What's new in libmpdata++ (towards the 2.0 release)

Sylwester Arabas, Anna Jaruga, Maciej Waruszewski

Atmospheric Physics Seminar
Faculty of Physics, University of Warsaw, Poland

Warsaw, October 23, 2015

Plan of the talk

1. what's libmpdata++
2. libmpdata++ 1.0: summary of features
3. libmpdata++ 2.0: new features under development
4. libmpdata++: a hello-world program
5. closing remarks

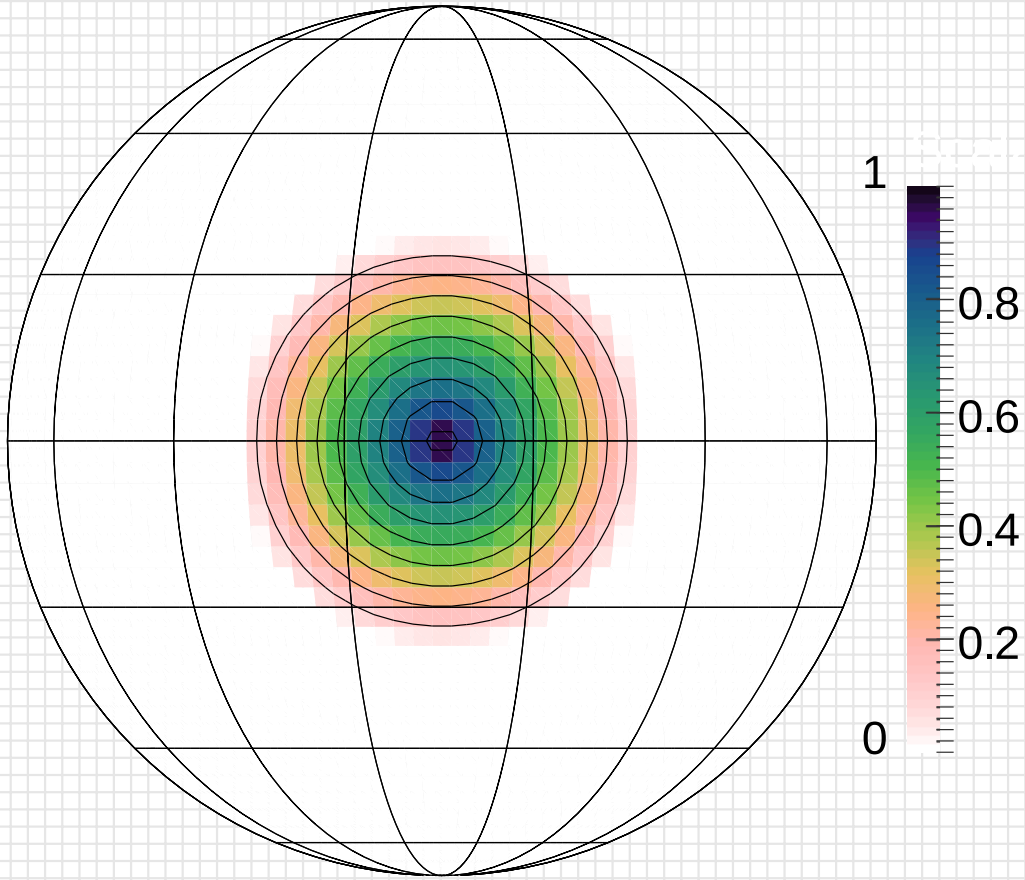
Plan of the talk

the generalised transport equation

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

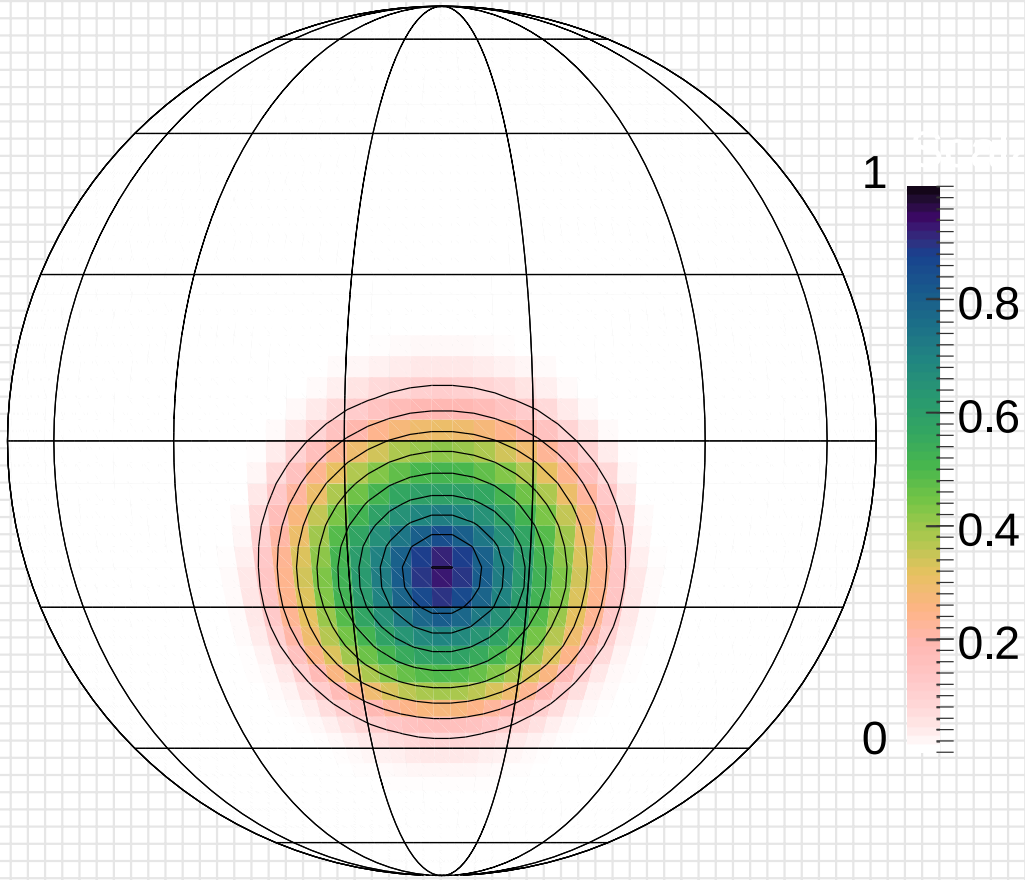
the generalised transport equation

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



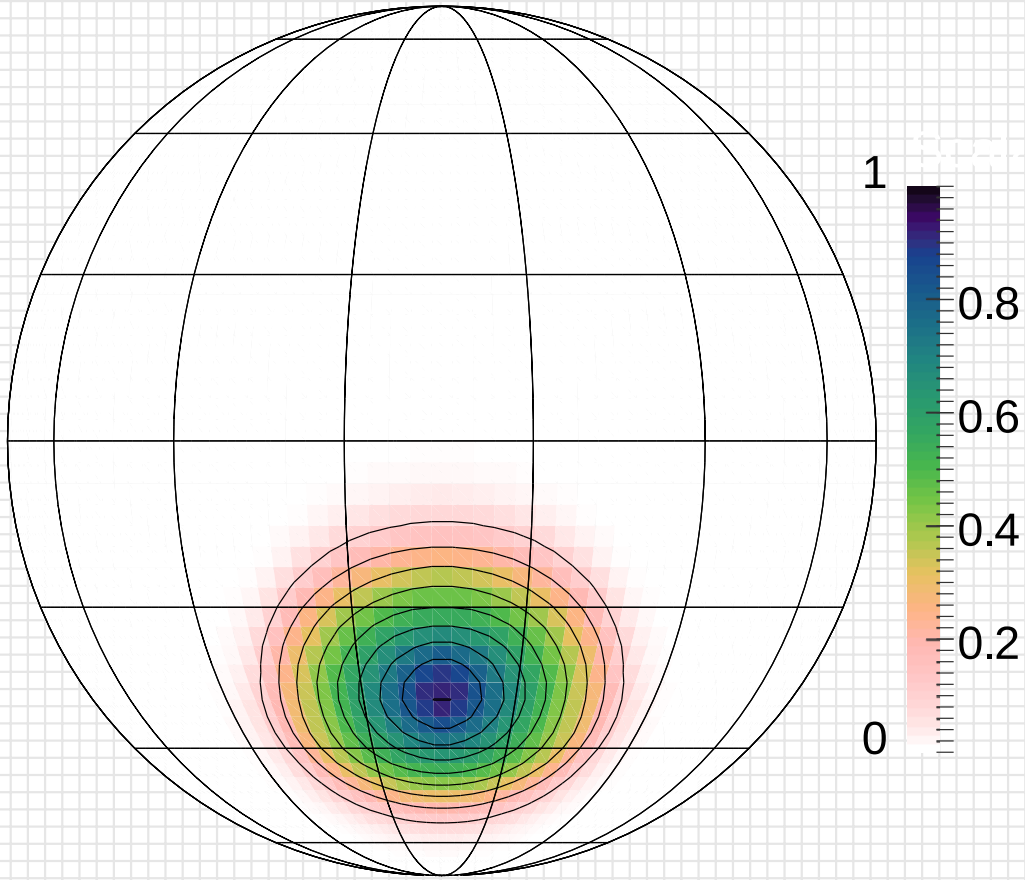
the generalised transport equation

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



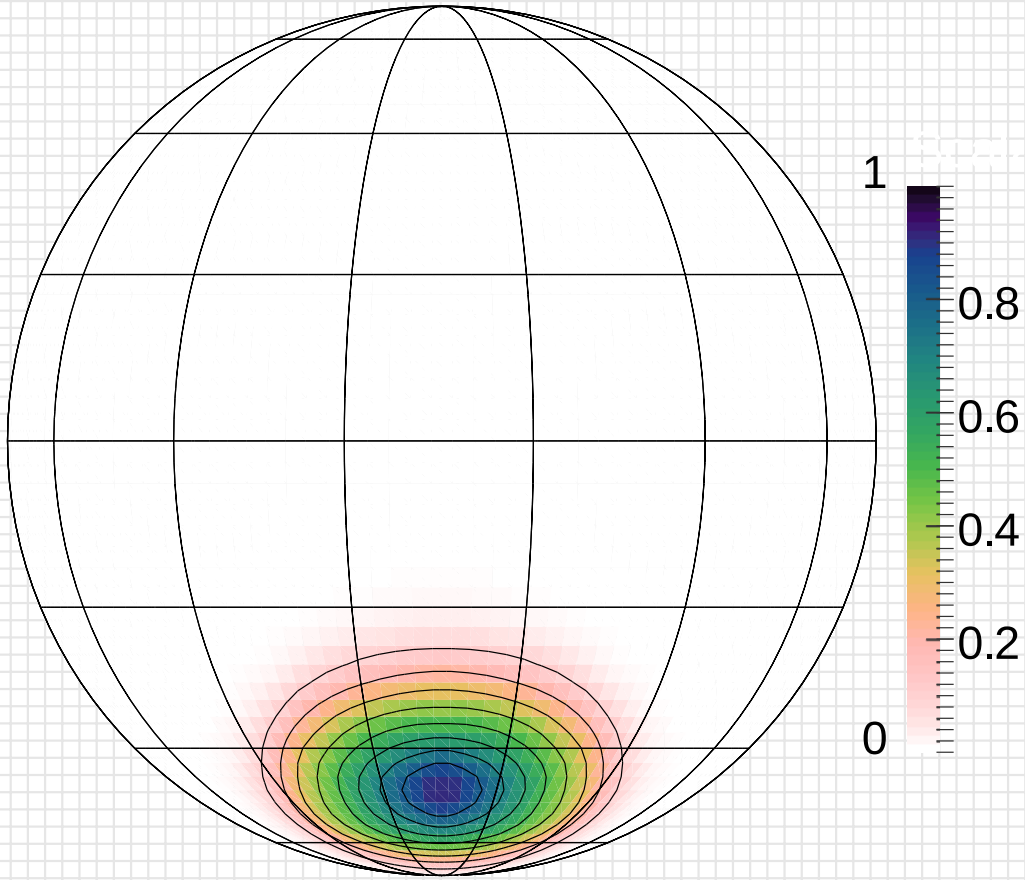
the generalised transport equation

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



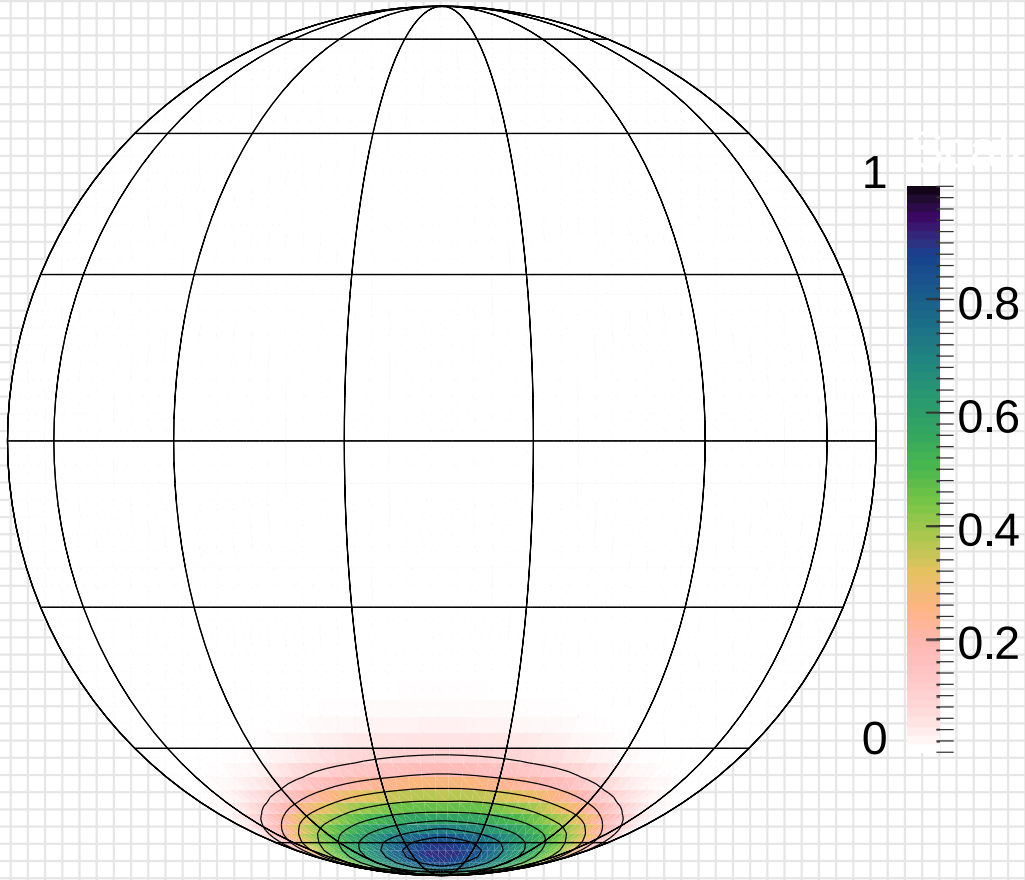
the generalised transport equation

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



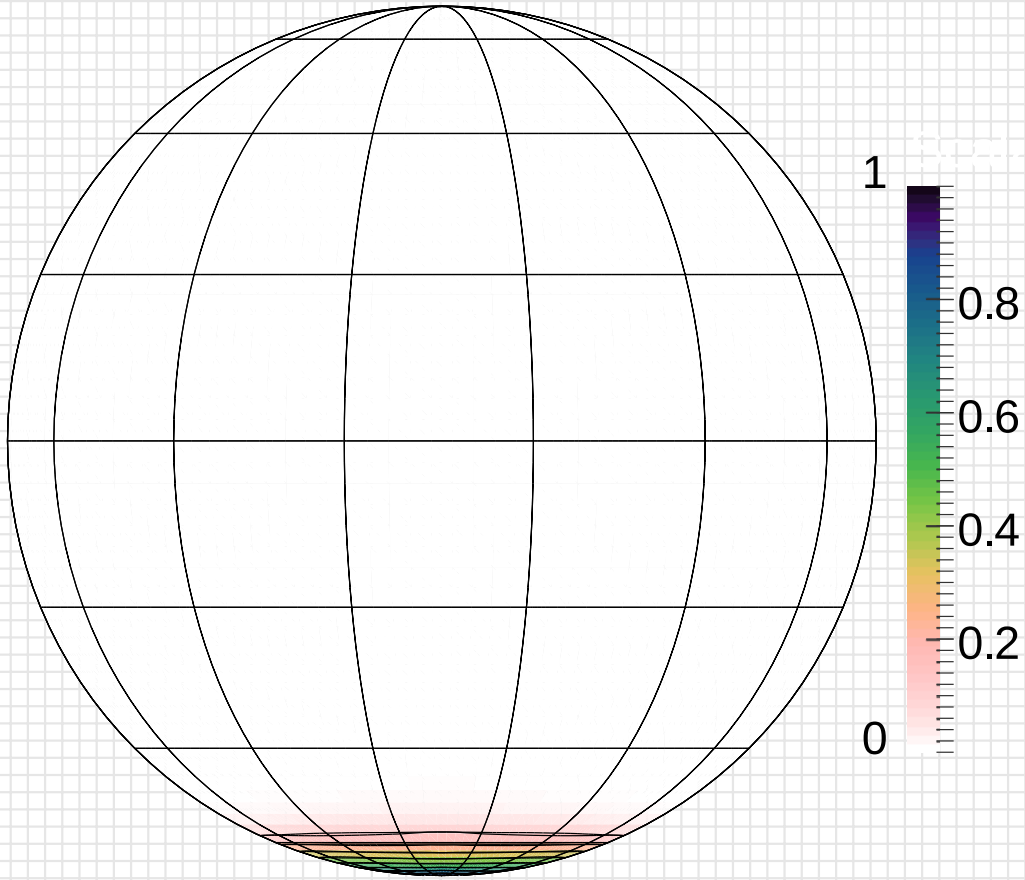
the generalised transport equation

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



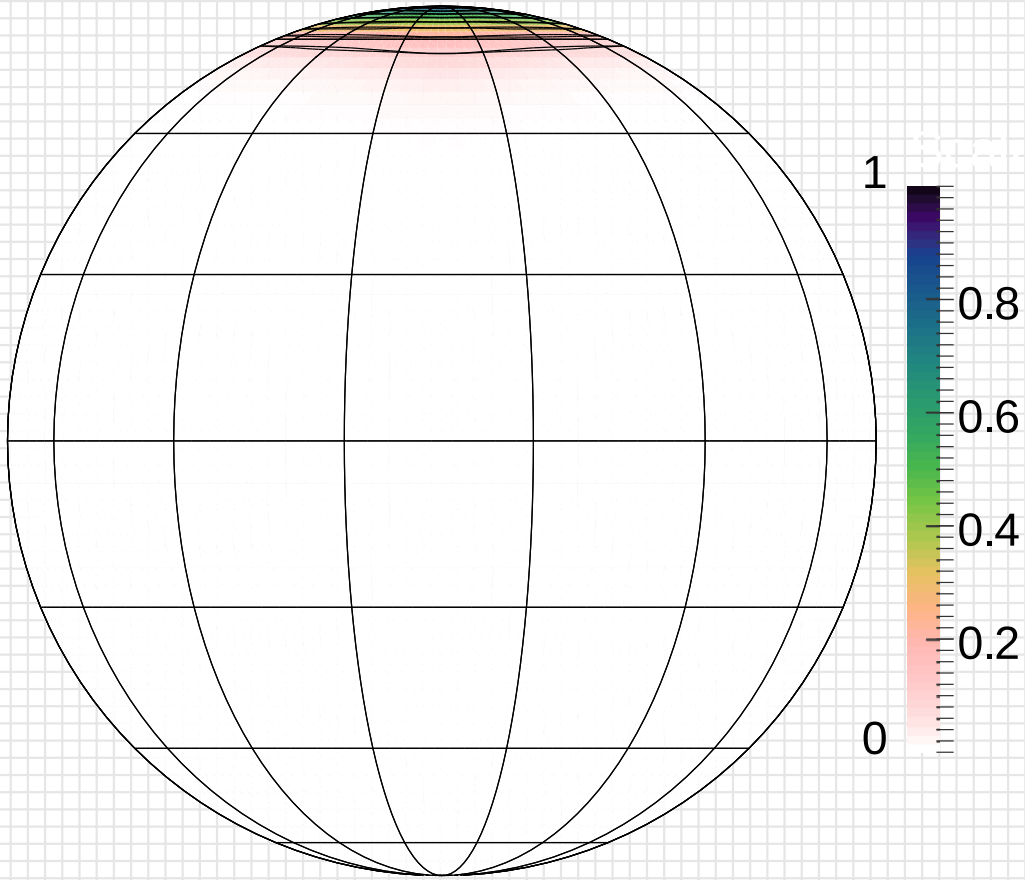
the generalised transport equation

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



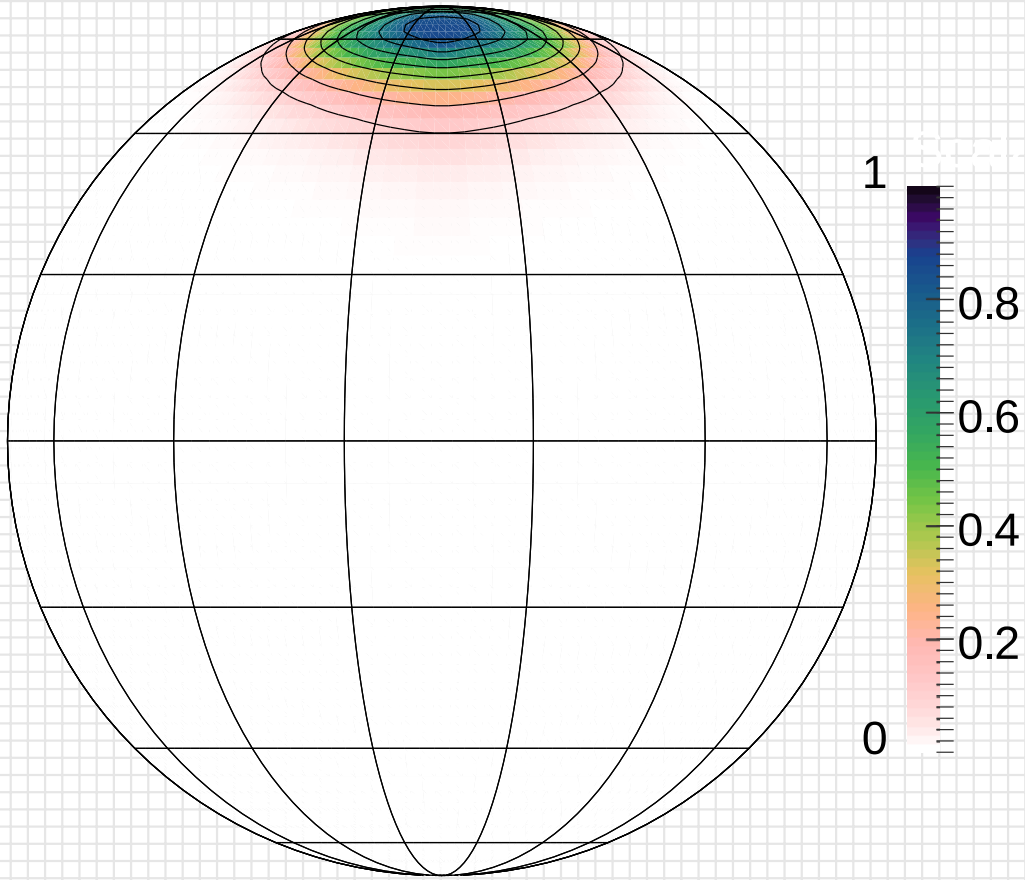
the generalised transport equation

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



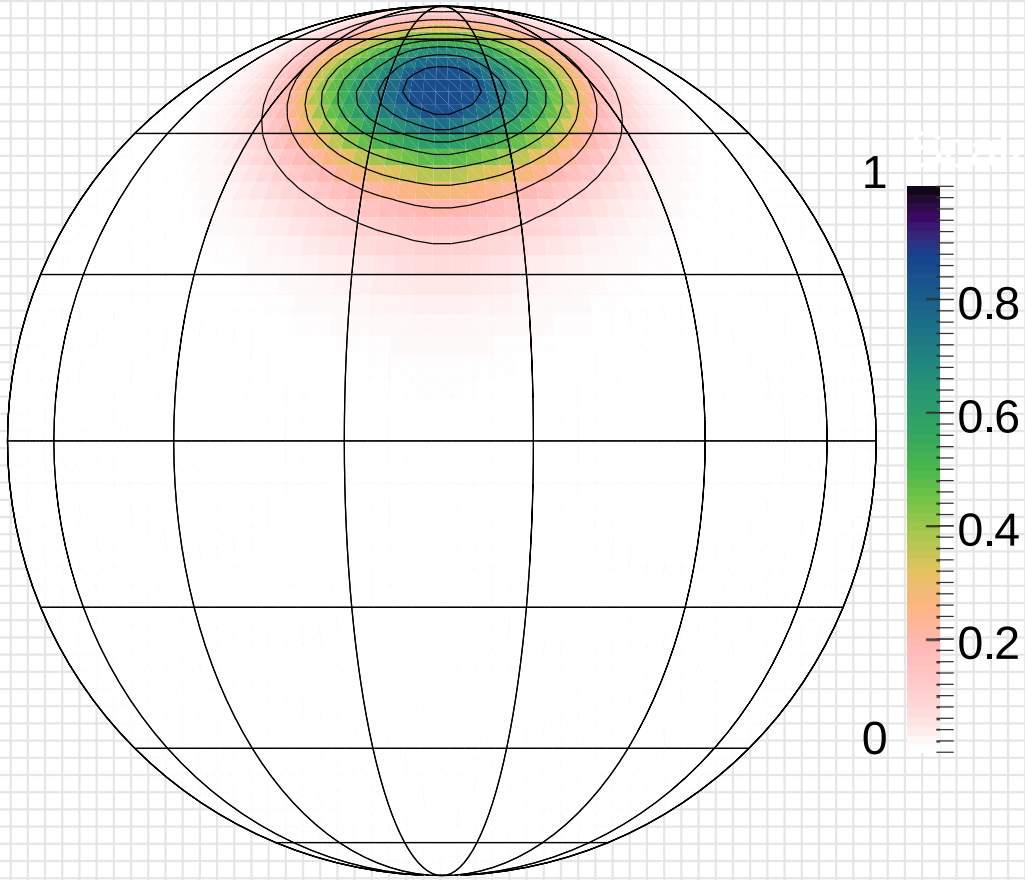
the generalised transport equation

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



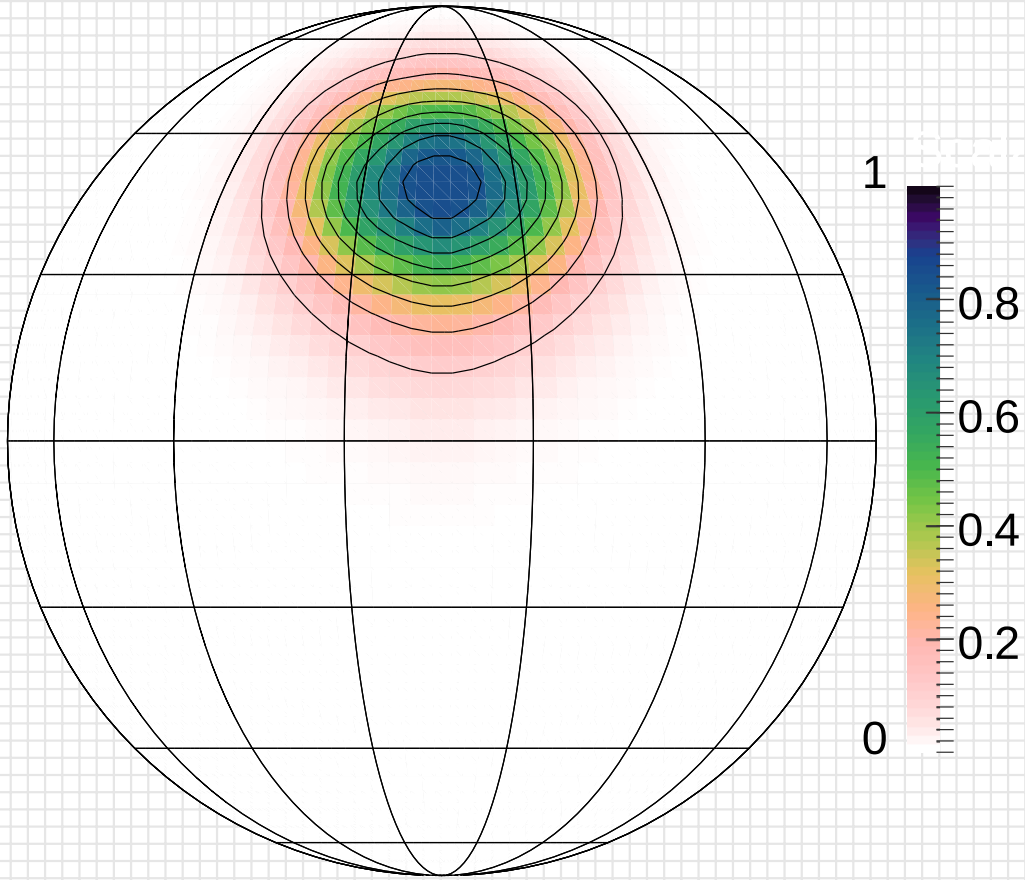
the generalised transport equation

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



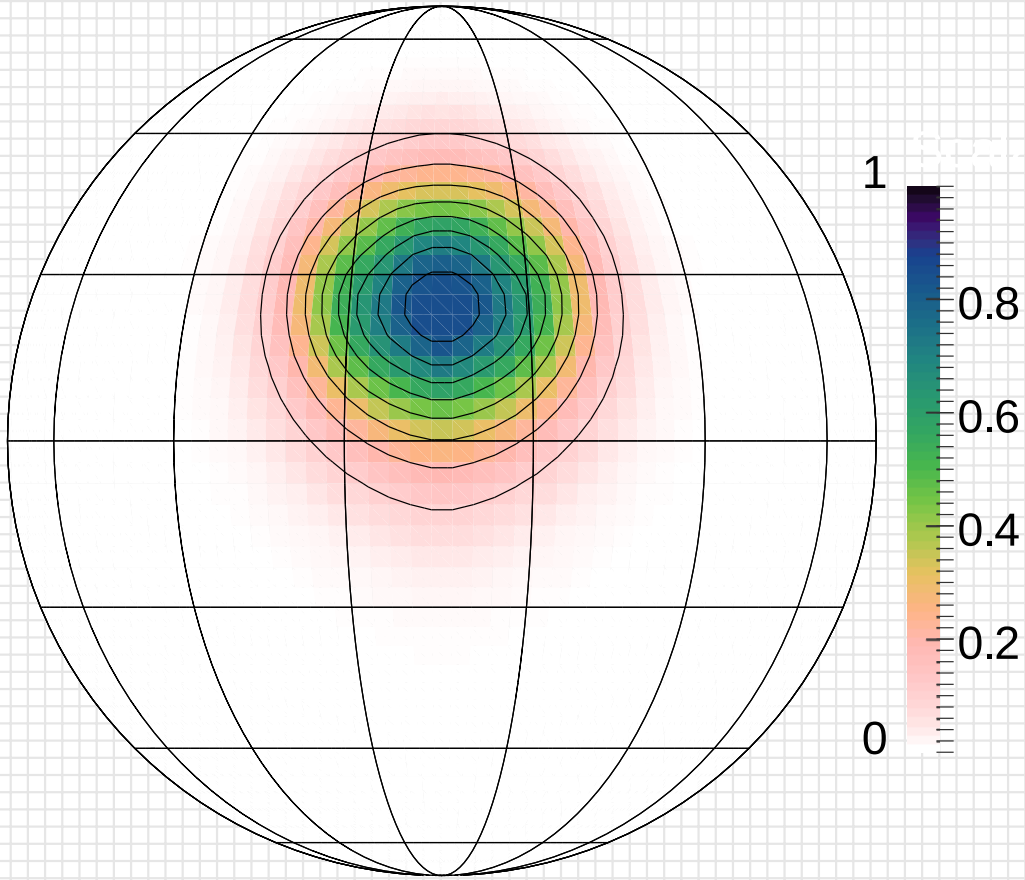
the generalised transport equation

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



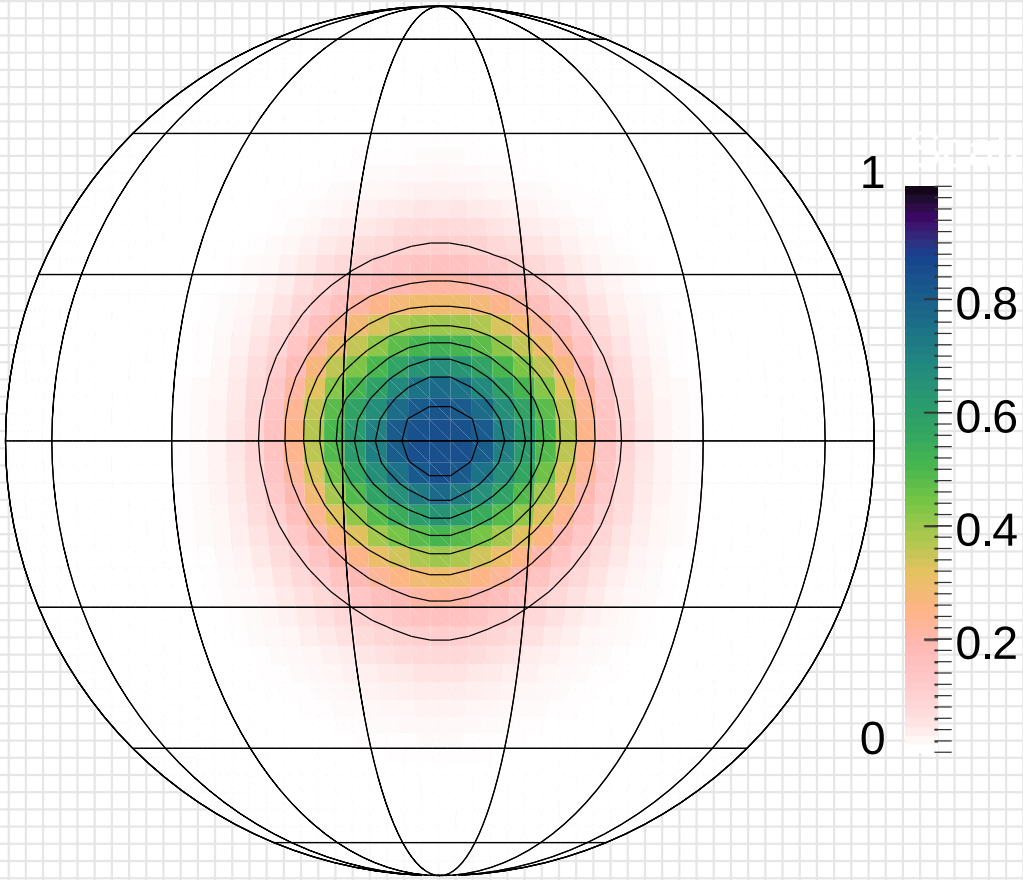
the generalised transport equation

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



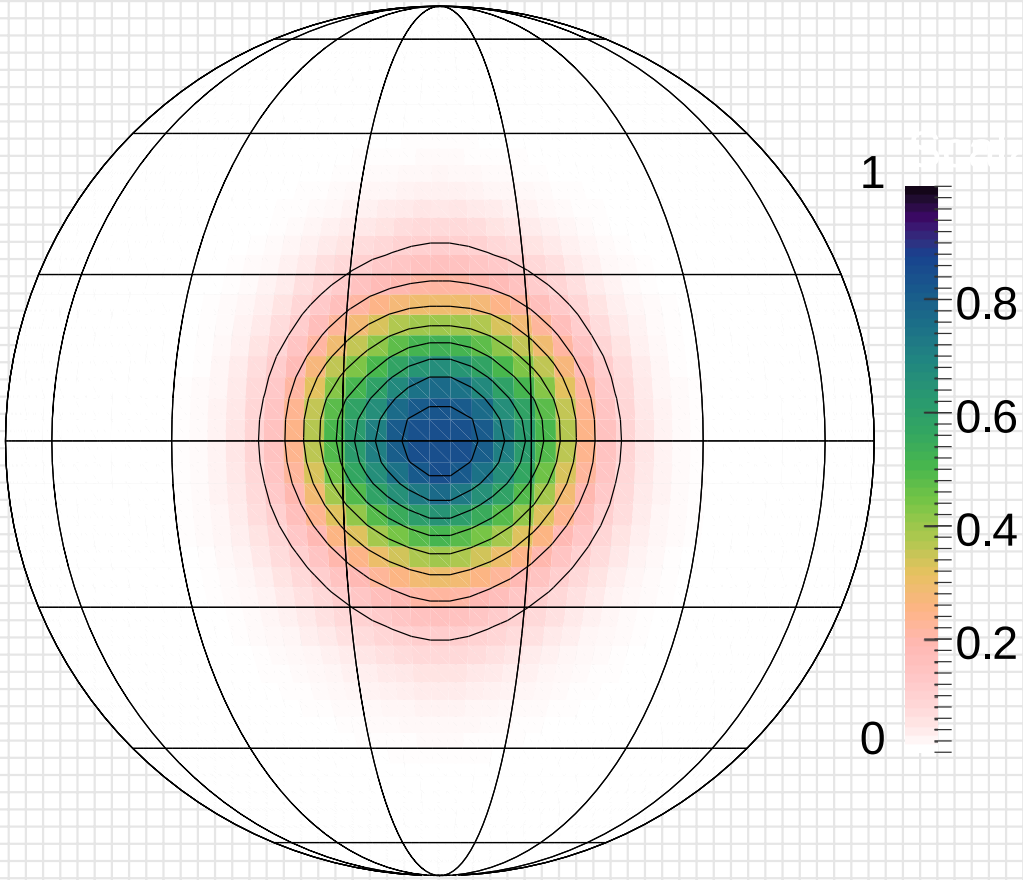
the generalised transport equation

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



the generalised transport equation

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



numerical integration using MPDATA

MPDATA (father: Piotr Smolarkiewicz)

