Pseudospectral roaming contour integral methods for convection-diffusion equations

Mattia Manucci
mattia.manucci@gssi.it

Gran Sasso Science Institute

2gglan 14th February, 2022.

With Nicola Guglielmi, Giancarlo Nino (GSSI L'Aquila), María López-Fernández (U. Málaga)

GRAN SASSO SCIENCE INSTITUTE
SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore

INδAM
GNCS

Problem formulation

$$\frac{\partial u}{\partial t}(x,t) = \mathcal{A}(x)[u(x,t)] + f(x,t)$$

Discretization in Space:

$$u'(t) = Au(t) + b(t), \quad u(0) = u_0 \tag{1}$$

▶How do we solve (1) when we are only interested in the solution at a given time t?

Time-Steps methods ⟶ expensive for high accuracy (small Δt) and/or large t.

▶ Alternative approach: solve with Laplace transform

$$\mathcal{L}[u'(t)] = z\hat{u} - u_0 = A\hat{u} + \hat{b}(z) \longrightarrow \hat{u}(z) = (z\mathrm{I} - A)^{-1}\left(u_0 + \hat{b}(z)\right)$$

Go back to time domain by Inverse Laplace transform:

$$u(t) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \mathrm{e}^{zt}\,\hat{u}(z)\,dz \tag{2}$$

Problem formulation

$$\frac{\partial u}{\partial t}(x,t) = \mathcal{A}(x)[u(x,t)] + f(x,t)$$

Discretization in Space:

$$u'(t) = Au(t) + b(t), \quad u(0) = u_0 \tag{1}$$

- How do we solve (1) when we are only interested in the solution at a given time t?

Time-Steps methods ⟶ expensive for high accuracy (small Δt) and/or large t.

- Alternative approach: solve with Laplace transform

$$\mathcal{L}[u'(t)] = z\hat{u} - u_0 = A\hat{u} + \hat{b}(z) \longrightarrow \hat{u}(z) = (z\mathrm{I} - A)^{-1}\left(u_0 + \hat{b}(z)\right)$$

Go back to time domain by Inverse Laplace transform:

$$u(t) = \frac{1}{2\pi \mathrm{i}}\int_{\Gamma} \mathrm{e}^{zt}\,\hat{u}(z)\,dz \tag{2}$$

Problem formulation

$$\frac{\partial u}{\partial t}(x,t) = \mathcal{A}(x)[u(x,t)] + f(x,t)$$

Discretization in Space:

$$u'(t) = Au(t) + b(t), \quad u(0) = u_0 \tag{1}$$

▶How do we solve (1) when we are only interested in the solution at a given time t?

Time-Steps methods ⟶ expensive for high accuracy (small Δt) and/or large t.

▶ Alternative approach: solve with Laplace transform

$$\mathcal{L}[u'(t)] = z\hat{u} - u_0 = A\hat{u} + \hat{b}(z) \longrightarrow \hat{u}(z) = (z\mathrm{I} - A)^{-1}\left(u_0 + \hat{b}(z)\right)$$

Go back to time domain by Inverse Laplace transform:

$$u(t) = \frac{1}{2\pi \mathrm{i}} \int_{\Gamma} \mathrm{e}^{zt}\hat{u}(z)\,dz \tag{2}$$

Problem formulation

$$\frac{\partial u}{\partial t}(x,t) = \mathcal{A}(x)[u(x,t)] + f(x,t)$$

Discretization in Space:

$$u'(t) = Au(t) + b(t), \quad u(0) = u_0 \tag{1}$$

▶How do we solve (1) when we are only interested in the solution at a given time t?

Time-Steps methods ⟶ expensive for high accuracy (small Δt) and/or large t.

▶ Alternative approach: solve with Laplace transform

$$\mathcal{L}[u'(t)] = z\hat{u} - u_0 = A\hat{u} + \hat{b}(z) \longrightarrow \hat{u}(z) = (z\mathrm{I} - A)^{-1}\left(u_0 + \hat{b}(z)\right)$$

Go back to time domain by Inverse Laplace transform:

$$u(t) = \frac{1}{2\pi \mathrm{i}} \int_{\Gamma} \mathrm{e}^{zt}\,\hat{u}(z)\,dz \tag{2}$$

Problem formulation

$$\frac{\partial u}{\partial t}(x,t) = \mathcal{A}(x)[u(x,t)] + f(x,t)$$

Discretization in Space:

$$u'(t) = Au(t) + b(t), \quad u(0) = u_0 \tag{1}$$

▶How do we solve (1) when we are only interested in the solution at a given time t?

Time-Steps methods ⟶ expensive for high accuracy (small Δt) and/or large t.

▶ Alternative approach: solve with Laplace transform

$$\mathcal{L}[u'(t)] = z\hat{u} - u_0 = A\hat{u} + \hat{b}(z) \longrightarrow \hat{u}(z) = (z\mathrm{I} - A)^{-1}\left(u_0 + \hat{b}(z)\right)$$

Go back to time domain by Inverse Laplace transform:

$$u(t) = \frac{1}{2\pi \mathrm{i}} \int_{\Gamma} \mathrm{e}^{zt}\hat{u}(z)\,dz \tag{2}$$

Problem formulation

$$\frac{\partial u}{\partial t}(x,t) = \mathcal{A}(x)[u(x,t)] + f(x,t)$$

Discretization in Space:

$$u'(t) = Au(t) + b(t), \quad u(0) = u_0 \tag{1}$$

▶How do we solve (1) when we are only interested in the solution at a given time t?

Time-Steps methods $\longrightarrow$ expensive for high accuracy (small Δt) and/or large t.

