MPDATA meets Black-Scholes: derivative pricing as a transport problem

Sylwester Arabas and Ahmad Farhat

background, affiliations

Sylwester Arabas

University of Warsaw
(MSc, physics)

↓

University of Warsaw
(PhD, [geo]physics)

↘

Ahmad Farhat

American University of Beirut
(MSc, mathematics)

↓

University of Wrocław
(PhD, topology)

↙

Chatham Financial, Cracow
(Models Dev. Team)

↙ AETHON, Athens
(H2020 MoveWise project)

↓

Jagiellonian Univ. Cracow
(from Oct. 2018)

↘ HSBC, Cracow
(Quant Team)

this talk (arXiv:1607.01751)

- MPDATA
- libmpdata++
- derivative pricing as a transport problem

this talk (arXiv:1607.01751)

- MPDATA
- libmpdata++
- derivative pricing as a transport problem

in line with the proposal put forward in Duffy 2004

to investigate robust and effective numerical schemes documented in the computational fluid dynamics literature as alternatives to commonly used numerical schemes in financial engineering, with the aim of “improving the finite difference methods gene pool as it were.”

(“A critique of the Crank-Nicolson scheme, strengths and weaknesses for financial instrument pricing”, WILMOTT 4)

MPDATA

MPDATA in a nutshell (Smolarkiewicz 1983, 1984, . . .)

transport PDE: $\dfrac{\partial \psi}{\partial t} + \dfrac{\partial}{\partial x}(v\psi) = 0$

MPDATA in a nutshell (Smolarkiewicz 1983, 1984, . . .)

transport PDE: $\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) = 0$

$$\psi_i^{n+1} = \psi_i^n - \left[F(\psi_i^n, \psi_{i+1}^n, \mathcal{C}_{i+1/2}) - F(\psi_{i-1}^n, \psi_i^n, \mathcal{C}_{i-1/2})\right]$$
$$F(\psi_L, \psi_R, \mathcal{C}) = \max(\mathcal{C}, 0) \cdot \psi_L + \min(\mathcal{C}, 0) \cdot \psi_R$$
$$\mathcal{C} = v\Delta t / \Delta x$$

upwind

MPDATA in a nutshell (Smolarkiewicz 1983, 1984, . . .)

transport PDE: $$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) = 0$$

$$\psi_i^{n+1} = \psi_i^n - \left[F(\psi_i^n, \psi_{i+1}^n, \mathcal{C}_{i+1/2}) - F(\psi_{i-1}^n, \psi_i^n, \mathcal{C}_{i-1/2})\right]$$
$$F(\psi_L, \psi_R, \mathcal{C}) = \max(\mathcal{C}, 0) \cdot \psi_L + \min(\mathcal{C}, 0) \cdot \psi_R$$
$$\mathcal{C} = v\Delta t/\Delta x$$

← upwind

modified eq.: $$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) + \underbrace{K\frac{\partial^2 \psi}{\partial x^2}}_{\text{numerical diffusion}} + \ldots = 0$$ ← MEA

MPDATA in a nutshell (Smolarkiewicz 1983, 1984, . . .)

transport PDE: $$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) = 0$$

$$\psi_i^{n+1} = \psi_i^n - \left[F(\psi_i^n, \psi_{i+1}^n, \mathcal{C}_{i+1/2}) - F(\psi_{i-1}^n, \psi_i^n, \mathcal{C}_{i-1/2})\right]$$
$$F(\psi_L, \psi_R, \mathcal{C}) = \max(\mathcal{C}, 0) \cdot \psi_L + \min(\mathcal{C}, 0) \cdot \psi_R$$
$$\mathcal{C} = v\Delta t/\Delta x$$

← upwind

modified eq.: $$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) + \underbrace{K\frac{\partial^2 \psi}{\partial x^2}}_{\text{numerical diffusion}} + \ldots = 0$$ ← MEA

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) + \frac{\partial}{\partial x}\underbrace{\left[\left(-\frac{K}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right]}_{\text{antidiffusive flux}} = 0 \;\leftarrow$$

MPDATA in a nutshell (Smolarkiewicz 1983, 1984, ...)

transport PDE: $\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) = 0$

$$\psi_i^{n+1} = \psi_i^n - \left[F(\psi_i^n, \psi_{i+1}^n, \mathcal{C}_{i+1/2}) - F(\psi_{i-1}^n, \psi_i^n, \mathcal{C}_{i-1/2})\right]$$
$$F(\psi_L, \psi_R, \mathcal{C}) = \max(\mathcal{C}, 0)\cdot\psi_L + \min(\mathcal{C}, 0)\cdot\psi_R$$
$$\mathcal{C} = v\Delta t/\Delta x$$

