Quelques méthodes d'intégration pour les équations différentielles

Françoise Richard-Jung

27 mai 2004

Comme son titre peut le laisser supposer, cet article fait suite à l'exposé paru dans le numéro 70 de cette revue, intitulé "Intégration numérique des équations différentielles : influence des instruments et des pratiques de calcul"[17]. Il a pour objectif de promouvoir des méthodes d'analyse des équations différentielles radicalement différentes et sans doute moins connues que les méthodes numériques "classiques". Fondamentalement, ces méthodes s'appliquent à des équations en variable complexe et peuvent être présentées en trois parties :

1. méthodes formelles : développement en séries dans le champ complexe
2. des solutions formelles aux "vraies" solutions : sommation de séries divergentes
3. visualisation graphique des solutions au voisinage de leurs singularités.

Ici, nous choisirons de suivre une autre démarche : partant de l'étude élémentaire d'une équation en variable réelle, nous passerons à la visualisation des solutions dans le champ complexe et comparerons les résultats obtenus en utilisant une fonction *spéciale* dont l'évaluation est réputée précise, des développements en séries convergentes ou divergentes, une intégration numérique (en une variable réelle) le long d'un chemin du plan complexe.
Le mode de représentation graphique adopté faisant appel à la couleur de manière essentielle, une version couleur de ce texte est disponible à l'adresse suivante : http ://www-lmc.imag.fr/lmc-cf/Francoise.Jung/index.html.

1 Les équations envisagées

L'étude que nous allons présenter ici se place dans le cadre général des équations scalaires du type $f(x, X, y, y', \ldots, y^{(d)}) = 0$, où f est une fonction de $d + 2 + r$ variables : x et X sont des variables indépendantes, x est scalaire et X un vecteur de dimension r, y est une fonction scalaire et $y', \ldots, y^{(d)}$ sont ses dérivées successives par rapport à x. Les différents cas que cela recouvre peuvent être visualisés sur le schéma suivant (Figure 1), avec les méthodes de résolution symbolique associées ; le lecteur intéressé pourra consulter [9] pour un panorama complet des équations traitées et des méthodes utilisées.

L'objectif de cet exposé est d'illustrer la branche centrale - équations différentielles linéaires - et plus particulièrement les méthodes d'analyse locale faisant intervenir des développements en séries. Dans ce but, nous prendrons comme fil conducteur l'équation différentielle suivante (équation d'Euler) :

$$x^2 y'(x) + y(x) = x. \qquad (1)$$

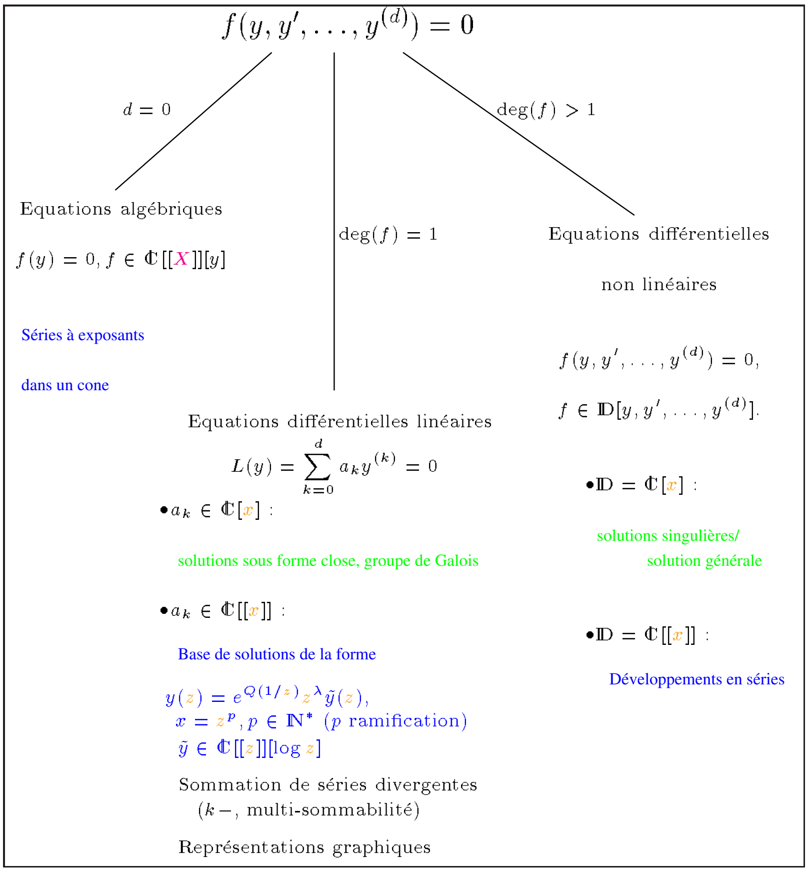


Figure 1 : le projet Irondel, organisation générale

2 Premiers résultats graphiques

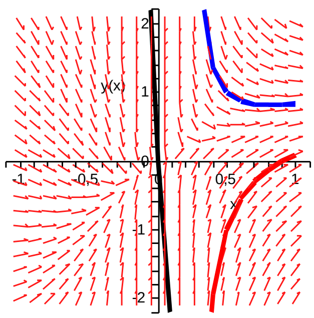
La façon la plus élémentaire d'appréhender une telle équation est de chercher à représenter graphiquement quelques solutions (réelles). Ce que l'on peut faire *facilement* avec la commande DEplot de MAPLE :

```
> eq1:=x^2*diff(y(x),x)+y(x)-x;
```

$$eq1 := x^2 \left(\frac{d}{dx}\,\mathrm{y}(x)\right) + \mathrm{y}(x) - x$$

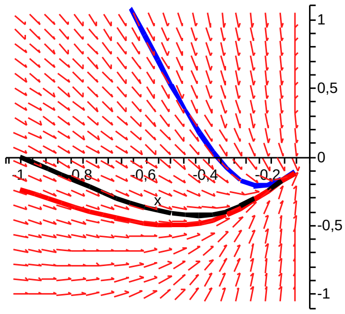
```
> with(DEtools):
> DEplot(eq1,y(x),x=-1..1,[[y(0)=0.],[y(0.5)=1.],[y(0.5)=-1.]],y=-2..2,
> linecolor=[black,blue,red]);
```

