A Review of Transparent and Artificial Boundary Conditions Techniques for Linear and Nonlinear Schrödinger Equations

Matthias Ehrhardt

Institut für Mathematik, Technische Universität Berlin

ehrhardt@math.tu-berlin.de

joint work with X. Antoine, A. Arnold, C. Besse and A. Schädle

Preprint No.18–2007 of the Institute for Mathematics, TU Berlin, May 2007

ICIAM

Minisymposium IC/MP/025/R/32

Artificial Boundary Conditions for linear and nonlinear Schrödinger equations

ICIAM07 – *6th International Congress on Industrial and Applied Mathematics*

16 – 20 July 2007, Zürich, Switzerland

Outline

- Numerical Computation on unbounded Domains
- Transparent BCs for Schrödinger Equation in 1D
  - Analytic Transparent BCs
  - temporally/spatially/fully discrete Transparent BCs
- Discretizations and Approximations of the Transparent BCs
  - discretization by quadrature rules
  - approximation of convolution kernel by sums of exponentials
  - rational approximations of the Fourier–symbol of the kernel
- Numerical Example
- Outlook: Cubic nonlinear case

Numerical Computation on unbounded Domains

- Many physical problems are described mathematically by a partial differential equation defined on an unbounded domain
- **Numerical computation**: one has to restrict the computational domain Ω by artificial boundary conditions or absorbing layers
- If approximate solution coincides on Ω with exact solution, these BCs are called transparent boundary conditions (TBCs)

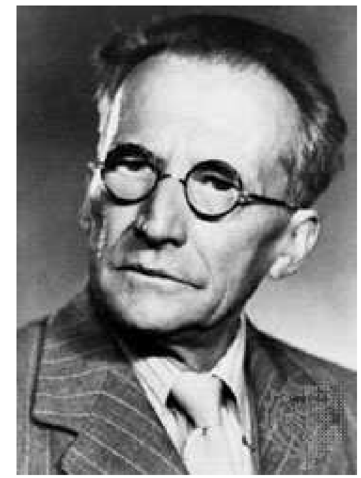
Numerical Computation on unbounded Domains

- Many physical problems are described mathematically by a partial differential equation defined on an unbounded domain
- **Numerical computation**: one has to restrict the computational domain Ω by artificial boundary conditions or absorbing layers
- If approximate solution coincides on Ω with exact solution, these BCs are called transparent boundary conditions (TBCs)
- Constructed artificial boundary conditions should
  - approximate the exact whole–space solution restricted to Ω
  - lead to well–posed (initial) boundary value problem
  - allow for an efficient (and easy) numerical implementation

The Schrödinger Equation

- SE is the fundamental equation of quantum mechanics developed 1926 by Austrian physicist Erwin Schrödinger
- describes form of probability waves that govern motion of small particles

  specifies how these waves are altered by external influences



(1887–1961)

- it is called Fresnel's equation in optics and 'parabolic equation' in acoustics & geophysics

$$i\partial_t u = -\partial_x^2 u + V(x,t)u, \quad x \in \mathbb{R},\ t > 0$$

$u(x,t) \in \mathbb{C}$ wave function, $V(x,t) \in \mathbb{R}$ given potential

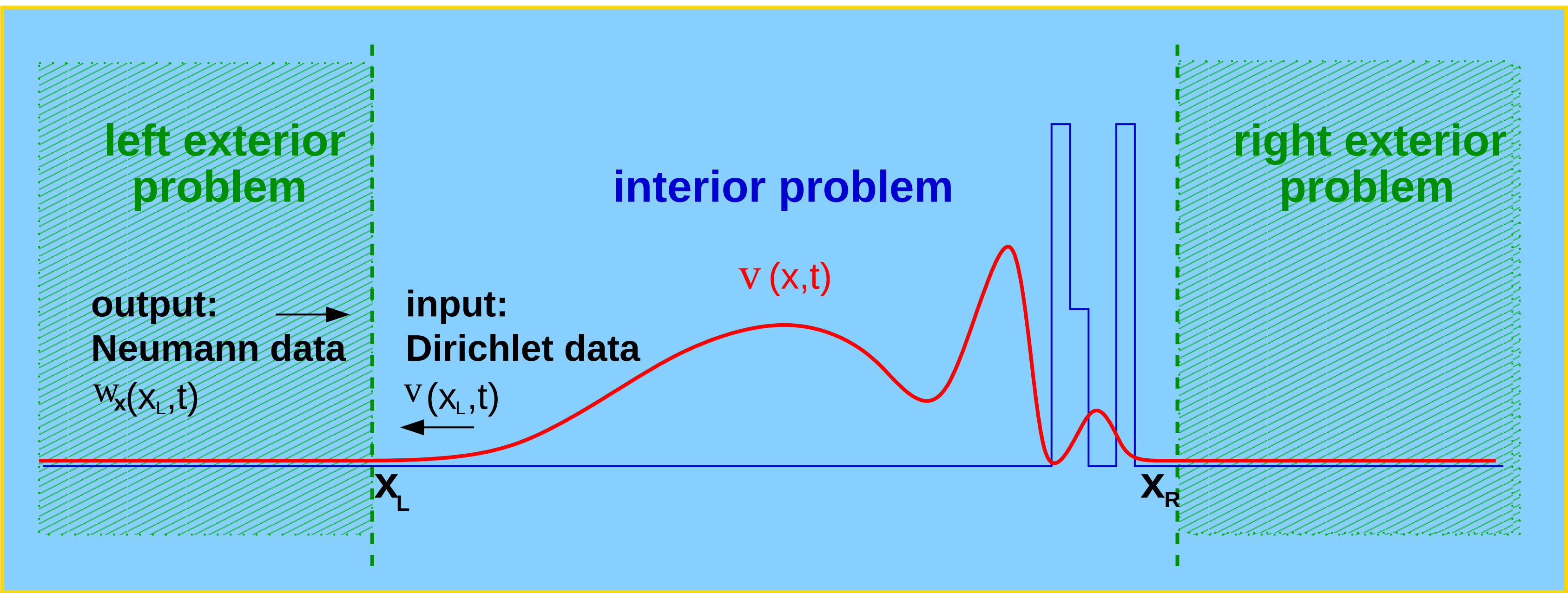
TBCs for the Schrödinger Equation

$$i\partial_t u = -\partial_x^2 u + V(x,t)u, \quad x \in \mathbb{R}, \quad t > 0$$

A1: $\text{supp } u(x,0) \subseteq (x_l, x_r) \quad \ldots$ computational domain

A2: $V(x,t) = V_l, \quad x \leq x_l, \qquad V(x,t) = V_r, \quad x \geq x_r$

(both assumptions can be relaxed significantly)



- **Goal:** reproduce $v = u_{[x_l,x_r]}$ with transparent BCs at $x = x_l, x_r$

The Derivation of the right TBC

- right TBC $\partial_x u(x_r, t) = (T\, u)(x_r, t)$ from exterior problem:

$$i\partial_t w = -\partial_x^2 w + V_r w, \quad x > x_r$$

$$w(x, 0) = 0, \quad x > x_r$$

$$w(x_r, t) = v(x_r, t), \quad \text{decay condition } \lim_{x\to\infty} w(x, t) = 0$$

The Derivation of the right TBC

- right TBC $\partial_x u(x_r,t) = (T\,u)(x_r,t)$ from exterior problem:

$$i\partial_t w = -\partial_x^2 w + V_r w, \quad x > x_r$$

$$w(x,0) = 0, \quad x > x_r$$

$$w(x_r,t) = v(x_r,t), \quad \text{decay condition } \lim_{x\to\infty} w(x,t) = 0$$

- explicit solution of the right exterior problem by

Laplace transformation: $\hat{w}(x,s) = \int_0^\infty w(x,t)e^{-st}\,dt$

$$\partial_x^2 \hat{w} = (-is + V_r)\hat{w}, \quad x > x_r$$

solution: $\hat{w}(x,s) = A^+(s)e^{\,{}^+\!\sqrt{-is+V_r}\,x} + A^-(s)e^{-\,{}^+\!\sqrt{-is+V_r}\,x}$

The Derivation of the right TBC

- right TBC $\partial_x u(x_r,t) = (T\,u)(x_r,t)$ from exterior problem:

$$i\partial_t w = -\partial_x^2 w + V_r w, \quad x > x_r$$

$$w(x,0) = 0, \quad x > x_r$$

$$w(x_r,t) = v(x_r,t), \quad \text{decay condition } \lim_{x\to\infty} w(x,t) = 0$$

- explicit solution of the right exterior problem by
Laplace transformation: $\hat{w}(x,s) = \int_0^\infty w(x,t)e^{-st}\,dt$