← upwind

modified eq.: $\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) + \underbrace{K\frac{\partial^2 \psi}{\partial x^2}}_{\text{numerical diffusion}} + \ldots = 0$ ← MEA

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) + \frac{\partial}{\partial x}\underbrace{\left[\left(-\frac{K}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right]}_{\text{antidiffusive flux}} = 0 \leftarrow$$

MPDATA: reverse numerical diffusion by integrating the antidiffusive flux using upwind (in a corrective iteration)

$$\mathcal{C}'_{i+1/2} = (|\mathcal{C}_{i+1/2}| - \mathcal{C}^2_{i+1/2})A_{i+1/2}$$
$$A_{i+1/2} = \frac{\psi_{i+1} - \psi_i}{\psi_{i+1} + \psi_i}$$

MPDATA: key features (review: e.g. Smolarkiewicz 2006)

MPDATA

Multidimensional **P**ositive **D**efinite Advection Transport Algorithm

MPDATA

Multidimensional **P**ositive **D**efinite Advection Transport Algorithm

- **M**ultidimensionality:
  antidiffusive fluxes include cross-dimensional terms, as opposed to dimensionally-split schemes

MPDATA

Multidimensional **P**ositive **D**efinite Advection Transport Algorithm

- **M**ultidimensionality:
  antidiffusive fluxes include cross-dimensional terms, as opposed to dimensionally-split schemes
- **P**ositive **D**efiniteness:
  sign-preserving + "infinite-gauge formulation for variable-sign fields

MPDATA

Multidimensional **P**ositive **D**efinite Advection Transport Algorithm

- **M**ultidimensionality:
  antidiffusive fluxes include cross-dimensional terms, as opposed to dimensionally-split schemes
- **P**ositive **D**efiniteness:
  sign-preserving + "infinite-gauge formulation for variable-sign fields
- Conservativeness:
  upstream for all iterations ($\rightsquigarrow$ stability cond.)

MPDATA

Multidimensional **P**ositive **D**efinite Advection Transport Algorithm

- **M**ultidimensionality:
  antidiffusive fluxes include cross-dimensional terms, as opposed to dimensionally-split schemes
- **P**ositive **D**efiniteness:
  sign-preserving + "infinite-gauge formulation for variable-sign fields
- Conservativeness:
  upstream for all iterations ($\rightsquigarrow$ stability cond.)
- High-Order Accuracy:
  up to 3rd-order in time and space (dep. on options & flow)

MPDATA

Multidimensional **P**ositive **D**efinite Advection Transport Algorithm

- **M**ultidimensionality:
  antidiffusive fluxes include cross-dimensional terms, as opposed to dimensionally-split schemes
- **P**ositive **D**efiniteness:
  sign-preserving + “infinite-gauge formulation for variable-sign fields
- Conservativeness:
  upstream for all iterations ($\rightsquigarrow$ stability cond.)
- High-Order Accuracy:
  up to 3rd-order in time and space (dep. on options & flow)
- Monotonicity:
  with Flux-Corrected Transport option

libmpdata++

libmpdata++: new open-source implementation in C++

Jaruga et al. 2015

Geosci. Model Dev., 8, 1005–1032, 2015
www.geosci-model-dev.net/8/1005/2015/
doi:10.5194/gmd-8-1005-2015

Geoscientific Model Development
Open Access

libmpdata++ 1.0: a library of parallel MPDATA solvers for systems of generalised transport equations

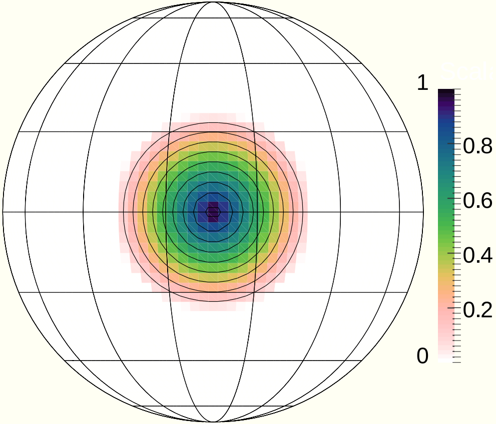
A. Jaruga[1], S. Arabas[1], D. Jarecka[1,2], H. Pawlowska[1], P. K. Smolarkiewicz[3], and M. Waruszewski[1]

[1]Institute of Geophysics, Faculty of Physics, University of Warsaw, Warsaw, Poland
[2]National Center for Atmospheric Research, Boulder, CO, USA
[3]European Centre for Medium-Range Weather Forecasts, Reading, UK

