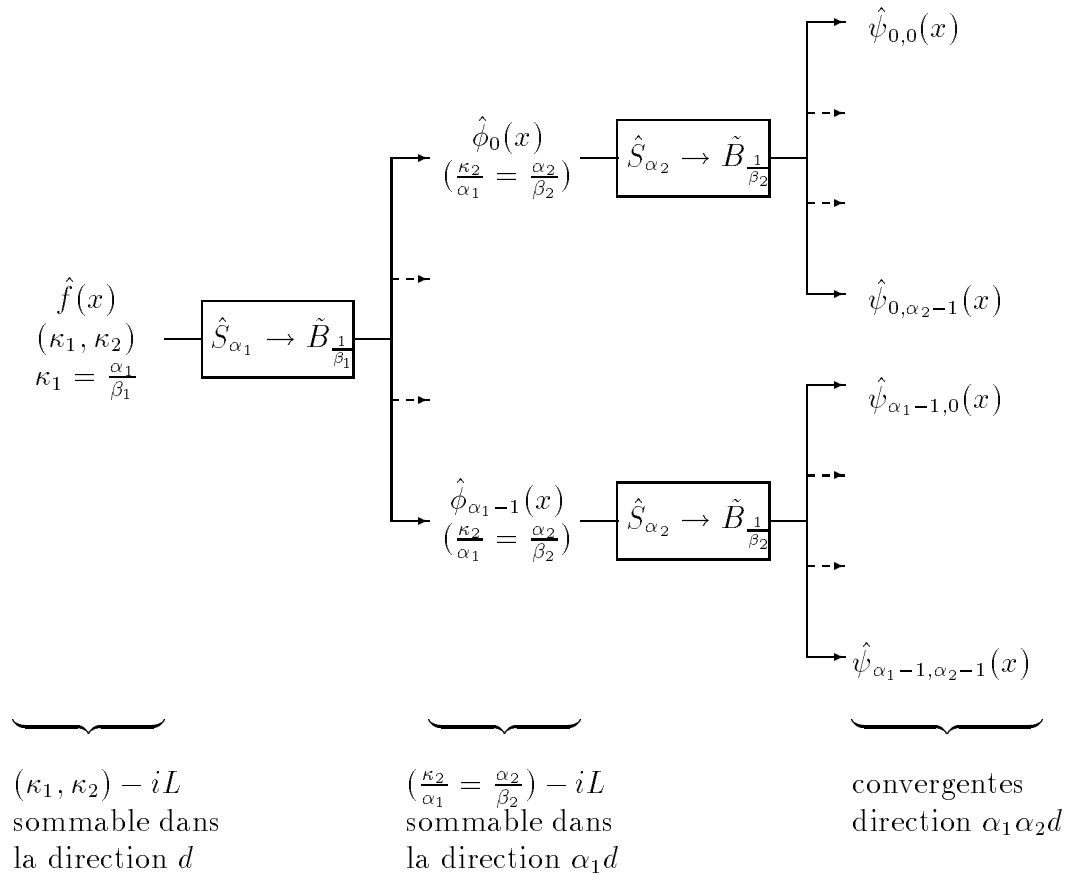
La série formelle $\hat{f}(x)$ est $(\kappa_1, \kappa_2) - iL$ sommable dans cette direction d .

La première étape de l'algorithme consiste à scinder la série formelle initiale en α_1 sous-séries $\hat{f}_0(x), \dots, \hat{f}_{\alpha_1-1}(x)$ et à calculer les transformées de Borel formelles $\tilde{B}_{\frac{1}{\beta_1}}(\hat{f}_q) = \hat{\phi}_q(x)$, $q = 0, \dots, \alpha_1 - 1$. Comme $d \neq d_i + \theta_1 l$ ($\forall l \in \mathbb{Z}$) Γ chaque série formelle $\hat{\phi}_q(x)$ est $(\frac{\kappa_2}{\alpha_1}) - iL$ sommable dans la direction $\alpha_1 d$.

La seconde étape consiste à scinder chaque série formelle $\hat{\phi}_q(x)$ en α_2 sous-séries puis à en prendre les transformées $\tilde{B}_{\frac{1}{\beta_2}}$. Comme $\alpha_1 d \neq \alpha_1 d_i + \theta_2 l$, $\forall l \in \mathbb{Z}$ Γ les sous-séries formelles sont $(\frac{1}{\beta_2}) - iL$ sommables Γ et les transformées de Borel d'ordre $\frac{1}{\beta_2}$ sont convergentes et peuvent être prolongées analytiquement le long de la direction $\alpha_1 \alpha_2 d$.

Ces deux étapes sont schématisées sur la figure 3.

Première remarque : A chaque niveau (ou étape) nous sommes amenés à calculer l'équation différentielle associée à chaque transformée de Borel formelle.



L'opérateur $\hat{S}_\alpha \rightarrow \tilde{B}_{\frac{1}{\beta}}$ appliqué à une série formelle consiste à la scinder en α sous-séries puis à calculer la transformée de Borel formelle de chaque sous-série :

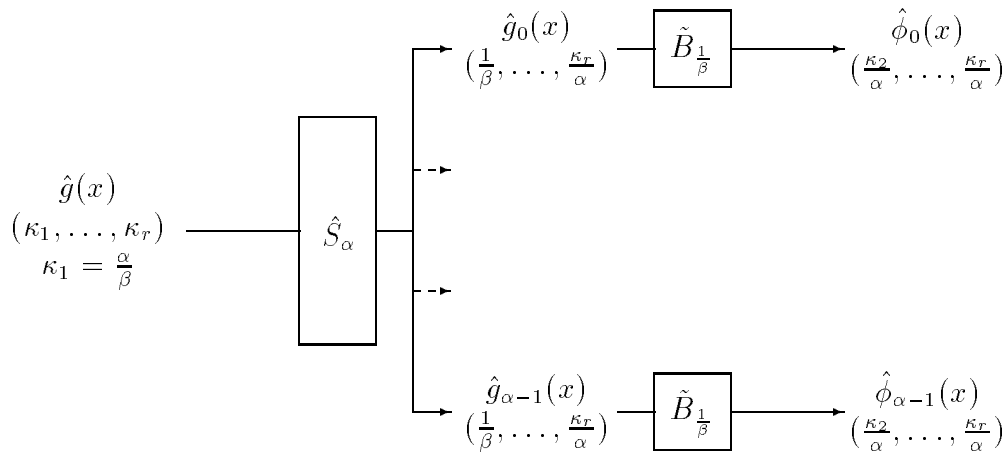


Figure 3 : Transformées de Borel itérées

Seconde remarque : Il est possible de voir à chaque niveau si les équations différentielles peuvent être factorisées ou s'il existe des solutions connues (sous forme exacte).

Troisième remarque : L'algorithme se généralise aisément à un nombre quelconque de niveaux. Mais la complexité de l'algorithme augmente très vite en pratique : le nombre de séries formelles à considérer augmente et en général Γ plus il y a de niveaux Γ plus le degré des équations différentielles augmente.

3.6. L'exemple de Ramis-Sibuya

La série formelle $\hat{f}(x) = x + 0.x^2 + 2x^3 - 7x^4 + 24x^5 - 118x^6 + 720x^7 + \dots$ peut être représentée par l'équation aux différences

$$(\Delta) \begin{cases} -(j^2 + j)a_j + (2j^2 + 7j + 5)a_{j+1} + 2a_{j+2} + (4j + 10)a_{j+3} + 4a_{j+4} = 0 \\ a_0 = 0, a_1 = 1, a_2 = 0, a_3 = 2, a_4 = -7 \end{cases}$$

La série $\hat{f}(x)$ est $(2, 1)$ -sommable Γ soit $(2, 2) - iL$ sommable. La partie formelle de l'algorithme consiste tout d'abord à scinder $\hat{f}(x)$ en deux sous-séries $\hat{f}_0(x)$ et $\hat{f}_1(x)$ lesquelles sont $(1, 1) - iL$ sommables Γ puis à calculer les transformées de Borel $\hat{\phi}_0 = \tilde{B}_1(\hat{f}_0)$ et $\hat{\phi}_1 = \tilde{B}_1(\hat{f}_1)$ qui sont $(1) - iL$ sommables. On calcule finalement les transformées de Borel $\hat{\psi}_0 = \tilde{B}_1(\hat{\phi}_0)$ et $\hat{\psi}_1 = \tilde{B}_1(\hat{\phi}_1)$.

Les sous-séries de $\hat{f}(x)$ sont $\hat{f}_0(x) = \sum_{j \geq 0} (-(2j+1)! + (-1)^j j!) x^{j+1}$ et $\hat{f}_1(x) = \sum_{j \geq 0} (2j)! x^j$. Les transformées de Borel $\hat{\phi}_0(x) = \sum_{j \geq 0} (-(2j+1)! + (-1)^j j!)/j! x^j$ et $\hat{\phi}_1(x) = \sum_{j \geq 0} (2j+2)!/j! x^j$ sont respectivement définies par :

$$\begin{cases} (4j+7)a_{j+2} - (16j^2 + 64j + 59)a_{j+1} - (16j^2 + 68j + 66)a_j = 0, & j \geq 0 \\ a_0 = 0, a_1 = -7 \end{cases}$$

$$\begin{cases} (j+1)a_{j+1} - 2(j+2)(2j+3)a_j = 0, & j \geq 0 \\ a_0 = 2 \end{cases}$$

Les équations différentielles associées sont respectivement égales à

$$\begin{aligned} & (-16x^4 - 16x^3)y^{(2)} + (-84x^3 - 48x^2 + 4x)y' + (-66x^2 - 11x - 1)y = -21x \\ \text{et} \quad & -4x^2y^{(2)} + (-18x + 1)y' - 12y = 0 \end{aligned}$$

La première équation admet comme singularités 0 et -1 .

Les transformées de Borel $\hat{\psi}_0(x)$ et $\hat{\psi}_1(x)$ sont respectivement définies par :

$$\begin{cases} (4j^3 + 23j^2 + 41j + 22)a_{j+2} - (16j^2 + 96j + 139)(j+1)a_{j+1} \\ \quad - (16j^2 + 100j + 150)a_j = 0, & j \geq 0 \\ a_0 = -7, a_1 = -59 \end{cases}$$

$$\begin{cases} (j+1)(j+2)a_{j+1} - 2(j+3)(2j+5)a_j = 0, & j \geq 0 \\ a_0 = 24 \end{cases}$$

et sont solutions respectivement des équations différentielles

$$\begin{aligned} & (-16x^2 + 4x)y^{(3)} + (-16x^2 - 112x + 11)y^{(2)} + (-116x - 139)y' - 150y = 0 \\ \text{et} \quad & (-4x^2 + x)y^{(2)} + (-26x + 2)y' - 30y = 0 \end{aligned}$$

Dans les deux cas $\Gamma x = 1/4$ est une singularité.

Les directions $\frac{\pi}{2}/\Gamma l \in \mathbb{Z}\Gamma$ constituent donc les directions singulières (éventuelles) pour $\hat{f}(x)$.

4. Prolongement analytique

4.1. Transformées de Laplace itérées

Soit $\hat{f}(x) = \sum_{n \geq 0} a_n x^n$ une série formelle $(\kappa_1, \dots, \kappa_r) - iL$ sommable. Supposons que les directions singulières de $\hat{f}(x)$ sont données par $d_i + 2\pi l \Gamma d_i \in [0, 2\pi[\Gamma i = 1, \dots, N$ et $l \in \mathbb{Z}$.

Soient $\alpha_1, \dots, \alpha_r \in \mathbb{N}$ et $\beta_1, \dots, \beta_r \in \mathbb{N}$ définis par $\kappa_1 = \frac{\alpha_1}{\beta_1}$ et $\frac{\kappa_j}{\alpha_1 \dots \alpha_{j-1}} = \frac{\alpha_j}{\beta_j} \Gamma j = 2, \dots, r$.

Soit $d \in \mathbb{R}$ une direction issue de l'origine Γ telle que

$$\forall i = 1 \dots N, \quad \forall l \in \mathbb{Z}, \quad d \neq d_i + \frac{2\pi}{\alpha_1 \alpha_2 \dots \alpha_r} l.$$

Soient $\hat{f}_0(x), \dots, \hat{f}_{\alpha_1-1}(x)$ les α_1 sous-séries formelles de $\hat{f}(x)$. Pour $q = 0, \dots, \alpha_1 - 1 \Gamma$ soit $\hat{\phi}_q(x) = \tilde{B}_{\frac{1}{\beta_1}}(\hat{f}_q)$ (cf 3). Soit $\phi_q(x)$ le prolongement analytique le long de la direction $\alpha_1 d$ de la somme de la série $\hat{\phi}_q(x)$. Sur un secteur convenable bissecté par d la somme $f(x)$ est égale à

$$f(x) = \left(a_0 + L_{\frac{1}{\beta_1}, \alpha_1 d}(\phi_0)(x^{\alpha_1}) \right) + \dots + x^{\alpha_1-1} \left(a_{\alpha_1-1} + L_{\frac{1}{\beta_1}, \alpha_1 d}(\phi_{\alpha_1-1})(x^{\alpha_1}) \right)$$

où

$$L_{\frac{1}{\beta_1}, \alpha_1 d}(\phi)(x) = \int_{\alpha_1 d} t^{1-\frac{1}{\beta_1}} \phi(t) \exp\left(-\left(\frac{t}{x}\right)^{\frac{1}{\beta_1}}\right) d\left(t^{\frac{1}{\beta_1}}\right) = \frac{1}{\beta_1} \int_{\alpha_1 d} \phi(t) \exp\left(-\left(\frac{t}{x}\right)^{\frac{1}{\beta_1}}\right) dt$$

Pour calculer la somme de la série $\hat{\phi}_q$ qui est $(\frac{\kappa_2}{\alpha_1}, \dots, \frac{\kappa_r}{\alpha_1}) - iL$ sommable Γ nous procédons de la même façon que précédemment Γ jusqu'à avoir des séries formelles convergentes.

Nous devons donc calculer à chaque niveau j ($j = 1, \dots, r$) des transformées de Laplace de la forme

$$\phi^j(x) = L_{\frac{1}{\beta_j}, \alpha_1 \dots \alpha_j d}(\phi^{j+1})(x).$$

Si nous avons calculé $\phi^{j+1}(t)$ avec une précision donnée pour chaque valeur de t dans la direction $\alpha_1 \dots \alpha_j d \Gamma$ nous pouvons espérer avoir une bonne approximation de $\phi^j(x)$ avec les méthodes classiques de Gauss-Legendre ou Gauss-Laguerre utilisées sur des sous-segments bien choisis de

la droite D_j de direction $\alpha_1 \dots \alpha_j d$ (si l'on s'aperçoit que pour t proche de l'origine la fonction $\phi^{j+1}(t)$ varie beaucoup on a intérêt à diviser la droite près de l'origine). Malheureusement si la droite D_j passe au voisinage d'une singularité de $\phi^{j+1}(t)$ l'évaluation numérique de $\phi^{j+1}(t)$ sera mauvaise. On a alors intérêt à déformer la droite D_j pour obtenir un chemin Π_j constitué de segments et d'arcs passant le plus loin possible des singularités de $\phi^{j+1}(t)$ (figure 4).

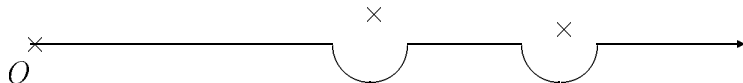


Figure 4 : Chemins d'intégrations Π_j

4.2. Localisation des singularités

Les racines de l'équation caractéristique de l'équation différentielle dont est solution $\phi^{j+1}(t)$ fournissent les singularités (éventuelles) de $\phi^{j+1}(t)$. L'équation différentielle est obtenue à partir de la série formelle initiale par des transformées de Borel et de Mellin et est calculée dans la partie formelle. Ces outils formels donnent donc les directions singulières éventuelles et nous permettent d'optimiser les chemins d'intégration Π_j des transformées de Laplace.

4.3. Prolongement analytique

Le prolongement analytique le long d'un chemin Π_j constitue le problème numérique essentiel de l'algorithme de Balser. A l'exception du dernier niveau où les séries sont convergentes l'origine est en général un point singulier irrégulier de l'équation différentielle dont est solution $\phi^{j+1}(t)$. Il est donc impossible de calculer $\phi^{j+1}(t)$ au voisinage de l'origine autrement que par la méthode des transformées de Laplace itérées elle-même. Mais pour calculer $\phi^{j+1}(t)$ aux abscisses définies par les formules de quadrature nous devons utiliser une méthode numérique efficace pour effectuer le prolongement analytique.

Le prolongement analytique va ainsi se faire en deux étapes. Soit P_j un point régulier du chemin Π_j ni trop loin ni trop près de l'origine excepté pour le dernier niveau où P_r est l'origine O . On calcule d'une part $\phi^{j+1}(t)$ aux abscisses de quadrature situées entre l'origine O et le point P_j en appelant récursivement l'algorithme de Balser. D'autre part le calcul de $\phi^{j+1}(t)$ aux abscisses de quadrature situées au-delà de P_j se fait par une méthode de prolongement analytique telle que la méthode de Runge-Kutta en utilisant l'équation différentielle et les conditions initiales au point P_j obtenues préalablement par l'algorithme de Balser lui-même (figure 5).

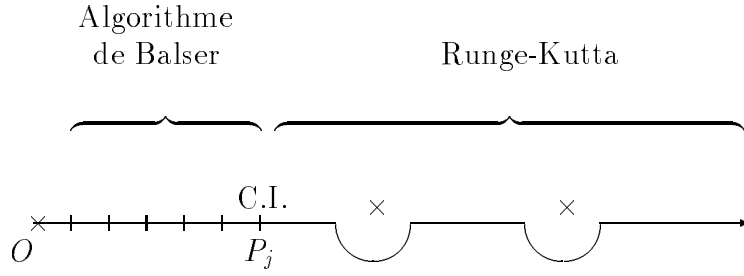


Figure 5 : Prolongement analytique

A chaque niveau le prolongement analytique se fait donc de la même manière : de l'origine au point P_j on fait appel récursivement à la méthode des transformées de Laplace itérées; au-delà de P_j on utilise une autre méthode. On peut également utiliser ce procédé pour calculer la somme $f(x)$ dans tout le plan complexe.

Différentes méthodes de prolongement analytique sont étudiées. Celle que nous utilisons est la méthode de Runge-Kutta à partir de conditions initiales (exactes à l'origine O pour le dernier niveau et numériques en P_j pour les autres). Elle est précise tant que le chemin d'intégration Π_j passe suffisamment loin des singularités et tant que l'ordre des équations différentielles est relativement faible.

Si l'ordre de l'équation différentielle dont est solution $\hat{\phi}^{j+1}(t)$ est élevé (de l'ordre de 7 ou 8 pour $\hat{\phi}^{r+1}(t)$ par exemple) et si le rayon de convergence de la série est faible (par exemple 0.25) la longueur du pas d'intégration utilisé dans la méthode de Runge-Kutta doit être choisie très petit. Ceci s'explique par le fait que dans ce cas les dérivées vont être très grandes en valeur absolue (par exemple de l'ordre de $k!4^k$ (pour la dérivée k -ième et pour un rayon de convergence de 0.25).

