On the design and cloud-modelling applications of libmpdata++

Anna Jaruga, Sylwester Arabas, Maciej Waruszewski

and the cloud-aerosol modelling team
Institute of Geophysics, Faculty of Physics, University of Warsaw

Workshop on Eulerian vs. Lagrangian methods for cloud microphysics
20th-22nd of April 2015

cloud-aerosol modelling team @ University of Warsaw

the team (http://foss.igf.fuw.edu.pl/)

- prof. Hanna Pawłowska (group leader)

cloud-aerosol modelling team @ University of Warsaw

the team (http://foss.igf.fuw.edu.pl/)

- prof. Hanna Pawłowska (group leader)
- in Warsaw:
  - Sylwester Arabas (postdoc)
  - Piotr Dziekan (postdoc)
  - Anna Jaruga (PhD student)
  - Maciej Waruszewski (PhD student)
  - Anna Zimniak (MSc student)

cloud-aerosol modelling team @ University of Warsaw

the team (http://foss.igf.fuw.edu.pl/)

- prof. Hanna Pawłowska (group leader)
- in Warsaw:
  - Sylwester Arabas (postdoc)
  - Piotr Dziekan (postdoc)
  - Anna Jaruga (PhD student)
  - Maciej Waruszewski (PhD student)
  - Anna Zimniak (MSc student)
- overseas:
  - prof. Piotr Smolarkiewicz @ ECMWF
  - prof. Wojciech Grabowski @ NCAR
  - Dorota Jarecka @ NCAR (postdoc)

cloud-aerosol modelling team @ University of Warsaw

the team (http://foss.igf.fuw.edu.pl/)

- prof. Hanna Pawłowska (group leader)
- in Warsaw:
  - Sylwester Arabas (postdoc)
  - Piotr Dziekan (postdoc)
  - Anna Jaruga (PhD student)
  - Maciej Waruszewski (PhD student)
  - Anna Zimniak (MSc student)
- overseas:
  - prof. Piotr Smolarkiewicz @ ECMWF
  - prof. Wojciech Grabowski @ NCAR
  - Dorota Jarecka @ NCAR (postdoc)

aim: developing tools for studying aerosol-cloud interactions

cloud-aerosol modelling team @ University of Warsaw

the team (http://foss.igf.fuw.edu.pl/)

- prof. Hanna Pawłowska (group leader)
- in Warsaw:
  - Sylwester Arabas (postdoc)
  - Piotr Dziekan (postdoc)
  - Anna Jaruga (PhD student)
  - Maciej Waruszewski (PhD student)
  - Anna Zimniak (MSc student)
- overseas:
  - prof. Piotr Smolarkiewicz @ ECMWF
  - prof. Wojciech Grabowski @ NCAR
  - Dorota Jarecka @ NCAR (postdoc)

aim: developing tools for studying aerosol-cloud interactions

- novel cloud/aerosol microphysics models,

cloud-aerosol modelling team @ University of Warsaw

the team (http://foss.igf.fuw.edu.pl/)

- prof. Hanna Pawłowska (group leader)
- in Warsaw:
  - Sylwester Arabas (postdoc)
  - Piotr Dziekan (postdoc)
  - Anna Jaruga (PhD student)
  - Maciej Waruszewski (PhD student)
  - Anna Zimniak (MSc student)
- overseas:
  - prof. Piotr Smolarkiewicz @ ECMWF
  - prof. Wojciech Grabowski @ NCAR
  - Dorota Jarecka @ NCAR (postdoc)

aim: developing tools for studying aerosol-cloud interactions

- novel cloud/aerosol microphysics models,
- state-of-the-art numerical schemes,

cloud-aerosol modelling team @ University of Warsaw

the team (`http://foss.igf.fuw.edu.pl/`)

- prof. Hanna Pawłowska (group leader)
- in Warsaw:
  - Sylwester Arabas (postdoc)
  - Piotr Dziekan (postdoc)
  - Anna Jaruga (PhD student)
  - Maciej Waruszewski (PhD student)
  - Anna Zimniak (MSc student)
- overseas:
  - prof. Piotr Smolarkiewicz @ ECMWF
  - prof. Wojciech Grabowski @ NCAR
  - Dorota Jarecka @ NCAR (postdoc)

aim: developing tools for studying aerosol-cloud interactions

- novel cloud/aerosol microphysics models,
- state-of-the-art numerical schemes,
- modern coding techniques
  $\leadsto$ priorities: researchers' productivity and result reproducibility

outline

On the design

and cloud-modelling applications of

libmpdata++

outline

On the design

and cloud-modelling applications of

lib**mpdata**++

MPDATA

Multi-dimensional Positive-Definite Advection Transport Algorithm

MPDATA

Multi-dimensional Positive-Definite Advection Transport Algorithm

- introduced in 1983 by Smolarkiewicz

MPDATA

Multi-dimensional Positive-Definite Advection Transport Algorithm

- introduced in 1983 by Smolarkiewicz
- at least second-order-accurate

MPDATA

Multi-dimensional Positive-Definite Advection Transport Algorithm

- introduced in 1983 by Smolarkiewicz
- at least second-order-accurate
- sign-preserving

MPDATA

Multi-dimensional Positive-Definite Advection Transport Algorithm

- introduced in 1983 by Smolarkiewicz
- at least second-order-accurate
- sign-preserving
- iterative scheme in which all iterations take the form of a first-order-accurate donor-cell pass

MPDATA

Multi-dimensional Positive-Definite Advection Transport Algorithm

- introduced in 1983 by Smolarkiewicz
- at least second-order-accurate
- sign-preserving
- iterative scheme in which all iterations take the form of a first-order-accurate donor-cell pass
- among the MPDATA options there are:
  - non-oscillatory option ensuring ripple-free solutions
  - accounting for third-order terms leading to more accurate solutions
  - option for transporting variable sign fields

MPDATA

Multi-dimensional Positive-Definite Advection Transport Algorithm

- introduced in 1983 by Smolarkiewicz
- at least second-order-accurate
- sign-preserving
- iterative scheme in which all iterations take the form of a first-order-accurate donor-cell pass
- among the MPDATA options there are:
  - non-oscillatory option ensuring ripple-free solutions
  - accounting for third-order terms leading to more accurate solutions
  - option for transporting variable sign fields

In our project it will serve as a dynamical core for our LES model

outline

On the design

and cloud-modelling applications of

lib**mpdata**++

outline

On the design

and cloud-modelling applications of

libmpdata++

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA-based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code ($\lesssim$ 10 kLOC)

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code (< 10 kLOC)

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code (< 10 kLOC)

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code (< 10 kLOC)

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code (< 10 kLOC)

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code ($<$ 10 kLOC)

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code ($<$ 10 kLOC)

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code ($<$ 10 kLOC)

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code (< 10 kLOC)

libmpdata++

free & open source C++ library of parallel MPDATA solvers

library

- leverage existing reusable software / let others reuse our code
- set of reusable components (solvers, output, concurrency)
- well-defined interface (documented in the paper)

key features

- variety of MPDATA based solvers in 1D, 2D & 3D
- rich set of options (MPDATA, boundary conditions, ...)
- shared-memory parallelisation using OpenMP or Boost.Thread
- built-in HDF5/XDMF output
- implemented using Blitz++ (no loops, expression templates)
- compact C++11 code ($<$ 10 kLOC)

outline

On the design

and cloud-modelling applications of

libmpdata++

outline

On the design

and cloud-modelling applications of

libmpdata++

software design and researchers’ productivity

software design and researchers’ productivity

- ease of use
- robustness
- result reproducibility

software design and researchers' productivity

- ease of use
- robustness
- result reproducibility

users' perspective

software design and researchers’ productivity

- ease of use
- robustness
- result reproducibility

users’ perspective

- extendability
- maintainability

software design and researchers' productivity

- ease of use
- robustness
- result reproducibility

users' perspective

- extendability
- maintainability

developers' perspective

software design and researchers’ productivity

- ease of use
- robustness
- result reproducibility

users’ perspective

- extendability
- maintainability

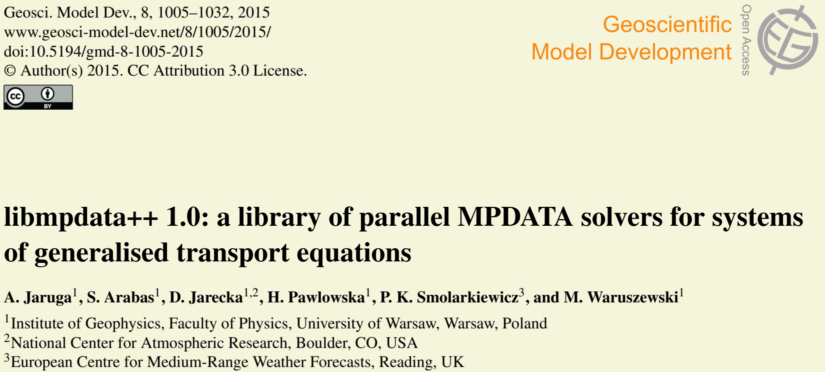
developers’ perspective

researcher = user & developer

software design and researchers’ productivity

- documentation

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

Geosci. Model Dev. policy (doi: 10.5194/gmd-6-1233-2013)

