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

Sylwester Arabas and Ahmad Farhat
Sylwester Arabas

University of Warsaw
(MSc, physics)

University of Warsaw
(PhD, [geo]physics)

Chatham Financial, Cracow
(models Dev. Team)

AETHON, Athens
(H2020 MoveWise project)

Jagiellonian Univ. Cracow
(from Oct. 2018)

Ahmad Farhat

American University of Beirut
(MSc, mathematics)

University of Wrocław
(PhD, topology)

HSBC, Cracow
(Quant Team)
• MPDATA

• libmpdata++

• derivative pricing as a transport problem
• 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)
transport PDE: \[
\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (v \psi) = 0
\]
transport PDE: \[
\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (\nu \psi) = 0
\]

\[
\psi_i^{n+1} = \psi_i^n - \left[ F(\psi_i^n, \psi_{i+1}^n, C_{i+1/2}) - F(\psi_{i-1}^n, \psi_i^n, C_{i-1/2}) \right]
\]

\[
F(\psi_L, \psi_R, C) = \max(C, 0) \cdot \psi_L + \min(C, 0) \cdot \psi_R
\]

\[
C = \nu \Delta t / \Delta x
\]
MPDATA in a nutshell (Smolarkiewicz 1983, 1984, ...)

Transport PDE:

\[
\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (\nu \psi) = 0
\]

\[
\psi_i^{n+1} = \psi_i^n - \left[ F(\psi_i^n, \psi_{i+1}^n, C_{i+1/2}) - F(\psi_{i-1}^n, \psi_i^n, C_{i-1/2}) \right]
\]

\[
F(\psi_L, \psi_R, C) = \max(C, 0) \cdot \psi_L + \min(C, 0) \cdot \psi_R
\]

\[
C = \nu \Delta t / \Delta x
\]

Modified eq.:

\[
\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (\nu \psi) + K \frac{\partial^2 \psi}{\partial x^2} + \ldots = 0
\]

(means)

numerical diffusion

upwind
MPDATA in a nutshell (Smolarkiewicz 1983, 1984, \ldots)

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

\[
\psi_i^{n+1} = \psi_i^n - \left[ F(\psi_i^n, \psi_{i+1}^n, C_{i+1/2}) - F(\psi_{i-1}^n, \psi_i^n, C_{i-1/2}) \right]
\]

\[ F(\psi_L, \psi_R, C) = \max(C, 0) \cdot \psi_L + \min(C, 0) \cdot \psi_R \]

\[ C = v \Delta t / \Delta x \]

modified eq.: \[ \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (v \psi) + K \frac{\partial^2 \psi}{\partial x^2} + \ldots = 0 \]

\[ \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (v \psi) + \frac{\partial}{\partial x} \left[ \left( -\frac{K}{\psi} \frac{\partial \psi}{\partial x} \right) \psi \right] = 0 \]
MPDATA in a nutshell (Smolarkiewicz 1983, 1984, . . .)

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

\[
\psi_{i}^{n+1} = \psi_{i}^{n} - \left[ F(\psi_{i}^{n}, \psi_{i+1}^{n}, C_{i+1/2}) - F(\psi_{i-1}^{n}, \psi_{i}^{n}, C_{i-1/2}) \right]
\]

\[
F(\psi_{L}, \psi_{R}, C) = \max(C, 0) \cdot \psi_{L} + \min(C, 0) \cdot \psi_{R}
\]

\[
C = v\Delta t/\Delta x
\]

modified eq.: \[
\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (v\psi) + K\frac{\partial^{2}\psi}{\partial x^{2}} + \ldots = 0 \quad \text{MEA}
\]

\[
\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (v\psi) + \frac{\partial}{\partial x} \left[ \left( -\frac{K}{\psi} \frac{\partial \psi}{\partial x} \right) \psi \right] = 0
\]

antidiffusive flux

\[
C'_{i+1/2} = (|C_{i+1/2}| - C^{2}_{i+1/2})A_{i+1/2}
\]

\[
A_{i+1/2} = \frac{\psi_{i+1} - \psi_{i}}{\psi_{i+1} + \psi_{i}}
\]

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

Arabas & Farhat: MPDATA meets Black-Scholes (arXiv:1607.01751)
MPDATA: key features (review: e.g. Smolarkiewicz 2006)

MPDATA

Multidimensional Positive Definite Advection Transport Algorithm
MPDATA

**Multidimensional Positive Definite Advection Transport Algorithm**

- **Multidimensionality:**
  antidiffusive fluxes include cross-dimensional terms, as opposed to dimensionally-split schemes
MPDATA: key features (review: e.g. Smolarkiewicz 2006)

**MPDATA**

Multidimensional Positive Definite Advection Transport Algorithm

- **Multidimensionality:**
  antidiffusive fluxes include cross-dimensional terms, as opposed to dimensionally-split schemes
- **Positive Definiteness:**
  sign-preserving + “infinite-gauge formulation for variable-sign fields
MPDATA

Multidimensional Positive Definite Advection Transport Algorithm

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

- **Positive Definiteness:**
  sign-preserving + “infinite-gauge formulation for variable-sign fields

- **Conservativeness:**
  upstream for all iterations (\(\Rightarrow\) stability cond.)
MPDATA: key features (review: e.g. Smolarkiewicz 2006)

MPDATA

Multidimensional Positive Definite Advection Transport Algorithm

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

- **Positive Definiteness:**
  sign-preserving + “infinite-gauge formulation for variable-sign fields

- **Conservativeness:**
  upstream for all iterations (\(\leadsto\) stability cond.)