D'autres méthodes sont en cours d'étude afin d'obtenir des accélérations de convergence en vue d'améliorer le prolongement analytique. Certaines ont été exposées dans un précédent article [16]. D'autres (les approximants de Padé notamment) sont étudiées par C. Chaffy [7]. J. Thomann et moi-même sommes également en train de tester la méthode d'accélération de convergence décrite par J.-P. Ramis et J. Martinet qui consiste à appliquer la transformée de Borel-Laplace $f(x) = L_{k,d} S \hat{B}_k \hat{f}(x)$ à une série convergente $\hat{f}(x)$ (k dépend de la localisation des singularités de $f(x)$) pour obtenir la somme $f(x)$ dans le k -disque de Borel maximal.

4.4. Logiciel Compas

Le logiciel Compas permet de choisir des méthodes de calcul sur des chemins du plan complexe et de calculer et représenter graphiquement les solutions d'équations différentielles ordinaires à coefficients polynomiaux sur ces chemins. La multisommation (méthode de Balser) est intégrée à Compas et utilisée en tant que méthode de prolongement analytique.

Pour calculer les transformées de Laplace itérées et donc la somme d'une série nous devons

choisir un chemin d'intégration pour chaque sous-série associée à un plan de Borel. Ce chemin doit éviter le voisinage des singularités. Une méthode de calcul est associée à chaque élément de chemin (segmentΓarc de cercle ou toute autre courbe). Par exempleΓsi la méthode est de type *Série*Γle prolongement analytique se fait en appelant l'algorithme de Balser. Nous utilisons ainsi le logiciel Compas comme outil interactif de prolongement analytiqueΓce logiciel permettant de lier à des chemins du plan complexe des méthodes de prolongement. Prévue dans CompasΓla visualisation des transformées de Borel sur les chemins dans les différents plans permet d'optimiser les chemins et donc d'améliorer la précision du résultat.

5. Implantation

Nous décrivons à présent les primitives informatiques que nous avons été amenés à définir en vue de l'implantation algorithmique du théorème 2.3.3 de W. Balser. Ces primitives sont implantées pour l'instant en A^\sharp . Elles pourront ultérieurement être écrites dans d'autres systèmes de calcul formel (MapleΓAxiomΓReduce).

L'un des intérêts de la réalisation en A^\sharp est de tester la librairie prototype *Sumit*Γdéveloppée par M. BronsteinΓimplantant notamment les polynômes d'Ore.

5.1. Primitives

5.1.1. BalserRing

BalserRing : Category

Description

BalserRing est la catégorie des anneaux complexes. Dans ce qui suitΓ S désigne un tel anneau. Il existe ainsi une application *coerce* : $S \rightarrow \text{Complex DoubleFloat}$ (utilisée dans la partie graphique et numérique); on peut définir le quotient de deux éléments de S ; l'anneau des polynômes à coefficients dans S est un anneau commutatif.

5.1.2. AutoDer

import from AutoDer(S)

Description

AutoDer(S) définit les dérivations et l'automorphisme utilisés.

Fonctions exportées

SUP ==> SparseUnivariatePolynomial

- $\text{sigma} : () \rightarrow \text{Automorphism}(\text{SUP}(\text{S}^\Gamma j^\Gamma))$
 $\text{sigma}()$ est l'automorphisme associé à la fonction $P \in \text{SUP}(S, "j") \rightarrow P(j + 1)$.

- $D : () \rightarrow \text{Derivation}(\text{SUP}(\text{SI}^{\circ}x))$
 $D()$ est la dérivation associée à la fonction $P \in \text{SUP}(S, "x") \rightarrow \frac{d}{dx}P$.
- $\text{Theta} : () \rightarrow \text{Derivation}(\text{SUP}(\text{SI}^{\circ}x))$
 $\text{Theta}()$ est la dérivation associée à la fonction $P \in \text{SUP}(S, "x") \rightarrow (x \frac{d}{dx})P$.

5.1.3. EulerLinearOrdinaryPolynomialDifferentialOperator

`import from EulerLinearOrdinaryPolynomialDifferentialOperator(S)`

Description

`EulerLinearOrdinaryPolynomialDifferentialOperator(S)` implémente les opérateurs linéaires différentiels dont les coefficients sont des polynômes en la variable x à coefficients dans S . L'opérateur de dérivation utilisé est l'opérateur d'Euler.

5.1.4. LinearOrdinaryPolynomialDifferenceOperator

`import from LinearOrdinaryPolynomialDifferenceOperator(S)`

Description

`LinearOrdinaryPolynomialDifferenceOperator(S)` implémente les opérateurs linéaires aux différences dont les coefficients sont des polynômes en la variable j à coefficients dans S . L'opérateur de translation utilisé est l'automorphisme associé à la fonction $P \in \text{SUP}(S, "j") \rightarrow P(j+1)$.

5.1.5. LinearOrdinaryPolynomialDifferentialOperator

`import from LinearOrdinaryPolynomialDifferentialOperator(S)`

Description

`LinearOrdinaryPolynomialDifferentialOperator(S)` implémente les opérateurs linéaires différentiels dont les coefficients sont des polynômes en la variable x à coefficients dans S . L'opérateur de dérivation utilisé est celui associé à la fonction $P \in \text{SUP}(S, "x") \rightarrow \frac{d}{dx}P(x)$.

5.1.6. PowerSeries

`import from PowerSeries(S)`

Description

`PowerSeries(S)` est l'anneau des séries formelles entières $\sum_{n \geq 0} a_n x^n$ définies par un opérateur aux différences et par les premiers termes $a_0, \dots, a_{m-1} \in S$.

Fonctions exportées

SI \implies SingleInteger
 Z \implies Integer
 SUP \implies SparseUnivariatePolynomial
 POLDELTO \implies LinearOrdinaryPolynomialDifferenceOperator

- `createPowerSeries` : (List Ratio(S) Γ POLDELTO(S)) \rightarrow %
createPowerSeries(cond, δ) crée une série formelle à partir de la liste *cond* contenant les premiers termes et de l'opérateur aux différences δ à condition que le nombre de conditions initiales soit supérieur au degré de l'opérateur δ .
- `dispose!` : % \rightarrow ()
dispose!(\hat{f}) indique que la série formelle ne sera plus utilisée.
- `coerce` : % \rightarrow Record(ic : List Ratio(S) Γ delt : POLDELTO(S))
coerce(\hat{f}) donne la représentation interne de la série \hat{f} . Le champ *ic* permet d'accéder à la liste des conditions initiales Γ *delt* à l'opérateur aux différences.
- `initcond` : % \rightarrow List Ratio(S)
initcond(\hat{f}) retourne la liste des premiers termes définissant la série.
- `delto` : % \rightarrow POLDELTO(S)
delto(\hat{f}) retourne l'opérateur aux différences.
- `coherent?` : % \rightarrow Boolean
coherent?(\hat{f}) teste si les premiers termes sont solutions de l'équation aux différences.
- `coefficients` : (% ISI) \rightarrow Partial List(Ratio(S))
coefficients(\hat{f} , n) calcule si c'est possible les nb premiers coefficients de la série et retourne la liste $[a_0, \dots, a_{nb-1}]$.
- `coefficients` : (% IZ) \rightarrow Partial List(Ratio(S))
coefficients(\hat{f} , n) calcule si c'est possible les nb premiers coefficients de la série et retourne la liste $[a_0, \dots, a_{nb-1}]$.
- `coefficient` : (% ISI) \rightarrow Partial Ratio(S)
coefficient(\hat{f} , n) retourne le coefficient a_{nb} de la série.
- `coefficient` : (% IZ) \rightarrow Partial Ratio(S)
coefficient(\hat{f} , n) retourne le coefficient a_{nb} de la série.
- `nextCoeff` : (List Ratio(S) Γ POLDELTO(S)) \rightarrow Partial Ratio(S)
nextCoeff(l , δ) calcule à partir de l'opérateur aux différences δ et des termes de la liste l le coefficient suivant.
- `<<` : (TextWriter I%) \rightarrow TextWriter

5.1.7. Borel

import from Borel(S)

Description

Borel(S) comprend les fonctions utilisées pour le calcul des transformées de Borel.

Fonctions exportées

SI \Rightarrow SingleInteger
 Z \Rightarrow Integer
 POLDELTO \Rightarrow LinearOrdinaryPolynomialDifferenceOperator

- *kappasummability* : List Ratio SI \rightarrow List Ratio SI
 Si $sum = [k_1, \dots, k_r]$ $\Gamma kappasummability(sum)$ retourne la liste $[\kappa_1, \dots, \kappa_r]$ où $\kappa_1 = k_1$ et $1/\kappa_l = 1/k_l - 1/k_{l-1}$ pour $l = 2, \dots, r$.
- *split* : (POLDELTO(S)FSI) \rightarrow POLDELTO(S)
 Si $\delta = P_0(j) + P_1(j)\sigma + \dots + P_r(j)\sigma^r$ $\Gamma split(\delta, \alpha)$ retourne un opérateur aux différences de la forme $Q_0(j) + Q_1(j)\sigma^\alpha + \dots + Q_{\bar{r}}(j)\sigma^{\bar{r}\alpha}$ multiple de δ ; si $P_0(j)a_j + P_1(j)a_{j+1} + \dots + P_r(j)a_{j+r} = 0$ alors $Q_0(j)a_j + Q_1(j)a_{j+\alpha} + \dots + Q_{\bar{r}}(j)a_{j+\bar{r}\alpha} = 0$ pour j assez grand.
- *split* : (PowerSeries(S)FSI) \rightarrow List PowerSeries(S)
 $split(\hat{f}, \alpha)$ retourne la liste des α sous-séries de \hat{f} . Ces sous-séries n'ont pas nécessairement un coefficient constant nul. La série $\hat{f}(x)$ est égale à

$$\sum_{q=0}^{\alpha-1} x^q \hat{f}_q(x).$$

- *borel* : (PowerSeries(S)FSI) \rightarrow PowerSeries(S)
 $borel(\hat{f}, \beta)$ retourne la transformée de Borel formelle d'ordre $1/\beta$ de $\hat{f}(x)$. Il ne s'agit pas exactement de la transformée de Borel mais de l'application qui à $\hat{f}(x) = \sum_{j \geq 0} a_j x^j$ associe la série formelle

$$borel(\hat{f}, \beta) = \sum_{j \geq 0} \frac{a_{j+1}}{\Gamma((j+1)\beta)} x^j.$$

5.1.8. Differential

import from Differential(S)

Description

Differential(S) comprend les fonctions qui permettent de passer d'une série formelle définie par un opérateur aux différences à une équation différentielle dont est solution cette série.

Fonctions exportées

SUP ==> SparseUnivariatePolynomial
 POLODO ==> LinearOrdinaryPolynomialDifferentialOperator
 LOEDO ==> EulerLinearOrdinaryPolynomialDifferentialOperator

- `diffequation : PowerSeries(S)`
 $\rightarrow \text{Record}(\text{eqdiff} : \text{POLODO}(S) \text{sec} : \text{SUP}(S[x]))$
diffequation(\hat{f}) détermine une équation différentielle dont est solution la série formelle \hat{f} ; *diffequation*(\hat{f}).*eqdiff* est l'opérateur différentiel le second membre étant donné par *diffequation*(\hat{f}).*sec*.
- `eulerequation : Powerseries(S)`
 $\rightarrow \text{Record}(\text{eqdiff} : \text{LOEDO}(S) \text{sec} : \text{SUP}(S[x]))$
eulerequation(\hat{f}) détermine une équation différentielle dont est solution la série formelle \hat{f} ; *eulerequation*(\hat{f}).*eqdiff* est l'opérateur différentiel (la dérivation est l'opérateur d'Euler) *eulerequation*(\hat{f}).*sec* est le second membre.

5.1.1.9. LagLeg

import from LagLeg

Description

LagLeg fournit les abscisses et les poids de Gauss-Laguerre et de Gauss-Legendre.

Fonctions exportées

F ==> DoubleFloat

- `laguerre : GeoLink \rightarrow Record(abscl : List Fweight : List F)`
laguerre(*gl*) retourne les abscisses (positives) et les poids de Laguerre utilisés pour la méthode de Gauss-Laguerre à *nlag* points (*nlag* est le nombre de points associés au chemin élémentaire *gl*). Nous nous limitons pour l'instant à 32 points.
- `legendre : GeoLink \rightarrow Record(abscl : List Fweight : List F)`
legendre(*gl*) retourne les abscisses (comprises entre 0 et 1) et les poids de Legendre utilisés pour la méthode de Gauss-Legendre à *nleg* points (*nleg* est le nombre de points associés au chemin élémentaire *gl*). Nous nous limitons pour l'instant à 32 points.

5.1.1.10. Laplace

import from Laplace(S)

Description

Laplace(S) comprend les fonctions utiles au calcul des transformées de Laplace.

Fonctions exportées

SI ==> SingleInteger
 Z ==> Integer
 CF ==> Complex DoubleFloat

- `readpath : () → Path(S)`
readpath() est le chemin d'intégration de l'intégrale de Laplace. Il est constitué de *maillons*. Un maillon est un chemin élémentaire (segmentArc de cercle...) auquel est associée une méthode de calcul. Si la méthode de calcul associée à un maillon est de type *Serie* la fonction à intégrer sera calculée itérativement par l'algorithme de Balser. Si elle est de type *Runge-Kutta* le prolongement sera fait par la méthode de Runge-Kutta. Nous nous limitons pour l'instant à ces deux seules méthodes de calcul. Si nous ne sommes pas au dernier niveau la méthode de calcul associée au premier maillon doit être de type *Serie* (l'origine étant généralement une singularité irrégulière). Le prolongement sur le dernier maillon du chemin sera effectué en les points de Gauss-Laguerre. Sur tous les autres maillons on utilise les points de Gauss-Legendre.
- `derivee : (List Record(listBorel : Record(vi : List CFF phi qvi : List CFF wi : List CF)) initerms : List Ratio(S)) (SI SIICFIZ) → List CF`
derivee(borellist, α, β, zr, n) calcule $f(zr)$ et les dérivées jusqu'à l'ordre n de la série; *borellist* contient les prolongements des transformées de Borel d'ordre $1/\beta$ des α sous-séries.

5.1.11. Runge_Kutta

`import from Runge_Kutta(S)`

Description

`Runge_Kutta(S)` comprend les fonctions utilisées pour effectuer le prolongement analytique par la méthode de Runge-Kutta.

Fonctions exportées

SI ==> SingleInteger
 F ==> DoubleFloat
 CF ==> Complex DoubleFloat
 SUP ==> SparseUnivariatePolynomial
 POLODO ==> LinearOrdinaryPolynomialDifferentialOperator
 LOEDO ==> EulerLinearOrdinaryPolynomialDifferentialOperator

- `generateDerivs : (POLODO(S) ISUP(SI"x")) (FF → CFF → CFFCF) → (List CFF) → List CF`
generateDerivs(eq, sec, γ, γ', t_0) transforme l'équation différentielle en un système différentiel d'ordre 1 le long de la courbe $x = \gamma(t)$ et retourne la valeur de ce vecteur dérivé en t_0 .

- `eulerDerivs` : $(\text{LOEDO}(\text{S})\text{ISUP}(\text{S}^{\text{r}}\text{x}))\Gamma\text{F} \rightarrow \text{CFTF} \rightarrow \text{CFTCF}) \rightarrow (\text{List CFTF}) \rightarrow \text{List CF}$
`eulerDerivs(eq, sec, γ , γ' , t_0)` transforme l'équation différentielle en un système différentiel d'ordre 1 le long de la courbe $x = \gamma(t)$ et retourne la valeur de ce vecteur dérivé en t_0 .
- `rk4f` : $(\text{List CFTFTFSIF}(\text{List CFTF}) \rightarrow \text{List CF}) \rightarrow \text{List CF}$
`rk4f(cond, t_0 , t_1 , nstep, derivs)` retourne la valeur du vecteur en t_1 connaissant sa valeur *cond* en t_0 par une méthode de Runge- Kutta à *nstep* étapes (nous nous limitons pour l'instant à *nstep*= 10.) La fonction *derivs* est la fonction dérivée de ce vecteur.

5.1.12. Balser

`import from Balser(S)`

Description

`Balser(S)` comprend les fonctions utilisées pour le calcul des sommes de séries formelles en utilisant l'algorithme de Balser.

Fonctions exportées

`SI` \Rightarrow `SingleInteger`
`Z` \Rightarrow `Integer`
`CF` \Rightarrow `Complex DoubleFloat`
`POLODO` \Rightarrow `LinearOrdinaryPolynomialDifferentialOperator`

- `balser` : $(\text{PowerSeries}(\text{S})\Gamma\text{List Ratio SIFCF}) \rightarrow \text{CF}$
`balser(\hat{f} , [k_1, \dots, k_r], zr)` calcule la somme de la série $\hat{f}(x)$ en zr la série étant (k_1, \dots, k_r) -sommable.
- `balser` : $(\text{PowerSeries}(\text{S})\Gamma\text{List Ratio SIFCFZ}) \rightarrow \text{List CF}$
`balser(\hat{f} , [k_1, \dots, k_r], zr , n)` calcule la somme f de la série $\hat{f}(x)$ en zr ainsi que toutes les dérivées jusqu'à l'ordre n (la série est supposée (k_1, \dots, k_r) -sommable) et retourne la liste $[f(zr), f'(zr), \dots, f^{(n)}(zr)]$.
- `iLbalser` : $(\text{PowerSeries}(\text{S})\Gamma\text{List Ratio SIFCF}) \rightarrow \text{CF}$
`iLbalser(\hat{f} , [$\kappa_1, \dots, \kappa_r$], zr)` calcule la somme de la série \hat{f} en zr ; la série est $(\kappa_1, \dots, \kappa_r) - iL$ sommable.
- `iLbalser` : $(\text{PowerSeries}(\text{S})\Gamma\text{List Ratio SIFCFZ}) \rightarrow \text{List CF}$
`iLbalser(\hat{f} , [$\kappa_1, \dots, \kappa_r$], zr , n)` calcule la somme f en zr ainsi que toutes les dérivées jusqu'à l'ordre n et retourne la liste $[f(zr), f'(zr), \dots, f^{(n)}(zr)]$. La série \hat{f} est $(\kappa_1, \dots, \kappa_r) - iL$ sommable.
- `balserpath` : $(\text{PowerSeries}(\text{S})\Gamma\text{List Ratio SIFPath}(\text{S})) \rightarrow \text{Record}(\text{abscisse} : \text{List CFTsum} : \text{List CF})$
Si $\hat{f}(x)$ est une série (k_1, \dots, k_r) -sommable `balserpath(\hat{f} , [k_1, \dots, k_r], γ)` calcule la somme de \hat{f} le long du chemin γ aux points (équidistants sur chaque chemin élémentaire) donnés

par le champ *abscisse*.

Le i -ième terme de $\text{balsepath}(\hat{f}, [k_1, \dots, k_r], \gamma).sum$ est la valeur de la somme au i -ième point de la liste $\text{balsepath}(\hat{f}, [k_1, \dots, k_r], \gamma).abscisse$.

- **analytic** : $(\text{PowerSeries}(S)\Gamma\text{Path}(S)) \rightarrow \text{Record}(abscisse : \text{List CF}, sum : \text{List CF})$
 $\text{analytic}(\hat{f}, \gamma)$ calcule le prolongement analytique de la série à partir de l'origine le long du chemin γ . La série est supposée convergente à l'origine.

5.2. Contenu algorithmique

Les programmes Γ écrits en $A^\sharp\Gamma$ sont donnés en annexe. Nous ne précisons ici que les points qui nous semblent importants.