Multi-dimensional Positive-Definite Advection Transport Algorithm

a family of robust schemes for solving transport problems

- the seminal MPDATA article (Smolarkiewicz, 1984): >600 citations
- Google Scholar: $\sim$ 700 research papers
- Google Books: $\sim$ 200 mentions in books

libmpdata++: a new C++11 / Blitz++ based implementation

features compared with the original F77 implementation:

- an over order-of-magnitude lower number of lines of code
- comparable performance
- major improvement in reusability and maintainability

MPDATA (father: Piotr Smolarkiewicz)

Multi-dimensional Positive-Definite Advection Transport Algorithm

a family of robust schemes for solving transport problems

- the seminal MPDATA article (Smolarkiewicz, 1984): >600 citations
- Google Scholar: $\sim$ 700 research papers
- Google Books: $\sim$ 200 mentions in books

libmpdata++: a new C++11 / Blitz++ based implementation

features compared with the original F77 implementation:

- an over order-of-magnitude lower number of lines of code
- comparable performance
- major improvement in reusability and maintainability

MPDATA (father: Piotr Smolarkiewicz)

Multi-dimensional Positive-Definite Advection Transport Algorithm

a family of robust schemes for solving transport problems

- the seminal MPDATA article (Smolarkiewicz, 1984): >600 citations
- Google Scholar: $\sim$ 700 research papers
- Google Books: $\sim$ 200 mentions in books

libmpdata++: a new C++11 / Blitz++ based implementation

features compared with the original F77 implementation:

- an over order-of-magnitude lower number of lines of code
- comparable performance
- major improvement in reusability and maintainability

MPDATA (father: Piotr Smolarkiewicz)

Multi-dimensional Positive-Definite Advection Transport Algorithm

a family of robust schemes for solving transport problems

- the seminal MPDATA article (Smolarkiewicz, 1984): >600 citations
- Google Scholar: $\sim$ 700 research papers
- Google Books: $\sim$ 200 mentions in books

libmpdata++: a new C++11 / Blitz++ based implementation

features compared with the original F77 implementation:

- an over order-of-magnitude lower number of lines of code
- comparable performance
- major improvement in reusability and maintainability

MPDATA (father: Piotr Smolarkiewicz)

Multi-dimensional Positive-Definite Advection Transport Algorithm

a family of robust schemes for solving transport problems

- the seminal MPDATA article (Smolarkiewicz, 1984): >600 citations
- Google Scholar: $\sim$ 700 research papers
- Google Books: $\sim$ 200 mentions in books

libmpdata++: a new C++11 / Blitz++ based implementation

features compared with the original F77 implementation:

- an over order-of-magnitude lower number of lines of code
- comparable performance
- major improvement in reusability and maintainability

MPDATA (father: Piotr Smolarkiewicz)

Multi-dimensional Positive-Definite Advection Transport Algorithm

a family of robust schemes for solving transport problems

- the seminal MPDATA article (Smolarkiewicz, 1984): >600 citations
- Google Scholar: $\sim$ 700 research papers
- Google Books: $\sim$ 200 mentions in books

libmpdata++: a new C++11 / Blitz++ based implementation

features compared with the original F77 implementation:

- an over order-of-magnitude lower number of lines of code
- comparable performance
- major improvement in reusability and maintainability

MPDATA (father: Piotr Smolarkiewicz)

Multi-dimensional Positive-Definite Advection Transport Algorithm

a family of robust schemes for solving transport problems

- the seminal MPDATA article (Smolarkiewicz, 1984): >600 citations
- Google Scholar: $\sim$ 700 research papers
- Google Books: $\sim$ 200 mentions in books

libmpdata++: a new C++11 / Blitz++ based implementation

features compared with the original F77 implementation:

- an over order-of-magnitude lower number of lines of code
- comparable performance
- major improvement in reusability and maintainability

MPDATA (father: Piotr Smolarkiewicz)

Multi-dimensional Positive-Definite Advection Transport Algorithm

a family of robust schemes for solving transport problems

- the seminal MPDATA article (Smolarkiewicz, 1984): >600 citations
- Google Scholar: $\sim$ 700 research papers
- Google Books: $\sim$ 200 mentions in books

libmpdata++: a new C++11 / Blitz++ based implementation

features compared with the original F77 implementation:

- an over order-of-magnitude lower number of lines of code
- comparable performance
- major improvement in reusability and maintainability

priority: researchers' productivity

researcher = user

- **ease of obtaining and using**
  ⇝ public repository, documentation, examples
- **result correctness**
  ⇝ multifaceted peer-reviewed automated tests
- **result reproducibility**
  ⇝ atomic versions, no legal nor tech. obstacles

researcher = developer

- **ease of extending**
  ⇝ concise OOP syntax, separation of concerns, automated tests, continuous integration

priority: researchers' productivity

researcher = user

- **ease of obtaining and using**
  $\rightsquigarrow$ public repository, documentation, examples
- **result correctness**
  $\rightsquigarrow$ multifaceted peer-reviewed automated tests
- **result reproducibility**
  $\rightsquigarrow$ atomic versions, no legal nor tech. obstacles

researcher = developer

- **ease of extending**
  $\rightsquigarrow$ concise OOP syntax, separation of concerns, automated tests, continuous integration

priority: researchers' productivity

researcher = user

- **ease of obtaining and using**
  $\rightsquigarrow$ public repository, documentation, examples
- **result correctness**
  $\rightsquigarrow$ multifaceted peer-reviewed automated tests
- **result reproducibility**
  $\rightsquigarrow$ atomic versions, no legal nor tech. obstacles

researcher = developer

- **ease of extending**
  $\rightsquigarrow$ concise OOP syntax, separation of concerns, automated tests, continuous integration

priority: researchers' productivity

researcher = user

- **ease of obtaining and using**
  ⇝ public repository, documentation, examples
- **result correctness**
  ⇝ multifaceted peer-reviewed automated tests
- **result reproducibility**
  ⇝ atomic versions, no legal nor tech. obstacles

researcher = developer

- **ease of extending**
  ⇝ concise OOP syntax, separation of concerns, automated tests, continuous integration

priority: researchers' productivity

researcher = user

- **ease of obtaining and using**
  $\rightsquigarrow$ public repository, documentation, examples
- **result correctness**
  $\rightsquigarrow$ multifaceted peer-reviewed automated tests
- **result reproducibility**
  $\rightsquigarrow$ atomic versions, no legal nor tech. obstacles

researcher = developer

- **ease of extending**
  $\rightsquigarrow$ concise OOP syntax, separation of concerns, automated tests, continuous integration

priority: researchers' productivity

researcher = user

- **ease of obtaining and using**
  ⇝ public repository, documentation, examples
- **result correctness**
  ⇝ multifaceted peer-reviewed automated tests
- **result reproducibility**
  ⇝ atomic versions, no legal nor tech. obstacles

researcher = developer

- **ease of extending**
  ⇝ concise OOP syntax, separation of concerns, automated tests, continuous integration

Plan of the talk

1. what's libmpdata++
2. libmpdata++ 1.0: summary of features
3. libmpdata++ 2.0: new features under development
4. libmpdata++: a hello-world program
5. closing remarks

Plan of the talk

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++ 1.0: 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"*

libmpdata++ 1.0: 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"*

libmpdata++ 1.0: 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"*

libmpdata++ 1.0: 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"*

libmpdata++ 1.0: 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"*

libmpdata++ 1.0: documentation

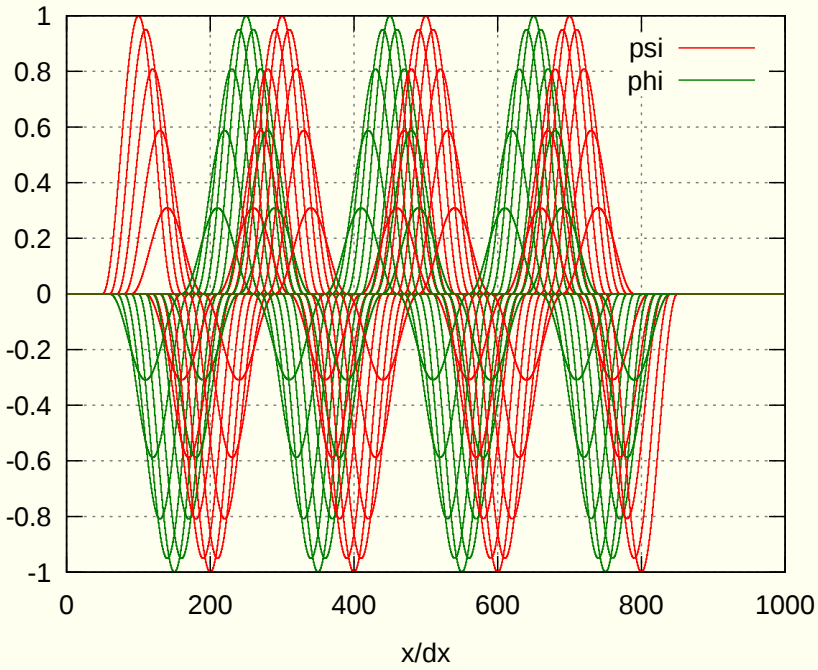


Figure 15. Simulation results of the example presented in Sect. 4.3. Abscissa marks the spatial dimension and ordinate represents the oscillator amplitude. The oscillator state is plotted every 20 time steps.

(partial differential equation) system (16) leads to the following system of coupled implicit algebraic equations:

$$\psi_i^{n+1} = \psi_i^* + 0.5\ \Delta t\ \omega\ \phi_i^{n+1},$$
$$\phi_i^{n+1} = \phi_i^* - 0.5\ \Delta t\ \omega\ \psi_i^{n+1}, \qquad (17)$$

```
#include <libmpdata++/solvers/mpdata_rhs.hpp>

template <class ct_params_t>
struct coupled_harmosc : public
  libmpdataxx::solvers::mpdata_rhs<ct_params_t>
{ // aliases
  using parent_t =
    libmpdataxx::solvers::mpdata_rhs<ct_params_t>;
  using ix = typename ct_params_t::ix;
  // member fields
  typename ct_params_t::real_t omega;

  // method called by mpdata_rhs
  void update_rhs(
    libmpdataxx::arrvec_t<
      typename parent_t::arr_t
    > &rhs,
    const typename parent_t::real_t &dt,
    const int &at
  ) {
    parent_t::update_rhs(rhs, dt, at);

    // just to shorten code
    const auto &psi = this->state(ix::psi);
    const auto &phi = this->state(ix::phi);
    const auto &i = this->i;

    switch (at)
    { // explicit solution for R^{n}
      // (note: with trapez used only at t=0)
      case (0):
      rhs.at(ix::psi)(i) += omega * phi(i);
      rhs.at(ix::phi)(i) -= omega * psi(i);
```

libmpdata++ 1.0: solver/algorithm hierarchy

libmpdata++ 1.0: solver/algorithm hierarchy

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

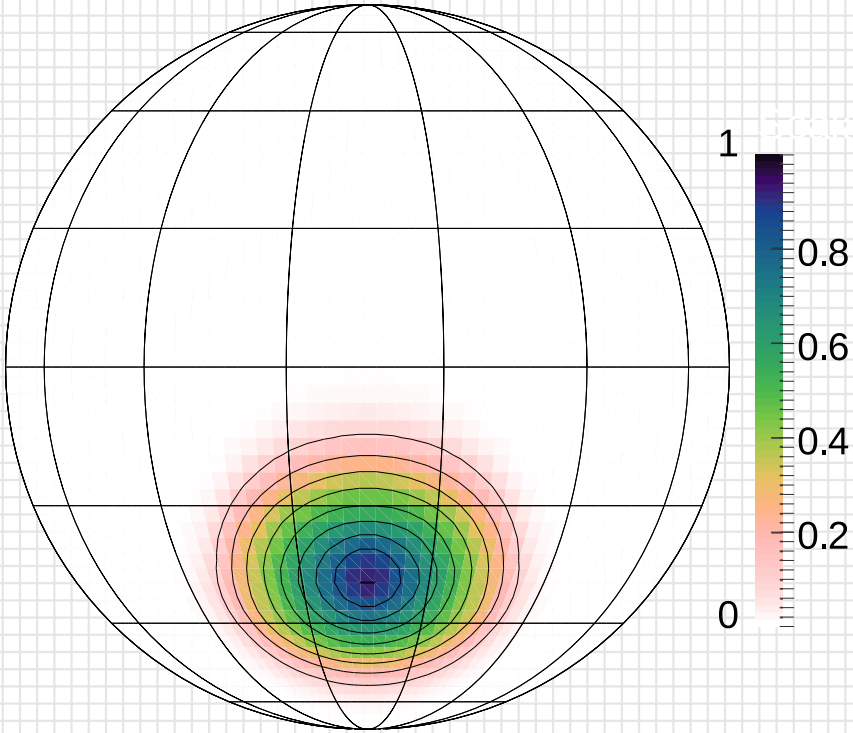
homogeneous advection

libmpdata++ 1.0: solver/algorithm hierarchy

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

homogeneous advection

user/test code

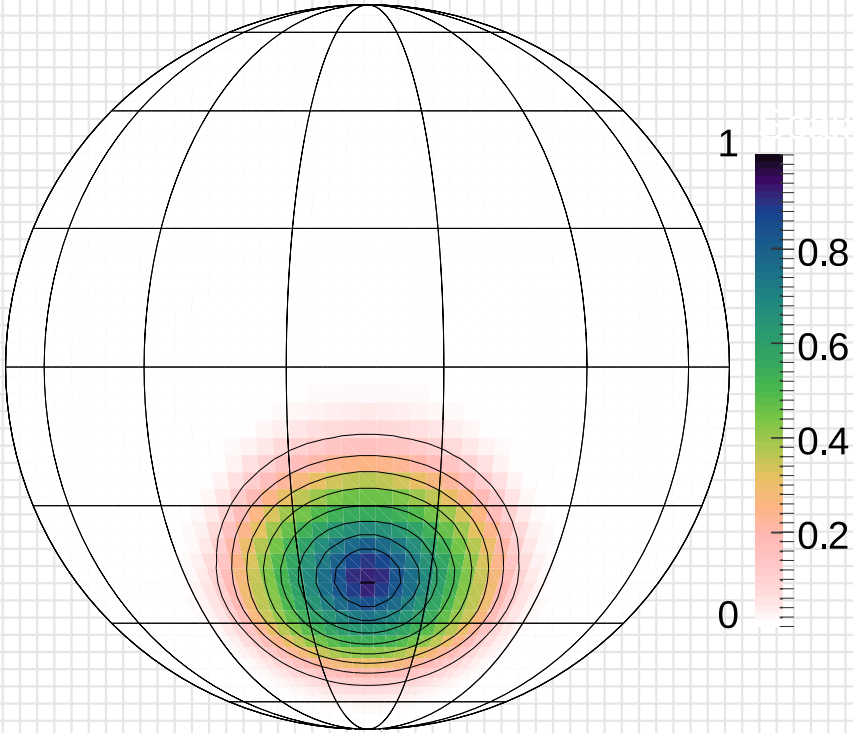


libmpdata++ 1.0: solver/algorithm hierarchy

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

homogeneous advection

user/test code



libmpdata++ 1.0: solver/algorithm hierarchy

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

homogeneous advection

+

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

source terms

libmpdata++ 1.0: solver/algorithm hierarchy

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

homogeneous advection

+

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

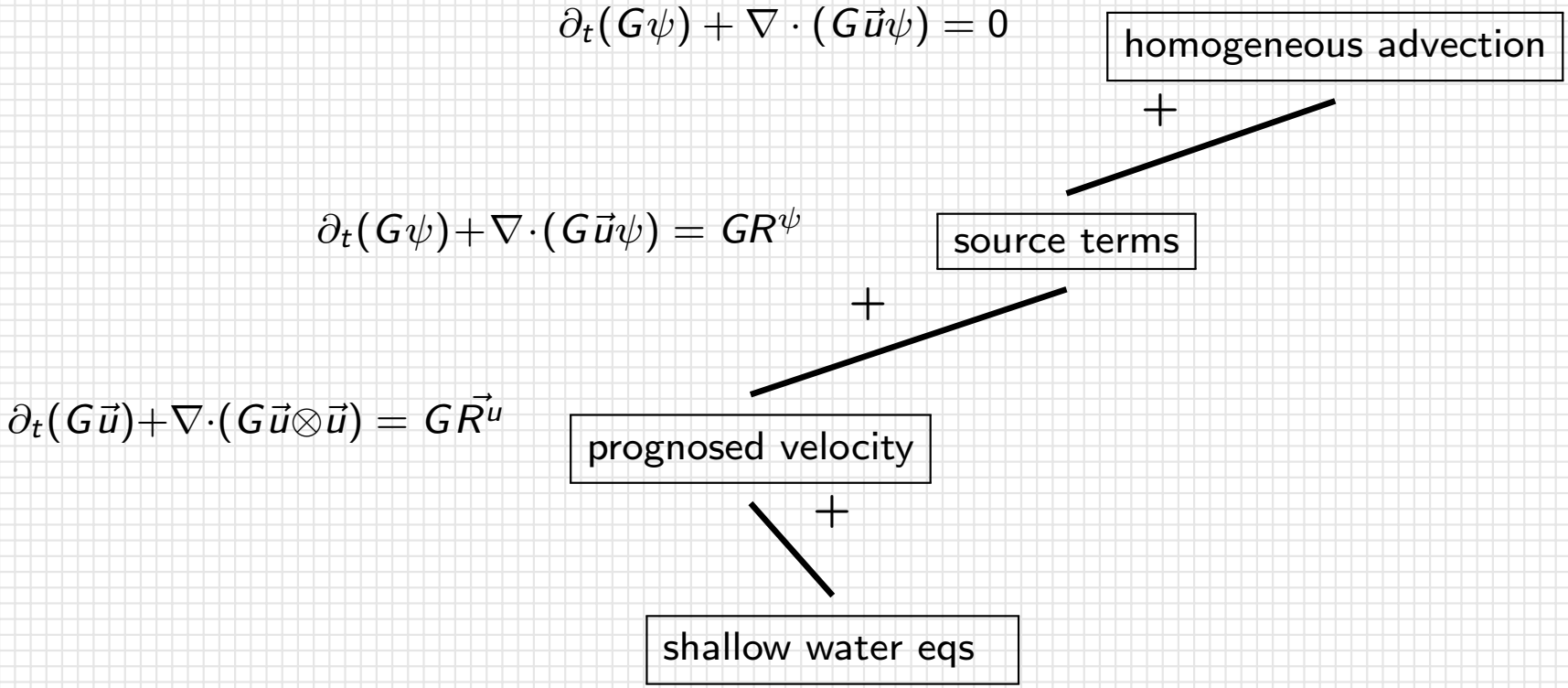
source terms

+

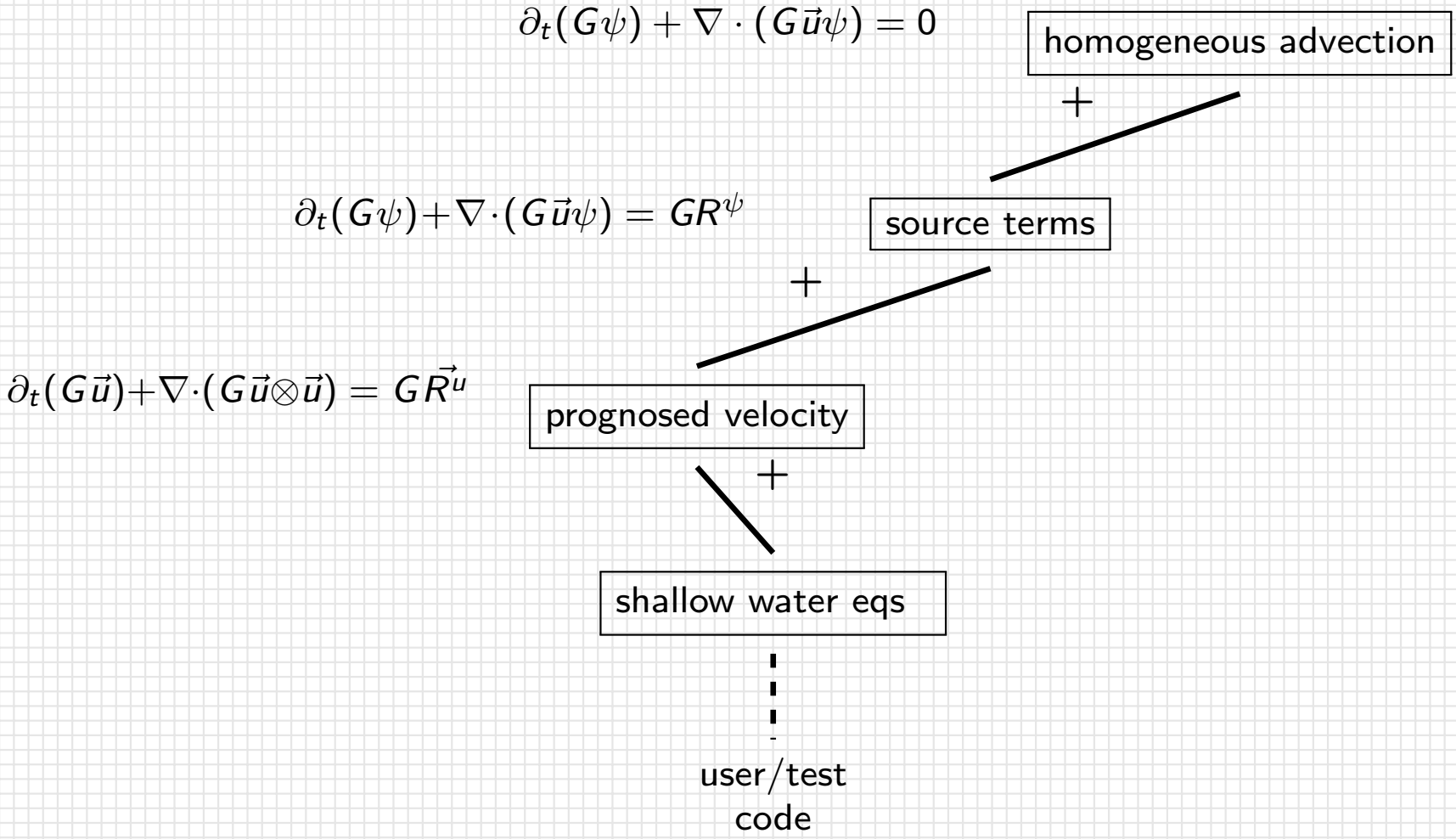
$$\partial_t(G\vec{u}) + \nabla \cdot (G\vec{u} \otimes \vec{u}) = G\vec{R}^{u}$$

prognosed velocity

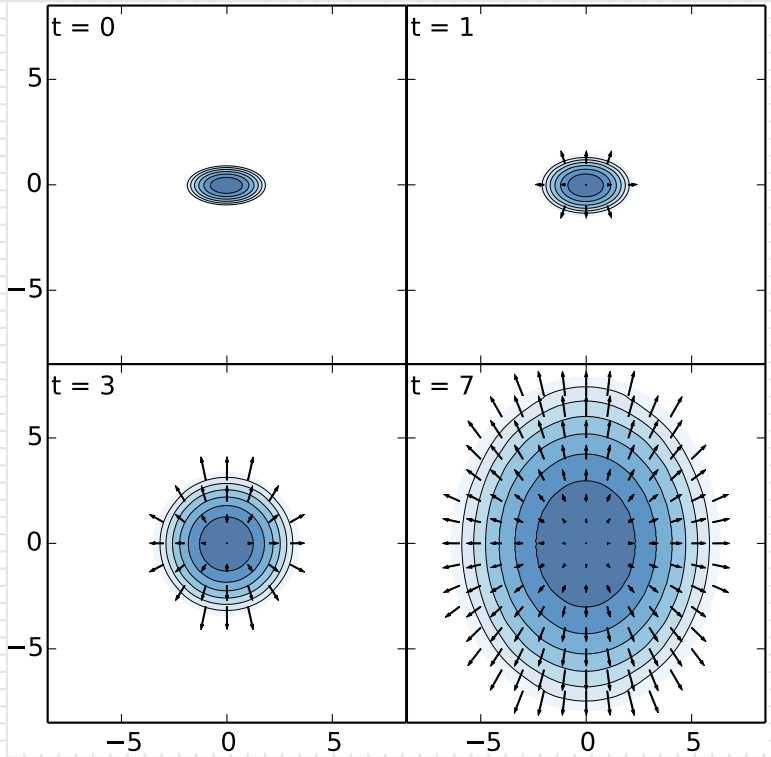
libmpdata++ 1.0: solver/algorithm hierarchy



libmpdata++ 1.0: solver/algorithm hierarchy

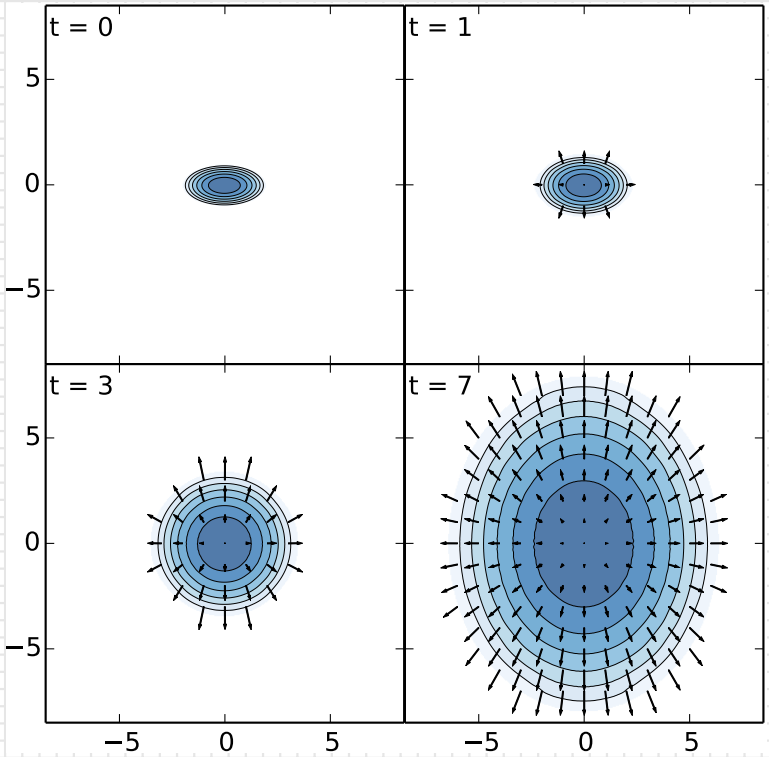


libmpdata++: 3D shallow-water system example



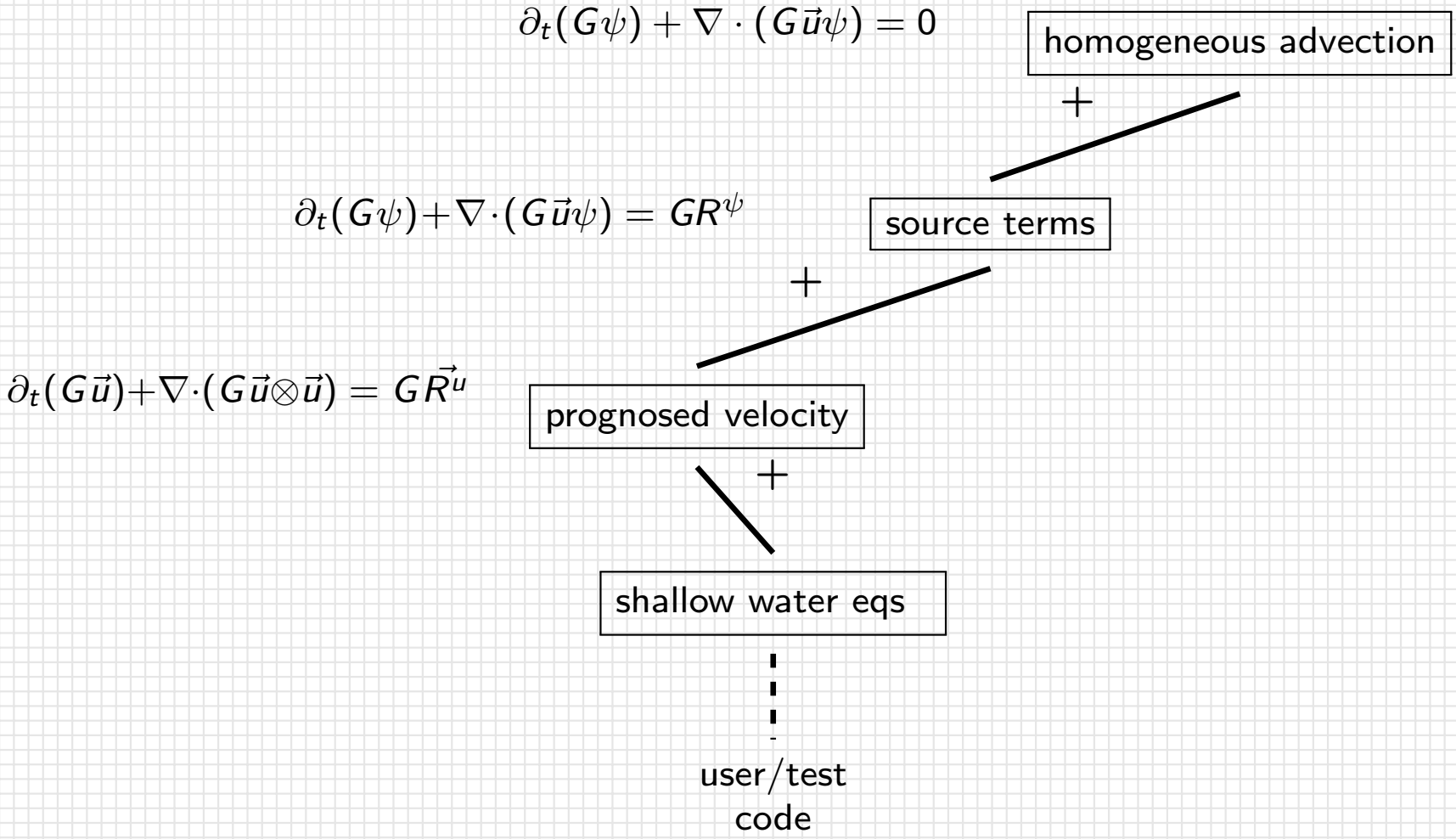
- inspired by 2D experiment of Schär and Smolarkiewicz, 1996
- example and original analytic solution by Dorota Jarecka / NCAR (Jarecka, Jaruga & Smolarkiewicz 2015, J. Comp. Phys. 289)

libmpdata++: 3D shallow-water system example

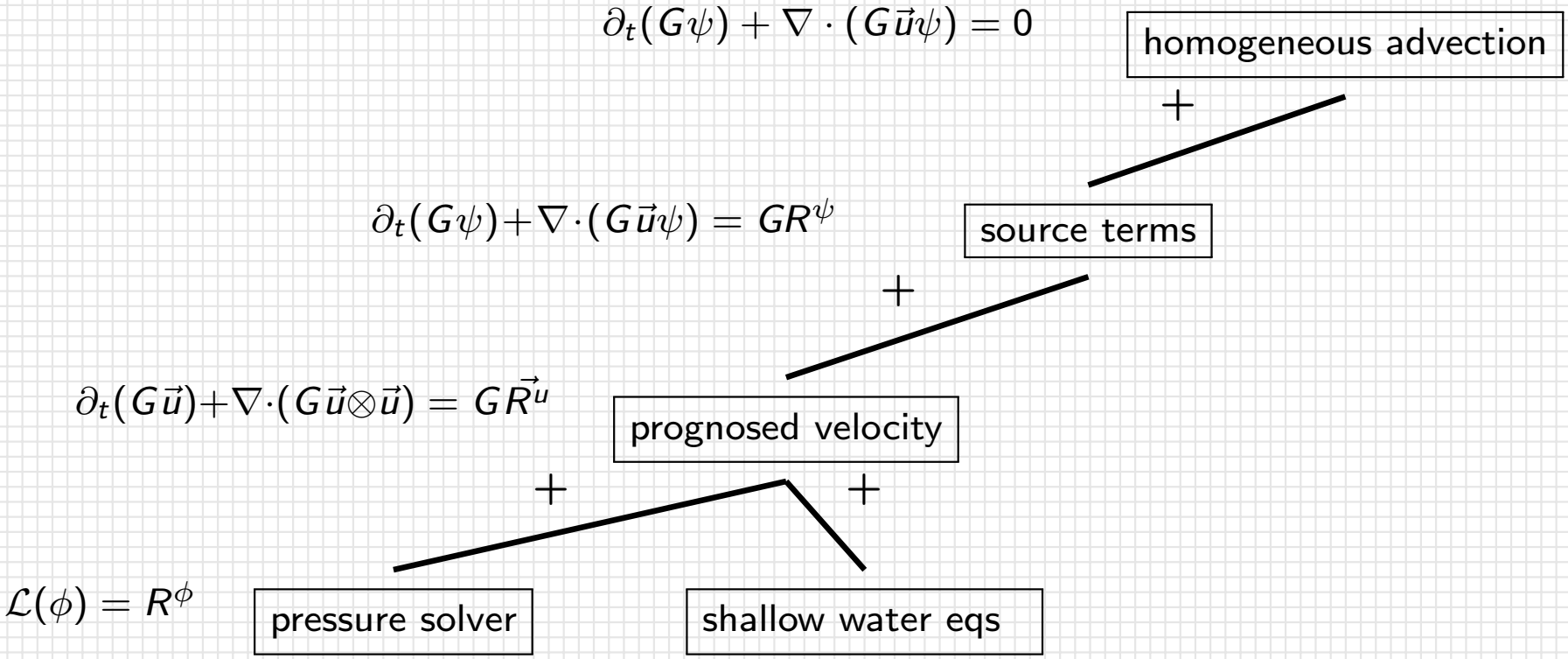


- inspired by 2D experiment of Schär and Smolarkiewicz, 1996
- example and original analytic solution by Dorota Jarecka / NCAR (Jarecka, Jaruga & Smolarkiewicz 2015, J. Comp. Phys. 289)

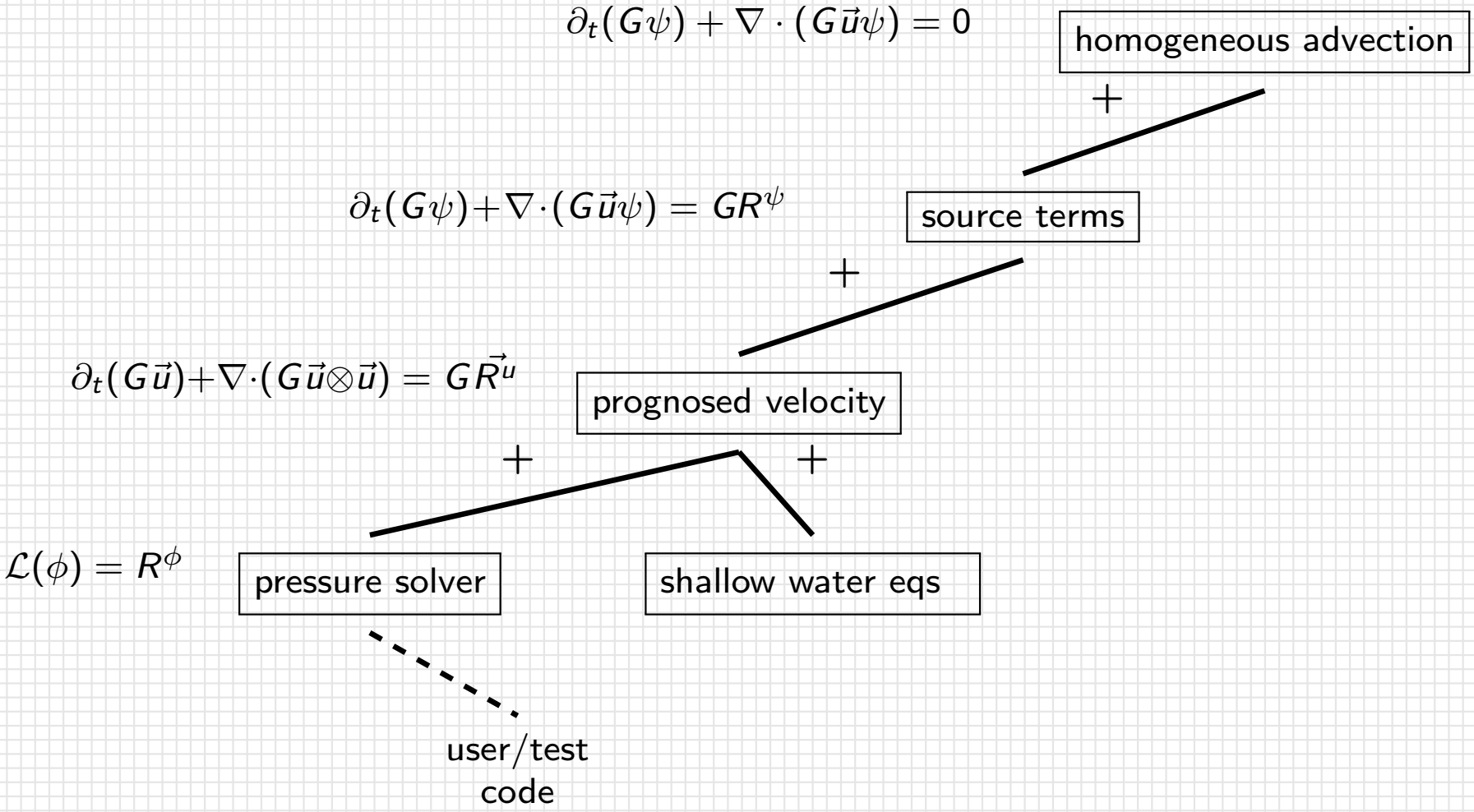
libmpdata++ 1.0: solver/algorithm hierarchy



