

TRANSFORMEES INVERSES DE LAPLACE DES FRACTIONS RATIONNELLES

par Charles Hubert

La transformation de Laplace

A une fonction du temps $f(t)$ nulle pour $t < 0$, la transformation de Laplace fait correspondre une fonction $F(p)$ d'une variable complexe p . La transformation et son inverse sont exprimées par les relations

$$(A) \quad F(p) = \int_0^{\infty} f(t) e^{-pt} dp \quad f(t) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} F(p) e^{pt} dp$$

Dans (A.2) le chemin d'intégration est une droite et c doit être supérieur aux parties réelles de toutes les singularités de $F(p)$ ce qui entraîne $f(t)=0$ pour $t < 0$; l'intégrale (A.1) est définie pour $\text{Re}(p) > c$ et la fonction obtenue s'étend pour $\text{Re}(p) \leq c$ par prolongement analytique. Dans la suite on utilisera une minuscule pour chaque fonction de t et la majuscule correspondante pour sa transformée de Laplace.

Cette transformation est linéaire :

$$\text{Lap}[f(t)+g(t)] = F(p)+G(p) \quad \text{Lap}[a f(t)] = a F(p)$$

Elle transforme la dérivation par rapport à t en multiplication :

$$\text{Lap}[f'(t)] = p F(p) \quad \text{Lap}\left[\int_0^t f(\tau) d\tau\right] = \frac{1}{p} F(p)$$

Les équations différentielles sont alors transformées en équations algébriques.

Exemples :

$$\text{Lap}[1] = \frac{1}{p} \quad \text{Lap}[t] = \frac{1}{p^2} \quad \text{Lap}[e^{\alpha t}] = \frac{1}{p-\alpha}$$

$$\text{Lap}\left[e^{\alpha t} \cos \omega t\right] = \frac{p-\alpha}{(p-\alpha)^2 + \omega^2} \quad \text{Lap}\left[e^{\alpha t} \sin \omega t\right] = \frac{\omega}{(p-\alpha)^2 + \omega^2}$$

Un système dynamique possède en général une ou plusieurs entrées et une ou plusieurs sorties. Par exemple dans un avion, le braquage de la gouverne de profondeur est une entrée et l'altitude est une sortie ; ces grandeurs et d'autres sont reliées par les équations de la mécanique du vol.

La transformation de Laplace s'applique aux systèmes dynamiques invariants (les fonctions du temps ne figurent dans les équations qu'aux entrées) et linéaires. Un tel système étant initialement au repos (tout à zéro), si on applique à une entrée un signal $e(t)$ dont la transformée de Laplace est $E(p)$ et si $S(p)$ est la transformée de Laplace du signal $s(t)$ qui en résulte à une sortie, on démontre qu'il existe une fonction $T(p)$ indépendante de $E(p)$ telle que

$$S(p) = E(p) T(p)$$

cette fonction $T(p)$ est la fonction de transfert de l'entrée vers la sortie considérées. Une simple multiplication de fonctions donne la transformée de Laplace de la réponse à un signal donné.

Si le système obéit à un système différentiel linéaire à coefficients constants, cette fonction de transfert $T(p)$ est une fraction rationnelle. Si de plus $E(p)$ est aussi une fraction rationnelle, ce qui est fréquent au cours de l'étude de ces systèmes, alors $S(p)$ est aussi une fraction rationnelle ; sa transformée inverse $s(t)$ est la réponse en fonction du temps. L'intérêt du calcul pratique de la transformée inverse d'une fraction rationnelle en résulte.

Et la transformation de Fourier ?

Faisons ici une remarque. Il existe un parenté entre les transformations de Laplace et de Fourier mais leurs domaines d'application sont différents. La transformation de Fourier et son inverse peuvent être décrites en posant $p = \omega i$ dans les formules (A), en intégrant de $-\infty$ à $+\infty$ dans (A.1) et en prenant $c=0$ dans (A.2).

La transformation de Fourier s'applique à des fonctions $f(t)$ pouvant être différentes de zéro aussi bien pour $t < 0$ que pour $t > 0$. Elle s'applique quand une cause peut produire des effets dans les deux directions, qui tendent vers zéro à l'infini ; t est une variable d'espace.

La transformation de Laplace s'applique à des fonctions $f(t)$ nulles pour $t < 0$. Elle s'applique quand une cause ne peut produire des effets que dans une seule direction, qui ne tendent pas forcément vers zéro quand t tend vers $+\infty$; t est le temps et une cause n'a aucun effet vers le passé.