5.2.1. CreatePowerSeries

```
import from PowerSeries(S)
```

Description

Soit $\hat{f}(x) = \sum_{j \geq 0} a_j x^j$ définie par

$$\begin{cases} \delta = P_0(j)a_j + \dots + P_r(j)a_{j+r} = 0, & j \geq \mu = m - r \\ cond = [a_0, \dots, a_{m-1}] \end{cases}$$

Si $\mu < 0$ il n'y a pas assez de conditions initiales et on sort un message d'erreur.

Si $P_0(j) = \dots = P_{r_0-1}(j) = 0$ pour $r_0 \leq r$ et $P_{r_0}(j) \neq 0$ on transforme δ en l'opérateur $P_{r_0}(j - r_0)\sigma^0 + \dots + P_r(j - r_0)\sigma^{r-r_0}$ que l'on simplifie ensuite par le plus grand commun diviseur des coefficients.

On suppose que le polynôme $P_r(j)$ ne s'annule pas pour $j \geq \mu$. Une série est donc nulle si et seulement si les premiers coefficients sont nuls. La série 0 est définie par $\text{createPowerSeries}(1, [])$ (ici $r = m = 0$).

La série 1 est définie par $\text{createPowerSeries}(1, [1])$ ($m = 1, r = 0$).

5.2.2. Coherent?

```
import from PowerSeries(S)
```

Description

Cette fonction teste si les premiers termes sont solutions de l'équation aux différences c'est-à-dire si $P_0(j)a_j + \dots + P_r(j)a_{j+r} = 0$ pour $j \geq 0$.

5.2.3. Coefficients

```
import from PowerSeries(S)
```

Description

$coefficients(\hat{f}, n)$ où $\hat{f} \in PowerSeries(S)$ calcule les n premiers coefficients de la série. Soit m le nombre des conditions initiales. Si $n > m$ nous devons calculer les termes en utilisant l'équation aux différences. Si le coefficient de tête $P_r(j)$ s'annule pour $j_0 \in [\mu, \mu + 1, \dots, \mu + n - 1 - m]$ ($\mu = m - r$) on ne peut pas calculer le coefficient a_{j_0+r} et on sort un message d'erreur.

5.2.4. Split

import from Borel(S)

Description

$split(\delta, \alpha)$ calcule un opérateur multiple de δ ne comportant que des termes de la forme $P_k(j)\sigma^{\alpha k}$. On suppose $\alpha \geq 1$. Si $\alpha = 1$ on retourne δ .

La fonction $apply(delt, l)$ retourne l'opérateur $\sigma^l delt$.

Soit (S) la liste des opérateurs $apply(\delta, l)$ pour $l = 0, \dots, r(\alpha - 1)$ (r est le degré de δ).
Tant qu'on n'a pas fini

- Soit $delt$ le premier opérateur de (S) .
- Chercher dans $delt$ le plus petit indice $k \in \mathbb{N}$ tel que le coefficient de $delt$ de degré k soit non nul et tel que k ne divise pas α .
- Si un tel indice existe on s'arrête; sinon
 - Soit $P_k(j)$ le coefficient de $delt$ de degré k .
 - Parcourir les autres opérateurs $delt_l$ ($l = 2, \dots, \#(S)$) de (S) . On remplace (S) par les $\#(S) - 1$ opérateurs obtenus en éliminant le cas échéant le coefficient de degré k (si le coefficient $P_k^l(j)$ de degré k de $delt_l$ est non nul on remplace $delt_l$ par $P_k(j) delt_l - P_k^l(j) delt$).

5.2.5. Split

import from Borel(S)

Description

$split(\hat{f}, \alpha)$ retourne sous forme de liste les α sous-séries de la série formelle. On suppose que $\hat{f} = \sum_{j \geq 0} a_j x^j$ est définie par $\delta = P_0(j) + \dots + P_r(j)\sigma^r$ et par ses premiers termes $[a_0, \dots, a_{m-1}]$. Soit δ^α l'opérateur $split(\delta, \alpha)$. Il est de la forme $Q_0(j) + Q_1(j)\sigma^\alpha + \dots + Q_{\bar{r}}(j)\sigma^{\bar{r}\alpha}$ et $Q_0(j)a_j + Q_1(j)a_{j+\alpha} + \dots + Q_{\bar{r}}(j)a_{j+\bar{r}\alpha} = 0$ pour $j \geq \mu = m - r$.

Pour $q = 0, \dots, \alpha - 1$ l'opérateur $Q_0(j\alpha + q) + Q_1(j\alpha + q)\sigma + \dots + Q_{\bar{r}}(j\alpha + q)\sigma^{\bar{r}}$ définit la q -ième sous-série $\hat{f}_q(x) = \sum_{j \geq 0} a_{j\alpha+q} x^j$. La récurrence est valable pour les indices $j \in \mathbb{N}$

tels que $j\alpha + q \geq \mu\Gamma$ c'est-à-dire $j \geq \mu_q = \max(0, (\mu - q)/\alpha)$. Les $m_q = \mu_q + \bar{r}$ conditions initiales $a_q, a_{q+\alpha}, \dots, a_{q+(m_q-1)\alpha}$ sont calculées à partir de l'équation aux différences définissant \hat{f} . On vérifie aisément que pour $q = 0, \dots, \alpha - 1$ $q + (m_q - 1)\alpha < \bar{r}\alpha + \mu$. Les premiers coefficients définissant chaque sous-série sont donc facilement calculés à partir de la liste $[a_0, a_1, \dots, a_{\bar{r}\alpha + \mu - 1}] = \text{coefficients}(\hat{f}, \bar{r}\alpha + \mu)$.

5.2.6. Borel

import from Borel(S)

Description

Soit $\hat{f}(x) = \sum_{j \geq 0} a_j x^j$ une série formelle et $\beta \in \mathbb{N}^*$. La fonction $\text{borel}(\hat{f}, \beta)$ retourne la série $\hat{\phi}(x) = \sum_{j \geq 0} b_j x^j$ avec $b_j = \frac{a_{j+1}}{(j\beta + \beta - 1)!}$.

Si $P_0(j)a_j + \dots + P_r(j)a_{j+r} = 0$ alors $P_0(j+1)a_{j+1} + \dots + P_r(j+1)a_{j+1+r} = 0$ et $P_0(j+1)(j\beta + \beta - 1)! b_j + \dots + P_r(j+1)((j+r)\beta + \beta - 1)! b_{j+r} = 0$.

Or pour $l = 1, \dots, r\Gamma$

$$Q_l(j) = \frac{((j+l)\beta + \beta - 1)!}{(j\beta + \beta - 1)!} = \prod_{m=1}^{l\beta} (j\beta + \beta - 1 + m) \in \text{SUP}(S, "j").$$

Donc $P_0(j+1)b_j + P_1(j+1)Q_1(j)b_{j+1} + \dots + P_r(j+1)Q_r(j)b_{j+r} = 0$: la série $\hat{\phi}(x)$ est représentée par un opérateur linéaire aux différences dont les coefficients sont des polynômes en la variable j à coefficients dans S .

5.2.7. Legendre

import from LagLeg

Description

Nous devons calculer des intégrales de la forme

$$I(x) = \frac{1}{\beta} \int_{t_0}^{t_1} \Psi(t) \exp \left(-\left(\frac{t}{x}\right)^{\frac{1}{\beta}} \right) dt$$

où t décrit une courbe intégrale de la forme $t = \gamma(u)\Gamma u \in [0 \dots 1]$ ($t_0 = \gamma(0)$ et $t_1 = \gamma(1)$). L'intégrale $I(x)$ est égale à

$$\frac{1}{\beta} \int_0^1 \Psi(\gamma(u)) \exp \left(-\left(\frac{\gamma(u)}{x}\right)^{\frac{1}{\beta}} \right) \gamma'(u) du$$

et est calculée numériquement par une quadrature de Gauss-Legendre à N points :

$$\frac{1}{\beta} \sum_{i=1}^N A_i \Psi(\gamma(u_i)) \exp \left(-\left(\frac{\gamma(u_i)}{x}\right)^{\frac{1}{\beta}} \right) \gamma'(u_i).$$

La fonction *legendre(gl)* retourne la liste des poids $(A_i)_{i=1,\dots,N}$ et des abscisses $(u_i)_{i=1,\dots,N}$ Γgl étant un chemin élémentaire défini par le nombre de points N et par les courbes γ et γ' .

5.2.8. Laguerre

import from LagLeg

Description

Nous devons calculer des intégrales de la forme

$$I(x) = \frac{1}{\beta} \int_{t_0}^{\infty} \Psi(t) \exp\left(-\left(\frac{t}{x}\right)^{\frac{1}{\beta}}\right) dt$$

où t décrit une courbe intégrale de la forme $t = \gamma(u)\Gamma u \in \mathbb{R}^+$ ($t_0 = \gamma(0)$). L'intégrale $I(x)$ est égale à

$$\frac{1}{\beta} \int_0^{+\infty} \Psi(\gamma(u)) \exp\left(-\left(\frac{\gamma(u)}{x}\right)^{\frac{1}{\beta}}\right) \gamma'(u) du$$

et est calculée numériquement par une quadrature de Gauss-Laguerre à N points :

$$\frac{1}{\beta} \sum_{i=1}^N A_i \Psi(\gamma(u_i)) \exp\left(-\left(\frac{\gamma(u_i)}{x}\right)^{\frac{1}{\beta}}\right) \gamma'(u_i).$$

La fonction *laguerre(gl)* retourne la liste des poids $(A_i)_{i=1,\dots,N}$ et des abscisses $(u_i)_{i=1,\dots,N}$ Γgl étant un chemin élémentaire défini par le nombre de points N et par les courbes γ et γ' .

5.2.9. Derivee

import from Laplace(S)

Description

Soit $\hat{f}(x) = \sum_{j \geq 0} a_j x^j$ une série $(\kappa_1, \dots, \kappa_r) - iL$ sommable Γ avec $\kappa_1 = \alpha_1/\beta_1$. Soient $\hat{f}_q(x) = \sum_{j \geq 0} a_{j\alpha_1+q} x^j$ ($q = 0, \dots, \alpha_1 - 1$) les α_1 sous-séries et $\hat{\phi}_q(x) = \sum_{j \geq 0} b_j^q x^j$ avec $b_j^q = \frac{a_{(j+1)\alpha_1+q}}{(j\beta_1+\beta_1-1)!}$ les transformées de Borel d'ordre $1/\beta_1$.

La somme f en x est égale à $f(x) = \sum_{q=0}^{\alpha_1-1} x^q (a_q + f_q(x^{\alpha_1}))$.

Chaque somme

$$f_q(x^{\alpha_1}) = \frac{1}{\beta_1} \int_{\Pi_q} \phi_q(t) \exp\left(-\left(\frac{t}{x^{\alpha_1}}\right)^{\frac{1}{\beta_1}}\right) dt$$

est égale à la somme des intégrales le long de chaque chemin élémentaire Γ que l'on calcule par les méthodes décrites en 5.2.7 et 5.2.8. On obtient alors une expression de la forme

$$f_q(x^{\alpha_1}) = \sum_{i=1}^{N_q} \phi_q(\gamma(u_i^q)) \exp\left(-\left(\frac{\gamma(u_i^q)}{x^{\alpha_1}}\right)^{\frac{1}{\beta_1}}\right) w_i^q$$

$$(w_i^q = A_i^q \gamma'(u_i^q)).$$

La dérivée n -ième de $f(x)$ est égale à

$$f^{(n)}(x) = \sum_{q=0}^{\alpha_1-1} \sum_{l=0}^n C_n^l(x^q)^{(l)} (a_q + f_q(x^{\alpha_1}))^{(n-l)}.$$

Les dérivées successives de la fonction $f_q(x^{\alpha_1})$ se calculent à partir de la formule intégrale :

$$\frac{d}{dx}(f_q(x^{\alpha_1})) = \frac{1}{\beta_1} \kappa_1 x^{-\kappa_1-1} \int_{\Pi_q} \phi_q(t) t^{\frac{1}{\beta_1}} \exp\left(-\left(\frac{t}{x^{\alpha_1}}\right)^{\frac{1}{\beta_1}}\right) dt.$$

Plus généralement la dérivée de $a x^b \int_{\Pi_q} \phi_q(t) t^c \exp\left(-\left(\frac{t}{x^{\alpha_1}}\right)^{\frac{1}{\beta_1}}\right) dt$ est égale à

$$a b x^{b-1} \int_{\Pi_q} \phi_q(t) t^c \exp\left(-\left(\frac{t}{x^{\alpha_1}}\right)^{\frac{1}{\beta_1}}\right) dt + a \kappa_1 x^{b-\kappa_1-1} \int_{\Pi_q} \phi_q(t) t^{c+\frac{1}{\beta_1}} dt.$$

Les dérivées de $f_q(x^{\alpha_1})$ se calculent donc par récurrence. Si nous connaissons la liste des coefficients $[a, b, c]$ définissant la dérivée l -ième la dérivée $(l+1)$ -ième sera définie par la liste des coefficients $[a, b-1, c]$ et $[a \kappa_1, b-\kappa_1-1, c+\frac{1}{\beta_1}]$. Connaissant les valeurs des transformées de Borel en les abscisses de quadrature et les poids associés nous pouvons ainsi calculer les dérivées à n'importe quel ordre de la somme.

La fonction *derivee* a pour premier argument *borellist* l'enregistrement des prolongements des α_1 sous-séries c'est-à-dire les abscisses $\gamma(u_i^q)$ les sommes $\phi_q(\gamma(u_i^q))$ et les poids w_i^q ($i = 1, \dots, N_q$); *derivee(borellist, α_1, β_1, x, N)* retourne la liste $[f(x), f'(x), \dots, f^{(n)}(x)]$.

5.2.10. iLbalser

import from balser(S)

Description

Le calcul de la somme et des dérivées d'une série $(\kappa_1, \dots, \kappa_r) - iL$ sommable suit le schéma suivant :

- Scindage de la série en α_1 sous-séries ($\kappa_1 = \frac{\alpha_1}{\beta_1}$);
- Calcul des transformées de Borel d'ordre $\frac{1}{\beta_1}$;
- Prolongement de ces transformées $(\frac{\kappa_2}{\alpha_1}, \dots, \frac{\kappa_r}{\alpha_1}) - iL$ sommables;
- Calcul de la somme et des dérivées par quadrature.

Le prolongement d'une série formelle convergente se fait par la méthode de Runge-Kutta à partir des conditions initiales calculées exactement à l'origine. La valeur calculée au bout d'un chemin élémentaire donne les conditions initiales pour le chemin élémentaire suivant.

Le prolongement d'une série $(\kappa_1, \dots, \kappa_r) - iL$ sommable le long d'un chemin se fait de la façon suivante :

- On parcourt le chemin;
- Si la méthode de calcul associée au chemin élémentaire est de type *Serie* le calcul de la somme se fait récursivement (calcul des transformées de Borel des sous-séries...). Les dérivées calculées au bout du chemin élémentaire par quadrature donnent les conditions initiales au début du chemin élémentaire suivant;
- Si la méthode de calcul associée au chemin élémentaire est de type *Runge-Kutta* on utilise la méthode de Runge-Kutta à partir des conditions initiales que l'on a calculées auparavant. Les dérivées calculées au bout du chemin élémentaire par la méthode de Runge-Kutta donnent les conditions initiales au début du chemin élémentaire suivant.

6. Résultats

6.1. Equation d'Euler

La série formelle $\hat{E}(x) = \sum_{n \geq 0} (-1)^n n! x^{n+1}$ est solution de l'équation d'Euler

$$x^2 y' + y = x.$$

Sa transformée de Borel formelle (d'ordre 1) est la série $\sum_{n \geq 0} (-1)^n x^n$ laquelle admet comme prolongement analytique la fonction $\frac{1}{1+x}$. La série $\hat{E}(x)$ est donc 1-sommable dans toutes les directions exceptées les directions $\pi + 2\pi l$ ($l \in \mathbb{Z}$).

Nous choisissons comme direction de resommation la droite \mathbb{R}^+ et calculons la somme en 0.01 et 1. Pour cela nous avons écrit un fichier "un.as" où l'on définit la série d'Euler et où l'on calcule la somme en les trois points. Voici l'historique de la session :

```
$ asharp -g loop -w gc
This is a pre-release of AXIOM-XL. 'axiomxl -h info' for more details.
%1 >> #include "sumit.as"

Comp : 2460 msec., Interp : 30 msec.

%2 >> #library un "un.ao"
%3 >> import from un

Comp : 40 msec., Interp : 0 msec.

%4 >> session()

Serie d'Euler :
Initial Conditions :list((0/1), (1/1))
Difference equation :tau+j

Liste des pentes :
```

```
list((1/1))
```

Point en lequel on veut calculer la somme :

```
zr = (0.01 + 0 %i)
```

Somme :

```
(0.0099019422867330075 + 0 %i)
```

Point en lequel on veut calculer la somme :

```
zr = (0.100000000000000001 + 0 %i)
```

Somme :

```
(0.091563333939787855 + 0 %i)
```

Point en lequel on veut calculer la somme :

```
zr = (1 + 0 %i)
```

Somme :

```
(0.5963473622644494 + 0 %i)
```

Comp : 10 msec., Interp : 346490 msec.

Nous obtenons ainsi les valeurs :

$$E(0.01) = 0.0099019422867$$

$$E(0.1) = 0.0915633339398$$

$$E(1) = 0.5963473623$$

Remarque : Ces valeurs peuvent d'autre part être calculées à l'aide de tables connues. Soit $E_1(z) = \int_z^\infty \frac{e^{-t}}{t} dt$. Comme

$$\int_0^{+\infty} \frac{e^{-at}}{b+t} dt = e^{ab} E_1(ab)$$

pour tout a et $b > 0$

$$E(x) = \int_0^{+\infty} e^{-\frac{t}{x}} \frac{1}{1+t} dt = e^{\frac{1}{x}} E_1\left(\frac{1}{x}\right), \quad x > 0.$$

En utilisant la table de E_1 nous obtenons :

- $E(0.01) = e^{100} E_1(100) = 0.01 * 0.9901942287 = 0.009901942287$
- $E(0.1) = e^{10} E_1(10) = 0.1 * 0.9156333394 = 0.09156333394$
- $E(1) = e^1 E_1(1) = e * 0.219383934 = 0.5963473612$

La précision relative pour $E(0.01)$, $E(0.1)$ et $E(1)$ est donc respectivement égale à 3.10^{-11} , 2.10^{-12} et 2.10^{-9} .

6.2. Exemple de 2-sommabilité

Il s'agit d'un exemple étudié par J. Thomann [16]. Une solution formelle fondamentale de l'équation différentielle

$$4x^5y'' + 2x^2y' + y = 0$$

est donnée par $e^{\frac{1}{2x}}x^{-\frac{1}{2}}\hat{f}(x)$ où $\hat{f}(x)$ est solution de

$$4x^3y'' - 2(2x^2 + 2x - 1)y' + 3(x + 2)y = 0.$$

La série formelle $\hat{f}(x)$ est 2-sommable dans toutes les directions exceptées $\frac{\pi}{2} + \pi l$ ($l \in \mathbb{Z}$).