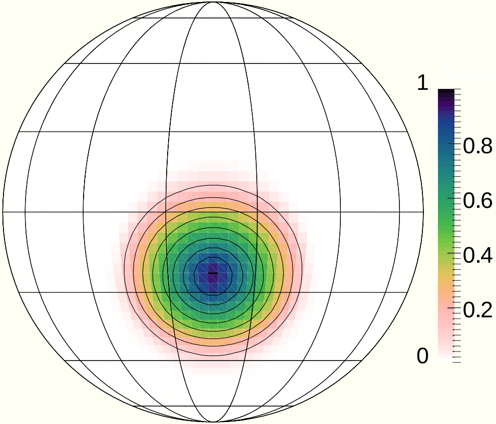
libmpdata++: generalised transport equation

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$

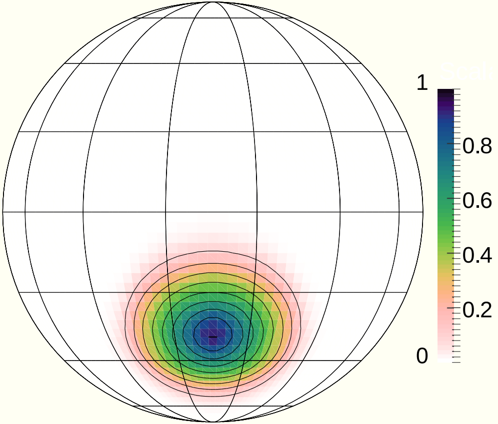


$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$

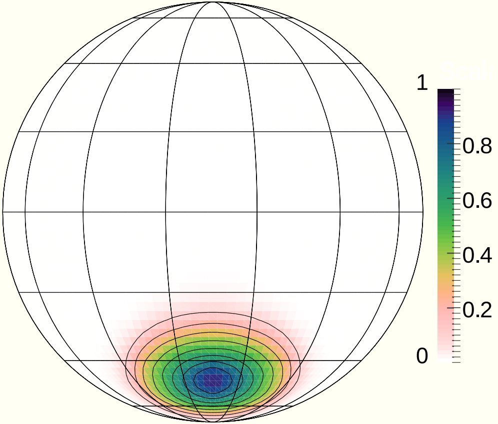


libmpdata++: generalised transport equation

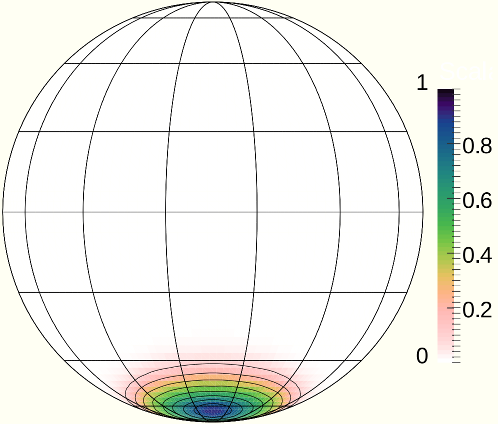
$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$



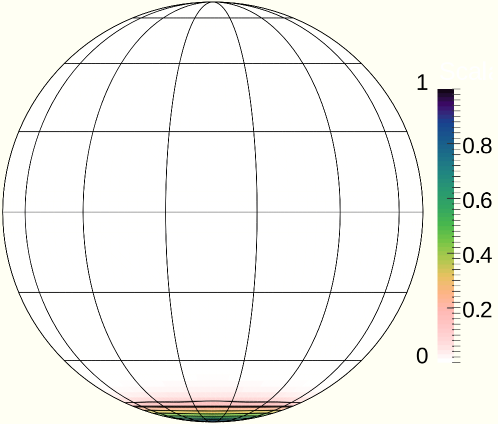
$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$



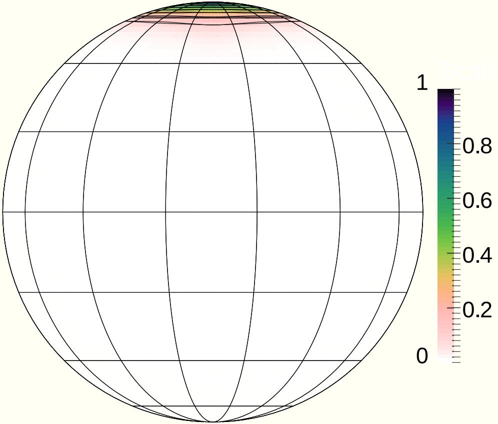
$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$