Fig. 2 – quelques solutions sur $\mathbb{R}^{*+}$



```
> DEplot(eq1,y(x),x=-1..-0.1,[[y(-1)=0.],[y(-0.5)=0.5],[y(-0.5)=-0.5]],
> y=-1..1,linecolor=[black,blue,red]);
```

FIG. 3 – quelques solutions sur $\mathbb{R}^{*-}$



Les commandes suivantes conduisent à des erreurs, ce qui est assez facile à comprendre...

```
> DEplot(eq1,y(x),x=-1..1,[[y(-1)=0.],[y(0.5)=1.]],y=-1..1);

Error, (in DEtools/DEplot/drawlines) Stopping integration due to
division by zero

> dsolve({eq1,y(0)=1.},y(x),type=numeric,output=array([1.]));

Error, (in dsolve/numeric/checksing) ode system has a removable
singularity at x=0. Initial data is restricted to {y(x) = 0}
```

Celle-ci donne par contre un résultat faux, assorti d'une mise en garde peu appropriée.

```
> dsolve({eq1,y(0)=0.},y(x),type=numeric,output=array([1.]));

Warning, cannot evaluate the solution further right of 1., probably a
singularity
```

$$\left[\begin{array}{c} [x, \mathrm{y}(x)] \\ \left[\begin{array}{cc} 1. & 0. \end{array} \right] \end{array} \right]$$

3 Solutions analytiques

Reprenons l'équation (1). La méthode de la *variation de la constante* sur $\mathbb{R}^{*+}$ donne la solution générale, définie sur $\mathbb{R}^{*+}$:

$$f_C(x) = e^{\frac{1}{x}}\Big(\int_0^x \frac{e^{\frac{-1}{t}}}{t}dt + C\Big).$$

Cette solution peut être exprimée à l'aide d'une *fonction spéciale*, la **fonction exponentielle intégrale**, définie dans le champ complexe par :

$$E_1(z) = \int_x^\infty \frac{e^{-u}}{u}du, \qquad |Arg(z)| < \pi.$$

Posons $f_0(x) = e^{\frac{1}{x}}E_1(\frac{1}{x})$, pour $x \in \mathbb{R}^{*+}$; on obtient $f_C(x) = f_0(x) + Ce^{\frac{1}{x}}$.
La fonction E_1 est connu du logiciel MAPLE sous le nom de $Ei(1,.)$.

```
> f0:=exp(1/x)*Ei(1,1/x);
```

$$f0 := e^{(\frac{1}{x})}\,\mathrm{Ei}(1, \frac{1}{x})$$

```
> subs(y(x)=f0,eq1):simplify(%);
```

$$0$$

On vérifie que f_0 est bien solution de l'équation (1) ! Mais aussi que f_0 est prolongeable par continuité en 0, et que la fonction ainsi prolongée est dérivable en 0.

```
> limit(f0,x=0);
```

$$0$$

```
> limit(diff(f0,x),x=0);
```

$$1$$

Il est *facile* de calculer et représenter les solutions de la figure 2.

```
> C1:=evalf((1.-subs(x=0.5,f0))*exp(-1/0.5));
```

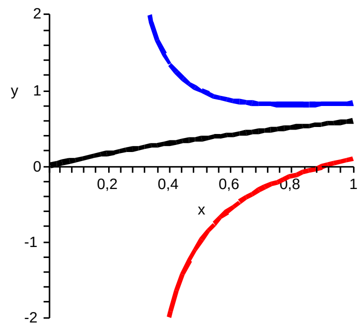
$$C1 := 0.08643477250$$

```
> C2:=evalf((-1.-subs(x=0.5,f0))*exp(-1/0.5));
```

$$C2 := -0.1842357939$$

```
> plot({f0,f0+C1*exp(1/x),f0+C2*exp(1/x)},x=0.01..1,y=-2..2,
> color=[black,blue,red],thickness=3);
```

FIG. 4 – solutions sur $\mathbb{R}^{*+}$, à comparer avec la figure 2



La solution *numérique* passant par le point $(0,0)$ ne correspond pas du tout à la solution f_0...
Le même travail sur $\mathbb{R}^{*-}$ conduit aux résultats suivants :

```
> g0:=exp(1/x)*int(exp(-1/t)/t,t=-1..x);
```

$$g0 := e^{\left(\frac{1}{x}\right)}\left(\mathrm{Ei}(1,\ \frac{1}{x}) - \mathrm{Ei}(1,\ -1)\right)$$

```
> C1:=evalf((0.5-subs(x=-0.5,g0))*exp(1/0.5));
```

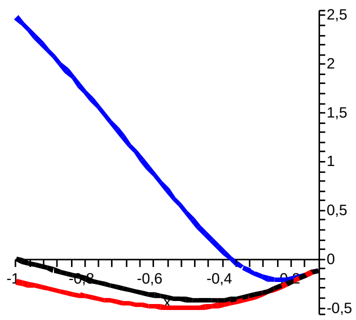
$$C1 := 6.753644589 - 0.\ I$$

```
> C2:=evalf((-0.5-subs(x=-0.5,g0))*exp(1/0.5));
```

$$C2 := -0.635411511 - 0.\ I$$

```
> plot({g0,g0+C1*exp(1/x),g0+C2*exp(1/x)},x=-1..-0.1,color=[black,blue,red],
> thickness=3);
```

FIG. 5 – solutions sur $\mathbb{R}^{*-}$, à comparer avec la figure 3



La comparaison avec les solutions *numériques* sur $\mathbb{R}^{*-}$ prouve que ces dernières sont correctes.