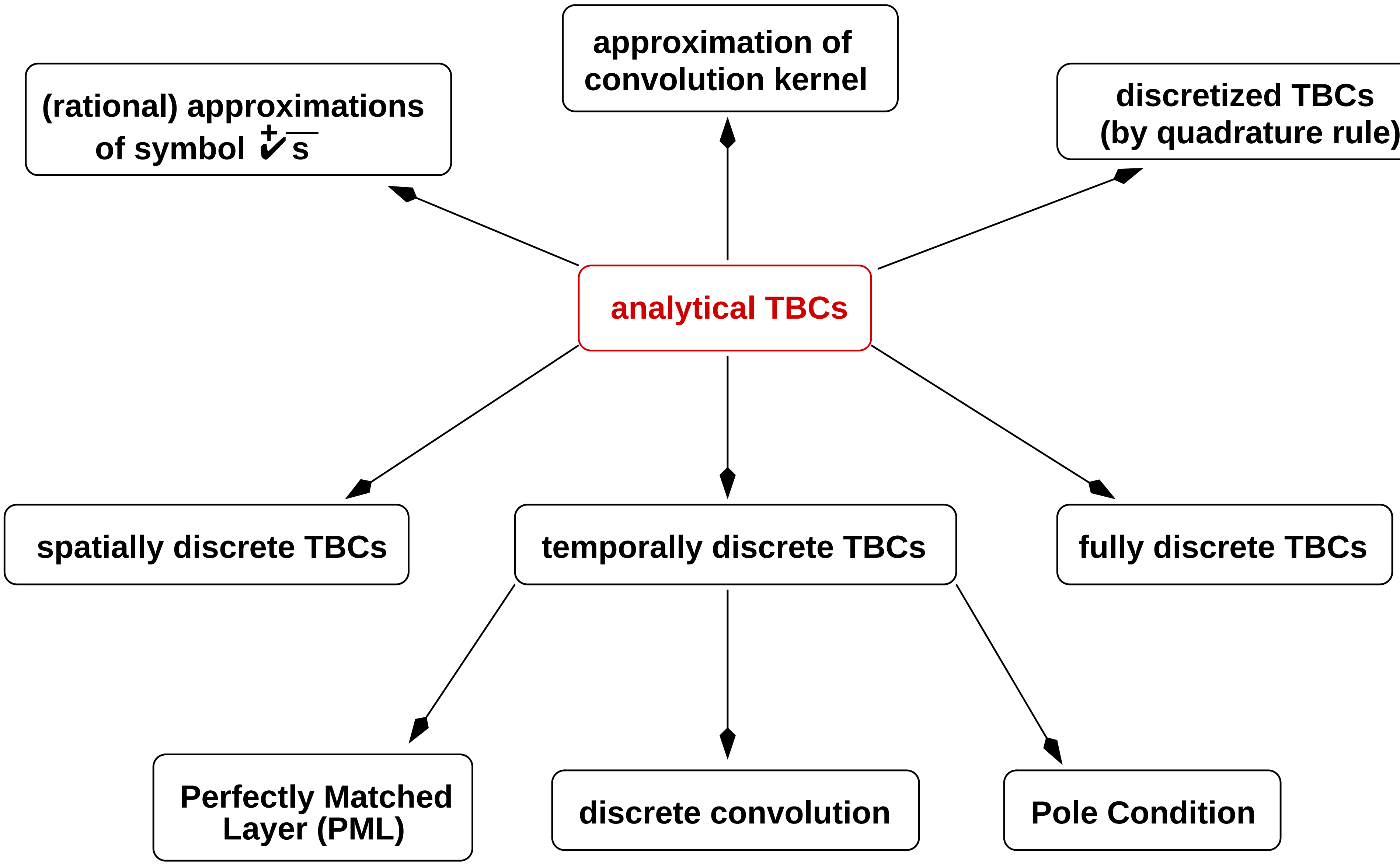
$$\partial_x^2 \hat{w} = (-is + V_r)\hat{w}, \quad x > x_r$$

solution must be in $L^2(\Omega)$: $\partial_x \hat{w}(x_r,s) = \hat{v}(x_r,s)\,e^{-\sqrt[+]{-is+V_r}\,(x-x_r)}$

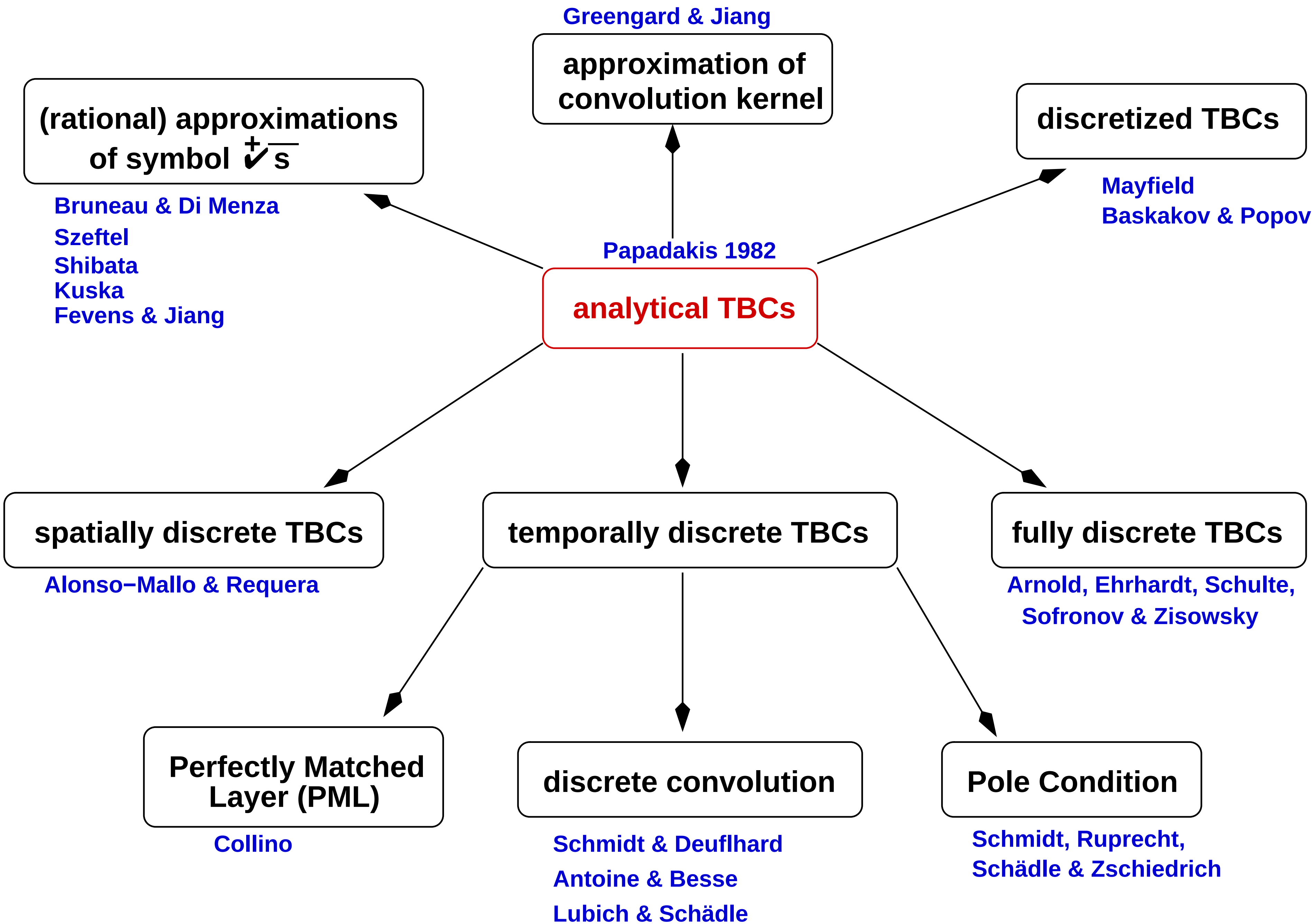
- inverse Laplace transformation $\rightsquigarrow$ right TBC: [Papadakis '82]

$$\boxed{\partial_x u(x_r,t) = -e^{-i\frac{\pi}{4}}\frac{1}{\sqrt{\pi}}\,e^{-iV_r t}\,\frac{d}{dt}\int_0^t \frac{u(x_r,\tau)e^{iV_r\tau}}{\sqrt{t-\tau}}\,d\tau}$$

Approaches for transient Schrödinger Equation



Approaches for transient Schrödinger Equation



Procedure to derive the analytic TBC

(1) Split problem into coupled equations: interior and exterior problems

(2) Apply a Laplace transformation in time t

(3) Solve the ordinary differential equations in x

(4) Allow only 'outgoing' waves by taking decaying solution as $|x| \to \infty$

(5) Match Dirichlet and Neumann values at $x = x_l$, $x = x_r$

(6) Apply the inverse Laplace transformation to obtain the TBC

Procedure to derive the analytic TBC

(1) Split problem into coupled equations: interior and exterior problems

(2) Apply a Laplace transformation in time t

(3) Solve the ordinary differential equations in x

(4) Allow only 'outgoing' waves by taking decaying solution as $|x| \to \infty$

(5) Match Dirichlet and Neumann values at $x = x_l$, $x = x_r$

(6) Apply the inverse Laplace transformation to obtain the TBC

- For the IBVP on Ω with a DtN or a NtD TBC, existence and uniqueness of the solution has been proved, e.g. [Antoine/Besse'03]
- Continuous TBCs fully solve the problem of confining the spatial domain to a bounded interval