Nous calculons la somme en $x = \frac{1}{3}$ (voir l'historique) $\Gamma x = \frac{1}{3}e^{i\frac{\pi}{8}}$ et $x = \frac{1}{3}e^{i\frac{\pi}{4}}$.

```
$ asharp -g loop -w gc
This is a pre-release of AXIOM-XL. 'axiomxl -h info' for more details.
%1 >> #include "sumit.as"

Comp : 2460 msec., Interp : 30 msec.

%2 >> #library deux "deux.ao"
%3 >> import from deux

Comp : 40 msec., Interp : 0 msec.

%4 >> session()

Serie etudiee :
Initial Conditions :list((0/1), (1/1))
Difference equation :tau+j

Liste des pentes :
list((2/1))

Point en lequel on veut calculer la somme :
zr = (0.33333333333333331 + 0 %i)
Somme :
(0.069524243326470447 + 1.8007867203910085e-08 %i)

Comp : 10 msec., Interp : 963780 msec.
```

La direction de resommation est respectivement $\mathbb{R}^+ \Gamma e^{i\frac{\pi}{8}} \mathbb{R}^+$ et $e^{i\frac{\pi}{4}} \mathbb{R}^+$. Nous rappelons également les résultats obtenus par J. Thomann par la méthode des approximants de Padé et celle des séries de factorielles généralisées SFG [16] :

arg x	Balser	Padé	SFG
0	0.06952424	0.0695243	0.0695117
$\pi/8$	0.130661 - 0.3370434i	0.130661 - 0.337043i	0.130627 - 0.337015i
$\pi/4$	0.304842 - 0.632041i	0.304843 - 0.632041i	0.306122 - 0.630481i

6.3. Exemple de Ramis-Sibuya

Les directions singulières de resommation de la série formelle solution de l'équation de Ramis-Sibuya sont les directions $\frac{\pi}{2} + \frac{\pi}{2}l$ ($l \in \mathbb{Z}$).

La série formelle étant égale à $\hat{f}(x) = \hat{E}(x) + \hat{E}(x^2)$ la somme est égale à $f(x) = E(x) + E(x^2) = e^{\frac{1}{x}}E_1(\frac{1}{x}) + e^{\frac{1}{x^2}}E_1(\frac{1}{x^2})$ (voir 6.1). Nous obtenons à l'aide de la table de E_1 :

- $f(0.25) = e^4 E_1(4) + e^{16} E_1(16) = 0.265353775$
- $f(0.5) = e^2 E_1(2) + e^4 E_1(4) = 0.567674269$
- $f(1) = 2e E_1(1) = 1.192694722$

Pour calculer la somme en ces points par l'algorithme de Balser Γ nous choisissons comme direction de resommation pour $\hat{\phi}_l(x)$ et $\hat{\psi}_l(x)$ ($l = 0, 1$) la droite $e^{i\frac{\pi}{8}}\mathbb{R}^+$ (il ne faut pas passer trop près du point $x = 1/4$ qui est une singularité pour les séries convergentes $\hat{\psi}_l(x)$ ($l = 0, 1$)). Nous obtenons ainsi :

- $f(0.25) = 0.26535392 + 4.7525175 \cdot 10^{-7}i$
- $f(0.5) = 0.56545058 + 0.0025003453i$
- $f(1) = 1.1387162 + 0.080168043i$

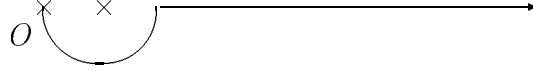
Nous remarquons que plus nous nous éloignons de l'origine Γ moins la précision est bonne. Nous pourrions choisir d'autres chemins (pas forcément les mêmes pour les séries $\hat{\phi}_l(x)$ et $\hat{\psi}_l(x)$) et un nombre de pas d'intégration plus élevé pour améliorer la précision du calcul. Une autre façon de calculer la somme en 1 consiste par exemple à calculer la somme et sa dérivée en 0.25 (où l'on a une assez bonne précision) et à effectuer le prolongement analytique de la somme le long du chemin $[0.25, 1]$ par une méthode de Runge-Kutta Γ à partir de l'équation différentielle dont est solution la somme et des conditions initiales en 0.25. Nous avons choisi une méthode de Runge-Kutta à pas fixe égal à 0.015 et obtenons

$$f(1) = 1.16035002 - 0.00016311899i$$

La précision obtenue est meilleure.

Nous souhaitons à présent calculer la somme au point $x = -0.25$.

On considère d'une part la direction de resommation $d^- = \pi - \epsilon$ ($\epsilon > 0$ petit). La direction de resommation au premier niveau est $\alpha_1 d^- = 2(\pi - \epsilon)$; celle du deuxième (et dernier) niveau est $\alpha_2 \alpha_1 d^- = 2(\pi - \epsilon)$. La droite de direction $\alpha_2 \alpha_1 d^-$ est proche de la singularité $\frac{1}{4}$ des fonctions $\psi_0(x)$ et $\psi_1(x)$. Nous avons testé l'algorithme avec différents chemins Γ et il se trouve que $\psi_0(x)$ et $\psi_1(x)$ se prolongent analytiquement dans la direction \mathbb{R}^+ (de plus Γ elles tendent vers 0). Nous allons considérer comme chemin d'intégration Π_2 :



Le chemin sur lequel nous allons intégrer les fonctions $\phi_0(x)$ et $\phi_1(x)$ est la droite de direction \mathbb{R}^+ . Le point P_1 à partir duquel est utilisée la méthode de Runge-Kutta est le point $(0.5, 0)$. Nous obtenons alors comme résultat :

$$f^-(-0.25) = -0.3009568828 + 0.05767302121i.$$

D'autre part nous pouvons également calculer la somme en -0.25 en considérant comme direction de resommation la direction $2(\pi + \epsilon)$. En prenant les symétriques par rapport à l'axe des réels des chemins précédents nous obtenons

$$f^+(-0.25) = -0.3009568828 - 0.05767302121i.$$

La différence des deux valeurs est

$$f^-(-0.25) - f^+(-0.25) = 0.1153460424i.$$

La différence n'est pas nulle : nous avons *traversé* la direction singulière \mathbb{R}^- . Il s'agit du phénomène de Stokes.

Nous allons comparer ce résultat avec la valeur exacte. La série formelle $\hat{f}(x)$ est égale à $\hat{f}(x) = \hat{E}(x) + \hat{E}(x^2)$. La série $\hat{E}(x)$ est 1-sommable $\Gamma \pi$ étant la direction singulière (modulo 2π). La série $\hat{E}(x^2)$ est 2-sommable Γ de directions singulières $\frac{\pi}{2} + \pi\mathbb{Z}$. La différence $f^-(-\rho) - f^+(-\rho)$ ($\rho > 0$) est égale à $E^-(-\rho) - E^+(-\rho)$ Γ ce que nous calculons aisément par la méthode des résidus. Nous obtenons :

$$E^-(-\rho) - E^+(-\rho) = 2\pi i \exp\left(-\frac{1}{\rho}\right).$$

Pour $\rho = 0.25$ Γ la différence est égale à $0.1150806i$. Le résultat obtenu est donc de l'ordre de 2.10^{-4} .

7. Conclusion

Les exemples que nous avons testés sont des cas d'école. Nous avons pu comparer nos résultats avec ceux obtenus par d'autres méthodes et ceux fournis par les tables. Ceci constitue un premier test de validité de l'ensemble des algorithmes.

Il reste un point important à étudier Γ à savoir le test de validité des résultats numériques

obtenus. Les résultats que nous avons obtenus pour la série d'Euler et la série formelle 2-sommable sont bons. Il semble cependant plus difficile d'obtenir une bonne précision pour la série de Ramis-Sibuya qui est $(2, 1)$ -sommable. Nous avons constaté une sensibilité des calculs par rapport aux chemins.

Il y a trois niveaux d'approximation qui ne sont pas encore maîtrisés actuellement : le calcul des conditions initiales floues au point P à un niveau intermédiaire (mis à part le dernier niveau) le prolongement analytique (pour l'instant il s'agit de la méthode de Runge-Kutta) et le calcul des transformées de Laplace. Dans l'état actuel d'avancement des travaux nous n'avons pas à notre disposition des calculs de bornes d'erreurs numériques.

L'apport de Compas permet pour l'instant de palier à l'absence de bornes d'erreurs. En pratique nous calculons un résultat en différents chemins (et par différentes méthodes). Par ses méthodes de visualisation Compas permet de tester la régularité des résultats obtenus sur les chemins dans les différents plans de Borel et de constater quand il y a un décrochement des méthodes numériques (ce qui arrive si le chemin est trop proche d'une singularité). Mon travail n'a pas porté sur la qualité des méthodes numériques de prolongement (par exemple la méthode de Runge-Kutta que nous avons implantée est une méthode d'ordre 4 à pas fixe). Ce travail reste encore à faire.

Quoi qu'il en soit l'algorithme de Balser que nous avons implanté est prêt à être utilisé. Nous avons ainsi à notre disposition un outil souple susceptible d'être interactif (cet aspect est en voie de réalisation) et pratique à l'emploi comprenant les bonnes primitives permettant de tester la validité des résultats par des recoupements.

Il fallait donner à l'utilisateur la possibilité de faire un choix entre différentes méthodes de prolongement analytique sur les chemins choisis dans les plans de Borel. Pour l'instant peu de méthodes sont disponibles plusieurs sont à l'étude. Le logiciel dans son état actuel permettra l'insertion et le test de ces méthodes dans le contexte général de la multisommation.

Références bibliographiques

- [1] W. Balser Γ *From divergent power series to analytic functions theory and application on multisummable power series* Γ Lecture notes in mathematics Γ Springer (1994).
- [2] W. Balser Γ *Summation of formal power series through iterated Laplace integrals* Γ Math. Scand. **70** (1992) 161-171
- [3] W. Balser Γ B.L.J. Braaksma Γ J.-P. Ramis Γ Y. Sibuya Γ *Multisummability of formal power series solutions of ordinary differential equations* Γ Asympt. Analysis **5** Γ Number 1 Γ Nov.(1991).
- [4] W. Balser Γ A. Tovbis Γ *Multisummability of iterated integrals* Γ Asympt. Anal. **7** (1993) 121-127.
- [5] A. Barkatou Γ A. Duval Γ *Sur les séries formelles solutions d'équations aux différences polynomiales* Γ Annal. Inst. Fourier Γ 44 (1994) Γ 495-524.
- [6] M. Bronstein Γ M. Petkovsek Γ *On Ore rings, linear operators and factorisation* Γ Preprint ETH Γ November 1993-200.
- [7] C. Chaffy-Camus Γ *Multivariate Padé approximation applied to analytic continuation* Γ à paraître dans les proceedings : Non linear numerical methods and rational approximation Γ Anvers 1993.
- [8] J. Della-Dora Γ C. Dicrescenzo Γ E. Tournier Γ *An algorithm to obtain formal solutions of a linear homogenous differential equation at an irregular singular point* Γ Lecture Notes in Computer Science. In : Calmet J.(ed.) Γ Lecture Notes in Computer Science Γ (Springer Γ 1982).
- [9] F. Jung Γ F. Naegele Γ J. Thomann Γ *An algorithm of multisummation of formal power series, solutions of linear ODE equations* Γ à paraître dans Proceedings IMACS Lille (1993).
- [10] B. Malgrange Γ J.-P. Ramis Γ *Fonctions multisommables* Γ Ann. Inst. Fourier Γ Grenoble **42** Γ 1-2 (1992).
- [11] J.-P. Ramis Γ *Les séries k-sommables et leurs applications* Γ Lecture Notes in Physics **126** Γ Springer (1980).
- [12] J.-P. Ramis Γ *Equations différentielles : Phénomène de Stokes et resommation* Γ C.R. Acad. Sc. Paris Γ t.301 Γ Série 1 Γ No.4 (1985).
- [13] J.-P. Ramis Γ Y. Sibuya Γ *Hukuhara domains and fundamental existence and uniqueness theorems for asymptotic solutions of Gevrey type* Γ Asymptotic Analysis **2** Γ 39-94 (1989).

- [14] F. Richard-Jung Γ *Représentations graphiques de solutions d'équations différentielles dans le champ complexe* Γ Thèse de l'Université de Strasbourg (1988).
- [15] F. Richard-Jung Γ *Le phénomène de Stokes en image* Γ RT 65 LMC Grenoble (1991).
- [16] J. Thomann Γ *Resommation des séries formelles solutions d'équations différentielles linéaires ordinaires du second ordre dans le champ complexe au voisinage de singularités irrégulières* Γ Numer.Math. **58** Γ 503-535 (1990).
- [17] J. Thomann Γ *Problèmes algorithmiques posés par la multisommation* Γ Journées de printemps de la PRC Math-Info (1990).
- [18] J. Thomann Γ *Procédés formels et numériques de sommation de séries solutions d'équations différentielles* Γ Expo. Math. 13 (1995) Γ 223-246.
- [19] E. Tournier Γ *Solutions formelles d'équations différentielles* Γ Thèse d'Etat de l'Université de Grenoble (1988).

Bibliographie

- [1] W. Balser Γ *From divergent power series to analytic functions theory and application on multisummable power series* Γ Lecture notes in mathematics Γ Springer (1994).
- [2] W. Balser Γ *Summation of formal power series through iterated Laplace integrals* Γ Math. Scand. **70** (1992) 161-171
- [3] W. Balser Γ B.L.J. Braaksma Γ J.-P. Ramis Γ Y. Sibuya Γ *Multisummability of formal power series solutions of ordinary differential equations* Γ Asympt. Analysis **5** Γ Number 1 Γ Nov.(1991).
- [4] W. Balser Γ A. Tovbis Γ *Multisummability of iterated integrals* Γ Asympt. Anal. **7** (1993) 121-127.
- [5] A. Barkatou Γ A. Duval Γ *Sur les séries formelles solutions d'équations aux différences polynomiales* Γ Annal. Inst. Fourier Γ 44 (1994) Γ 495-524.
- [6] J.-P. Bézivin Γ *Sur les équations fonctionnelles aux q -différences*. Preprint Paris VI (1990).
- [7] M. Bronstein Γ M. Petkovsek Γ *On Ore rings, linear operators and factorisation* Γ Preprint ETH Γ November 1993-200.
- [8] C. Chaffy-Camus Γ *Multivariate Padé approximation applied to analytic continuation* Γ à paraître dans les proceedings : Non linear numerical methods and rational approximation Γ Anvers 1993.
- [9] J. Della-Dora Γ C. Dicrescenzo Γ E. Tournier Γ *An algorithm to obtain formal solutions of a linear homogenous differential equation at an irregular singular point* Γ Lecture Notes in Computer Science. In : Calmet J.(ed.) Γ Lecture Notes in Computer Science Γ (Springer Γ 1982).
- [10] R. Douady Γ *Produits tensoriels topologiques et espaces nucléaires*. Séminaire de géométrie analytique. Astérisque 16 (1974).
- [11] G. Gasper Γ M. Rahman Γ *Basic Hypergeometric Series*. Encyclopedia of Mathematics and its Applications Γ G.C. Rota ed. Γ Vol. 35 (1990).
- [12] A.-O. Gelfond Γ *Calcul des différences finies* Γ Dunod Γ Paris Γ 1963.
- [13] P. Grisvard Γ *Opérateurs à indice, Lemme de Compacité*. Séminaire Cartan-Schwartz Γ 16ème année Γ 1963-64 Γ n° 12.
- [14] W. Hahn Γ *Über die Funktional-differentialgleichung $f'(z) = f(qz)$ und verwandte Funk-*

- tionalgleichungen*. Ann. Univ.Sc. Budapest Eötvös Sect. Math. (1973) Γp.3-21.
- [15] F. Jung Γ F. Naegele Γ J. Thomann Γ *An algorithm of multisummation of formal power series, solutions of linear ODE equations* Γ à paraître dans Proceedings IMACS Lille (1993).
- [16] T. Kato Γ J.-B. McLeod Γ *The functional differential difference equation $y'(x) = ay(\lambda x) - by(x)$* . Bull. Amer. Math. Soc. Γ77 (1971) Γp. 891-937.
- [17] H. Komatsu Γ *On the index of differential operators*. J. Fac. Sci. Tokyo IA (1971) Γp.379-398.
- [18] K. Mahler Γ *On a special functionalequation*. J. London Math. Soc. 15 (1940) Γ115-123. MR 2 Γ133.
- [19] B. Malgrange Γ *Sur les points singuliers des équations différentielles*. L'Enseignement Mathématique Γ tome 20 Γ n°1-2 (1974) Γp. 147-176.
- [20] B. Malgrange Γ J.-P. Ramis Γ *Fonctions multisommables* Γ Ann. Inst. Fourier Γ Grenoble Γ **42** Γ1-2 (1992).
- [21] J. Martinet Γ J.-P. Ramis Γ *Théorie de Galois différentielle et resommation* Γ E. Tournier (ed.) Γ Computer Algebra and Differential Equations Γ117-214 Γ (Academic Press Γ1988).
- [22] J. Martinet Γ J.-P. Ramis Γ *Elementary acceleration and multisummability* Γ Ann. Inst. Henri Poincaré Γ Physique Théorique **54** Γ N.1 (1991).
- [23] F. Naegele Γ *Théorèmes d'indices pour les équations q -différences-différentielles* Γ C.R. Acad. Sci. Paris Γ t. 317 Γ Serie II Γp. 579-582 Γ1993.
- [24] F. Naegele Γ *Théorèmes d'indices pour les équations q -différences-différentielles* Γ Preprint Institut Recherche Mathématique Avancée (1995).
- [25] F. Naegele Γ J. Thomann Γ *Algorithmic approach of the multisummation of formal power series solutions of linear ODE, applied to the Stokes phenomena* Γ à paraître dans Proceedings workshop Groningen : "Stokes phenomena and their applications" (World scientific pub. comp.).
- [26] N.-E. Nörlund Γ *Leçons sur les séries d'interpolation* Γ Gauthiers Villars et Cie Γ Paris 1926.
- [27] J.-P. Ramis Γ *Déviage Gevrey*. Société Mathématique de France. Asterisque n° 59-60 (1978) Γp.173-204.
- [28] J.-P. Ramis Γ *Les séries k -sommables et leurs applications* Γ Lecture Notes in Physics **126** Γ Springer (1980).

- [29] J.-P. Ramis Γ *Théorèmes d'indices Gevrey pour les équations différentielles ordinaires*. Memoirs of the American Mathematical Society n°296 Γ volume 48 (1984).
- [30] J.-P. Ramis Γ *Equations différentielles : Phénomène de Stokes et resommation* Γ C.R. Acad. Sc. Paris Γ t.301 Γ Série 1 Γ No.4 (1985).
- [31] J.-P. Ramis Γ Y. Sibuya Γ *Hukuhara domains and fundamental existence and uniqueness theorems for asymptotic solutions of Gevrey type* Γ Asymptotic Analysis Γ **2** Γ 39-94 (1989).
- [32] J.-P. Ramis Γ *About the growth of entire functions solutions of linear algebraic q -difference equations*. Annales de la Faculté des Sciences de Toulouse. Série 6 Γ Vol. II n°1 Γ 1992.
- [33] F. Richard-Jung Γ *Représentations graphiques de solutions d'équations différentielles dans le champ complexe* Γ Thèse de l'Université de Strasbourg (1988).
- [34] F. Richard-Jung Γ *Le phénomène de Stokes en image* Γ RT 65 LMC Grenoble (1991).
- [35] J. Thomann Γ *Resommation des séries formelles solutions d'équations différentielles linéaires ordinaires du second ordre dans le champ complexe au voisinage de singularités irrégulières* Γ Numer.Math. **58** Γ 503-535 (1990).
- [36] J. Thomann Γ *Problèmes algorithmiques posés par la multisommation* Γ Journées de printemps de la PRC Math-Info (1990).
- [37] J. Thomann Γ *Procédés formels et numériques de sommation de séries solutions d'équations différentielles* Γ Expo. Math. 13 (1995) Γ 223-246.
- [38] E. Tournier Γ *Solutions formelles d'équations différentielles* Γ Thèse d'Etat de l'Université de Grenoble (1988).
- [39] K. Yosida Γ *Functional Analysis*. Berlin Γ Springer-Verlag (1971).
- [40] C. Zhang Γ *Quelques études en théorie des équations fonctionnelles et en analyse combinatoire* Γ Thèse de l'Université Louis Pasteur (Strasbourg I) Γ 1994.