▶ Alternative approach: solve with Laplace transform

$$\mathcal{L}[u'(t)] = z\hat{u} - u_0 = A\hat{u} + \hat{b}(z) \longrightarrow \hat{u}(z) = (z\mathrm{I} - A)^{-1}\left(u_0 + \hat{b}(z)\right)$$

Go back to time domain by Inverse Laplace transform:

$$u(t) = \frac{1}{2\pi \mathrm{i}}\int_{\Gamma} \mathrm{e}^{zt}\hat{u}(z)\,dz \tag{2}$$

The integration contour

We need to identify an opportune contour Γ and then to construct a map $z : \mathbb{R} \longrightarrow \Gamma$ such as:

- Elliptic: [N. Guglielmi, M. López-Fernández, G. Nino]

$$z(x) = \begin{cases} \ell_1(x), & x \in \left[-\infty, -\frac{\pi}{2}\right] ; \\ (a_1 + a_2)\cos x + i(a_2 - a_1)\sin x + a_3, & x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] ; \\ \ell_2(x)\,, & x \in \left[\frac{\pi}{2}, +\infty\right] ; \end{cases}$$

  $\ell_{1,2}(x)$ upper and lower half-lines

- Parabolic: [N. Guglielmi, M. López-Fernández, M. M.]

$$z(x) = -x^2 - 2\mathrm{i}xa_1 + a_2,$$

- Hyperbolic: [N. Guglielmi, M. López-Fernández, M. M.]

$$z(x) = a_3 - a_2\sin(a_1)\cosh x - \mathrm{i}a_2\cos(a_1)\sinh x.$$

The integration contour

We need to identify an opportune contour Γ and then to construct a map $z : \mathbb{R} \longrightarrow \Gamma$ such as:

- Elliptic: [N. Guglielmi, M. López-Fernández, G. Nino]

$$z(x) = \begin{cases} \ell_1(x), & x \in \left[-\infty, -\frac{\pi}{2}\right] ; \\ (a_1 + a_2)\cos x + i(a_2 - a_1)\sin x + a_3, & x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] ; \\ \ell_2(x)\,, & x \in \left[\frac{\pi}{2}, +\infty\right] ; \end{cases}$$

$\ell_{1,2}(x)$ upper and lower half-lines

- Parabolic: [N. Guglielmi, M. López-Fernández, M. M.]

$$z(x) = -x^2 - 2\mathrm{i}xa_1 + a_2,$$

- Hyperbolic: [N. Guglielmi, M. López-Fernández, M. M.]

$$z(x) = a_3 - a_2\sin(a_1)\cosh x - \mathrm{i}a_2\cos(a_1)\sinh x.$$

The integration contour

We need to identify an opportune contour Γ and then to construct a map $z : \mathbb{R} \longrightarrow \Gamma$ such as:

- Elliptic: [N. Guglielmi, M. López-Fernández, G. Nino]

$$z(x) = \begin{cases} \ell_1(x), & x \in \left[-\infty, -\frac{\pi}{2}\right] ; \\ (a_1 + a_2)\cos x + i(a_2 - a_1)\sin x + a_3, & x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] ; \\ \ell_2(x)\,, & x \in \left[\frac{\pi}{2}, +\infty\right] ; \end{cases}$$

$\ell_{1,2}(x)$ upper and lower half-lines

- Parabolic: [N. Guglielmi, M. López-Fernández, M. M.]

$$z(x) = -x^2 - 2\mathrm{i}xa_1 + a_2,$$

- Hyperbolic: [N. Guglielmi, M. López-Fernández, M. M.]

$$z(x) = a_3 - a_2\sin(a_1)\cosh x - \mathrm{i}a_2\cos(a_1)\sinh x.$$

The integration contour

We need to identify an opportune contour Γ and then to construct a map $z : \mathbb{R} \longrightarrow \Gamma$ such as:

- Elliptic: [N. Guglielmi, M. López-Fernández, G. Nino]

$$z(x) = \begin{cases} \ell_1(x), & x \in \left[-\infty, -\frac{\pi}{2}\right] \; ; \\ (a_1 + a_2)\cos x + i(a_2 - a_1)\sin x + a_3, & x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \; ; \\ \ell_2(x)\,, & x \in \left[\frac{\pi}{2}, +\infty\right] \; ; \end{cases}$$

$\ell_{1,2}(x)$ upper and lower half-lines

- Parabolic: [N. Guglielmi, M. López-Fernández, M. M.]

$$z(x) = -x^2 - 2\mathrm{i}xa_1 + a_2,$$

- Hyperbolic: [N. Guglielmi, M. López-Fernández, M. M.]

$$z(x) = a_3 - a_2\sin(a_1)\cosh x - \mathrm{i}\,a_2\cos(a_1)\sinh x.$$

Trapezoidal rule for analytic functions

After parametrization, contour integration gives

$$u(t) = I \approx \frac{1}{2\pi \mathrm{i}} \int_{-c\pi}^{c\pi} F(z(x))\, dx\,, \quad 0 < c < c_{max}$$

with $F(z(x)) = \mathrm{e}^{z(x)t}\hat{u}(z(x))z'(x)$.

Integral approximation:

$$I_N = \frac{c}{\mathrm{i}N}\sum_{j=1}^{N-1} F(z(x_j)) \quad \text{with} \quad x_j = -c\pi + j\frac{2c\pi}{N}, \quad j = 1, \ldots, N-1.$$

Error:

$$\|u(t) - I_N\| \lessapprox \underbrace{\frac{P}{\mathrm{e}^{\frac{a}{c}N} - 1}}_{\text{quadrature err.}} + \underbrace{Mc\,tol}_{\text{truncation err.}} + \underbrace{\max_j \delta F(x_j)}_{\text{noise err.}}.$$

Note: each quadrature node $z(x_j)$ corresponds to the solution of the linear system $(z(x_j)\mathrm{I} - A)\hat{u} = u_0 + \hat{b}(z(x_j))$.

Trapezoidal rule for analytic functions

After parametrization, contour integration gives

$$u(t) = I \approx \frac{1}{2\pi \mathrm{i}} \int_{-c\pi}^{c\pi} F(z(x))\, dx\,, \quad 0 < c < c_{max}$$

with $F(z(x)) = \mathrm{e}^{z(x)t}\hat{u}(z(x))z'(x)$.

Integral approximation:

$$I_N = \frac{c}{\mathrm{i}N} \sum_{j=1}^{N-1} F(z(x_j)) \quad \text{with} \quad x_j = -c\pi + j\frac{2c\pi}{N}, \quad j = 1, \dots, N-1.$$

Error:

$$\|u(t) - I_N\| \lessapprox \underbrace{\frac{P}{\mathrm{e}^{\frac{a}{c}N} - 1}}_{\text{quadrature err.}} + \underbrace{Mc\,tol}_{\text{truncation err.}} + \underbrace{\max_j \delta F(x_j)}_{\text{noise err.}}.$$

Note: each quadrature node $z(x_j)$ corresponds to the solution of the linear system $(z(x_j)\mathrm{I} - A)\hat{u} = u_0 + \hat{b}(z(x_j))$.