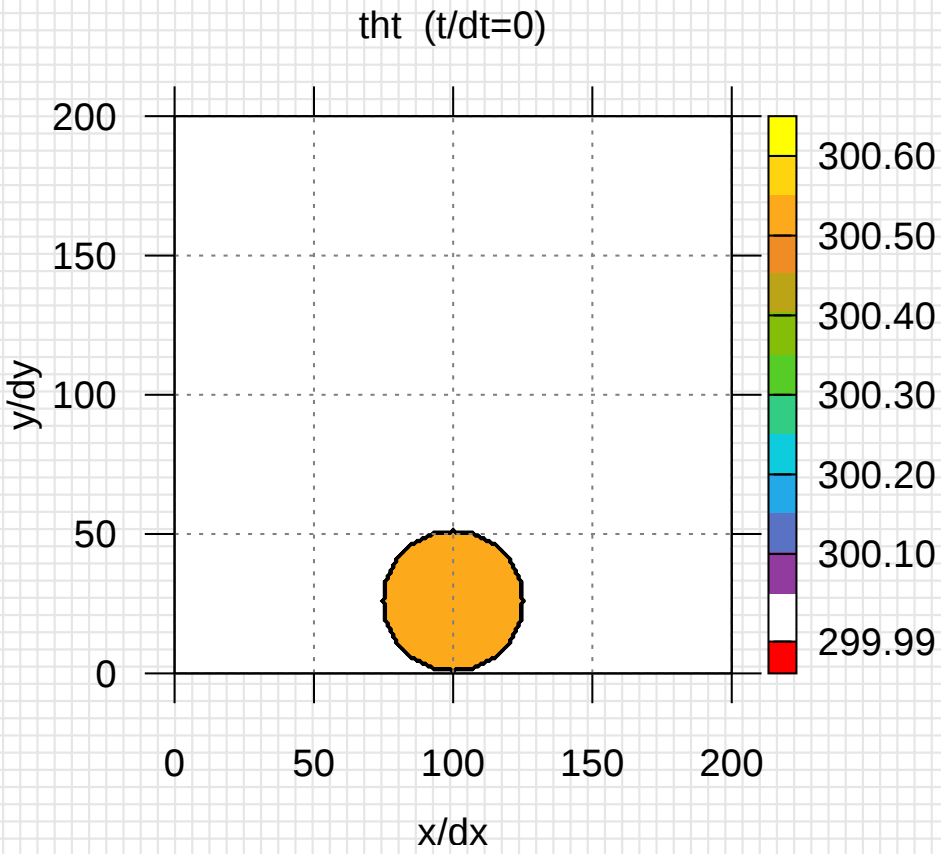
libmpdata++ 1.0: solver/algorithm hierarchy



libmpdata++ 1.0: solver/algorithm hierarchy



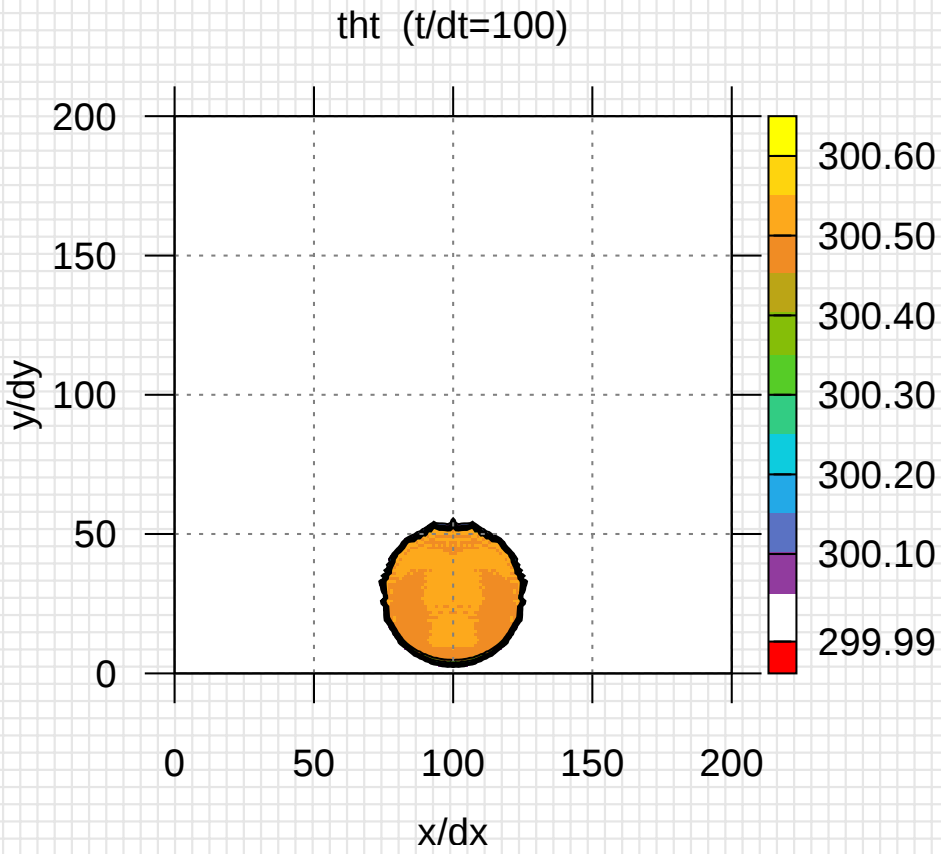
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

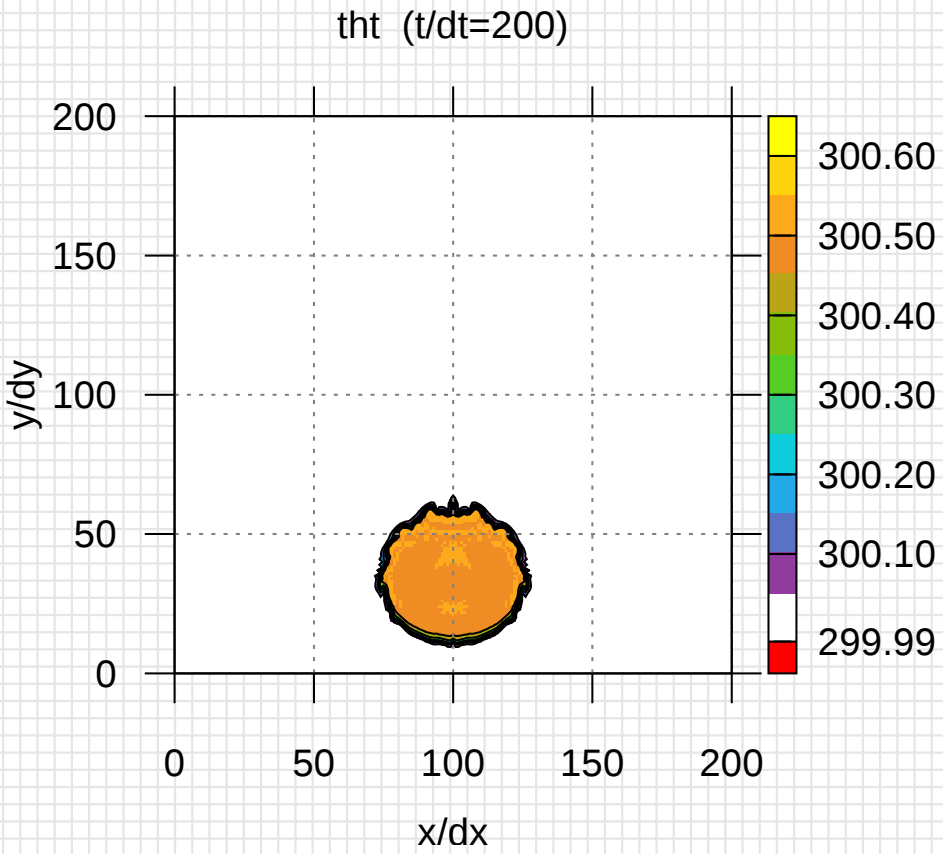
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

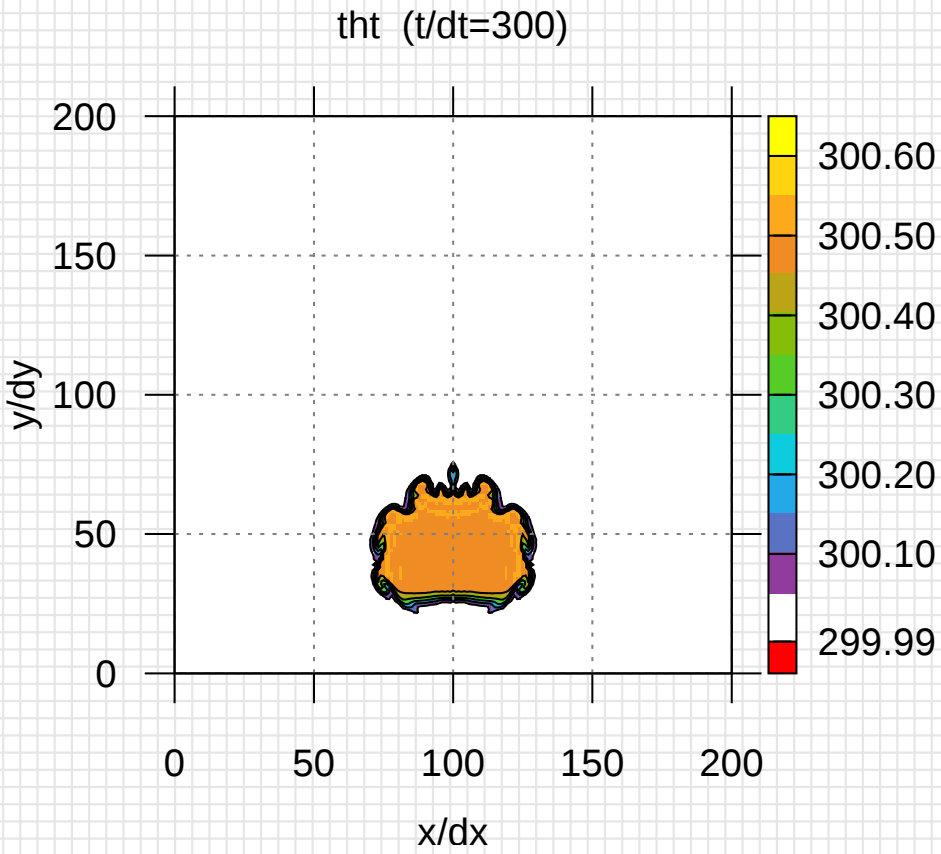
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

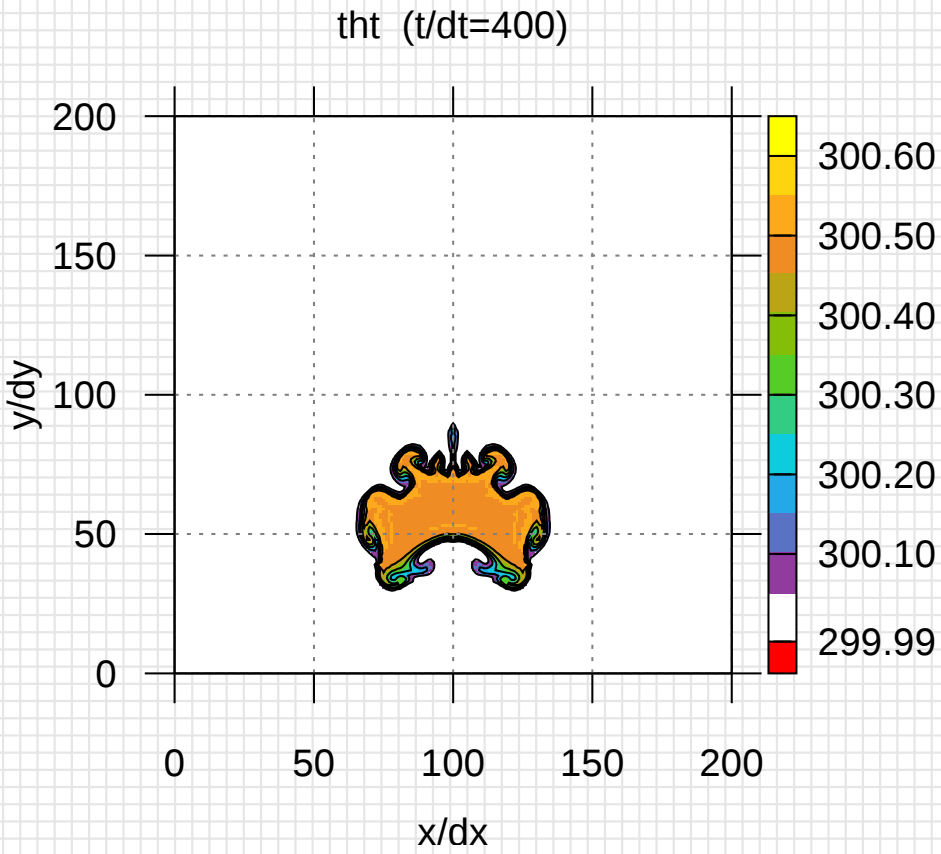
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

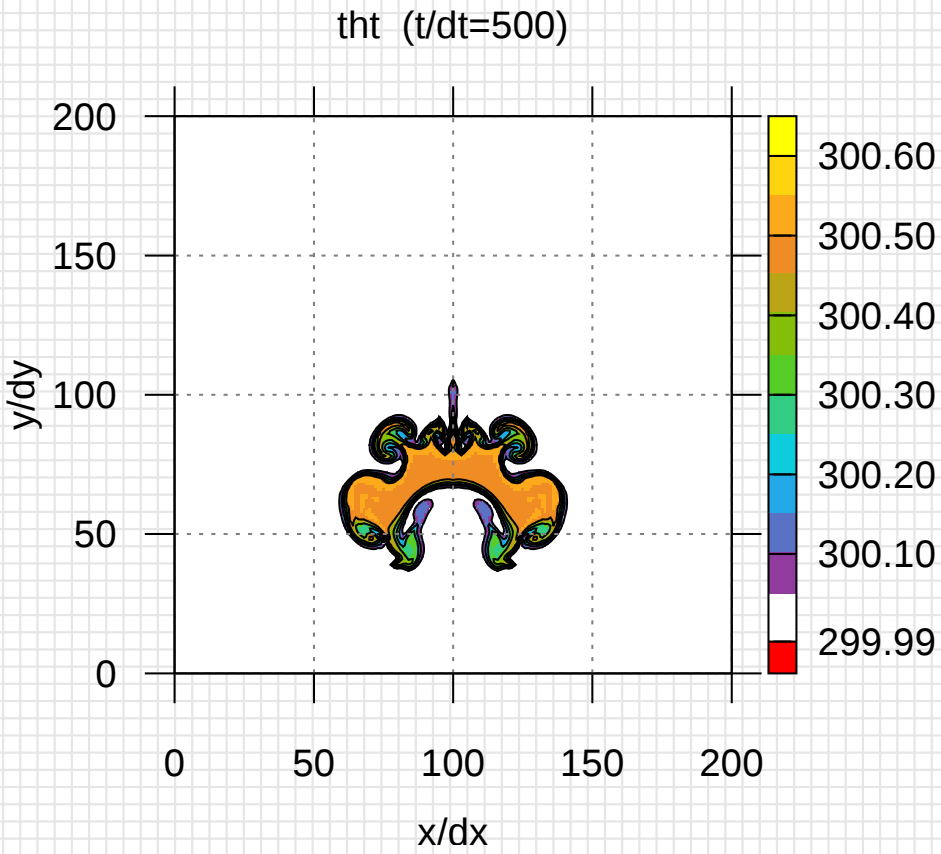
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

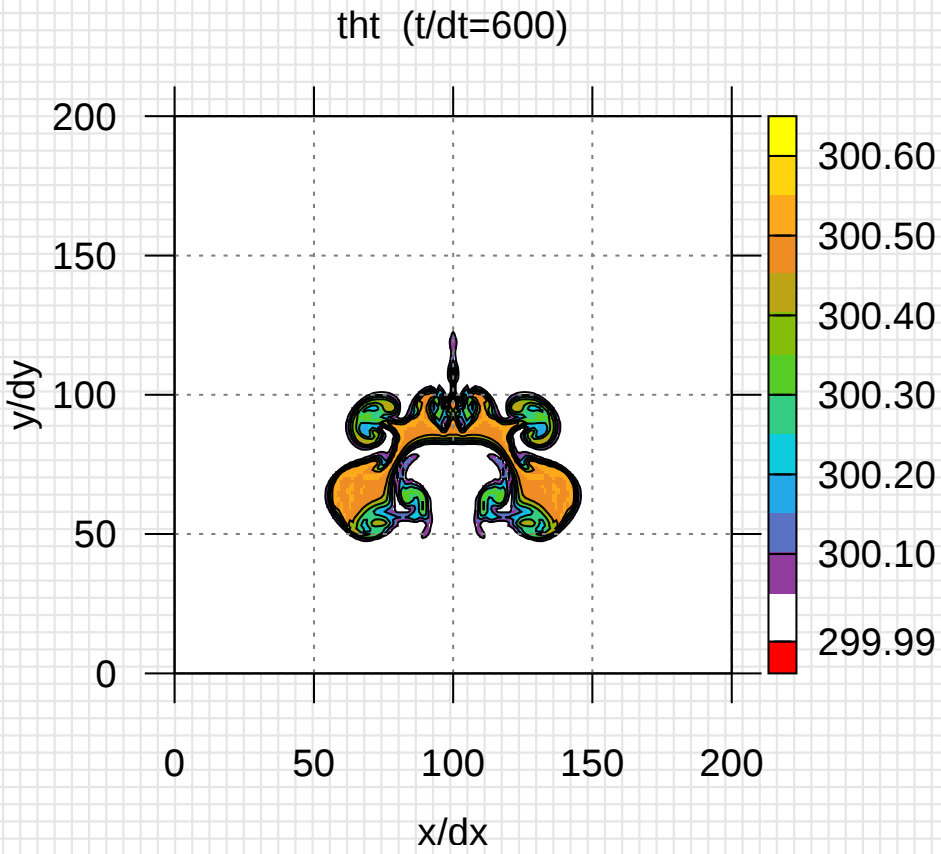
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

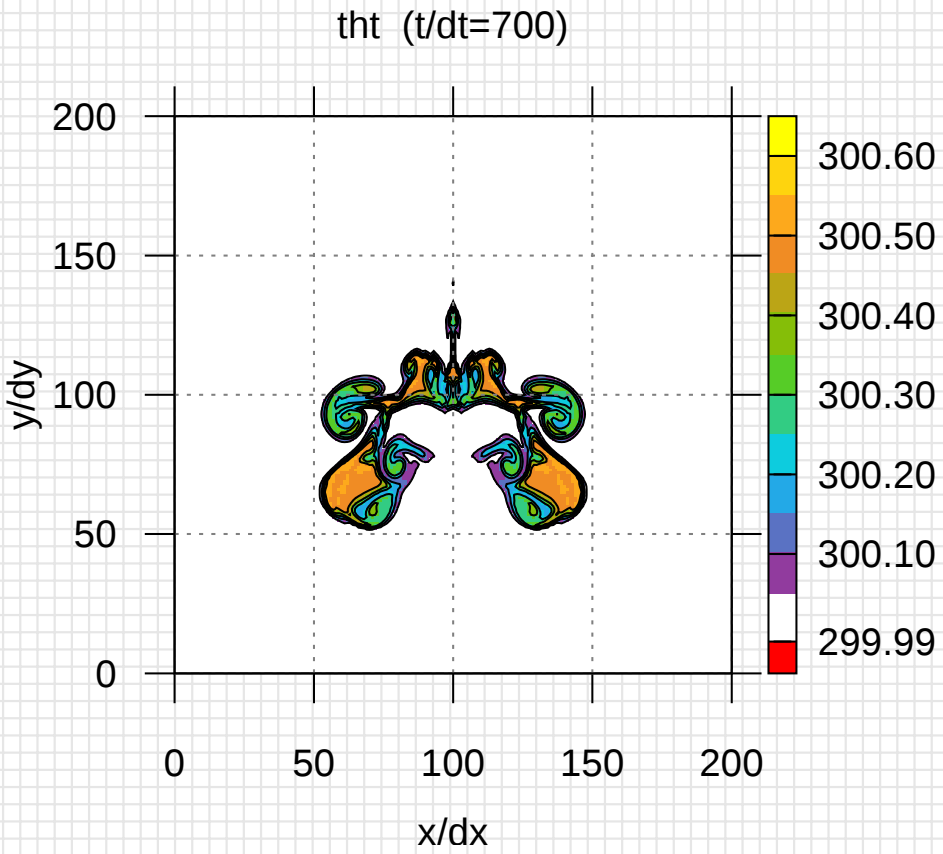
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

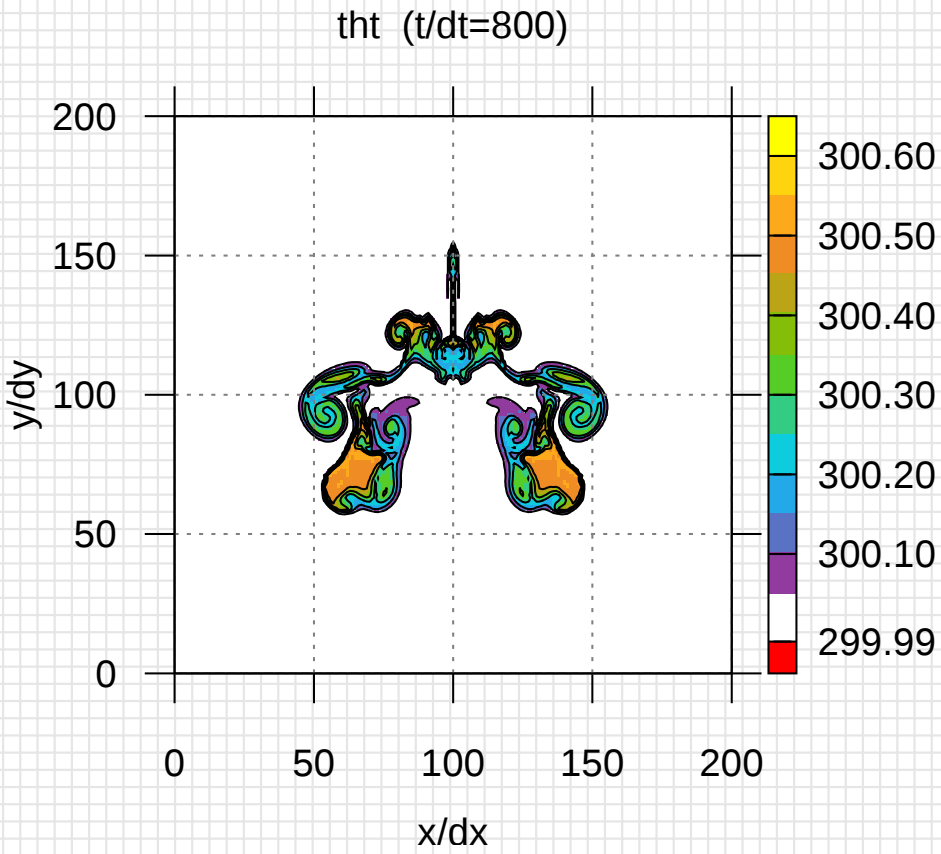
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

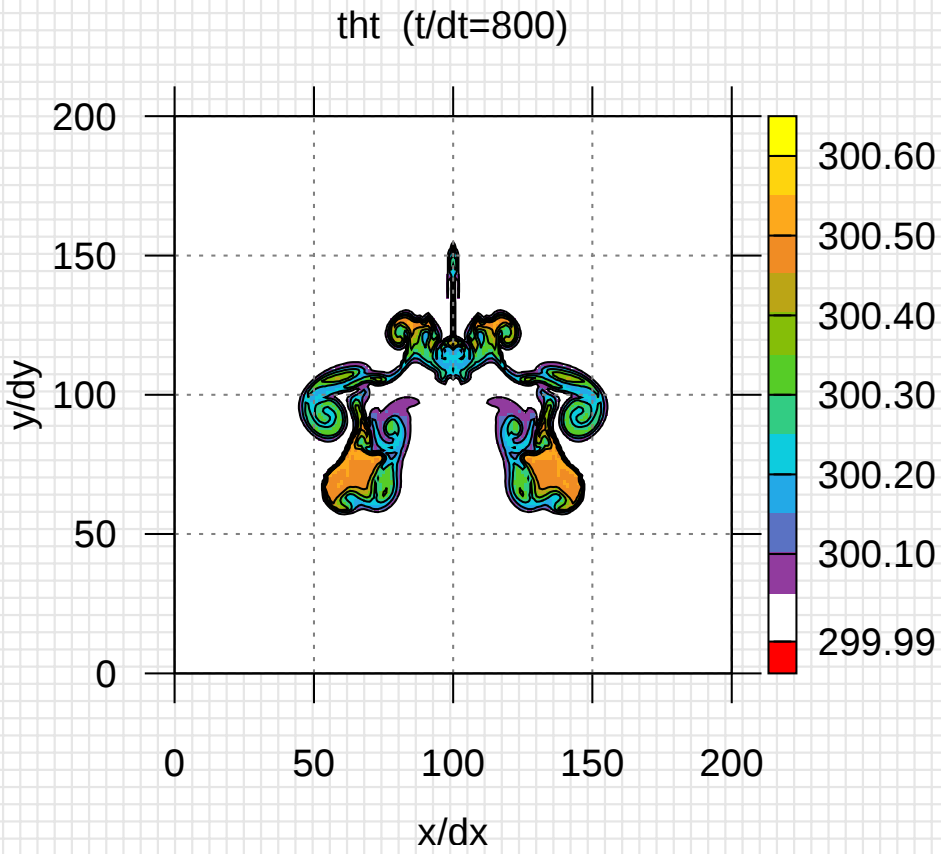
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

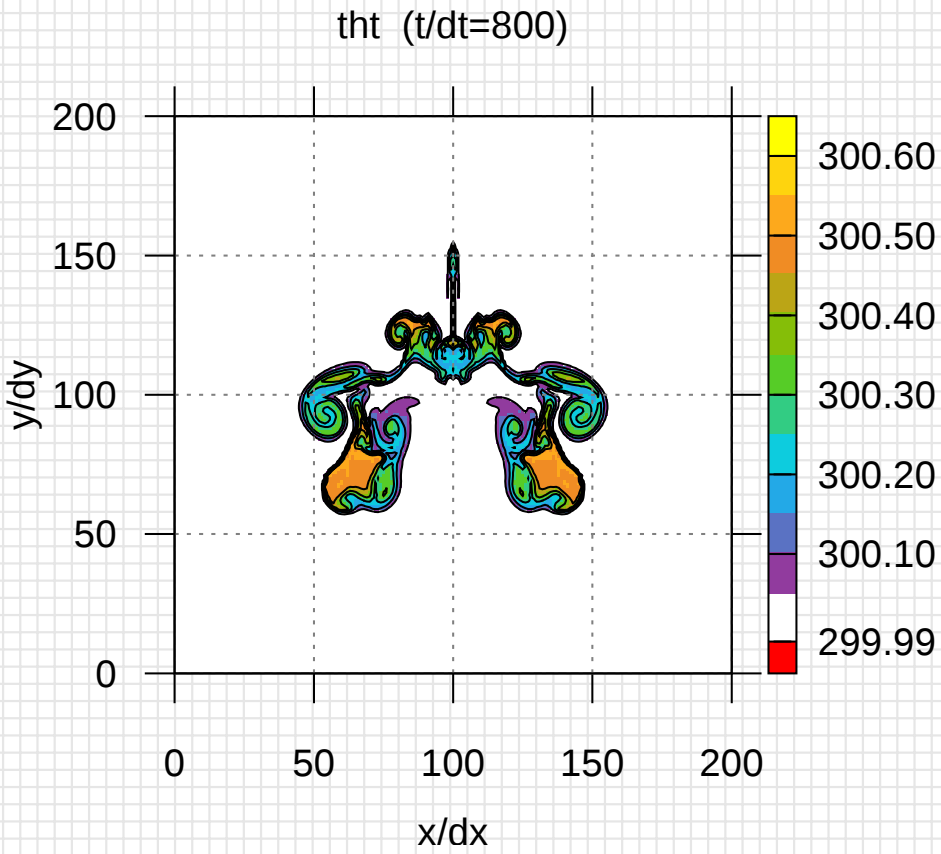
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d

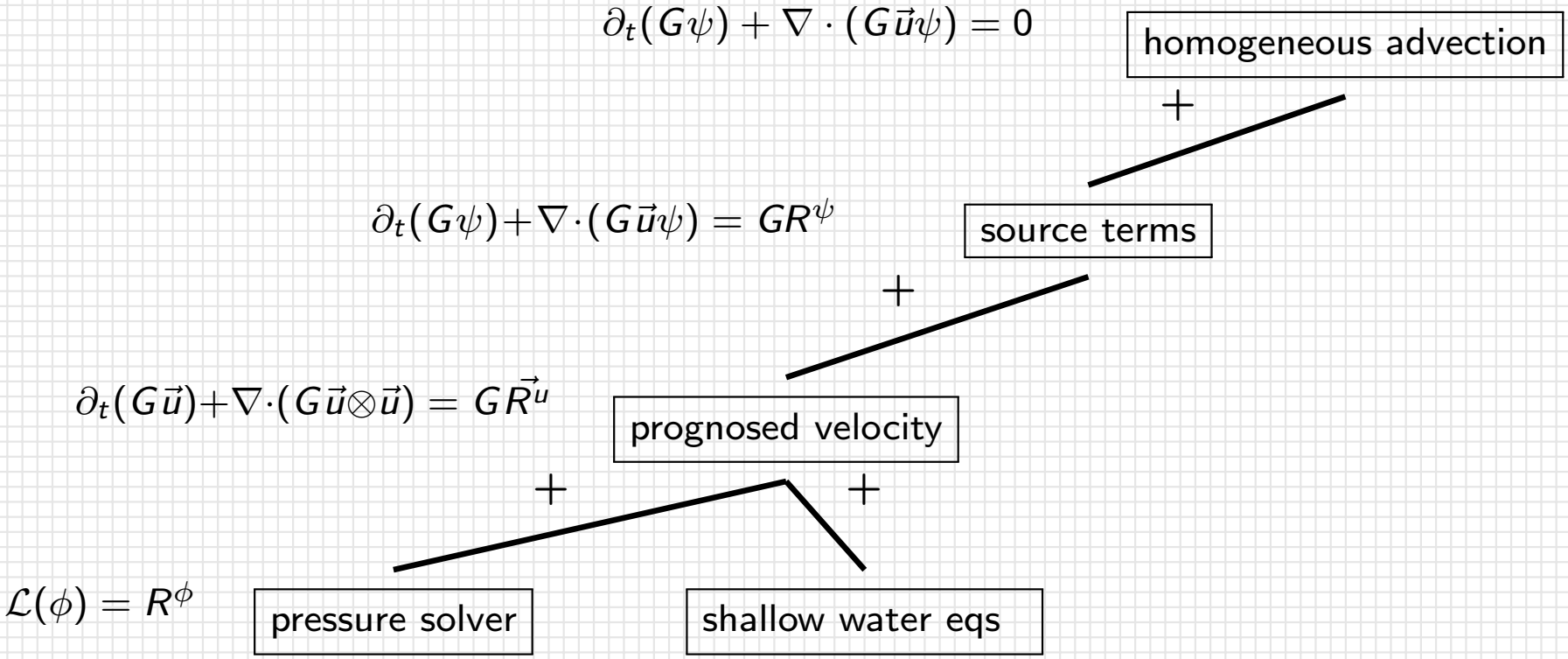
libmpdata++: 2D Boussinesq convection example



- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- <200 lines of code with libmpdata++

`https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d`

libmpdata++ 1.0: solver/algorithm hierarchy



libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - non-oscillatory option
  - third-order terms option
  - infinite gauge option
  - flux-corrected transport option
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - non-oscillatory option
  - infinite gauge option
  - third-order terms option
  - flux-corrected transport option
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - arbitrary number of corrective iterations
  - Flux-Corrected Transport (FCT, non-oscillatory) option
  - infinite-gauge variant (for sing-changing fields)
  - third-order terms
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - arbitrary number of corrective iterations
  - Flux-Corrected Transport (FCT, non-oscillatory) option
  - infinite-gauge variant (for sing-changing fields)
  - third-order terms
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - arbitrary number of corrective iterations
  - Flux-Corrected Transport (FCT, non-oscillatory) option
  - infinite-gauge variant (for sing-changing fields)
  - third-order terms
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - arbitrary number of corrective iterations
  - Flux-Corrected Transport (FCT, non-oscillatory) option
  - infinite-gauge variant (for sing-changing fields)
  - third-order terms
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - arbitrary number of corrective iterations
  - Flux-Corrected Transport (FCT, non-oscillatory) option
  - infinite-gauge variant (for sing-changing fields)
  - third-order terms
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - arbitrary number of corrective iterations
  - Flux-Corrected Transport (FCT, non-oscillatory) option
  - infinite-gauge variant (for sing-changing fields)
  - third-order terms
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - arbitrary number of corrective iterations
  - Flux-Corrected Transport (FCT, non-oscillatory) option
  - infinite-gauge variant (for sing-changing fields)
  - third-order terms
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - arbitrary number of corrective iterations
  - Flux-Corrected Transport (FCT, non-oscillatory) option
  - infinite-gauge variant (for sing-changing fields)
  - third-order terms
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

libmpdata++ 1.0: summary

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:
  - arbitrary number of corrective iterations
  - Flux-Corrected Transport (FCT, non-oscillatory) option
  - infinite-gauge variant (for sing-changing fields)
  - third-order terms
- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

Plan of the talk

1. what's libmpdata++
2. libmpdata++ 1.0: summary of features
3. libmpdata++ 2.0: new features under development
4. libmpdata++: a hello-world program
5. closing remarks