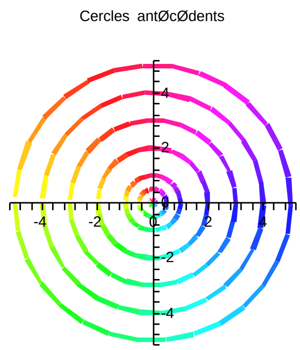
Toutes les solutions $g_C(x) = g_0(x) + Ce^{\frac{1}{x}}$, définies a priori sur $\mathbb{R}^{*-}$, sont prolongeables en 0 (par la valeur 0) et deviennent dérivables sur $\mathbb{R}^-$, de dérivée 1 en 0. Elles peuvent donc toutes être raccordées en 0 à la fonction f_0. On obtient ainsi une infinité de solutions de classe $\mathcal{C}^1(\mathbb{R})$ passant par le point $(0,0)$ (mais aucune solution passant par chacun des points $(0, y_0)$, $y_0 \neq 0$).

4 Visualisation dans le champ complexe

Dans ce paragraphe, nous allons nous intéresser au comportement des solutions de l'équation (1) dans le champ complexe. La fonction $Ei(1,.)$ est définie dans tout le plan complexe, sauf en 0, avec une coupure sur l'axe réel négatif, la valeur de la fonction aux points de la coupure étant définie par continuité dans le sens des arguments croissants. Par définition, f_0 est donc définie en tout point du plan complexe, la valeur de la fonction sur l'axe réel négatif étant calculée en considérant que les points antécédents ont pour argument $-Pi$.
Cherchons maintenant à visualiser ces fonctions, en tant que fonctions complexes de la variable complexe.
Le principe de la représentation est le suivant : on affiche dans le plan complexe l'image par la fonction étudiée f d'un arc de cercle dans le plan complexe de départ. La courbe antécédent et la courbe image sont représentées sur deux graphiques séparés, la couleur liant les points entre eux : chaque point $f(z)$ est affiché avec la même couleur que z, celle-ci étant déterminée par l'argument de z [16].
Les fonctions considérées étant généralement multi-variées, les points antécédents sont représentés par leur coordonnées d'Euler, alors que les points images sont calculés en coordonnées cartésiennes. Ainsi, nous manipulons des listes de points colorés de deux types $(\rho, \theta, couleur)$ et $(x, y, couleur)$, et les deux fonctions *draw_rhotheta* et *draw_xy* permettent d'afficher de telles listes de points. La fonction *create_circle*$(\rho, \theta^-, \theta^+, nbpoints)$ crée un arc de cercle de centre l'origine, de rayon ρ, comme une liste de *nbpoints* de module ρ et d'argument compris entre θ^- et θ^+.

```
> read desiragile:
> ## dans le champ complexe : image de cercles
> for i from 1 to 5 do circ[i]:=create_circle(i,-Pi,3.1,30): od:
> circ[6]:=create_circle(0.5,-Pi,3.1,50):
> circ[7]:=create_circle(0.1,-Pi,3.1,30):
> draw_rhotheta([seq(op(circ[i]),i=1..7)],"Cercles antécédents");
```

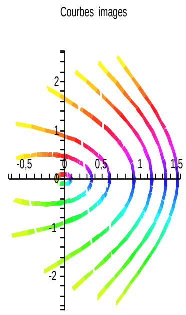
FIG. 6 – 7 cercles autour de l'origine




```
> for i from 1 to 7 do
> seq1[i]:=map(proc(pp) local x,y; x:=pp[1]*(cos(pp[2])+I*sin(pp[2]));
> y:=exp(1/x)*(Ei(1,1/x));
> [Re(y),Im(y),pp[3]] end proc,circ[i]):
> od:
```

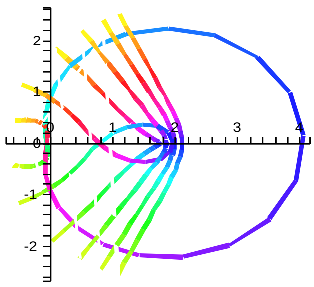
```
> draw_xy([seq(op(seq1[i]),i=1..7)],"Courbes images");
```

FIG. 7 – image par f_0 des 7 cercles



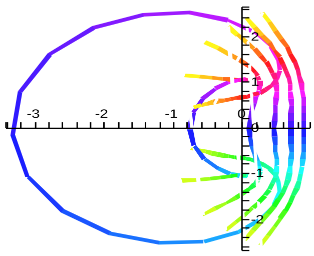

```
> for i from 1 to 6 do
> seq2[i]:=map(proc(pp) local x,y; x:=pp[1]*(cos(pp[2])+I*sin(pp[2]));
> y:=exp(1/x)*(Ei(1,1/x)+0.5);
> [Re(y),Im(y),pp[3]] end proc,circ[i]):
> od:
> draw_xy([seq(op(seq2[i]),i=1..6)],"");
```

FIG. 8 – image par $f_0 + 0.5e^{\frac{1}{x}}$ des 6 premiers cercles



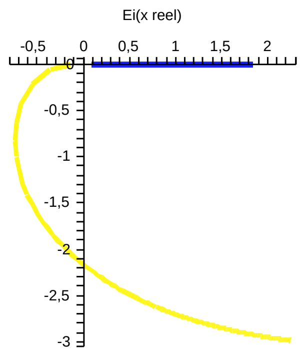
```
> for i from 1 to 6 do
> seq2[i]:=map(proc(pp) local x,y; x:=pp[1]*(cos(pp[2])+I*sin(pp[2]));
> y:=exp(1/x)*(Ei(1,1/x)-0.5);
> [Re(y),Im(y),pp[3]] end proc,circ[i]): od:
> draw_xy([seq(op(seq2[i]),i=1..6)],"");
```