$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$

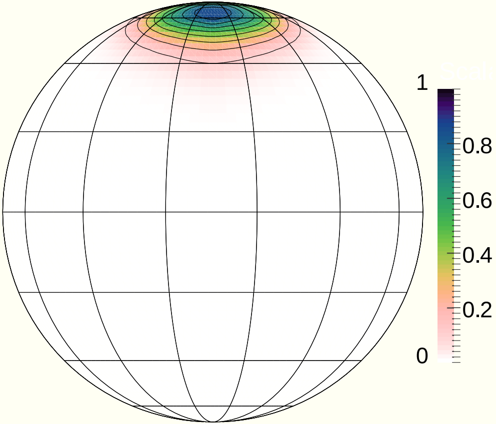


$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$



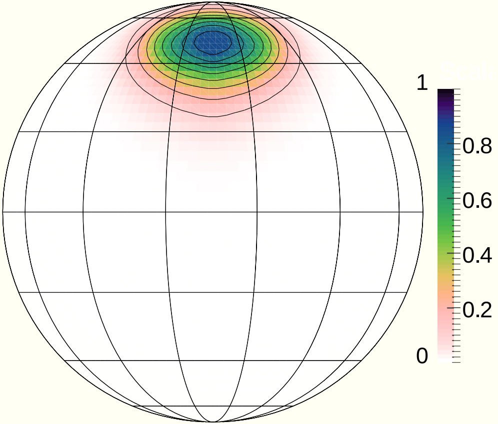
libmpdata++: generalised transport equation

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$



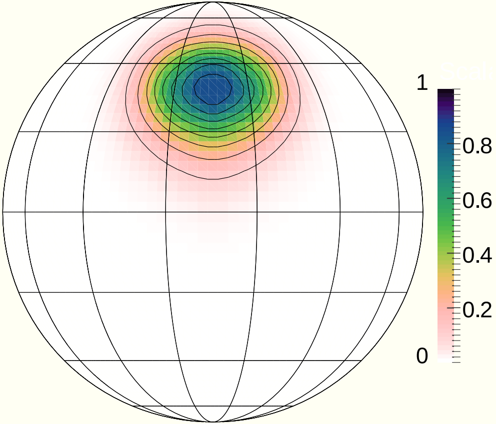
libmpdata++: generalised transport equation

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$

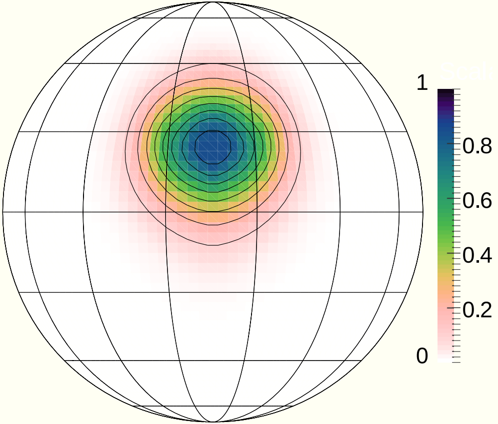


libmpdata++: generalised transport equation

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$

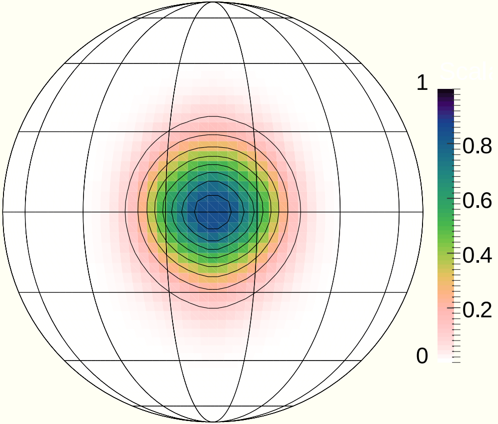


$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$



libmpdata++: generalised transport equation

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = GR$$



libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, ...)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, . . .)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, . . .)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, ...)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, ...)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, ...)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, . . .)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, . . .)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code ($<$ 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, . . .)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, . . .)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, . . .)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, . . .)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, . . .)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code (< 10 kLOC)

libmpdata++: summary & some technicalities

key features (as of v1.0):