Procedure to derive the analytic TBC

(1) Split problem into coupled equations: interior and exterior problems

(2) Apply a Laplace transformation in time t

(3) Solve the ordinary differential equations in x

(4) Allow only ‘outgoing’ waves by taking decaying solution as $|x| \to \infty$

(5) Match Dirichlet and Neumann values at $x = x_l$, $x = x_r$

(6) Apply the inverse Laplace transformation to obtain the TBC

- For the IBVP on Ω with a DtN or a NtD TBC, existence and uniqueness of the solution has been proved, e.g. [Antoine/Besse’03]
- Continuous TBCs fully solve the problem of confining the spatial domain to a bounded interval

but: their numerical discretization is not trivial at all!

here: two different approaches to derive (semi)discrete TBCs and ABCs

Temporally discrete TBCs

- SE discretized uniformly in time with A–stable multi–step method

$$\frac{i}{\Delta t}\sum_{j=0}^{K}\alpha_j u^{n-j} = \sum_{j=0}^{K}\beta_j\left(-\partial_x^2 + V\right)u^{n-j}, \quad n \geq K$$

(2) instead of Laplace-transformation we apply a $\mathcal{Z}$-transformation

$$\mathcal{Z}(u^n) = \hat{u}(z) := \sum_{n=0}^{\infty} u^n\, z^{-n}, \quad z \in \mathbb{C}, \quad |z| > R\big(\mathcal{Z}(u^n)\big)$$

(3) 2nd order ODE $$i\,\underbrace{\frac{\delta(z)}{\Delta t}}_{\leftrightarrow s}\,\hat{w}(z) = \left(-\partial_x^2 + V_r\right)\hat{w}(z), \quad x > x_r$$

$$\delta(z) = \frac{\sum_{j=0}^{K}\alpha_j z^{K-j}}{\sum_{j=0}^{K}\beta_j z^{K-j}}$$ generating function of time integration scheme

- Assumption on startup: $\mathrm{supp}(u^j) \subset [x_l, x_r], \quad j = 0, 1, \ldots, K-1$

Example: trapezoidal rule discretization

$$i\frac{u^{n+1}-u^n}{\Delta t} = -\partial_x^2 \frac{u^{n+1}+u^n}{2} + \frac{V^{n+1}(x)u^{n+1}+V^n(x)u^n}{2},\ x \in \mathbb{R}$$

method (Crank-Nicolson) has generating function $\delta(z) = 2\frac{z-1}{z+1}$

Example: trapezoidal rule discretization

$$i\frac{u^{n+1}-u^n}{\Delta t} = -\partial_x^2 \frac{u^{n+1}+u^n}{2} + \frac{V^{n+1}(x)u^{n+1}+V^n(x)u^n}{2},\ x \in \mathbb{R}$$

method (Crank-Nicolson) has generating function $\delta(z) = 2\frac{z-1}{z+1}$

(3) solving the ODE yields general solution

$$\hat{w}(x,z) = A^+(z)\, e^{i \sqrt[+]{i\frac{\delta(z)}{\Delta t}-V_r}\, x} + A^-(z)\, e^{-i \sqrt[+]{i\frac{\delta(z)}{\Delta t}-V_r}\, x}, \quad x > x_r$$

Example: trapezoidal rule discretization

$$i\frac{u^{n+1}-u^n}{\Delta t} = -\partial_x^2 \frac{u^{n+1}+u^n}{2} + \frac{V^{n+1}(x)u^{n+1}+V^n(x)u^n}{2},\ x \in \mathbb{R}$$

method (Crank-Nicolson) has generating function $\delta(z) = 2\frac{z-1}{z+1}$

(3) solving the ODE yields general solution

$$\hat{w}(x,z) = A^{+}(z)\, e^{i \sqrt[+]{i\frac{\delta(z)}{\Delta t} - V_r}\, x} + A^{-}(z)\, e^{-i \sqrt[+]{i\frac{\delta(z)}{\Delta t} - V_r}\, x}, \quad x > x_r$$

(4) $u^n \in L^2(]x_r, \infty[) \quad \rightsquigarrow \quad A^-$ must vanish

Example: trapezoidal rule discretization

$$i\frac{u^{n+1}-u^n}{\Delta t} = -\partial_x^2 \frac{u^{n+1}+u^n}{2} + \frac{V^{n+1}(x)u^{n+1}+V^n(x)u^n}{2}, \; x \in \mathbb{R}$$

method (Crank-Nicolson) has generating function $\delta(z) = 2\frac{z-1}{z+1}$

(3) solving the ODE yields general solution

$$\hat{w}(x,z) = A^+(z)\, e^{i \sqrt[+]{i\frac{\delta(z)}{\Delta t} - V_r}\, x} + A^-(z)\, e^{-i \sqrt[+]{i\frac{\delta(z)}{\Delta t} - V_r}\, x}, \quad x > x_r$$

(4) $u^n \in L^2(]x_r, \infty[) \quad \rightsquigarrow \quad A^-$ must vanish

$$\mathcal{Z}(\partial_x w^n)(z) = i \sqrt[+]{i\frac{\delta(z)}{\Delta t} - V_r}\, \mathcal{Z}(w^n)(z), \quad x = x_r$$

(6) inverse $\mathcal{Z}$-transformation $\rightsquigarrow$ expression $\partial_x w^n(x_r)$ in terms of $w^k(x_r)$

Example: trapezoidal rule discretization

$$i\frac{u^{n+1}-u^n}{\Delta t} = -\partial_x^2 \frac{u^{n+1}+u^n}{2} + \frac{V^{n+1}(x)u^{n+1}+V^n(x)u^n}{2}, \; x \in \mathbb{R}$$

method (Crank-Nicolson) has generating function $\delta(z) = 2\frac{z-1}{z+1}$

(3) solving the ODE yields general solution

$$\hat{w}(x,z) = A^+(z)\, e^{i \sqrt[+]{i\frac{\delta(z)}{\Delta t} - V_r}\, x} + A^-(z)\, e^{-i \sqrt[+]{i\frac{\delta(z)}{\Delta t} - V_r}\, x}, \quad x > x_r$$

(4) $u^n \in L^2(]x_r, \infty[) \quad \rightsquigarrow \quad A^-$ must vanish

(6) $\partial_x v^{n+1} = \sum_{k=0}^{n+1} u_k v^{n+1-k}$ at $x = x_r$ with weights u_n (for $V_r = 0$)

$$u_k = -e^{-\frac{i\pi}{4}} \frac{\sqrt{2}}{\sqrt{\Delta t}} (-1)^k \tilde{u}_k, \quad k \in \mathbb{N}_0,$$

$$(\tilde{u}_0\, \tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \tilde{u}_4, \tilde{u}_5, \dots) = \left(1, 1, \frac{1}{2}, \frac{1}{2}, \frac{1\cdot 3}{2\cdot 4}, \frac{1\cdot 3}{2\cdot 4}, \dots\right)$$

Fully discrete TBCs

- SE discretized in space and time by Crank-Nicolson scheme

$$i\frac{u_j^{n+1}-u_j^n}{\Delta t} = -D_x^2\frac{u_j^{n+1}+u_j^n}{2} + \frac{V_j^{n+1}u_j^{n+1}+V_j^n u_j^n}{2}, \; j\in\mathbb{Z}$$

right artificial boundary is located at $x_r = x_l + J\Delta x$

(2) instead of Laplace-transformation we apply a $\mathcal{Z}$-transformation

(3) 2nd order difference equation with constant coefficients

$$\hat{w}_{j+1}(z) - 2\left(1-\frac{\Delta x^2}{2}\left(i\frac{\delta(z)}{\Delta t}+V_r\right)\right)\hat{w}_j(z) + \hat{w}_{j-1}(z) = 0, \; j > J$$

- general solution
  $\hat{w}_j(z) = A^+(z)\chi^{j-J}(z) + A^-(z)\chi^{-(j-J)}(z), \quad j \geq J-1$
  where $\chi(z)$ and $\chi(z)^{-1}$ are the roots of the quadratic equation

$$X^2 - 2\left(1-\frac{\Delta x^2}{2}\left(i\frac{\delta(z)}{\Delta t}+V_r\right)\right)X + 1 = 0$$

(4) decaying solutions $\hat{u}_j(z)$, $j \to \infty$ $\leadsto$ choose branch with $|\chi(z)| < 1$

- $\mathcal{Z}$-transformed right discrete TBC [Arnold '95, Ehrhardt '01]

$$\boxed{\hat{u}_{J-1}(z) = \chi(z)\hat{u}_J(z)}$$

- transformed boundary kernel $\chi(z)$

$$\chi(z) = 1 - \frac{\Delta x^2}{2}\left(i\frac{\delta(z)}{\Delta t} - V_r\right) - \sqrt[+]{\frac{\Delta x^2}{2}\left(i\frac{\delta(z)}{\Delta t} - V_r\right)\left(\frac{\Delta x^2}{2}\left(i\frac{\delta(z)}{\Delta t} - V_r\right) - 2\right)}$$