FIG. 9 – image par $f_0 - 0.5e^{\frac{1}{x}}$ des 6 premiers cercles



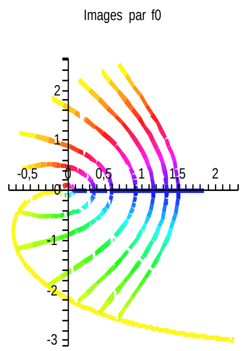
```
> ##dans le champ complexe : image de l'axe réel
> seg1:=create_seg(0,-20.,0.,160,0.16):
> seg2:=create_seg(0,0.,8.,80,0.66):
> seg:=[op(seg1),op(seg2)]:
> seq3:=map(x -> [exp(1/x[1])*Ei(1,1/x[1]),x[3]],seg):
> seq33:=map(x -> [Re(x[1]),Im(x[1]),x[2]],seq3):
> draw_xy(seq33, "Ei(x réel)");
```

FIG. 10 – image par f_0 de l'axe réel




```
> draw_xy([seq(op(seq1[i]),i=1..7),op(seq33)],"Images par f0");
```

FIG. 11 – superposition des figures 7 et 10



L'étude de $E_1(x)$ au voisinage de l'origine prouve que cette fonction admet le développement suivant :

$$E_1(x) = -\gamma - \ln x - \sum_{n=1}^{\infty} \frac{(-1)^n x^n}{n n!}, \tag{2}$$

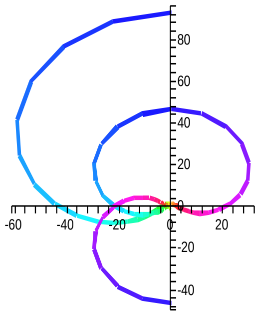
ce qui permet de la prolonger sur toute la surface de Riemann du logarithme[1]. On peut donc aussi prolonger f_0, avec la relation : $f_0(xe^{2i\pi}) = f_0(x) + 2i\pi e^{\frac{1}{x}}$. Cette définition "étendue" permet de calculer l'image par f_0 de cercles faisant plusieurs tours autour de l'origine.

```
> circ1:=create_circle(0.5,-2*Pi,4*Pi,150):
```

[1] γ est la constante d'Euler

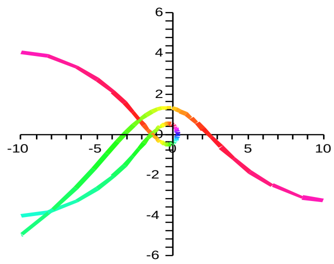
```
> seq5:=map(proc(pp) local x,y,k; x:=pp[1]*(cos(pp[2])+I*sin(pp[2]));
> y:=exp(1/x)*Ei(1,1/x);
> k:=floor((pp[2]+Pi)/(2*Pi));
> y:=y+2*I*Pi*k*exp(1/x);
> [Re(y),Im(y),pp[3]] end proc,circ):
> draw_xy([op(seq5)],"");
```

FIG. 12 – f_0 sur la surface de Riemann du logarithme.




```
> draw_xy([op(seq5)],"",-10.,10.,-6.,6.);
```

FIG. 13 – zoom de la figure 12 près de l'origine.



5 Développement en séries

Dans ce paragraphe, nous allons voir comment retrouver tous ces résultats en *oubliant* la fonction exponentielle intégrale, avec la seule connaissance des développements en séries des solutions.

5.1 avec l'option "series" de MAPLE

```
> sol:=op(2,dsolve({eq1,y(0)=0.},y(x),'series'));
```

$$\mathrm{y}(x) = x - x^2 + 2\,x^3 - 6\,x^4 + 24\,x^5 + \mathrm{O}(x^6)$$

On reconnaît la *série d'Euler*

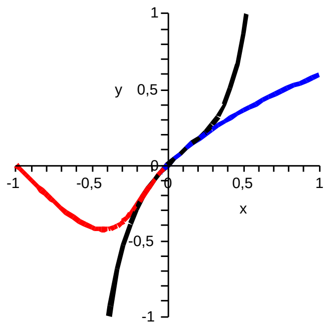
$$\sum_{n=0}^{\infty} (-1)^n n! x^{n+1},$$

qui est bien sûr divergente.
Pourtant, le lien avec les *vraies* solutions ne paraît pas fortuit, comme on peut le voir sur le

graphique suivant[2]...

```
> P:=convert(sol,'polynom');
> plot([P,f0,g0],x=-1..1,y=-1..1,color=[black,blue,red],thickness=3);
```

FIG. 14 – solutions au voisinage de 0



5.2 avec le logiciel DESIR

Celui-ci s'adresse aux équations différentielles linéaires (à coefficients polynomiaux) *homogènes* et à leur analyse locale dans le champ complexe. Pour entrer dans ce cadre, nous pouvons "transformer" l'équation (1) par dérivation en l'équation (non équivalente !) suivante :

$$x^3 y''(x) + x(x+1)y'(x) - y(x) = 0 \tag{3}$$

Nous ne souhaitons pas décrire ici les algorithmes de résolution formelle [10, 7], mais plutôt expliquer comment exploiter les résultats, d'un point de vue numérique et graphique.

```
> read desir2:
> eq2:=x^3*diff(y(x),x$2)+x*(x+1)*diff(y(x),x)-y(x);
```

$$eq2 := x^3 \left(\frac{d^2}{dx^2}\,\mathrm{y}(x)\right) + x\,(x+1)\left(\frac{d}{dx}\,\mathrm{y}(x)\right) - \mathrm{y}(x)$$

```
> desir(eq2,y(x),'t',res1);
```

$$[[\mathrm{x}(t) = t,\ \mathrm{y}(t) = t\,(1 - t + 2\,t^2 + \mathrm{O}(t^3))],\ [\mathrm{x}(t) = t,\ \mathrm{y}(t) = e^{(\frac{1}{t})}\,(1 + \mathrm{O}(t^3))]]$$

```
> sol1:=`convert/gsol`(res1):
```