- reusable – API documented in the paper; out-of-tree setups
- comprehensive set of MPDATA opts (incl. FCT, infinite-gauge, ...)
- 1D, 2D & 3D integration; optional coordinate transformation
- four types of solvers:
  - adv (homogeneous advection)
  - adv+rhs (+ right-hand-side terms)
  - adv+rhs+vip (+ prognosed velocity)
  - adv+rhs+vip+prs (+ elliptic pressure solver)
- implemented using Blitz++ (no loops, expression templates)
- built-in HDF5/XDMF output
- shared-memory parallelisation using OpenMP or Boost.Thread
- separation of concerns (numerics / boundary cond. / io / concurrency)
- compact C++11 code ($<$ 10 kLOC)

libmpdata++: documented applications

libmpdata++: documented applications

- Jarecka et al. 2015 (J. Comp. Phys.):
  shallow water eqs, 3D liquid drop spreading under gravity

libmpdata++: documented applications

- Jarecka et al. 2015 (J. Comp. Phys.):
  shallow water eqs, 3D liquid drop spreading under gravity
- Arabas, Jaruga et al. 2015 (Geosci. Model. Dev.);
  Jaruga & Pawlowska 2018 (″):
  particle-based/Monte-Carlo simulations of clouds

libmpdata++: documented applications

- Jarecka et al. 2015 (J. Comp. Phys.):
  shallow water eqs, 3D liquid drop spreading under gravity
- Arabas, Jaruga et al. 2015 (Geosci. Model. Dev.);
  Jaruga & Pawlowska 2018 (″):
  particle-based/Monte-Carlo simulations of clouds
- Waruszewski et al. 2018 (J. Comp. Phys.):
  MPDATA ext. for 3rd-order accuracy for variable flows

libmpdata++: documented applications

- Jarecka et al. 2015 (J. Comp. Phys.):
  shallow water eqs, 3D liquid drop spreading under gravity

- Arabas, Jaruga et al. 2015 (Geosci. Model. Dev.);
  Jaruga & Pawlowska 2018 (″):
  particle-based/Monte-Carlo simulations of clouds

- Waruszewski et al. 2018 (J. Comp. Phys.):
  MPDATA ext. for 3rd-order accuracy for variable flows

- Dziekan et al. 2018 (AMS Cloud Physics Conf.):
  3D LES for atm. boundary layer simulations

libmpdata++: documented applications

- Jarecka et al. 2015 (J. Comp. Phys.):
  shallow water eqs, 3D liquid drop spreading under gravity
- Arabas, Jaruga et al. 2015 (Geosci. Model. Dev.);
  Jaruga & Pawlowska 2018 (″):
  particle-based/Monte-Carlo simulations of clouds
- Waruszewski et al. 2018 (J. Comp. Phys.):
  MPDATA ext. for 3rd-order accuracy for variable flows
- Dziekan et al. 2018 (AMS Cloud Physics Conf.):
  3D LES for atm. boundary layer simulations
- MPDATA meets Black-Scholes!

derivative pricing as a transport problem

Black-Scholes equation and pricing formulæ

Black-Scholes equation and pricing formulæ

- asset price SDE: $$dS = S(\mu dt + \sigma dw)$$

Black-Scholes equation and pricing formulæ

- asset price SDE: $dS = S(\mu dt + \sigma dw)$
- derivative price: $f(S, t)$

Black-Scholes equation and pricing formulæ

- asset price SDE: $dS = S(\mu dt + \sigma dw)$
- derivative price: $f(S, t)$
- riskless portfolio (asset + option): $\Pi = -f + \Delta_t S$

Black-Scholes equation and pricing formulæ

- asset price SDE: $dS = S(\mu dt + \sigma dw)$
- derivative price: $f(S, t)$
- riskless portfolio (asset + option): $\Pi = -f + \Delta_t S$
- Itô's lemma: SDE $\rightsquigarrow$ PDE

Black-Scholes equation and pricing formulæ

- asset price SDE: $dS = S(\mu dt + \sigma dw)$
- derivative price: $f(S, t)$
- riskless portfolio (asset + option): $\Pi = -f + \Delta_t S$
- Itô's lemma: SDE $\rightsquigarrow$ PDE
- no arbitrage (riskless interest rate): $d\Pi = \Pi r dt$

Black-Scholes equation and pricing formulæ

- asset price SDE: $dS = S(\mu dt + \sigma dw)$
- derivative price: $f(S, t)$
- riskless portfolio (asset + option): $\Pi = -f + \Delta_t S$
- Itô's lemma: SDE $\rightsquigarrow$ PDE
- no arbitrage (riskless interest rate): $d\Pi = \Pi r dt$

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

Black-Scholes equation and pricing formulæ

- asset price SDE: $dS = S(\mu dt + \sigma dw)$
- derivative price: $f(S, t)$
- riskless portfolio (asset + option): $\Pi = -f + \Delta_t S$
- Itô's lemma: SDE $\rightsquigarrow$ PDE
- no arbitrage (riskless interest rate): $d\Pi = \Pi r dt$

