Exponential Integrators for Fractional Partial Differential Equations

Roberto Garrappa

Department of Mathematics - University of Bari - Italy
roberto.garrappa@uniba.it

International Symposium on Fractional PDEs:
Theory, Numerics and Applications,
Newport (RI), June 3–5 2013
Outline

1. Exponential integrators for ODEs
2. Time–fractional PDEs
3. Exponential Integrators for fractional–order problems
4. Numerical approximation of the Mittag–Leffler function with matrix arguments
5. Numerical experiments
Exponential integrators for ODEs

\[ U'(t) = AU(t) + F(t, U(t)), \quad U(0) = U_0 \]

- the size of the system is very large \( (10^3 \sim 10^6) \)
- \( A \) is a sparse matrix
- \( A \) stiff but \( F(t, U(t)) \) non stiff: implicit or explicit methods?

Problems from spatial discretization of PDEs

Main idea of Exponential integrators

\[ U(t) = e^{tA}U_0 + \int_0^t e^{(t-\tau)A}F(\tau, U(\tau))d\tau \]

Device numerical schemes that use the evaluation of \( e^{tA} \)

Advantages: stability (stiff term evaluated exactly) with explicit schemes
Disadvantages: computation of \( e^{tA} \)

Non zero entries of A
Exponential integrators for ODEs

\[ U'(t) = AU(t) + F(t, U(t)), \quad U(0) = U_0 \]

- The size of the system is very large (\(10^3 \sim 10^6\))
- \(A\) is a sparse matrix
- \(A\) stiff but \(F(t, U(t))\) non-stiff: implicit or explicit methods?

Problems from spatial discretization of PDEs

Main idea of Exponential integrators

\[
U(t) = e^{tA}U_0 + \int_0^t e^{(t-\tau)A}F(\tau, U(\tau))d\tau
\]

Device numerical schemes that use the evaluation of \(e^{tA}\)

Advantages: stability (stiff term evaluated exactly) with explicit schemes

Disadvantages: computation of \(e^{tA}\)
Exponential integrators for ODEs: examples

\[ U(t) = e^{tA}U_0 + \int_0^t e^{(t-\tau)A} F(\tau, U(\tau)) d\tau \]

Integration on a grid \( t_n = nh \)

- \( F(\tau, U(\tau)) \) approximated by \( F_n = F(t_n, U_n) \) on \([t_n, t_{n+1}]\)
  - Exponential Euler: \( U_{n+1} = e^{hA}U_n + h\varphi_1(hA)F_n \)

- \( F(\tau, U(\tau)) \) approximated by linear interpolation \( F_n + \frac{t_n-\tau}{t_n-t_{n-1}}(F_n - F_{n-1}) \)
  - Second order E.I.: \( U_{n+1} = e^{hA}U_n + h\varphi_1(hA)F_n + h\varphi_2(hA)(F_n - F_{n-1}) \)

- Other approaches: Exponential Runge–Kutta, Exponential Adams methods, Exponential Rosenbrock-type methods, and so on ...

\[ \varphi_1(z) = \frac{e^z - 1}{z}, \quad \varphi_2(z) = \frac{e^z - z - 1}{z^2} \]
Exp. integrators

Marlis Hochbruck
Karlsruher Inst. für Technologie,
Institut für Angewandte und Numerische Mathematik,
D-76128 Karlsruhe,
Germany
E-mail: marlis.hochbruck@kit.edu

Alexander Ostermann
Institut für Mathematik,
Universität Innsbruck,
A-6020 Innsbruck,
Austria
E-mail: alexander.ostermann@uibk.ac.at

EXPINT: A MATLAB package for exponential integrators.1


Fractional partial differential equation

Time–fractional diffusion

\[
\frac{\partial^{\alpha}}{\partial t^{\alpha}} u(t, x) = \nu \nabla^2 u(t, x) + f(t, x)
\]

I.C. : \( u(0, x) = u_0(x) \)

B.C. : \( u(t, x) = g(t, x), \ t > 0, \ x \in \partial \Omega \)

0 < \( \alpha < 1 \): fractional diffusion equation

1 < \( \alpha < 2 \): fractional wave equation

Papers of Schneider and Wyss (1989)\(^2\) and Mainardi (1996)\(^3\)