(6) $\mathcal{Z}^{-1}$-transformation yields convolution coefficients for discrete TBC

$$(\chi_n) := \mathcal{Z}^{-1}\big(\chi(z)\big), \quad n \in \mathbb{N}_0$$

- right discrete TBC reads in physical space

$$\boxed{u_{J-1}^n = \sum_{k=1}^{n} \chi_{n-k} u_J^k, \quad n \in \mathbb{N}}$$

Discretizations and Approximations

- TBCs completely solve problem of confining unbounded domain (theoretically)
- for an (efficient) implementation the TBCs have to be discretized and/or approximated
- three main approaches in the literature
  - discretizations of the TBC by quadrature rules
  - approximation of convolution kernel by sums of exponentials
  - rational approximations of Fourier–symbol of convolution kernel

Discretizations by quadrature formulas

- first idea to incorporate TBC in scheme is an ad–hoc discretization

$$\int_0^{t_{n+1}} \frac{u_x(x_r, t_{n+1}-\tau)\, e^{-iV_r\tau}}{\sqrt{\tau}}\, d\tau$$

$$\approx \frac{1}{\Delta x} \sum_{k=0}^{n} (u_J^{n+1-k} - u_{J-1}^{n+1-k})\, e^{-iV_r k\Delta t} \int_{t_k}^{t_{k+1}} \frac{d\tau}{\sqrt{\tau}}$$

$$= \frac{2\sqrt{\Delta t}}{\Delta x} \sum_{k=0}^{n} \frac{(u_J^{n+1-k} - u_{J-1}^{n+1-k})\, e^{-iV_r k\Delta t}}{\sqrt{k+1}+\sqrt{k}}$$

Discretizations by quadrature formulas

- first idea to incorporate TBC in scheme is an ad–hoc discretization

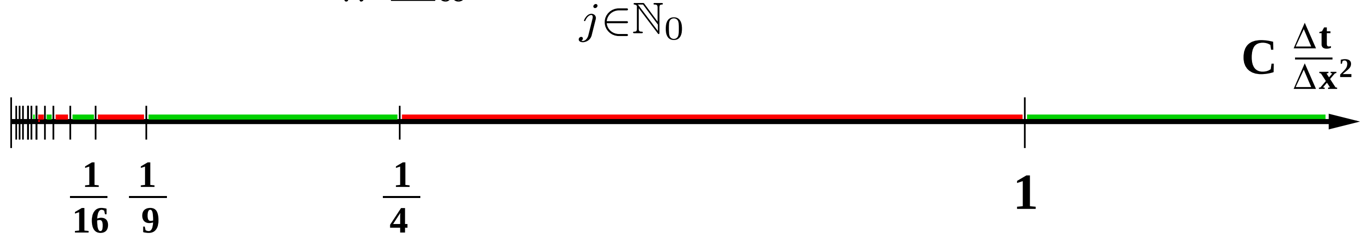
$\rightsquigarrow$ discretized TBC for the SE

$$u_J^{n+1} - u_{J-1}^{n+1} = \frac{\Delta x}{2B\sqrt{\Delta t}}\, u_J^{n+1} - \sum_{k=1}^{n} \bigl(u_J^{n+1-k} - u_{J-1}^{n+1-k}\bigr)\, \tilde{\ell}_k$$

with $\quad B = -\frac{1}{\sqrt{2\pi}}\, e^{i\frac{\pi}{4}}, \qquad \tilde{\ell}_k = \frac{e^{-iV_r k\Delta t}}{\sqrt{k+1}+\sqrt{k}}$

- **Theorem** [Mayfield ’89]: CN–FD scheme for the Schrödinger equation with a certain discretized analytic TBCs is stable $\Longleftrightarrow$

$$\frac{4}{\pi}\frac{\Delta t}{\Delta x^2} \in \bigcup_{j\in\mathbb{N}_0} \left[(2j+1)^{-2}, (2j)^{-2}\right]$$



Rational approximations of the Fourier symbol

- In pseudodifferential calculus the Laplace transform of the kernel $\sqrt[+]{s}$ is identified with the Fourier symbol $\sqrt[+]{i\omega}$
- now: rational approximations of Fourier symbol of convolution kernel
- fractional derivative operator $\partial_t^{1/2}$ in analytic TBC is nonlocal–in–time (due to the non–polynomial nature of its Fourier symbol $\sqrt[+]{i\omega}$)
- **Example:** In the spatial discrete case the Fourier symbol is given in

$$\chi(s) = 1 - \frac{\Delta x^2}{2}(is - V_r) + \sqrt[+]{\frac{\Delta x^2}{2}(is - V_r)\left(\frac{\Delta x^2}{2}(is - V_r) - 2\right)}$$

- rational approximation of these symbols allows for a local–in–time approximated convolution
- For all of the subsequent methods some a-priori information on the dominant wavenumber of the solution at the boundary is needed

Bruneau–Di Menza, Szeftel, Shibata, and Kuska

- right analytic TBC in Fourier space ($V_r = 0$)

$$\boxed{\partial_x \hat{u}(x_r, \omega) = -e^{i\pi/4} \sqrt[+]{i\omega}\, \hat{u}(x_r, \omega)}$$

- approximate the symbol $\sqrt[+]{i\omega}$ by a rational function

$$R_m(i\omega) = \sum_{k=0}^{m} a_k^m - \sum_{k=1}^{m} \frac{a_k^m d_k^m}{i\omega + d_k^m}, \qquad a_k^m, d_k^m > 0$$

ru–DiMe require $R_m(i\omega)$ to interpolate $\sqrt[+]{i\omega}$ at $2m+1$ distinct points

Szeftel also uses a rational approximation of $\sqrt[+]{i\omega}$, but different coefficients:

- Padé approximation
- coefficients by minimizing the reflection coefficient

Shibata linear approximation with two intersection points

Kuska 1/1–Padé approximation about dominant frequency ω_0 to choose

Approach of Fevens and Jiang

- family of absorbing boundary conditions (ABCs) of order p

$$\prod_{l=1}^{p}\left(i\frac{\partial}{\partial x}+a_l\right)u=0, \quad p\in\mathbb{N}$$

- from the shape $u=e^{i(kx-\omega t)}$ of a plane wave one sees:
  all waves with wavenumber $k=a_l$ are perfectly absorbed
- well–posedness for this class of (analytic) ABCs was established
- with low order choices $p=2$ or $p=3$; $a_1=a_2=a_3$ one recovers the ABCs of Shibata and of Kuska

Numerical Example

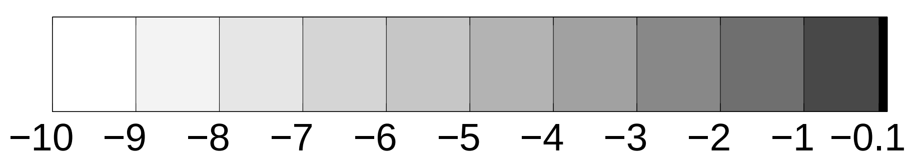
- SE with a vanishing potential $V \equiv 0$ and a Gaussian initial condition

$$u^I(x) = \exp(-x^2 + ik_0 x), \quad x \in \mathbb{R}$$

  - analytic solution can be calculated explicitly

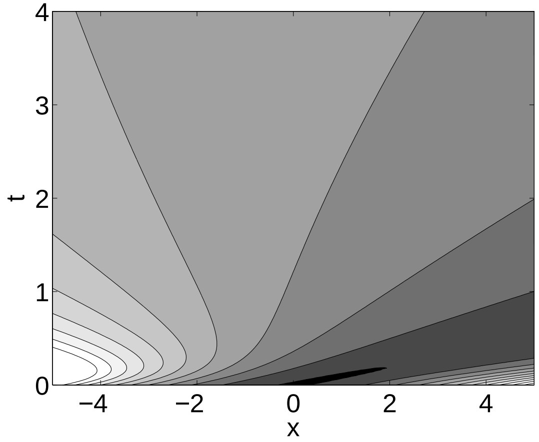
$$u_{\text{ex}}(x,t) = \sqrt{\frac{i}{-4t+i}} \exp\left(\frac{-ix^2 - k_0 x + k_0^2 t}{-4t+i}\right), \quad x \in \mathbb{R},\ t > 0$$

- comput. domain $\Omega_{int} =]-5,5[$, frequency $k_0 = 8$, final time $T_f = 4$
- high frequency of solution needs a very good approximation of $\partial_t^{1/2}$
- $J = 40000$ grid points in spatial direction and time step $\Delta t = 10^{-4}$
- contour of $\log_{10}(|u|)$ to show small level of reflections
  (numerical reflections cannot be visualized in traditional contour plot)