Trapezoidal rule for analytic functions

After parametrization, contour integration gives

$$u(t) = I \approx \frac{1}{2\pi \mathrm{i}} \int_{-c\pi}^{c\pi} F(z(x))\, dx\,, \quad 0 < c < c_{max}$$

with $F(z(x)) = \mathrm{e}^{z(x)t}\hat{u}(z(x))z'(x)$.

Integral approximation:

$$I_N = \frac{c}{\mathrm{i}N} \sum_{j=1}^{N-1} F(z(x_j)) \quad \text{with} \quad x_j = -c\pi + j\frac{2c\pi}{N}, \quad j = 1, \ldots, N-1.$$

Error:

$$\|u(t) - I_N\| \lessapprox \underbrace{\frac{P}{\mathrm{e}^{\frac{a}{c}N} - 1}}_{\text{quadrature err.}} + \underbrace{M c\, tol}_{\text{truncation err.}} + \underbrace{\max_j \delta F(x_j)}_{\text{noise err.}}.$$

Note: each quadrature node $z(x_j)$ corresponds to the solution of the linear system $(z(x_j)\mathrm{I} - A)\hat{u} = u_0 + \hat{b}(z(x_j))$.

Trapezoidal rule for analytic functions

After parametrization, contour integration gives

$$u(t) = I \approx \frac{1}{2\pi \mathrm{i}} \int_{-c\pi}^{c\pi} F(z(x))\, dx\,, \quad 0 < c < c_{max}$$

with $F(z(x)) = \mathrm{e}^{z(x)t}\hat{u}(z(x))z'(x)$.

Integral approximation:

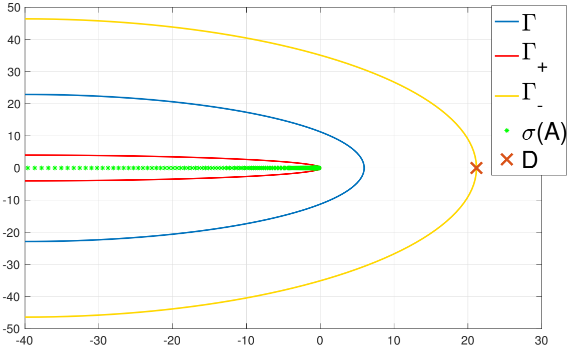
$$I_N = \frac{c}{\mathrm{i}N} \sum_{j=1}^{N-1} F(z(x_j)) \quad \text{with} \quad x_j = -c\pi + j\frac{2c\pi}{N}, \quad j = 1, \dots, N-1.$$

Error:

$$\|u(t) - I_N\| \lessapprox \underbrace{\frac{P}{\mathrm{e}^{\frac{a}{c}N} - 1}}_{\text{quadrature err.}} + \underbrace{Mc\,tol}_{\text{truncation err.}} + \underbrace{\max_j \delta F(x_j)}_{\text{noise err.}}\,.$$

Note: each quadrature node $z(x_j)$ corresponds to the solution of the linear system $(z(x_j)\mathrm{I} - A)\hat{u} = u_0 + \hat{b}(z(x_j))$.

The three integration contours



Three ellipses:

$$\Gamma_{right} \rightarrow \text{ bound on } D = \left|\mathrm{e}^{zt}\right| , \quad \Gamma_{left} \rightarrow \text{ bound on } \mathrm{e}^{\mathrm{Re}(z)t} \left\|(z\mathrm{I} - A)^{-1}\right\|$$

$$\Gamma \rightarrow \text{ Integration profile}$$

The weighted ϵ-Pseudospectrum

The $\varepsilon-$pseudospectrum is the set defined as:

$$\sigma_\varepsilon(A) := \left\{ z \in \mathbb{C} \,:\, \left\| (z\mathrm{I} - A)^{-1} \right\| > \tfrac{1}{\varepsilon} \right\}$$

We define the "weighted" $\varepsilon-$pseudospectrum as:

$$\sigma_{\varepsilon,t}(A) := \left\{ z \in \mathbb{C} \,:\, \mathrm{e}^{\mathrm{Re}(z)t} \left\| (z\mathrm{I} - A)^{-1} \right\| > \tfrac{1}{\varepsilon} \right\}$$

The boundary of this set, denoted as $\partial\sigma_{\varepsilon,t}(A)$, is crucial in the construction of the integration contour.

Recall that $\left\| (z\mathrm{I} - A)^{-1} \right\|^{-1} = \sigma_{\min}(z\mathrm{I} - A)$, $\sigma_{\min}$ smallest singular value.

The weighted ϵ-Pseudospectrum

The $\varepsilon-$pseudospectrum is the set defined as:

$$\sigma_{\varepsilon}(A) := \left\{ z \in \mathbb{C} \,:\, \left\| (z\mathrm{I} - A)^{-1} \right\| > \frac{1}{\varepsilon} \right\}$$

We define the “weighted” $\varepsilon-$pseudospectrum as:

$$\sigma_{\varepsilon,t}(A) := \left\{ z \in \mathbb{C} \,:\, \mathrm{e}^{\mathrm{Re}(z)t} \left\| (z\mathrm{I} - A)^{-1} \right\| > \frac{1}{\varepsilon} \right\}$$

The boundary of this set, denoted as $\partial\sigma_{\varepsilon,t}(A)$, is crucial in the construction of the integration contour.

Recall that $\left\| (z\mathrm{I} - A)^{-1} \right\|^{-1} = \sigma_{\min}(z\mathrm{I} - A)$, $\sigma_{\min}$ smallest singular value.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{\mathrm{e}^{-\phi t}\sigma_{\min}((\phi+\mathrm{i}\psi)\mathrm{I}-A)}_{\tilde{\sigma}} - \varepsilon \to 0; \qquad (3)$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{e^{-\phi t}\sigma_{\min}((\phi+i\psi)I-A)}_{\tilde{\sigma}} - \varepsilon \to 0; \tag{3}$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{\mathrm{e}^{-\phi t}\sigma_{\min}((\phi+\mathrm{i}\psi)\mathrm{I}-A)}_{\tilde\sigma}-\varepsilon\to 0; \tag{3}$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{\mathrm{e}^{-\phi t}\sigma_{\min}((\phi + \mathrm{i}\psi)\mathrm{I} - A)}_{\tilde{\sigma}} - \varepsilon \to 0; \tag{3}$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{\mathrm{e}^{-\phi t}\sigma_{\min}((\phi+\mathrm{i}\psi)\mathrm{I}-A)}_{\tilde{\sigma}}-\varepsilon \to 0; \qquad (3)$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{\mathrm{e}^{-\phi t}\sigma_{\min}((\phi+\mathrm{i}\psi)\mathrm{I}-A)}_{\tilde{\sigma}}-\varepsilon\to 0; \tag{3}$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{e^{-\phi t}\sigma_{\min}((\phi + i\psi)I - A)}_{\tilde{\sigma}} - \varepsilon \to 0; \qquad (3)$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{\mathrm{e}^{-\phi t}\sigma_{\mathsf{min}}((\phi + \mathrm{i}\psi)\mathrm{I} - A)}_{\tilde{\sigma}} - \varepsilon \to 0; \tag{3}$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{\mathrm{e}^{-\phi t}\sigma_{\text{min}}((\phi+\mathrm{i}\psi)\mathrm{I}-A)}_{\tilde{\sigma}}-\varepsilon \to 0; \tag{3}$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Weighted pseudospectral computation