From a FPDE to a system of FDEs: the linear case

\[
\frac{\partial^\alpha}{\partial t^\alpha} u(t, x) = \nu \nabla^2 u(t, x) + f(t, x)
\]

\[
\frac{d^\alpha}{dt^\alpha} U(t) = AU(t) + F(t), \quad U(0) = U_0
\]

\[
A \longleftrightarrow \nu \nabla^2 \quad F(t) \longleftrightarrow \text{Linear source term and B.C.} \quad U_0 \longleftrightarrow \text{I.C.}
\]
From a FPDE to a system of FDEs: the linear case

\[
\frac{\partial^\alpha}{\partial t^\alpha} u(t, x) = \nu \nabla^2 u(t, x) + f(t, x)
\]

\[
\frac{d^\alpha}{dt^\alpha} U(t) = AU(t) + F(t), \quad U(0) = U_0
\]

\[A \longleftrightarrow \nu \nabla^2 \quad F(t) \longleftrightarrow \text{Linear source term and B.C.} \quad U_0 \longleftrightarrow \text{I.C.}\]

Finite difference methods

\[\nabla^2 u_{i,j}(t) \approx \frac{u_{i-1,j}(t) - 2u_{i,j}(t) + u_{i+1,j}(t)}{(\Delta x_1)^2} + \frac{u_{i,j-1}(t) - 2u_{i,j}(t) + u_{i,j+1}(t)}{(\Delta x_2)^2}\]

\[u_{i,j}(t) = u(t, x_{1i}, x_{2j})\]

\[U(t) = \left( u_{1,1}(t), u_{2,1}(t), \ldots, u_{N_1,1}(t), \ldots, u_{1,N_2}(t), \ldots, u_{N_1,N_2}(t) \right)^T\]
From a FPDE to a system of FDEs: the linear case

\[
\frac{\partial^\alpha}{\partial t^\alpha} u(t,x) = \nu \nabla^2 u(t,x) + f(t,x)
\]

\[
\frac{d^\alpha}{dt^\alpha} U(t) = AU(t) + F(t), \quad U(0) = U_0
\]

\[A \longleftrightarrow \nu \nabla^2 \quad F(t) \longleftrightarrow \text{Linear source term and B.C.} \quad U_0 \longleftrightarrow \text{I.C.}\]

Finite elements methods

\[
u(t, x) \approx u_h(t, x) = \sum_{j=1}^{N_h} U_j(t) \Phi_j(x)
\]

\[
\sum_{j=1}^{N_h} \frac{d^\alpha}{dt^\alpha} U_j(t)(\Phi_j, \Phi_k) + \sum_{j=1}^{N_h} U_j(t)(\nabla^2 \Phi_j, \nabla^2 \Phi_k) = (f, \Phi_k)
\]

\[k = 1, \ldots, N_h\]
From a FPDE to a system of FDEs: non linear case

\[ \frac{\partial^\alpha}{\partial t^\alpha} u(t, x) = \nu \nabla^2 u(t, x) + f(t, u(t, x)) \]

\[ \frac{d^\alpha}{dt^\alpha} U(t) = AU(t) + F(t, U(t)), \quad U(0) = U_0 \]

Examples of non linear time–fractional PDEs:
- Bonhoeffer–van der Pol and Brusselator\(^4\)
- other problems\(^5\)


Generalization of Exponential Integrators to FDEs

\[ \frac{d^\alpha}{dt^\alpha} U(t) = A \cdot U(t) + F(t) \]

\[ \frac{d^\alpha}{dt^\alpha} U(t) = A \cdot U(t) + F(t, U(t)) \]

Linear term: 
\( A \cdot U(t) \) stiff
\( A \) large and sparse

Source term: 
\( F(t) \) linear
\( F(t, U(t)) \) non linear but non–stiff

Main steps:

1. Derivation of a Variation–of–constant formula for FDEs
2. Application of a discretization scheme
3. Evaluation of a counterpart of the exponential function
Generalization of Exponential Integrators to FDEs

\[
\frac{d^\alpha}{dt^\alpha} U(t) = A \cdot U(t) + F(t)
\]

\[
\frac{d^\alpha}{dt^\alpha} U(t) = A \cdot U(t) + F(t, U(t))
\]

Linear term: \(A \cdot U(t)\) stiff  
\(A\) large and sparse  
Source term:  
\(F(t)\) linear  
\(F(t, U(t))\) non-linear but non-stiff