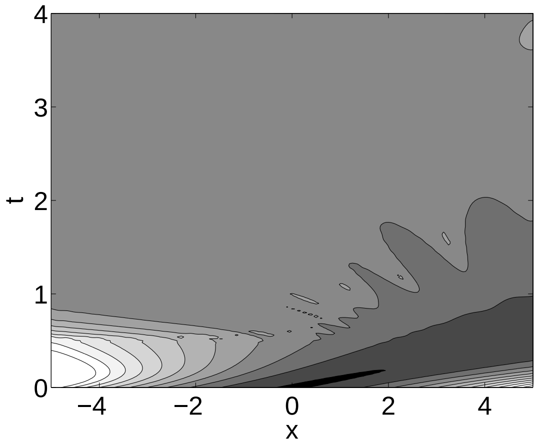


Example: Fevens–Jiang family

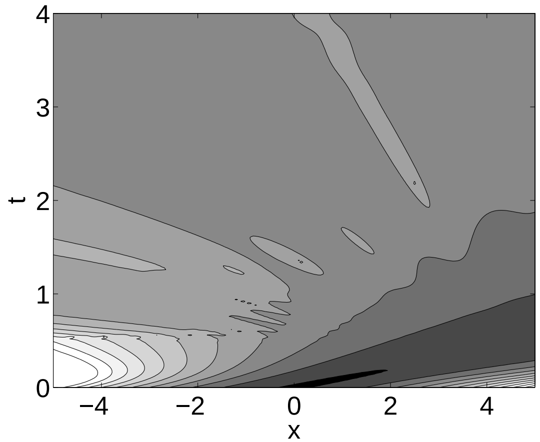
- Figure (a) contour plot of the solution for the fully discrete scheme
- By construction there are no reflections at all from the discrete TBC, so it serves as a reference for the other methods



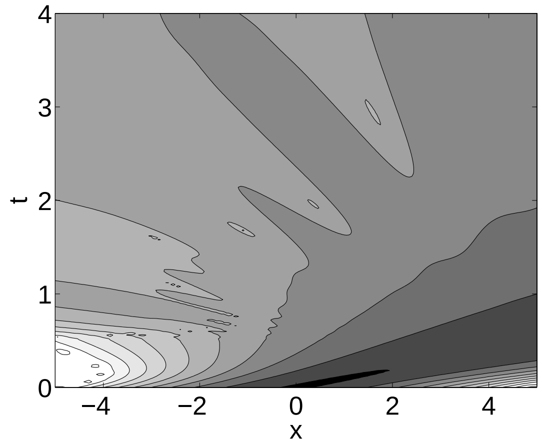
(a) Arnold–Ehrhardt



(b) Shibata



(c) Kuska

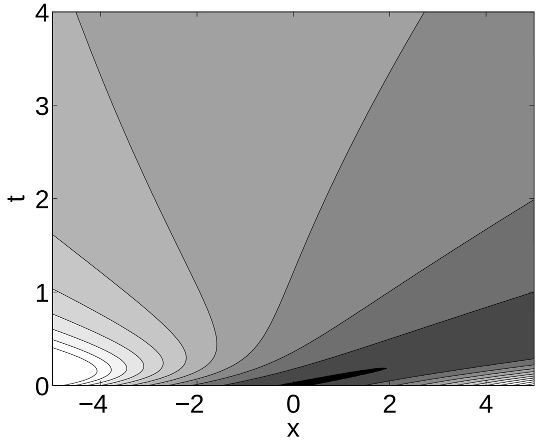


(d) Fevens

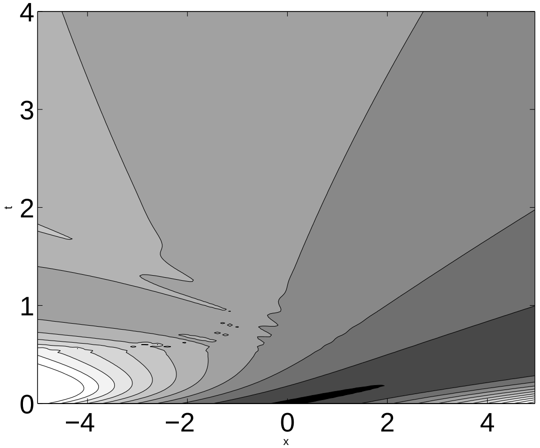
- a levels increase improves significantly the solution
- these older methods are less competitive

Example: Approximation of the square root

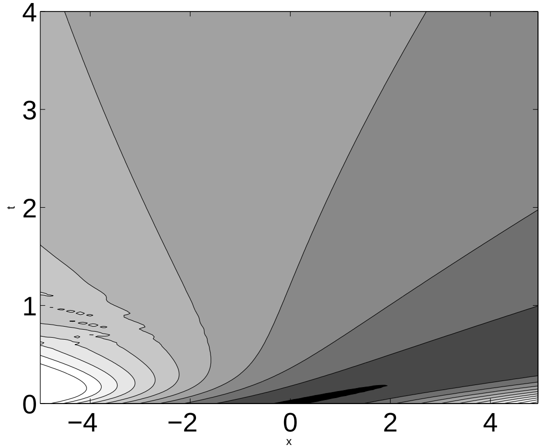
- Figure (e) again as reference solution
- method (f) needs 27 coefficients
- method (g) uses 20 Padé coefficients
- method (h) only uses 3 coefficients (minimization of reflection coefficient)