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

- terminal value prob., analytic solutions for vanilla options

Black-Scholes equation and pricing formulæ

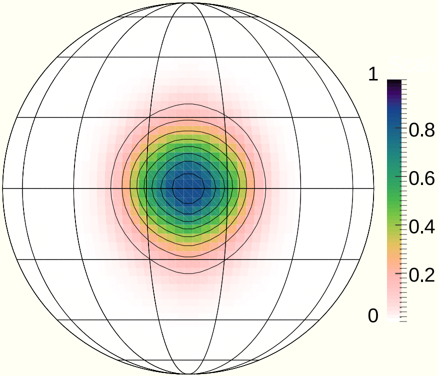
- asset price SDE: $dS = S(\mu dt + \sigma dw)$
- derivative price: $f(S, t)$
- riskless portfolio (asset + option): $\Pi = -f + \Delta_t S$
- Itô's lemma: SDE $\rightsquigarrow$ PDE
- no arbitrage (riskless interest rate): $d\Pi = \Pi r dt$

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

- terminal value prob., analytic solutions for vanilla options


SVERIGES RIKSBANK TILL ALFRED NOBELS MINNE 1968

?


1
0.8
0.6
0.4
0.2
0

Black-Scholes ⇝ ("advection-only") transport problem

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

Black-Scholes ⇝ ("advection-only") transport problem

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\xrightarrow{x=\ln S} \frac{\partial f}{\partial t} + \underbrace{\left(r - {\sigma^2}/{2}\right)}_{u}\frac{\partial f}{\partial x} + \underbrace{{\sigma^2}/{2}}_{-\nu}\frac{\partial^2 f}{\partial x^2} - rf = 0$$

Black-Scholes $\rightsquigarrow$ ("advection-only") transport problem

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\xrightarrow{x = \ln S} \frac{\partial f}{\partial t} + \underbrace{\left(r - \sigma^2/2\right)}_{u}\frac{\partial f}{\partial x} + \underbrace{\sigma^2/2}_{-\nu}\frac{\partial^2 f}{\partial x^2} - rf = 0$$

$$\xrightarrow{\psi = e^{-rt}f} \frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} - \nu\frac{\partial^2 \psi}{\partial x^2} = 0$$

Black-Scholes ⇝ ("advection-only") transport problem

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\xrightarrow{x = \ln S} \frac{\partial f}{\partial t} + \underbrace{(r - \sigma^2/2)}_{u}\frac{\partial f}{\partial x} + \underbrace{\sigma^2/2}_{-\nu}\frac{\partial^2 f}{\partial x^2} - rf = 0$$

$$\xrightarrow{\psi = e^{-rt} f} \frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} - \nu\frac{\partial^2 \psi}{\partial x^2} = 0$$

$$\longrightarrow \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

Black-Scholes ⇝ ("advection-only") transport problem

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\xrightarrow{x = \ln S} \quad \frac{\partial f}{\partial t} + \underbrace{\left(r - \sigma^2/2\right)}_{u}\frac{\partial f}{\partial x} + \underbrace{\sigma^2/2}_{-\nu}\frac{\partial^2 f}{\partial x^2} - rf = 0$$

$$\xrightarrow{\psi = e^{-rt}f} \quad \frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} - \nu\frac{\partial^2 \psi}{\partial x^2} = 0$$

$$\longrightarrow \quad \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

re last step: Smolarkiewicz and Clark (1986, JCP), Sousa (2009, IJNMF), Smolarkiewicz and Szmelter (2005, JCP), Cristiani (2015, JCSMD)

same trick!

MPDATA in a nutshell (Smolarkiewicz 1983, 1984, . . .)

transport PDE: $\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) = 0$

$$\psi_i^{n+1} = \psi_i^n - \left[F(\psi_i^n, \psi_{i+1}^n, \mathcal{C}_{i+1/2}) - F(\psi_{i-1}^n, \psi_i^n, \mathcal{C}_{i-1/2})\right]$$
$$F(\psi_L, \psi_R, \mathcal{C}) = \max(\mathcal{C}, 0) \cdot \psi_L + \min(\mathcal{C}, 0) \cdot \psi_R$$
$$\mathcal{C} = v\Delta t/\Delta x$$

← upwind

modified eq.: $\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) + \underbrace{K\frac{\partial^2 \psi}{\partial x^2}}_{\text{numerical diffusion}} + \ldots = 0$ ← MEA

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}(v\psi) + \frac{\partial}{\partial x}\underbrace{\left[\left(-\frac{K}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right]}_{\text{antidiffusive flux}} = 0$$

Black-Scholes ⇝ ("advection-only") transport problem

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$x = \ln S$ →

$$\frac{\partial f}{\partial t} + \underbrace{(r - \sigma^2/2)}_{u}\frac{\partial f}{\partial x} + \underbrace{\sigma^2/2}_{-\nu}\frac{\partial^2 f}{\partial x^2} - rf = 0$$