Calcul de la transformée inverse d'une fraction rationnelle

Considérons par exemple un système dont la fonction de transfert est

$$T(p) = \frac{15}{15 + 17p + 7p^2 + p^3}$$

On lui applique un échelon d'amplitude 2 : à l'instant 0 l'entrée passe de la constante 0 à la constante 2. On déduit d'un exemple précédent que sa transformée de Laplace est $E(p) = 2/p$. La transformée de Laplace de la sortie est donc

$$(B) \quad S(p) = E(p) T(p) = \frac{30}{15p + 17p^2 + 7p^3 + p^4}$$

Pour calculer numériquement la transformée inverse $s(t)$, il faut représenter cette fraction sous la forme d'une matrice à deux lignes : numérateur puis dénominateur. On représente le polynôme de chaque ligne par tous ses coefficients, zéros compris, suivant les puissances croissantes. Il faut donc que S soit sous la forme

$$\begin{bmatrix} 30 & 0 & 0 & 0 & 0 \\ 0 & 15 & 17 & 7 & 1 \end{bmatrix}$$

qu'on peut générer par l'expression APL

$$\Rightarrow 30 (0 \ 15 \ 17 \ 7 \ 1)$$

La transformée inverse de $s(t)$ pour $t \in [0,4]$ subdivisé en 200 intervalles égaux s'obtient par l'expression

```
s← 0 4 200 FntLap⇒ 30 (0 15 17 7 1)
```

si on veut que le résultat soit affecté à une variable s . Ce résultat a la forme d'une matrice à deux colonnes : les valeurs de t dans $s[;0]$, puis celles de $s(t)$ correspondantes dans $s[;1]$. Vérification avec l'expression analytique (en origine 0) :

```

R
R(C)      s(t) = 2 -5 e-3t + e-2t (3 cos t -9 sin t)
      [ / | s[;1] - 2 + (5 * 3 * s[;0]) + (2 * s[;0]) * (3 * 2 * cos[;0]) - 9 * 1 * cos[;0]
1.998401444E-15

```

qui est ainsi le maximum des valeurs absolues des différences entre les valeurs de $s(t)$ calculées par FntLap et celles calculées par l'expression qui développe la formule analytique.

L'une des bornes de l'intervalle temporel n'est pas obligatoirement zéro ; on peut calculer

```
2 5 200 FntLap ... ou encore ^1 5 400 FntLap ...
```

si l'on veut. Quand zéro fait partie de l'intervalle temporel, il figure deux fois dans le résultat, pour $0-\varepsilon$ et $0+\varepsilon$, afin de rendre compte de la discontinuité éventuelle de la transformée inverse. Autre exemple :

```

^-0.25 1 5 FntLap ⇒ 1 (0.5 1)
^-0.25 0
0 0
0 1
0.25 0.8824969026
0.5 0.7788007831
0.75 0.6872892788
1 0.6065306597

```

et puisque

$$\frac{1}{0.5+p} = \text{Lap} \left[e^{-0.5t} \right]$$

on peut vérifier :

```

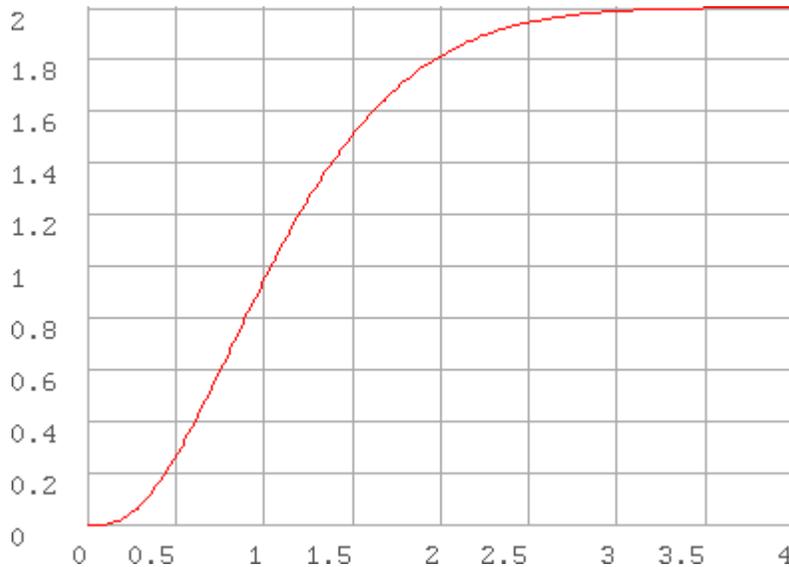
*^-0.5× 0 0.25 0.5 0.75 1
1 0.8824969026 0.7788007831 0.6872892788 0.6065306597

```

Bien entendu les arguments droit et gauche de FntLap peuvent résulter d'expressions calculées, et le résultat de FntLap, au lieu d'être affecté à une variable, peut servir d'argument à une fonction :

```
Graf 0 Plot 0 4 200 FntLap⇒ 30 (0 15 17 7 1)
```

par exemple, si 0 Plot construit le graphe de $s(t)$ et si Graf l'affiche à l'écran.



L'argument peut ne contenir qu'une valeur ; alors FntLap donne la matrice correspondant à cette valeur de t :

```
0.75 FntLap => 1 (0.5 1)
0.75 0.6872892788
```

La fonction FntLap ne se limite pas à inverser une seule transformée de Laplace ; elle se généralise de deux façons.