$$x\,(1 - x + 2\,x^2 + \mathrm{O}(x^3))$$

$$e^{(\frac{1}{x})}\,(1 + \mathrm{O}(x^3))$$

On a ainsi obtenu une **base** de solutions formelles au voisinage de l'origine. La fonction de conversion *convert/gsol* est utilisée pour passer de solutions *formelles rationnelles* obtenues grâce à l'algorithme de Newton rationnel ([2, 14]) à des solutions formelles *classiques* (ici, en l'absence de ramification, il n'y a pas de différence), utilisables par les fonctions de calcul. Dans la suite, on notera $\hat{f}_0$ la première solution et $\hat{f}_1$ la deuxième solution[3].

Au voisinage de l'infini, on obtient une autre base de solutions formelles. Avec $t = 1/x$, on notera $\hat{g}_0(t)$ le premier élément et $\hat{g}_1(t)$ le deuxième. Ces deux solutions comportent des séries de puissance en la variable t, et dans le cas de la deuxième, un facteur logarithmique.

```
> desir(eq2,y(x),'t',x=infinity,res2):
> sol2:=`convert/gsol`(res2):
```

$$1 + \frac{1}{x} + \frac{1}{2\,x^2} + \mathrm{O}(\frac{1}{x^3})$$

[2]Remarque : le polynôme P semble proche de g_0, mais ni plus ni moins que d'un autre g_C ; par contre, du côté réel positif, il semble bien distinguer f_0 parmi tous les f_C.

[3]Remarque : pour $\hat{f}_1$, $O(x^3) = 0$.

$$\ln(\frac{1}{x})\,(1+\frac{1}{x}+\frac{1}{2\,x^2})-\frac{1}{x}-\frac{3}{4\,x^2}+\mathrm{O}(\frac{1}{x^3})$$

5.3 Sommation des séries et visualisation des résultats

Au voisinage de l'infini, les séries qui apparaissent dans les solutions sont convergentes, de rayon de convergence infini. En particulier, on reconnaît $\hat{g}_0(t) = \exp t$ et

$$\hat{g}_1(t) = -\ln t \exp t - t - \frac{3}{4}t + \ldots$$

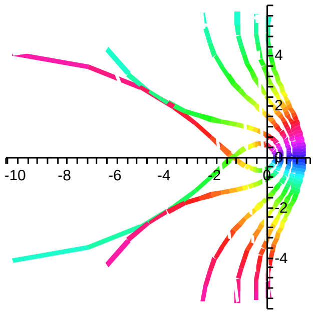
La comparaison avec le développement (2) permet d'établir que la fonction f_0 s'exprime comme combinaison linéaire des solutions $\hat{g}_0$ et $\hat{g}_1$ de la façon suivante :

$$f_0(x) = -\gamma\hat{g}_0(1/x) - \hat{g}_1(1/x).$$

La fonction *compute_solution(sol,combli,liste_de_points)* permet de calculer l'image de la liste de points *liste_de_points* par **la** solution qui s'exprime dans la base *sol* grâce à la combinaison linéaire *combli*. Les séries qui apparaissent dans $\hat{g}_0$ et $\hat{g}_1$ sont tronquées à l'ordre fixé par la variable globale de MAPLE "Order". On peut ainsi obtenir à nouveau l'image par f_0 d'arcs de cercle autour de l'origine.

```
> for i from 1 to 5 do circ[i]:=create_circle(i,-5.,5.,60): od:
> circ[6]:=create_circle(0.5,-5.,5.,60):
> Order:=21:
> for i from 1 to 6 do
> seq1[i]:=compute_solution(sol2,[-gamma,-1.],circ[i]):od:
> draw_xy([seq(op(seq1[i]),i=1..6)],"");
```

FIG. 15 – séries convergentes "au voisinage de l'infini".



Bien entendu, ces calculs deviennent plus difficiles plus on s'approche de l'origine. Pour des cercles de rayon très petit, il faudrait augmenter le nombre de termes utilisés dans les séries.

Au voisinage de l'origine, la série $\hat{f}_0$ est 1-sommable, ce qui signifie que l'on peut lui associer une fonction holomorphe, définie sur un secteur du plan complexe, unique, qui admette $\hat{f}_0$ comme développement asymptotique Gevrey, et que l'on peut calculer par divers procédés de sommation. Le procédé de sommation utilisé dans la suite est le plus simple qui existe, et connu sous le nom de *sommation au plus petit terme* ou *sommation des astronomes* [15]. Il permet de calculer $f_0(x)$ pour $|Arg(x)| < \pi$. Pour calculer $f_0(x)$ à l'extérieur de ce secteur, on utilise la relation $f_0(xe^{2i\pi}) = f_0(x) + 2i\pi e^{\frac{1}{x}}$, ce qui revient à exprimer f_0 grâce à une combinaison linéaire *variable* des fonctions $\hat{f}_0$ et $\hat{f}_1$. Une telle combinaison est décrite par la procédure *combli_Euler*.
Bien entendu, ces calculs sont plutôt valables au voisinage de l'origine. Nous allons donc les tester et les comparer à ceux obtenus par combinaison des séries au voisinage de l'infini sur un cercle de rayon *intermédiaire*, en l'occurence 0.2. L'image calculée en sommant la série d'Euler est affichée en trait pointillé.

```
> circ1:=create_circle(0.2,-2*Pi,4*Pi,150):
```

```
> seq2[1]:=compute_solution(sol2,[-gamma,-1.],circ1):
```