Existing literature relying on the concept of pseudospectrum:

- `Eigtool` (Wright, 2002)→ Too expensive;
- Contour tracing methods (Brühl, 1996)→ Problems with disconnected components;
- Approximation of pseudospectral abscissa (Guglielmi, Overton 2011, Kressner, Vandereycken 2014)→ difficulties with ill conditioned eigenvalues Problems with `eigs` (ARPACK)

A computationally cheaper strategy:

- Section the complex plane with vertical lines;
- Search on each line with a fixed ϕ the highest intersection with $\sigma_\varepsilon(A)$

$$\underbrace{\mathrm{e}^{-\phi t}\sigma_{\mathsf{min}}((\phi+\mathrm{i}\psi)\mathrm{I}-A)}_{\tilde{\sigma}}-\varepsilon \to 0; \tag{3}$$

- Apply Newton's method exploiting a derivative formula for singular values (see for instance Kato, 1995) to find the zero of (3) moving ψ.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Summary of the method

- set initial data (A, b, u_0). Fix t, tol;
- compute Γ_{left} (based on pseudospectral computation);
- compute Γ (minimizing the number of quadrature nodes N to reach the target accuracy tol);
- compute the truncation parameter c;
- apply the quadrature formula.

Advantages:

- no a priori knowledge about the resolvent norm of A is needed;
- the profile of integration does not depend on N;
- the method is stable w.r.t. N;
- the method can be extended to time intervals $t \in [t_0, t_1]$;
- the main computational effort, i.e. the computation of the $\hat{u}(z_j)$, can be parallelized in a straightforward way;
- the method is designed to achieve a given target accuracy tol and check whether this is possible.

Black–Scholes equation

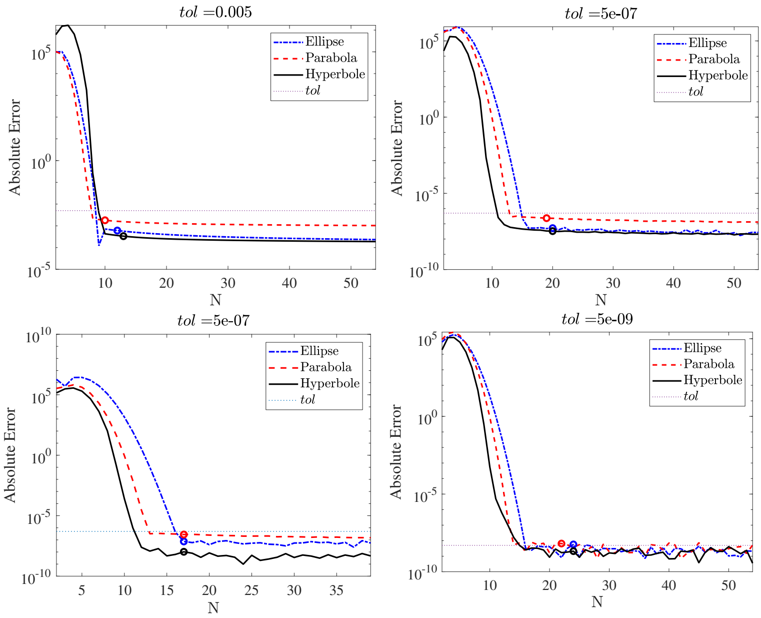
For $u = u(s, \tau)$,

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 u}{\partial s^2} + rs\frac{\partial u}{\partial s} - ru, \quad \tau \geq 0,\ L \leq s \leq S.$$

With initial and boundary conditions

$$\begin{aligned} u(s,0) &= \max(0, s-K) \\ u(0,\tau) &= 0\,, \quad \tau \geq 0; \\ u(S,\tau) &= S - e^{-r\tau}K\,, \quad \tau \geq 0. \end{aligned}$$

Spatial discretization: centered finite differences



Comparison at $t = 1$.

Heston equation

For $u = u(s, v, \tau)$,

$$\frac{\partial u}{\partial \tau} = \frac{1}{2} s^2 v \frac{\partial^2 u}{\partial s^2} + \rho \sigma s v \frac{\partial^2 u}{\partial s \partial v} + \frac{1}{2} \sigma^2 v \frac{\partial^2 u}{\partial v^2} + (r_d - r_f) s \frac{\partial u}{\partial s} + \kappa(\eta - v) \frac{\partial u}{\partial v} - r_d u\,,$$

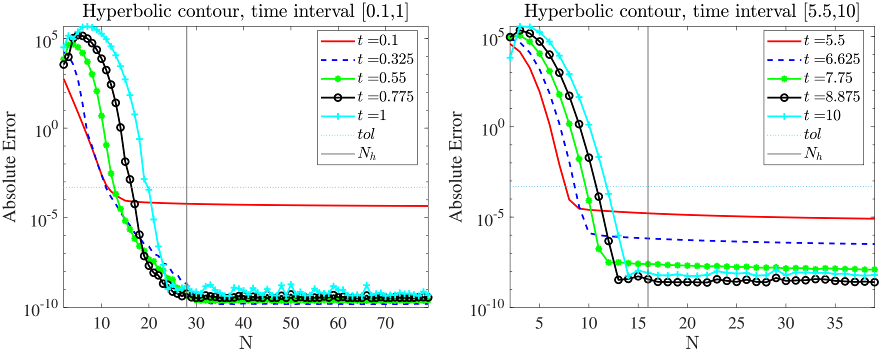
for $\tau \geq 0, \quad 0 \leq s \leq S, \quad 0 \leq v \leq V.$

With initial and boundary conditions

$$\begin{aligned} u(s, 0) &= \max(0, s - K) \\ u(0, v, \tau) &= 0\,, \quad \frac{\partial u}{\partial s}(S, v, \tau) = 1\,, \quad \tau \geq 0,\ 0 \leq v \leq V; \\ u(s, 0, \tau) &= 0\,, \quad u(s, V, \tau) = s\,, \qquad \tau \geq 0,\ 0 \leq s \leq S. \end{aligned}$$

Spatial discretization: ADI difference scheme from in 'Hout & Foulon 2010.