1) FntLap permet de calculer en même temps les transformées inverses de plusieurs fractions qui ont le même dénominateur. Dans un système invariant linéaire ayant plusieurs entrées et-ou plusieurs sorties, il y a plusieurs fonctions de transfert ; celles-ci ont le même dénominateur et cette généralisation de FntLap s'y applique. Le dénominateur commun doit se trouver dans la dernière ligne de l'argument droit et les numérateurs dans les autres lignes. Exemple avec deux numérateurs :

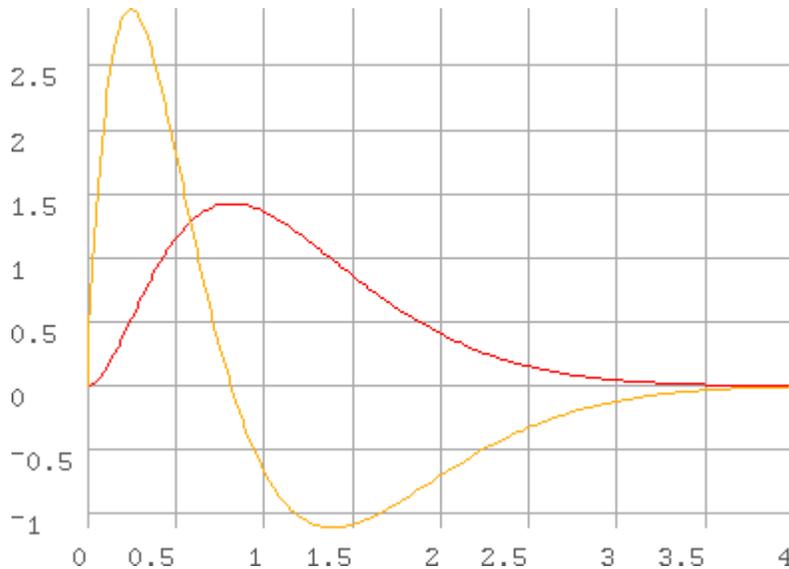
```
=> (0 30) (0 0 30) (0 15 17 7 1)
0 30 0 0 0
0 0 30 0 0
0 15 17 7 1
```

si bien que

```
s<- 0 4 200 FntLap=> (0 30) (0 0 30) (0 15 17 7 1)
```

calcule en même temps les transformées inverses de

$$\frac{30p}{15p^2 + 17p^3 + 7p^4 + p^4} \quad \text{et} \quad \frac{30p^2}{15p^2 + 17p^3 + 7p^4 + p^4}$$



et place les résultats dans s, matrice à trois colonnes : le temps t puis les deux transformées inverses s'(t) et s''(t) ; le nombre de colonnes de s est égal au nombre de lignes de l'argument droit. Vérification avec les expressions analytiques :

```

R
R      -3t      -2t
R      s'(t) = 15 e      + e      (15 sin t  -15 cos t)
R      Γ/|s[;1]-(15*^-3*s[;0])+(^-2*s[;0])*(15*cos[;0])-15*2*cos[;0]
2.442490654E-15

```

```

R
R      -3t      -2t
R      s''(t) = -45 e      + e      (45 cos t  -15 sin t)
R      Γ/|s[;2]-(-45*^-3*s[;0])+(^-2*s[;0])*(45*2*cos[;0])-15*1*cos[;0]
8.437694987E-15

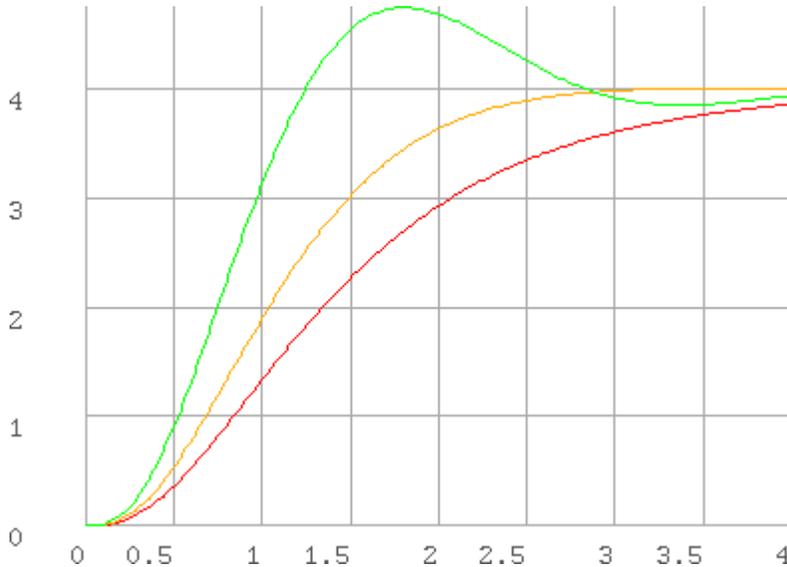
```