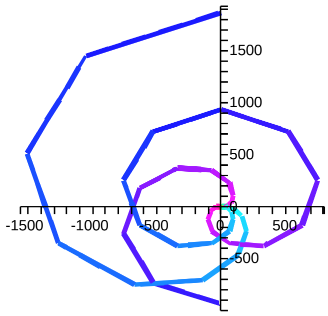
21
21

```
> seq2[2]:=compute_solution(sol1,combli_Euler,circ1,linestyle=3):
```

5[4]
5

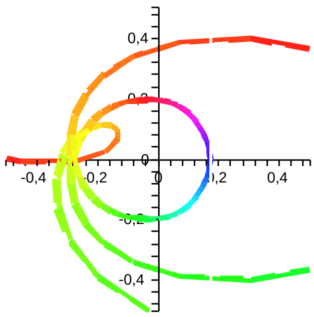
```
> draw_xy([seq(op(seq2[i]),i=1..2)],"");
```

FIG. 16 – images d'un arc de cercle de rayon 0.2.




```
> draw_xy([seq(op(seq2[i]),i=1..2)],"",-0.5,0.5,-0.5,0.5);
```

FIG. 17 – zoom



Les courbes se superposent de façon presque parfaite : les deux méthodes donnent des résultats précis pour un tel cercle. A noter aussi que même si celle-ci n'est pas affichée sur les dessins, l'image que l'on aurait obtenue en utilisant $f_0(x) = e^{\frac{1}{x}} Ei(1, 1/x)$ coïncide également avec les courbes précédentes.

Sur le cercle de rayon 0.5, les résultats sont encore assez bons, sauf près de l'axe réel négatif.

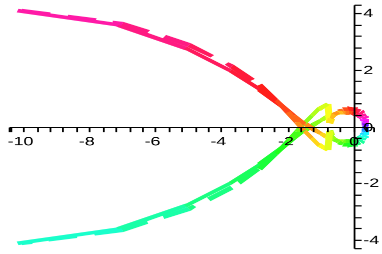
```
> seq22:=compute_solution(sol1,combli_Euler,circ[6],linestyle=3):
```

2[5]
2

```
> draw_xy([op(seq1[6]),op(seq22)],"");
```

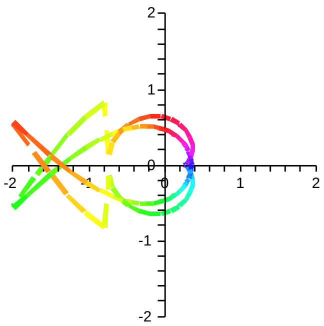
[4] Seulement les 5 premiers termes *des* séries sont utilisés.
[5] A noter que les résultats suivants sont obtenus en utilisant seulement les 2 premiers termes des séries !

FIG. 18 – image d'un arc de cercle de rayon 0.5.




```
>  draw_xy([op(seq1[6]),op(seq22)],"", -2.,2.,-2.,2.);
```

FIG. 19 – zoom



6 Intégration numérique dans le plan complexe

Dans ce paragraphe, nous allons comparer les résultats obtenus précédemment avec ceux obtenus par intégration numérique le long du chemin antécédent. Dans ce but, nous utilisons deux méthodes de transformation d'une équation en variable complexe en un système d'équations en la variable réelle qui paramètre le chemin antécédent.

La procédure *compute_num({eq,ICs},y1,opt,liste_de_points)* permet de calculer l'image de la *liste_de_points* par la solution de l'équation différentielle *eq* vérifiant les conditions initiales *ICs*. La variable *y1* désigne la fonction inconnue. Le paramètre *opt* donne le choix entre un chemin antécédent de type circulaire (*opt*='RHO') ou linéaire (*opt*='THETA'). Les conditions initiales sont données au premier point de *liste_de_points*. Deux paramètres optionnels permettent de fixer la méthode d'intégration numérique (par défaut rkf45) et le type de trait.

Notons x la variable complexe initiale, t la variable réelle d'intégration et $\gamma(t)$ la fonction décrivant le chemin antécédent. Notons $y_1(t)$ et $y_2(t)$ les parties réelles et imaginaires de la fonction inconnue $y(\gamma(t))$. Dans cette première procédure, le système intégré est celui vérifié par $u_k(t) = \dfrac{d^k y_1(t)}{dt^k}$, $v_k(t) = \dfrac{d^k y_2(t)}{dt^k}$.

La procédure *compute_num1({eq,ICs},y1,opt,liste_de_points)* réalise le même travail en posant $y_k(x) = \dfrac{d^k y(x)}{dx^k}$, et en intégrant le système différentiel vérifié par $u_k(t) = \Re(y_k(\gamma(t)))$ et $v_k(t) = \Im(y_k(\gamma(t)))$ [13].

Voici les résultats obtenus.

```
>  circ2:=create_circle(0.5,-3.1,5.1,60):
>  rho0:=circ2[1][1]:theta0:=circ2[1][2]:
```

```
> x0:=rho0*(cos(theta0)+I*sin(theta0));
```

$$x0 := -0.4995675752 - 0.02079033122\,I$$

Les conditions initiales sont évaluées au premier point de l'arc de cercle grâce à l'expression anlytique de la fonction f_0.

```
> y0:=exp(1/x0)*Ei(1,1/x0);
```

$$y0 := -0.6356750428 - 0.4386302374\,I$$

```
> yy0:=evalc(subs(x=x0,diff(exp(1/x)*Ei(1,1/x),x)));
```

$$yy0 := 0.681419557 + 1.620343872\,I$$

```
> ICs:=y(x0)=y0,D(y)(x0)=yy0:
> seq2:=compute_num({eq2, ICs},y(x),'RHO',circ2):
```