Results on time windows for Heston equation



Heston equation in time intervals $[0.1, 1]$ (left) and $[5.5, 10]$ (right), for $tol = 5 \cdot 10^{-4}$.

References

- N. Guglielmi, M. Lopéz-Fernández, M. Manucci, *Pseudospectral roaming contour integral methods for convection-diffusion equations*, Journal of Scientific Computing 89 (22), 2021.

- Manucci, M.: Accompanying codes published at GitHub (2020). https://github.com/MattiaManucci/Contour_Integral_Methods.git

- N. Guglielmi, M. Lopéz-Fernández, G. Nino, *Numerical inverse Laplace transform for convection-diffusion equations in finance*, Math. Comput., 2020.

Thanks for your attention!

Extension to time windows

$$F(z(x_j)) = \mathrm{e}^{z(x_j)t}\hat{u}(z(x_j))z'(x_j)$$

Note that:

- the main effort is due to the computation of $\hat{u}(z(x_j)) = (z(x_j)I - A(\mu))^{-1}\left(u_0 + \hat{b}(z(x_j),\mu)\right)$;
- the dependence on time is only in the scalar term $\mathrm{e}^{z(x_j)t}$.

Therefore: it is possible to construct a unique profile of integration for a time window

$$[t_0, \Lambda t_0],\ \Lambda > 1.$$

Once computed $\hat{u}(z(x))$ on the quadrature nodes the solution u can be quickly evaluated $\forall t \in [t_0, t_1]$.

Extension to time windows

$$F(z(x_j)) = \mathrm{e}^{z(x_j)t}\hat{u}(z(x_j))z'(x_j)$$

Note that:

- the main effort is due to the computation of $\hat{u}(z(x_j)) = (z(x_j)I - A(\mu))^{-1}\left(u_0 + \hat{b}(z(x_j), \mu)\right)$;
- the dependence on time is only in the scalar term $\mathrm{e}^{z(x_j)t}$.

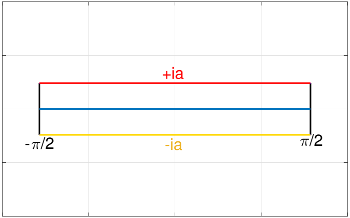
Therefore: it is possible to construct a unique profile of integration for a time window

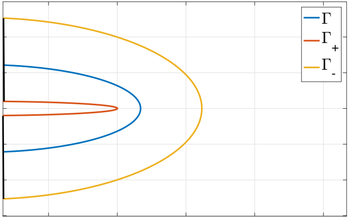
$$[t_0, \Lambda t_0],\ \Lambda > 1.$$

Once computed $\hat{u}(z(x))$ on the quadrature nodes the solution u can be quickly evaluated $\forall t \in [t_0, t_1]$.

Conformal mapping

$$\frac{1}{2\pi\mathrm{i}}\int_{-\pi/2}^{\pi/2} \mathrm{e}^{z(x)t}\,\left(z(x)\mathrm{I}-A\right)^{-1}\left(u_0+\hat{b}(z(x))\right) z'(x)\,dx$$



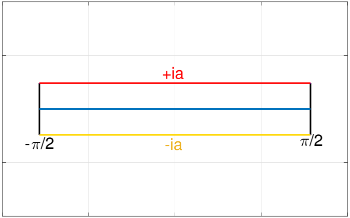


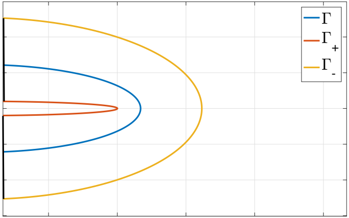
$$z(x+\mathrm{i}y) = A_1(y)\cos x + \mathrm{i}A_2(y)\sin x + A_3(y)$$

with suitable $A_1, A_2, A_3 \in \mathbb{R}$ provides a parametrization of the ellipses.

Conformal mapping

$$\frac{1}{2\pi \mathrm{i}} \int_{-\pi/2}^{\pi/2} \mathrm{e}^{z(x)t} \left(z(x)\mathrm{I} - A\right)^{-1} \left(u_0 + \hat{b}(z(x))\right) z'(x)\, dx$$



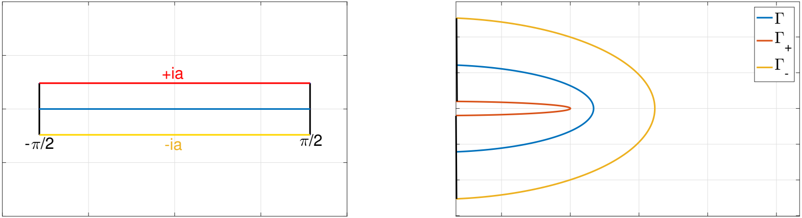


$$z(x + \mathrm{i}y) = A_1(y)\cos x + \mathrm{i}A_2(y)\sin x + A_3(y)$$

with suitable $A_1, A_2, A_3 \in \mathbb{R}$ provides a parametrization of the ellipses.