2) FntLap accepte dans l'argument droit une matrice gigogne à condition que les tableaux contenus dans les cases de cette matrice correspondant à une même fraction rationnelle aient des structures compatibles. FntLap calcule en même temps les transformées inverses des fractions dont les coefficients ont les valeurs contenues dans les cases correspondantes. Cela permet d'étudier l'influence sur la réponse de la variation des paramètres d'un système. Par exemple l'argument droit de

$$\Leftarrow 0 \ 4 \ 200 \text{ FntLap} \Rightarrow (c \ 40 \ 60 \ 100) \ (0 \ (10 \ 15 \ 25)(16 \ 17 \ 15) \ 7 \ 1)$$

décrit une fraction dont le numérateur prend trois valeurs et dont le dénominateur prend trois valeurs aux degrés 1 et 2 et une valeur aux degrés 0, 3 et 4 ; cela correspond aux trois fractions

40	60	100
$\frac{2 \quad 3 \quad 4}{10p + 16p + 7p + p}$	$\frac{2 \quad 3 \quad 4}{15p + 17p + 7p + p}$	$\frac{2 \quad 3 \quad 4}{25p + 15p + 7p + p}$



Après exécution s est une matrice à deux colonnes : t et s(t) ; chaque case de la colonne s(t) contient un vecteur à trois composantes, c'est la dimension résultant des dimensions compatibles des cases de l'argument droit. Vérification avec les expressions analytiques :

```

pour la fraction 40/... :
  A
  A      -t      -3t
  A      4 -8 e  + e      (4 cos t  +4 sin t)
  A      Γ/(0="s[;1])-4+(-8×*-s[;0])+(*-3×s[;0])×(4×2os[;0])+4×1os[;0]
  4.884981308E-15

pour la fraction 60/... :
  A
  A      -3t      -2t
  A      4 -10 e  + e      (6 cos t  -18 sin t)
  A      Γ/(1="s[;1])-4+(-10×*-3×s[;0])+(*-2×s[;0])×(6×2os[;0])-18×1os[;0]
  3.996802889E-15

pour la fraction 100/... :
  A
  A      -5t      -t
  A      4 -e  + e      (-3 cos 2t  -4 sin 2t)
  A      Γ/(2="s[;1])-4+(-*-5×s[;0])+(*-s[;0])×(-3×2o2×s[;0])-4×1o2×s[;0]
  4.884981308E-15

```

Les méthodes rejetées

Voyons les méthodes de calcul déconseillées et qu'on n'a donc pas programmées dans FntLap.

1) Pour calculer à la main l'expression analytique, pour $t > 0$, de la transformée inverse de Laplace d'une fraction rationnelle $F(p)$ la méthode

la plus simple consiste à faire la somme des résidus de $F(p)e^{pt}$ pour tous ses pôles c'est-à-dire les racines de son dénominateur. C'est toujours une combinaison linéaire de fonctions exponentielles-circulaires déterminées par les pôles, chaque coefficient est un polynôme dont le nombre de termes (degré moins un) est l'ordre du pôle correspondant. Par exemple les pôles de la fraction (B) sont 0, -3, -2+i, -2-i et nous avons vu dans les vérifications que sa transformée inverse (C) est une combinaison linéaire

de e^{0t} , e^{-3t} , e^{-2t+it} , e^{-2t-it} . Informatiquement, il faut calculer les racines du dénominateur ; cela se révèle difficile en cas de racines

multiples ou presque multiples, et dans ce dernier cas le calcul est amené à faire la différence de grands nombres, ce qui détériore la précision du résultat. De plus pour t proche de zéro, si c'est le but d'un calcul pratique, la précision est généralement dégradée ; on peut le voir sur l'exemple précédent, la formule

$$(C) \quad s(t) = 2 - 5e^{-3t} + e^{-2t} (3 \cos t - 9 \sin t)$$

calcule numériquement

$s(t) = 2 - (5+\dots) + (1+\dots)(3+\dots) = 0 + \dots$
 et les valeurs 2, 5, 1, 3 font perdre des bits significatifs utiles pour le résultat. Informatiquement l'expression (C) ne permet pas de trouver pour $t < 10^{-6}$ que

$$s(t) = 5t + \dots$$

2) En effectuant indéfiniment la division du numérateur de la fraction (B) par son dénominateur suivant les puissances croissantes de 1/p et en appliquant à chaque terme de la série obtenue la formule

$$\text{Lap} \left[\frac{t^n}{p} \right] = \frac{t^n}{n!}$$

on obtient d'abord le développement en série de la fraction du même exemple :

$$S(p) = 30p^{-4} - 210p^{-5} + 960p^{-6} - 3600p^{-7} + \dots$$

puis, terme à terme, celui de sa transformée inverse

$$s(t) = 5t^3 - \frac{35}{4}t^4 + 8t^5 - 5t^6 + \dots$$

On démontre que cette méthode est mathématiquement valable pour n'importe quelle fraction. On ne calcule pas les racines du dénominateur et on ne rencontre aucune difficulté due à leurs particularités. La méthode est précise au voisinage de zéro mais elle nécessite d'autant plus de termes que t est plus grand. Si t est grand la précision peut se dégrader parce que des termes de signes différents et de grandes valeurs absolues font perdre des bits significatifs utiles pour le résultat.