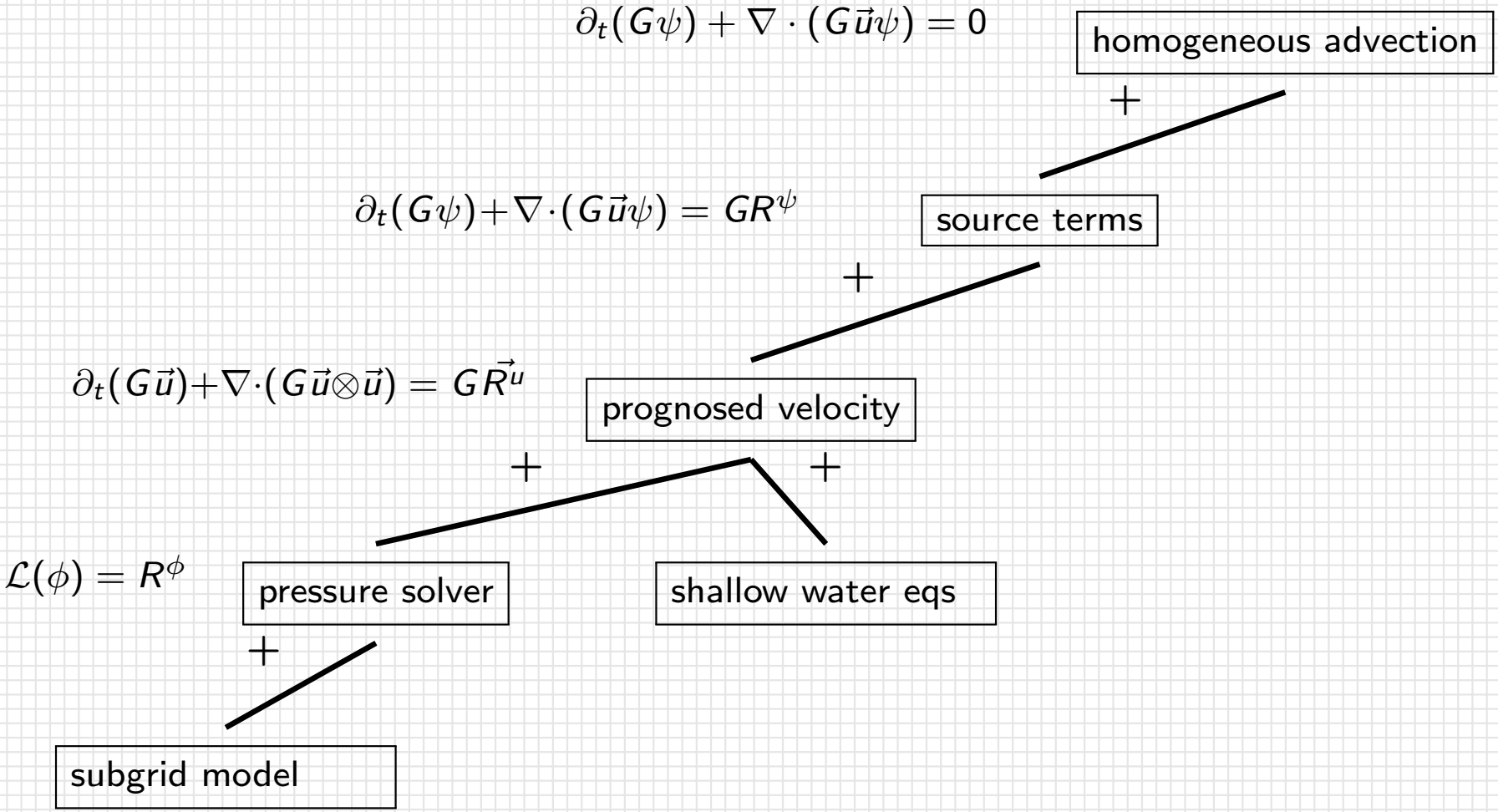
Plan of the talk

libmpdata++ 2.0: summary of features under development

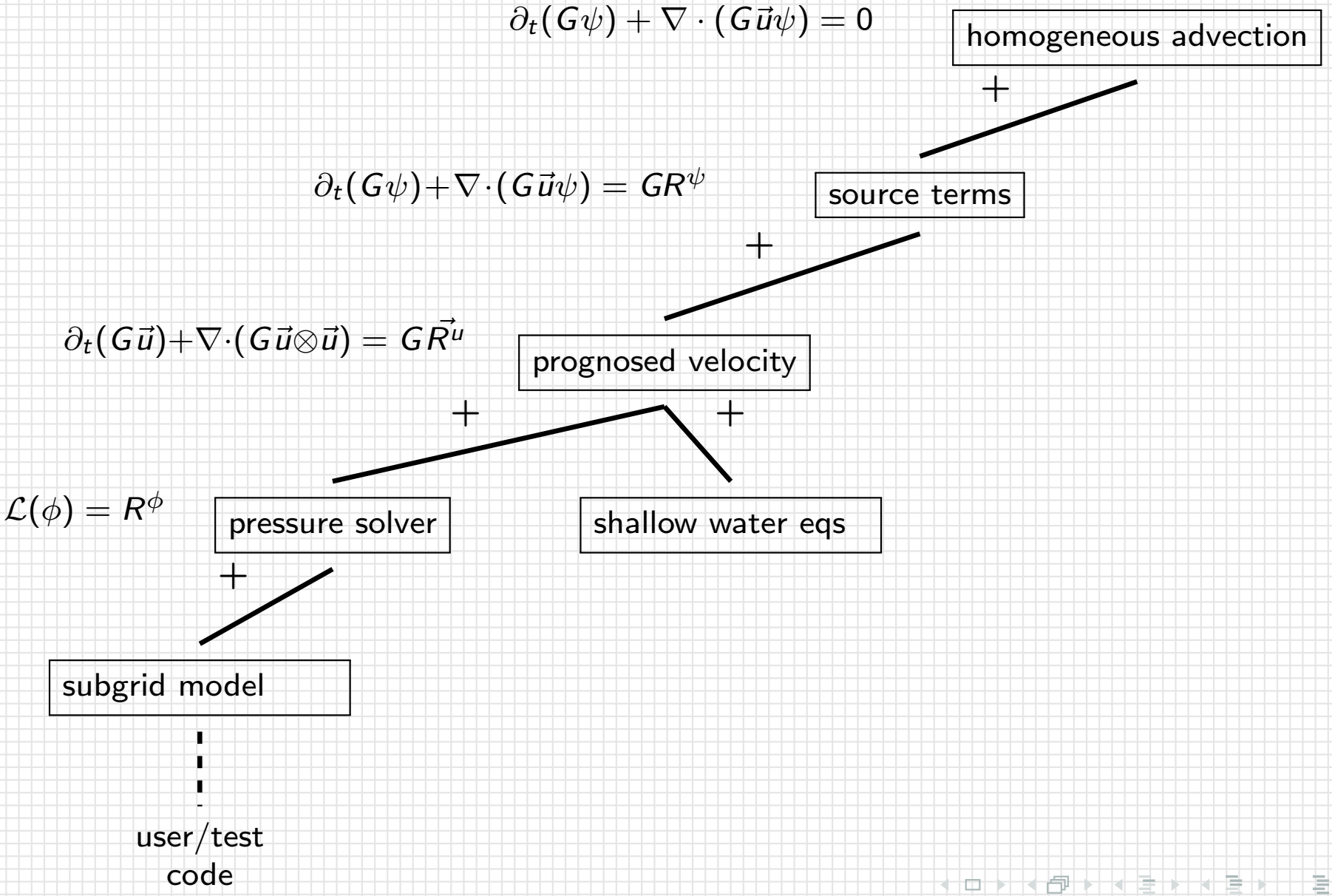
libmpdata++ 2.0: summary of features under development

- higher-order operators
  (for DNS/iLES simulations)
- implicit treatment of absorbers
  (for immersed-boundary method)
- adaptive timestepping
- distributed-memory parallelisation
  (using Boost.MPI & HDF5/MPI-IO)

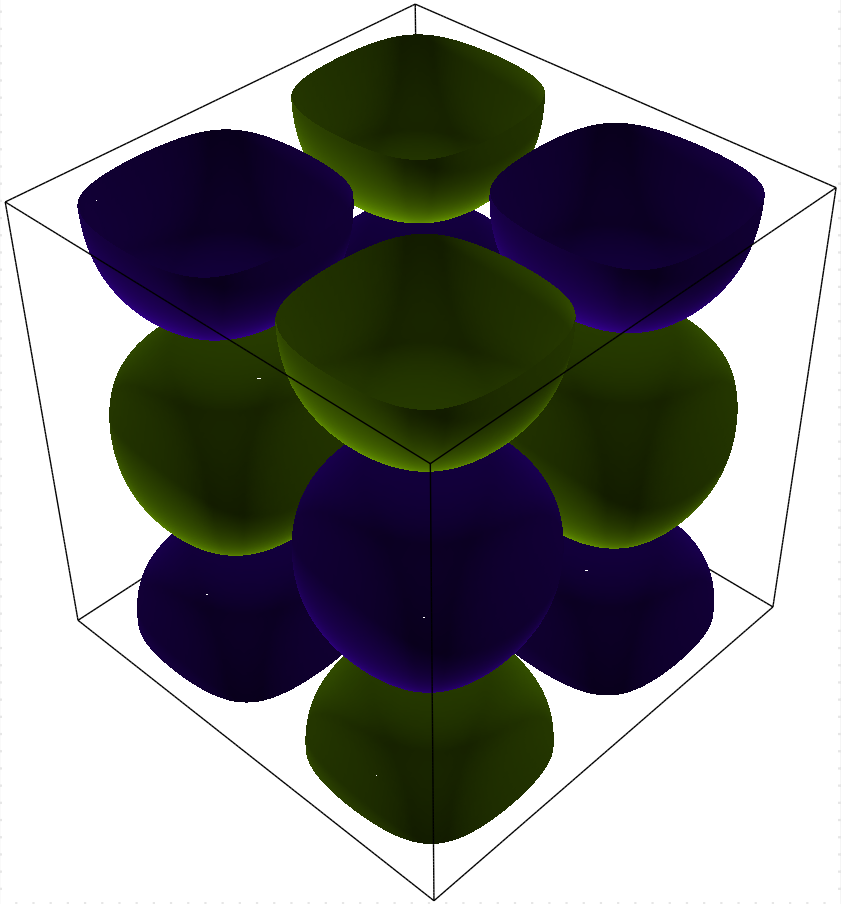
libmpdata++ 2.0: solver/algorithm hierarchy



libmpdata++ 2.0: solver/algorithm hierarchy



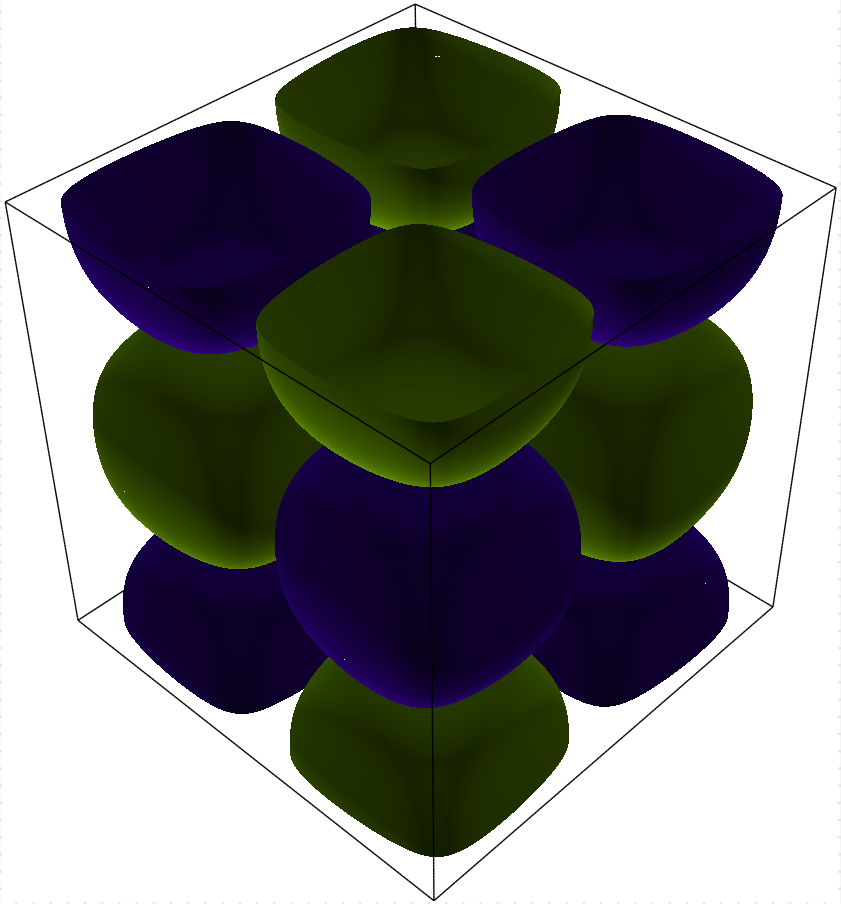
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

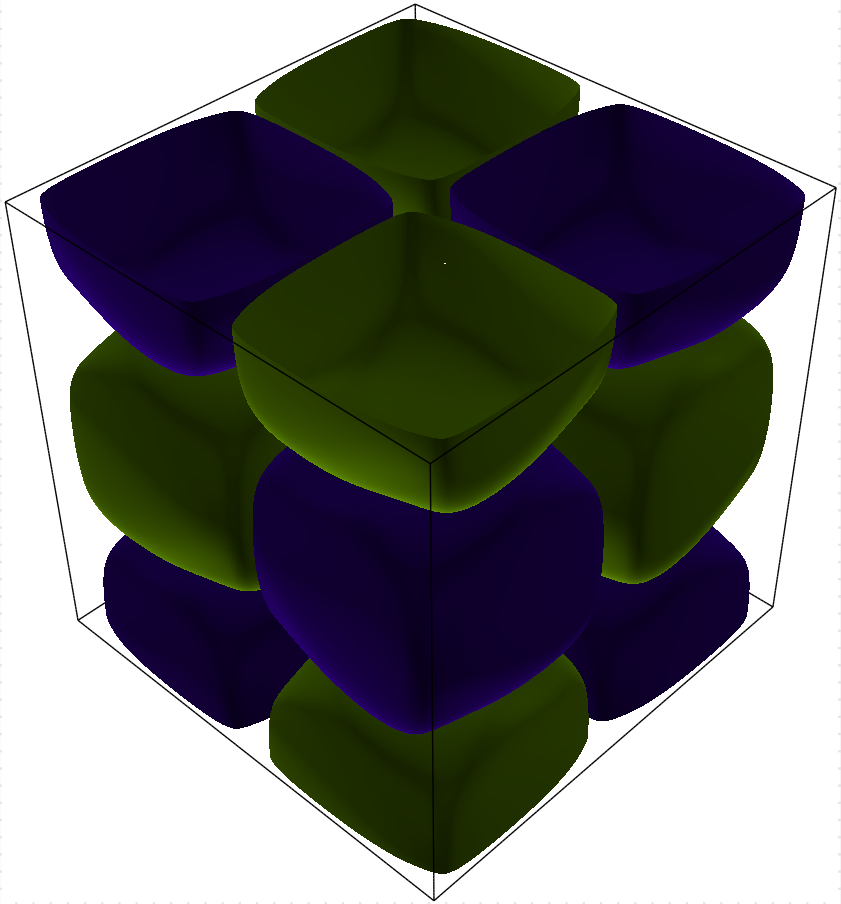
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

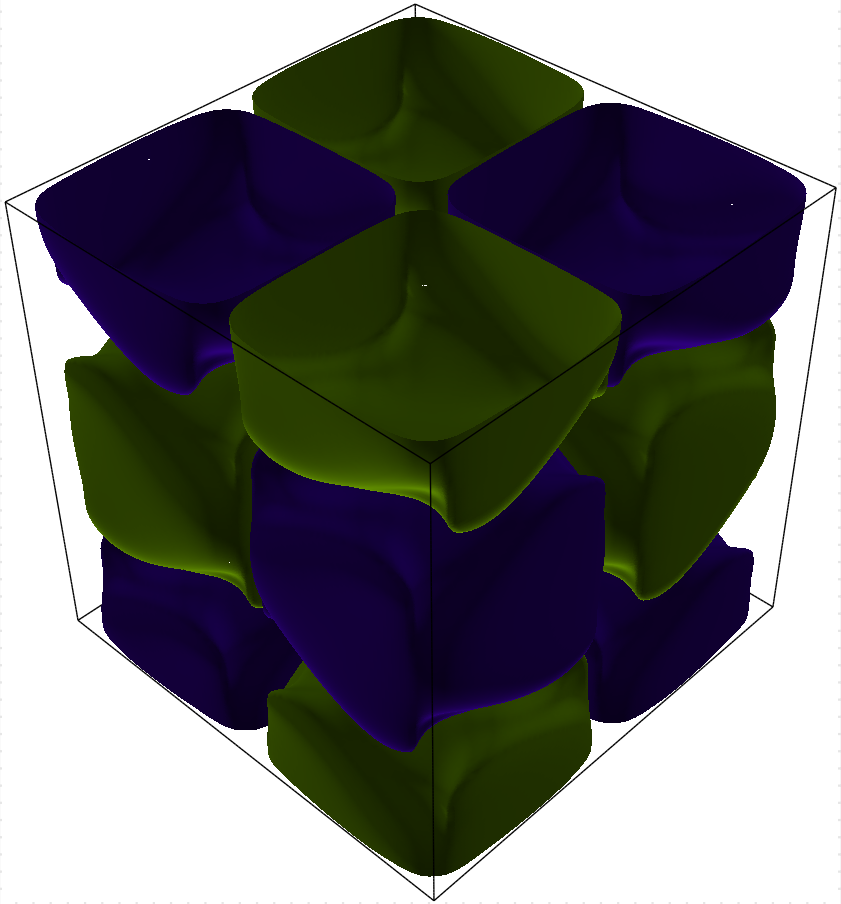
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

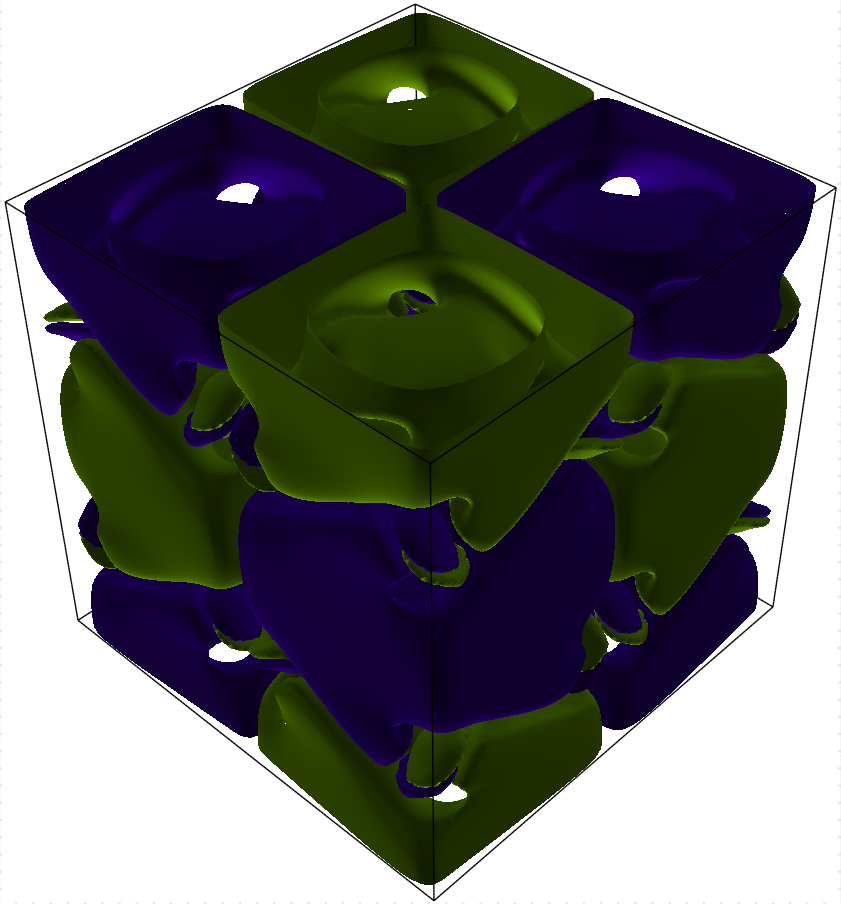
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

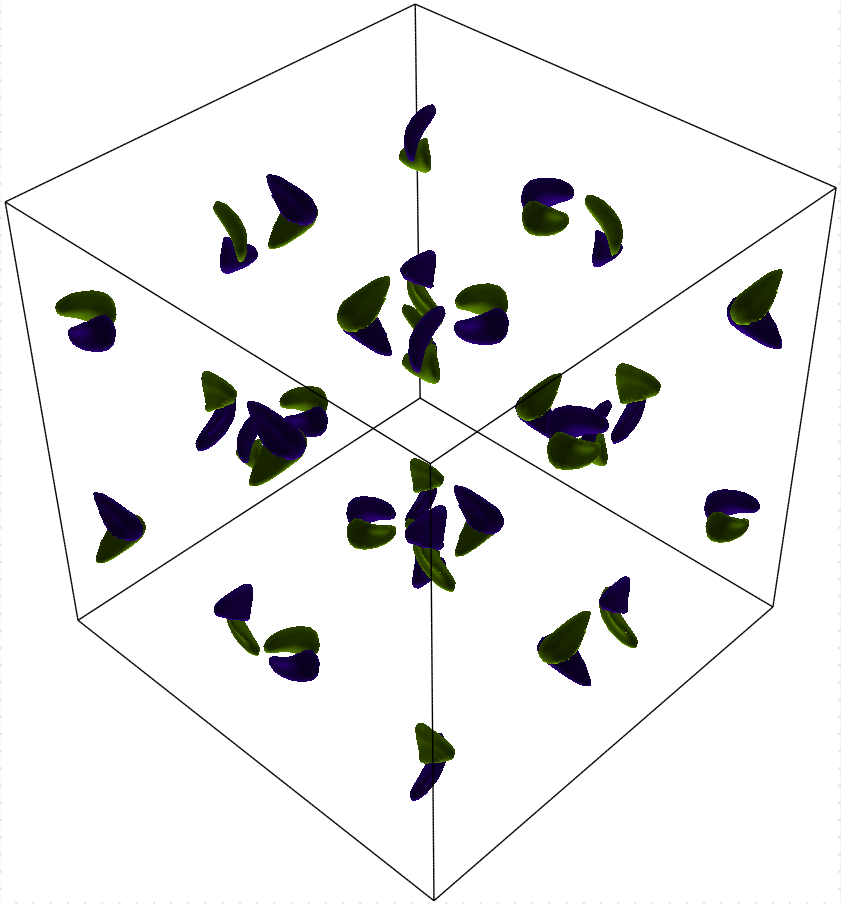
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

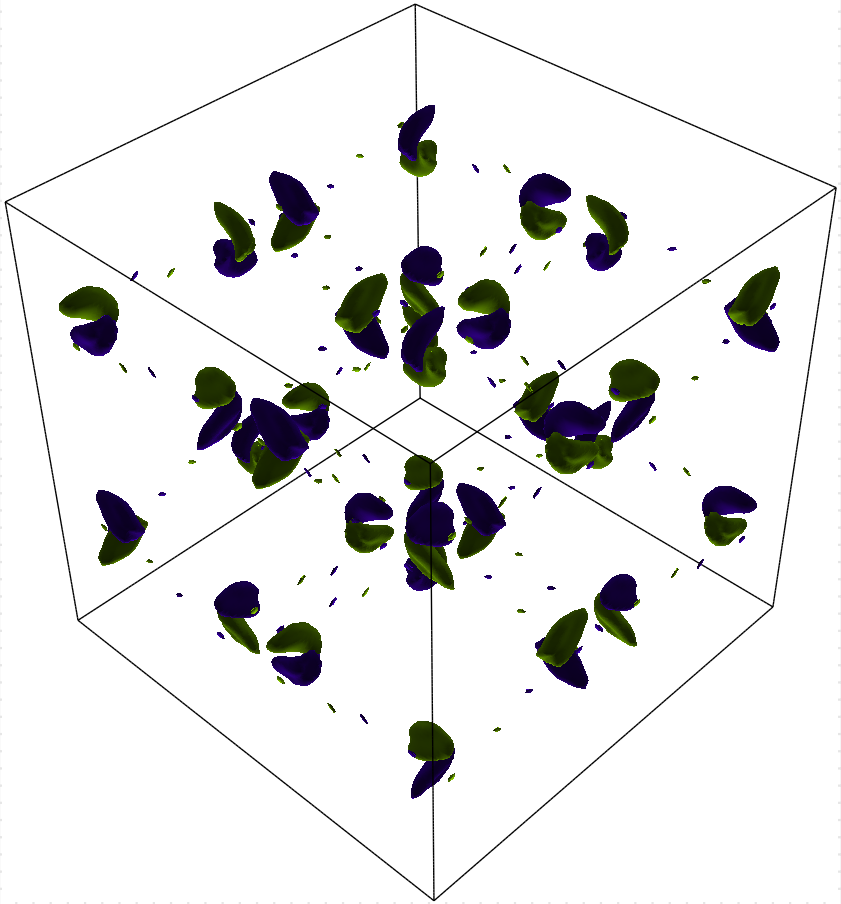
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

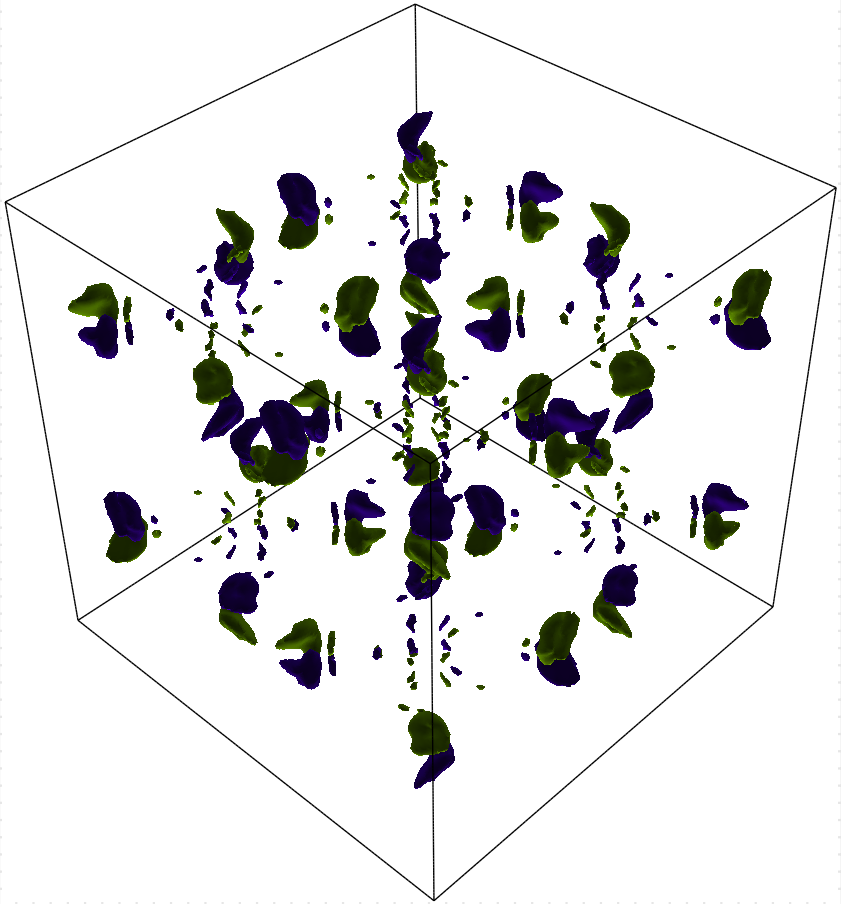
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

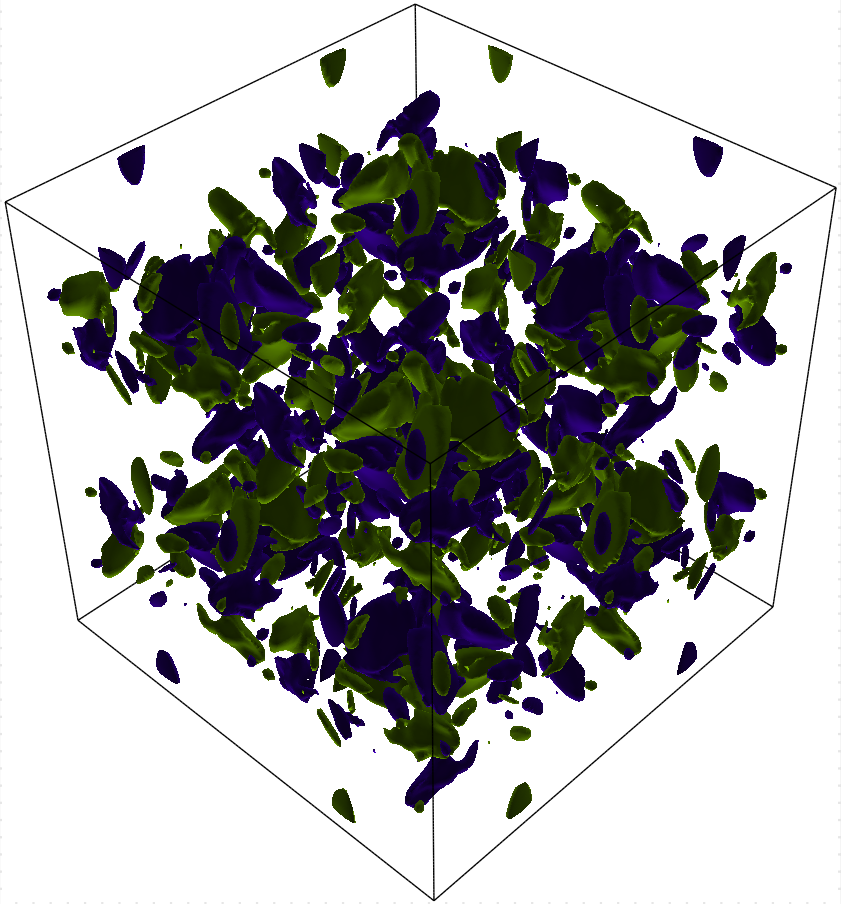
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

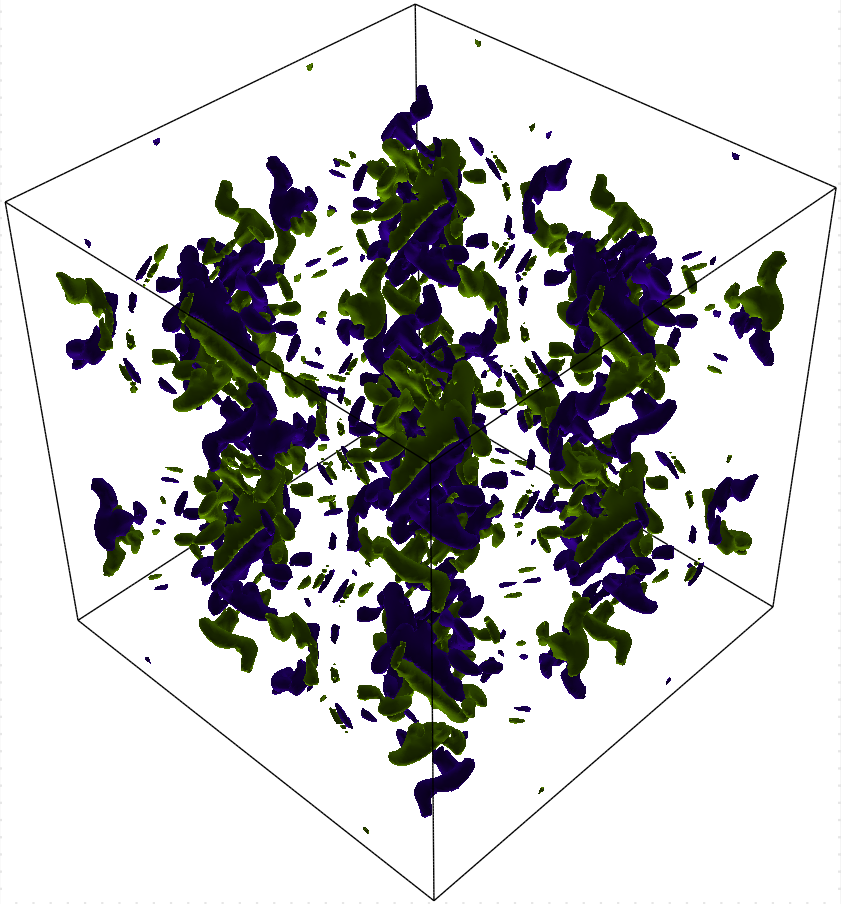
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

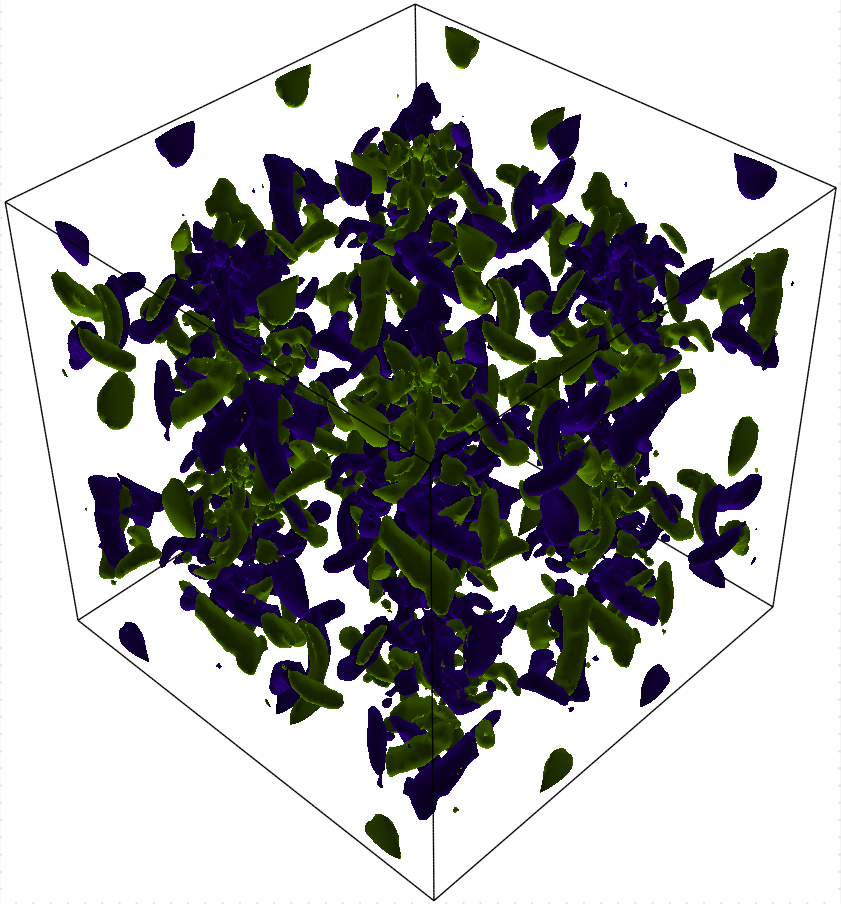
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

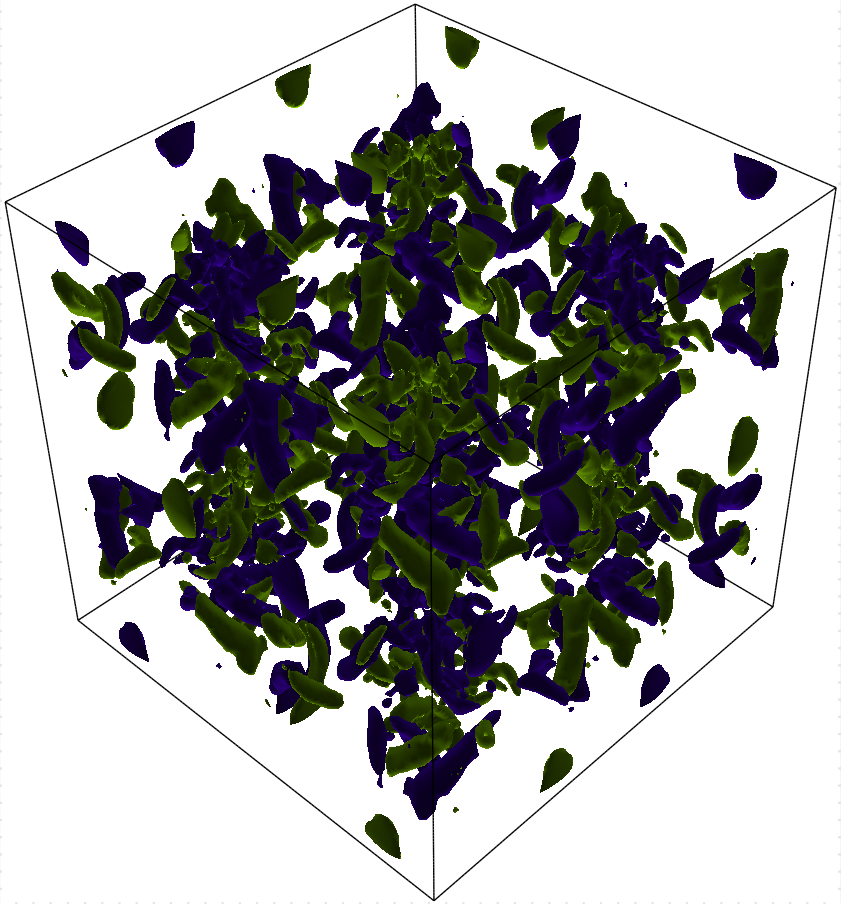
libmpdata++ 2.0: Taylor-Green vortex example