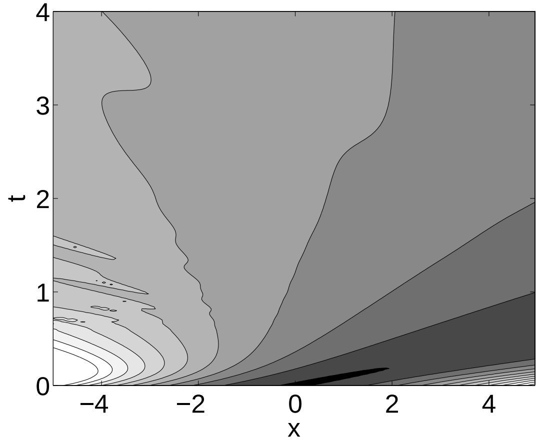
(e) Arnold–Ehrhardt



(f) Di Menza



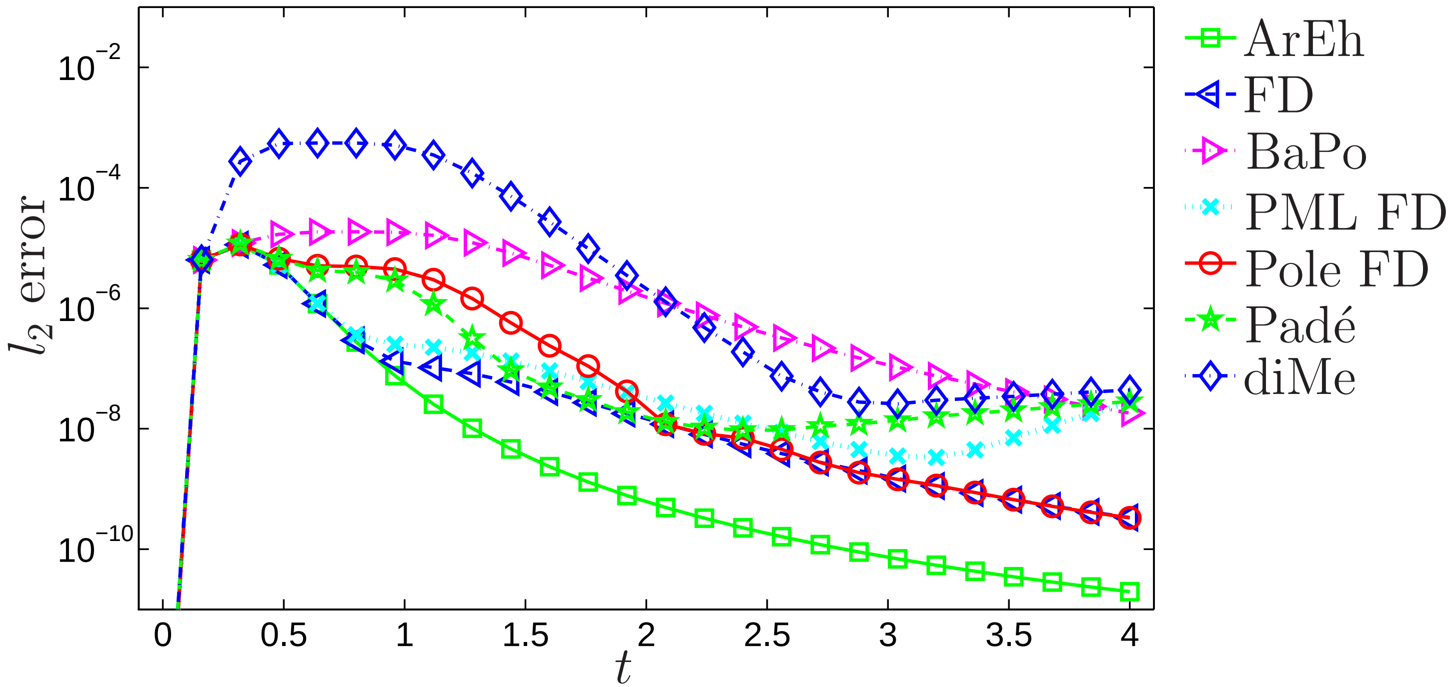
(g) Padé



(h) Szeftel

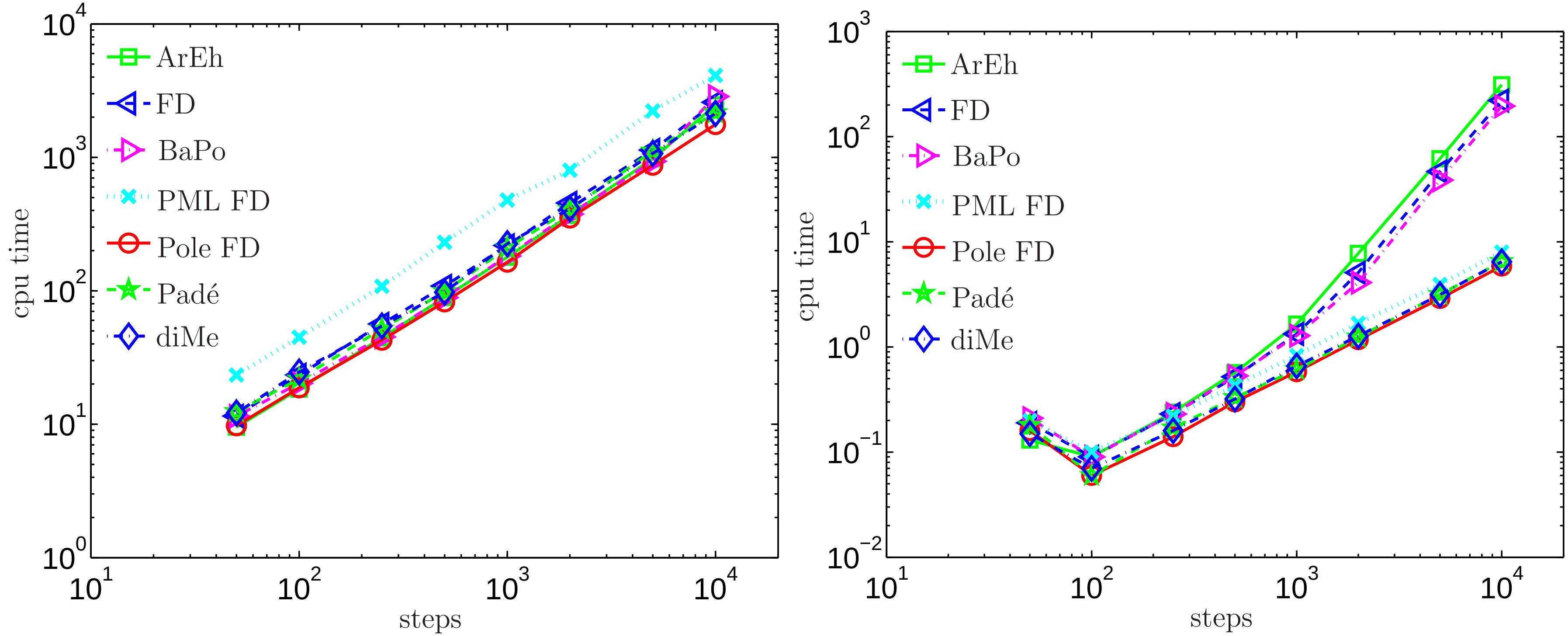
- solutions built with a square root approximation are far better

Example: first group of methods



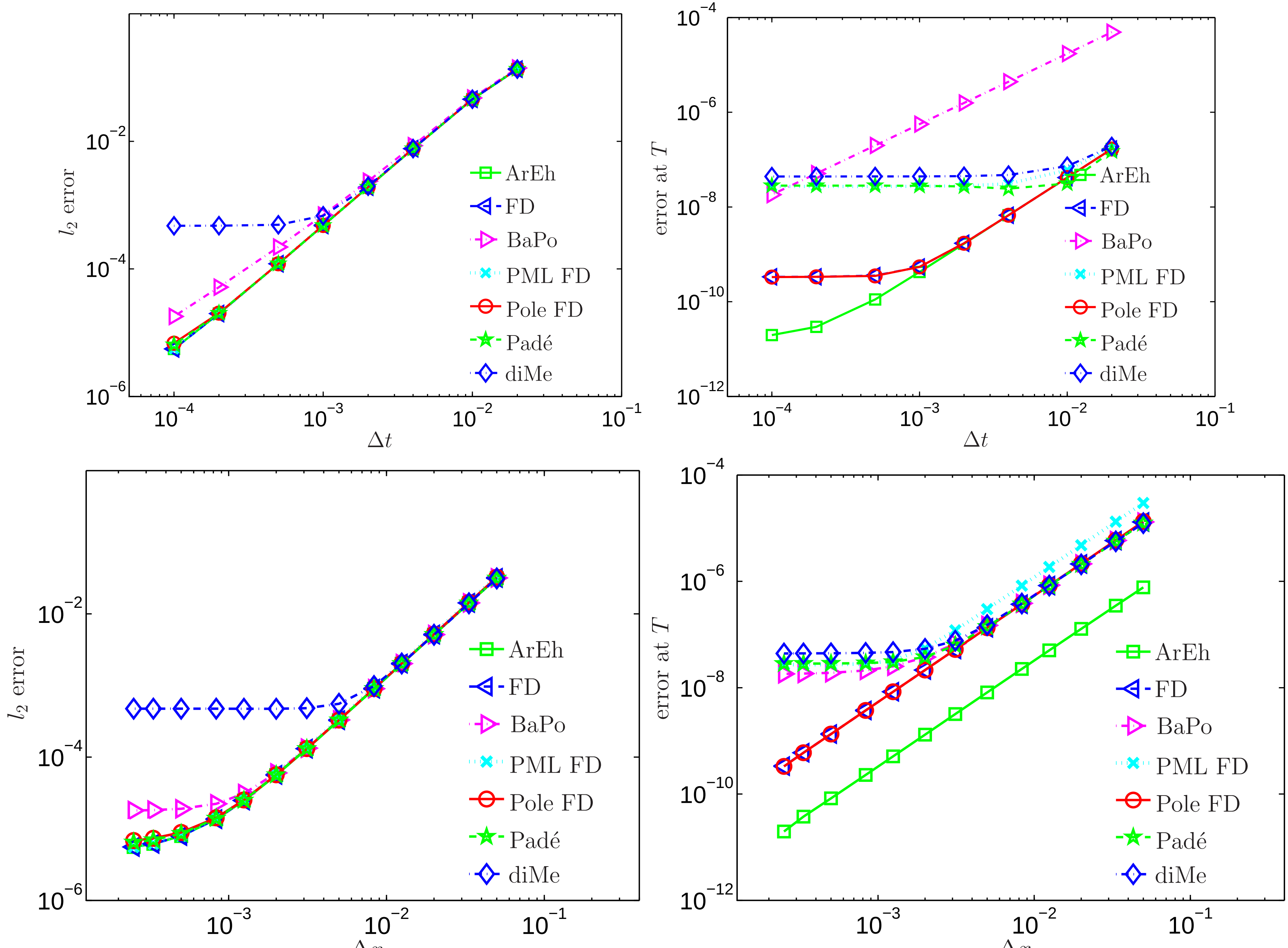
- Time evolution of the spatial l_2 error for various finite difference methods and the fixed step sizes $\Delta x = 2.5 \cdot 10^{-4}$, $\Delta t = 10^{-4}$
- method by Di Menza (**diMe**) shows strong reflection producing an error about 10^2 times larger than the interior discretization error
- method of Baskakov–Popov (**BaPo**) induces a reflection that is about the same magnitude as the interior discretization error