- **High-Order Accuracy:**
  up to 3rd-order in time and space (dep. on options & flow)
MPDATA: key features (review: e.g. Smolarkiewicz 2006)

**MPDATA**

Multidimensional Positive Definite Advection Transport Algorithm

- **Multidimensionality:** antidiffusive fluxes include cross-dimensional terms, as opposed to dimensionally-split schemes
- **Positive Definiteness:** sign-preserving + “infinite-gauge formulation for variable-sign fields
- **Conservativeness:**
  upstream for all iterations (\(\swarrow\) stability cond.)
- **High-Order Accuracy:**
  up to 3rd-order in time and space (dep. on options & flow)
- **Monotonicity:**
  with Flux-Corrected Transport option
libmpdata++ 1.0: a library of parallel MPDATA solvers for systems of generalised transport equations

A. Jaruga¹, S. Arabas¹, D. Jarecka¹,², H. Pawlowska¹, P. K. Smolarkiewicz³, and M. Waruszewski¹

¹Institute of Geophysics, Faculty of Physics, University of Warsaw, Warsaw, Poland
²National Center for Atmospheric Research, Boulder, CO, USA
³European Centre for Medium-Range Weather Forecasts, Reading, UK
\[ \partial_t (G \psi) + \nabla \cdot (G \mathbf{u} \psi) = GR \]
\[ \partial_t (G \psi) + \nabla \cdot (G \vec{u} \psi) = GR \]
\[ \partial_t (G\psi) + \nabla \cdot (G\tilde{u}\psi) = GR \]
\partial_t (G \psi) + \nabla \cdot (G \vec{u} \psi) = GR
\[ \partial_t (G\psi) + \nabla \cdot (G\vec{u}\psi) = GR \]
\[ \partial_t (G\psi) + \nabla \cdot (G\vec{u}\psi) = GR \]
\[ \partial_t (G \psi) + \nabla \cdot (G \vec{u} \psi) = GR \]
∂_t(Gψ) + ∇ \cdot (G \vec{u} \psi) = GR
\[ \partial_t (G\psi) + \nabla \cdot (G\vec{u}\psi) = GR \]
\[ \partial_t (G\psi) + \nabla \cdot (G\vec{u}\psi) = GR \]
\[ \partial_t (G \psi) + \nabla \cdot (G \tilde{u} \psi) = GR \]
\[
\partial_t (G \psi) + \nabla \cdot (G \tilde{u} \psi) = GR
\]
\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)
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)
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)
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)
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)
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)
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)
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)
## 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

Arabas & Farhat: MPDATA meets Black-Scholes (arXiv:1607.01751)

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

Arabas, Jaruga et al. 2015 (Geosci. Model. Dev.): 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!
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

\[ dS = S(\mu dt + \sigma dw) \]

\[ f(S, t) \]

riskless portfolio (asset + option):
\[ \Pi = -f + \Delta_t S \]

Itô's lemma: SDE \( \Rightarrow \) PDE

no arbitrage (riskless interest rate):
\[ d\Pi = \Pi r dt \]

terminal value prob., analytic solutions for vanilla options
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

no arbitrage (riskless interest rate): \( d\Pi = \Pi r dt \)
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:
- 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:

\[ \text{SDE} \Rightarrow \text{PDE} \]

- no arbitrage (riskless interest rate):

\[ d\Pi = \Pi r dt \]

\[ \frac{\partial f}{\partial t} + rS \frac{\partial f}{\partial S} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 f}{\partial S^2} - rf = 0 \]
- 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: \[ \text{SDE} \Rightarrow \text{PDE} \]
- no arbitrage (riskless interest rate): \[ d\Pi = \Pi rd\tau \]

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

- 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: $\text{SDE} \rightsquigarrow \text{PDE}$
- no arbitrage (riskless interest rate): $d\Pi = \Pi rdt$
- terminal value prob., analytic solutions for vanilla options

\[ \frac{\partial f}{\partial t} + rS \frac{\partial f}{\partial S} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 f}{\partial S^2} - rf = 0 \]
Black-Scholes $\rightsquigarrow$ ("advection-only") transport problem

\[
\frac{\partial f}{\partial t} + rS \frac{\partial f}{\partial S} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 f}{\partial S^2} - rf = 0
\]
Black-Scholes $\mapsto ("\text{"advection-only"}"")$ transport problem