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

https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence

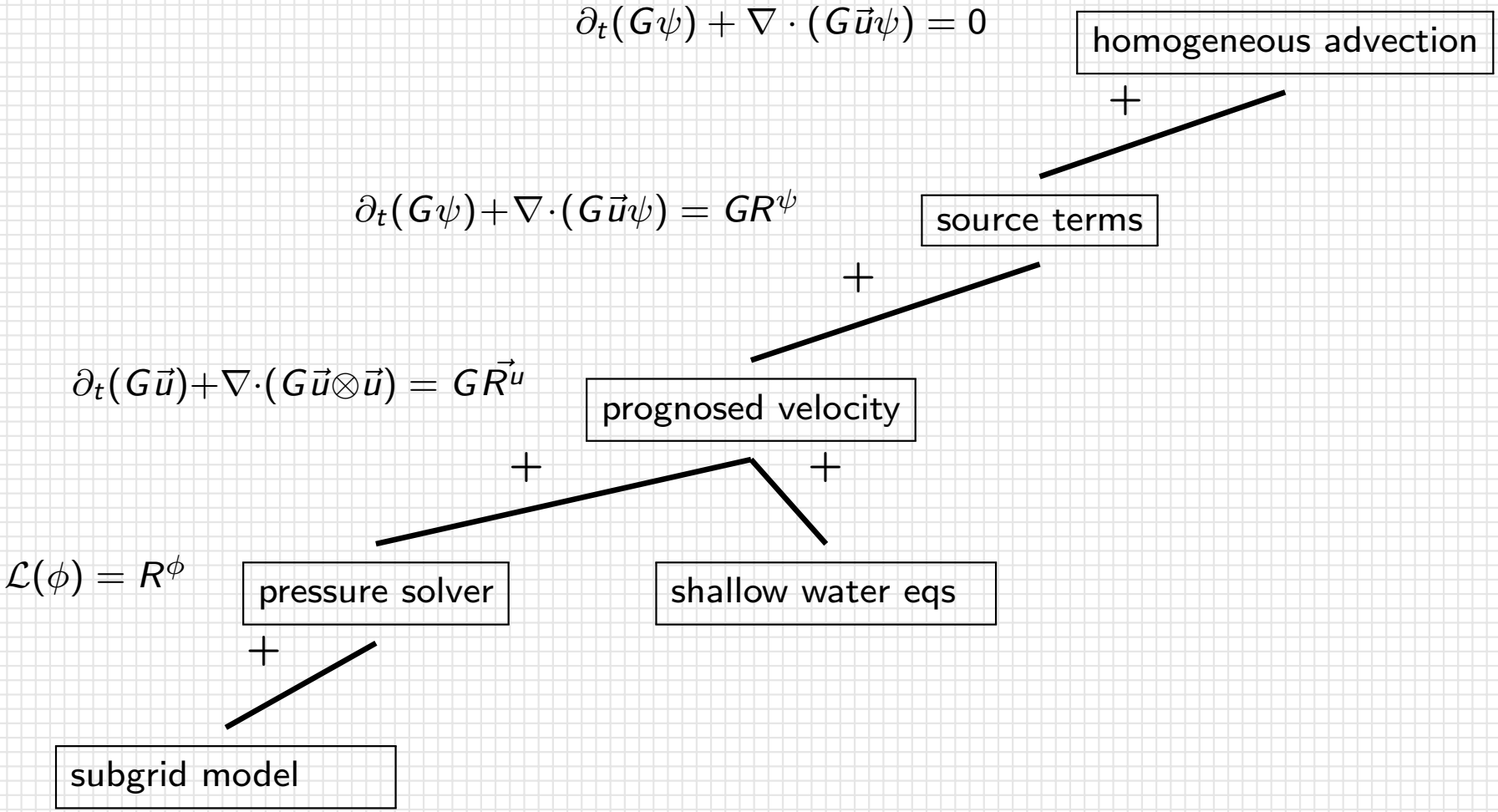
libmpdata++ 2.0: Taylor-Green vortex example



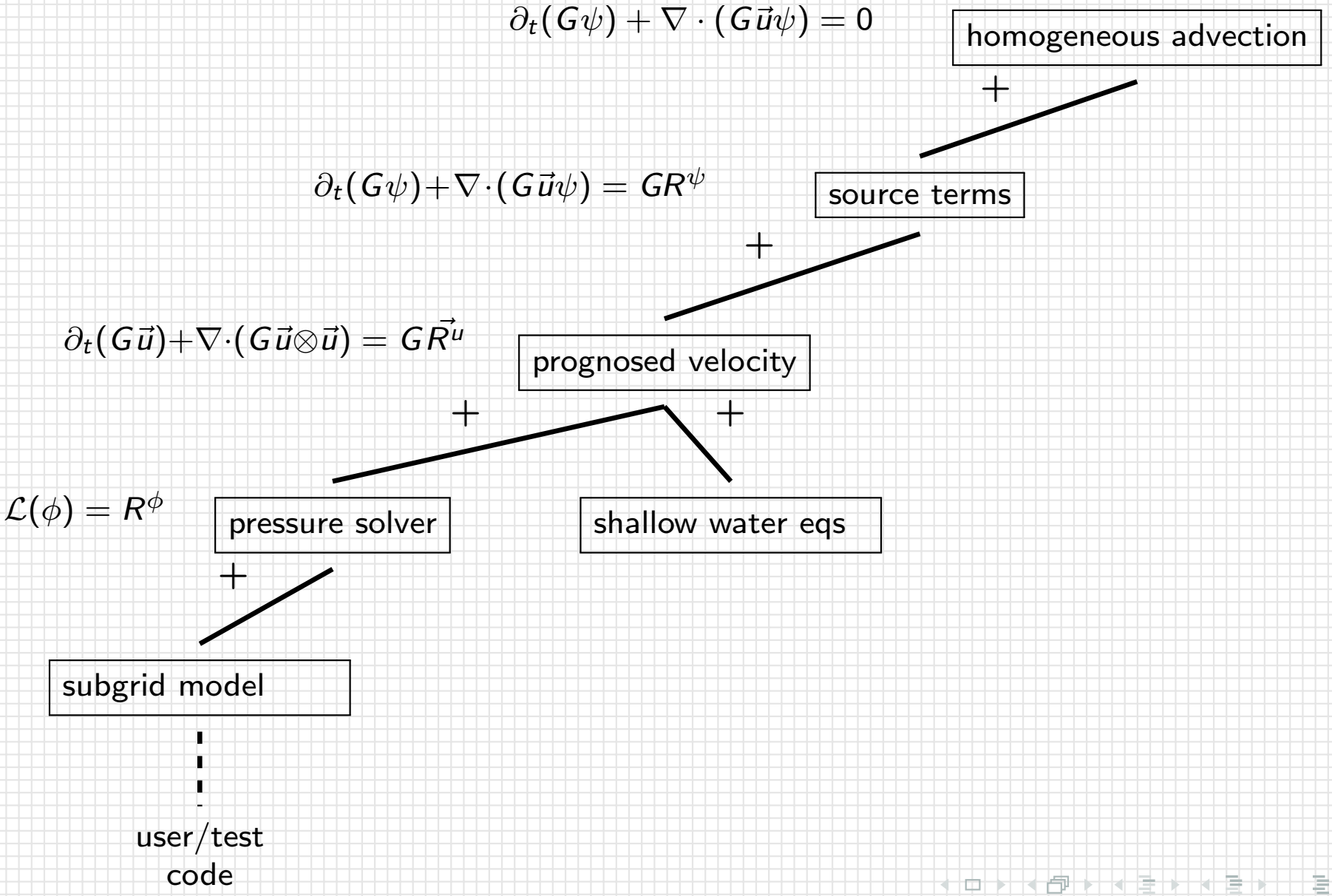
- inspired by spectral calculations of Brachet et al., 1983
- <100 lines of code with libmpdata++

`https://github.com/igfuw/libmpdataxx/tree/master/tests/sandbox/turbulence`

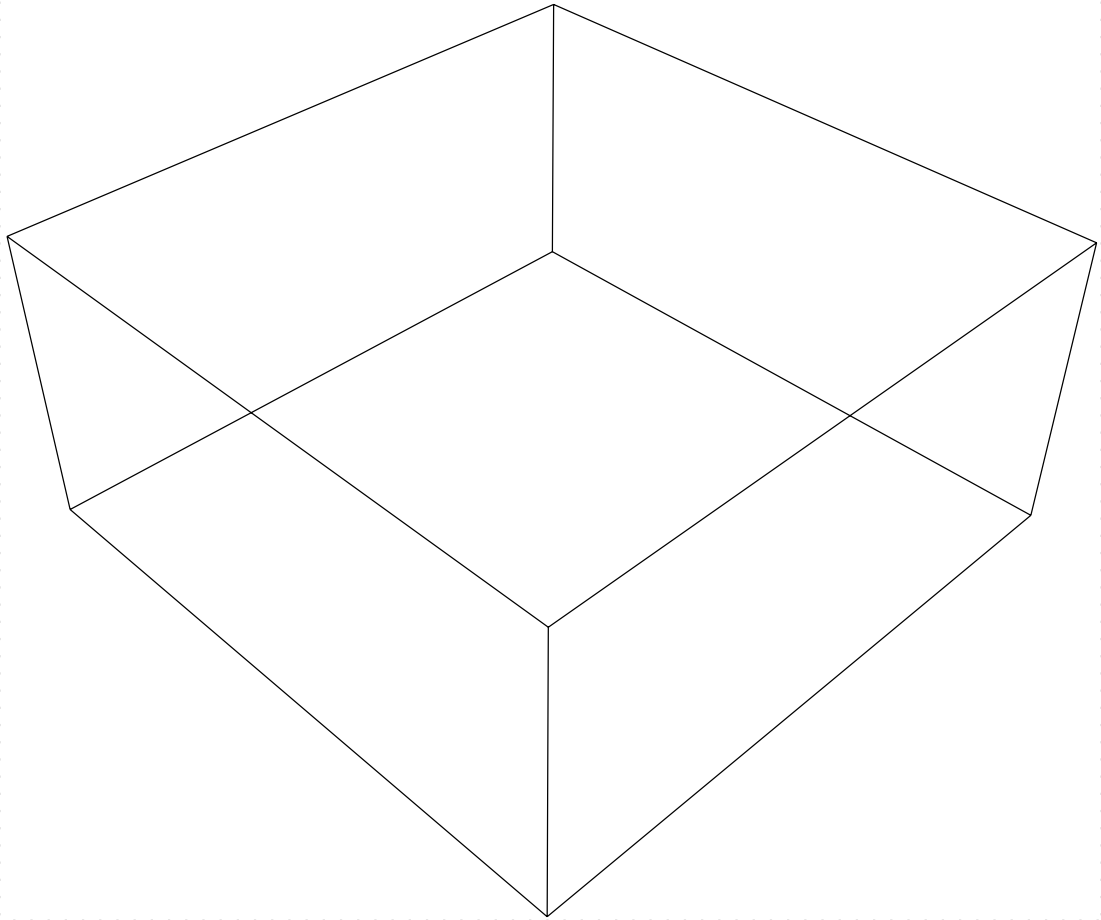
libmpdata++ 2.0: solver/algorithm hierarchy



libmpdata++ 2.0: solver/algorithm hierarchy

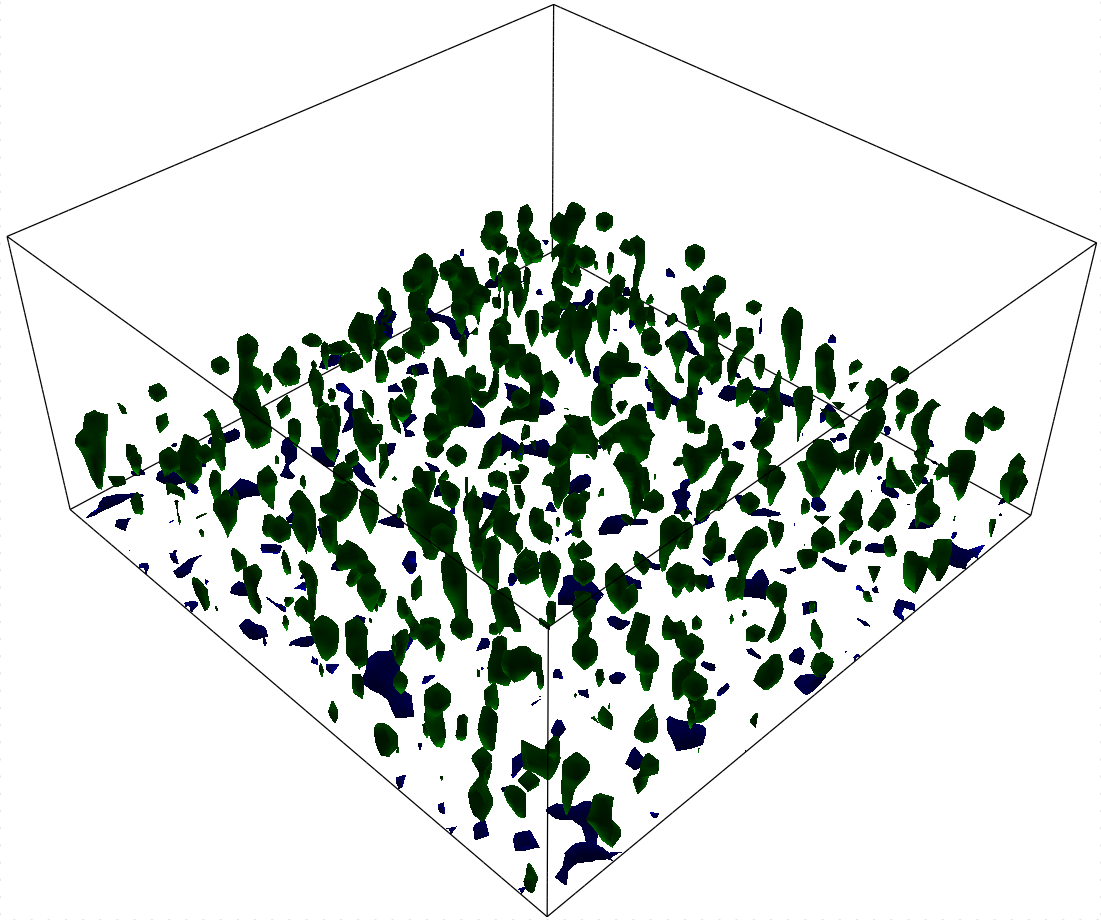


libmpdata++ 2.0: convective boundary layer example



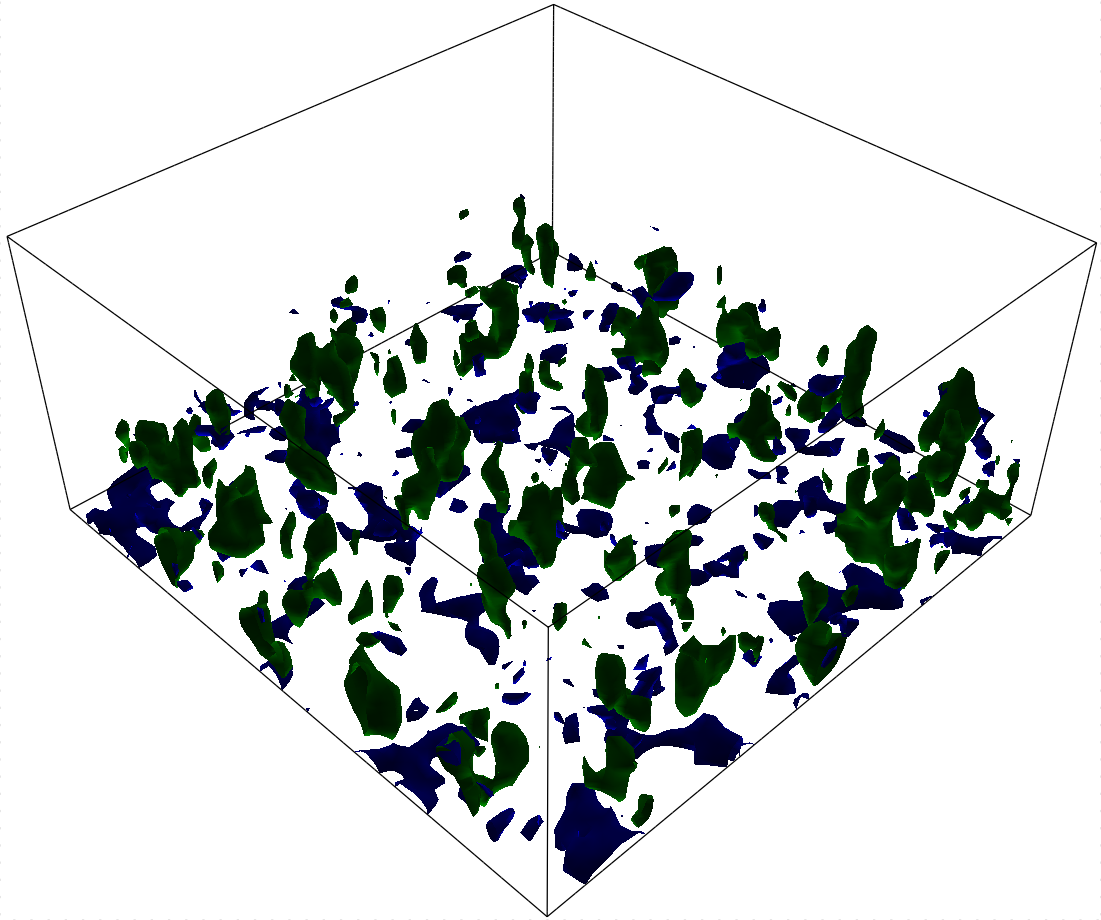
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



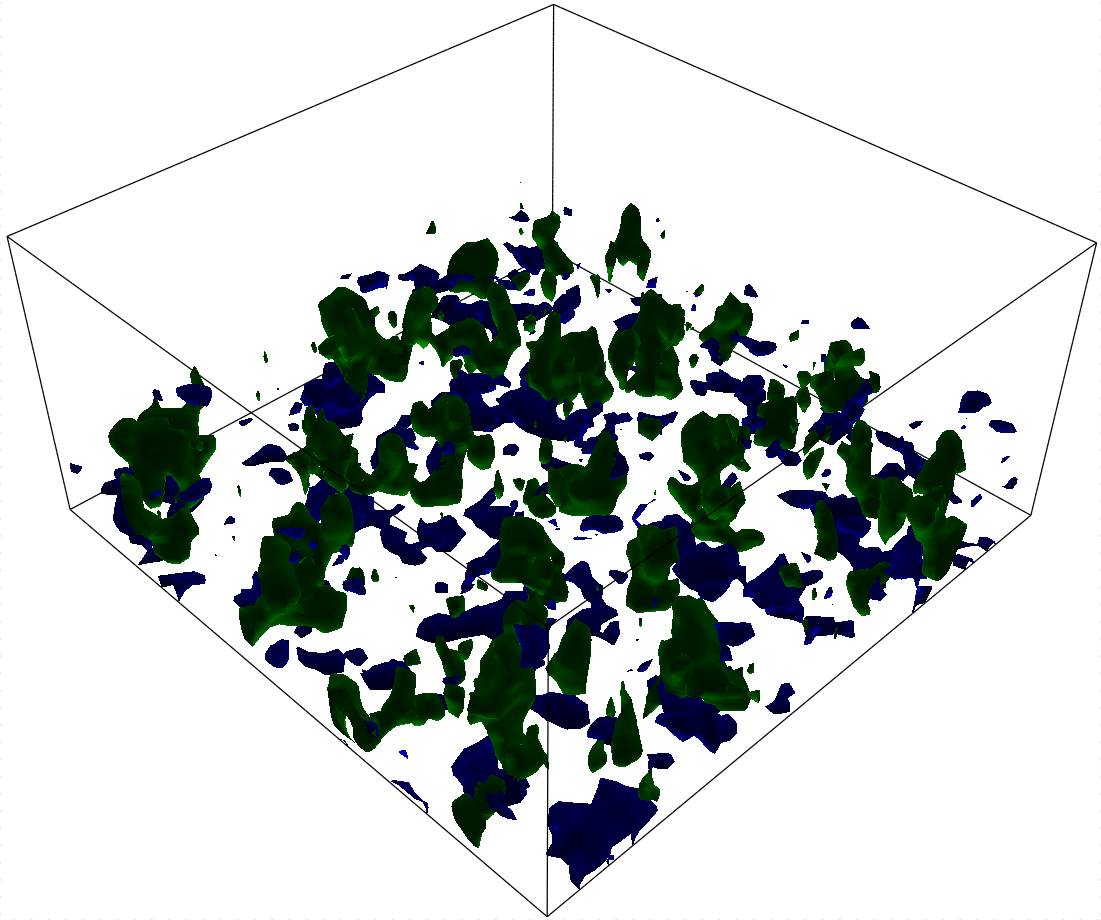
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



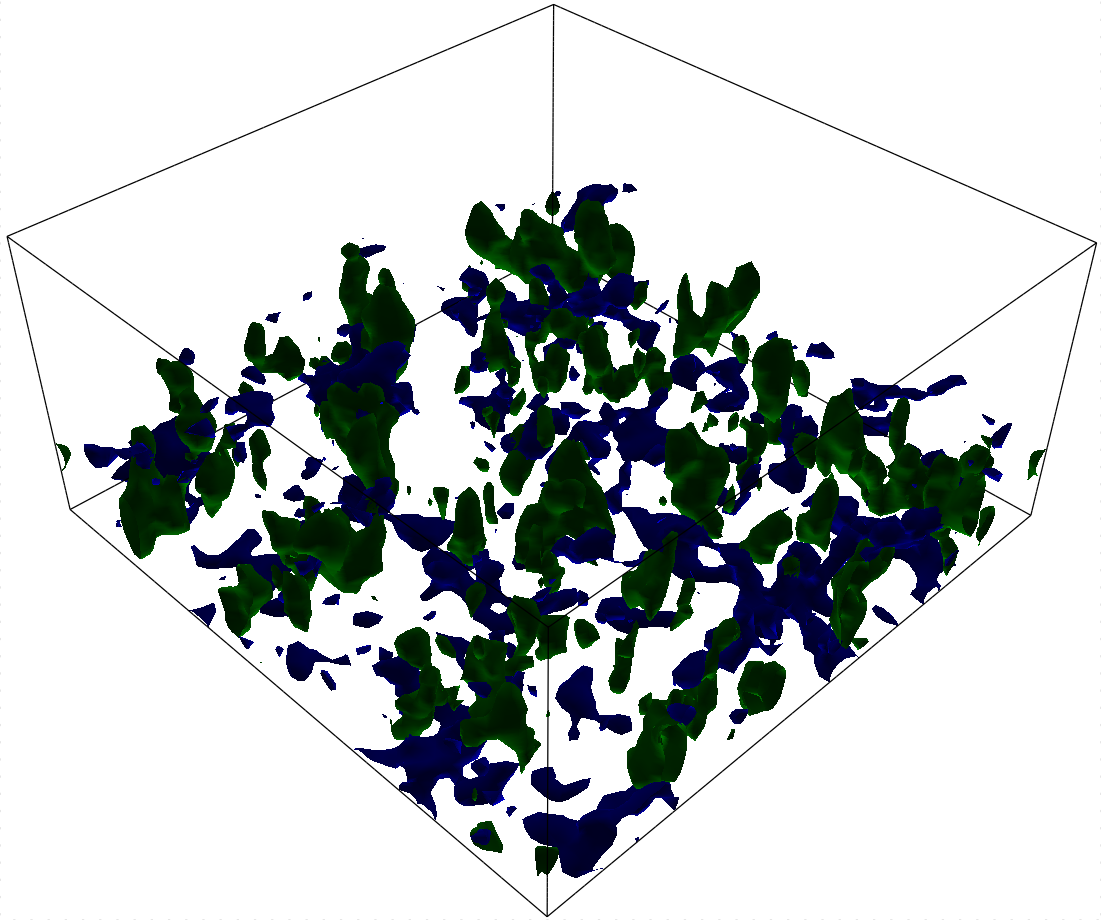
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



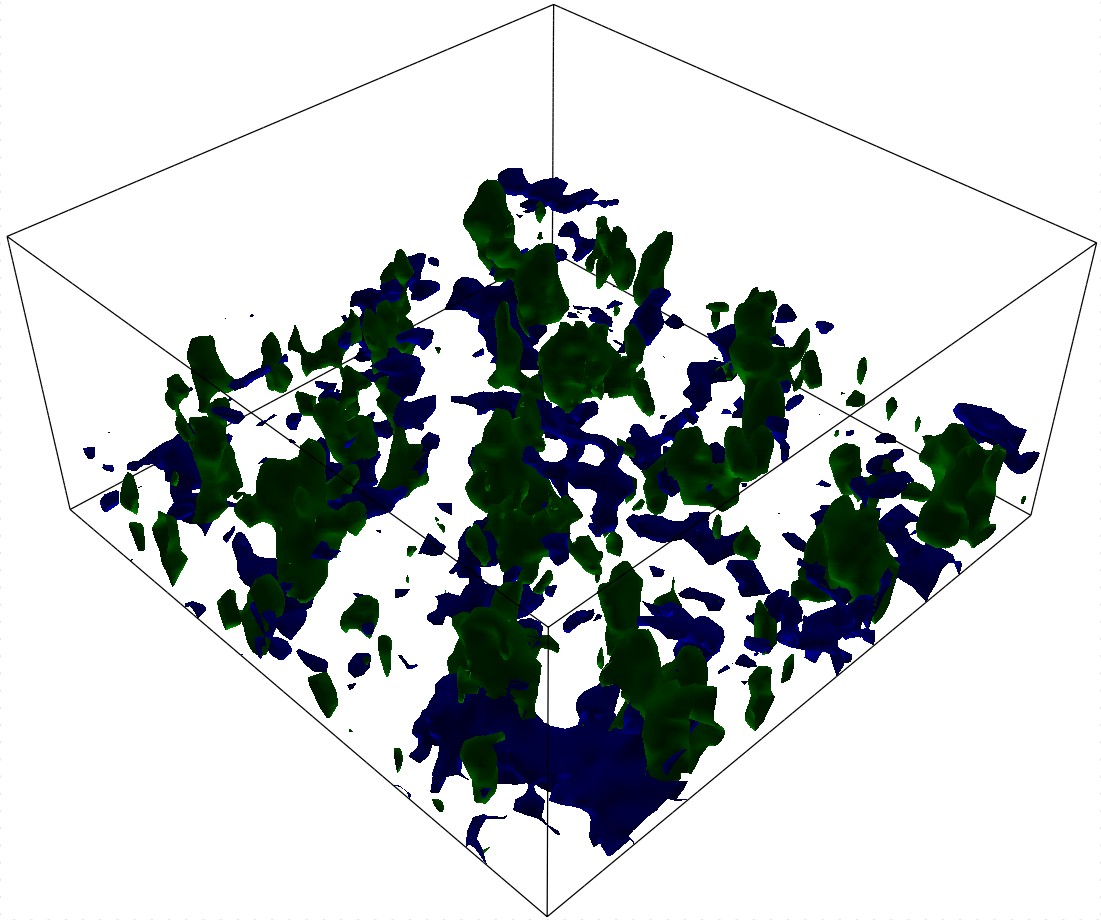
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



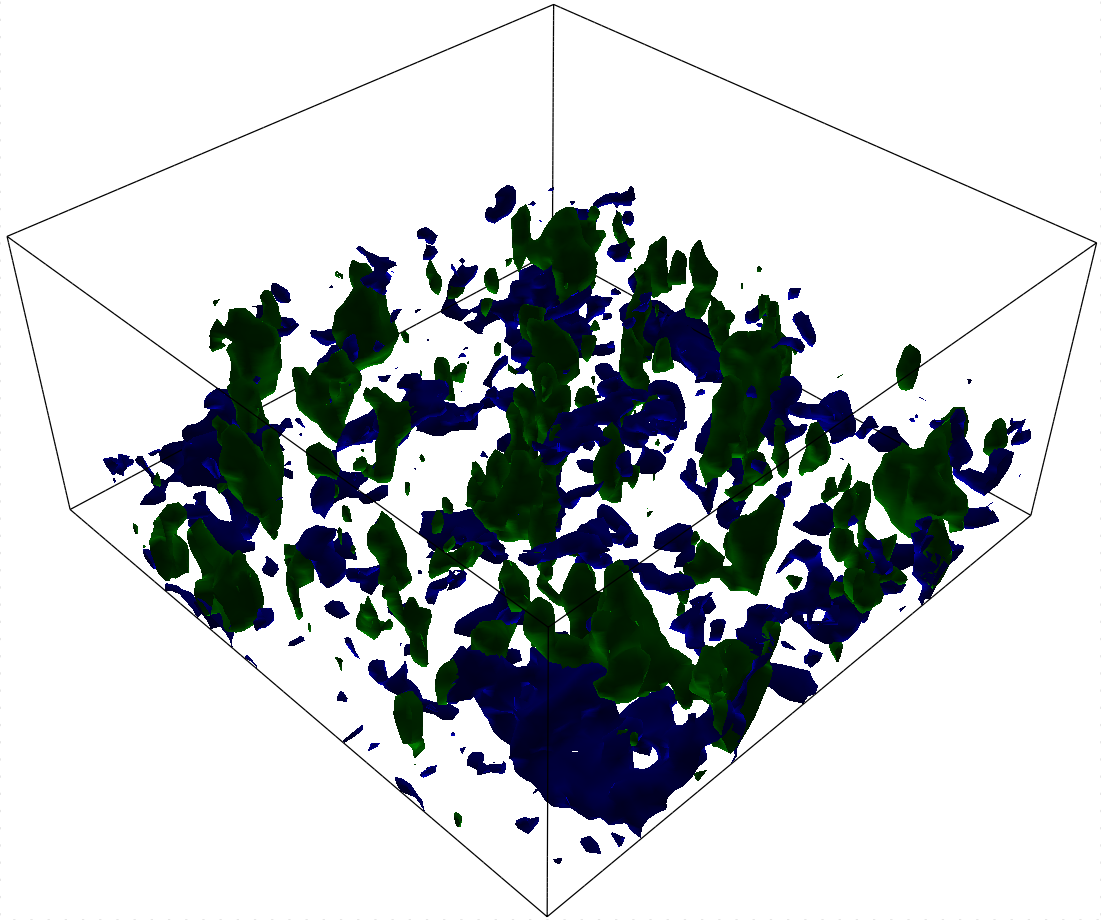
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



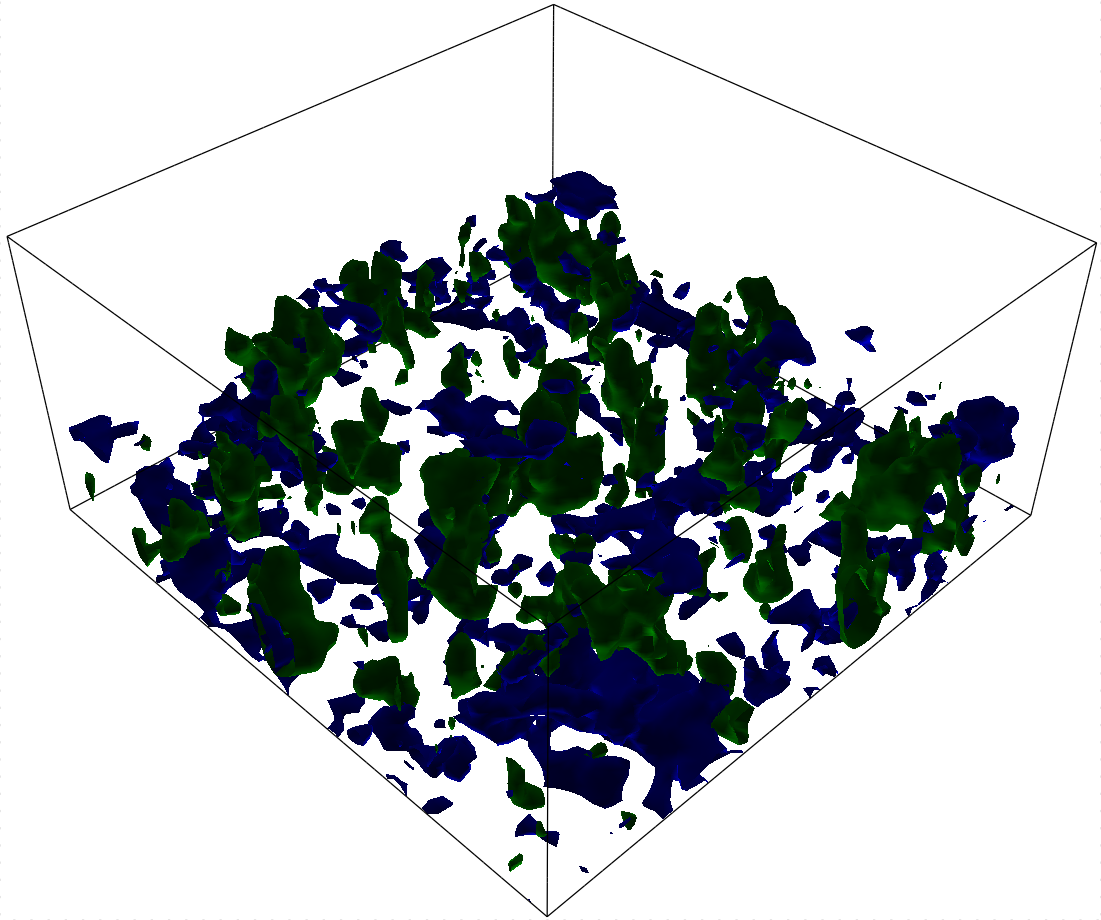
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



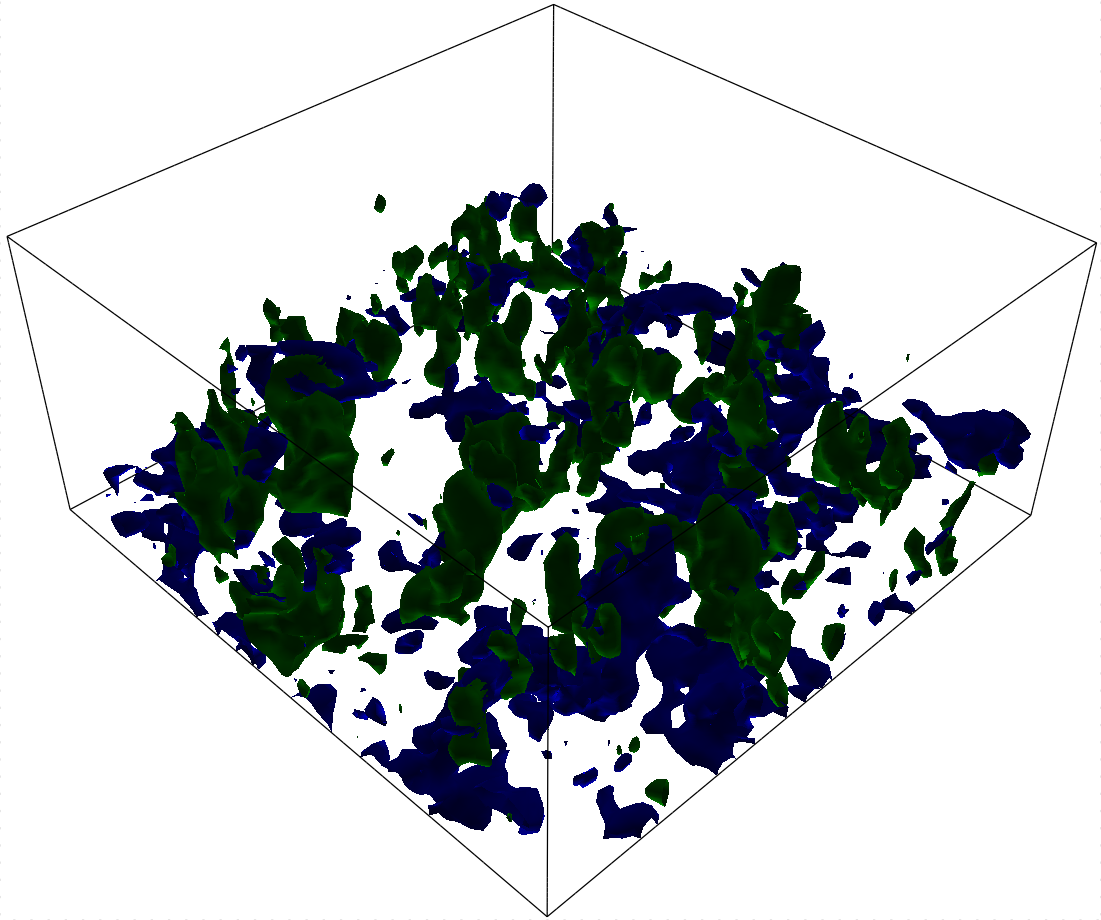
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



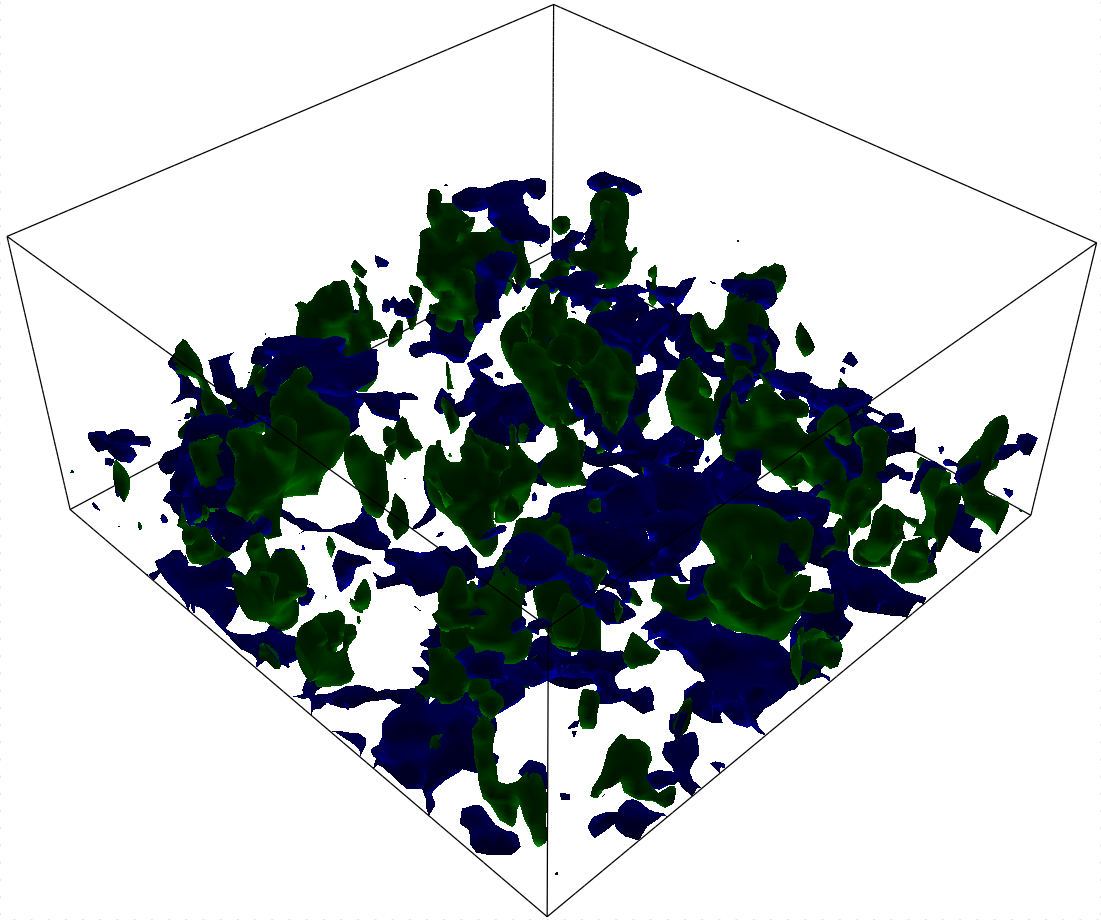
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