Example: cpu time as function of number of steps

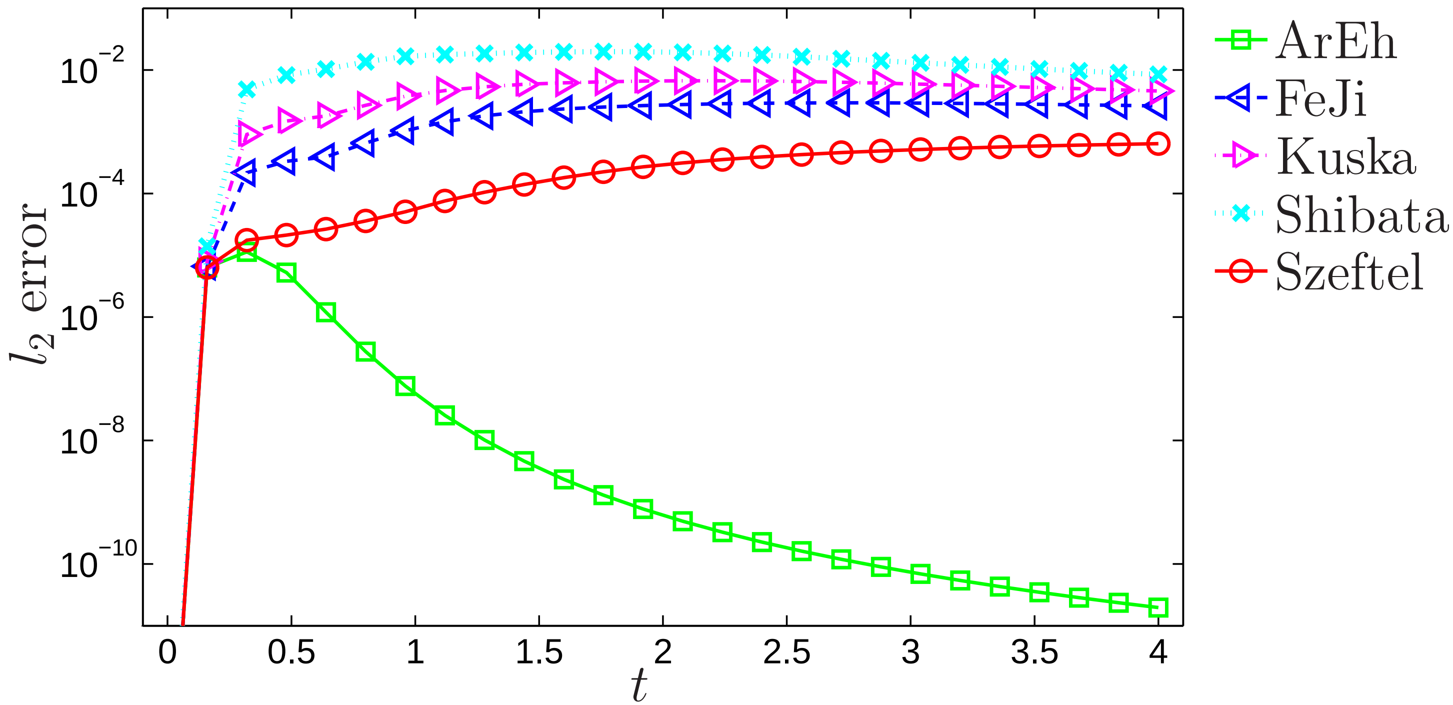


- **left figure:** fine spatial mesh (#unknowns 40.000)
  - solution of the linear system is most time consuming part
    $\Longrightarrow$ the different methods can hardly be distinguished
- **right figure:** coarse spatial mesh (about 200 unknowns)
  - **ArEh**, **FD** and **BaPo** methods do not scale linearly
    quadratic operation count of the convolution starts to dominate

Example: convergence of the methods

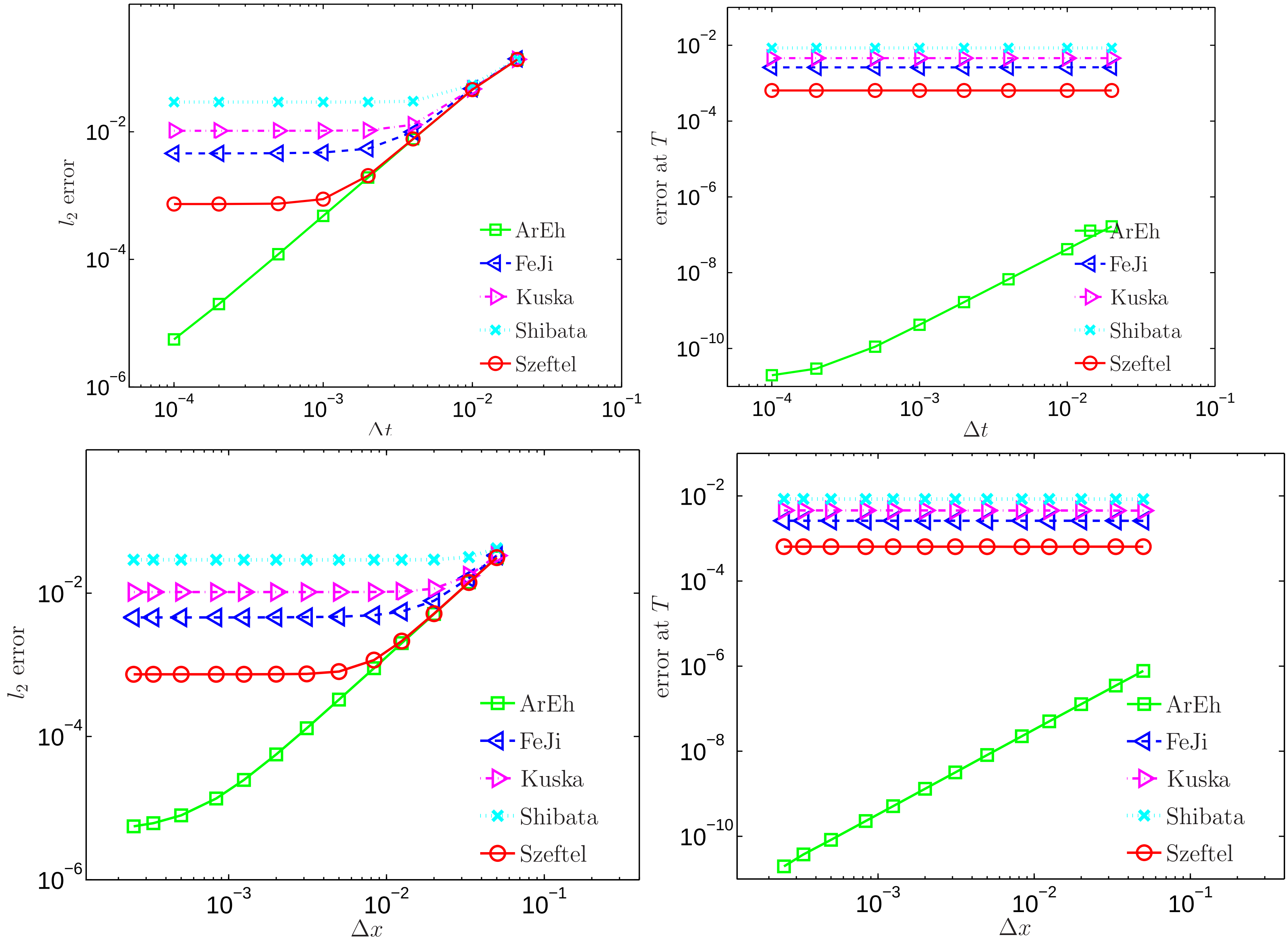


Example: second group of methods

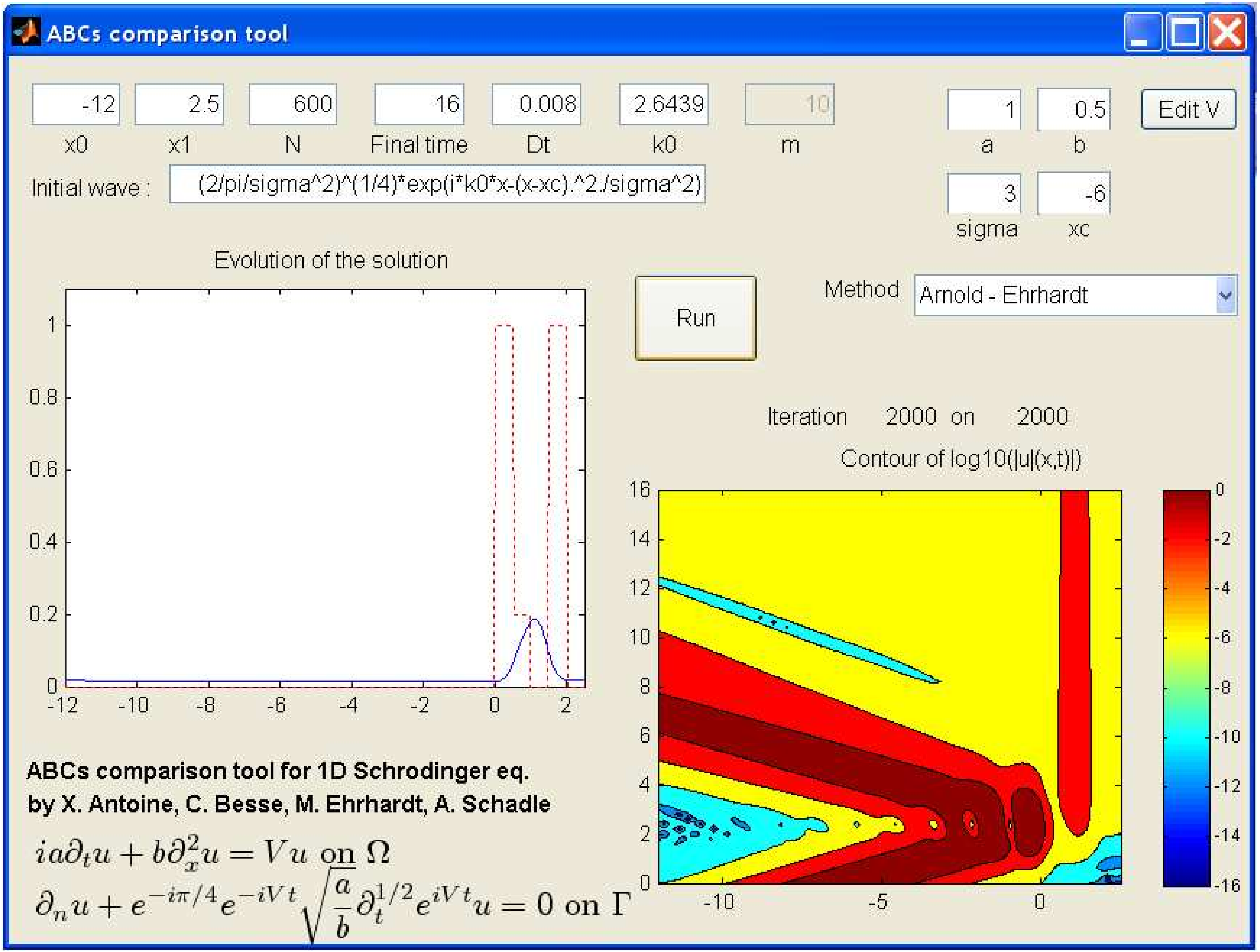


- Time evolution of the spatial l_2 error for various finite difference methods and the fixed step sizes $\Delta x = 2.5 \cdot 10^{-4}$, $\Delta t = 10^{-4}$
- methods of the Fevens–Jiang family **FeJi**, Kuska and Shibata all show strong reflections
- method by Szeftel performs better (uses only 3 coefficients!)