Conformal mapping

$$\frac{1}{2\pi \mathrm{i}}\int_{-\pi/2}^{\pi/2} \mathrm{e}^{z(x)t}\,(z(x)\mathrm{I}-A)^{-1}\left(u_0+\hat{b}(z(x))\right) z'(x)\,dx$$

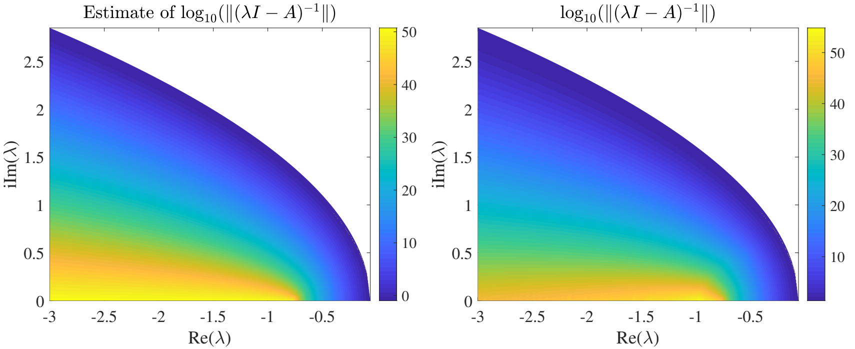


$$z(x+\mathrm{i}y) = A_1(y)\cos x + \mathrm{i}A_2(y)\sin x + A_3(y)$$

with suitable $A_1, A_2, A_3 \in \mathbb{R}$ provides a parametrization of the ellipses.

Estimates of the resolvent of the BS operator

We generalize the analysis in Reddy & Trefethen, 1994 for the canonical convection-diffusion equation and derive theoretical estimates for the resolvent of the BS operator.



Theoretical estimate of the resolvent norm (left) and computed resolvent norm (right). This also provides us a good guess for Γ_{left}.

A variational result for simple singular values

Lemma (Kato, 1995)

Let $D(t)$ be a differentiable matrix-valued function in a neighborhood of t_0. Let

$$D(t) = U(t)\Sigma(t)V(t)^* = \sum_i u_i(t)\sigma_i(t)v_i(t)^*$$

be a smooth (with respect to t) singular value decomposition of the matrix $D(t)$ and $\sigma(t)$ be a certain singular value of $D(t)$ converging to a simple singular value $\hat{\sigma}$ of $D_0 = D(t_0)$.
If $\hat{u}, \hat{v}$ are the associated left and right singular vectors, respectively, the function $\sigma(t)$ is differentiable near $t = t_0$ with

$$\dot{\sigma}(t_0) = \Re\left(\hat{u}^* \dot{D}_0 \hat{v}\right) \qquad \text{with } \dot{D}_0 = \dot{D}(t_0).$$

Guglielmi, López-Fernández & MM: construction of Γ_{left}

We fix:

- z^R intersection of Γ_{left} with the real axis.
- z^L with $e^{z^L t} <$ eps, being eps the machine precision of the solver used.
- Choose a grid of M points z_k, with $k = 1, \dots, M$, $\operatorname{Im} z_k > 0$,

$$z^R > \operatorname{Re} z_1 > \operatorname{Re} z_2 > \cdots > \operatorname{Re} z_M.$$

- A **control point** $d + \mathrm{i} r$ on Γ_{left} with

$$d = \frac{1}{M} \sum_{k=1}^{M} \operatorname{Re} z_k \qquad \text{fixed.}$$

If any of the z_k lays in the wrong pseudospectral level set, we move the ordinate r of the control point by solving

$$\tilde{\sigma}^k(d, r) - \epsilon = 0, \quad \text{with respect to } r,$$

for $\tilde{\sigma}^k(d, r)$ the smallest weighted singular value of $A - z_k(d, r)I$.

Guglielmi, López-Fernández & MM: construction of Γ_{left}

We fix:

- z^R intersection of Γ_{left} with the real axis.
- z^L with $\mathrm{e}^{z^L t} <$ eps, being eps the machine precision of the solver used.
- Choose a grid of M points z_k, with $k = 1, \ldots, M$, $\operatorname{Im} z_k > 0$,

$$z^R > \operatorname{Re} z_1 > \operatorname{Re} z_2 > \cdots > \operatorname{Re} z_M.$$

- A **control point** $d + \mathrm{i} r$ on Γ_{left} with

$$d = \frac{1}{M} \sum_{k=1}^{M} \operatorname{Re} z_k \qquad \text{fixed.}$$

If any of the z_k lays in the wrong pseudospectral level set, we move the ordinate r of the control point by solving

$$\tilde{\sigma}^k(d, r) - \epsilon = 0, \quad \text{with respect to } r,$$

for $\tilde{\sigma}^k(d, r)$ the smallest weighted singular value of $A - z_k(d, r)I$.

Guglielmi, López-Fernández & MM: construction of Γ_{left}

We fix:

- z^R intersection of Γ_{left} with the real axis.
- z^L with $\mathrm{e}^{z^L t} < \text{eps}$, being eps the machine precision of the solver used.
- Choose a grid of M points z_k, with $k = 1, \dots, M$, $\operatorname{Im} z_k > 0$,

$$z^R > \operatorname{Re} z_1 > \operatorname{Re} z_2 > \cdots > \operatorname{Re} z_M.$$

- A **control point** $d + \mathrm{i} r$ on Γ_{left} with

$$d = \frac{1}{M} \sum_{k=1}^{M} \operatorname{Re} z_k \qquad \text{fixed.}$$

If any of the z_k lays in the wrong pseudospectral level set, we move the ordinate r of the control point by solving

$$\tilde{\sigma}^k(d, r) - \epsilon = 0, \quad \text{with respect to } r,$$

for $\tilde{\sigma}^k(d, r)$ the smallest weighted singular value of $A - z_k(d, r)I$.

Guglielmi, López-Fernández & MM: construction of Γ_{left}

We fix:

- z^R intersection of Γ_{left} with the real axis.
- z^L with $\mathrm{e}^{z^L t} <$ eps, being eps the machine precision of the solver used.
- Choose a grid of M points z_k, with $k = 1, \ldots, M$, $\operatorname{Im} z_k > 0$,

$$z^R > \operatorname{Re} z_1 > \operatorname{Re} z_2 > \cdots > \operatorname{Re} z_M.$$

- A **control point** $d + \mathrm{i} r$ on Γ_{left} with

$$d = \frac{1}{M} \sum_{k=1}^{M} \operatorname{Re} z_k \qquad \text{fixed.}$$

If any of the z_k lays in the wrong pseudospectral level set, we move the ordinate r of the control point by solving

$$\tilde{\sigma}^k(d,r) - \epsilon = 0, \quad \text{with respect to } r,$$

for $\tilde{\sigma}^k(d,r)$ the smallest weighted singular value of $A - z_k(d,r)I$.