Main steps:
1. Derivation of a Variation–of–constant formula for FDEs
2. Application of a discretization scheme
3. Evaluation of a counterpart of the exponential function
The variation–of–constant formula for FDEs

\[
\frac{d^\alpha}{dt^\alpha} U(t) = AU(t) + F(t)
\]

\[
\downarrow \quad \text{in the Laplace transform domain}
\]

\[
s^\alpha \hat{U}(s) - s^{\alpha-1} U_0 = A\hat{U}(s) + \hat{F}(s)
\]

\[
\downarrow \quad \text{solve w.r.t. } \hat{U}(s)
\]

\[
\hat{U}(s) = s^{\alpha-1}(s^\alpha I - A)^{-1} U_0 + (s^\alpha I - A)^{-1} \hat{F}(s)
\]

\[
\downarrow \quad \text{in the time domain}
\]

V.o.C. : \[U(t) = e_{\alpha,1}(t; -A) U_0 + \int_0^t e_{\alpha,\alpha}(t - s; -A) F(s) ds\]

The kernel \(e_{\alpha,\beta}(t; A)\) and the Mittag–Leffler (ML) function

\[
e_{\alpha,\beta}(t; A) = \mathcal{L}^{-1} \left(s^{\alpha-\beta} \left(s^\alpha I + A\right)^{-1}\right) = t^{\beta-1} E_{\alpha,\beta}(-t^\alpha A)
\]

\[
E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}
\]
The variation–of–constant formula for FDEs

\[ \frac{d^{\alpha}}{d{t^{\alpha}}} U(t) = AU(t) + F(t) \]

\[ \downarrow \quad \text{in the Laplace transform domain} \]

\[ s^{\alpha} \hat{U}(s) - s^{\alpha-1} U_0 = A\hat{U}(s) + \hat{F}(s) \]

\[ \downarrow \quad \text{solve w.r.t. } \hat{U}(s) \]

\[ \hat{U}(s) = s^{\alpha-1}(s^{\alpha} I - A)^{-1} U_0 + (s^{\alpha} I - A)^{-1} \hat{F}(s) \]

\[ \downarrow \quad \text{in the time domain} \]

V.o.C. : \[ U(t) = e_{\alpha,1}(t; -A) U_0 + \int_0^t e_{\alpha,\alpha}(t - s; -A) F(s) ds \]

The kernel \( e_{\alpha,\beta}(t; A) \) and the Mittag–Leffler (ML) function

\[ e_{\alpha,\beta}(t; A) = \mathcal{L}^{-1} \left( s^{\alpha-\beta} \left( s^{\alpha} I + A \right)^{-1} \right) = t^{\beta-1} E_{\alpha,\beta}(-t^\alpha A) \]

\[ E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}. \]
Fractional exponential Euler method

1. Write the Variation–of–Constants (VoC) formula in a piecewise form

\[ U(t_n) = e_{\alpha,1}(t_n; -A)U_0 + \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} e_{\alpha,\alpha}(t_n - s; -A)F(s)ds \]

2. Approximate \( F(s) \) by a constant value

\[ F(s) \approx F_j = F(t_j), \quad s \in [t_j, t_{j+1}] \]

3. Exactly integrate

\[ U(t_n) = e_{\alpha,1}(t_n; -A)U_0 + h^\alpha \sum_{j=0}^{n-1} W_{n-j}F_j \]

\[ W_n = e_{\alpha,\alpha+1}(n; -h^\alpha A) - e_{\alpha,\alpha+1}(n-1; -h^\alpha A) \]
Adams Exponential Integrators for FDEs

1. Write the Variation–of–Constants (VoC) formula in a piecewise form

\[ U(t_n) = e_{\alpha,1}(t_n; -A)U_0 + \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} e_{\alpha,\alpha}(t_n - s; -A)F(s)ds \]

2. Approximate \( F(s) \) by a \( L \)-degree polynomial in each interval \([t_j, t_{j+1}]\)

\[ F(s) = F(t_j + \theta h) \approx \sum_{\ell=0}^{L} (-1)^\ell \binom{-\theta + 1}{\ell} \nabla^\ell F_{j+1} \]

- \( F_n = F(t_n) \)
- \( \nabla^\ell F_n \) : backward differences of order \( \ell \)
- \( \binom{-\theta + 1}{\ell} \) : binomial coefficients

3. Exactly integrate
Adams Exponential Integrators for FDEs

\[ U_n = S.T. + h^\alpha \sum_{j=L}^{n-1} \sum_{\ell=0}^{L} \varphi_{\alpha,\ell}(n-j; -h^\alpha A) \nabla^\ell F_{j+1} \]

\[
\varphi_{\alpha,\ell}(n; z) = (-1)^\ell \int_0^1 \left( -\theta + 1 \right)^\ell e_{\alpha,\alpha}(n - \theta; z) d\theta
\]

\[
= \sum_{k=0}^{\ell} \left( p_{\ell,k} e_{\alpha,\alpha+k+1}(n; z) - q_{\ell,k} e_{\alpha,\alpha+k+1}(n-1; z) \right)
\]