On peut le voir sur l'exemple simple

$$F(p) = \frac{1}{p+1} = p^{-1} - p^{-2} + p^{-3} - p^{-4} + \dots + (-1)^n p^{-n-1} + \dots$$

qui donne

$$f(t) = 1 - \frac{t}{1!} + \frac{t^2}{2!} - \frac{t^3}{3!} + \dots + (-1)^n \frac{t^n}{n!} + \dots$$

-t

ce qui développe e^{-t} bien entendu.

Pour t=15 par exemple il faut prendre 62 termes dont le plus grand vaut

$$\frac{\Gamma(15+1)}{15!} = 334864.6277$$

La somme et la bonne valeur sont

$$2.1 \rho(15+1) = 3.059249138E^{-7}$$

$$3.059023205E^{-7}$$

L'erreur relative

$$^{-1+(*15)*+/(^{-15*162})+162} \\ 7.385803447E^{-5}$$

est loin de la précision qu'on pourrait attendre de l'ordinateur.
L'erreur relative de FntLap est

$$^{-1+(*15)*(15 \text{ FntLap } \Rightarrow 1(1 \ 1))[;1]} \\ ^{-2.220446049E^{-16}}$$

3) Toujours à propos du même exemple considérons la solution du système différentiel

$$w' = x \quad x' = y \quad y' = z \quad z' = -7z - 17y - 15x$$

avec les conditions initiales

$$w(0) = 0 \quad x(0) = 0 \quad y(0) = 0 \quad z(0) = 1$$

La transformée inverse de (B) est alors

$$s = 30w$$

On peut appliquer à ce système différentiel une méthode numérique d'intégration. Ici aussi on ne se soucie pas des racines du dénominateur, mais on ne profite ni de l'invariance ni de la linéarité du système. La précision obtenue, plus grossière que celle de l'arithmétique de l'ordinateur, et éventuellement la convergence du calcul dépendent du pas d'intégration et de la méthode choisie ; et quelle méthode choisir ? Cela dépendra généralement de la position des pôles dans le plan complexe.

La méthode choisie pour FntLap

Cette méthode commence par diviser les numérateurs et le dénominateur par le coefficient de la plus haute puissance de celui-ci pour que ce coefficient devienne égal à 1. On peut examiner la suite sur le même exemple non limitatif. Le système différentiel explicité dans la troisième méthode précédente peut être mis sous la forme matricielle

$$X' = A X$$

avec

$$A = \begin{bmatrix} . & 1 & . & . \\ . & . & 1 & . \\ . & . & . & 1 \\ 0 & -15 & -17 & -7 \end{bmatrix} \quad X = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \quad X(0) = \begin{bmatrix} . \\ . \\ . \\ 1 \end{bmatrix}$$

les zéros par principe ont été notés ici par des points ; A est souvent appelée la matrice compagne du dénominateur.

Toute solution de ce système vérifie la relation

$$X(t+\theta) = \exp(\theta A) X(t) \quad (\forall t, \theta)$$

FntLap doit calculer

$$X(\tau), X(\tau+\theta), \dots X(\tau+k\theta), \dots X(\tau+n\theta)$$

Notons que les matrices de la forme $\exp(\theta A)$ pour toutes les valeurs de θ commutent entre elles.

Pour calculer $X(\tau+k\theta)$ avec $k \in \{0, 1, \dots, n\}$, FntLap calcule d'abord $\exp(\tau A)$ et $E = \exp(\theta A)$. Alors

$$X(\tau) = \exp(\tau A) X(0) = \text{la dernière colonne de } \exp(\tau A)$$

Par élévations au carré successives FntLap calcule E^2, E^4, E^8, \dots ce

qui donne $\exp(2\theta A)$, $\exp(4\theta A)$, $\exp(8\theta A)$, ...

Par simple multiplication matricielle E^k permet un décalage temporel $k\theta$.
 Par multiplications successives FntLap déduit de proche en proche les $X(\tau+k\theta)$:

τ par $E \Rightarrow \tau+\theta$
 $\tau, \tau+\theta$ par $E^2 \Rightarrow \tau+2\theta, \tau+3\theta$
 $\tau, \tau+\theta, \tau+2\theta, \tau+3\theta$ par $E^4 \Rightarrow \tau+4\theta, \tau+5\theta, \tau+6\theta, \tau+7\theta$
 etc...

Pour une raison de programmation, FntLap travaille avec les transposées des matrices ci-dessus. A la fin de FntLap la transformée inverse est calculée par combinaison linéaire des composantes des $X(\tau+k\theta)$, les coefficients étant tirés du numérateur de la transformée directe (l'argument droit).