- *"paper must be accompanied by the code, or means of accessing the code, for the purpose of peer-review"*
- *"we strongly encourage referees to compile the code, and run test cases supplied by the authors"*

software design and researchers’ productivity

- documentation

software design and researchers' productivity

- documentation
- separation of user and library code (ease of updates)

software design and researchers' productivity

- documentation

- separation of user and library code (ease of updates)
- use of third-party libraries (Blitz++, Boost, OpenMP, HDF5)

building blocks of libmpdata++

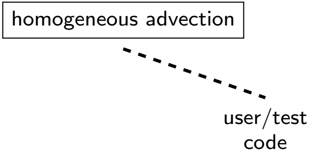
building blocks of libmpdata++

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

building blocks of libmpdata++

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$



homogeneous advection

user/test code

building blocks of libmpdata++

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

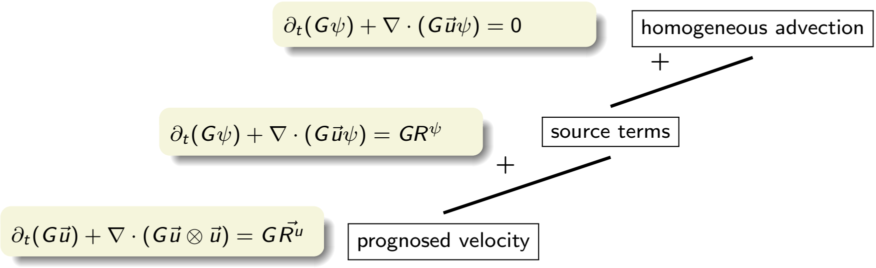
homogeneous advection

+

source terms

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

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

+

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

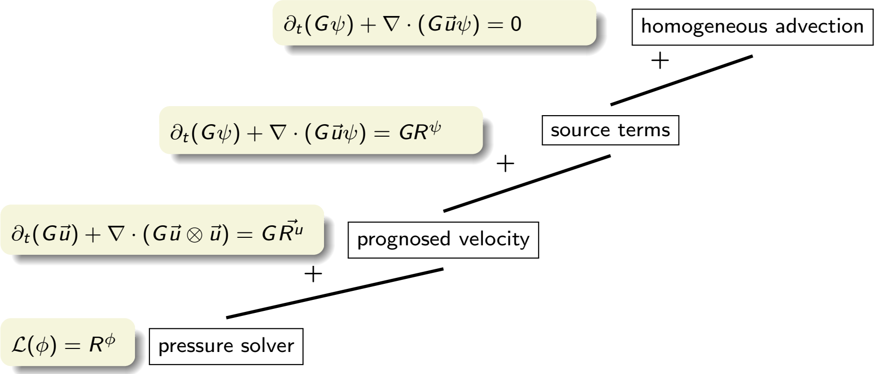
source terms

+

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R}^{u}$$

prognosed velocity

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

+

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

source terms

+

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^{u}}$$

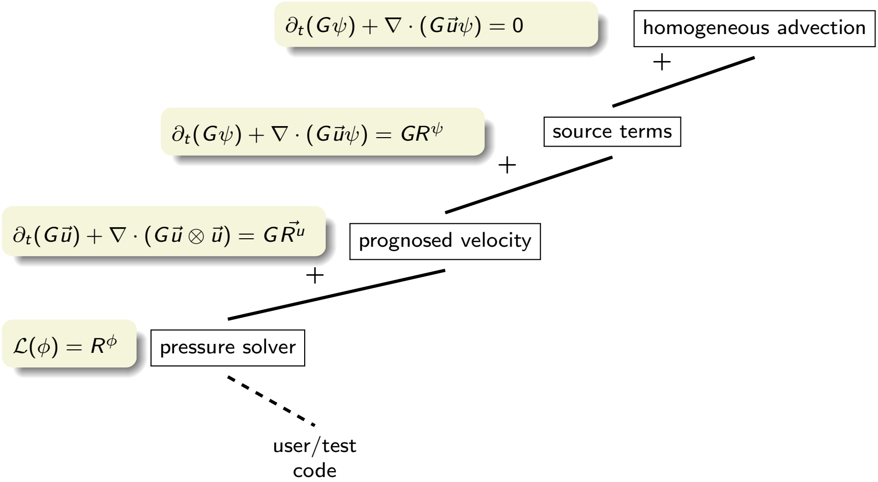
prognosed velocity

+

$$\mathcal{L}(\phi) = R^{\phi}$$

pressure solver

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla\cdot(G\vec{u}\psi) = 0$$

homogeneous advection

+

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

source terms

+

$$\partial_t(G\vec{u}) + \nabla\cdot(G\vec{u}\otimes\vec{u}) = G\vec{R^u}$$

prognosed velocity

+

$$\mathcal{L}(\phi) = R^{\phi}$$

pressure solver

user/test
code

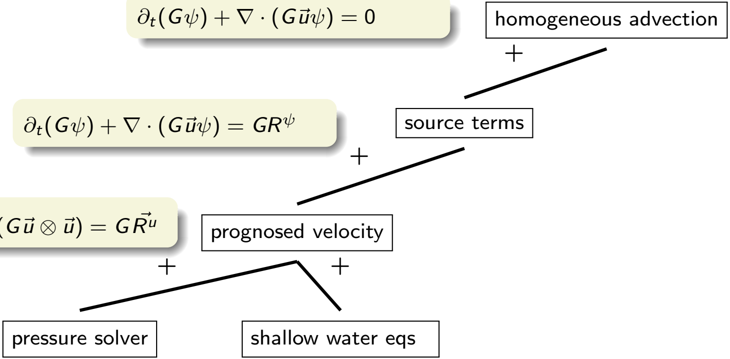
building blocks of libmpdata++

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

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

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^{u}}$$

$$\mathcal{L}(\phi) = R^{\phi}$$



homogeneous advection

+

source terms

+

prognosed velocity

+

+

pressure solver

shallow water eqs

software design and researchers' productivity

- documentation

- separation of user and library code (ease of updates)
- use of third-party libraries (Blitz++, Boost, OpenMP, HDF5)

software design and researchers' productivity

- documentation

- separation of user and library code (ease of updates)
- use of third-party libraries (Blitz++, Boost, OpenMP, HDF5)

- public code repository with history of changes

software design and researchers' productivity

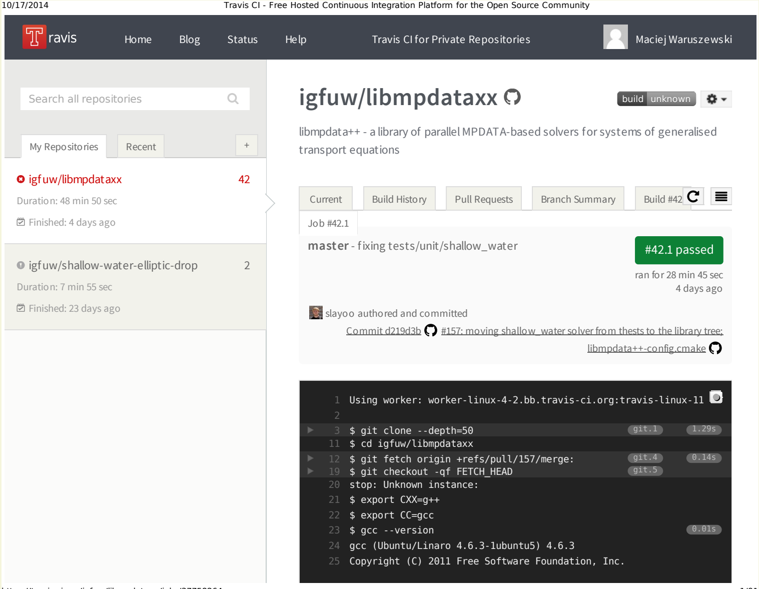
- documentation

- separation of user and library code (ease of updates)
- use of third-party libraries (Blitz++, Boost, OpenMP, HDF5)

- public code repository with history of changes
- automated builds and testing after changes

tests

Travis CI



10/17/2014 Travis CI - Free Hosted Continuous Integration Platform for the Open Source Community

igfuw/libmpdataxx 42
Duration: 48 min 50 sec
Finished: 4 days ago

igfuw/shallow-water-elliptic-drop 2
Duration: 7 min 55 sec
Finished: 23 days ago

igfuw/libmpdataxx

libmpdata++ - a library of parallel MPDATA-based solvers for systems of generalised transport equations

master - fixing tests/unit/shallow_water

#42.1 passed

ran for 28 min 45 sec
4 days ago

slayoo authored and committed

Commit d219d3b #157: moving shallow_water solver from thests to the library tree; libmpdata++-config.cmake

```
Using worker: worker-linux-4-2.bb.travis-ci.org:travis-linux-11

$ git clone --depth=50                                    git.1   1.29s
$ cd igfuw/libmpdataxx
$ git fetch origin +refs/pull/157/merge:                  git.4   0.14s
$ git checkout -qf FETCH_HEAD                             git.5
stop: Unknown instance:
$ export CXX=g++
$ export CC=gcc
$ gcc --version                                                   0.01s
gcc (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3
Copyright (C) 2011 Free Software Foundation, Inc.
```