ANNEXES

Annexe 1

Convergence des séries de factorielles q -analogues

Nous allons étudier la convergence de séries de factorielles q -analogues de la forme

$$\hat{f}_q(x) = \sum_{n=0}^{+\infty} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n+1)}, \quad |q| \neq 0, 1.$$

Ces séries n'ont un sens que si x est différent des valeurs $-l + \frac{2\pi i}{\log q} k$, $\forall l \in \mathbb{N}, \forall k \in \mathbb{Z}$. Nous excluons toujours dans ce qui suit ces points exceptionnels.

Les séries de q -factorielles sont des q -analogues des séries de facultés $\hat{f}(x) = \sum_{n \geq 0} \frac{a_n}{x(x+1)\dots(x+n)} = \sum_{n \geq 0} a_n \frac{\Gamma(x)}{\Gamma(x+n+1)}$.

Théorème 1 : Si une série de q -factorielles $\sum_{n \geq 0} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n+1)}$ converge en un point x_0 alors :

- (i) Si $0 < |q| < 1$, elle converge dans tout le plan;
- (ii) Si $|q| > 1$, elle converge dans le demi-plan $\Re(x \log q) > \Re(x_0 \log q)$.

Démonstration : On pose $b_n(x) = a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n+1)}$ et $c_n(x, x_0) = \frac{b_n(x)}{b_n(x_0)} = \frac{(q^{x_0}; q)_{n+1}}{(q^x; q)_{n+1}}$. Alors $\sum_{n \geq 0} b_n(x) = \sum_{n \geq 0} b_n(x_0) c_n(x, x_0)$ et la différence $c_n(x, x_0) - c_{n+1}(x, x_0)$ est égale à $c_n(x, x_0) \frac{q^{x_0+1} - q^{x+1}}{1 - q^{x+n+1}} q^n$.

On suppose que la série $\sum_{n \geq 0} b_n(x_0)$ converge.

Si $0 < |q| < 1$ alors $c_n(x, x_0)$ tend vers $\frac{(q^{x_0}; q)_{+\infty}}{(q^x; q)_{+\infty}}$ quand n tend vers $+\infty$. Il existe donc une constante $C(x, x_0) > 0$ telle que $\forall n \in \mathbb{N}$, $|c_n(x, x_0) - c_{n+1}(x, x_0)| \leq C(x, x_0) |q|^n$ et la série $\sum_{n \geq 0} |c_n(x, x_0) - c_{n+1}(x, x_0)|$ converge. En utilisant le test de Bois-Reymond (voir par exemple le théorème I p.126 [1]) la série $\sum_{n \geq 0} b_n(x)$ converge.

Si $|q| > 1$ $c_n(x, x_0) = q^{(n+1)(x_0-x)} \frac{(p^{x_0}; p)_{n+1}}{(p^x; p)_{n+1}}$ ($p = q^{-1}$) et dans ce cas il existe une constante $C(x, x_0)$ telle que $\forall n \in \mathbb{N}$, $|c_n(x, x_0) - c_{n+1}(x, x_0)| \leq C(x, x_0) |q^{x_0-x}|^n$. Si $|q^{x_0-x}| < 1$ la série $\sum_{n \geq 0} b_n(x)$ converge. La série converge donc dans le demi-plan limité à gauche par la droite passant par x_0 de pente $\frac{\log |q|}{\bar{q}}$ ($q = |q| e^{i\bar{q}} 0 \leq \bar{q} < 2\pi$; la droite est parallèle à l'axe des imaginaires purs si $\bar{q} = 0$).

Nous pouvons définir l'abscisse de convergence d'une série de q -factorielles $\hat{f}_q(x)$ par

$$\sigma = \inf \{ \sigma_0 \in \mathbb{R} / \hat{f}_q(x) \text{ converge dans le demi-plan } \Re(x \log q) > \sigma_0 \log |q| \}.$$

Si $0 < |q| < 1$ $\Gamma\sigma$ est égale soit à $-\infty$ (la série converge dans tout le plan) Γ soit à $+\infty$ (elle diverge partout).

N.-E. Nörlund montre que l'abscisse de convergence de la série de facultés $\sum_{n \geq 0} a_n \frac{\Gamma(x)}{\Gamma(x+n+1)}$ est la même que celle de la série de Newton associée $\sum_{n \geq 0} \frac{(-1)^n a_n}{(\Gamma(n+1))^2} \frac{\Gamma(x)}{\Gamma(x-n)}$ (paragraphe 84 p.177 [2]).

Considérons les séries de Newton q -analogues de la forme

$$\hat{g}_q(x) = \sum_{n \geq 0} \frac{a_n}{\Gamma_q(n+1)} \frac{\Gamma_q(x)}{\Gamma_q(x-n)}.$$

Elles ont été étudiées par C. Zhang [3]. Pour q réel $0 < q < 1$ il a montré que le domaine de convergence de ces séries est un demi-plan.

Théorème 2 : Soit $\hat{f}_q(x) = \sum_{n \geq 0} a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n+1)}$ une série de q -factorielles ($|q| \neq 0, 1$). Considérons la série de Newton p -analogue associée

$$\hat{g}_p(x) = \sum_{n \geq 0} \frac{(-1)^n p^{n^2}}{(\Gamma_p(n+1))^2} a_n \frac{\Gamma_p(x)}{\Gamma_p(x-n)}, \quad (p = q^{-1}).$$

Ces deux séries sont simultanément convergentes ou divergentes.

Démonstration : Nous allons une fois de plus utiliser le test de Bois-Reymond. Posons $b_n(x) = a_n \frac{\Gamma_q(x)}{\Gamma_q(x+n+1)}$ et $c_n(x) = \frac{(-1)^n p^{n^2}}{(\Gamma_p(n+1))^2} \frac{\Gamma_p(x)}{\Gamma_p(x-n)} \frac{\Gamma_q(x+n+1)}{\Gamma_q(x)}$. Posons $\beta_n(x) = \frac{(-1)^n p^{n^2}}{(\Gamma_p(n+1))^2} a_n \frac{\Gamma_p(x)}{\Gamma_p(x-n)}$ et $\delta_n(x) = \frac{1}{c_n(x)}$. Alors $b_n(x)c_n(x) = \beta_n(x)$ et $\beta_n(x)\delta_n(x) = b_n(x)$.

D'autre part $\Gamma c_n(x) - c_{n+1}(x) = c_n(x) \frac{-2+q^x+q^{-x}}{(1-q^{n+1})^2} q^{n+1}$ et $\delta_n(x) - \delta_{n+1}(x) = \frac{q^{n+1}}{c_n(x)} \frac{2-q^x-q^{-x}}{(1-q^{-x+n+1})(1-q^{x+n+1})}$.

Si $0 < |q| < 1$ $\Gamma c_n(x) = \frac{(q^{-x+1}; q)_n (q^x; q)_{n+1}}{((q; q)_n)^2 (1-q)}$ tend vers une constante et il existe $C(x) > 0$ telle que $\forall n \in \mathbb{N} \Gamma |c_n(x) - c_{n+1}(x)| \leq C(x) |q|^n$ et $|\delta_n(x) - \delta_{n+1}(x)| \leq C(x) |q|^n$. Les séries $\sum_{n \geq 0} |c_n(x) - c_{n+1}(x)|$ et $\sum_{n \geq 0} |\delta_n(x) - \delta_{n+1}(x)|$ convergent.

Si $|q| > 1$ $\Gamma c_n(x) = -\frac{q^x}{((p; p)_n)^2 (1-q)} (p^{-x+1}; p)_n (p^x; p)_{n+1}$ et il existe une constante $C(x) > 0$ telle que $\forall n \in \mathbb{N} \Gamma |c_n(x) - c_{n+1}(x)| \leq C(x) |q|^{-n}$ et $|\delta_n(x) - \delta_{n+1}(x)| \leq C(x) |q|^{-n}$. Les séries $\sum_{n \geq 0} |c_n(x) - c_{n+1}(x)|$ et $\sum_{n \geq 0} |\delta_n(x) - \delta_{n+1}(x)|$ convergent.

Alors Γ si $\sum_{n \geq 0} b_n(x) = \hat{f}_q(x)$ converge $\Gamma \sum_{n \geq 0} b_n(x)c_n(x) = \hat{g}_p(x)$ converge et si $\sum_{n \geq 0} \beta_n(x) = \hat{g}_p(x)$ converge $\Gamma \sum_{n \geq 0} \beta_n(x)\delta_n(x) = \hat{f}_q(x)$ converge.

Théorème 3 : Si $\sum_{n \geq 0} a_n \frac{\Gamma_q(x)}{\Gamma_q(x-n)}$ est une série de Newton q -analogue avec $|q| > 1$, soit elle converge dans tout le plan, soit elle diverge partout.

Pour q réel $0 < q < 1$ C. Zhang a calculé l'abscisse de convergence σ d'une série de Newton q -analogue $\hat{g}_q(x)$. Plus précisément si $\hat{g}_q(x) = \sum_{n \geq 0} \frac{a_n}{\Gamma_q(n+1)} \frac{\Gamma_q(x)}{\Gamma_q(x-n)}$ alors

$$(i) \quad \text{Si } \sigma \geq 0 \quad \sigma = \overline{\lim}_{n \rightarrow \infty} \frac{\log \left| \sum_{k=0}^n (-1)^k a_k q^{-\frac{k(k+1)}{2}} \right|}{n \log q^{-1}};$$

$$(ii) \quad \text{Si } \sigma < 0 \quad \sigma = \overline{\lim}_{n \rightarrow \infty} \frac{\log \left| \sum_{k=n}^{+\infty} (-1)^k a_k q^{-\frac{k(k+1)}{2}} \right|}{n \log q^{-1}}.$$

Ce résultat se généralise à q complexe $0 < |q| < 1$ en définissant l'abscisse de convergence d'une série de Newton q -analogue par $\inf \{ \sigma_0 \in \mathbb{R} / \text{la série converge pour } \Re(x \log q) > \sigma_0 \log |q| \}$.

Théorème 4 : Soit $\hat{f}_q(x) = \sum_{n \geq 0} a_n \Gamma_q(n+1) \frac{\Gamma_q(x)}{\Gamma_q(x+n+1)}$ une série de q -factorielles avec $|q| > 1$. Soit σ son abscisse de convergence.

$$(i) \quad \text{Si } \sigma \geq 0 \text{ alors } \sigma = \overline{\lim}_{n \rightarrow \infty} \frac{\log \left| \sum_{k=0}^n a_k \right|}{n \log |q|};$$

$$(ii) \quad \text{Si } \sigma < 0 \text{ alors } \sigma = \overline{\lim}_{n \rightarrow \infty} \frac{\log \left| \sum_{k=n}^{+\infty} a_k \right|}{n \log |q|}.$$

Démonstration : D'après le théorème 2 l'abscisse de convergence de $\hat{f}_q(x)$ est égale à l'abscisse de convergence de la série $\hat{g}_p(x) = \sum_{n \geq 0} \frac{(-1)^n p^{\frac{n(n+1)}{2}}}{(\Gamma_p(n+1))^2} a_n \Gamma_q(n+1) \frac{\Gamma_p(x)}{\Gamma_p(x-n)} = \sum_{n \geq 0} \frac{(-1)^n p^{\frac{n(n+1)}{2}}}{\Gamma_p(n+1)} a_n \frac{\Gamma_p(x)}{\Gamma_p(x-n)}$. On utilise alors le résultat de C. Zhang.

Références bibliographiques

- [1] A.-O. Gelfond *Calcul des différences finies* Dunod Paris 1963.
- [2] N.-E. Nörlund *Leçons sur les séries d'interpolation* Gauthiers Villars et Cie Paris 1926.
- [3] C. Zhang *Quelques études en théorie des équations fonctionnelles et en analyse combinatoire* Thèse de l'Université Louis Pasteur (Strasbourg I) 1994.

Annexe 2

Fichiers "source" de l'algorithme de Balser

```
----- autoder.as -----

#include "sumit.as"
#library ci "ci.ao"
import from ci;

SUP ==> SparseUnivariatePolynomial;

AutoDer(S: BalserRing) : with {

  sigma: () -> Automorphism(SUP(S,"j"));
  ++ Automorphisme associé à l'opérateur usuel de translation.
  D: () -> Derivation SUP(S,"x");
  ++ Dérivation associée à la dérivée usuelle.
  Theta: () -> Derivation SUP(S,"x");
  ++ Dérivation associée à la dérivée d'Euler.
}

== add {

  import from Integer;
  import from Derivation SUP(S,"x");
  import from Automorphism SUP(S,"j");

  fsigma(pol: SUP(S,"j"), n: Integer) : SUP(S,"j") == {
    res: SUP(S,"j") := 0;
    for l in 0..degree(pol) repeat {
      aux : SUP(S,"j") := 1;
      for i in 1..l repeat {
        aux:=aux*(monomial(coerce(n)$S,0)+monomial(1$S,1))
      }
      res := res+monomial(coeffcient(pol,l),0) * aux
    }
    res
  }

  +++  $\sigma$  définit l'automorphisme:  $P(j) \rightarrow P(j+1)$ .
```

```
sigma():Automorphism SUP(S,"j") == morphism(fsigma);
```

```
fTheta(pol: SUP(S,"x")): SUP(S,"x") == {
  import from S;
  res: SUP(S,"x") := 0;
  for l in 1..degree(pol) repeat {
    res := res + monomial(coerce(l)$S*coefficient(pol,l), l)
  }
  res
}
```

```
+++  $\Theta$  définit l'opérateur d'Euler:  $P(x) \rightarrow xP'(x)$ .
Theta():Derivation SUP(S,"x") == derivation fTheta;
```

```
fD(pol: SUP(S,"x")): SUP(S,"x") == {
  import from S;
  res: SUP(S,"x") := 0;
  for l in 1..degree(pol) repeat {
    res := res + monomial(coerce(l)$S*coefficient(pol,l), l-1)
  }
  res
}
```

```
+++  $D$  définit la dérivation usuelle:  $P(x) \rightarrow P'(x)$ .
D():Derivation SUP(S,"x") == derivation fD;
```

```
}
```

----- balser.as -----

```
#include "sumit.as"

#library path "path.ao";
#library lagleg "lagleg.ao"
#library differential "differential.ao"
#library borel "borel.ao"
#library laplace "laplace.ao"
#library runge "runge.ao"

import from complexring, path, link, geolink, method;
import from autoder,ci;
import from poldelto,Powerseries,borel;
import from polodo,differential,runge;
import from lagleg,laplace;

SUP ==> SparseUnivariatePolynomial;
POLDELTO ==> LinearOrdinaryPolynomialDifferenceOperator;
POLODO ==> LinearOrdinaryPolynomialDifferentialOperator;
LOEDO ==> EulerLinearOrdinaryPolynomialDifferentialOperator;
SI ==> SingleInteger;
Z ==> Integer;
F ==> DoubleFloat;
CF ==> Complex F;

Balser(S: BalserRing): with {

  balser: (Powerseries(S), List Ratio SI, CF) -> CF;
  ++ Si  $\hat{f}$  est une série formelle  $(k_1, \dots, k_r)$ -sommable,
  ++  $balser(\hat{f}, [k_1, \dots, k_r], zr)$  calcule la somme en  $zr$ .
  balser: (Powerseries(S), List Ratio SI, CF, Z) -> List CF;
  ++ Si  $\hat{f}$  est une série formelle  $(k_1, \dots, k_r)$ -sommable,
  ++  $balser(\hat{f}, [k_1, \dots, k_r], zr, n)$  calcule la somme en  $zr$ ,
  ++ ainsi que toutes ses dérivées jusqu'à l'ordre  $n$  et retourne
  ++ la liste  $[f(zr), f'(zr), \dots, f^{(n)}(zr)]$ .
  balserpath: (Powerseries(S), List Ratio SI, Path(S))
    -> Record(points : List CF , value : List CF);
  ++ Si  $\hat{f}$  est une série formelle  $(k_1, \dots, k_r)$ -sommable,
  ++  $balserpath(\hat{f}, [k_1, \dots, k_r], ch)$  calcule la somme le long du chemin  $ch$ .
  iLbalser: (Powerseries(S), List Ratio SI, CF) -> CF;
  ++ Si  $\hat{f}$  est une série formelle  $(\kappa_1, \dots, \kappa_r) - iL$  sommable,
  ++  $balser(\hat{f}, [\kappa_1, \dots, \kappa_r], zr)$  calcule la somme en  $zr$ .
```

```

iLbalser: (Powerseries(S), List Ratio SI, CF, Z) -> List CF;
++ Si  $\hat{f}$  est une série formelle  $(\kappa_1, \dots, \kappa_r) - iL$  sommable,
++ balser( $\hat{f}$ ,  $[\kappa_1, \dots, \kappa_r]$ ,  $zr$ ,  $n$ ) calcule la somme en  $zr$ , ainsi que
++ toutes ses dérivées jusqu'à l'ordre  $n$  et retourne la liste
++  $[f(zr), f'(zr), \dots, f^{(n)}(zr)]$ .
iLbalserpath: (Powerseries(S), List Ratio SI, Path(S))
--> Record(points : List CF , value : List CF);
++ Si  $\hat{f}$  est une série formelle  $(\kappa_1, \dots, \kappa_r) - iL$  sommable,
++ iLbalserpath( $\hat{f}$ ,  $[\kappa_1, \dots, \kappa_r]$ ,  $ch$ ) calcule la somme
++ le long du chemin  $ch$ .
analyticpath: (Powerseries(S), Path(S))
--> Record(points : List CF , value : List CF);
++ Si  $\hat{f}$  est une série formelle convergente,
++ analyticpath( $\hat{f}$ ,  $ch$ ) calcule le prolongement analytique de la
++ somme le long du chemin  $ch$  par la méthode de Runge-Kutta.
}

== add {

+++ Calcule la somme en  $zr$  d'une série formelle  $(k_1, \dots, k_r)$ -sommable.
balser(ps : Powerseries(S) , sum : List Ratio SI , zr : CF ) : CF == {
  import from Z;
  import from List CF;
  bal := balser(ps, sum, zr, 0);
  first(bal)
}

+++ Calcule la somme en  $zr$  ainsi que les dérivées jusqu'à l'ordre
+++  $n$  d'une série  $(k_1, \dots, k_r)$ -sommable.
balser(ps:Powerseries(S), sum:List Ratio SI , zr:CF , n:Z): List CF == {
  import from Borel(S);
  kappasum := kappasummability(sum);
  iLbalser(ps, kappasum, zr, n)
}

+++ Calcule la somme en  $zr$  ainsi que les dérivées jusqu'à l'ordre
+++  $n$  d'une série  $(\kappa_1, \dots, \kappa_r) - iL$  sommable.
iLbalser(ps:Powerseries(S), kappasum:List Ratio SI, zr:CF):CF == {
  import from Z;
  import from List CF;
  bal := iLbalser(ps, kappasum, zr, 0);
  first(bal)
}