Le calcul de $\exp(\tau A)$ et $\exp(\theta A)$ n'est pas fait en diagonalisant A car cela ramènerait au calcul des racines du dénominateur avec les difficultés signalées dans la première méthode précédente ; de plus en cas de racine multiple A n'est jamais diagonalisable. Ce calcul est assuré par une fonction PolExp (exponentielle de polynôme) parce qu'elle est appelée deux fois (τ et θ) dans FntLap et appelée aussi par d'autres fonctions. PolExp ne travaille pas sur la matrice τA elle-même mais profite de la structure polynômiale du dénominateur. Disons sommairement que PolExp procède par calcul d'un nombre suffisant de puissances successives de $x=\tau p$ modulo le dénominateur $D(x)$ et leur applique le développement en série de l'exponentielle limité aux puissances utiles.

Pour que le nombre de termes ne soit pas trop grand, PolExp calcule un majorant des modules de toutes les racines de $D(x)$ par une méthode très

simple et divise x par une puissance entière de 2, disons 2^m , afin que toutes les racines du polynôme ainsi modifié aient un module inférieur à $\log 2$; l'exponentielle obtenue est ensuite soumise à m élévations au carré successives afin d'obtenir l'exponentielle cherchée.

Pour que des bits utiles ne soient pas perdus dans le calcul de l'exponentielle, on décompose chaque nombre en deux parties : le multiple entier de $1/16$ le plus proche y_0 et le reste y_1 , de manière que $y = y_0 + y_1$; alors y_0 est représenté sans arrondi dans l'ordinateur. En fin de calcul seulement les deux parties de chaque nombre sont additionnées pour donner le nombre sous la forme habituelle. Pour voir l'utilité de cette décomposition on peut considérer le dénominateur

$$1 + p + 10^{-20} p^2$$

dont les racines -1 et -10^{-10} (à la précision de l'ordinateur) sont fortement disproportionnées. Si on rapporte la matrice A à ses directions propres, on voit que cela implique le calcul de $\exp(-\tau)$ et $\exp(-10^{-10} \tau)$. Si $\tau=0.01$ par exemple il faut trouver $\exp(-0.01)=0.9900498337$ et $\exp(-10^{-10})=0$.

Mais à cause de 10^{-18} , il faut diviser τ par 2^m avec $m=61$ ce qui entraîne $\exp(-\tau/2^{61})=1-\tau/2^{61}$ qui serait représenté par 1 exactement dans un seul nombre et 61 élévations au carré donneraient 1 au lieu de 0.9900498337. La

Exemple :

```
prod← ^1 8 7 ^10 ^3 PolMul 7 4 ^8 ^9
prod
^7 52 89 ^97 ^189 5 114 27
```

réalise

$$\begin{aligned} & (-1 + 8p^2 + 7p^3 - 10p^4 - 3p^5)(7 + 4p^2 - 8p^3 - 9p^4) \\ &= -7 + 52p^2 + 89p^3 - 97p^4 - 189p^5 + 5p^6 + 114p^7 + 27p^8 \end{aligned}$$

Cette fonction n'est appelée ni par FntLap ni par DecLap, mais elle est souvent utile dans les travaux où ces deux fonctions sont nécessaires.

Les arguments ne sont pas restreints à des vecteurs. Ils peuvent être des tableaux de rang plus élevé : la dernière dimension développe les coefficients de chaque polynôme, les autres dimensions décrivent une liste de polynômes. Ainsi un tableau de dimension 5 9 7 décrit une matrice à 5 lignes et 9 colonnes de polynômes à 7 coefficients. Les dimensions des polynômes (privées des nombres de coefficients) des deux arguments doivent être compatibles ; par exemple un tableau de dimension 5 9 7 est compatible avec un tableau de dimension 6 : ce sont une matrice de dimension 5 9 et un scalaire qui contiennent des polynômes et PolMul donnera un produit de dimension 5 9 12 (degré 6 + 5 = 11).

De plus les arguments peuvent être gigognes à la condition que les structures des cases des polynômes à multiplier soient compatibles.

La fonction calcule tous les produits des coefficients d'un argument par ceux de l'autre, et en même temps les degrés de ces produits. Elle rassemble ensuite les produits d'un même degré et en fait la somme.

Division de polynômes

La fonction PolDiv effectue la division du polynôme gauche par le polynôme droit suivant les puissances décroissantes et fournit le quotient et le reste dans les deux cases d'un vecteur de dimension 2 :

```
(quotient reste) ← numérateur PolDiv dénominateur
```

Exemple :

```
num← 2 ^7 34 ^4 30 ^50 ◊ den← ^4 1 ^1 5
(quo res)←num PolDiv den
```

En multipliant le dénominateur par le quotient et en ajoutant le reste on retrouve bien le numérateur :

```
num-+>(den PolMul quo) res
0 0 0 0 0 0
```

Les arguments ne sont pas restreints à des vecteurs. Ils peuvent être des tableaux de rang plus élevé : la dernière dimension développe les coefficients de chaque polynôme, les autres dimensions décrivent une liste de polynômes. Ainsi un tableau de dimension 5 9 14 décrit une matrice à 5 lignes et 9 colonnes de polynômes à 14 coefficients. Les arguments doivent être compatibles ; par exemple un tableau de dimension 5 9 12 est compatible avec un tableau de dimension 6 : ce sont une matrice de dimension 5 9 et un scalaire qui contiennent des polynômes et PolDiv

donnera un quotient de dimension 5 9 7 (degrés 11 - 5 = 6) et un reste de dimension 5 9 5 (degrés 5 - 1 = 4).