<table>
<thead>
<tr>
<th></th>
<th>( p_{0,0} )</th>
<th>( p_{0,1} )</th>
<th>( p_{0,2} )</th>
<th>( p_{0,3} )</th>
<th>( p_{0,4} )</th>
<th>( q_{0,0} )</th>
<th>( q_{0,1} )</th>
<th>( q_{0,2} )</th>
<th>( q_{0,3} )</th>
<th>( q_{0,4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \ell = 0 )</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \ell = 1 )</td>
<td>-1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \ell = 2 )</td>
<td>0</td>
<td>-12\frac{1}{2}</td>
<td>1</td>
<td></td>
<td></td>
<td>0</td>
<td>12\frac{1}{2}</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \ell = 3 )</td>
<td>0</td>
<td>-12\frac{1}{6}</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>12\frac{1}{3}</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( \ell = 4 )</td>
<td>0</td>
<td>-12\frac{1}{12}</td>
<td>-12\frac{1}{12}</td>
<td>12\frac{1}{2}</td>
<td>1</td>
<td>0</td>
<td>12\frac{1}{4}</td>
<td>12\frac{11}{12}</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Adams Exponential Integrators for FDEs

\[ U_n = S.T. + h^\alpha \sum_{j=L}^{n-1} \sum_{\ell=0}^{L} \varphi_{\alpha,\ell}(n - j; -h^\alpha A) \nabla^\ell F_{j+1} \]

Main advantages: accuracy and stability\(^6\)

- Order depends on the smoothness of the linear source term and not of the true solution
  \[ F(t) \in C^{L+1} \implies \text{Error} = O(h^{L+1}) \]

- Stiffness due to large values in \( \sigma(A) \) does not affect the stability

Some problems to solve:

- evaluation of \( \varphi_{\alpha,\ell}(n; -h^\alpha A) \) (and hence the ML functions)

---

Application in control theory problems

Linear time–invariant system of FDEs

\[
\begin{align*}
D_0^\alpha x(t) &= Ax(t) + Bu(t) \\
y(t) &= C^T x(t), \quad x(0) = x_0
\end{align*}
\]

\[A \in \mathbb{R}^{N \times N}, \quad B \in \mathbb{R}^{N \times m}, \quad C \in \mathbb{R}^{N \times p}\]

No need for direct evaluation of the state \(x(t)\):

\[V.o.C.: \quad y(t) = C^T e_{\alpha,1}(t; -A)x_0 + \int_0^t C^T e_{\alpha,\alpha}(t - s; -A)Bu(s)ds\]

Fractional Exp. Adams: \(y_n = S.T. + h^{\alpha} \sum_{j=L}^{n-1} \sum_{\ell=0}^{L} \varphi_{\alpha,\ell}(n - j; -h^{\alpha}A)\nabla^\ell u_{j+1}\)

\[\varphi_{\alpha,\ell}(n - j; Z) = C^T \sum_{k=0}^{\ell} (p_{\ell,k} e_{\alpha,\alpha+k+1}(n; Z) - q_{\ell,k} e_{\alpha,\alpha+k+1}(n - 1; Z))B\]

MIMO : external dimensions \(m\) and \(p\) smaller than the internal dimension \(N\)

SISO : scalar weights
From a FPDE to a system of FDEs: the non linear case

\[ \frac{\partial^\alpha}{\partial t^\alpha} u(t, x) = \nu \nabla^2 u(t, x) + f(t, u(t, x)) \]

\[ \frac{d^\alpha}{d t^\alpha} U(t) = AU(t) + F(t, U(t)), \quad U(0) = U_0 \]

Fractional exponential Euler scheme:

\[ U_n = e_{\alpha,1}(t_n; -A)U_0 + h^\alpha \sum_{j=0}^{n-1} W_{n-j}^{(1)}F_j \]