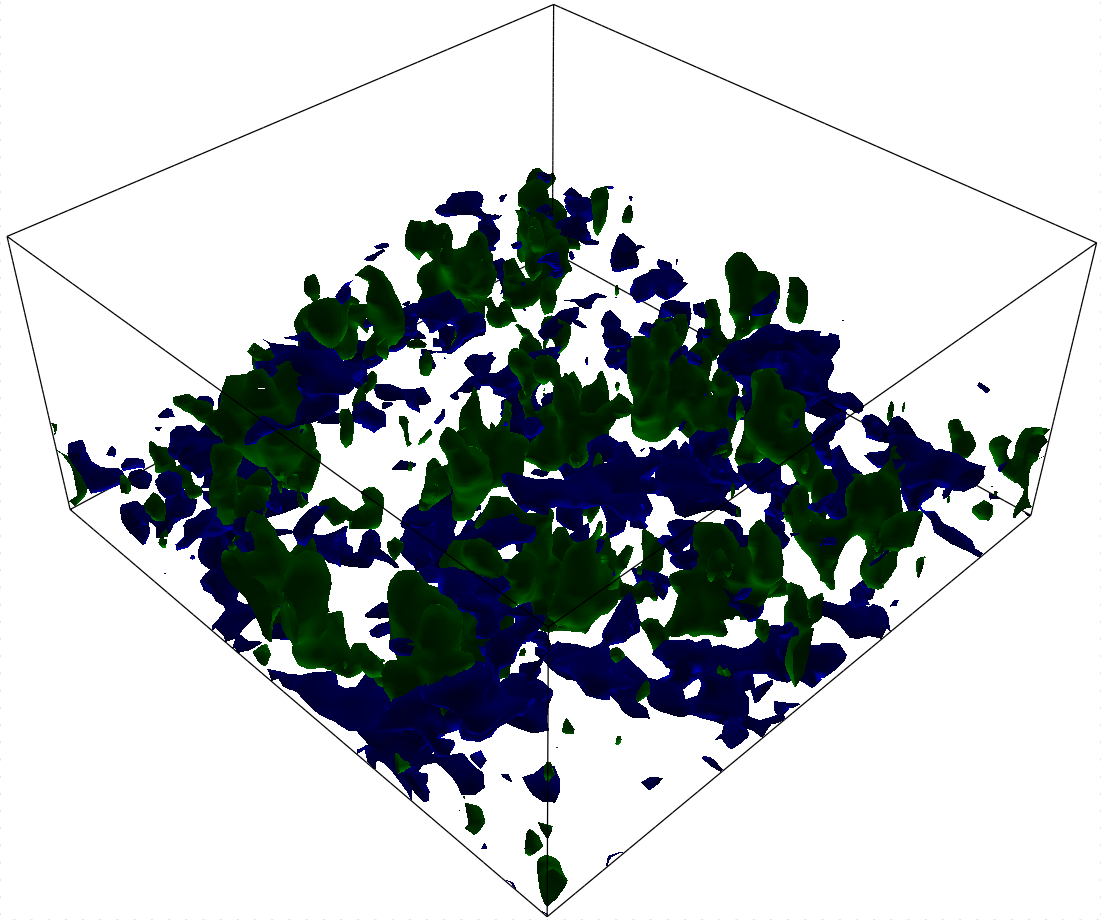
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



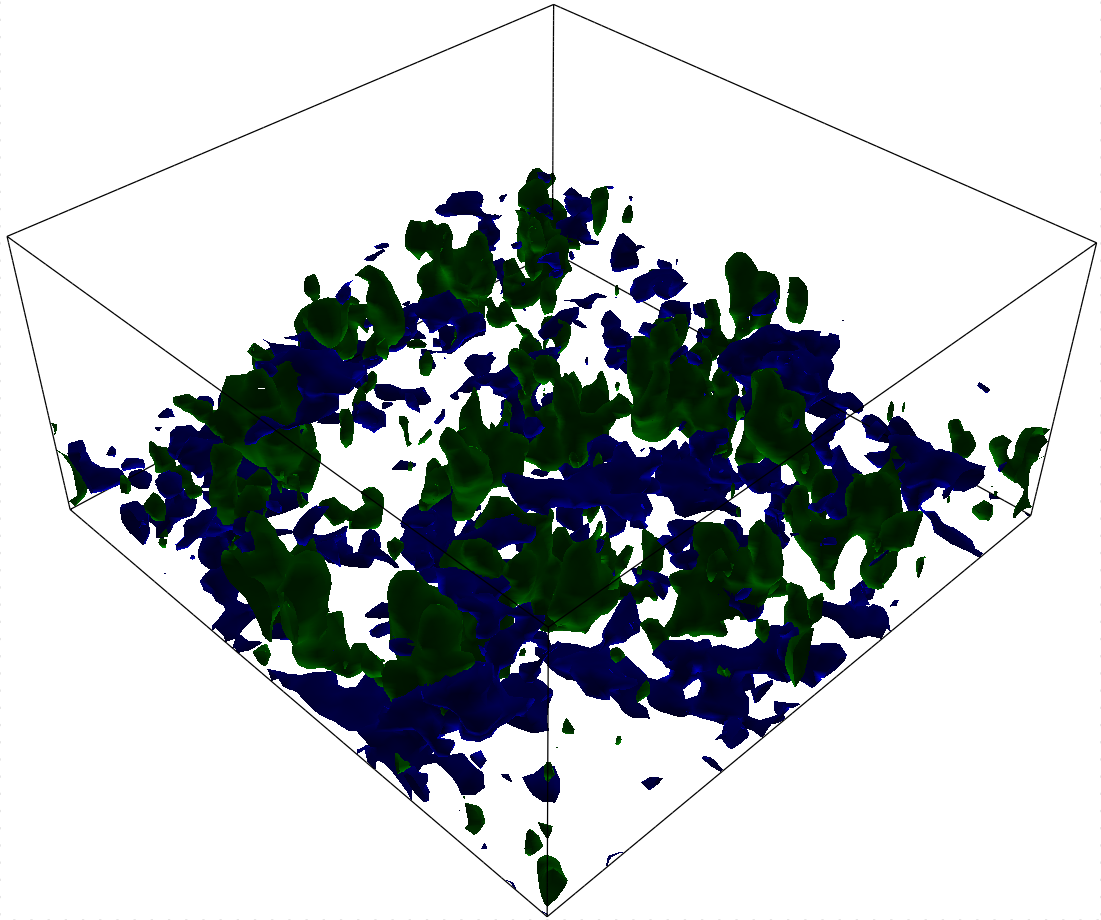
- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

libmpdata++ 2.0: convective boundary layer example



- iLES setup following Margolin, Smolarkiewicz & Sorbjan 1999
- <250 lines of code with libmpdata++

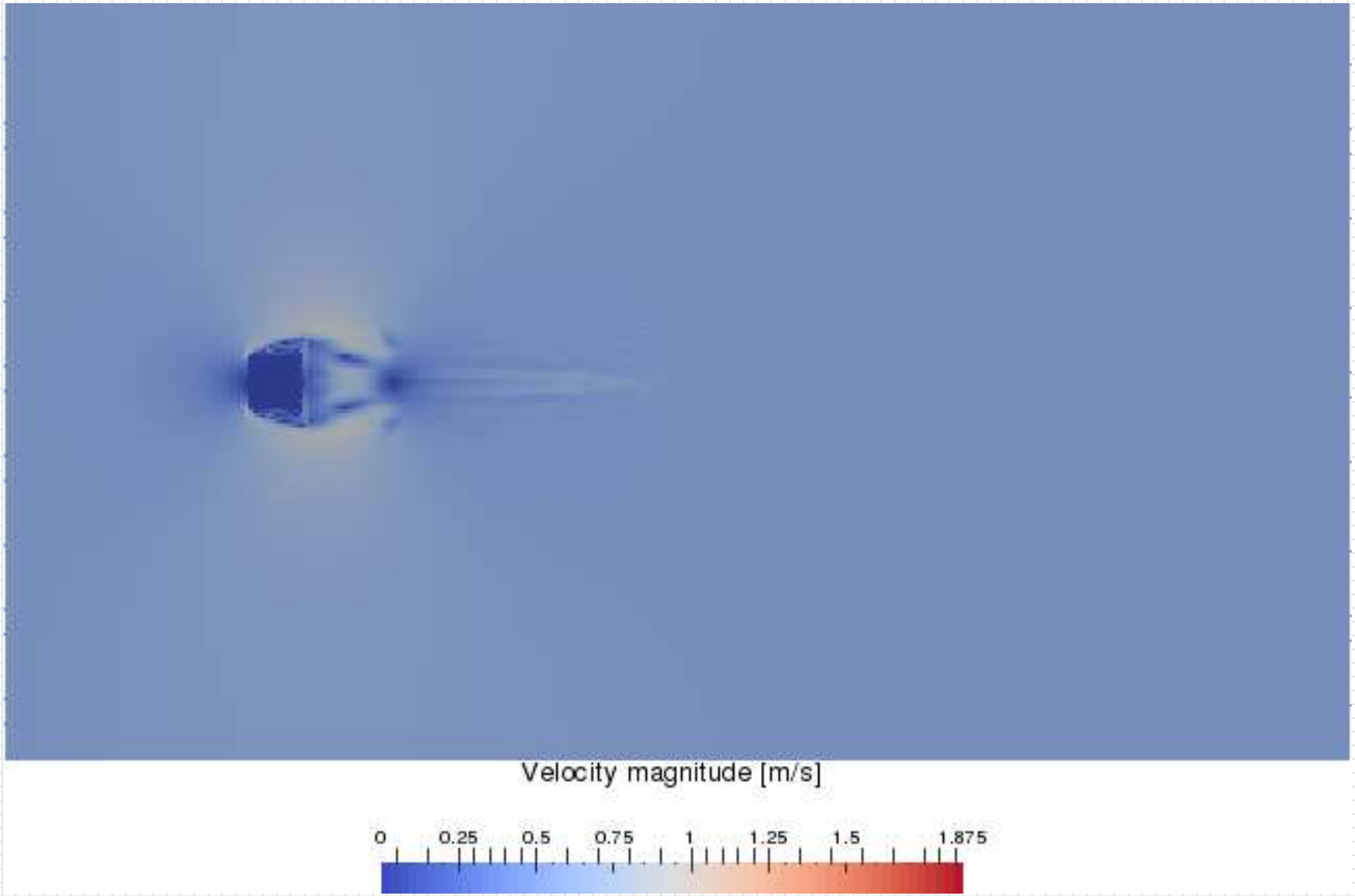
libmpdata++ 2.0: summary of features under development

- higher-order operators
  (for DNS/iLES simulations)
- implicit treatment of absorbers
  (for immersed-boundary method)
- adaptive timestepping
- distributed-memory parallelisation
  (using Boost.MPI & HDF5/MPI-IO)

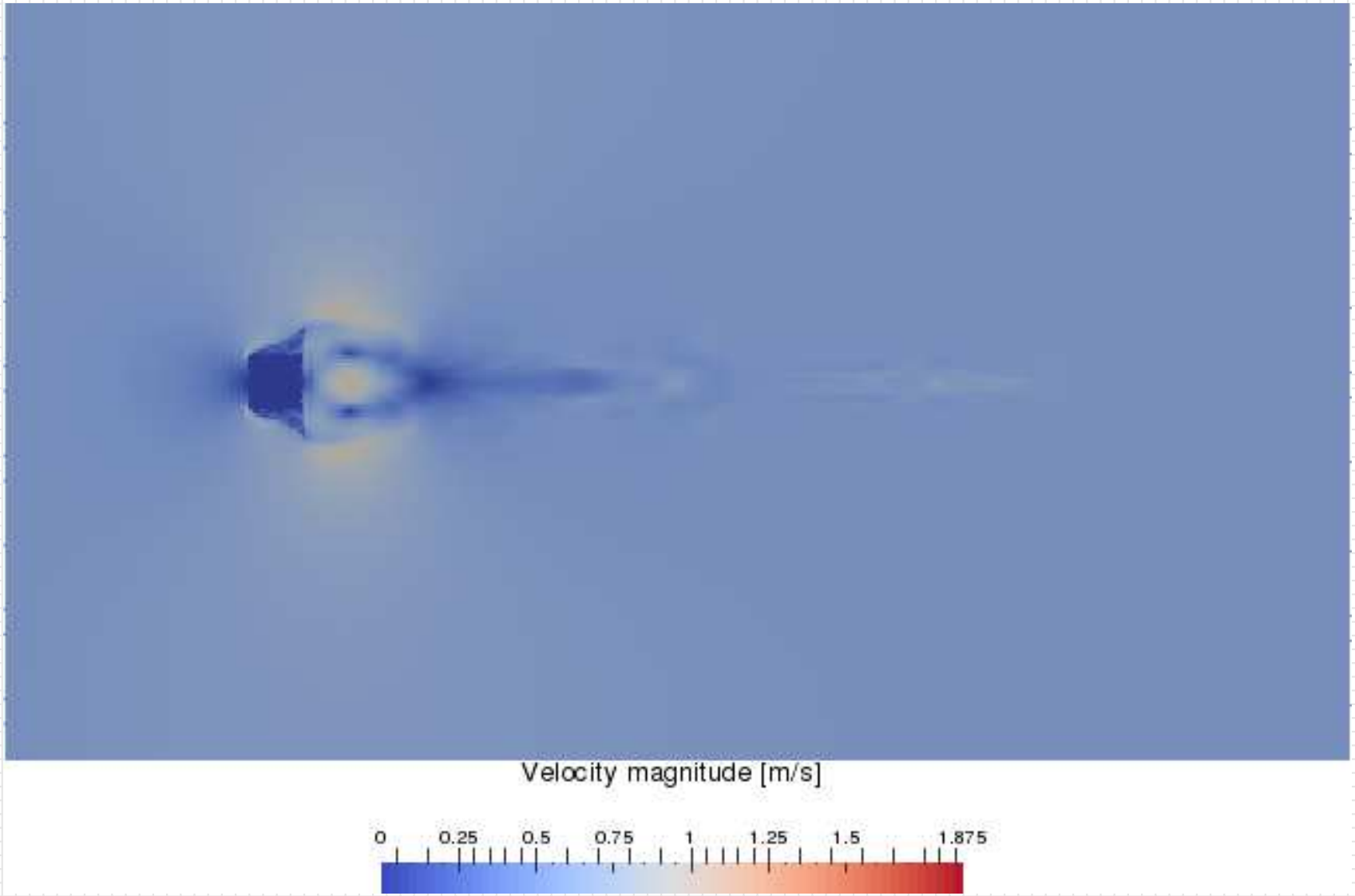
libmpdata++ 2.0: summary of features under development

- higher-order operators
  (for DNS/iLES simulations)
- implicit treatment of absorbers
  (for immersed-boundary method)
- adaptive timestepping
- distributed-memory parallelisation
  (using Boost.MPI & HDF5/MPI-IO)

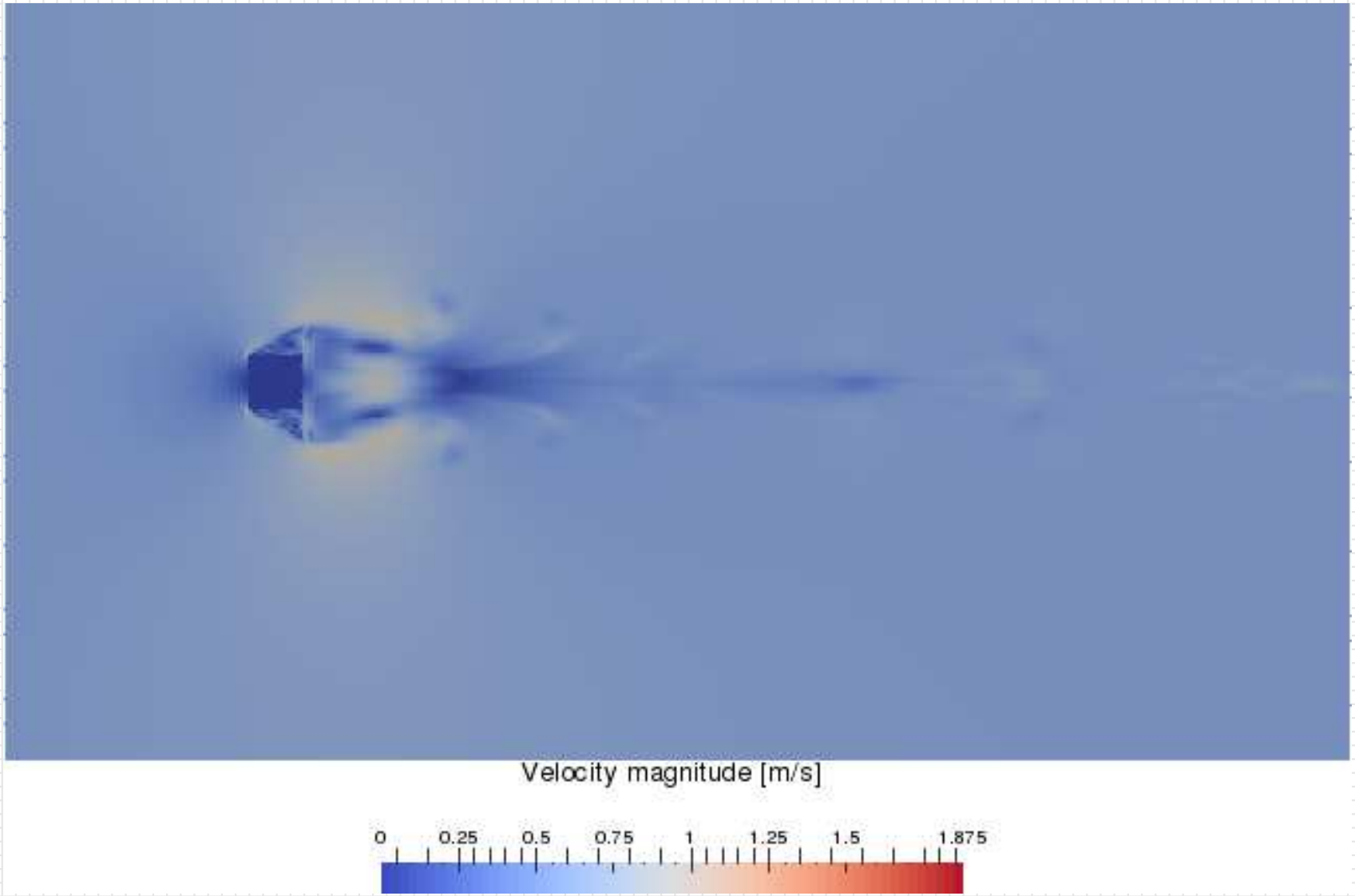
libmpdata++ 2.0: immersed boundary teaser



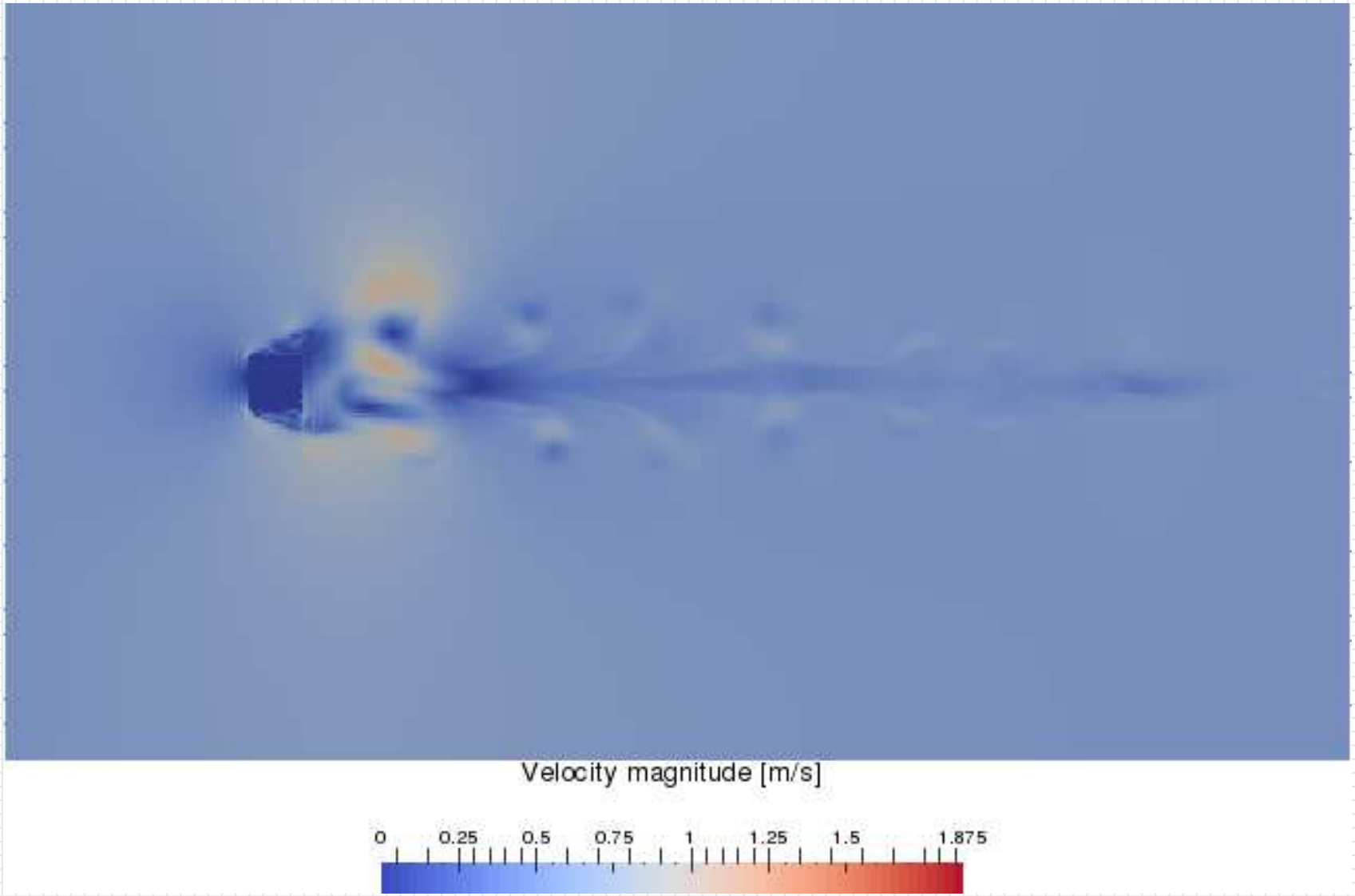
libmpdata++ 2.0: immersed boundary teaser



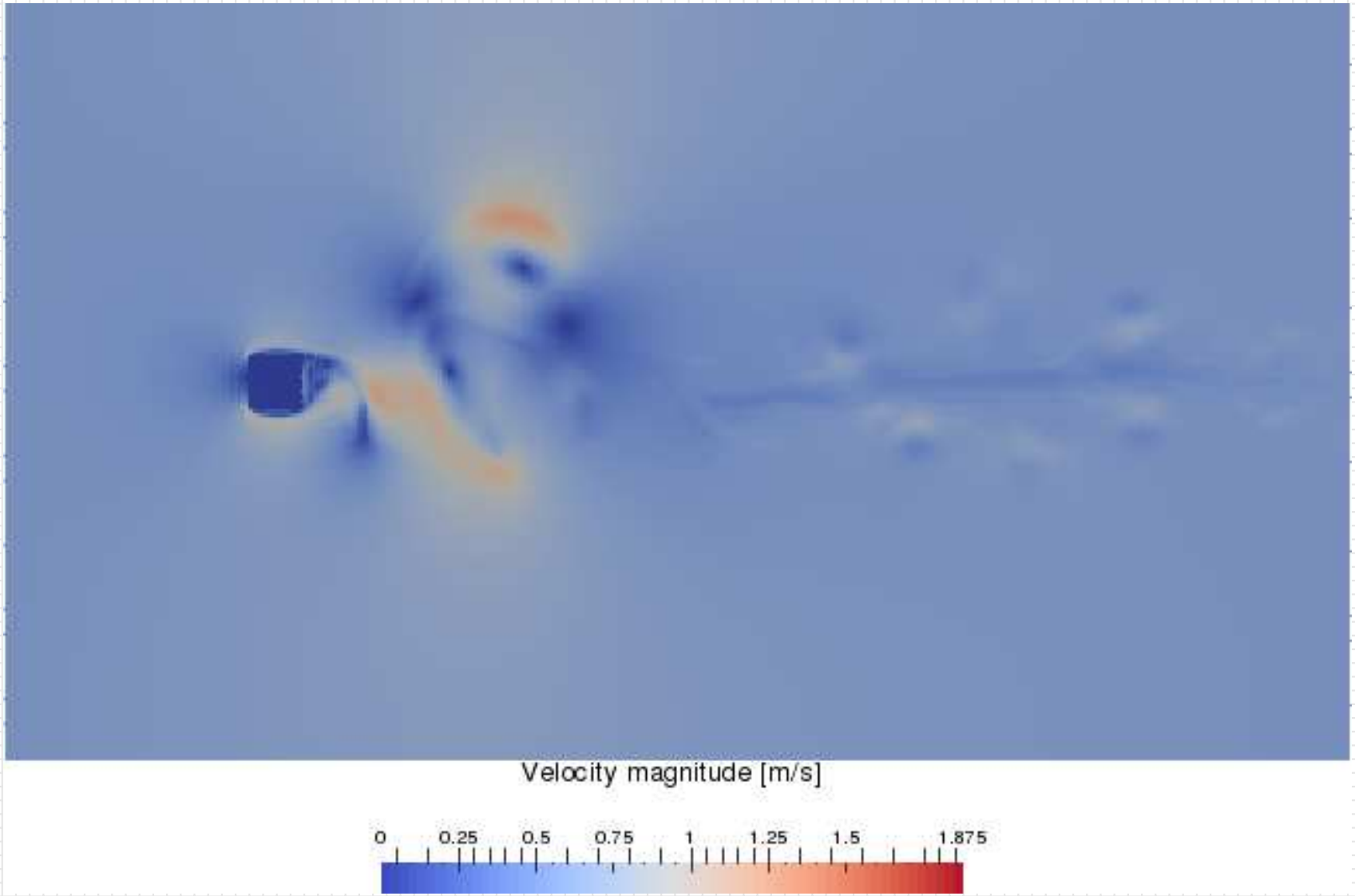
libmpdata++ 2.0: immersed boundary teaser



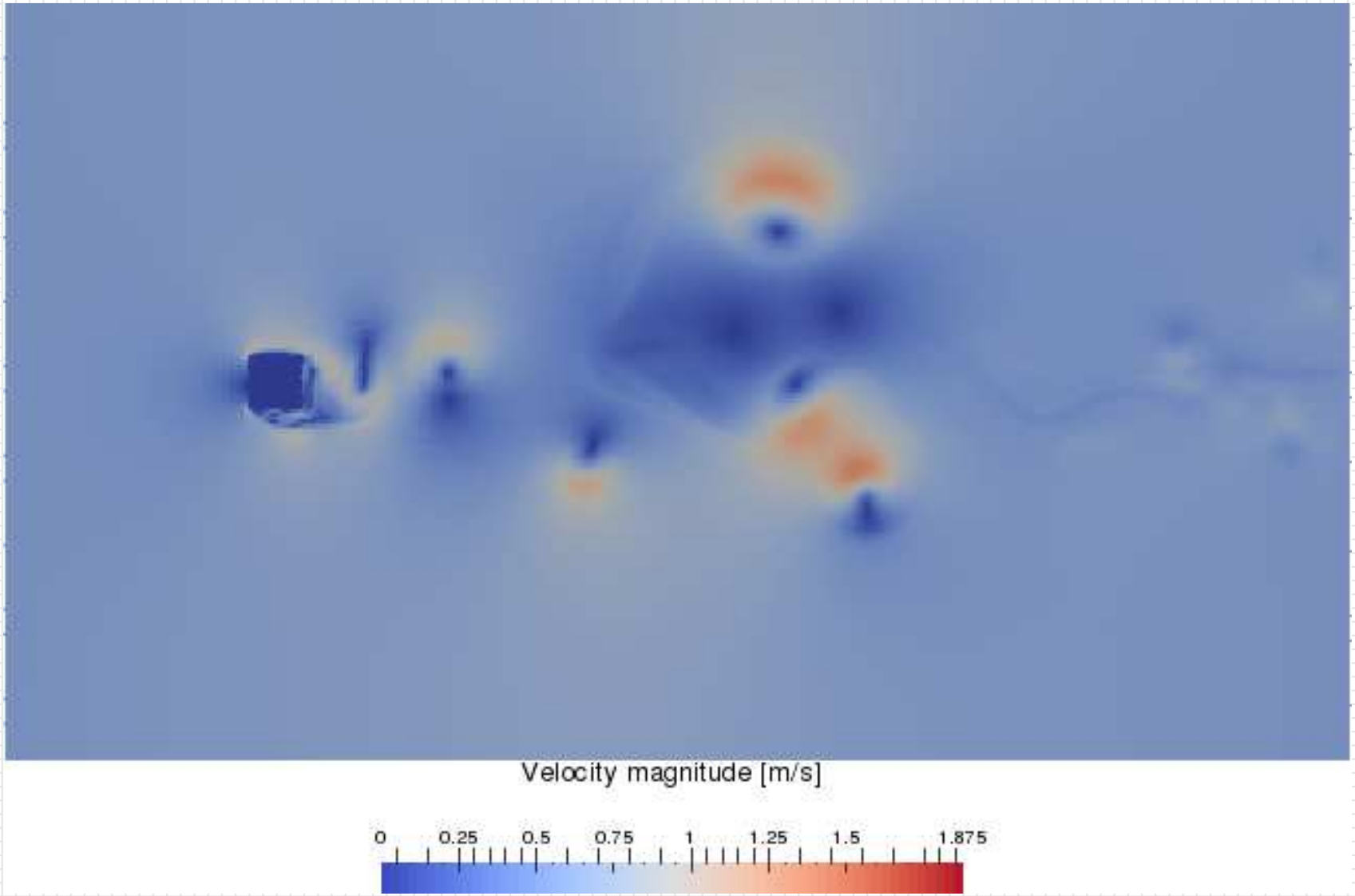
libmpdata++ 2.0: immersed boundary teaser



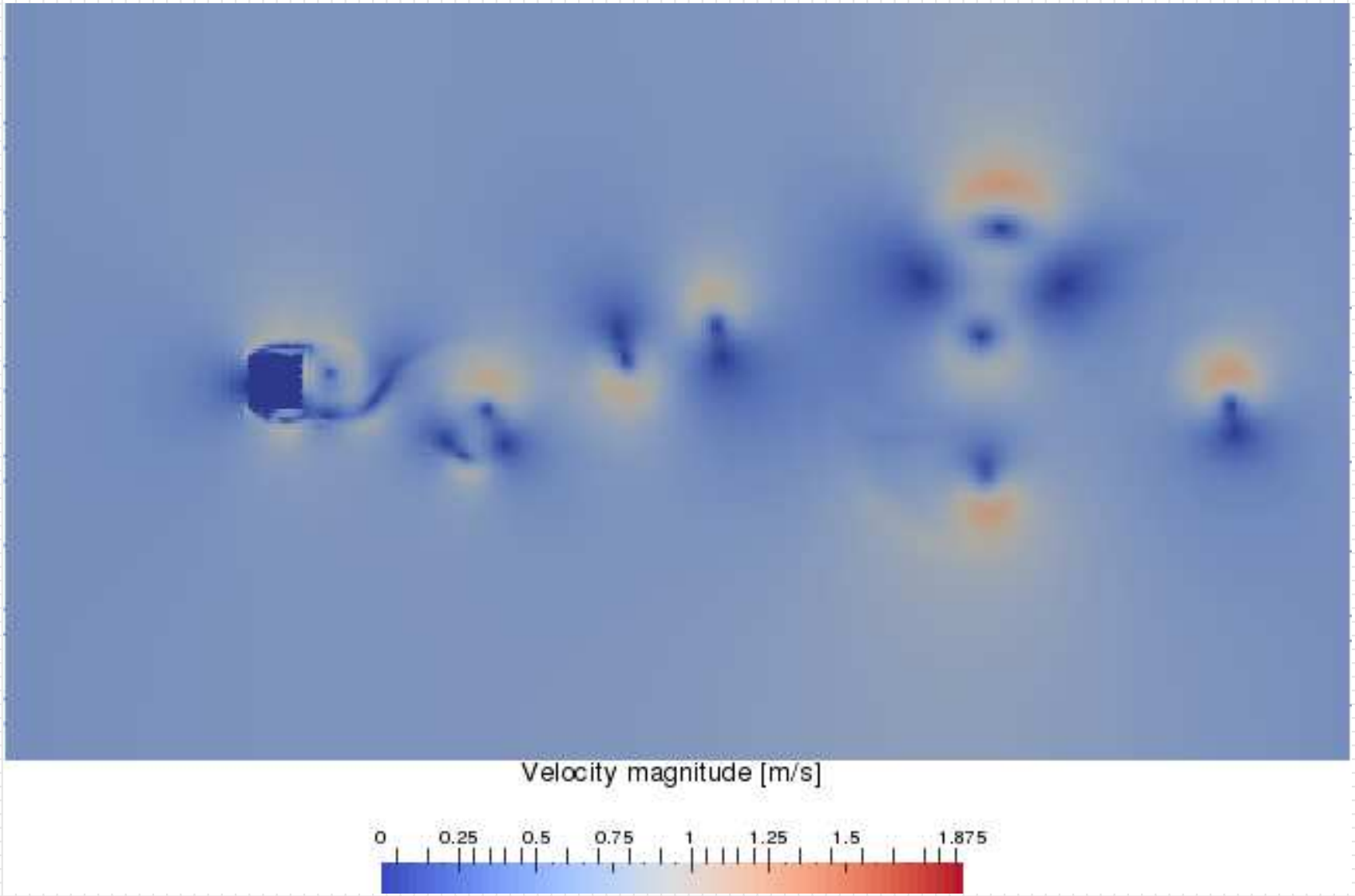
libmpdata++ 2.0: immersed boundary teaser



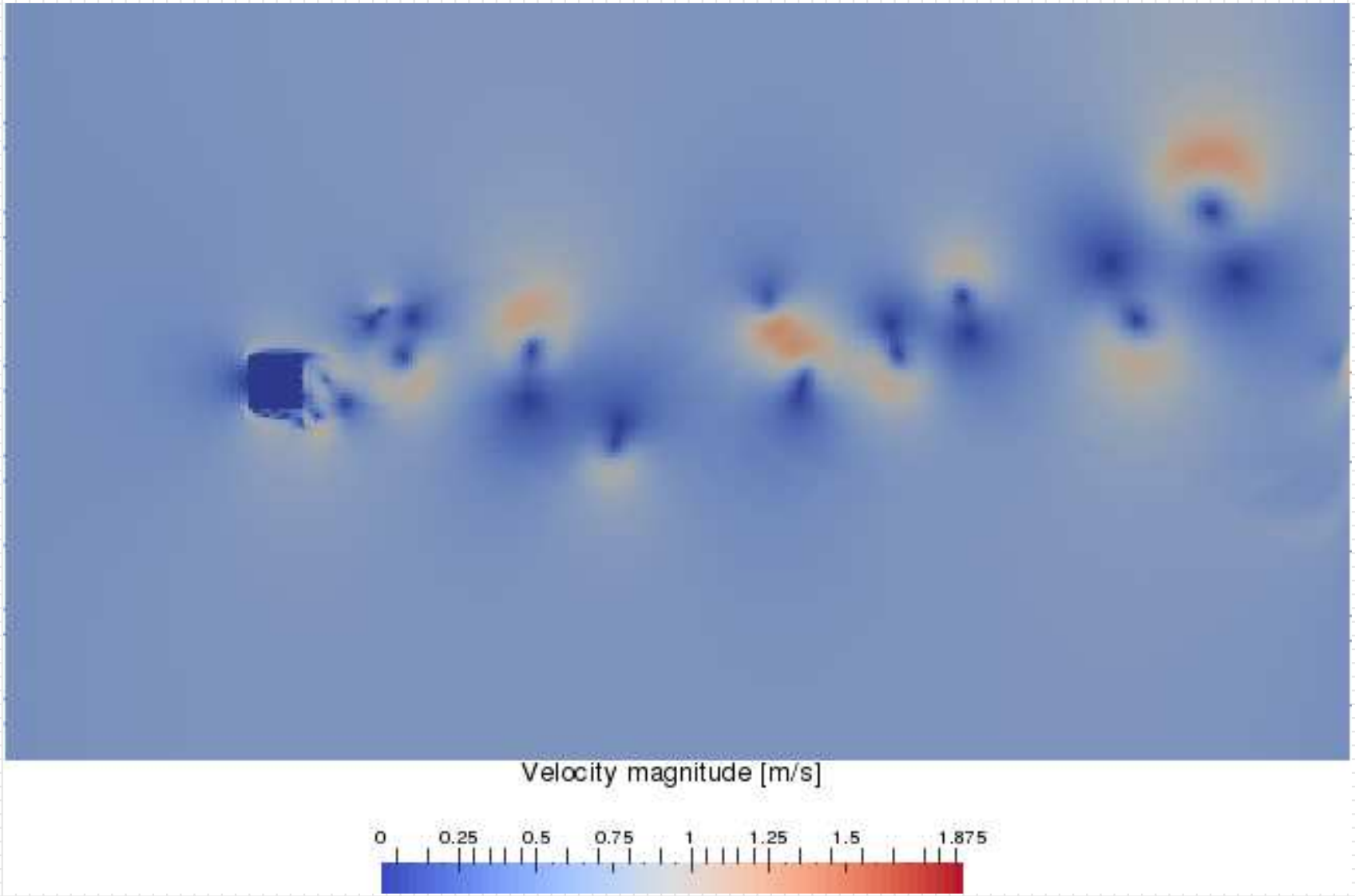
libmpdata++ 2.0: immersed boundary teaser