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

$$x = \ln S \quad \frac{\partial f}{\partial t} + (r - \frac{\sigma^2}{2}) \frac{\partial f}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial x^2} - rf = 0$$
Black-Scholes $\leadsto ("\text{advection-only}")$ transport problem

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

$x = \ln S$

$$\frac{\partial f}{\partial t} + \left( r - \frac{\sigma^2}{2} \right) \frac{\partial f}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial x^2} - rf = 0$$

$$\psi = e^{-rt}$$

$$\frac{\partial \psi}{\partial t} + u \frac{\partial \psi}{\partial x} - \nu \frac{\partial^2 \psi}{\partial x^2} = 0$$
Black-Scholes $\rightsquigarrow$ ("advection-only") transport problem

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

$$x = \ln S \quad \frac{\partial f}{\partial t} + \left( r - \frac{\sigma^2}{2} \right) \frac{\partial f}{\partial x} + \frac{\sigma^2}{2} \left[ -\frac{\partial^2 f}{\partial x^2} - rf \right] = 0$$

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

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} \left[ \left( u - \frac{\nu \partial \psi}{\psi \partial x} \right) \psi \right] = 0$$
Black-Scholes \( \leadsto ("\text{"advection-only"}) \) transport problem

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

\[
x = \ln S \quad \frac{\partial f}{\partial t} + (r - \frac{\sigma^2}{2}) \frac{\partial f}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 f}{\partial x^2} - rf = 0
\]

\[
\psi = e^{-rf} \quad \frac{\partial \psi}{\partial t} + u \frac{\partial \psi}{\partial x} - \nu \frac{\partial^2 \psi}{\partial x^2} = 0
\]

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

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:**
\[
\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (\nu \psi) = 0
\]

\[
\psi_{i}^{n+1} = \psi_{i}^{n} - \left[ F(\psi_{i}^{n}, \psi_{i+1}^{n}, C_{i+1/2}) - F(\psi_{i-1}^{n}, \psi_{i}^{n}, C_{i-1/2}) \right]
\]

\[
F(\psi_{L}, \psi_{R}, C) = \max(C, 0) \cdot \psi_{L} + \min(C, 0) \cdot \psi_{R}
\]

\[
C = \nu \Delta t / \Delta x
\]

modified eq.:
\[
\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} (\nu \psi) + \underbrace{K \frac{\partial^2 \psi}{\partial x^2}}_{\text{numerical diffusion}} + \ldots = 0
\]

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

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

**Black-Scholes \text{\textasciitilde}\text{\textasciitilde} (“advection-only”) transport problem**

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

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

\[
\psi = e^{-\nu t} \frac{\partial \psi}{\partial t} + u \frac{\partial \psi}{\partial x} - \nu \frac{\partial^2 \psi}{\partial x^2} = 0
\]
payoff function: corridor

truncation error est. ($\psi_a$: B-S formula):

$$E = \sqrt{\sum_{i=1}^{n_x} [\psi_n(x_i) - \psi_a(x_i)]^2 / (n_x \cdot n_t)_{t=0}}$$
Truncation error as a function of the Courant number $C = u \frac{\Delta t}{\Delta x}$ which, for fixed $\lambda^2$, is proportional to the gridstep.

Truncation error as a function of the $\lambda^2$ parameter which, for fixed $C$, is proportional to the timestep.
MPDATA meets Black-Scholes: some takeaways

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

\[ \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} \left[ \left( u - \frac{\nu}{\psi} \frac{\partial \psi}{\partial x} \right) \psi \right] = 0 \]
MPDATA meets Black-Scholes: some takeaways

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

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

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

consistent (and simple) discretisation for advective and Fickian terms

\( \Rightarrow \) for linear payoffs, relevant terms cancel out “before” timestepping
MPDATA meets Black-Scholes: some takeaways

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

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

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

\[ C = \left| \frac{r - \sigma^2}{2} + \frac{\sigma^2}{2} \Delta x \right| \Delta t \frac{\Delta x}{A} < \frac{1}{2} \]

\[ \lambda^2 = \frac{\sigma^2}{2} \Delta x^2 \Delta t \gg 2 \]

Black-Scholes put formula = “standard model for the transport of an unreactive solute in a soil column”

\( \Rightarrow \) Hogarth et al. (1990, Comp. Math. Applic.)
explicit, non-oscillatory, positive-definite (option pricing!), second-order solution

consistent (and simple) discretisation for advective and Fickian terms

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

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

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

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

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

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

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

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

- 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: \( C = \left| r - \frac{\sigma^2}{2} + \frac{\sigma^2}{\Delta x} A \right| \frac{\Delta t}{\Delta x} < \frac{1}{2} \) \( \rightsquigarrow \) \( \chi^2 = \frac{1}{\sigma^2} \frac{\Delta x^2}{\Delta t} \geq 2 \)

- 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.)
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 your attention!
libmpdata++:

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

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 your attention!
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
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)
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)
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!