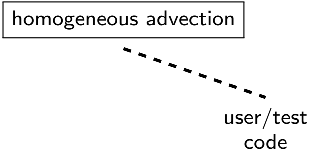
building blocks of libmpdata++

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

building blocks of libmpdata++

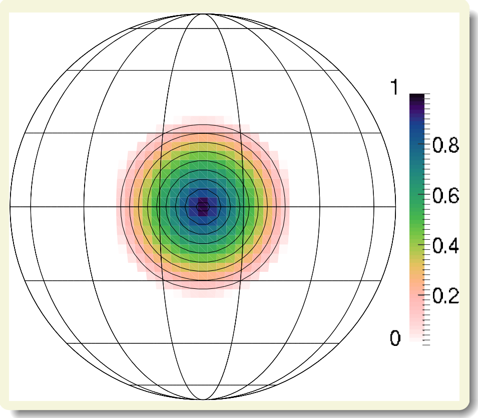
$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$



homogeneous advection

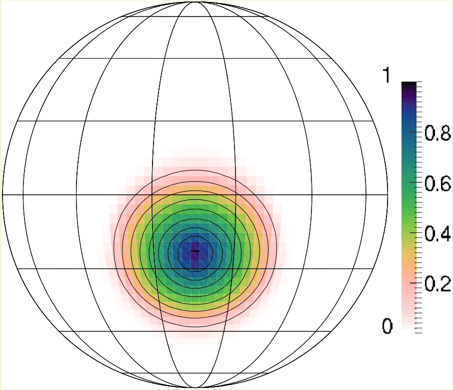
user/test code

example: 2D advection on a sphere



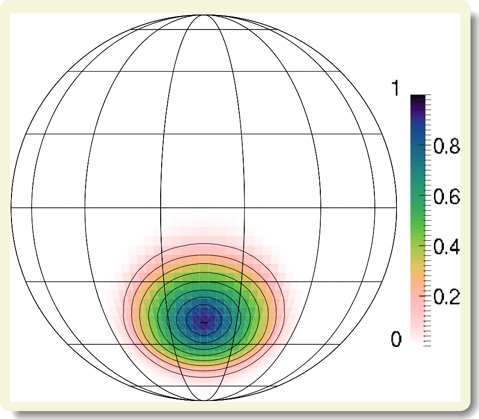
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



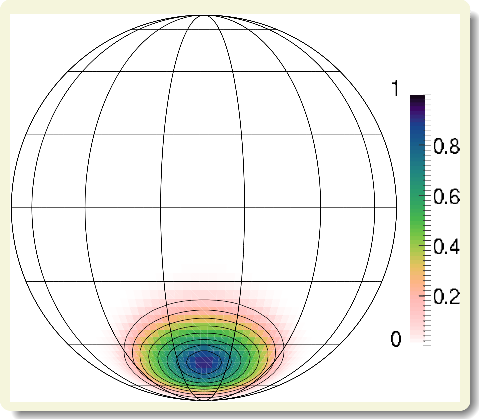
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



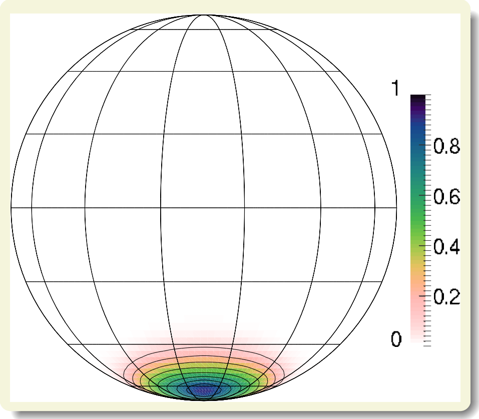
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



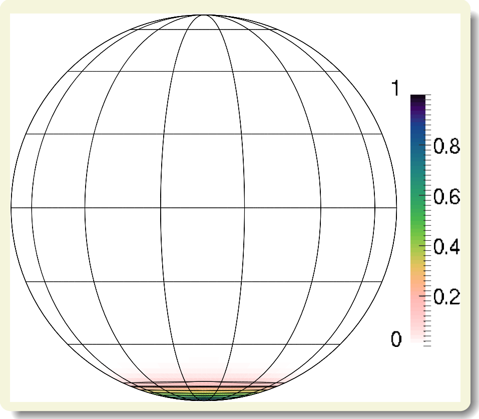
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



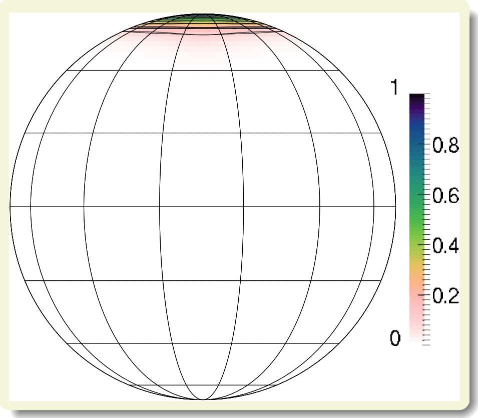
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



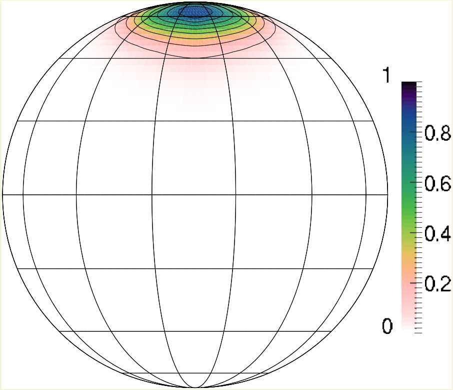
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



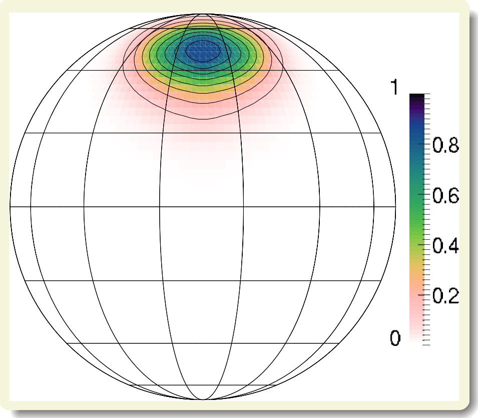
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



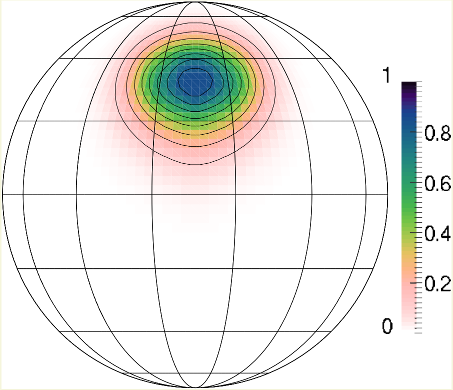
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



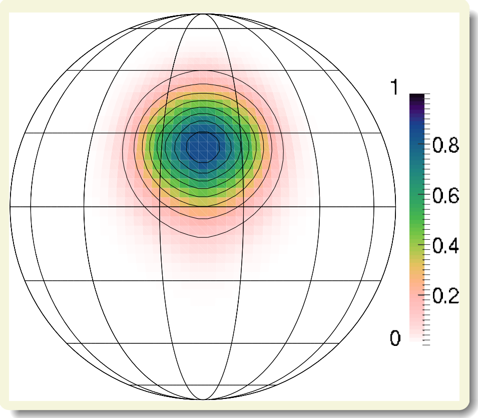
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



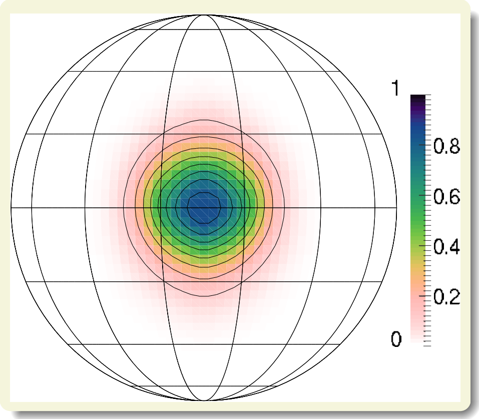
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



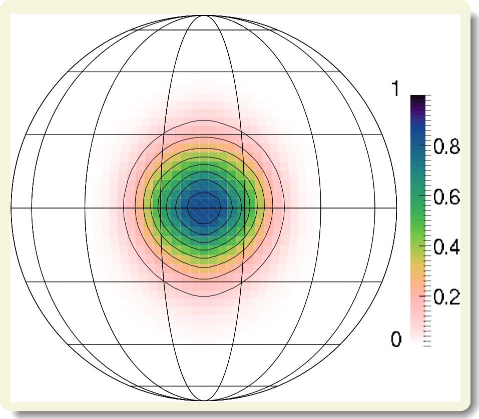
- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