Fractional exponential trapezoidal scheme:

\[ U_n = e_{\alpha,1}(t_n; -A)U_0 + h^\alpha \tilde{W}_n F_0 + h^\alpha \left( \sum_{j=1}^{n-1} W_{n-j}^{(2)}F_j - W_0^{(2)}F_{n-2} + 2W_0^{(2)}F_{n-1} \right) \]

\[ W_{n}^{(1)} = e_{\alpha,\alpha+1}(n; -h^\alpha A) - e_{\alpha,\alpha+1}(n-1; -h^\alpha A) \]
\[ W_{0}^{(2)} = e_{\alpha,\alpha+2}(1; -h^\alpha A) \]
\[ W_{n}^{(2)} = e_{\alpha,\alpha+2}(n-1; -h^\alpha A) - 2e_{\alpha,\alpha+2}(n; -h^\alpha A) + e_{\alpha,\alpha+2}(n+1; -h^\alpha A) \]
\[ \tilde{W}_n = e_{\alpha,\alpha+2}(n-1; -h^\alpha A) + e_{\alpha,\alpha+1}(n; -h^\alpha A) - e_{\alpha,\alpha+2}(n; -h^\alpha A) \]
The Mittag–Leffler function

Introduced in 1903 by Magnus Gösta Mittag–Leffler

\[ E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}, \quad \alpha \in \mathbb{C}, \quad \mathcal{R}(\alpha) > 0 \]

The generalization with two parameters

\[ E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta \in \mathbb{C}, \quad \mathcal{R}(\alpha) > 0, \mathcal{R}(\beta) > 0 \]

A generalized Mittag–Leffler function

\[ e_{\alpha,\beta}(t; \lambda) = t^{\beta-1}E_{\alpha,\beta}(-t^\alpha \lambda) \]

\( t \) is the independent variable \quad \lambda is a real/complex scalar/matrix argument

We will focus on \( \alpha, \beta \in \mathbb{R} \) and \( \beta = \alpha + k, \ k = 0, 1, \ldots \)
The Mittag–Leffler function

Introduced in 1903 by Magnus Gösta Mittag–Leffler

\[ E_{\alpha}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}, \quad \alpha \in \mathbb{C}, \quad \Re(\alpha) > 0 \]

The generalization with two parameters

\[ E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta \in \mathbb{C}, \quad \Re(\alpha) > 0, \Re(\beta) > 0 \]

A generalized Mittag–Leffler function

\[ e_{\alpha,\beta}(t; \lambda) = t^{\beta-1}E_{\alpha,\beta}(-t^\alpha \lambda) \]

\( t \) is the independent variable \quad \lambda is a real/complex scalar/matrix argument

We will focus on \( \alpha, \beta \in \mathbb{R} \) and \( \beta = \alpha + k, \ k = 0, 1, \ldots \)
The ML function and FDEs

\[ e_{\alpha,\beta}(t; \lambda) = t^{\beta-1}E_{\alpha,\beta}(-t^\alpha \lambda) \quad E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)} \]

The ML function plays for FDEs the same role of the \text{exp} function for ODEs

ODE : \[ \begin{align*}
 Dy(t) + \lambda y(t) &= 0 \\
 y(0) &= y_0 \end{align*} \quad \text{True solution} \quad y(t) = e^{-t\lambda} y_0 \]

FDE : \[ \begin{align*}
 0 < \alpha < 1 \quad \begin{cases} 
 0 \mathcal{D}^{\alpha} y(t) + \lambda y(t) &= 0 \\
  y(0) &= y_0 \end{cases} \quad \text{True solution} \quad y(t) = e_{\alpha,1}(t; \lambda) y_0 \]

ML function generalizes the \text{exp} function \( e_{1,1}(t; \lambda) = e^{-t\lambda} \)

- Books of Podlubny (1999) and Mainardi (2011)
- Code mlf.m available in the File Exchange service of MATLAB Central
Numerical evaluation of the ML function

Some challenging problems: 
- fast evaluation of $e_{\alpha,\beta}(t; \lambda)$
- extension to matrix arguments

$$e_{\alpha,\beta}(t; \lambda) = t^{\beta-1}E_{\alpha,\beta}(-t^\alpha \lambda), \quad E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}$$

Convergence of the series can be extremely slow

The Laplace transform

$$\mathcal{L}\left(e_{\alpha,\beta}(t; \lambda); s\right) = \frac{s^{\alpha-\beta}}{s^\alpha + \lambda}$$

Numerical quadrature in the inversion formula

$$e_{\alpha,\beta}(t, \lambda) = \frac{1}{2\pi i} \int_C e^{st} \frac{s^{\alpha-\beta}}{s^\alpha + \lambda} ds,$$
Numerical evaluation of the ML function

Some challenging problems: fast evaluation of $e_{\alpha,\beta}(t; \lambda)$
extension to matrix arguments

$$e_{\alpha,\beta}(t; \lambda) = t^{\beta-1}E_{\alpha,\beta}(-t^\alpha \lambda), \quad E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}$$