```

```

+++ Calcule la somme en  $zr$  ainsi que les dérivées jusqu'à l'ordre
+++  $n$  d'une série  $(\kappa_1, \dots, \kappa_r) - iL$  sommable.
iLbalser(ps: Powerseries(S), kappasum: List Ratio SI, zr: CF, n: Z): List(CF) == {
  import from Laplace(S);
  (#kappasum=0) => error("You should not use Balser's algorithm in the
  convergent case");
  borellist := boreltransform(ps, kappasum);
  a : SI := numer(kappasum.1);
  b : SI := denom(kappasum.1);
  derivee(borellist, a, b, zr, n)
}

+++ Si  $\hat{f} = \sum_{n \geq 0} a_n x^n$  est une série  $(\kappa_1, \dots, \kappa_r) - iL$  sommable avec
+++  $\kappa_1 = \alpha_1 / \beta_1$ ,  $boreltransform(\hat{f}, [\kappa_1, \dots, \kappa_r])$  scinde  $\hat{f}$  en  $\alpha_1$  sous-séries,
+++ puis calcule leur transformée de Borel d'ordre  $1/\beta_1$  le long d'un
+++ chemin demandé à l'utilisateur; initerms représente la liste
+++  $[a_0, \dots, a_{\alpha_1-1}]$  et listBorel les  $\alpha_1$  transformées de Borel.
boreltransform(ps: Powerseries(S), kappasum: List Ratio SI):
  Record( listBorel: List Record(vi: List(CF), phiqvi: List(CF), wi: List(CF)),
    initerms: List Quotient(S) ) == {

  import from Borel(S);
  import from Partial Quotient(S);
  import from POLDELTO(S);
  import from Laplace(S);
  import from SUP(S, "j");
  import from S, Z;

  a : SI := numer(kappasum.1);
  b : SI := denom(kappasum.1);
  listBorel: List Record(vi: List CF, phiqvi: List CF, wi: List CF) := nil;
  initerms: List Quotient(S) := nil;
  subseries : List Powerseries(S) := split(ps, a);
  kappa := [numer(kappasum.1)/(denom(kappasum.1)*a) for 1 in 2..#kappasum];

  for q in 1..a repeat {
    print << "On traite la sous-série " << q;
    print << newline;
    print << kappa << newline;
    fq := subseries.q;
    print << fq << newline;
    phiq := borel(fq, b);
    print << "Transformée de borel:";
    print << phiq;
  }
}

```

```

    print << newline;
    print << newline;
    ch := readpath();
    if (#kappasum=1) then res := analytic(phiq,ch)
    else res := balser1(phiq,kappa,ch);
    listBorel := concat!( listBorel, list res);
    initerms := concat!( initerms, list coerce(coefficients(fq,0@Z)))
  }
[listBorel, initerms]
}

```

+++ *analytic(\hat{f} ,ch)* calcule le prolongement de la somme de \hat{f} le long du
 +++ chemin *ch*, en les abscisses de quadrature de Gauss-Laguerre pour le
 +++ dernier maillon du chemin, de Gauss-Legendre ailleurs. La série
 +++ est supposée convergente.

```

analytic( phiq : Powerseries(S), ch: Path(S)):
  Record(vi:List CF, phiqvi:List CF, wi:List CF) == {

    import from Differential(S),Laplace(S);
    import from Record(eqdiff: POLODO(S),sec:SUP(S,"x"));
    import from Integer, SingleInteger,S;
    import from Path(S),List Link(S),GeoLink;
    import from Record(vi:List CF,phiqvi:List CF,wi:List CF,ycond:List CF);

    diffeq := diffequation(phiq);
    differentialeq : POLODO(S) := diffeq.eqdiff;
    r: Integer := degree(differentialeq);
    sec: SUP(S,"x") := diffeq.sec;
    if (r = 0) then {
      r := 1;
      deriv(differentialeq, sec)
    }

    listcoef : Partial List Quotient(S) := coefficients(phiq, r+1);
    failed? listcoeff ==> error("You should give more initial conditions");
    coeff : List Quotient(S) := coerce(listcoef);
    ycond : List CF := nil;
    for i in 0..r-1 repeat {
      coef := first coeff;
      ycond:=concat!(ycond, list(coerce(factorial(i)*numerator(coef))$S
                                /coerce(denominator(coef))));
      coeff := rest coeff
    }
    coef := first coeff;
  }

```

```

znieme : CF := coerce(factorial r * numerator(coef))$S
              /coerce(denominator(coef));

rch: List Link(S) := coerce(ch);
nbpath := #(rch);
weightlist : List CF := nil;
vilist : List CF := nil;
phiqvilist : List CF := nil;
for l in 1..nbpath repeat {
  gl : GeoLink:=coerce(rch.l);
  aux := analyticlocal(gl,nbpath-l,ycond, znieme,differentialeq,sec);
  vilist := concat!(vilist,aux.vi);
  phiqvilist := concat!(phiqvilist,aux.phiqvi);
  weightlist := concat!(weightlist,aux.wi);
  if (l < nbpath) then ycond := aux.ycond
}
[vilist,phiqvilist,weightlist] ;
}

+++ Prolongement sur un chemin élémentaire.
analyticlocal( gl:GeoLink, l:SingleInteger, ycond:List CF, znieme:CF,
              differentialeq:POLODO(S), sec:SUP(S,"x")):
Record(vi:List CF, phiqvi:List CF, wi:List CF, ycond:List CF) == {

import from Record(np: SI, gamma: F -> CF, gammaderiv: F -> CF);
import from S;
import from LagLeg;
import from Record(absci: List F, weight: List F);
import from Runge_Kutta(S);
import from F;

rgl := coerce(gl);
gamma := rgl.gamma;
gammaderiv := rgl.gammaderiv;
derivs := generateDerivs(differentialeq, sec,gamma,gammaderiv,znieme);
if (l=0) then {
  lag := laguerre(gl);
  abscisse := lag.absci;
  weight := lag.weight
}
else {
  leg := legendre(gl);
  abscisse := leg.absci;
  weight := leg.weight
}
}

```



```

    }
tbegin: F := 0;
nbpoint := #(abscisse);
weightlocal : List CF := nil;
vilocal : List CF := nil;
phiqlocal : List CF := nil;
ycondlocal: List CF := nil;

for i in 1..nbpoint repeat {
    ti := abscisse.i;
    vi := gamma(ti);
    wi := weight.i;
    ycond := rk4f(ycond, tbegin, ti, 10@SI, derivs);
    yi := first ycond;
    vilocal := concat!(vilocal, list vi);
    weightlocal := concat!(weightlocal, list(wi * gammaderiv(ti)));
    phiqlocal := concat!(phiqlocal, list yi);
    print << "x= "<<$CF vi << newline;
    print << "y= "<<$CF yi << newline;
    tbegin := ti
}
if (l = 0) then ycondlocal := rk4f(ycond,tbegin, 1., 10@SI, derivs);
[vilocal,phiqlocal,weightlocal,ycondlocal]
}

+++ Si  $\hat{\phi}$  est une série  $(\kappa_1, \dots, \kappa_r) - iL$  sommable, balser1( $\hat{\phi}, [\kappa_1, \dots, \kappa_r], ch$ )
+++ calcule la somme de  $\hat{\phi}$  le long du chemin ch, en les abscisses de
+++ Gauss-Laguerre sur le dernier maillon, de Gauss-Legendre ailleurs.
balser1(phiq:Powerseries(S) , kappasum : List Ratio SI, ch : Path(S)):
Record(vi:List CF, phiqvi:List CF, wi:List CF) == {

import from Differential(S);
import from Record(eqdiff: POLODO(S), sec: SUP(S,"x"));
import from Laplace(S);
import from List Link(S);
import from LagLeg;
import from Record(absci: List F, weight: List F);
import from Record(
    listBorel:List Record(vi:List(CF),phiqvi:List(CF),wi:List(CF)),
    initerms:List Quotient(S));
import from Runge_Kutta(S);

diffeq := diffequation(phiq);
differentialeq := diffeq.eqdiff;

```

```

r: Integer := degree(differentialeq);
sec := diffeq.sec;
if (r = 0) then {
    r := 1;
    deriv(differentialeq, sec)
}
rch : List Link(S) := coerce(ch);
weightlist : List CF := nil;
vilist : List CF := nil;
phiqvilist : List CF := nil;

for l in 1..#rch repeat {
    gl : GeoLink := coerce(rch.l);
    rgl: Record(np:SI, gamma:F->CF, gammaderiv:F->CF):=coerce(gl);
    gamma: F -> CF := rgl.gamma;
    gammaderiv: F -> CF := rgl.gammaderiv;
    if (l=#rch) then {
        lag := laguerre(gl);
        abscisse := lag.absci;
        weight := lag.weight
    }
    else {
        leg := legendre(gl);
        abscisse := leg.absci;
        weight := leg.weight
    }

    mch: Method(S) := coerce(rch.l);
    if typeSerie?(mch) then {
        a : SI := numer(kappasum.1);
        b : SI := denom(kappasum.1);
        borellist := boreltransform(phiq, kappasum);
        for i in 1..#abscisse repeat {
            ti: F := abscisse.i;
            vi: CF := gamma(ti);
            yi := first derivee(borellist,a,b,vi,0);
            wi: CF := weight.i * gammaderiv(ti);
            vilist := concat!(vilist, list vi);
            weightlist := concat!(weightlist, list wi);
            phiqvilist := concat!(phiqvilist, list yi);
            print << "x= "<<$CF vi << newline;
            print << "y= "<<$CF yi << newline
        }
        ycond := derivee(borellist,a,b,gamma(1),r-1)
    }
}

```

```

    }
  else {
    tbegin: F := 0.;
    derivs:=generateDerivs(differentialeq,sec,gamma,gammaderiv,0);
    v0 := beginGeoLink(gl);

    for i in 1..#abscisse repeat {
      ti: F := abscisse.i;
      vi: CF := gamma(ti);
      wi: CF := weight.i * gammaderiv(ti);
      ycond := rk4f(ycond, tbegin, ti,10@SI, derivs);
      yi := first ycond;
      vilist := concat!(vilist, list vi);
      weightlist := concat!(weightlist, list wi);
      phiqvilst := concat!(phiqvilst,list yi);
      tbegin := ti;
      print << "x= "<<$CF vi << newline;
      print << "y= "<<$CF yi << newline
    }

    if l < #rch then ycond := rk4f(ycond,tbegin,1.,10@SI,derivs)
  }
}
[vilist,phiqvilst,weightlist]
}

deriv(differentialeq: POLODO(S), sec: SUP(S,"x")):() == {
  import from AutoDer(S);
  import from Derivation SUP(S,"x");
  import from Z;

  a0: SUP(S,"x") := coefficient(differentialeq, 0);
  a0deriv: SUP(S,"x") := function(D()) a0;
  differentialeq := monomial(a0deriv, 0) + monomial(a0,1);
  sec:= function(D()) sec;
  ()
}

+++ Calcul de la somme d'une série multisommable le long du chemin ch,
+++ les points en lesquels est calculée la somme étant équidistants
+++ sur chaque chemin élémentaire.
balserpath(ps:Powerseries(S), sum:List Ratio SI, ch:Path(S)):
  Record(points:List CF, value:List CF) == {
    import from Borel(S);

```

```

kappasum := kappasummability(sum);
iLbalserpath(ps, kappasum, ch)
}

+++ Calcul de la somme d'une série  $iL$ -multisommable le long du chemin
+++  $ch$ , les points en lesquels est calculée la somme étant
+++ équidistants sur chaque chemin élémentaire.
iLbalserpath(ps:Powerseries(S), kappasum>List Ratio SI, ch:Path(S)):
  Record(points>List CF, value>List CF) == {

import from Differential(S);
import from Record(eqdiff: POLODO(S), sec: SUP(S,"x"));
import from Laplace(S),LagLeg;
import from List Link(S);
import from Runge_Kutta(S);
import from Record(
  listBorel>List Record(vi>List(CF),phiqvi>List(CF),wi>List(CF)),
  initerms>List Quotient(S));

diffeq := diffequation(ps);
differentialeq := diffeq.eqdiff;
r: Integer := degree(differentialeq);
sec := diffeq.sec;
if (r = 0) then {
  r := 1;
  deriv(differentialeq, sec)
}
rch : List Link(S) := coerce(ch);
abs: List CF := nil;
val : List CF := nil;

for l in 1..#rch repeat {
  gl : GeoLink := coerce(rch.l);
  rgl: Record(np:SI, gamma:F->CF, gammaderiv:F->CF) := coerce(gl);
  gamma: F -> CF := rgl.gamma;
  gammaderiv: F -> CF := rgl.gammaderiv;
  n := rgl.np;
  nF: F := coerce(n);
  abscisse : List F := [coerce(i)/nF for i in 1..n];

  mch: Method(S) := coerce(rch.l);
  if typeSerie?(mch) then {
    a : SI := numer(kappasum.1);
    b : SI := denom(kappasum.1);
  }
}

```

```

        borellist := boreltransform(ps, kappasum);
        for i in 1..#abscisse repeat {
            ti: F := abscisse.i;
            vi: CF := gamma(ti);
            abs := concat!(abs, list vi);
            val := concat!(val, derivee(borellist,a,b,vi,0))
        }
        ycond := derivee(borellist,a,b,gamma(1),r-1)
    }
else {
    tbegin: F := 0.;
    derivs := generateDerivs(differentialeq, sec, gamma, gammaderiv,0);
    v0 := beginGeoLink(gl);
    for i in 1..#abscisse repeat {
        ti: F := abscisse.i;
        vi: CF := gamma(ti);
        abs := concat!(abs, list vi);
        ycond := rk4f(ycond, tbegin, ti,10@SI, derivs);
        val := concat!(val,list ycond(1));
        tbegin := ti
    }
}
}
[abs,val]
}

```

+++ Calcul de la somme d'une série convergente le long du chemin *ch*.

+++ Les points en lesquels est calculée la somme sont équidistants

+++ sur chaque chemin élémentaire.

analyticpath(ps: Powerseries(S), ch: Path(S)):

```

    Record(points:List CF, value:List CF) == {

```

```

        import from Differential(S),Laplace(S);

```

```

        import from Record(eqdiff: POLODO(S),sec:SUP(S,"x"));

```

```

        import from Integer, SingleInteger,S;

```

```

        import from Path(S),List Link(S),GeoLink;

```

```

        import from Record(vi : List CF , phiqvi : List CF ,ycond:List CF);

```

```

        diffeq := diffequation(ps);

```

```

        differentialeq : POLODO(S) := diffeq.eqdiff;

```

```

        r: Integer := degree(differentialeq);

```

```

        sec: SUP(S,"x") := diffeq.sec;

```

```

        if (r = 0) then {

```

```

            r := 1;

```

```

    deriv(differentialeq, sec)
  }

ycond : List CF := nil;
for i in 0..r-1 repeat {
  coeff : Partial Quotient(S) := coefficient(ps,i);
  failed? coeff ==> error("You should give more initial conditions");
  coefff : Quotient(S) := coerce(coeff);
  ycond:=concat!(ycond, list(coerce(factorial(i)*numerator(coefff))$S ))
    /coerce(denominator(coefff))
}
coeff : Partial Quotient(S) := coefficient(ps,r);
failed? coeff ==> error("You should give more initial conditions");
coefff : Quotient(S) := coerce(coeff);
znieme : CF := coerce(factorial r * numerator(coefff))$S ;
    /coerce(denominator(coefff))

rch: List Link(S) := coerce(ch);
nbpath := #(rch);
abs : List CF := nil;
val : List CF := nil;

for l in 1..nbpath repeat {
  gl : GeoLink:=coerce(rch.l);
  aux := analyticloc(gl,nbpath-l,ycond, znieme,differentialeq,sec);
  abs := concat!(abs,aux.vi);
  val := concat!(val,aux.phiqvi);
  if (l < nbpath) then ycond := aux.ycond
}
[abs,val] ;
}

analyticloc( gl:GeoLink, l:SingleInteger, ycond:List CF, znieme:CF, ):
    differentialeq:POLODO(S), sec:SUP(S,"x")
Record(vi:List CF, phiqvi:List CF,ycond:List CF) == {

import from Record(np: SI, gamma: F -> CF, gammaderiv: F -> CF);
import from S;
import from LagLeg;
import from Runge_Kutta(S);
import from F;

rgl := coerce(gl);
gamma := rgl.gamma;

```

```

gammaderiv := rgl.gammaderiv;
n := rgl.np;
nF: F := coerce(n);
abscisse : List F := [coerce(i)/nF for i in 1..n];
derivs := generateDerivs(differentialeq, sec,gamma,gammaderiv,znieme);
tbegin: F := 0;
vilocal : List CF := nil;
phiqlocal : List CF := nil;
ycondlocal: List CF := nil;
for i in 1..n repeat {
    ti := abscisse.i;
    ycond := rk4f(ycond, tbegin, ti, 10@SI, derivs);
    vilocal := concat!(vilocal,list gamma(ti));
    phiqlocal := concat!(phiqlocal, list ycond.1);
    tbegin := ti
}
ycondlocal := ycond;
[vilocal,phiqlocal,ycondlocal]
}
}

```

```

----- borel.as -----

#include "sumit.as"
#library powerseries "powerseries.ao"

import from powerseries;
import from poldelto;
import from autoder, ci;

SUP ==> SparseUnivariatePolynomial;
POLDELTO ==> LinearOrdinaryPolynomialDifferenceOperator;
SI ==> SingleInteger;
Z ==> Integer;

Borel(S: BalserRing): with {

  borel: (PowerSeries(S), SI) -> PowerSeries(S);
  ++ Calcule la transformée de Borel d'une série formelle.
  kappasummability : List Ratio SI -> List Ratio SI;
  ++ Définit les ordres des transformées de Borel-Laplace.
  split: (PowerSeries(S), SI) -> List PowerSeries(S);
  ++ Scinde une série formelle.
  split: (POLDELTO(S), SI) -> POLDELTO(S);
  ++ Scinde un opérateur aux différences.
}

== add {

  +++ Retourne la liste  $[\kappa_1, \dots, \kappa_p]$ .
  kappasummability(sum : List Ratio SI ) : List Ratio SI == {
    ASSERT (#sum >=1);
    kappasum : List Ratio SI := list (sum.1);
    #sum = 1 => kappasum;
    for i in 2..#sum repeat {
      kappasum := concat!(kappasum, list(1/(1/sum.i-1/sum.(i-1))))
    }
    kappasum
  }

  gamma(l: SI): S == {
    res: SI := 1;
    for i in 2..l repeat res := res*(i-1);
    coerce(res)
  }
}

```



```

}

facteur(l:Integer,b:Integer): SUP(S,"j") == {
  import from SUP(S,"j");
  import from S;

  (l = 0) => 1;
  res: SUP(S,"j") := 1;
  for i in 1..l*b repeat {
    fact : SUP(S,"j") := monomial(coerce(b)$S,1);
    fact := fact+coerce(coerce(b+i-1)$S)$SUP(S,"j");
    res := res*fact
  }
  res
}

+++ Calcule la pseudo transformée de Borel formelle d'ordre  $1/b$  d'une série
+++ formelle.
borel(ps: PowerSeries(S), b: SI): PowerSeries(S) == {

  import from Partial List Quotient(S),S,SI,Z;
  import from List Quotient(S);
  import from SUP(S,"j");
  import from Automorphism SUP(S,"j");

  ASSERT (b > 0);
  cond := initcond(ps);
  m : SI := #cond;
  delta : POLDELTO(S) := delto(ps);
  r := degree(delta);
  condd:Partial List Quotient(S):=coefficients(ps,m+1)$PowerSeries(S);
  failed? condd => error( "Impossible to calculate the next coefficient
                        in order to define the Borel transform");
  condaux : List Quotient(S) := rest(coerce(condd));
  condbo: List Quotient(S) := nil;
  for l in 1..m repeat {
    condbo:=concat!(condbo, list(numerator(first condaux)
                                /(denominator(first condaux)*gamma(b*l))));
    condaux := rest condaux
  }
  deltabo: POLDELTO(S) := 0;
  for k in 0..r repeat {
    mon: SUP(S,"j") := apply(sigma())$AutoDer(S),coefficient(delta,k));
    mon := mon * facteur(k,coerce b);