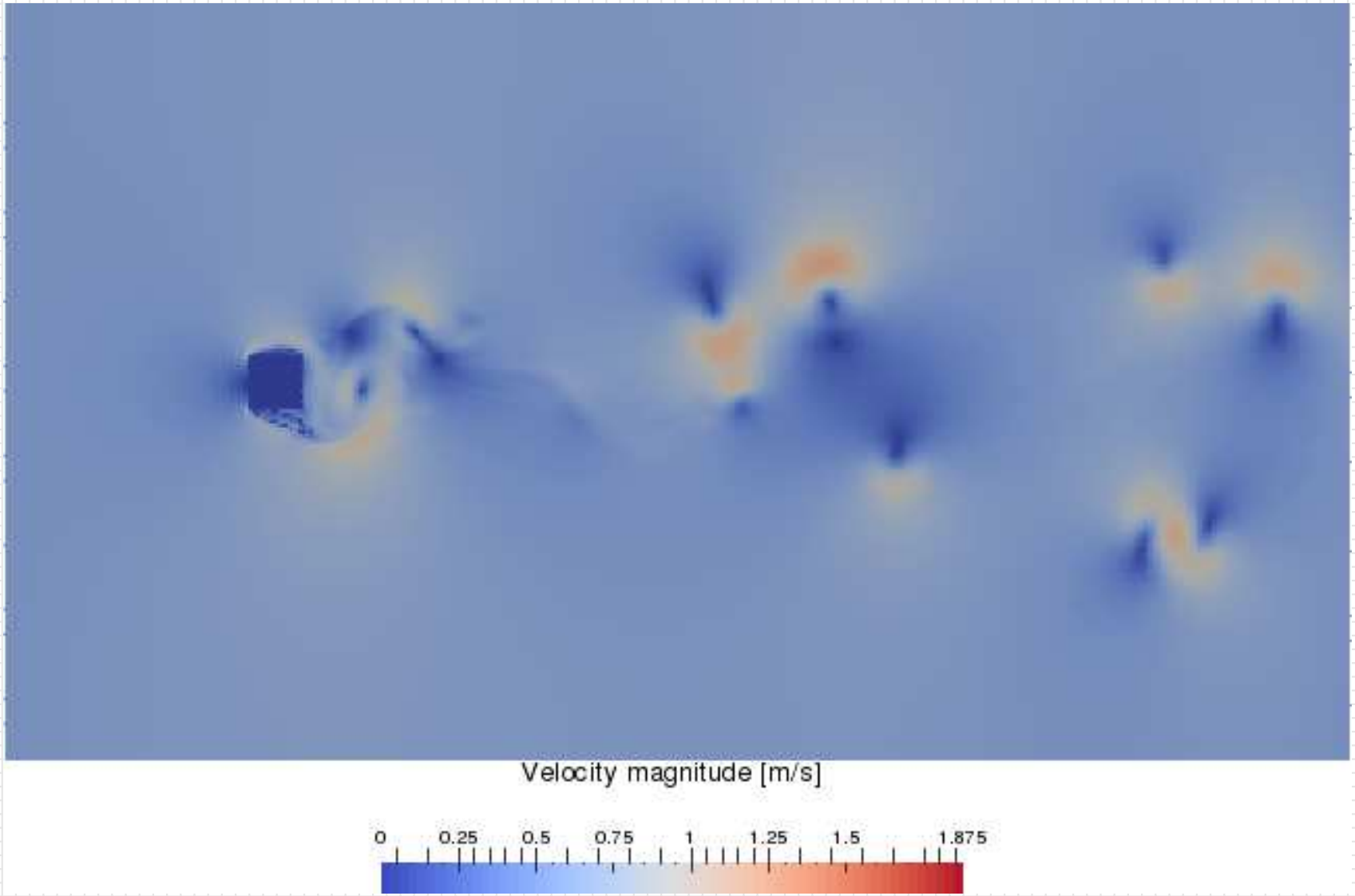
libmpdata++ 2.0: immersed boundary teaser



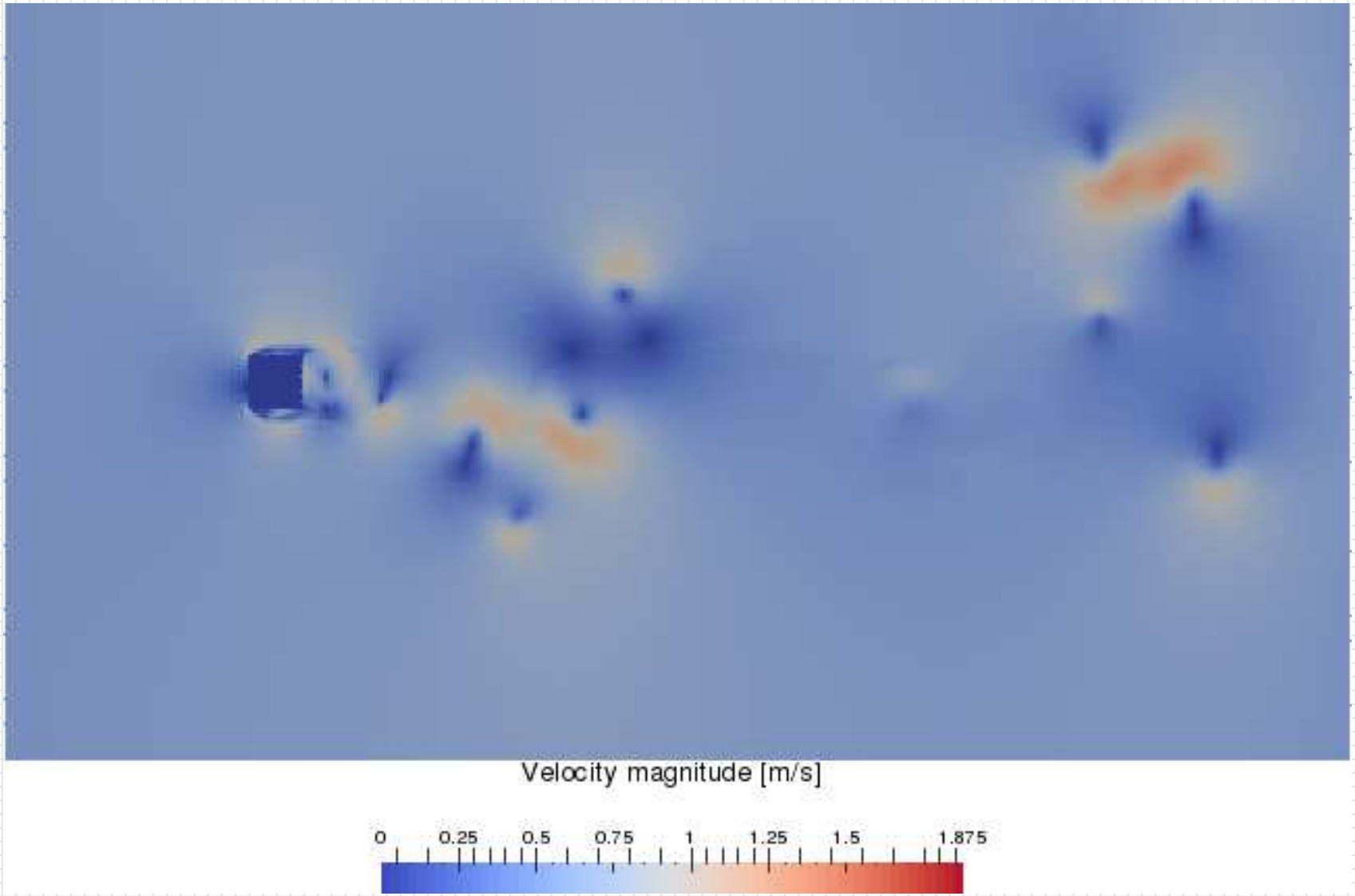
libmpdata++ 2.0: immersed boundary teaser



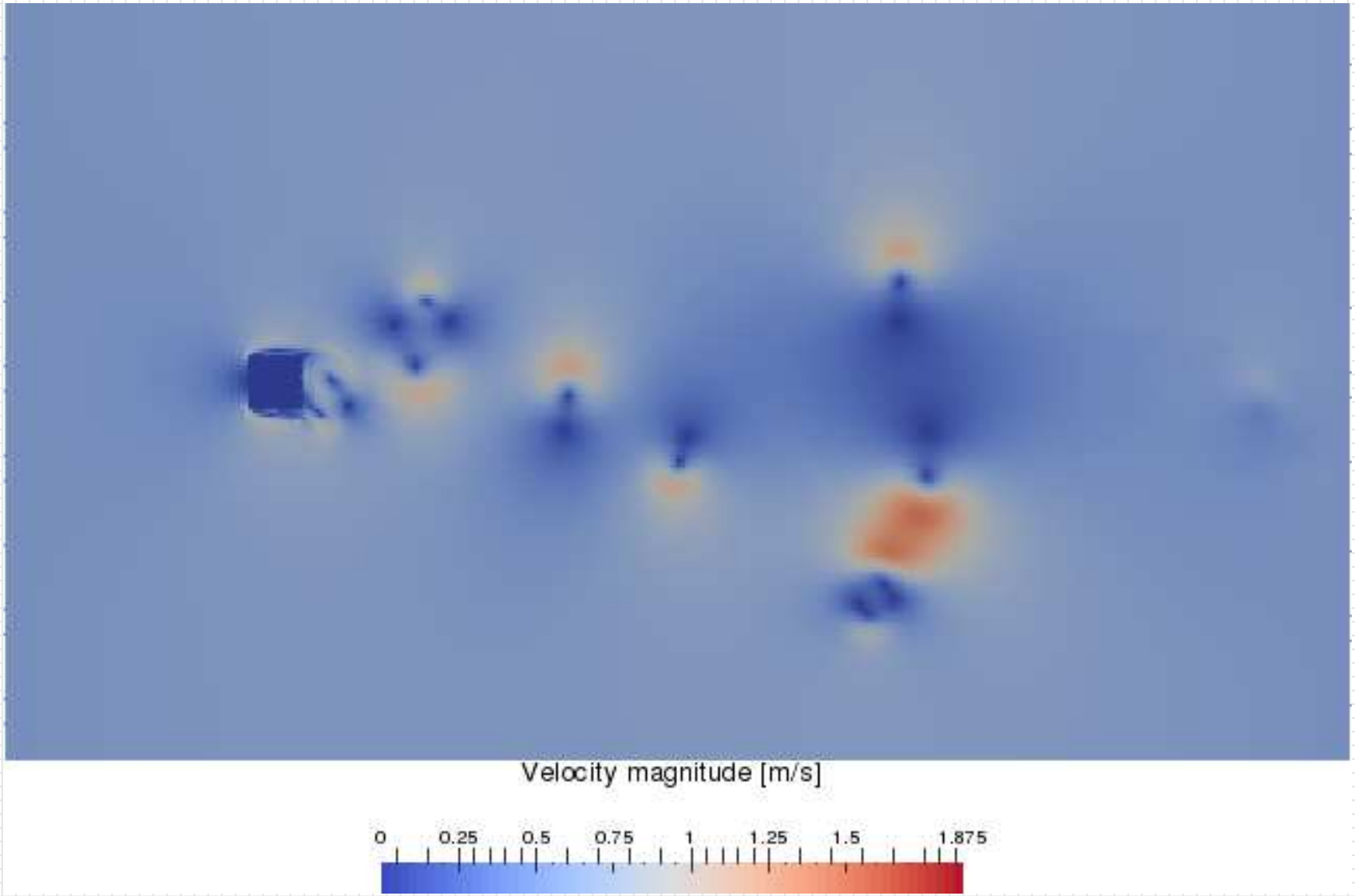
libmpdata++ 2.0: immersed boundary teaser



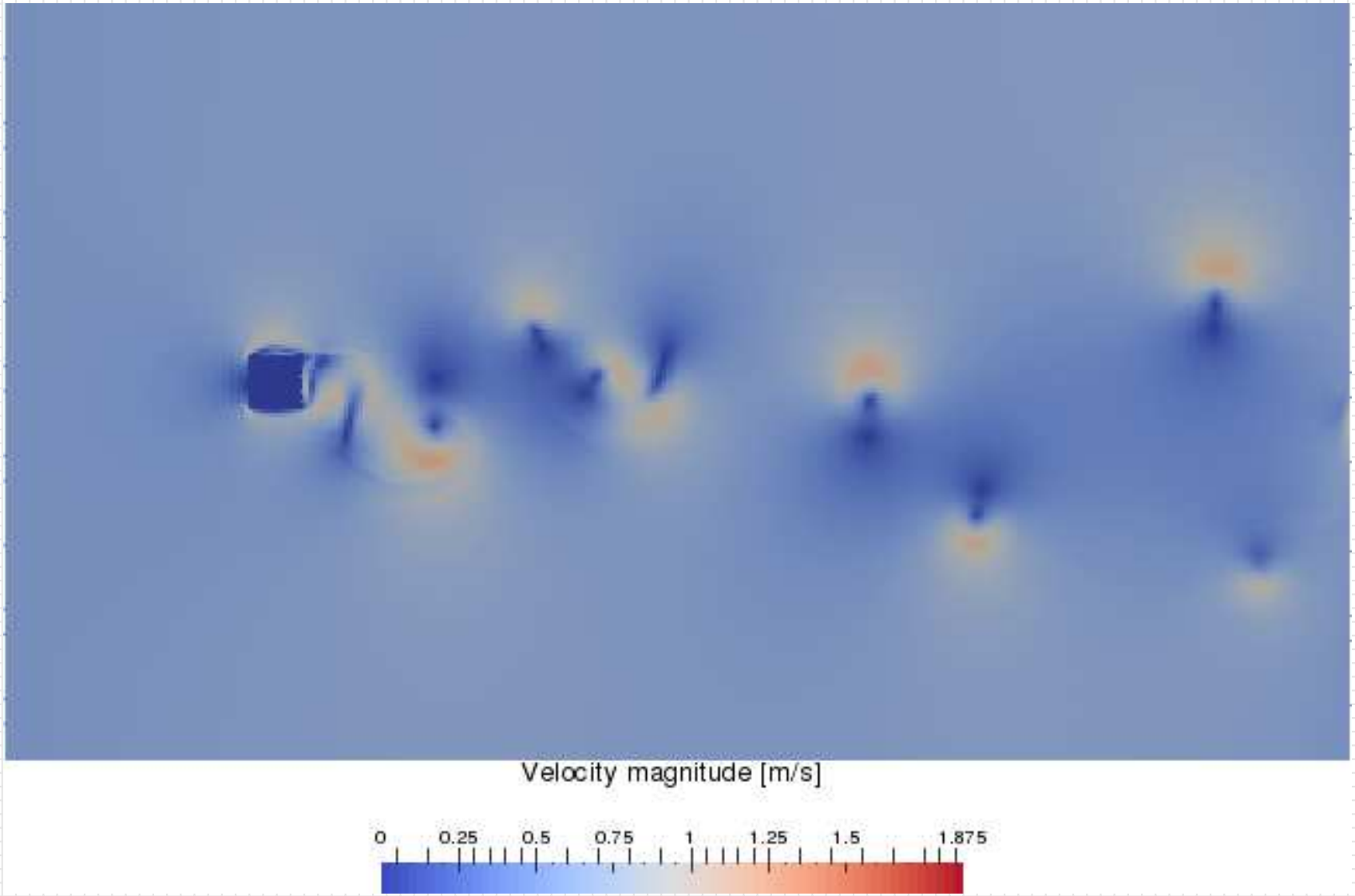
libmpdata++ 2.0: immersed boundary teaser



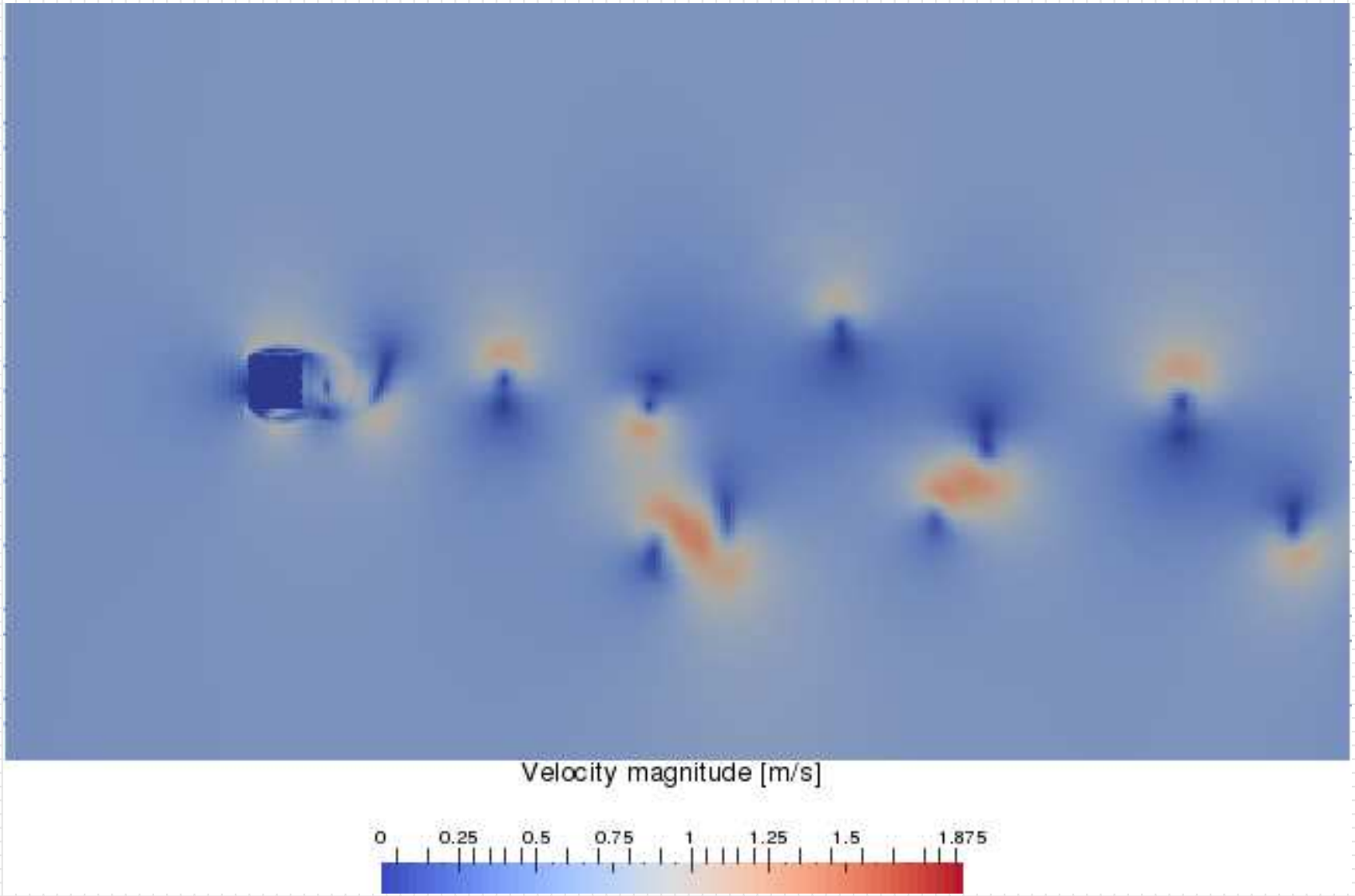
libmpdata++ 2.0: immersed boundary teaser



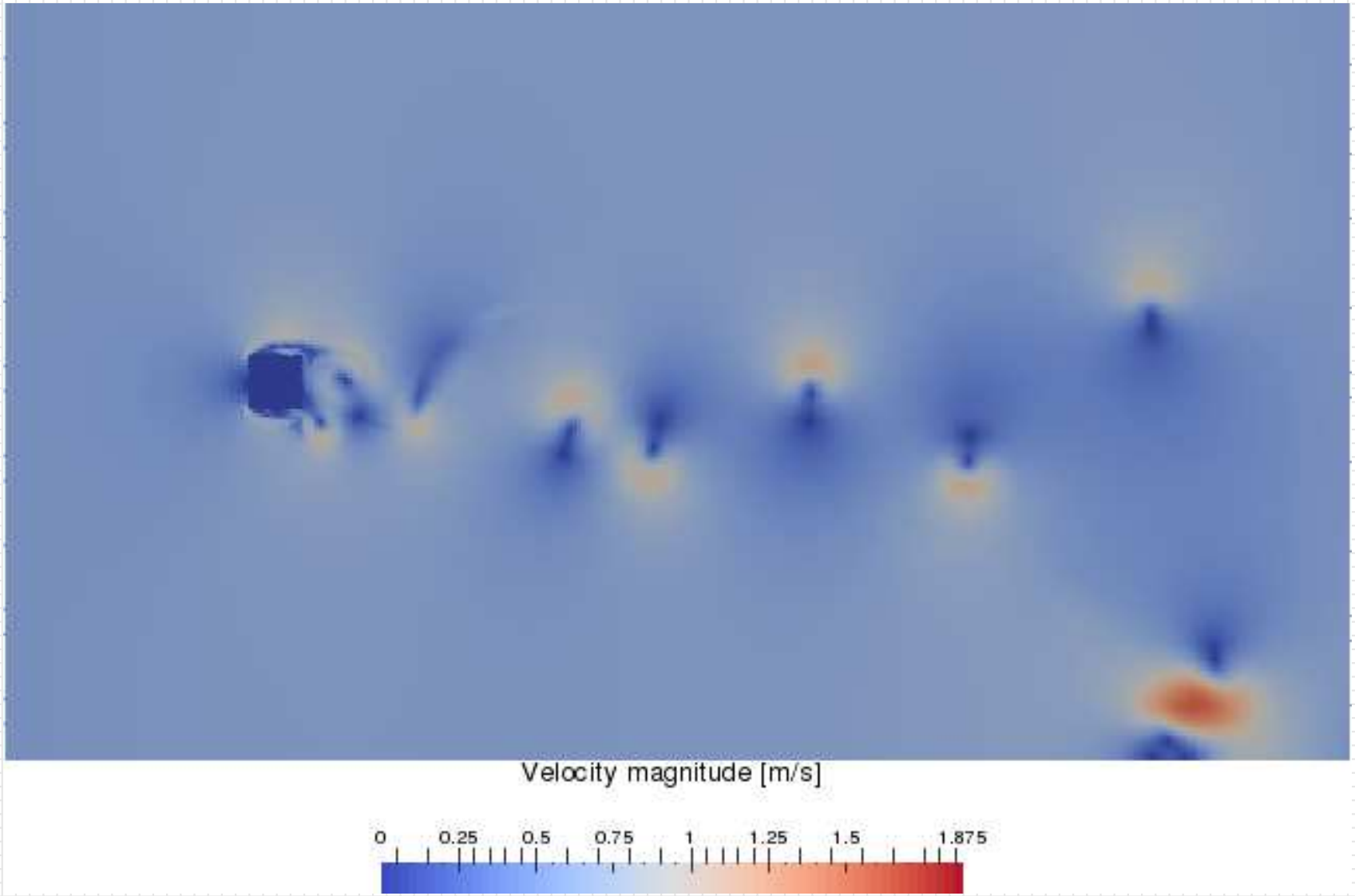
libmpdata++ 2.0: immersed boundary teaser



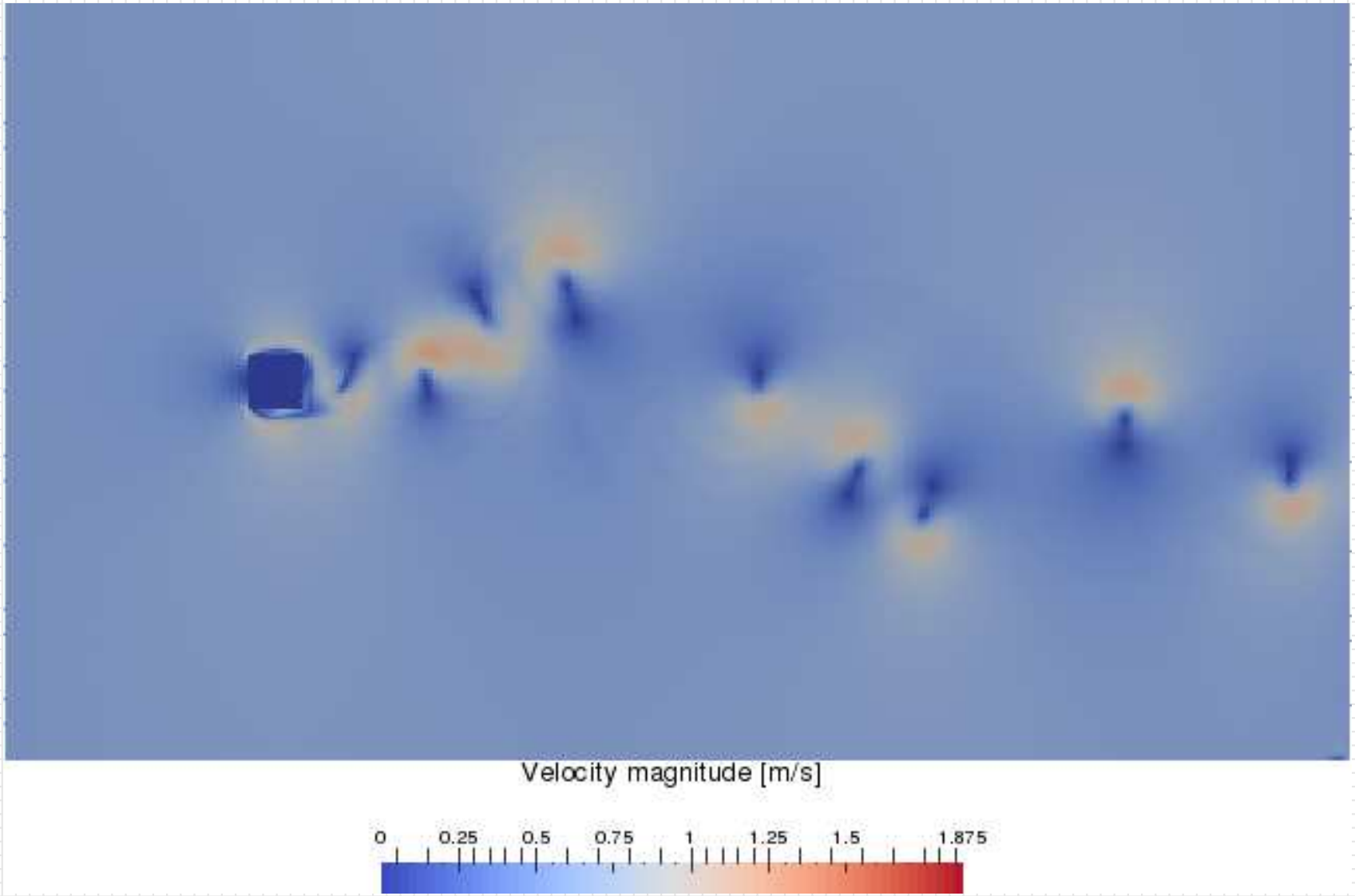
libmpdata++ 2.0: immersed boundary teaser



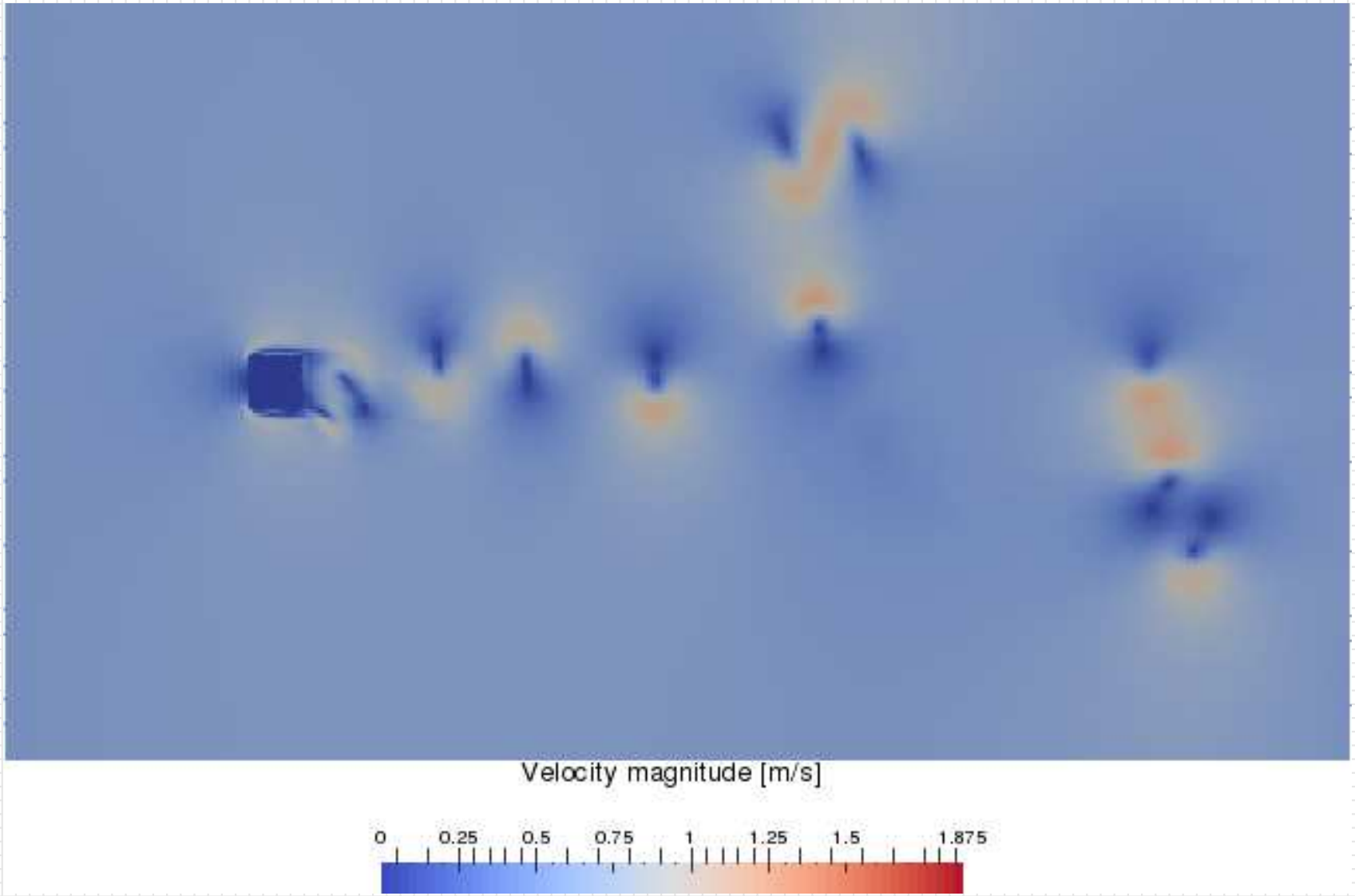
libmpdata++ 2.0: immersed boundary teaser



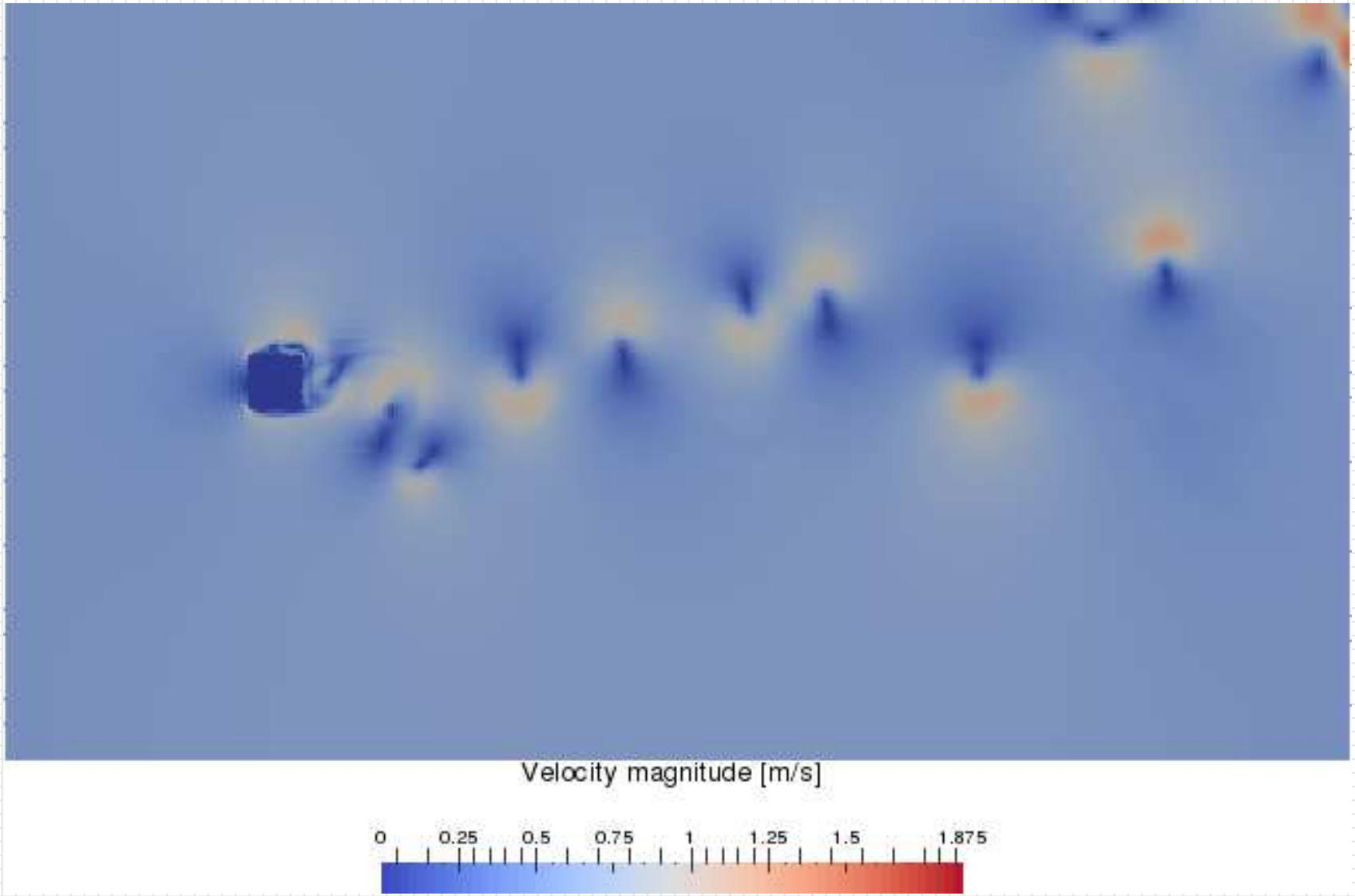
libmpdata++ 2.0: immersed boundary teaser