De plus les arguments peuvent être gigognes à la condition que les structures des cases des polynômes à diviser soient compatibles.

La fonction procède par itération comme on ferait manuellement.

Calcul sur un tableau gigogne

Pour pouvoir traiter un tableau gigogne, FntLap développe tous les dénominateurs décrits par l'argument droit afin de les traiter en toute sécurité. L'expression

```
den←c[0]⊃e''den+z-z←+/den
```

affecte à z un scalaire contenant un tableau de structure compatible avec toutes les cases du dénominateur, en déduit un tableau nul de même structure, répercute cette structure à toutes les cases du dénominateur, puis affecte à den la liste de tous les dénominateurs distincts. L'indice du dernier coefficient non nul de chaque dénominateur donne son degré ; les degrés sont placés dans d et les coefficients correspondants dans z. En divisant la fraction X par ces coefficients on ramène ceux-ci à 1. Pour supprimer les termes impulsionsnels (les parties entières) la fonction PolDiv calcule le reste de la division des numérateurs par les dénominateurs.

La suite de FntLap calcule $\exp(\tau A)$ et $\exp(\theta A)$ par PolExp puis les $\exp(\tau+k\theta)$ définis précédemment qu'elle place dans y.

La fin de FntLap réarrange dans z avec la bonne structure les $\exp(\tau+k\theta)$ contenus dans y, puis calcule les transformées inverses en multipliant par la matrice des numérateurs et achève en caténant une colonne de t.

La programmation de DecLap, PolMul et PolDiv utilise les mêmes recettes que FntLap pour gérer la structure des données.