Example: convergence of the methods



Comparison tool for 1D Schrödinger equation



ABCs for Nonlinear Schrödinger Equations

new field [Szeftel ’05], [Antoine, Besse & Descombes ’05], [Zheng ’05]

- Schrödinger equation $iu_t + u_{xx} + f(|u|^2)u = 0$ (NLS)
  e.g. the cubic Schrödinger equation ($f = \lambda|u|^2$)
- linear TBCs work well for reaction–diffusion eqs. but not for NLS
- Problem: integral transformation methods for (discrete) TBCs for linear equations do not work for nonlinear problems!

ABCs for Nonlinear Schrödinger Equations

new field [Szeftel '05], [Antoine, Besse & Descombes '05], [Zheng '05]

- Schrödinger equation $iu_t + u_{xx} + f(|u|^2)u = 0$ (NLS)
  e.g. the cubic Schrödinger equation ($f = \lambda|u|^2$)
- linear TBCs work well for reaction–diffusion eqs. but not for NLS
- Problem: integral transformation methods for (discrete) TBCs for linear equations do not work for nonlinear problems!
- Idea: transform NLS into a linear equation (with variable coefficients)
  - potential strategy
  - phase function approach
  - paralinearization

  apply strategy of [Engquist & Majda '79] (adapted for Schrödinger eq.) and backtransform the obtained ABC

The Potential Strategy

We consider the cubic nonlinear Schrödinger equation

$$\begin{cases} iu_t + u_{xx} + \lambda |u|^2 u = 0, & x \in \mathbb{R}, \quad t > 0 \\ u(x,0) = u^I(x), & x \in \mathbb{R} \end{cases}$$

- Nonlinearity = potential multiplied by the unknown function u
  $\rightsquigarrow$ linear Schrödinger equation with a potential $V(x,t) = \lambda |u(x,t)|^2$

The Potential Strategy

We consider the cubic nonlinear Schrödinger equation

$$\begin{cases} iu_t + u_{xx} + \lambda |u|^2 u = 0, & x \in \mathbb{R}, \quad t > 0 \\ u(x,0) = u^I(x), & x \in \mathbb{R} \end{cases}$$

- Nonlinearity = potential multiplied by the unknown function u
  $\rightsquigarrow$ linear Schrödinger equation with a potential $V(x,t) = \lambda |u(x,t)|^2$
- Method of [Engquist & Majda '79] $\rightsquigarrow$ second order absorbing BC

$$u_x + e^{-i\pi/4} \partial_t^{1/2} u - \frac{V}{2} e^{i\pi/4} I_t^{1/2} u = 0, \qquad x = 0 \qquad \text{(ABC2)}$$

The Potential Strategy

We consider the cubic nonlinear Schrödinger equation

$$\begin{cases} iu_t + u_{xx} + \lambda |u|^2 u = 0, & x \in \mathbb{R}, \quad t > 0 \\ u(x,0) = u^I(x), & x \in \mathbb{R} \end{cases}$$

- Nonlinearity = potential multiplied by the unknown function u
  $\rightsquigarrow$ linear Schrödinger equation with a potential $V(x,t) = \lambda |u(x,t)|^2$
- Method of [Engquist & Majda '79] $\rightsquigarrow$ second order absorbing BC

$$u_x + e^{-i\pi/4} \partial_t^{1/2} u - \frac{V}{2} e^{i\pi/4} I_t^{1/2} u = 0, \qquad x = 0 \qquad \text{(ABC2)}$$

- Recalling $V = \lambda |u|^2$ (ABC2) becomes

$$\boxed{u_x + e^{-i\pi/4} \partial_t^{1/2} u - \lambda \frac{|u|^2}{2} e^{i\pi/4} I_t^{1/2} u = 0, \qquad x = 0}$$

$$\partial_t^{1/2} \varphi(t) = \frac{1}{\sqrt{\pi}} \partial_t \int_0^t \frac{\varphi(\tau)}{\sqrt{t-\tau}} \, d\tau, \qquad I_t^{1/2} \varphi(t) = \frac{1}{\sqrt{\pi}} \int_0^t \frac{\varphi(\tau)}{\sqrt{t-\tau}} \, d\tau$$

The Phase Function Transformation Strategy

We consider the cubic nonlinear Schrödinger equation

$$\begin{cases} iu_t + u_{xx} + \lambda |u|^2 u = 0, & x \in \mathbb{R}, \quad t > 0 \\ u(x,0) = u^I(x), & x \in \mathbb{R} \end{cases}$$

- introduce phase function $\mathcal{V}(x,t) = \lambda \int_0^t |u(x,\tau)|^2 \, d\tau$

The Phase Function Transformation Strategy

We consider the cubic nonlinear Schrödinger equation

$$\begin{cases} iu_t + u_{xx} + \lambda |u|^2 u = 0, & x \in \mathbb{R}, \quad t > 0 \\ u(x,0) = u^I(x), & x \in \mathbb{R} \end{cases}$$

- introduce phase function $\mathcal{V}(x,t) = \lambda \int_0^t |u(x,\tau)|^2 \, d\tau$
- new unknown function $v(x,t) = e^{i\mathcal{V}(x,t)} u(x,t)$ satisfies

$$\begin{cases} iv_t + v_{xx} + 2i\mathcal{V}_x v_x + (i\mathcal{V}_x x - (\mathcal{V}_x)^2) v = 0, & x \in \mathbb{R}, \quad t > 0 \\ v(x,0) = u^I(x), & x \in \mathbb{R} \end{cases}$$

$\leadsto$ linear variable coefficients Schrödinger equation

The Phase Function Transformation Strategy

We consider the cubic nonlinear Schrödinger equation

$$\begin{cases} iu_t + u_{xx} + \lambda |u|^2 u = 0, & x \in \mathbb{R}, \quad t > 0 \\ u(x,0) = u^I(x), & x \in \mathbb{R} \end{cases}$$

- introduce phase function $\mathcal{V}(x,t) = \lambda \int_0^t |u(x,\tau)|^2 \, d\tau$
- new unknown function $v(x,t) = e^{i\mathcal{V}(x,t)} u(x,t)$ satisfies

$$\begin{cases} iv_t + v_{xx} + 2i\mathcal{V}_x v_x + (i\mathcal{V}_x x - (\mathcal{V}_x)^2) v = 0, & x \in \mathbb{R}, \quad t > 0 \\ v(x,0) = u^I(x), & x \in \mathbb{R} \end{cases}$$

$\rightsquigarrow$ linear variable coefficients Schrödinger equation

- Method of [Engquist & Majda '79] $\rightsquigarrow$ second order absorbing BC

$$\boxed{u_x + e^{-i\pi/4} e^{i\mathcal{V}} \partial_t^{1/2} (e^{-i\mathcal{V}} u) + i \frac{V_x}{4} e^{i\mathcal{V}} I_t^{1/2} (e^{-i\mathcal{V}} u) = 0, \quad x = 0}$$

asymptotic ABCs by Leibnitz derivation rule for fractional operators

Conclusions

We have presented several different techniques to solve numerically the time–dependent SE on unbounded domains

- comparison of several implementations of the classical TBC and ABCs into finite difference (and finite element discretizations)
- mainly the one-dimensional case but also the cubic nonlinear case (cf. review paper for the 2D case)

Conclusions

We have presented several different techniques to solve numerically the time–dependent SE on unbounded domains

- comparison of several implementations of the classical TBC and ABCs into finite difference (and finite element discretizations)
- mainly the one-dimensional case but also the cubic nonlinear case (cf. review paper for the 2D case)
- several numerical examples comparing the proposed boundary conditions w.r.t. different kind of errors and the cpu time
- freely available graphical user interface (written in MATLAB) covering most of the presented approaches

Conclusions

We have presented several different techniques to solve numerically the time–dependent SE on unbounded domains

- comparison of several implementations of the classical TBC and ABCs into finite difference (and finite element discretizations)
- mainly the one-dimensional case but also the cubic nonlinear case (cf. review paper for the 2D case)
- several numerical examples comparing the proposed boundary conditions w.r.t. different kind of errors and the cpu time
- freely available graphical user interface (written in MATLAB) covering most of the presented approaches
- review paper / software: `http://www.tbc-review.de.vu`

Conclusions

We have presented several different techniques to solve numerically the time–dependent SE on unbounded domains

- comparison of several implementations of the classical TBC and ABCs into finite difference (and finite element discretizations)
- mainly the one-dimensional case but also the cubic nonlinear case (cf. review paper for the 2D case)
- several numerical examples comparing the proposed boundary conditions w.r.t. different kind of errors and the cpu time
- freely available graphical user interface (written in MATLAB) covering most of the presented approaches
- review paper / software: `http://www.tbc-review.de.vu`

Thank You for Your Attention!