$\psi = e^{-rt}f$ →

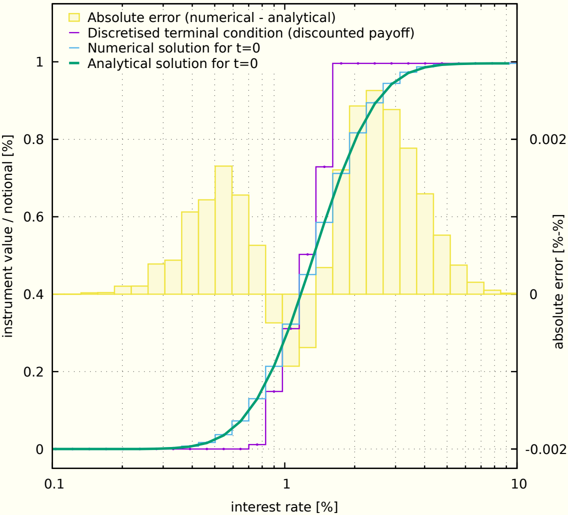
$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} - \nu\frac{\partial^2 \psi}{\partial x^2} = 0$$

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

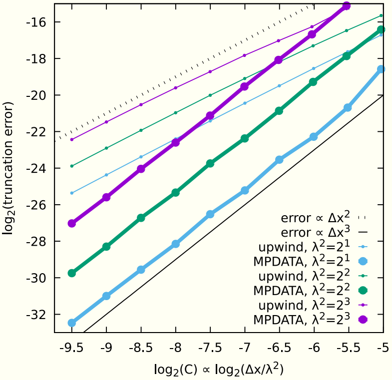
MPDATA meets Black-Scholes: test case

- payoff function: corridor
- truncation error est. (ψ_a: B-S formula):

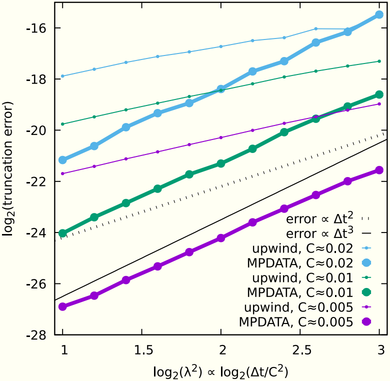
$$E = \sqrt{\sum_{i=1}^{n_x} \left[\psi_n(x_i) - \psi_a(x_i)\right]^2 / (n_x \cdot n_t)}\Bigg|_{t=0}$$



MPDATA meets Black-Scholes: convergence analysis



Truncation error as a function of the Courant number $C = u\frac{\Delta t}{\Delta x}$ which, for fixed λ^2, is proportional to the gridstep.



Truncation error as a function of the λ^2 parameter which, for fixed C, is proportional to the timestep.

MPDATA meets Black-Scholes: some takeaways

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\longrightarrow \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

MPDATA meets Black-Scholes: some takeaways

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\longrightarrow \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

- explicit, non-oscillatory, positive-definite (option pricing!), second-order solution

MPDATA meets Black-Scholes: some takeaways

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\searrow \quad \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

- explicit, non-oscillatory, positive-definite (option pricing!), second-order solution
- consistent (and simple) discretisation for advective and Fickian terms
  $\rightsquigarrow$ for linear payoffs, relevant terms cancel out "before" timestepping

MPDATA meets Black-Scholes: some takeaways

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\longrightarrow \quad \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

- explicit, non-oscillatory, positive-definite (option pricing!), second-order solution
- consistent (and simple) discretisation for advective and Fickian terms
  ⇝ for linear payoffs, relevant terms cancel out "before" timestepping
- classic: B-S ⇝ heat eq. (Lagrangian frame of ref., here: Eulerian)
  ⇝ no time-dependent coordinate transformation

MPDATA meets Black-Scholes: some takeaways

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\searrow \quad \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

- explicit, non-oscillatory, positive-definite (option pricing!), second-order solution
- consistent (and simple) discretisation for advective and Fickian terms
  $\rightsquigarrow$ for linear payoffs, relevant terms cancel out "before" timestepping
- classic: B-S $\rightsquigarrow$ heat eq. (Lagrangian frame of ref., here: Eulerian)
  $\rightsquigarrow$ no time-dependent coordinate transformation
- divergent flow field ($\rightsquigarrow$ relevant MPDATA option required for 2nd-order)

MPDATA meets Black-Scholes: some takeaways

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\longrightarrow \quad \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