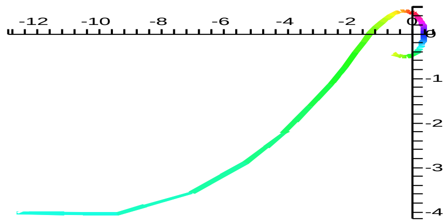
$$\{\frac{d}{d\theta}\,\mathrm{u1}(\theta) = -2.\,\mathrm{v0}(\theta)\sin(\theta) - 2.\,\mathrm{u1}(\theta)\sin(\theta) + 2.\,\mathrm{v1}(\theta)\cos(\theta) - 2.\,\mathrm{u0}(\theta)\cos(\theta),$$
$$\frac{d}{d\theta}\,\mathrm{v1}(\theta) = -2.\,\mathrm{v1}(\theta)\sin(\theta) + 2.\,\mathrm{u0}(\theta)\sin(\theta) - 2.\,\mathrm{v0}(\theta)\cos(\theta) - 2.\,\mathrm{u1}(\theta)\cos(\theta),$$
$$\frac{d}{d\theta}\,\mathrm{v0}(\theta) = \mathrm{v1}(\theta),\ \frac{d}{d\theta}\,\mathrm{u0}(\theta) = \mathrm{u1}(\theta)\}$$

```
> seq22:=compute_num1({eq1,ICs},y(x),'RHO',circ2,linestyle=3):
```

$$\{\frac{d}{d\theta}\,\mathrm{u0}(\theta) = -2.000000000\sin(\theta)\,\mathrm{u0}(\theta) + 2.000000000\cos(\theta)\,\mathrm{v0}(\theta),\ \frac{d}{d\theta}\,\mathrm{v0}(\theta) =$$
$$-2.000000000\cos(\theta)\,\mathrm{u0}(\theta) - 2.000000000\sin(\theta)\,\mathrm{v0}(\theta) + 1.000000000\}$$

```
> draw_xy([op(seq2),op(seq22)],"");
```

FIG. 20 – image par f_0 d'un arc de cercle de rayon 0.5.



Pour un rayon antécédent de 0.5, les deux courbes images sont parfaitement superposées. On se propose de faire les mêmes calculs pour un arc de cercle de rayon 0.1, plus proche de la singularité.

```
> circ3:=create_circle(0.1,-3.1,5.1,150):
> rho0:=circ3[1][1]:theta0:=circ3[1][2]:
> x0:=rho0*(cos(theta0)+I*sin(theta0)):
> y0:=exp(1/x0)*Ei(1,1/x0):
> yy0:=evalc(subs(x=x0,diff(exp(1/x)*Ei(1,1/x),x))):
> ICs:=y(x0)=y0,D(y)(x0)=yy0;
```

$$ICs := \mathrm{y}(-0.09991351503 - 0.004158066243\,I) = -0.1129305001 - 0.005595792948\,I,$$
$$\mathrm{D}(y)(-0.09991351503 - 0.004158066243\,I) = 1.309143338 + 0.035118167\,I$$

```
> seq3:=compute_num({eq2, ICs},y(x),'RHO',circ3):
```

$$\{\frac{d}{d\theta}\,\mathrm{u0}(\theta) = \mathrm{u1}(\theta),\ \frac{d}{d\theta}\,\mathrm{v0}(\theta) = \mathrm{v1}(\theta),\ \frac{d}{d\theta}\,\mathrm{v1}(\theta) =$$
$$10.\,\mathrm{u0}(\theta)\sin(\theta) - 10.\,\mathrm{v1}(\theta)\sin(\theta) - 10.\,\mathrm{v0}(\theta)\cos(\theta) - 10.\,\mathrm{u1}(\theta)\cos(\theta),\ \frac{d}{d\theta}\,\mathrm{u1}(\theta)$$
$$= -10.\,\mathrm{v0}(\theta)\sin(\theta) - 10.\,\mathrm{u1}(\theta)\sin(\theta) - 10.\,\mathrm{u0}(\theta)\cos(\theta) + 10.\,\mathrm{v1}(\theta)\cos(\theta)\}$$

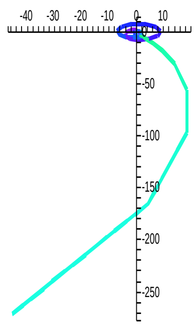
```
> seq33:=compute_num1({eq1, ICs},y(x),'RHO',circ3,linestyle=3):
```

[6] Cette deuxième procédure permet d'intégrer une équation non homogène. Ce qui explique la différence d'ordre du système.

$$\{\frac{d}{d\theta}\,\mathrm{u0}(\theta) = -10.00000000\sin(\theta)\,\mathrm{u0}(\theta) + 10.00000000\cos(\theta)\,\mathrm{v0}(\theta),\ \frac{d}{d\theta}\,\mathrm{v0}(\theta) = -10.00000000\cos(\theta)\,\mathrm{u0}(\theta) - 10.00000000\sin(\theta)\,\mathrm{v0}(\theta) + 1.000000000\}$$

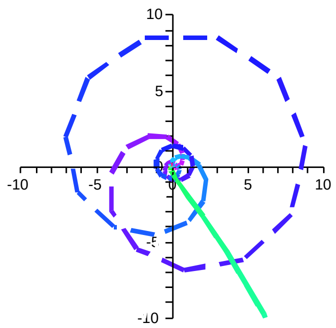
```
> draw_xy([op(seq3),op(seq33)],"");
```

FIG. 21 – image par f_0 d'un arc de cercle de rayon 0.1.



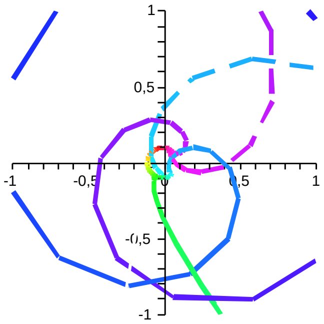

```
> draw_xy([op(seq3),op(seq33)],"",-10.,10.,-10.,10.);
```

FIG. 22 – zoom.




```
> draw_xy([op(seq3),op(seq33)],"",-1.,1.,-1.,1.);
```

FIG. 23 – zoom du précédent.