- reproduced experiment of Williamson and Rasch, 1989

example: 2D advection on a sphere



- reproduced experiment of Williamson and Rasch, 1989
- <100 lines of code with libmpdata++

building blocks of libmpdata++

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection
user/test code

building blocks of libmpdata++

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

+

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

source terms

building blocks of libmpdata++

$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

+

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

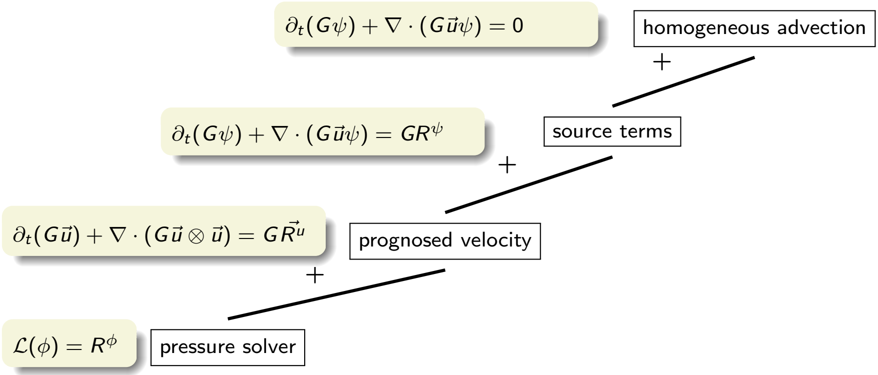
source terms

+

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^{u}}$$

prognosed velocity

building blocks of libmpdata++



$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$

homogeneous advection

+

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

source terms

+

$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^{u}}$

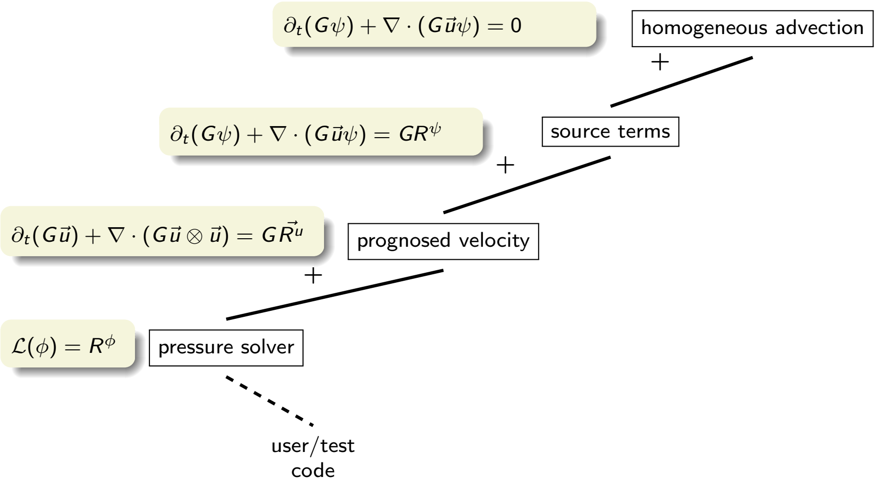
prognosed velocity

+

$\mathcal{L}(\phi) = R^{\phi}$

pressure solver

building blocks of libmpdata++



$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$

homogeneous advection

+

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

source terms

+

$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^u}$

prognosed velocity

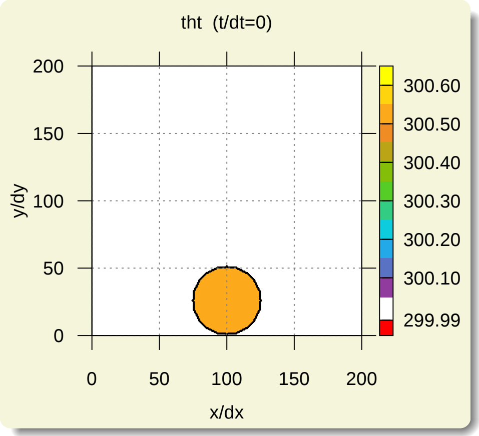
+

$\mathcal{L}(\phi) = R^{\phi}$

pressure solver

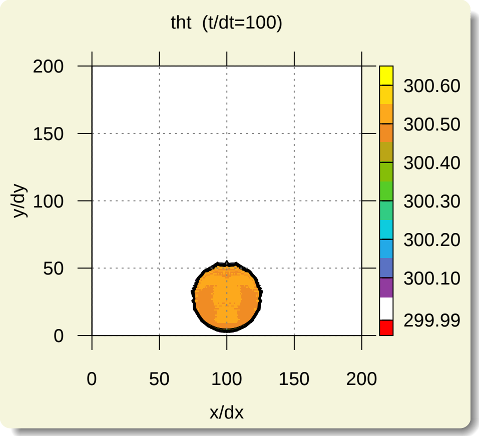
user/test
code

example: 2D Boussinesq convection



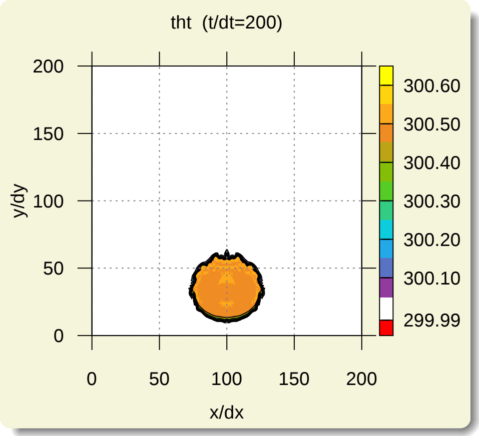
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



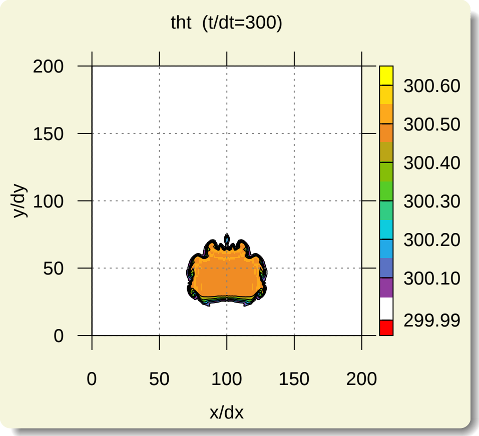
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



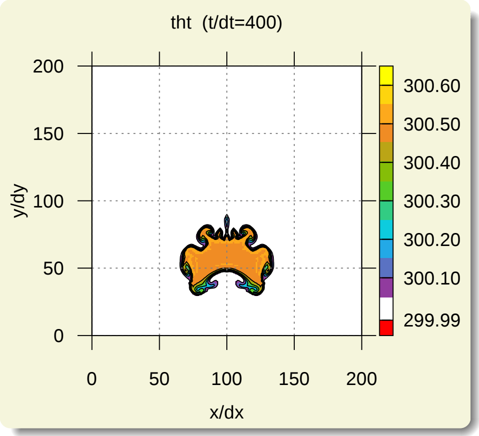
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



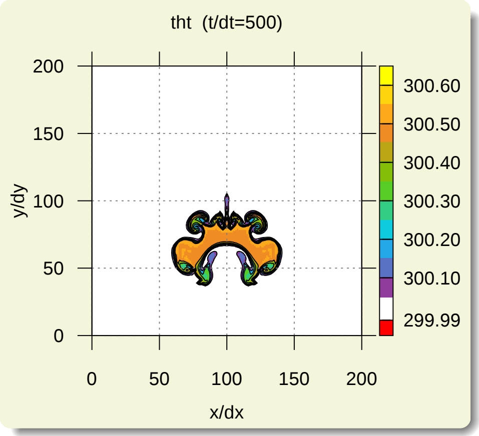
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



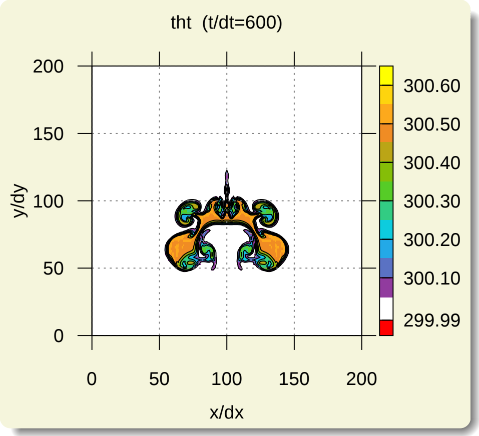
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



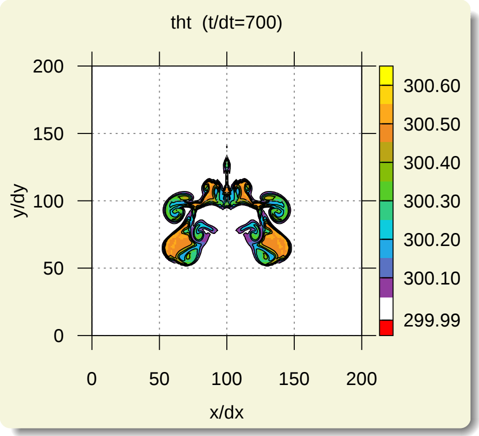
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



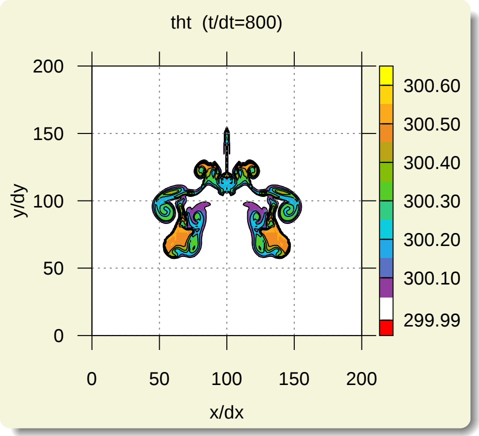
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