```

```

        deltabo := deltabo + monomial(mon,k)
    }
bor:  PowerSeries(S) := createPowerSeries(condbo,deltabo);
bor
}

apply(delt:  POLDELTO(S), l:  Integer):  POLDELTO(S) == {
    import from Automorphism(SUP(S,"j"));
    import from AutoDer(S);
    res :  POLDELTO(S) := 0;
    for i in 0..degree(delt) repeat {
        res := res + monomial(apply(sigma()̂l, coefficient(delt,i)), i+1)
    }
    res
}

inconnues( delt:  POLDELTO(S), a:  SingleInteger):  Partial Integer == {
    import from Integer;
    inc :  Boolean := false;
    l :  Integer := 1;
    while inc and (l <= degree delt) repeat {
        coef :  SUP(S,"j") := coefficient(delt,l);
        if coef = 0 then l := l+1
        else {
            l rem coerce a = 0 => inc := true;
            l := l + 1
        }
    }
    inc => failed;
    coerce(l)
}

+++ Calcule un multiple d'un opérateur aux différences tel que les
+++ coefficients des termes  $\sigma^k$  avec  $k$  non congru à  $a$  soient nuls.
split(delt:  POLDELTO(S), a:SingleInteger) :  POLDELTO(S) == {
    import from SUP(S,"j");
    import from Integer;

    (a = 1) => delt;
    r := degree(delt);
    (r = 0) => delt;
    ai :  Integer := coerce a;
    equations :  List POLDELTO(S) := list delt;
    for i in 1..r*(ai-1) repeat {

```

```

        equations := concat!(equations,list apply(delt,i))
    }
fini : Boolean := false;
while  fini repeat {
    pol := first equations;
    ind : Partial Integer := inconnues(pol, a);
    if failed? ind then fini := true
    else {
        l : Integer := coerce(ind);
        Pl := coefficient(pol,l);
        eqaux := rest equations;
        equations := nil;
        for i in 1..#eqaux repeat {
            delti := eqaux.i;
            if coefficient(delti,l) = 0 then
                equations := concat!(equations, list delti)
            else
                equations:=concat!(equations,
                    list(Pl*delti-coefficient(delti,l)*pol))
        }
    }
}
pol
}

transform(q:SingleInteger, a:SingleInteger, l:Integer):SUP(S,"j") == {
    import from S;
    res : SUP(S,"j") := 1;
    for i in 1..l repeat {
        res := res * (monomial(coerce(q)$S,0) + monomial(coerce(a)$S,1))
    }
    res
}

apply(pol:SUP(S,"j"), q:SingleInteger, a:SingleInteger):SUP(S,"j") == {
    import from Integer;
    res : SUP(S,"j") := 0;
    for l in 0..degree(pol) repeat {
        res := res + monomial(coefficient(pol,l),0)*transform(q,a,l)
    }
    res
}

+++ Scinde une série formelle en a sous-séries.

```

```

split(ps:PowerSeries(S), a:SingleInteger) : List PowerSeries(S) == {
  import from POLDELTO(S);
  import from Integer;
  import from List Quotient(S);

  ASSERT( a >= 1);
  a = 1 => list ps;
  delt := delto(ps);
  cond := initcond(ps);
  deltsplit := split(delt,a);
  ai : Integer := coerce a;
  k : Integer := degree(deltsplit) quo ai;
  resdelt : List PowerSeries(S) := nil;
  m: Integer := coerce(#cond);
  r: Integer := degree(delt);
  mu := m - r;
  conddd : Partial List Quotient(S) := coefficients(ps,mu+ai*k);
  failed? conddd => error( "Not enough initial conditions
                          in order to split the PowerSeries");
  condd : List Quotient(S) := coerce(conddd);
  card := #condd;
  for q in 0..a-1 repeat {
    muq : Integer := 0;
    while (muq*ai < mu- coerce(q)) repeat muq := muq+1;
    mq := k + muq;
    deltq : POLDELTO(S) := 0;
    for l in 0..k repeat {
      deltq:=deltq+monomial(apply(coefficient(deltsplit,l*ai),q,a),l)
    }
    initcondq : List Quotient(S) := nil;
    i:SingleInteger := 0;
    while (coerce(i) <= mq-1) repeat {
      initcondq := concat!(initcondq, list condd.(q+1+i*a));
      i := i + 1
    }
    resdelt := concat!(resdelt,list createPowerSeries(initcondq,deltq));
  }
  resdelt
}

```

```

----- ci.as -----

#include "sumit"

#library complexring "complexring.ao"
import from complexring;

F ==> DoubleFloat;
CF ==> Complex DoubleFloat;

BalserRing : Category == Join(IntegralDomain,ComplexRing);

BalInt : BalserRing == Integer add {
  Rep ==> Integer;
  import from Rep;
  import from F;
  import from CF;

  coerce(i:%):CF == {ii : Integer := rep(i);
    iii: F:=coerce(ii)$F;
    complex(iii,0.)
  }

  extree(x:%):ExpressionTree == extree rep x;

  coerce(i:Integer):% == per(i);
}

```

----- differential.as -----

```
#include "sumit.as"

#library powerseries "powerseries.ao"
#library polodo "polodo.ao"
#library eulerlodo "eulerlodo.ao"

import from ci,autoder;
import from poldelto,powerseries;
import from polodo,eulerlodo;

Z ==> Integer;
SUP ==> SparseUnivariatePolynomial;
POLDELTO ==> LinearOrdinaryPolynomialDifferenceOperator;
POLODO ==> LinearOrdinaryPolynomialDifferentialOperator;
AUTOMOR ==> Automorphism;
LOEDO ==> EulerLinearOrdinaryPolynomialDifferentialOperator;

Differential(S: BalserRing ):  with {

  diffequation:  PowerSeries(S) -> Record(eqdiff:POLODO(S),sec:SUP(S,"x"));
    ++ Retourne une équation différentielle dont est solution la série
    ++ formelle.
  eulerequation:  PowerSeries(S) -> Record(eqdiff:LOEDO(S),sec:SUP(S,"x"));
    ++ Retourne une équation différentielle (avec l'opérateur d'Euler)
    ++ dont est solution la série formelle.
}

== add {

  diffequation(ps:PowerSeries(S)):Record(eqdiff:POLODO(S),sec:SUP(S,"x"))=={

    import from Integer,S,Quotient S,Partial Quotient S,
    import from UnivariateSkewPolynomialCategoryOps(SUP(S,"x"),POLODO(S));
    import from POLDELTO(S);
    import from Derivation SUP(S,"x");
    import from AUTOMOR SUP(S,"x");
    import from AutoDer(S);

    delta := delto(ps);
    r := degree(delta);
    lodop:  POLODO(S) := 0;
```

```

m : Integer := coerce(#initcond(ps));
for i in 0..r repeat {
  Q: SUP(S,"j") := coefficient(delta,r-i);
  q := degree(Q);
  for j in 0..q repeat {
    lodoi: POLODO(S) := 1;
    li:POLODO(S):=monomial(monomial(coerce(i-r)$S,0)$SUP(S,"x"),0);
    li := li + monomial(monomial(1$S,1)$SUP(S,"x"),1);
    for l in 1..j repeat {
      lodoi := times( lodoi,li,1$AUTOMOR(SUP(S,"x")),
                     function D()$Derivation(SUP(S,"x")))
    }
    lodop := lodop +
      monomial(monomial(coefficient(Q,j),i)$SUP(S,"x"),0)*lodoi
  }
}
pol: List Quotient(S) := nil;
for l in 0..r-1 repeat {
  pl: Quotient(S) := 0;
  for i in r-l..r repeat {
    alpha: Quotient(S) := coerce(coefficient(ps,i-r+l));
    beta:S:=apply(coefficient(delta,i),coerce(l-r)$S)$SUP(S,"j");
    pl := pl + (numerator(alpha)*beta)/ denominator(alpha)
  }
  pol := concat!(pol,list(pl))
}
for l in r..m-1 repeat {
  pl: Quotient(S) := 0;
  for i in 0..r repeat {
    alpha: Quotient(S) := coerce(coefficient(ps,i-r+l));
    beta:S:=apply(coefficient(delta,i),coerce(l-r)$S)$SUP(S,"j");
    pl := pl + (numerator(alpha)*beta)/ denominator(alpha)
  }
  pol := concat!(pol,list(pl))
}
div: S := 1;
for a in pol repeat div := div * denominator(a);
divv: Quotient(S) := coerce(div);
poly: SUP(S,"x") := 0;
for l in 0..m-1 repeat {
  poly:=poly+monomial(numerator(divv*first(pol)),l);
  pol := rest pol
}
dispose!(pol);

```

```

lodop:=monomial(monomial(div,0)$SUP(S,"x"),0)*lodop;
[lodop,poly]
}

eulerequation(ps:PowerSeries(S)):Record(eqdiff:LOEDO(S),sec:SUP(S,"x"))=={

import from Integer,S,Quotient S,Partial Quotient S;
import from UnivariateSkewPolynomialCategoryOps(SUP(S,"x"),LOEDO(S));
import from POLDELTO(S);
import from Derivation SUP(S,"x");
import from AUTOMOR SUP(S,"x");
import from AutoDer(S);

delta := delto(ps);
r := degree(delta);
lodo: LOEDO(S) := 0;
m : Integer := coerce(#initcond(ps));
for i in 0..r repeat {
    Q: SUP(S,"j") := coefficient(delta,r-i);
    q := degree(Q);
    for j in 0..q repeat {
        lodoi: LOEDO(S) := 1;
        li:LOEDO(S):=monomial(monomial(coerce(i-r)$S,0)$SUP(S,"x"),0);
        li:=li+monomial(monomial(1$S,0)$SUP(S,"x"),1);
        for l in 1..j repeat {
            lodoi:=times( lodoi,li,1$AUTOMOR(SUP(S,"x")),
                        function Theta()$Derivation(SUP(S,"x")))
        }
        lodo := lodo +
            monomial(monomial(coefficient(Q,j),i)$SUP(S,"x"),0)*lodoi
    }
}
pol: List Quotient(S) := nil;
for l in 0..r-1 repeat {
    pl: Quotient(S) := 0;
    for i in r-l..r repeat {
        alpha: Quotient(S) := coerce(coefficient(ps,i-r+l));
        beta:S:=apply(coefficient(delta,i),coerce(l-r)$S)$SUP(S,"j");
        pl := pl + (numerator(alpha)*beta)/ denominator(alpha)
    }
    pol := concat!(pol,list(pl))
}
for l in r..m-1 repeat {
    pl: Quotient(S) := 0;

```



```

    for i in 0..r repeat {
      alpha: Quotient(S) := coerce(coefficient(ps,i-r+1));
      beta:S:=apply(coefficient(delta,i),coerce(1-r)$S)$SUP(S,"j");
      pl := pl + (numerator(alpha)*beta)/ denominator(alpha)
    }
    pol := concat!(pol,list(pl))
  }
  div: S := 1;
for a in pol repeat div := div * denominator(a);
divv: Quotient(S) := coerce(div);
poly: SUP(S,"x") := 0;
for l in 0..m-1 repeat {
  poly := poly + monomial(numerator(divv * first(pol)),l);
  pol := rest pol
}
dispose!(pol);
lodo := monomial(monomial(div,0)$SUP(S,"x"),0) * lodo;
[lodo,poly]
}
}

```

```

----- eulerlodo.as -----

#include "sumit.as"

#library autoder "autoder.ao"
import from autoder,ci;

LODO ==> LinearOrdinaryDifferentialOperator;
SUP ==> SparseUnivariatePolynomial;

EulerLinearOrdinaryPolynomialDifferentialOperator(S: BalserRing):
    UnivariateSkewPolynomialCategory SUP(S,"x")

== LODO(SUP(S,"x"),Theta()$AutoDer(S),"Theta");

```

----- lagleg.as -----

```
#include "sumit.as"
#library geolink "geolink.ao";
import from geolink;
F ==> DoubleFloat;
SI ==> SingleInteger;
CF ==> Complex F;

LagLeg: with {

  laguerre: GeoLink -> Record(absci: List F, weight: List F);

  legendre: GeoLink -> Record(absci: List F, weight: List F)
}

== add {

+++ Définit les abscisses ( $t_i$ ) et les poids de la formule de quadrature
+++ de Laguerre. Les poids sont ceux des tables de quadrature.
+++ Par la suite, les poids seront le produit de ces poids par
+++  $\gamma'(t_i)\exp(\dots)$ . On se limite pour l'instant à 32 abscisses.
laguerre(gl: GeoLink): Record(absci: List F, weight: List F) == {
  rgl: Record(np: SI, gamma: F->CF, gammaderiv: F->CF) := coerce(gl);
  nlag := rgl.np;

  abscisses: List F :=
  [ 0.044489365833267025,      0.234526109519618520,
    0.576884629301886420,      1.07244875381781750,
    1.72240877644464520,      2.52833670642579450,
    3.49221327302199450,      4.61645676974976730,
    5.90395850417424360,      7.35812673318624040,
    8.98294092421259580,      10.7830186325399720,
    12.7636979867427250,      14.9311397555225550,
    17.2924543367153130,      19.8558609403360540,
    22.6308890131967750,      25.6286360224592470,
    28.8621018163234710,      32.3466291539647350,
    36.1004948057519710,      40.1457197715394420,
    44.5092079957549340,      49.2243949873086430,
    54.3337213333969110,      59.8925091621340210,
    65.9753772879350480,      72.6876280906627170,
    80.1874469779135280,      88.7353404178923960,
    98.8295428682839720,      111.751398097937680 ] ;
```

```

weights: List F :=
[ 0.114187105768104840,      0.266065216897615110,
  0.418793137324853030,      0.572532846499804740,
  0.727648788380971380,      0.884536719340249710,
  1.04361887589207680,       1.20534927415235260,
  1.37022133852178110,       1.53877725646864450,
  1.71161935268645730,       1.88942406344948410,
  2.07295934024653360,       2.26310663399696350,
  2.46088907248823600,       2.66750812639711680,
  2.88439209292204150,       3.11326132703958600,
  3.35621769259580220,       3.61586985648426860,
  3.89551304494854950,       4.19939410471158470,
  4.53311497853436170,       4.90427028761124460,
  5.32350097202366590,       5.80633321423362120,
  6.37661467415965270,       7.07352658070724300,
  7.96769350929590010,       9.20504033127818850,
  11.1630130907678720,       15.3901804152606420 ] ;

[abscisses,weights]
}

```

+++ Définit les abscisses (t_i) et les poids de la formule de quadrature

+++ de Legendre. Les poids sont ceux des tables de quadrature.

+++ Par la suite, les poids seront le produit de ces poids par

+++ $\gamma'(t_i) \exp(\dots)$ On se limite pour l'instant à 32 abscisses.

```

legendre(gl:GeoLink):Record(absci:List F,weight:List F)=={

```

```

  abscisses: List F :=

```

```

[ 0.001368069075259215,      0.00719424422736580920,
  0.0176188722062468040,      0.0325469620311301620,
  0.0518394221169739520,      0.0753161931337150130,
  0.102758102016028790,      0.133908940629855150,
  0.168477866534892390,      0.206142121379618830,
  0.246550045533885320,      0.289324361934682360,
  0.334065698858936190,      0.380356318873931410,
  0.427764019208601710,      0.475846167156130860,
  0.524153832843869130,      0.572235980791398280,
  0.619643681126068580,      0.665934301141063800,
  0.710675638065317630,      0.753449954466114650,
  0.793857878620381170,      0.831522133465107630,
  0.866091059370144850,      0.897241897983971090,
  0.924683806866285000,      0.948160577883026070,
  0.967453037968869830,      0.982381127793753120,
  0.992805755772634250,      0.998631930924740830 ] ;

weights: List F :=
[ 0.00350930500473504810,      0.00813719736545283530,
  0.0126960326546310290,      0.0171369314565107150,
  0.0214179490111133410,      0.0254990296311880870,
  0.0293420467392677730,      0.0329111113881809210,
  0.0361728970544242570,      0.0390969478935351570,
  0.0416559621134733720,      0.0438260465022019060,
  0.0455869393478819490,      0.0469221995404022890,
  0.0478193600396374300,      0.0482700442573639040,
  0.0482700442573639040,      0.0478193600396374300,
  0.0469221995404022890,      0.0455869393478819490,
  0.0438260465022019060,      0.0416559621134733720,
  0.0390969478935351570,      0.0361728970544242570,
  0.0329111113881809210,      0.0293420467392677730,
  0.0254990296311880870,      0.0214179490111133410,
  0.0171369314565107150,      0.0126960326546310290,
  0.00813719736545283530,      0.00350930500473504810 ] ;

[abscisses,weights]
}
}

```

```

----- laplace.as -----

#include "sumit.as"

#library path "path.ao"
#library point "point.ao"
#library ci "ci.ao"

import from complexring, path, method , localisation, minisup, point,ci;

SI ==> SingleInteger;
Z ==> Integer;
F ==> DoubleFloat;
CF ==> Complex F;

Laplace(S : BalserRing): with {

  readpath:() -> Path(S);
  ++ Lecture du chemin d'integration. Devrait se faire interactivement.
  derivee:( Record( listBorel:List Record(vi:List CF,phiqvi:List CF,wi:List CF),
    initerms:List Quotient(S)),
    SI, SI, CF, Z) -> List CF;
  ++ Calcule les derivees successives de la somme connaissant le
  ++ prolongement de chacune de ses sous-séries.
};

== add {

  derivee( borellist:
    Record( listBorel:List Record(vi:List(CF),phiqvi:List(CF),wi:List(CF)),
      initerms: List Quotient(S)),
    a: SI, b: SI, zr: CF, n: Z) : List CF == {

    import from Ratio Z;

    listres := borellist.listBorel;
    initerms := borellist.initerms;
    quadr: List(List(CF)) := nil;
    for q in 1..a repeat {
      quadr:=concat!(quadr,list quadrv(listres.q,a,b,zr,n))
    }
    derivn : List CF := nil;
    for i in 0..n repeat {
      derivaux : CF := 0;

```

```

    for q in 1..a repeat {
        alpha0 := initerms.q;
        quadrq := quadr.q;
        der: CF := first(quadrq);
        der:=der+ coerce(numerator(alpha0))$S
                /coerce(denominator(alpha0))$S;
        der := der * deriv(q-1, i, zr);
        for k in 0..i-1 repeat {
            bin := binome(i,k);
            binF:=coerce(numer(bin))$F/coerce(denom(bin))$F;
            deraux := element(quadrq,i-k+1) ;
            deraux := deraux * deriv(q-1, k, zr) * binF;
            der := der + deraux
        };
        derivaux := derivaux + der
    };
    derivn := concat!(derivn, list derivaux)
};
derivn
};

binome(n: Z, k: Z) : Ratio Z == {
    factorial(n) / (factorial(k) * factorial(n-k))
};

deriv(q: SI, k: Integer, xr: CF) : CF == {
    qi : Z := coerce(q);
    if (k=0) then coef : Z := 1
    else {
        coef : Z := qi;
        for i in 1..k-1 repeat coef := coef*(qi-i)
    };
    coeff : CF := coerce(coef);
    res : CF := coeff*xr^ (qi-k);
    res
};

quadr( res:Record(vi:List CF, phiqvi:List CF, wi:List CF), ):List CF=={
    a:SI, b:SI, zr:CF, n:Z

import from DoubleFloatElementaryFunctions;

viabs : List CF := res.vi;
phiqvi : List CF := res.phiqvi;
weightvi : List CF := res.wi;