Convergence of the series can be extremely slow

The Laplace transform

$$\mathcal{L}\left(e_{\alpha,\beta}(t; \lambda); s\right) = \frac{s^{\alpha-\beta}}{s^\alpha + \lambda}$$

Numerical quadrature in the inversion formula

$$e_{\alpha,\beta}(t, \lambda) = \frac{1}{2\pi i} \int_C e^{st} \frac{s^{\alpha-\beta}}{s^\alpha + \lambda} ds,$$
Numerical inversion of the LT: choice of the contour

\[ e_{\alpha,\beta}(t, \lambda) = \frac{1}{2\pi i} \int_{C} e^{st} \frac{s^{\alpha-\beta}}{s^{\alpha} + \lambda} ds, \]

Parabolic contour:

- simplicity of the contour
  \[ C : z(u) = \mu(iu + 1)^2 \]
- extension to matrices with real spectrum
- good accuracy with fast computation
- availability of error estimates for \( \exp \)
- extension to the Mittag–Leffler function

---


Numerical inversion of the LT on parabolic contours

\[ e_{\alpha,\beta}(t, \lambda) = \frac{1}{2\pi i} \int_{C} e^{st} \mathcal{E}_{\alpha,\beta}(s) ds, \quad \mathcal{E}_{\alpha,\beta}(s) = \mathcal{L}\left(e_{\alpha,\beta}(t, \lambda) ; s \right) = \frac{s^{\alpha-\beta}}{s^{\alpha} + \lambda} \]

Main steps in numerical inversion of LT on parabolic contours:

Determination of:

- Geometry parameter \( \mu \) for \( C : z(u) = \mu (iu + 1)^2 \)
- Number \( N \) of quadrature nodes \( z_k = \mu (ikh + 1)^2, k = -N, \ldots, N \)
- Step-size \( h \)

\[ e^{[h,N]}_{\alpha,\beta}(t, \lambda) = \frac{h}{2\pi i} \sum_{k=-N}^{N} e^{t z_k} z_k^{\alpha-\beta} (z_k^{\alpha} + \lambda)^{-1} \]
Numerical inversion of the LT on parabolic contours

\[ e_{\alpha,\beta}(t, \lambda) = \frac{1}{2\pi i} \int_C e^{st} E_{\alpha,\beta}(s) ds, \quad E_{\alpha,\beta}(s) = \mathcal{L}\left( e_{\alpha,\beta}(t, \lambda) ; s \right) = \frac{s^{\alpha-\beta}}{s^\alpha + \lambda} \]

Main steps in numerical inversion of LT on parabolic contours:

Determination of:
- Geometry parameter \( \mu \) for \( C : z(u) = \mu(iu + 1)^2 \)
- Number \( N \) of quadrature nodes \( z_k = \mu(ikh + 1)^2, k = -N, \ldots, N \)
- Step-size \( h \)

\[ e^{[h,N]}_{\alpha,\beta}(t, A) = \frac{h}{2\pi i} \sum_{k=-N}^{N} e^{tz_k} z_k z_k^{\alpha-\beta} (z_k^\alpha I + A)^{-1} \]
Numerical inversion of the LT on parabolic contours

\[ e_{\alpha,\beta}(t, \lambda) = \frac{1}{2\pi i} \int_{C} e^{st} E_{\alpha,\beta}(s) ds, \quad E_{\alpha,\beta}(s) = \mathcal{L}\left( e_{\alpha,\beta}(t, \lambda); s \right) = \frac{s^{\alpha-\beta}}{s^{\alpha} + \lambda} \]

Main steps in numerical inversion of LT on parabolic contours:

Determination of:
- Geometry parameter \( \mu \) for \( C : z(u) = \mu(iu + 1)^2 \)
- Number \( N \) of quadrature nodes \( z_k = \mu(i kh + 1)^2, \ k = -N, \ldots, N \)
- Step-size \( h \)

\[
e^{[h,N]}_{\alpha,\beta}(t, A) \cdot F_j = \frac{h}{2\pi i} \sum_{k=-N}^{N} e^{tz_k} z_k^{\alpha-\beta} (z_k^{\alpha} I + A)^{-1} \cdot F_j \quad \text{Solve } (z_k^{\alpha} I + A) y = F_j
\]
Numerical inversion of the LT on parabolic contours

Selection of contour and quadrature parameters. Main goal:

$$\text{Error} = e_{\alpha,\beta}(t, \lambda) - e_{[h,n]}^{\alpha,\beta}(t, \lambda) = DE + TE \approx \epsilon$$