Guglielmi, López-Fernández & MM: construction of Γ_{left}

We fix:

- z^R intersection of Γ_{left} with the real axis.
- z^L with $\mathrm{e}^{z^L t} <$ eps, being eps the machine precision of the solver used.
- Choose a grid of M points z_k, with $k = 1, \ldots, M$, $\operatorname{Im} z_k > 0$,

$$z^R > \operatorname{Re} z_1 > \operatorname{Re} z_2 > \cdots > \operatorname{Re} z_M.$$

- A **control point** $d + \mathrm{i} r$ on Γ_{left} with

$$d = \frac{1}{M} \sum_{k=1}^{M} \operatorname{Re} z_k \qquad \text{fixed.}$$

If any of the z_k lays in the wrong pseudospectral level set, we move the ordinate r of the control point by solving

$$\tilde{\sigma}^k(d, r) - \epsilon = 0, \quad \text{with respect to } r,$$

for $\tilde{\sigma}^k(d, r)$ the smallest weighted singular value of $A - z_k(d, r)I$.

An example: the parabolic profile

$$\Gamma_{left}(x) = -x^2 + z^R + \frac{\mathrm{i} r x}{\sqrt{z^R - d}}, \qquad x \in \mathbb{R}. \tag{4}$$

Setting $\Gamma_{left}(x) = \phi + \mathrm{i}\psi$ and fixing the abscissa $\phi = \mathrm{Re}\left(\Gamma_{left}\right)$ we obtain

$$\psi = \frac{r x}{\sqrt{z^R - d}},$$

which depends on r and d. We easily obtain

$$\begin{aligned} \frac{\partial \psi}{\partial d} &= \frac{x r}{2(z^R - d)^{3/2}} \\ \frac{\partial \psi}{\partial r} &= \frac{x}{\sqrt{z^R - d}}. \end{aligned}$$

Applying Lemma 1 to $\tilde{\sigma}(d, r)$- with u and v left and right associated singular vectors - we get

$$\frac{d}{dr}\tilde{\sigma}\left(d, r\right) = -\mathrm{e}^{-\operatorname{Re}(z_k)t}\operatorname{Re}(\mathrm{i}u^* v)\, g$$

with

$$g = \frac{x_k}{\sqrt{z^R - d}}.$$

In order to accurately compute r such that $\tilde{\sigma}(d, r) = \epsilon$ do a few (say m) Newton iterations

$$r^{\ell+1} = r^{\ell} + \frac{\mathrm{e}^{-\operatorname{Re}(z_k)t}\sigma_{\min}\left(A - z(d, r^{\ell})\mathrm{I}\right) - \epsilon}{\mathrm{e}^{-\operatorname{Re}(z_k)t}\operatorname{Re}\left(\mathrm{i}(u^{\ell})^* v^{\ell}\right) g}, \quad \ell = 1, \ldots, m-1 \qquad (5)$$

with u^{ℓ} and v^{ℓ} singular vectors associated to $\sigma_{\min}(A - z(d, r^{\ell})\mathrm{I})$ and r^{ℓ} the actual ordinate of the control point.
Then we compute a new parabola, which interpolates $d + \mathrm{i}r^{m}$, reparametrize it and compute a new set of points.

Applying Lemma 1 to $\tilde{\sigma}(d,r)$- with u and v left and right associated singular vectors - we get

$$\frac{d}{dr}\tilde{\sigma}\left(d,r\right) = -\mathrm{e}^{-\,\mathrm{Re}(z_k)t}\,\mathrm{Re}(\mathrm{i}u^* v)\,g$$

with

$$g = \frac{x_k}{\sqrt{z^R - d}}.$$

In order to accurately compute r such that $\tilde{\sigma}(d,r) = \epsilon$ do a few (say m) Newton iterations

$$r^{\ell+1} = r^{\ell} + \frac{\mathrm{e}^{-\,\mathrm{Re}(z_k)t}\sigma_{\min}\left(A - z(d,r^{\ell})\mathrm{I}\right) - \epsilon}{\mathrm{e}^{-\,\mathrm{Re}(z_k)t}\,\mathrm{Re}\Big(\mathrm{i}(u^{\ell})^* v^{\ell}\Big)\,g}, \quad \ell = 1,\ldots,m-1 \qquad (5)$$

with u^{ℓ} and v^{ℓ} singular vectors associated to $\sigma_{\min}(A - z(d,r^{\ell})\mathrm{I})$ and r^{ℓ} the actual ordinate of the control point.

Then we compute a new parabola, which interpolates $d + \mathrm{i}r^m$, reparametrize it and compute a new set of points.

Parameters selection: computation of a and c

1. We first find a maximal value for c, from $\mathrm{Re}(z(c_{\max}\pi)) = z^L$, where $\mathrm{e}^{tz^L} = eps$. It is $c_{\max}(a)$.
2. **Compute a:** For a given target accuracy tol we have

$$N \leq \frac{c_{\max}(a)}{a}\Big(\log\Big(2\pi c_{\max}(a)\tilde{M}_{right} + \pi\tilde{M}_{left}\Big) - \log(tol)\Big),$$

   We minimize numerically the right hand side. The interval of minimization for a is chosen in such a way that stability of the method is ensured.

3. **Compute c:** From

$$|F(c\pi)| = tol.$$

   Determine c by fixed point iterations.

4. Set $N = \left\lceil \frac{c}{a}\Big(\log\Big(2\pi c\tilde{M}_{right} + \pi\tilde{M}_{left}\Big) - \log(tol)\Big)\right\rceil$.

Parameters selection: computation of a and c

1. We first find a maximal value for c, from $\mathrm{Re}(z(c_{\max}\pi)) = z^L$, where $\mathrm{e}^{tz^L} = eps$. It is $c_{\max}(a)$.
2. **Compute a:** For a given target accuracy tol we have

$$N \le \frac{c_{\max}(a)}{a}\left(\log\left(2\pi c_{\max}(a)\tilde{M}_{right} + \pi\tilde{M}_{left}\right) - \log\left(tol\right)\right),$$

   We minimize numerically the right hand side. The interval of minimization for a is chosen in such a way that stability of the method is ensured.
3. **Compute c:** From

$$|F(c\pi)| = tol.$$

   Determine c by fixed point iterations.
4. Set $N = \left\lceil \frac{c}{a}\left(\log\left(2\pi c\tilde{M}_{right} + \pi\tilde{M}_{left}\right) - \log\left(tol\right)\right)\right\rceil$.