```

```

quadr : List CF := nil;
aF : F := coerce(a);
bF : F := coerce(b);
derivn: List Record(coef: F, expx: F, expt: F) := [[1/bF,0.,0.]];
for l in 0..n repeat {
  resquadr: CF := 0;
  derivnn: List Record(coef: F, expx: F, expt: F) := nil;
  for j in 1..#derivn repeat {
    aux := derivn.j;
    if (aux.expx = 0) then {
      coeff := aux.coef * aux.expx;
      expxx := aux.expx - 1.;
      exptt := aux.expt;
      derivnn := cons([coeff,expxx,exptt], derivnn)
    };
    coeff := aux.coef * aF/bF;
    expxx := aux.expx - aF/bF - 1.;
    exptt := aux.expt + 1./bF;
    derivnn := cons([coeff,expxx,exptt], derivnn);
    resaux: CF := 0;
    for i in 1..#viabs repeat {
      vi := viabs.i;
      resaux:=resaux+ phiqvi.i*weightvi.i*exp(aux.expt*log(vi))
                      exp(-exp(log(vi)/bF)/exp(log(zr)*aF/bF))
    };
    resaux:=resaux*aux.coef*exp(aux.expx*log(zr));
    resquadr := resquadr + resaux
  };
  quadr := concat!(quadr, list resquadr);
  derivn := [derivnn.i for i in 1..#derivnn]
};
quadr
};

element(quadruu: List CF, n: Z): CF == {

  quadr := quadruu;
  for i in 2..n repeat quadr := rest(quadr);
  first(quadr)
};

+++ Lecture d'un chemin de sommation.
readpath():Path(S) == {

```



```

import from SI,F,CF,List F,S;
import from Method(S),Localisation(S),MiniSUP(S),Point(S);

ch:Path(S):=createSeg( complex(0.,0.),complex(1,0),32,
                      createSerie(createLoc(point(0@S)),0,[1$F],0));
ch:=ch+createSeg( complex(1,0),complex(2.,0),32,
                  createRK(createLoc(point(1@S))));
ch:=ch+createSeg( complex(2.,0),complex(5.,0),32,
                  createRK(createLoc(point(1@S))));
ch
}
}

```

```

----- poldelto.as -----

#include "sumit.as"

#library autoder "autoder.ao"
import from autoder,ci;

LDELTO ==> LinearOrdinaryDifferenceOperator;
SUP ==> SparseUnivariatePolynomial;

LinearOrdinaryPolynomialDifferenceOperator(S: BalserRing):
    UnivariateSkewPolynomialCategory SUP(S,"j")

== LDELTO(SUP(S,"j"),sigma()$AutoDer(S),"tau");

```

```

----- polodo.as -----

#include "sumit.as"

#library autoder "autoder.ao"
import from autoder,ci;

LODO ==> LinearOrdinaryDifferentialOperator;
SUP ==> SparseUnivariatePolynomial;

LinearOrdinaryPolynomialDifferentialOperator(S: BalserRing):
    UnivariateSkewPolynomialCategory SUP(S,"x")

== LODO(SUP(S,"x"),D()$AutoDer(S),"D");

```

```

----- powerseries.as -----

#include "sumit.as"

#library poldelto "poldelto.ao"
import from poldelto;
import from autoder,ci;

SUP ==> SparseUnivariatePolynomial;
POLDELTO ==> LinearOrdinaryPolynomialDifferenceOperator;

PowerSeries(S: BalserRing): Ring with {

  createPowerSeries:(List Quotient(S), POLDELTO(S)) -> %;
    ++ Crée une série formelle.
  dispose!: % -> ();
    ++ Détruit une série formelle.
  coerce: % -> Record(ic: List Quotient(S), delt: POLDELTO(S));
    ++ Donne la représentation interne.
  initcond: % -> List Quotient(S);
    ++ Retourne la liste des conditions initiales.
  delto: % -> POLDELTO(S);
    ++ Retourne l'opérateur aux différences.
  coherent?: % -> Boolean;
    ++ Teste si les premiers termes sont solutions de l'équation
    ++ aux différences.
  coefficients : (% , SingleInteger) -> Partial List(Quotient(S));
    ++ Retourne  $[a_0, \dots, a_{nb-1}]$ .
  coefficient : (% , SingleInteger) -> Partial Quotient(S);
    ++ Retourne  $a_{nb}$ .
  coefficients : (% , Integer) -> Partial List(Quotient(S));
    ++ Retourne  $[a_0, \dots, a_{nb-1}]$ .
  coefficient : (% , Integer) -> Partial Quotient(S);
    ++ Retourne  $a_{nb}$ .
  nextCoeff: (List Quotient(S), POLDELTO(S)) -> Partial Quotient(S);
  <<: (TextWriter, %) -> TextWriter
}

== add {

  Rep == Record(ic: List Quotient(S), delt: POLDELTO(S));
  import from Rep;

```

```

import from Integer,SUP(S,"j"),Quotient(S);

0 : % == per([], monomial(1,0));

1 : % == per([1] , monomial(1,0));

(ps1:%)+(ps2:%):%==error("The addition is not yet implemented");

(ps1:%)*(ps2:%):%==error("The multiplication is not yet implemented");

coerce(z:Integer) : % == error("not yet implemented");

coerce(i:SingleInteger): % == error("not yet implemented");

(ps : % )^(z:Integer) : % == error("not yet implemented");

<<(p:TextWriter, ps:%): TextWriter == {
  delta := delto(ps);
  cond := initcond(ps);
  p := p << "Initial Conditions:" << cond << newline;
  p := p << "Difference equation:" << delta << newline
}

-(ps:%) : % == {
  import from SingleInteger;
  cond := initcond(ps);
  condaux : List Quotient(S) := nil;
  delta := delto(ps);
  for l in 1.. #cond repeat {
    condaux:=concat!(condaux,list(-cond.l))
  }
  per([condaux,delta])
}

(s1:Quotient(S)) * (ps:%) : % == {
  import from SingleInteger;
  cond := initcond(ps);
  condaux : List Quotient(S) := nil;
  delta := delto(ps);
  for l in 1..#cond repeat {
    condaux := concat!(condaux,list(s1*cond.l))
  }
  per([condaux,delta])
}

```

```

(ps:%) * (s1:Quotient(S)) : % == s1*ps;

(ps1:%) = (ps2:%):Boolean == {
  import from SingleInteger;
  import from Quotient(S);
  cond := initcond(ps1-ps2);
  equalseries := true;
  l : SingleInteger := 1;
  while equalseries and (l <= #cond) repeat
    if (cond.l=0) then l:=l+1 else equalseries:=false
  equalseries
}

coerce(ps:%):Record(ic: List Quotient(S), delt: POLDELTO(S)) == rep(ps);

initcond(ps:%):List Quotient(S) == rep(ps).ic;

delto(ps:%): POLDELTO(S) == rep(ps).delt;

valuation(delt: POLDELTO(S)): Integer == {
  val: Boolean := false;
  l: Integer := 0;
  r := degree(delt);
  while val and (l <= r) repeat
    if (coefficient(delt,l) = 0) then l:=l+1
    else val := true
  l
}

simplify(delt: POLDELTO(S)): POLDELTO(S) == {
  import from AutoDer(S);
  import from Automorphism SUP(S,"j");
  r0 := valuation(delt);
  r := degree(delt);
  res : POLDELTO(S) := 0;
  for l in r0..r repeat
    res := res +
      monomial(apply((inv sigma())^ r0,coefficient(delt,l)),l-r0)
  res
}

createPowerSeries(cond:List Quotient(S), delta:POLDELTO(S)): % == {
  m : Integer := coerce(#cond);

```

```

r : Integer := degree(delta);
(m < r) => error("Not enough initial conditions");
ps := per([cond,simplify delta]);
ps
}

dispose!(ps:%):() == {
  dispose!(initcond(ps));
  dispose!(rep(ps))
}

coherent?(ps:%):Boolean == {
  import from Partial(Quotient(S));
  import from S;

  cond := initcond(ps);
  delta := delto(ps);
  m : Integer := coerce(#cond);
  r := degree(delta);
  mu := m-r-1;
  coh : Boolean := true;
  j: Integer := 0;
  while (j <= mu) and coh repeat {
    aux : Quotient(S) := 0;
    for i in 0..r repeat {
      aux:=aux+(apply(coefficient(delta,i),coerce(j))*
        numerator(coerce(element(cond,i+j+1))))/
        denominator(coerce(element(cond,i+j+1)))
    }
    if (aux = 0) then coh := false
    else j := j+1
  }
  coh
}

element(cond: List Quotient(S), i:Integer): Partial Quotient(S) == {
  import from List(Quotient(S));

  if (i<1) or (i > coerce(#cond)) then failed
  else
  if (1 <= i) and (i <= coerce(#cond)) then {
    condaux := cond;
    for j in 2..i repeat condaux := rest(condaux);
    el : Quotient(S) := first(condaux)$List(Quotient(S))
  }
}

```

```

    }
    if (1<=i) and (i<=coerce(#cond)) then coerce(el) else failed
  }

nextCoeff(l: List Quotient(S), delta:POLDELTO(S)):
  Partial Quotient(S) == {

    import from S;
    import from Partial Quotient(S);
    import from SUP(S,"j");

    m: Integer := coerce(#l);
    r: Integer := degree(delta);
    mu := m-r;
    mus : S := coerce(mu);
    val : S := apply(leadingCoefficient(delta),mus);
    if (mu >= 0) and (val = 0) then {
      res : Quotient(S) := 0;
      for j in 0..r-1 repeat {
        el : Quotient(S) := coerce(element(l,mu+j+1));
        ell : S := apply(coefficient(delta,j),mus);
        res:=res-(ell*numerator(el))/denominator(el)
      }
      coerce(numerator(res)/(denominator(res)*val))
    }
    else failed
  }

coefficients(ps:%, nb: Integer): Partial List Quotient(S) == {
  import from Partial(Quotient(S));

  cond := initcond(ps);
  delta := delto(ps);
  m : Integer := coerce(#cond);
  res: List Quotient(S) := nil;
  welldefined: Boolean := true;
  if (nb<=m) then res:=[coerce(element(cond,i)) for i in 1..nb]
  else {
    res := cond;
    i : Integer := 1;
    while (i <= nb-m) and welldefined repeat {
      nextco := nextCoeff(res,delta);
      if failed? nextco then welldefined := false
      else res := concat(res, list(coerce(nextco)));
    }
  }
}

```



```

        i := i + 1
      }
    }
    if welldefined then coerce(res) else failed
  }

coefficients(ps:%, nb:SingleInteger):Partial List Quotient(S) ==
  coefficients(ps,coerce(nb)$Integer);

coefficient(ps: %, nb: Integer): Partial Quotient(S) == {
  lcoef:Partial List Quotient(S):=coefficients(ps,nb+1);
  if failed? lcoef then failed
  else coerce(last(coerce(lcoef)))
}

coefficient(ps:%, nb:SingleInteger):Partial Quotient(S) ==
  coefficient(ps,coerce(nb)$Integer)
}

```

```
----- runge.as -----
```

```
#include "sumit.as"
#library powerseries "powerseries.ao"
#library polodo "polodo.ao"
#library eulerlodo "eulerlodo.ao"
#library complexring "complexring.ao"

import from ci,autoder,poldelto,powerseries;
import from polodo,eulerlodo;
import from complexring;

SUP ==> SparseUnivariatePolynomial;
POLODO ==> LinearOrdinaryPolynomialDifferentialOperator;
LOEDO ==> EulerLinearOrdinaryPolynomialDifferentialOperator;
F ==> DoubleFloat;
CF ==> Complex F;
SI ==> SingleInteger;

Runge_Kutta(S: BalserRing): with {

    generateDerivs:(POLODO(S),SUP(S,"x"),F->CF,F->CF,CF)->
        (List CF, F) -> List CF;
    eulerDerivs:(LOEDO(S),SUP(S,"x"),F->CF,F->CF,CF ->
        (List CF, F) -> List CF;
    rk4f:(List CF,F,F,SI,(List CF,F)->List CF)->List CF
}

== add {

    generateDerivs(eqdiff: POLODO(S), sec:SUP(S,"x"),gamma:F->CF,
        gammaderiv:F->CF,znieme:CF):(List CF,F)->List CF=={

        (y: List CF, t: F):List CF +->
            functionDerivs(eqdiff,sec,gamma,gammaderiv,y,t,znieme)
    }

    functionDerivs(eqdiff:POLODO(S),sec:SUP(S,"x"),gamma:F->CF,
        gammaderiv:F->CF,y:List CF,t:F,znieme:CF):List CF=={

        r: Integer := degree(eqdiff);
        i: SI := 1;
        res: CF := 0;
```

```

dydx: List CF := nil;
coefr := eval(coefficient(eqdiff,r),gamma(t));
while coerce(i) <= r-1 repeat {
    dydx:=concat!(dydx,list(y(i+1)*gammaderiv(t)));
    res:=res+eval(coefficient(eqdiff,coerce(i)-1),gamma(t))*y(i);
    i := i + 1
}
if (gamma(t)=0) then dydx:=concat!(dydx,list(znieme*gammaderiv(t)))
else {
    res:=res+eval(coefficient(eqdiff,r-1),gamma(t))*y(i);
    res := eval(sec,gamma(t))-res;
    res := gammaderiv(t) * res / coefr;
    dydx:=concat!(dydx, list res)
}
dydx
}

eval(pol: SUP(S,"x"), xt: CF): CF == {
    import from Integer,S;
    at: CF := 0;
    for i in 0..degree(pol) repeat {
        coeff := coefficient(pol,i);
        coefff : CF := coerce(coeff);
        at := at + coefff * xt^$CF i
    }
    at
}

eulerDerivs(eqdiff:LOEDO(S),sec:SUP(S,"x"),gamma:F->CF,
    gammaderiv: F -> CF, a1: CF): (List CF, F)->List CF=={

    (y: List CF, t: F):List CF +->
        eulerfunctionDerivs(eqdiff,sec,gamma,gammaderiv,y,t,a1)
}

eulerfunctionDerivs(eqdiff:LOEDO(S),sec:SUP(S,"x"),gamma:F->CF,
    gammaderiv:F->CF,y>List CF,t:F,a1:CF):List CF=={

    r: Integer := degree(eqdiff);
    dydx: List CF := nil;
    if (t = 0) then {
        i: SI := 1;
        while coerce(i) <= r repeat {
            dydx:=concat!(dydx,list(a1*gammaderiv(0)));

```

```

        i := i+1;
    }
}
else {
    xt := gamma(t);
    i:  SI := 1;
    alphas := gammaderiv(t) / xt;
    res: CF := 0;
    while coerce(i) <= r-1 repeat {
        dydx := concat!(dydx,list(y(i+1) * alphas));
        i := i+1
    }
    coefer := eval(coefficient(eqdiff,r),xt);
    i:=1;
    while coerce(i) <= r repeat {
        tt: CF := 1;
        coef: CF := 0;
        coefi := coefficient(eqdiff,coerce(i)-1);
        di := degree(coefi);
        for j in 0..di repeat {
            coef:=coef+coerce(coefficient(coefi,j))$S *tt;
            tt := tt*xt
        }
        res := res + coef * y(i);
        i := i+1;
    }
    res := eval(sec, xt) - res;
    res := res / coefer;
    res := res * alphas;
    dydx:=concat!(dydx, list res)
}
dydx
}

```

```

rk4f(cond:List CF,xbegin:F,xend:F,nstep:SI,derivs:(List CF,F)->List CF):
List CF == {

h : F := (xend-xbegin) / coerce(nstep);
y0 := cond;
x0 := xbegin;
for i in 1..nstep repeat {
    x1 := x0 + h;
    y0 := rk4step(y0,x0,x1,derivs);
    x0 := x1
}

```

```

    }
    y0
  }

rk4step(y0:List CF,x0:F,x1:F,derivs:(List CF,F)->List CF):
  List CF == {

    h := x1-x0;
    k1 := h * derivs(y0 , x0);
    k2 := h * derivs(y0 + 0.5*k1 , x0 + 0.5*h);
    k3 := h * derivs(y0 + 0.25*(k1 + k2) , x0 + 0.5*h);
    k4 := h * derivs(y0 - k2 + 2. *k3, x0 + h);
    y1 := y0 + 1./6. * (k1 + 4. * k3 + k4);
    dispose!(k1);
    dispose!(k2);
    dispose!(k3);
    dispose!(k4);
    y1
  }

(h: F) * (x: List CF) : List CF == {
  import from SingleInteger;
  [h*xx for xx in x]
}

(x: List CF) + (y: List CF) : List CF == {
  import from SingleInteger;
  cardx := #(x);
  [x(i) + y(i) for i in 1..cardx]
}

(x: List CF) - (y: List CF) : List CF == {
  import from SingleInteger;
  cardx := #(x);
  [x(i) - y(i) for i in 1..cardx]
}

}

```

----- un.as -----

```
#include "sumit.as"

#library balser "balser.ao"
#library poldelto "poldelto.ao"

import from balser;
import from ci,poldelto,powerseries;

SUP ==> SparseUnivariatePolynomial;
POLDELTO ==> LinearOrdinaryPolynomialDifferenceOperator;
CF==>Complex DoubleFloat;
S ==> BalInt;

import from POLDELTO(S);
import from Balser(S);
import from Integer;
import from S;

session():() == {
    coef1:SUP(S,"j"):=monomial(1,0);
    coef0:SUP(S,"j"):=monomial(1,1);
    print << "Polynome aux differences: " << newline;
    pol:POLDELTO(S):=monomial(coef0,0)+monomial(coef1,1);
    print << "Liste des premiers coefficients générant la serie ";
    print << newline;
    inip:List(Quotient(S)):=list((0,1));
    print << inip << newline;
    print << "Série d'Euler:" << newline;
    ps:PowerSeries(S):=createPowerSeries(inip,pol);
    print << ps << newline;
    print << "Liste des pentes:" << newline;
    sum:List Ratio SingleInteger:=list((1));
    print << sum << newline;
    print << newline;
    print << "Point en lequel on veut calculer la somme:" << newline;
    zr: CF := complex(0.01,0.);
    print << "zr = " << zr << newline;
    print << "Somme:" << newline;
    s := balser(ps,sum,zr);
    print << s << newline;
    print << newline;
```

```

print << "Point en lequel on veut calculer la somme:" << newline;
zr: CF := complex(0.1,0.);
print << "zr = " << zr << newline;
print << "Somme:" << newline;
s := balser(ps,sum,zr);
print << s << newline;
print << newline;
print << "Point en lequel on veut calculer la somme:" << newline;
zr: CF := complex(1,0.);
print << "zr = " << zr << newline;
print << "Somme:" << newline;
s := balser(ps,sum,zr);
print << s << newline
}

```