Discretization error:
- \(DE = DE_+ + DE_-\)
  - \(DE_+ = \frac{M_+(c)}{e^{2\pi c h / h - 1}}\) \(0 < c \leq 1\)
  - \(DE_- = \frac{M_-(d)}{e^{2\pi d h / h - 1}}\) \(d > 0\)

Truncation error:
- \(TE = O(\|E_{\alpha,\beta}(z_N)\|)\)

\[M_+(c) = \max_{0 \leq r \leq c} \int_{-\infty}^{\infty} |E_{\alpha,\beta}(z(u+ir))| \, du\]
\[M_-(d) = \max_{0 \leq s \leq d} \int_{-\infty}^{\infty} |E_{\alpha,\beta}(z(u-is))| \, du\]
Singularities of the LT of the ML

\[
\mathcal{E}_{\alpha,\beta}(s) = \mathcal{L}\left(e_{\alpha,\beta}(t; \lambda); s\right) = \frac{s^{\alpha-\beta}}{s^\alpha + \lambda}
\]

\[0 < \alpha < 1 \quad \beta > \alpha\]

\[
DE_+ = \frac{M_+(c)}{e^{2\pi c / h} - 1} \quad \text{but} \quad M_+(c) \to \infty \quad \text{when} \quad c \to 1
\]
Error analysis for the ML function

\[ DE_+ \approx DE_- \approx TE \approx \varepsilon \]

\[
\text{Error} = M_+(c)e^{-2\pi c/h}
\]

\[ h = \frac{\sqrt{1 + 4c(1 + c)}}{N} \]

Balance \( e^{-2\pi c/\sqrt{1+4c(1+c)}} \) and \( M_+(c) \) to keep \( N \) at a low value\(^9\)

Krylov subspace methods

Originally devised to find the solution of

\[ Ax = b \]

- Iterative methods
- Look for approximation \( x_m \approx A^{-1}b \) in

\[ \mathcal{K}_m(A, b) = \text{span}\{ b, Ab, A^2b, \ldots, A^{m-1}b \} \]
Main steps in Krylov subspace methods

1. Construction of $\mathcal{K}_m(A, b) = \text{span}\{b, Ab, \ldots, A^{m-1}b\}$
2. Choice of $x_m \in \mathcal{K}_m(A, b)$
Krylov subspace methods

Main steps in Krylov subspace methods

1. Construction of $\mathcal{K}_m(A, b) = \text{span}\{b, Ab, \ldots, A^{m-1}b\}$
2. Choice of $x_m \in \mathcal{K}_m(A, b)$

1. Construction of $\mathcal{K}_m(A, b)$
   - Numerical instability with the basis $\{b, Ab, A^2b, \ldots, A^{m-1}b\}$
   - Need of orthogonal basis $v_i^T v_j = 0, i \neq j$

Arnoldi or Lanczos algorithm

$$\{v_1, v_2, \ldots, v_m\} = \mathcal{K}_m(A, b)$$

$\begin{align*}
v_1 &= \frac{b}{\|b\|_2} \\
\text{for } k = 1, 2, \ldots \\
H_{i,k} &= v_k^T A^T v_i, \quad i = 1, \ldots, k \\
\hat{v}_{k+1} &= A v_k - \sum_{i=1}^{k} H_{i,k} v_i \\
H_{k+1,k} &= \|\hat{v}_{k+1}\|_2 \\
v_{k+1} &= \hat{v}_{k+1}/H_{k+1,k}
\end{align*}$

end
Krylov subspace methods

Main steps in Krylov subspace methods

1. Construction of $\mathcal{K}_m(A, b) = \text{span}\{b, Ab, \ldots, A^{m-1}b\}$
2. Choice of $x_m \in \mathcal{K}_m(A, b)$

② Choice of $x_m \in \mathcal{K}_m(A, b)$
- Minimize the residual $r_m = \|b - Ax_m\|$
- Different norms
- GMRES, MINRES, FOM, CG, ...
Projections on Krylov subspaces

\[ V_m^T A V_m = H_m \in \mathbb{R}^{m \times m} \]

- \( V_m = [v_1, v_2, \ldots, v_m] \in \mathbb{R}^{N \times m} \) vectors of the basis \( \mathcal{K}_m(A, b) \)
- \( V_m^T V_m = I \in \mathbb{R}^{m \times m} \)
- \( H_m \) is upper Hessenberg
Projections on Krylov subspaces

\[ AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T \]

\[ e_{\alpha,\beta}(t; A)b \approx V_m e_{\alpha,\beta}(t; H_m) V_m^T b = \|b\|_2 V_m e_{\alpha,\beta}(t; H_m) e_1 \]
Krylov subspaces methods