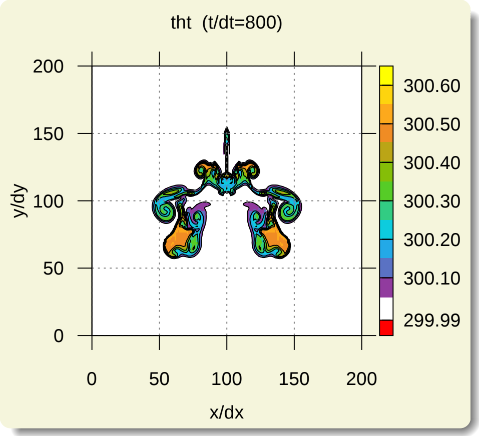
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



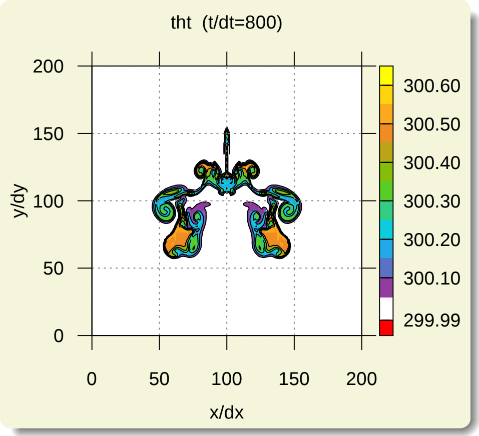
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



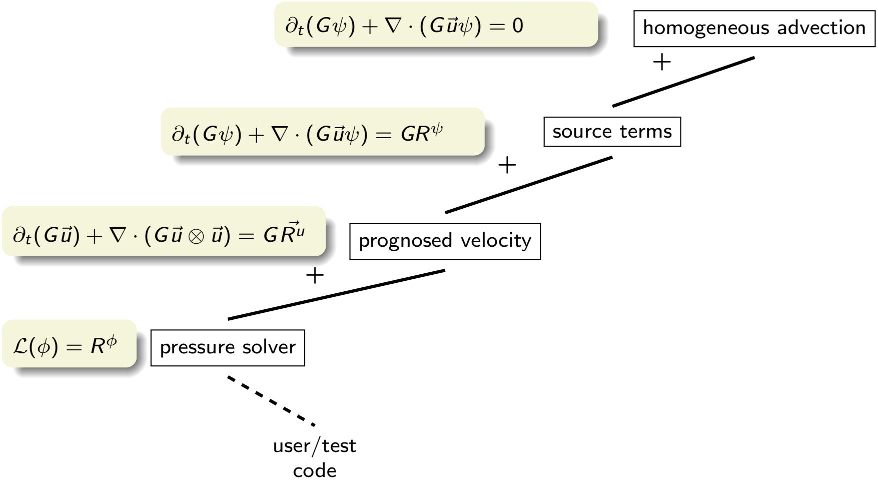
- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992

example: 2D Boussinesq convection



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++
  (using built-in elliptic pressure solver)

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

+

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

source terms

+

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^u}$$

prognosed velocity

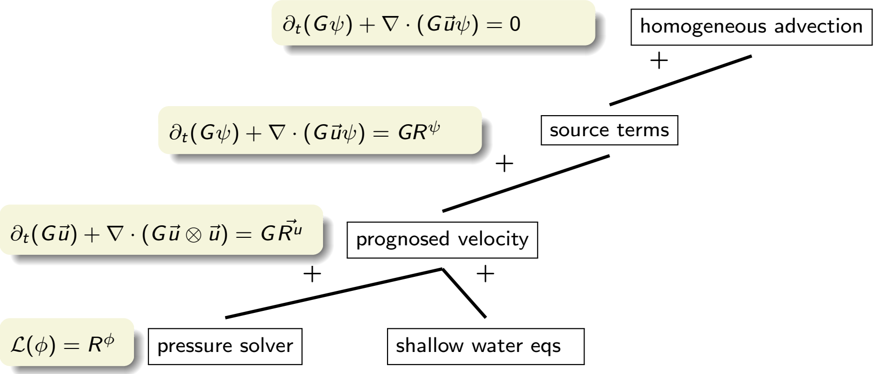
+

$$\mathcal{L}(\phi) = R^{\phi}$$

pressure solver

user/test
code

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

+

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

source terms

+

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^{u}}$$

prognosed velocity

+

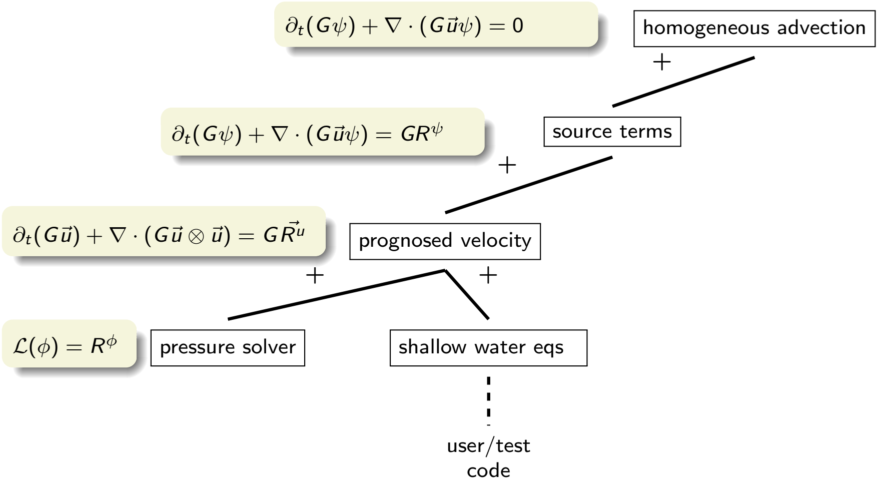
+

$$\mathcal{L}(\phi) = R^{\phi}$$

pressure solver

shallow water eqs

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla\cdot(G\vec{u}\psi) = 0$$

homogeneous advection

+

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

source terms

+

$$\partial_t(G\vec{u}) + \nabla\cdot(G\vec{u}\otimes\vec{u}) = G\vec{R^u}$$

prognosed velocity

+ +

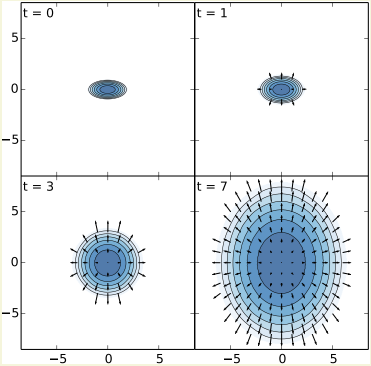
$$\mathcal{L}(\phi) = R^{\phi}$$

pressure solver

shallow water eqs

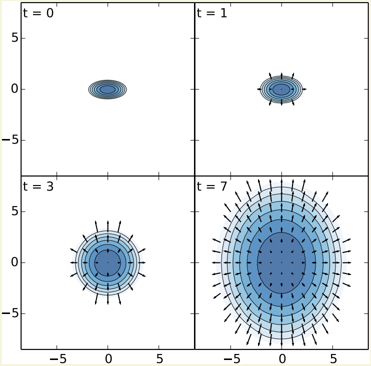
user/test code

example: spreading drop of shallow water in 3D



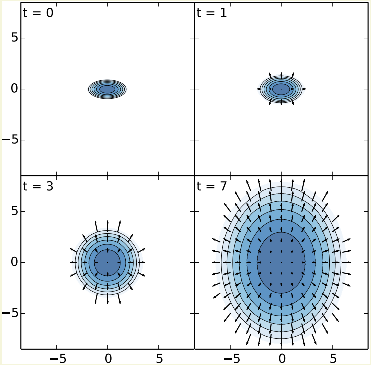
- inspired by 2D experiment of Schär and Smolarkiewicz, 1996

example: spreading drop of shallow water in 3D



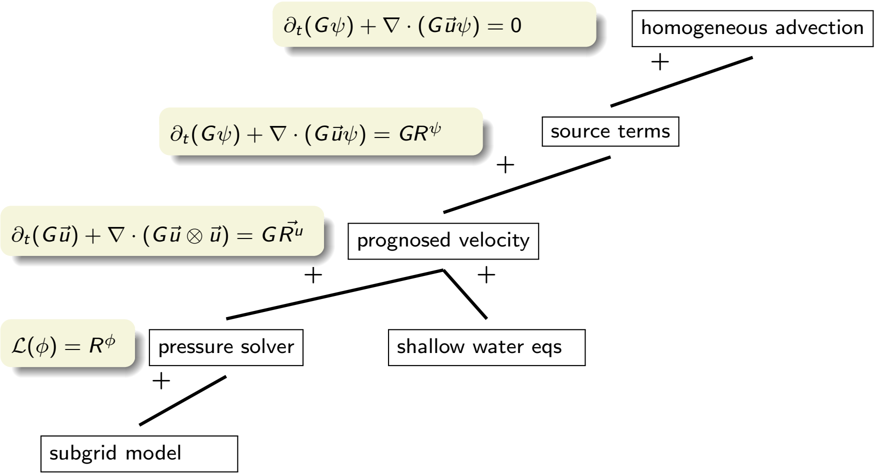
- inspired by 2D experiment of Schär and Smolarkiewicz, 1996
- example and original analytic solution by Dorota Jarecka / NCAR (paper Jarecka D., Jaruga A., Smolarkiewicz P.K. JCP 2015)

example: spreading drop of shallow water in 3D



- inspired by 2D experiment of Schär and Smolarkiewicz, 1996
- example and original analytic solution by Dorota Jarecka / NCAR (paper Jarecka D., Jaruga A., Smolarkiewicz P.K. JCP 2015)
- <120 lines of code with libmpdata++