Parameters selection: computation of a and c

1. We first find a maximal value for c, from $\mathrm{Re}(z(c_{\max}\pi)) = z^L$, where $\mathrm{e}^{tz^L} = eps$. It is $c_{\max}(a)$.
2. **Compute a:** For a given target accuracy tol we have

$$N \leq \frac{c_{\max}(a)}{a}\Big(\log\Big(2\pi c_{\max}(a)\tilde{M}_{right} + \pi\tilde{M}_{left}\Big) - \log\left(tol\right)\Big),$$

   We minimize numerically the right hand side. The interval of minimization for a is chosen in such a way that stability of the method is ensured.

3. **Compute c:** From

$$|F(c\pi)| = tol.$$

   Determine c by fixed point iterations.

4. Set $N = \left\lceil \frac{c}{a}\Big(\log\Big(2\pi c\tilde{M}_{right} + \pi\tilde{M}_{left}\Big) - \log\left(tol\right)\Big)\right\rceil$.

Parameters selection: computation of a and c

1. We first find a maximal value for c, from $\mathrm{Re}(z(c_{\max}\pi)) = z^L$, where $\mathrm{e}^{tz^L} = eps$. It is $c_{\max}(a)$.
2. **Compute a:** For a given target accuracy tol we have

$$N \le \frac{c_{\max}(a)}{a}\left(\log\left(2\pi c_{\max}(a)\tilde{M}_{right} + \pi\tilde{M}_{left}\right) - \log\left(tol\right)\right),$$

   We minimize numerically the right hand side. The interval of minimization for a is chosen in such a way that stability of the method is ensured.

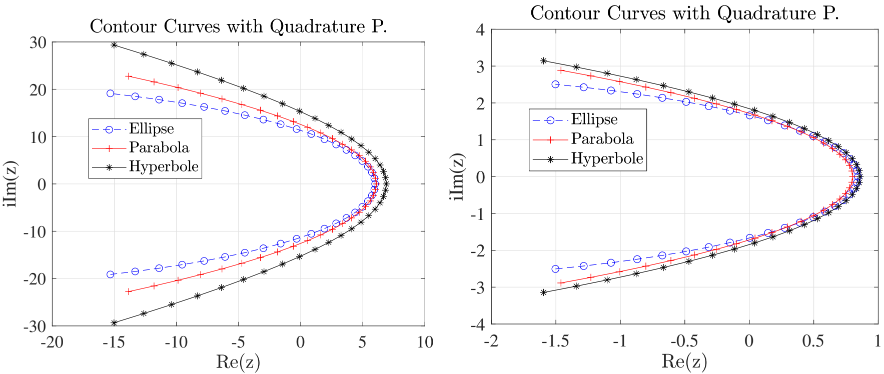
3. **Compute c:** From

$$|F(c\pi)| = tol.$$

   Determine c by fixed point iterations.

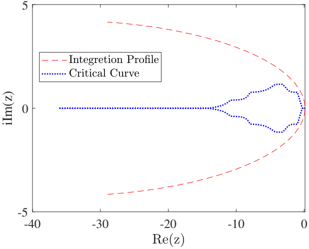
4. Set $N = \left\lceil \frac{c}{a}\left(\log\left(2\pi c\tilde{M}_{right} + \pi\tilde{M}_{left}\right) - \log\left(tol\right)\right)\right\rceil$.

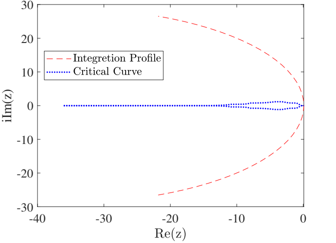
Contours and nodes for BS

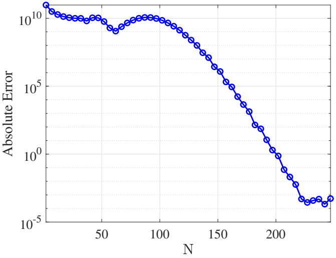


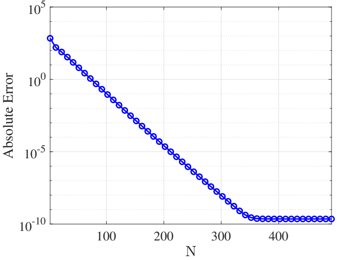
Example of integration profiles for the Black-Scholes problem for tolerance $tol = 5 \cdot 10^{-6}$ at time $t = 1$ (left) and $t = 10$ (right).

Examples of constructed integration contours

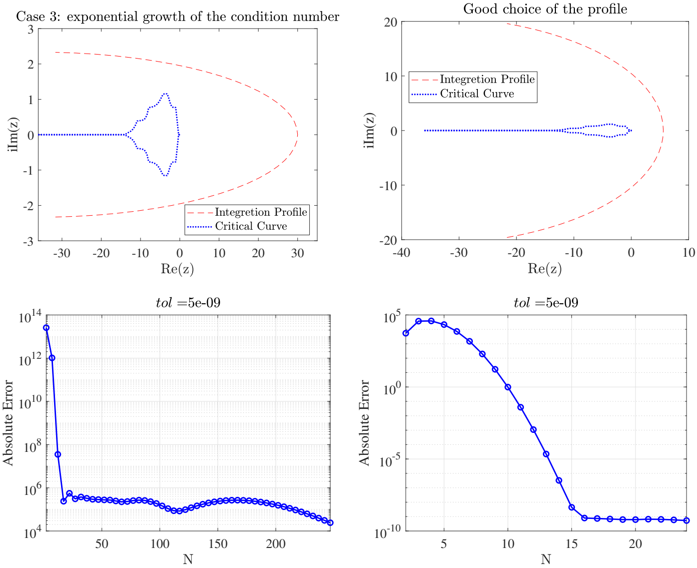

Case 1: profile close to the pseudospectrum level curve


Case 2: profile far from the pseudospectrum level curve


tol =5e-09


tol =5e-09

Examples of constructed integration contours


Case 3: exponential growth of the condition number
iIm(z)
Re(z)
Integretion Profile
Critical Curve
Good choice of the profile
iIm(z)
Re(z)
Integretion Profile
Critical Curve
tol =5e-09
Absolute Error
N
10^{14}
10^{12}
10^{10}
10^{8}
10^{6}
10^{4}
tol =5e-09
Absolute Error
N
10^{5}
10^{0}
10^{-5}
10^{-10}