- explicit, non-oscillatory, positive-definite (option pricing!), second-order solution
- consistent (and simple) discretisation for advective and Fickian terms
  $\rightsquigarrow$ for linear payoffs, relevant terms cancel out "before" timestepping
- classic: B-S $\rightsquigarrow$ heat eq. (Lagrangian frame of ref., here: Eulerian)
  $\rightsquigarrow$ no time-dependent coordinate transformation
- divergent flow field ($\rightsquigarrow$ relevant MPDATA option required for 2nd-order)
- stability condition: $\mathcal{C} = \left| r - \frac{\sigma^2}{2} + \frac{\sigma^2}{\Delta x} A \right| \frac{\Delta t}{\Delta x} < \frac{1}{2} \quad \rightsquigarrow \quad \lambda^2 = \frac{1}{\sigma^2}\frac{\Delta x^2}{\Delta t} \gtrsim 2$

MPDATA meets Black-Scholes: some takeaways

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 f}{\partial S^2} - rf = 0$$

$$\longrightarrow \quad \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x}\left[\left(u - \frac{\nu}{\psi}\frac{\partial \psi}{\partial x}\right)\psi\right] = 0$$

- explicit, non-oscillatory, positive-definite (option pricing!), second-order solution
- consistent (and simple) discretisation for advective and Fickian terms
  $\rightsquigarrow$ for linear payoffs, relevant terms cancel out "before" timestepping
- classic: B-S $\rightsquigarrow$ heat eq. (Lagrangian frame of ref., here: Eulerian)
  $\rightsquigarrow$ no time-dependent coordinate transformation
- divergent flow field ($\rightsquigarrow$ relevant MPDATA option required for 2nd-order)
- stability condition: $\mathcal{C} = \left| r - \frac{\sigma^2}{2} + \frac{\sigma^2}{\Delta x}A \right| \frac{\Delta t}{\Delta x} < \frac{1}{2} \quad \rightsquigarrow \quad \lambda^2 = \frac{1}{\sigma^2}\frac{\Delta x^2}{\Delta t} \gtrsim 2$

- Black-Scholes put formula = *"standard model for the transport of an unreactive solute in a soil column"* $\rightsquigarrow$ Hogarth et al. (1990, Comp. Math. Applic.)

the last slide

the last slide

- libmpdata++:
  - Jaruga et al. 2015 (Geosci. Model. Dev.)
  - http://github.com/igfuw/libmpdataxx

the last slide

- libmpdata++:
  - Jaruga et al. 2015 (Geosci. Model. Dev.)
  - http://github.com/igfuw/libmpdataxx
- MPDATA meets B-S (arXiv:1607.01751):
  - addressed at math-finance & MPDATA (geo) communities
  - + American option valuation example

the last slide

- libmpdata++:
  - Jaruga et al. 2015 (Geosci. Model. Dev.)
  - http://github.com/igfuw/libmpdataxx
- MPDATA meets B-S (arXiv:1607.01751):
  - addressed at math-finance & MPDATA (geo) communities
  - + American option valuation example
- math-finance prospects:
  - applying the new third-order-accurate MPDATA (Fickian term)
  - application to multi-dimensional problems (M in MPDATA)

the last slide

- libmpdata++:
  - Jaruga et al. 2015 (Geosci. Model. Dev.)
  - `http://github.com/igfuw/libmpdataxx`
- MPDATA meets B-S (arXiv:1607.01751):
  - addressed at math-finance & MPDATA (geo) communities
  - + American option valuation example
- math-finance prospects:
  - applying the new third-order-accurate MPDATA (Fickian term)
  - application to multi-dimensional problems (M in MPDATA)
- acknowledgements:
  - Piotr Smolarkiewicz (ECMWF, Reading)
  - libmpdata++ team @ Faculty of Physics, University of Warsaw
  - Chatham Financial Corporation, Cracow (hack-week projects)
  - AETHON Engineering Consultants, Athens (H2020 funding)

the last slide

- libmpdata++:
  - Jaruga et al. 2015 (Geosci. Model. Dev.)
  - `http://github.com/igfuw/libmpdataxx`
- MPDATA meets B-S (arXiv:1607.01751):
  - addressed at math-finance & MPDATA (geo) communities
  - + American option valuation example
- math-finance prospects:
  - applying the new third-order-accurate MPDATA (Fickian term)
  - application to multi-dimensional problems (M in MPDATA)

- acknowledgements:
  - Piotr Smolarkiewicz (ECMWF, Reading)
  - libmpdata++ team @ Faculty of Physics, University of Warsaw
  - Chatham Financial Corporation, Cracow (hack-week projects)
  - AETHON Engineering Consultants, Athens (H2020 funding)

- thanks to organisers, thank you for and your attention!