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

+

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

source terms

+

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^u}$$

prognosed velocity

+

+

$$\mathcal{L}(\phi) = R^{\phi}$$

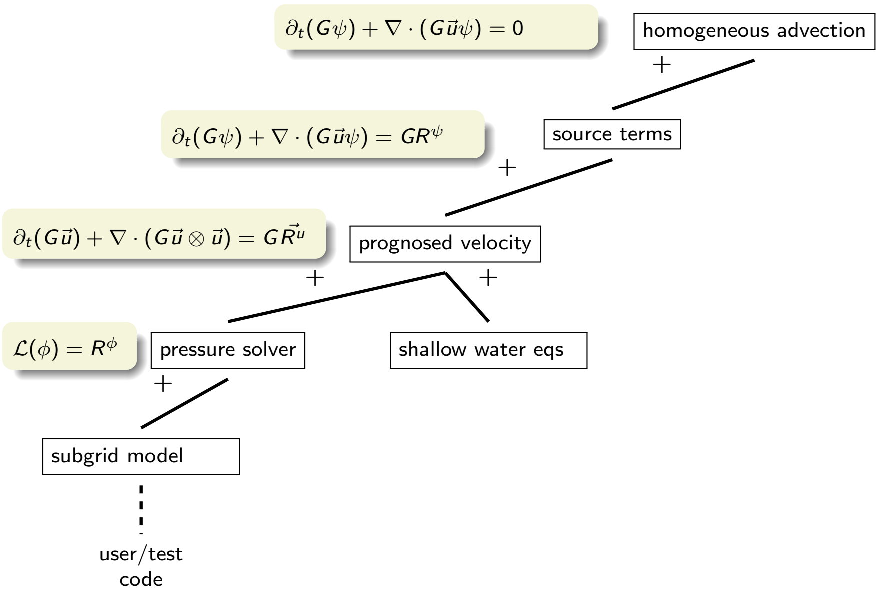
pressure solver

shallow water eqs

+

subgrid model

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

+

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

source terms

+

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^{u}}$$

prognosed velocity

+

+

$$\mathcal{L}(\phi) = R^{\phi}$$

pressure solver

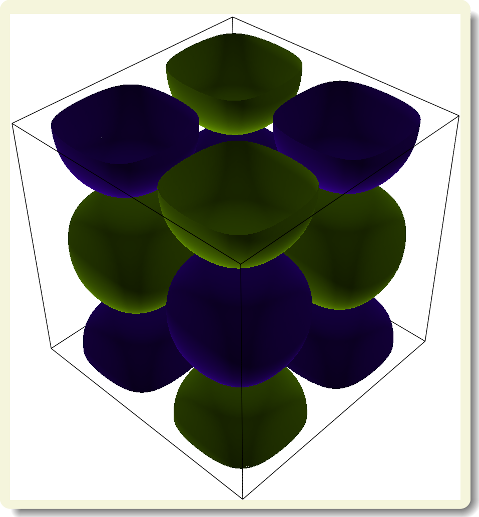
shallow water eqs

+

subgrid model

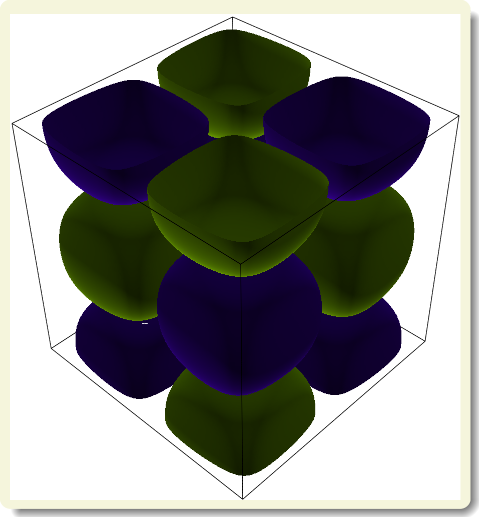
user/test code

example: Taylor-Green vortex at Re = 800



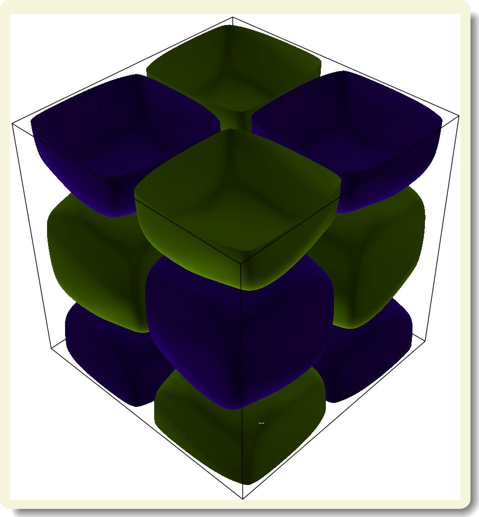
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



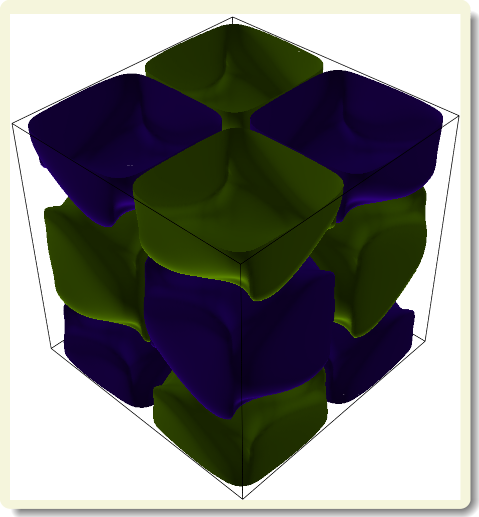
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



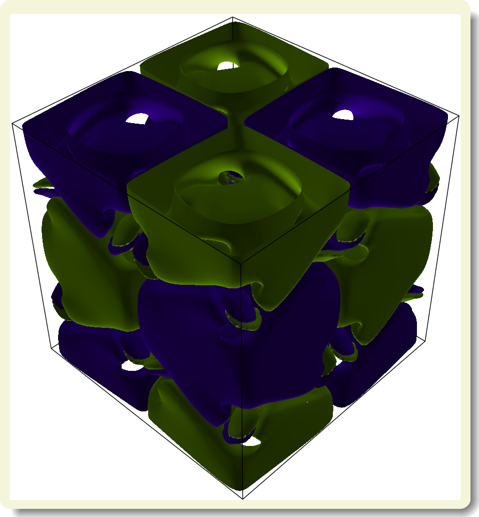
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



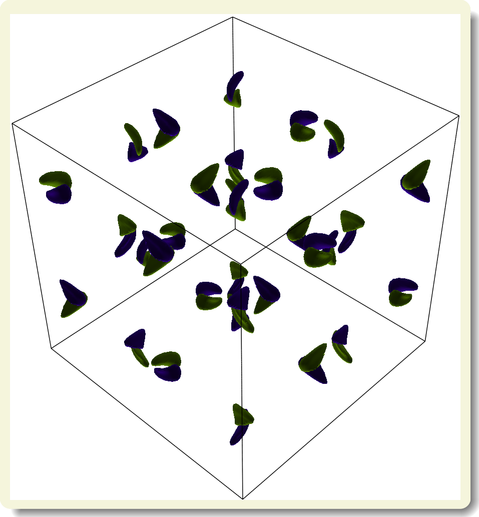
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



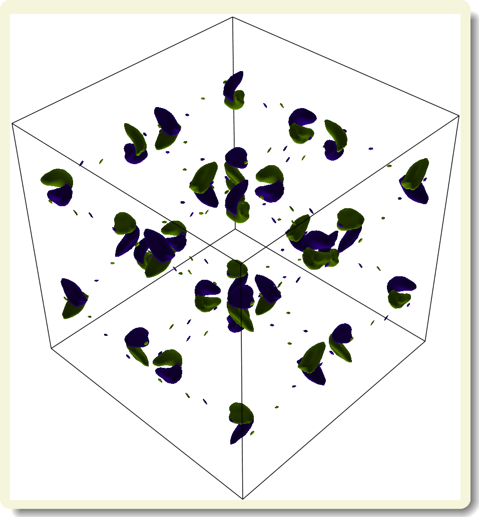
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



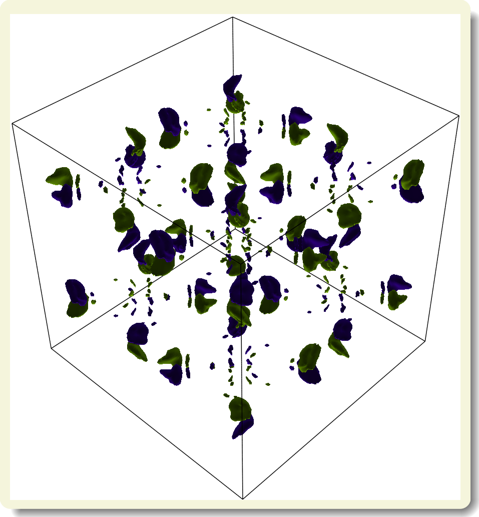
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