Les fonctions

On peut les trouver dans le fichier APL*PLUS 'laplace.sf' ; la composante 1 contient la table des matières, les autres composantes contiennent les \square_{vr} de ces fonctions.

```

    ▽ x←a FntLap X;Dio;num;den;c;d;e;g;h;i;n;t;u;v;w;y;z
[1]  a∇ (t,x)←a FntLap X a [0]≤t≤a[1] en a[2] intervalles
[2]  a∇ X(p) = Lap(x(t)) ; transformée inverse de Laplace
[3]  Dio←0 ◊ (u v n)←3↑(2pa), (2,a), 1 ◊ →(0≠e←v-u)ρA ◊ n←0
[4]  A:den←c[0]◊e"den+z-z←+/den←XΓ-1+0◊pX;]
[5]  Derror(Γ-1ed←(Γ-1+0◊pX)-(0≠φ"den)u"1)/'DENOMINATEUR NUL'
[6]  (ez)←d◊"den ◊ X←X+z
[7]  num←1◊(Γ-1 0 ↓X)PolDiv den←(1+Γ/d)ρXΓ-1+0◊pX;] ◊ den←e"den
[8]  t←((0≥u×v)/ 0 0), ((uLv)+(w≠0)×((le)×u×e1+Γw+w←ln)~0
[9]  t←t[4t] ◊ →(0≤e)ρB ◊ t←φt
[10] B:y←(pt)ρcdp"0 ◊ (((-0≤e)φ 0 0 et)/y)←c(-d)↑-1
[11] →(ρi←(t>0)/Lpt)↓G ◊ →(0≤e)ρC ◊ i←φi
[12] C:c←PolExp den×g←φ1, h←x\(\Γ/d)ρu←t[↑i]
[13] y[1ρi]←c(-d)↑-1c[0]◊c[Γ-1+Γ/d;]×1↓g
[14] →(Γ-1+ρi)↓G ◊ →(0≠u-v←(le)+w)↓D ◊ c←PolExp den×φ1, h←x\(\Γ/d)ρv
[15] D:c←c(2ρ-d)↑-1c[0 1]◊c×h◊. +h
[16] d←1 ◊ h←1↓2ρi ◊ →F
[17] E:c←c+.x""c
[18] F:y[h]←y[(ph)ρi]+.x""c ◊ d←d+d
[19] →(0<ρh←d)↓((ρi)Ld+d)ρE
[20] G:y←c[1]◊◊y
[21] z←(py)ρz ◊ (ez)←ey
[22] x←t, ◊z+.x◊[0](1◊pZ)↑-1c[1]num
    ▽
    ▽ z←PolExp p;Dio;c;d;e;g;i;k;l;m;n;x;y
[1]  a∇ utilisé par FntLap Declap EchLap
[2]  Dio←0 ◊ g←3×2*47 ◊ p←◊e"p+z-z←+/p←, p
[3]  p←c[1](-c[0]0=ep)u"0)ep
[4]  p←◊(Γ-1↓p)◊-1↑p
[5]  (m 1)←pp ◊ (d n)←m+ -1 16
[6]  k←0.5*Γ2◊Γ/"(x←c[0](lp);-1)×c(+m-um), 0 ◊ i←u1 ◊ →B
[7]  A:k[i]←0.5×k[i]
[8]  B:→(0<ρi←(0<(1.4427×k[i])L"x[i])/i)ρA
[9]  a
    +◊2
[10] e← m 1 n ↑ 0 2 1 ⊗ m m 1 ρ((m×m)ρ(1+i←m)↑1)×c←x-1 0 ↓1;k←(pp)ρk
[11] x←e[;;d]
[12] C:x←k×(Γ-1 0 ↓0;x)-p×(pp)ρx[d;]
[13] e[;;i]←x ◊ →(n>i←i+1)ρC
[14] x←(c[1 0]e)◊.x""c[0]φ◊(um)↓-1cφ+×\1Γun)+[1 3]⊗c
[15] e←x[u2;];] ◊ i←u1 ◊ k←lρk ◊ →E
[16] D:x←c[2 3]e[;;i];] ◊ x←◊x[0 0 1 1 ;]+.x""x[0 1 0 1 ;]
[17] E:e[;;i];]←(x/y), [Γ-0.5]×x-y←(x+g)-g
[18] k←k+k
[19] →(0<ρi←(1.5≥k)/u1)ρD
[20] z←(Γ-2↑pe)ρz ◊ (ez)←, 2 0 1 ⊗x/e
    ▽
    ▽ L←d Declap L;Dio;num;den;i;y
[1]  a∇ L1←dt Declap L0 a L0(p) = Lap(f(t)) ; L1(p) = Lap(f(t+dt)u(t))
[2]  a∇ décalage temporel
[3]  Dio←0 ◊ Derror(0∈eΓ/0, |den←LΓ-1+0◊pL;])/'DENOMINATEUR NUL'
[4]  num←1◊(Γ-1 0 ↓L)PolDiv den
[5]  L←(PolExp(c"den)×φ1, y)×y◊. +y←x\1↓(ρden)ρcd←, d+y-y←+/den
[6]  (((ρeL)ρi)/eL)←(+/i←0=ed)/(ρ, L)ρ(1+py)↑1
[7]  (eL)←e◊[0 1](cρL)↑-1(2ρ""(0≠c[0]e◊e"c[0](d-d)◊. +den)u"1)↓-1c[0 1]◊e" L
[8]  L←(◊""c[0 1]◊L)+.x""(1◊pL)↑-1c[1]num, 0), cden
    ▽

```

```

      ▽ p←a PolMul b;Dio;Delx
[1]  n (A×B) ← A PolMul B  n en polynômes
[2]  Dio←0 ◊ Delx←'Derror(Ddm.Dtonl)ρDdm'
[3]  a←c[1]↓∪ρρa]a
[4]  b←c[1]↓∪ρρb]b
[5]  p←(, (ερ'' a b)↑(a, (ρa)↓0)◊.×b, (ρb)↓0), [0.5], (∪ρa)◊.+1+∪ρb
[6]  p←p[4p[1];1];] ◊ p←p[1]cρ[0]
[7]  p←(ρp)↑+/'p, (ρp)↓0
[8]  p←[∪ρρ↑p]p
      ▽
▽ x←a PolDiv b;Dio;c;i;n;p;q;r;z
[1]  n (quotient reste)←numérateur PolDiv dénominateur n en polynômes
[2]  Dio←0
[3]  a←c[(-1+ρρa)~1]φa
[4]  b←c[(-1+ρρb)~1]φb
[5]  →(0ερq←a, 'b)↓C ◊ x←2ρc=0ρ''q ◊ →0
[6]  C:z←z-z←+/'q
[7]  a←c[0]◊ε''c[0]◊, 'a+z
[8]  b←c[0]◊ε''c[0]◊, 'b+z
[9]  a←((0≠a)∪''1)↓''a
[10] b←((0≠b)∪''1)↓''b
[11] c←↑''b ◊ b←1↓''b
[12] n←ερ''b ◊ q←(ρb)ρc, 0 ◊ →B
[13] A:q[i]←(p←(↑''a[i])÷c[i]), ''q[i]
[14] a[i]←r-(ρ''r←1↓''a[i])↑''p×b[i]
[15] B:→(0<ρi←(n←ερ''a)/∪ρa)ρA
[16] a←[0]φ''a
[17] q← -1 0 ↓=[0]q
[18] x←(0=ρq)ρc=z ◊ (εx)←, q ◊ x←[∪ρρz]x
[19] r←(0=ρa)ρc=z ◊ (εr)←, a ◊ r←[∪ρρz]r
[20] x← x r
      ▽

```