Pour un rayon antécédent de 0.1, on constate une déviation des deux méthodes numériques (plus forte pour la seconde), qui donnent des valeurs fausses pour les points voisins de l'axe réel positif. Cette déviation est à mettre en relation avec les figures 8 et 9, ce qui laisse penser que les méthodes numériques sont perturbées par la présence dans la solution générale de (1) du terme en $exp(1/x)$.
A noter enfin que la méthode numérique utilisée par défaut (rkf45) peut être remplacée par toutes les autres méthodes disponibles sous MAPLE sans que le comportement des courbes résultats soit fondamentalement différent.

7 Conclusion

Revenons au cadre général pour préciser dans quelle mesure les méthodes précédentes peuvent être également appliquées aux deux autres branches de la figure 1.

- **les méthodes formelles** qui conduisent à des développements en séries plus ou moins complexes reposent sur des algorithmes de Newton, eux-mêmes basés sur des figures géométriques (polyhèdre de Newton) qu'il faut savoir construire et manipuler d'un point de vue algorithmique. Des versions complètes des algorithmes correspondants existent pour le cas algébrique en une ou plusieurs variables [3] et pour le cas différentiel linéaire homogène. Des généralisations ont été étudiées pour le cas différentiel non linéaire [11, 5, 6, 8], et sont en cours pour les équations aux dérivées partielles [1].
  Des techniques similaires sont étudiées par A.D. Bruno depuis une trentaine d'années, et exposées de façon très complète dans [4]. Elles aboutissent à la description des comportements asymptotiques potentiels des solutions d'équations très générales (systèmes d'équations aux dérivées partielles, avec paramètres).
- **la sommation** de séries divergentes repose sur les caractéristiques Gevrey des dites séries, qui sont connues pour toutes les solutions d'équations différentielles linéaires d'ordre inférieur ou égal à 2. Ces techniques concernent donc une grande majorité des fonctions spéciales : fonctions de Bessel, fonctions d'Airy, fonctions hypergéométriques. Des méthodes de sommation plus sophistiquées (sommation de Borel, multi-sommation) pourraient également être mises en œuvre (comme elles l'ont été de manière expérimentale dans d'autres langages de programmation [12]).
- les fonctions de **visualisation graphique** ont été conçues pour permettre la représentation de solutions d'équations différentielles linéaires, mais aussi de fonctions définies par n'importe quelle série de puissance (le type powseries de MAPLE).

Références

1. F. Aroca, J. Cano, and F. Jung. Power series solutions for non-linear PDE's. In *International Symposium on Symbolic and Algebraic Computation, ISSAC' 03*, pages 15–22, Philadelphia, Pennsylvania, USA, August 2003. ACM.

2. M.A. Barkatou. Rational Newton algorithm for computing formal solutions of linear differential equations. In *Lecture Notes in Computer Science*, volume 358, proceeding of ISSAC 88, Italy, 1989.
3. F. Béringer and F. Richard-Jung. Multi-variate polynomials and Newton-Puiseux expansions. In F. Winkler and U. Langer, editors, *Symbolic and Numerical Scientific Computation, SNSC 01*, volume 2630 of *Lecture Notes in Computer Science*, pages 240–254, Hagenberg, Austria, September 2003. Springer.
4. A.D. Bruno. *Power Geometry in Algebraic and Differential Equation*. North-Holland Mathematical Library, 2000.
5. J. Cano. An extension of the Newton-Puiseux Polygon construction to give solutions of Pfaffian forms. *Annales de l'Institut Fourier*, 43(1) :125–142, 1993.
6. J. Cano. On the series defined by differential equations, with an extension of the Puiseux Polygon construction to these equations. *Analysis*, 13 :103–119, 1993.
7. J. Della Dora, C. Dicrescenzo, and E. Tournier. An algorithm to obtain formal solutions of a linear homogeneous differential equation at an irregular singular point. In *Lecture Notes in Computer Science*, volume 174, proceeding of Eurocam'82, Marseille, France. Springer Verlag.
8. J. Della Dora and F. Richard-Jung. About the Newton algorithm for non linear ordinary differential equations. In W. Küchlin, editor, *International Symposium on Symbolic and Algebraic Computation, ISSAC'97*, pages 298–304, Maui, Hawaii, July 1997. ACM Press.
9. J. Della Dora and F. Richard-Jung. An Integration and Reduction for Ordinary Nonlinear Differential Equations Library. In V.G. Ganzha, E.W. Meyr, and E.V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing, CASC'2000*, pages 113–128, Samarkand, Uzbekistan, October 2000. Springer.
10. J. Della Dora and E. Tournier. Formal solutions of differential equations in the neighborhood of singular points. In *SYMSAC 81, proceeding of the 1981 ACM Symposium on Symbolic and Algebraic Computation*. Edition Paul S. Wang.
11. D.Yu Grigorèv and M. Singer. Solving Ordinary Differential Equations in Terms of Series with Real Exponents. *Trans A.M.S.*, 1991.
12. F. Jung, F. Naegele, and J. Thomann. An algorithm of multisummation of formal power series, solutions of linear ODE equations. 42(4) :409–425, 1996.
13. F. Naegele. *Autour de quelques équations fonctionnelles analytiques*. PhD thesis, Institut National Polytechnique de Grenoble, 1995.
14. E. Pflügel. On the latest version of DESIR. *Theoretical Computer Science*, 187 :81–86, 1997.
15. J.P. Ramis and R. Schäfke. Gevrey separation of fast and slow variables. *Nonlinearity*, 9 :353–384, 1996.
16. F. Richard-Jung. *Représentations graphiques de solutions d'équations différentielles dans le champ complexe*. PhD thesis, Université Louis Pasteur, 1988.
17. Dominique Tournès. Intégration numérique des équations différentielles : influence des instruments et des pratiques de calcul. *MATAPLI*, (70), Janvier 2003.