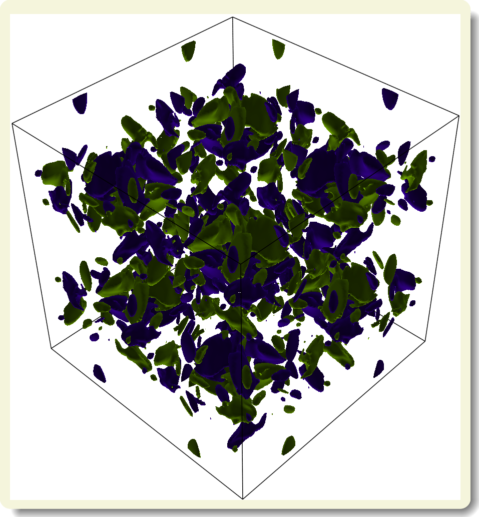
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



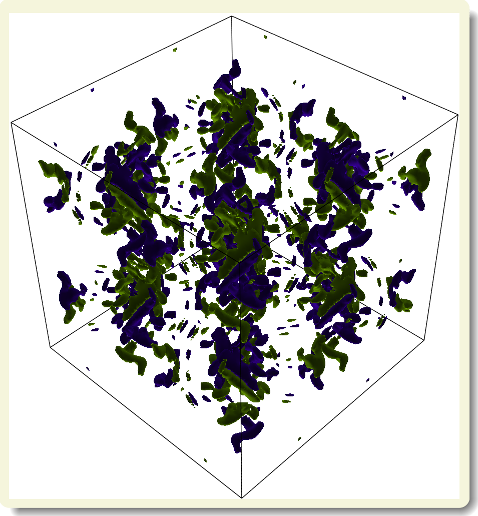
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



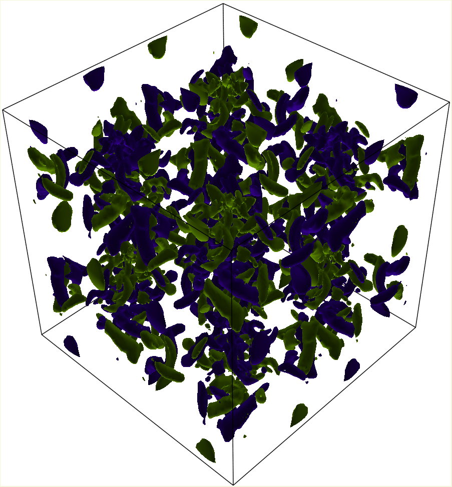
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



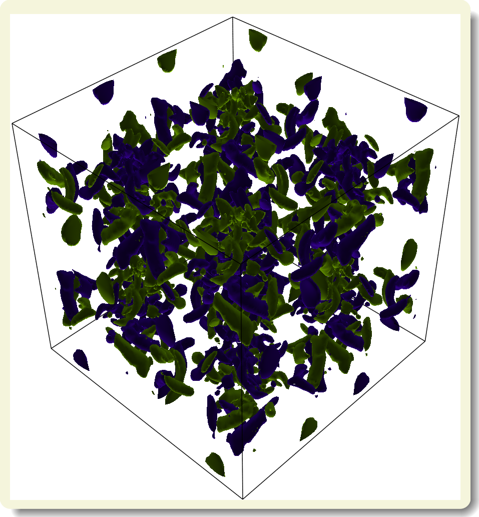
- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



- inspired by spectral calculations of Brachet et al., 1983

example: Taylor-Green vortex at Re = 800



- inspired by spectral calculations of Brachet et al., 1983
- <100 lines of code with libmpdata++