Main features:

✓ Only matrix–vector products
✓ Exploit the sparsity of \( A \)
✓ Convergence in a finite number of steps
✓ Recent applications to the Mittag–Leffler function

---


Numerical test: 2D diffusion time–fractional equation

\[ \frac{\partial^\alpha}{\partial t^\alpha} u(t, x, y) = \nu \left( \frac{\partial^2}{\partial x^2} u(t, x, y) + \frac{\partial^2}{\partial y^2} u(t, x, y) \right) \]

\( \rightarrow \) 2D square domain : \( \Omega = [0, 1] \times [0, 1] \)

\( \rightarrow \) Initial conditions : \( u(0, x, y) = x(1 - x)y(1 - y) \)

\( \rightarrow \) Homogeneous boundary conditions : \( u(t, x, y) = 0, \ (x, y) \in \partial \Omega \)

Spatial discretization - Equispaced grid on \( \Omega \):

\[ \frac{d^\alpha}{dt^\alpha} U(t) = AU(t), \quad U(0) = U_0 \quad \implies \quad U(t) = e_{\alpha,1}(t; -A)U_0 \]
Numerical test: 2D diffusion time–fractional equation

Time simulation at $t = 1$ for $\alpha = 0.7$ and $\nu = 0.1$

Numerical Solution

Difference with GL

Spatial mesh–grid $N_x = N_y = 40$ - Step–size $h = 2^{-4} = 0.625 \times 10^{-1}$

Comparison with the implicit Grünwald–Letnikov scheme (first order)
Numerical test: 2D diffusion time–fractional equation

Time simulation at $t = 1$ for $\alpha = 0.7$ and $\nu = 0.1$

Spatial mesh–grid $N_x = N_y = 40$ - Step–size $h = 2^{-4} = 0.625 \times 10^{-1}$

Comparison with the implicit Grünwald–Letnikov scheme (first order)
Numerical test: 2D diffusion time–fractional equation

Time of execution for $t \in [0, 1]$

<table>
<thead>
<tr>
<th>Step-size $h$</th>
<th>Run-time (sec.) $N_x=20$</th>
<th>Run-time (sec.) $N_x=30$</th>
<th>Run-time (sec.) $N_x=40$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>$10^{0}$</td>
<td>$10^{0}$</td>
<td>$10^{0}$</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>$10^{-2}$</td>
<td>$10^{-2}$</td>
<td>$10^{-2}$</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
</tr>
</tbody>
</table>

EI = Exponential integrator
G-L = Grünwald–Letnikov
Numerical test: 1D time–fractional ADR equation

\[
\frac{\partial^\alpha}{\partial t^\alpha} u + \varepsilon \frac{\partial}{\partial x} u = \nu \frac{\partial^2}{\partial x^2} u + \frac{1}{2} u(u - \frac{1}{2})(u - 1)
\]

\[\rightarrow\] 1D domain : \( \Omega = [0, 1] \)
\[\rightarrow\] Initial conditions : \( u(0, x) = x(1 - x) \)
\[\rightarrow\] Boundary conditions : \( u(t, 0) = 0, \ u(t, 1) = 0 \)

Spatial mesh–grid \( N_x = 80 \)

\[
\frac{d^\alpha}{dt^\alpha} U(t) = AU(t) + F(t, U(t)), \ \ U(0) = U_0
\]
Numerical test: 1D time–fractional ADR equation

Time simulation at $t = 1$ for $\alpha = 0.7$ and $\nu = 0.1$

Numerical Solution

Difference with GL

Spatial mesh–grid $N_x = 80$ - Step–size $h = 2^{-6} = 0.156 \times 10^{-1}$

Comparison with the implicit Grünwald–Letnikov scheme (first order)
Numerical test: 1D time–fractional ADR equation

Time of execution for $t \in [0, 1]$

EI = Exponential integrator
G-L = Grünwald–Letnikov
Concluding remarks

- Exponential integrators seem to be promising for FPDEs
  - Good stability properties
  - Explicit schemes
  - Competitive computation with respect to classical methods
  - Higher accuracy for linear problems

- Open problems
  - Enhance the computation of the generalized ML functions
  - Efficient algorithms for the ML function with matrix arguments
Some references