outline

On the design

and cloud-modelling applications of

libmpdata++

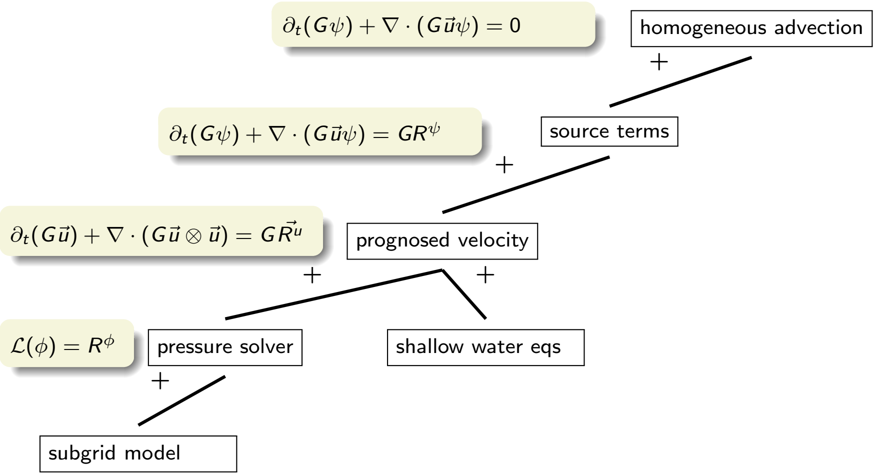
outline

On the design

and cloud-modelling applications of

libmpdata++

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

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

source terms

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^{u}}$$

prognosed velocity

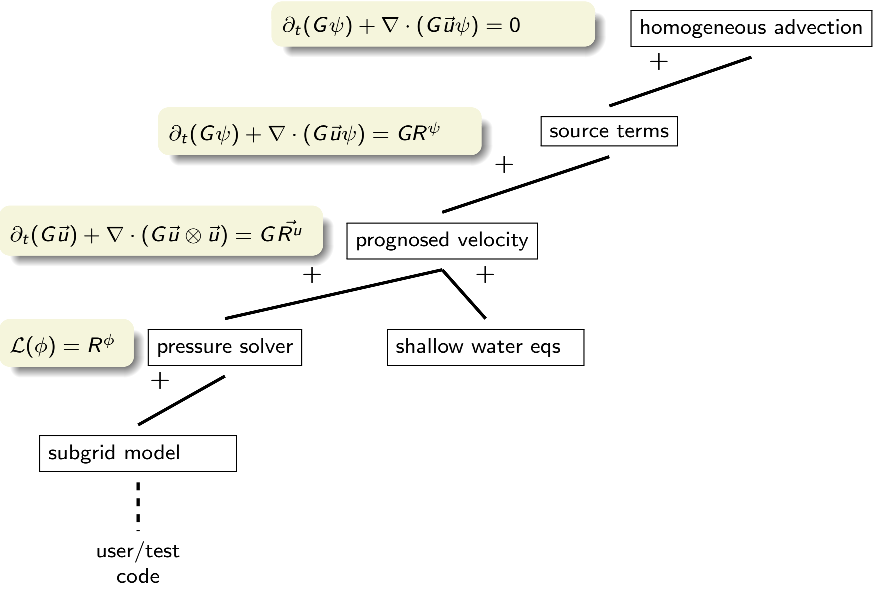
$$\mathcal{L}(\phi) = R^{\phi}$$

pressure solver

shallow water eqs

subgrid model

building blocks of libmpdata++



$$\partial_t(G\psi) + \nabla \cdot (G\vec{u}\psi) = 0$$

homogeneous advection

+

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

source terms

+

$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R^u}$$

prognosed velocity

+

+

$$\mathcal{L}(\phi) = R^{\phi}$$

pressure solver

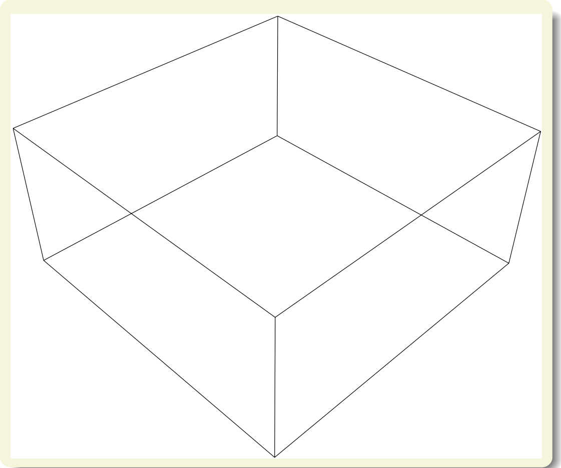
shallow water eqs

+

subgrid model

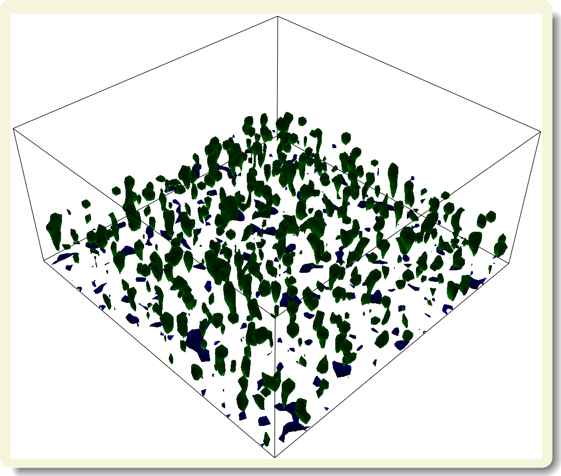
user/test
code

example: convective PBL



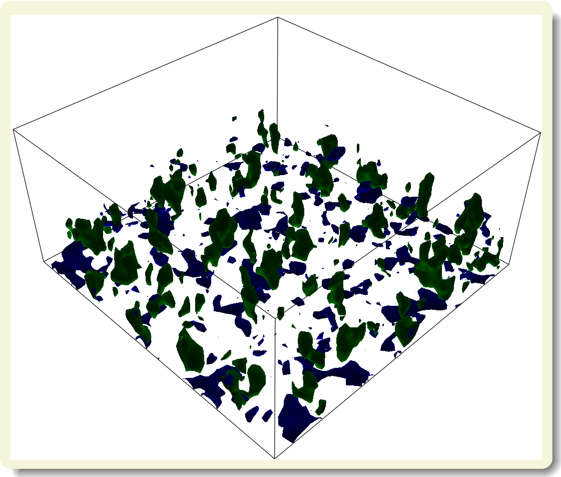
- setup following Margolin et al., 1999

example: convective PBL



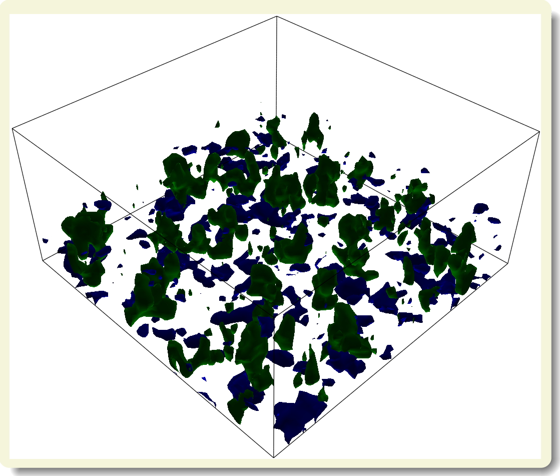
- setup following Margolin et al., 1999

example: convective PBL



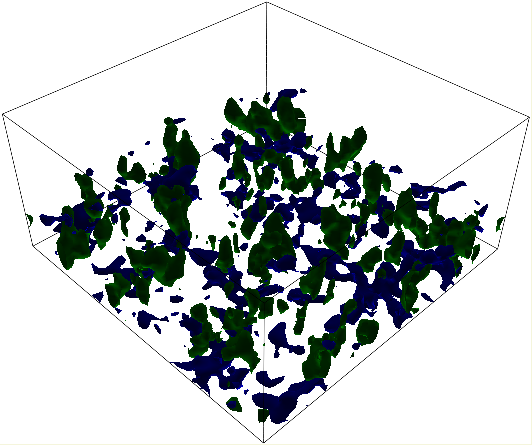
- setup following Margolin et al., 1999

example: convective PBL



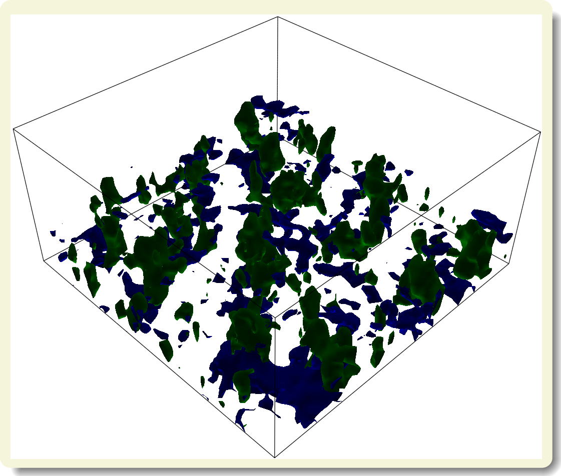
- setup following Margolin et al., 1999

example: convective PBL



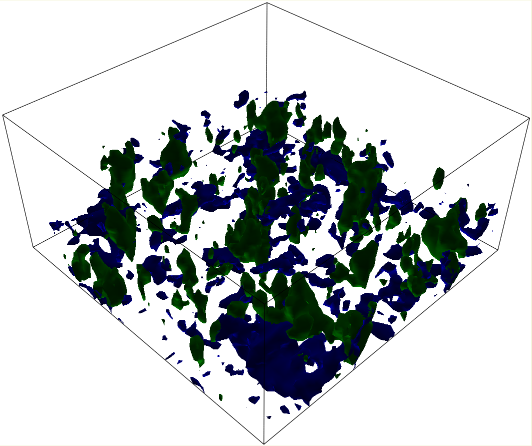
- setup following Margolin et al., 1999

example: convective PBL



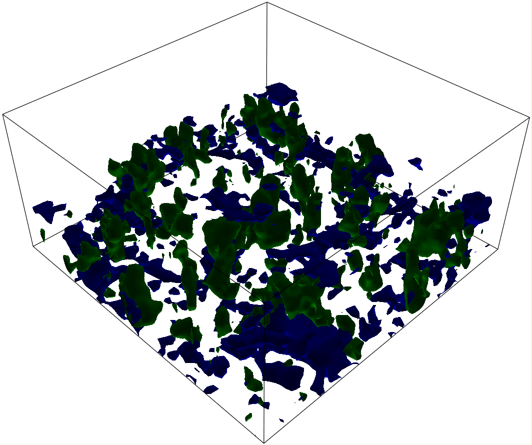
- setup following Margolin et al., 1999

example: convective PBL



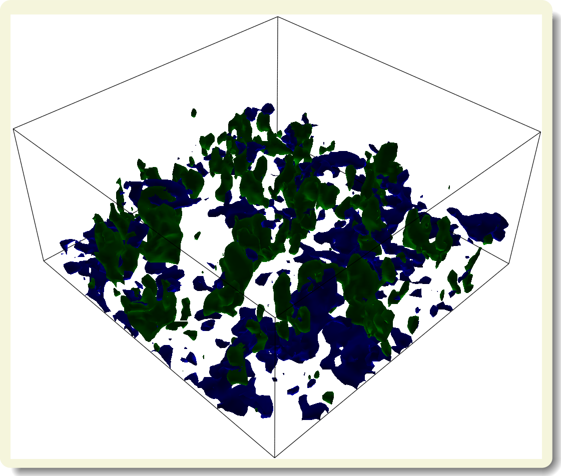
- setup following Margolin et al., 1999

example: convective PBL



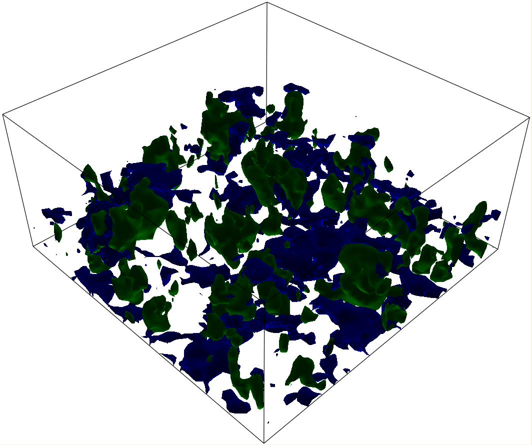
- setup following Margolin et al., 1999

example: convective PBL



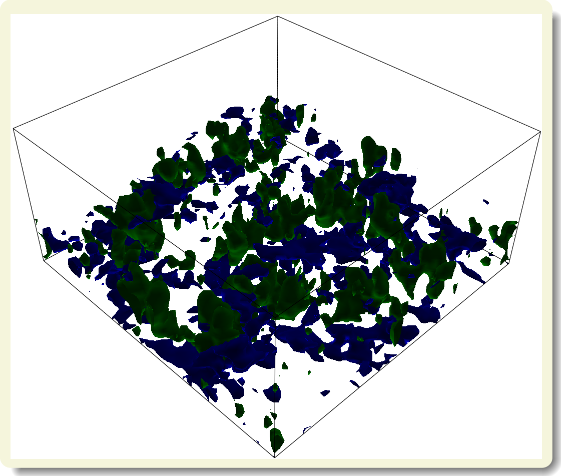
- setup following Margolin et al., 1999

example: convective PBL



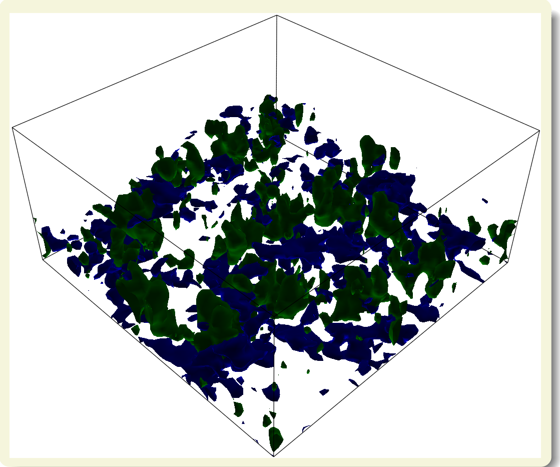
- setup following Margolin et al., 1999

example: convective PBL



- setup following Margolin et al., 1999

example: convective PBL



- setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

ongoing work

ongoing work

- higher-order operators for subgrid-scale modelling

ongoing work

- higher-order operators for subgrid-scale modelling
- distributed memory parallelisation (Boost.MPI)

ongoing work

- higher-order operators for subgrid-scale modelling
- distributed memory parallelisation (Boost.MPI)
- complex geometries using immersed boundary method

ongoing work

- higher-order operators for subgrid-scale modelling
- distributed memory parallelisation (Boost.MPI)
- complex geometries using immersed boundary method
- extension to anelastic model

ongoing work

- higher-order operators for subgrid-scale modelling
- distributed memory parallelisation (Boost.MPI)
- complex geometries using immersed boundary method
- extension to anelastic model

Thank you for your attention!