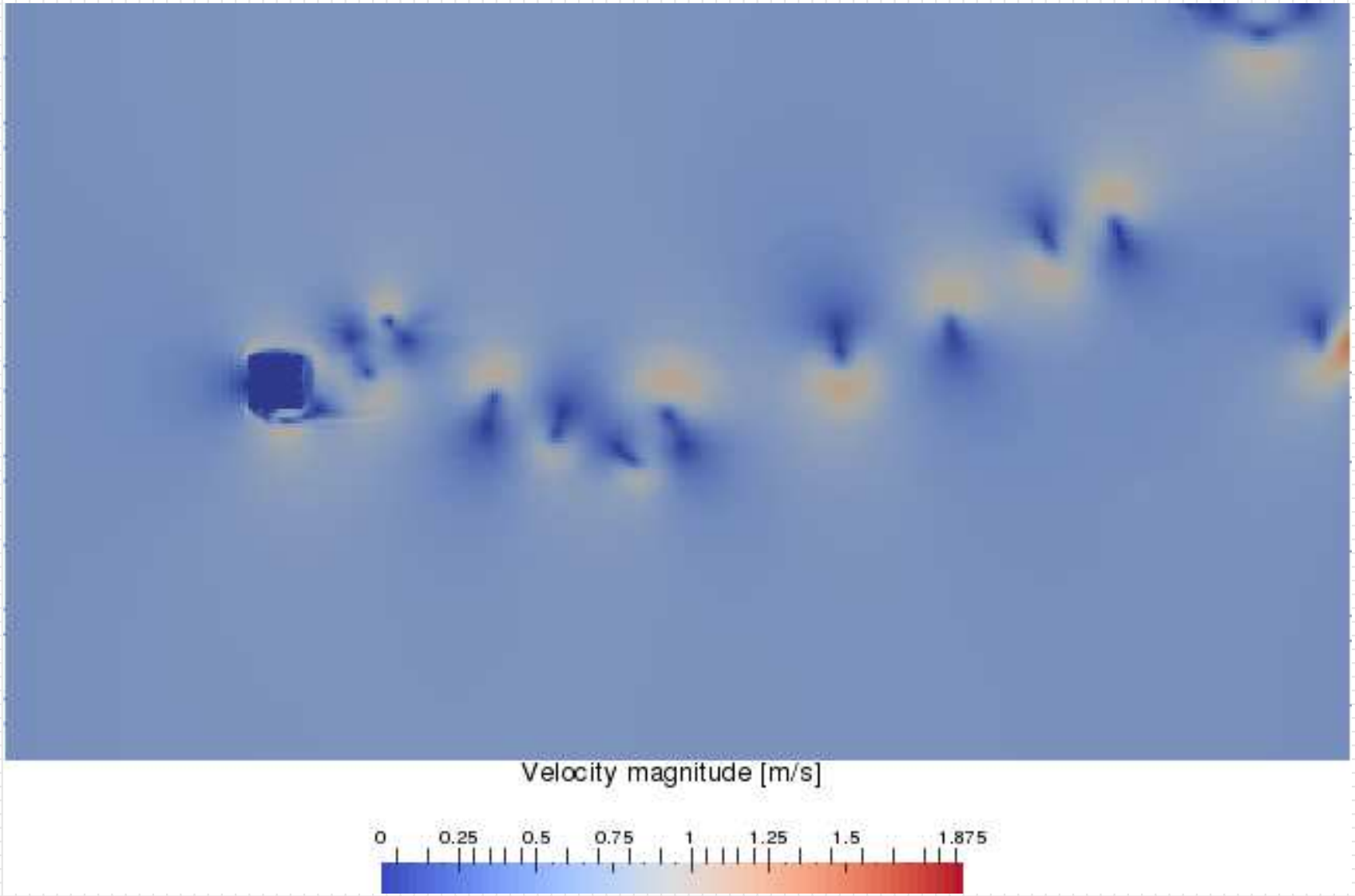
libmpdata++ 2.0: immersed boundary teaser



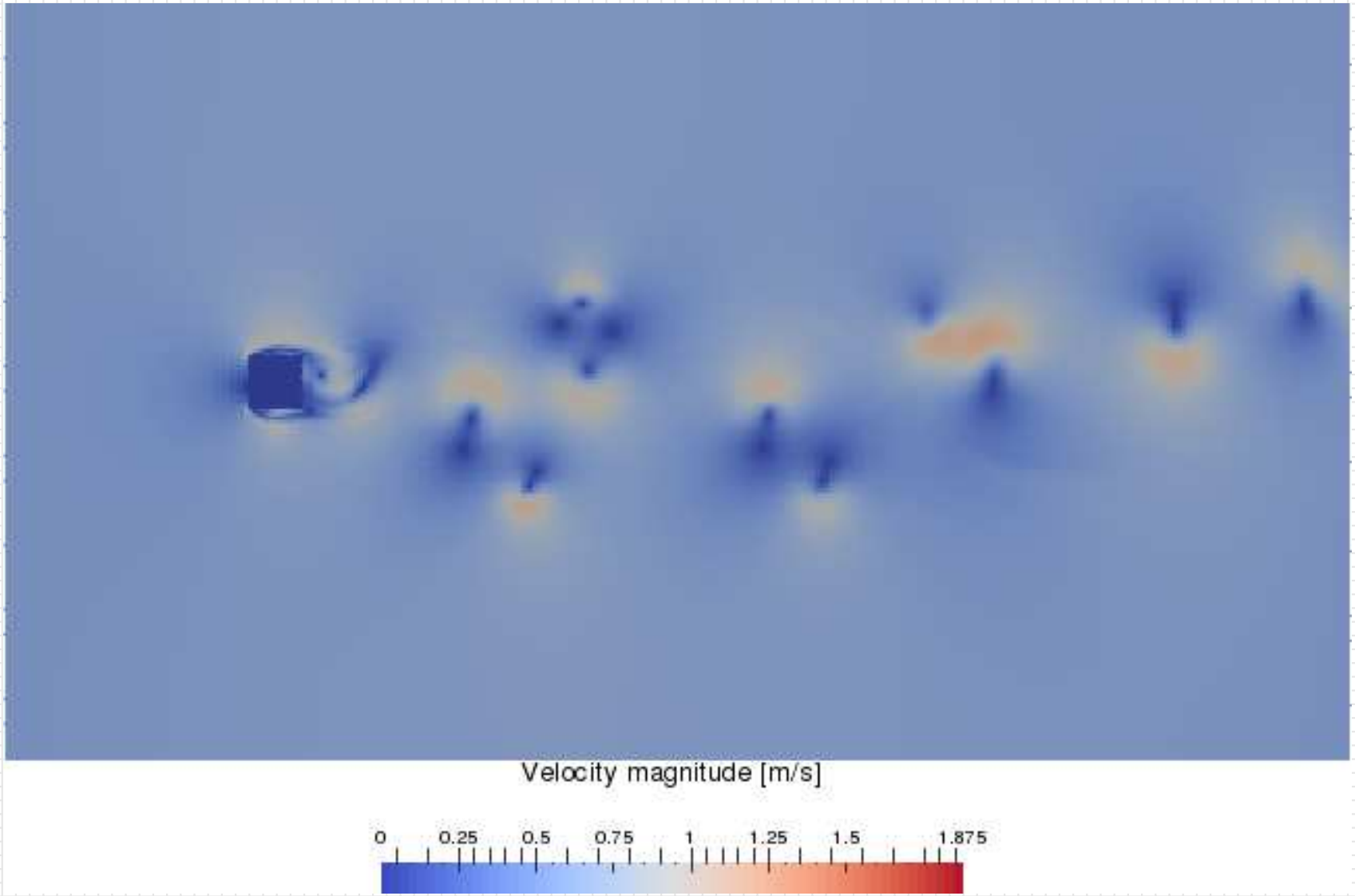
libmpdata++ 2.0: immersed boundary teaser



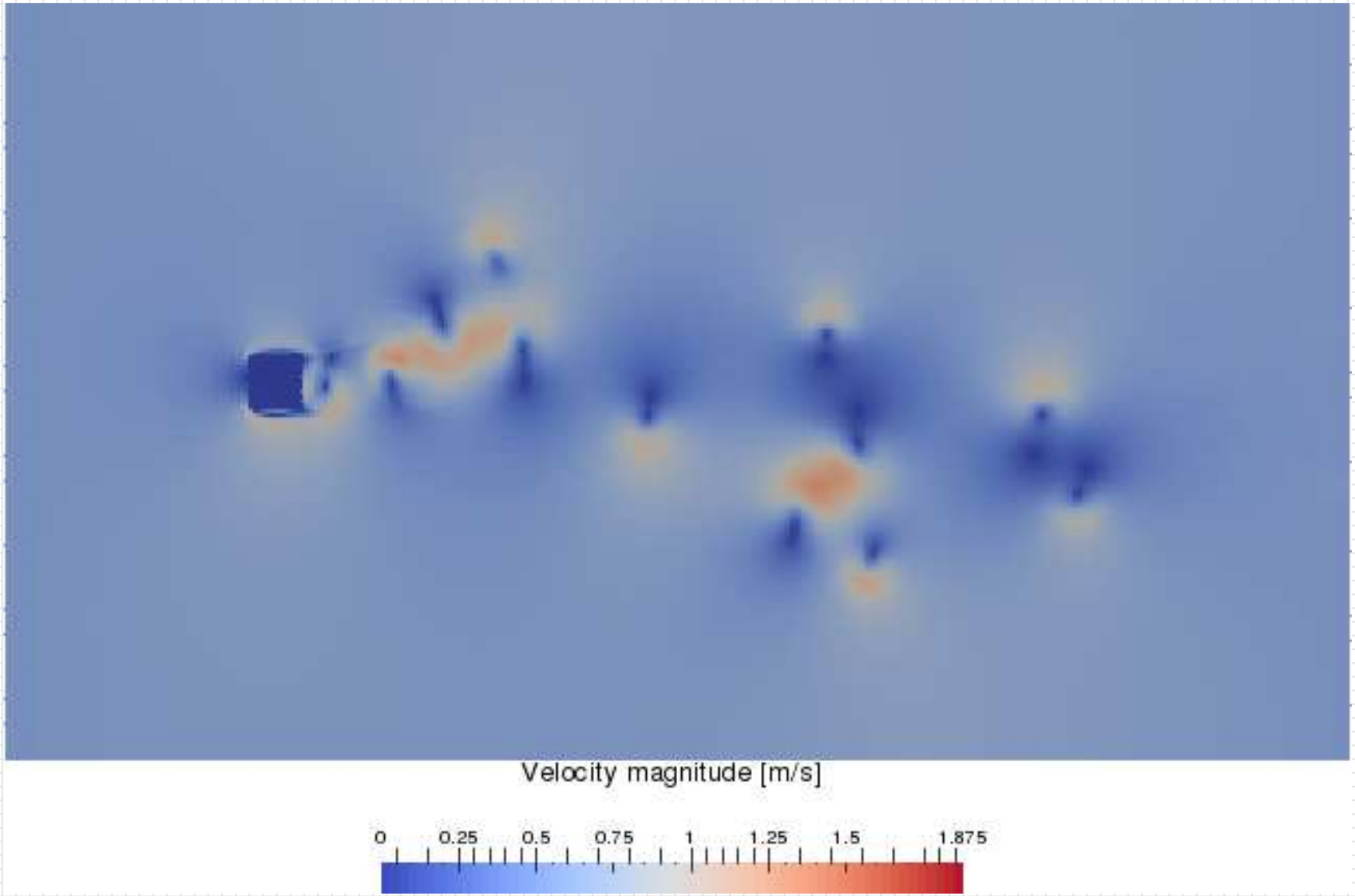
libmpdata++ 2.0: immersed boundary teaser



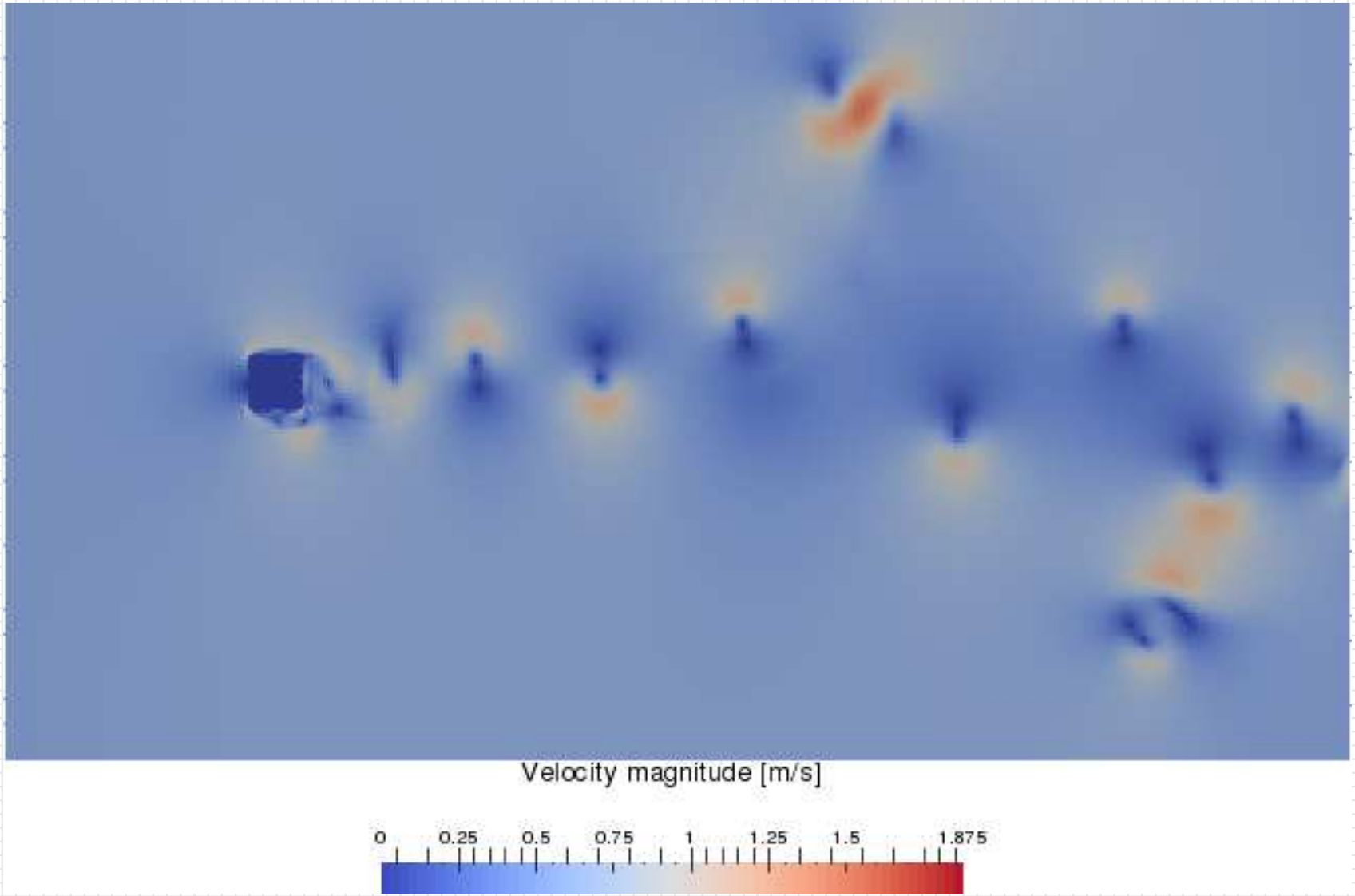
libmpdata++ 2.0: immersed boundary teaser



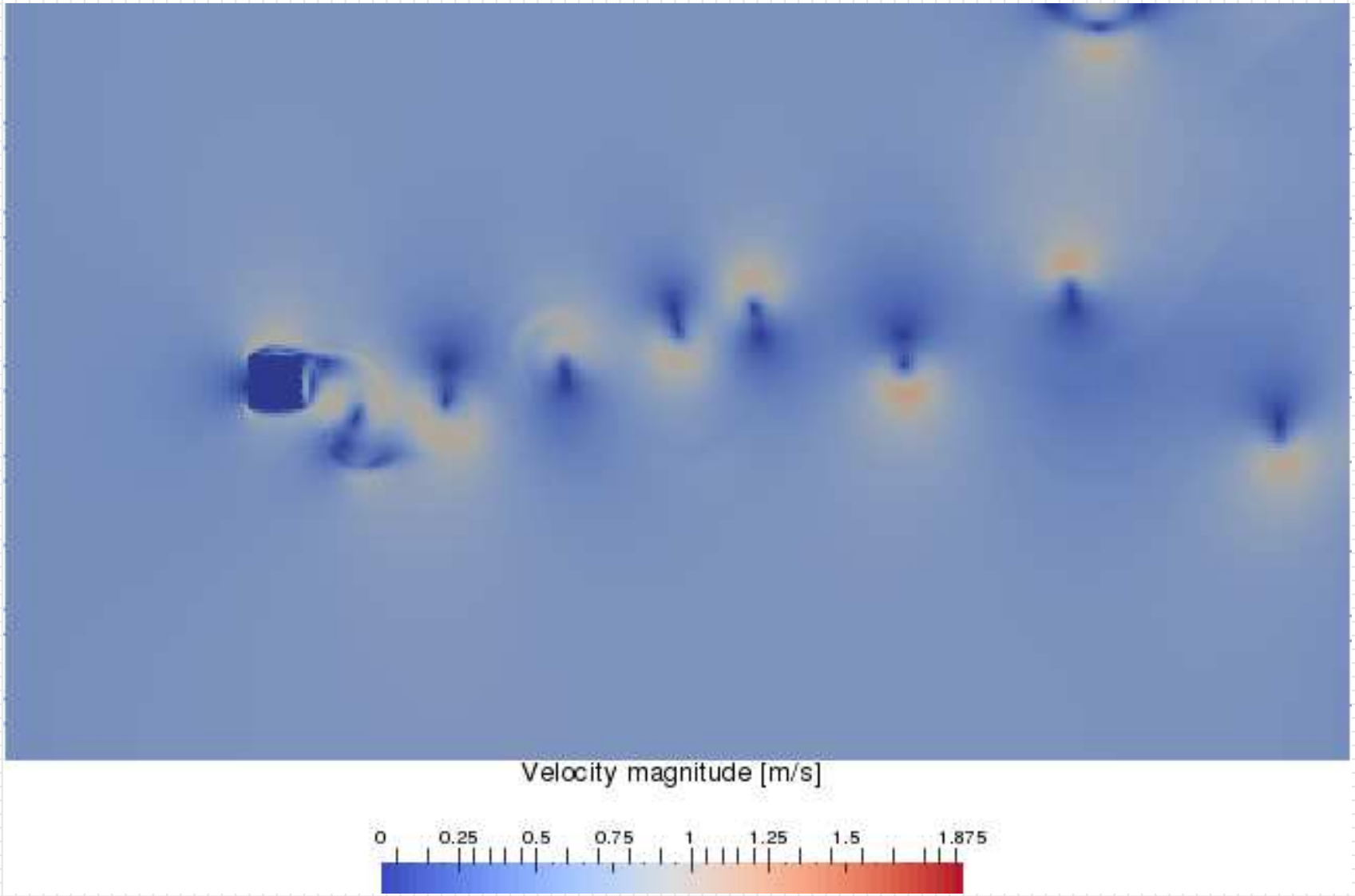
libmpdata++ 2.0: immersed boundary teaser



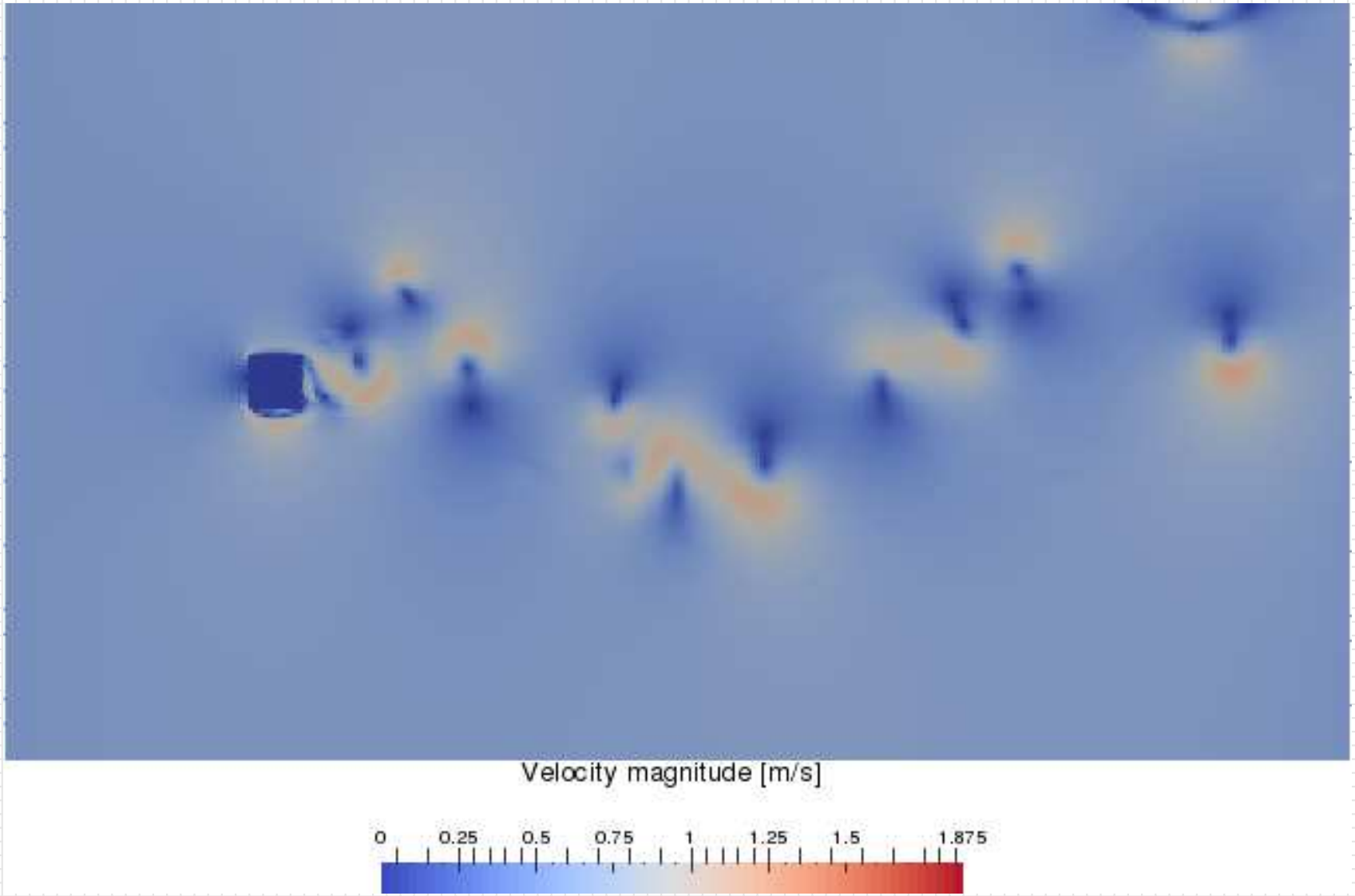
libmpdata++ 2.0: immersed boundary teaser



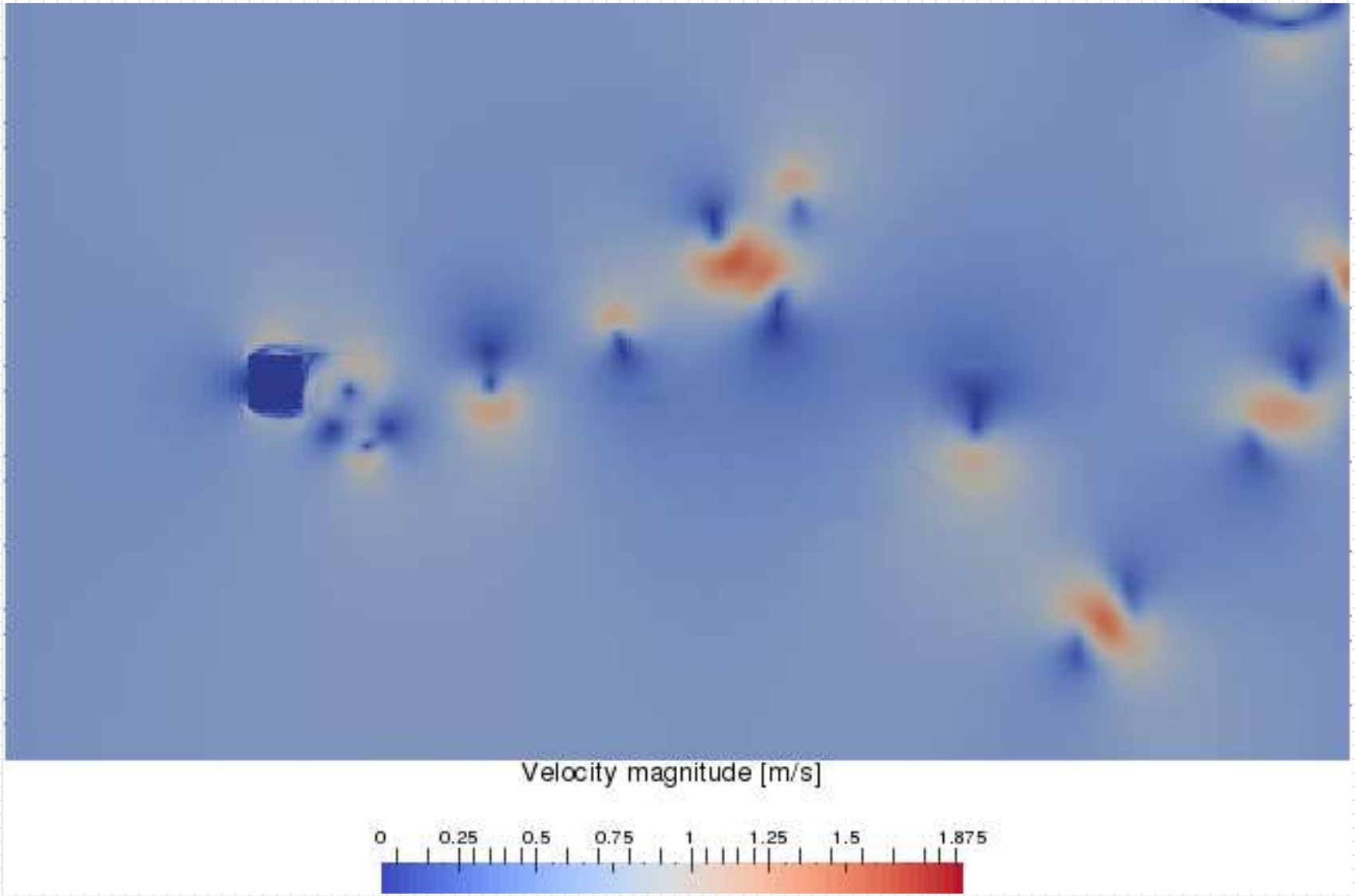
libmpdata++ 2.0: immersed boundary teaser



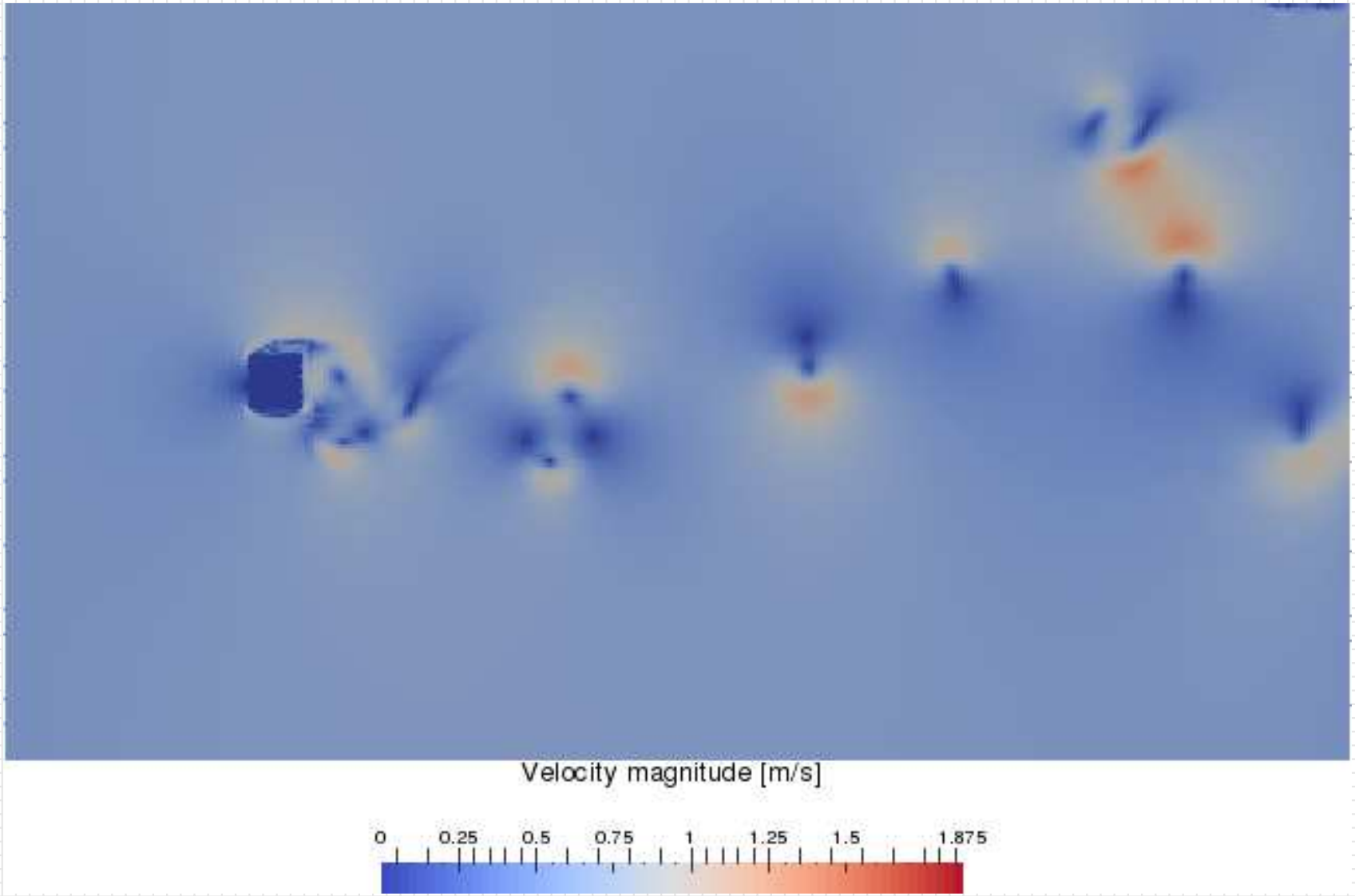
libmpdata++ 2.0: immersed boundary teaser



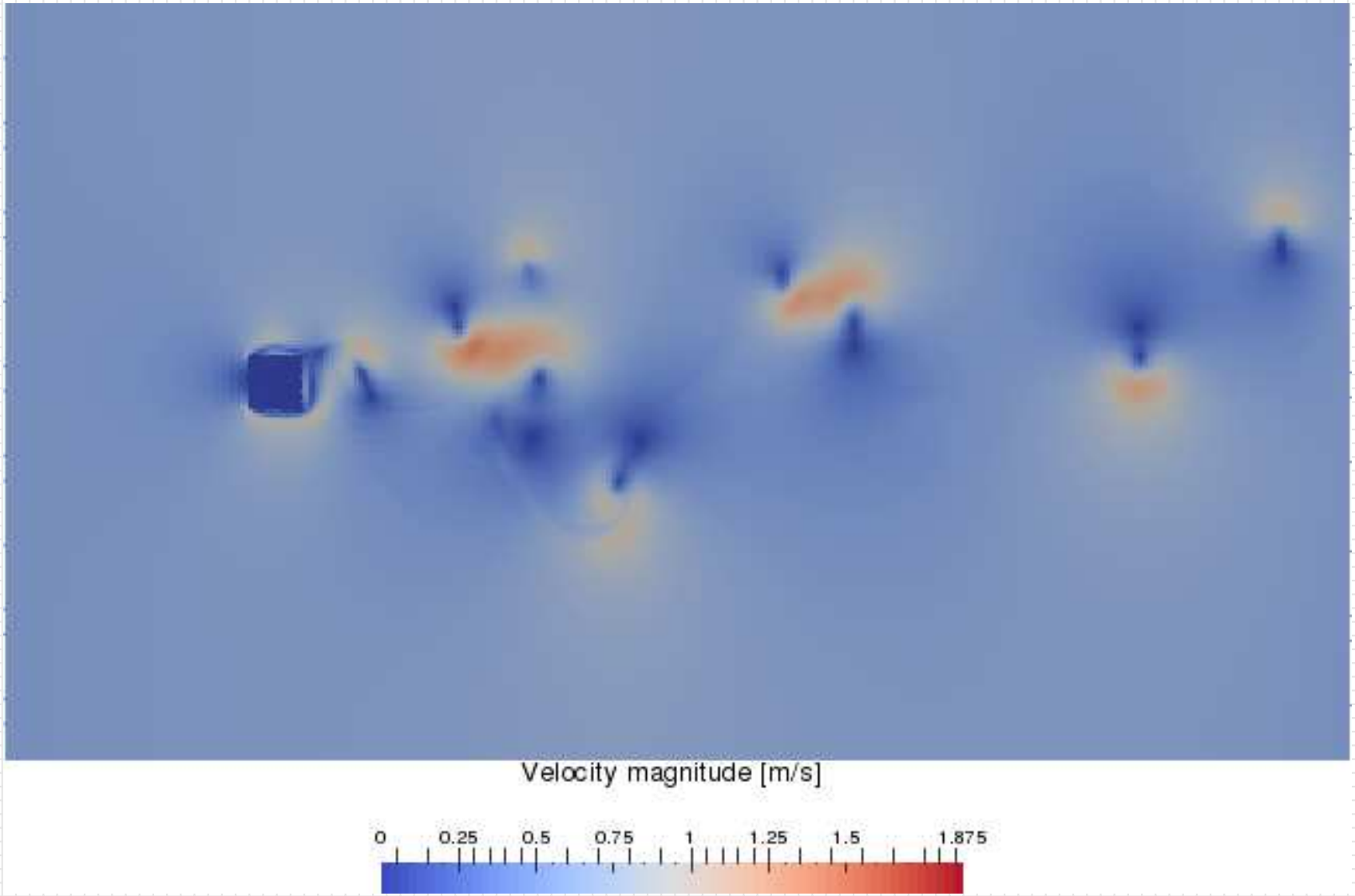
libmpdata++ 2.0: immersed boundary teaser



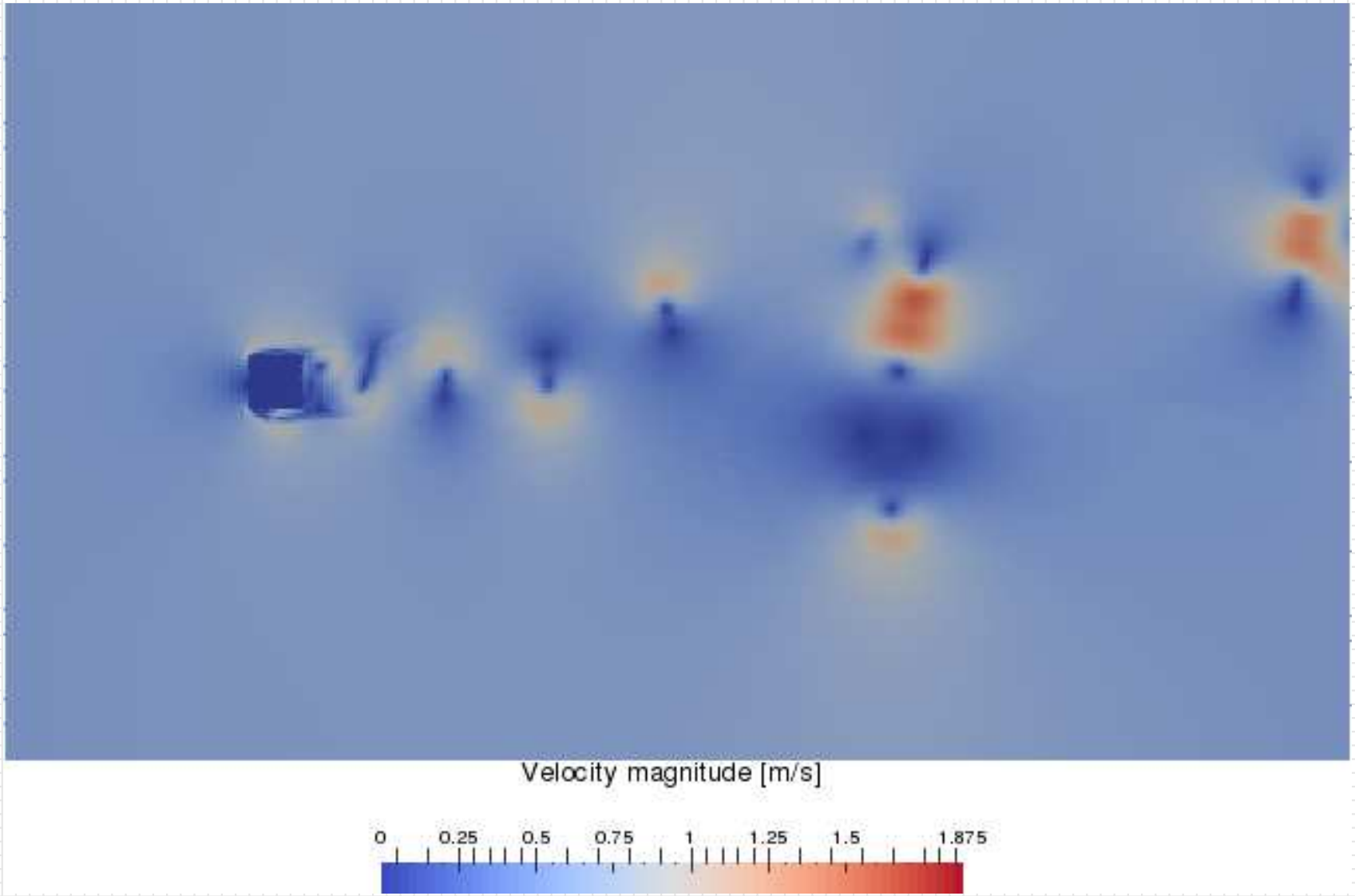
libmpdata++ 2.0: immersed boundary teaser



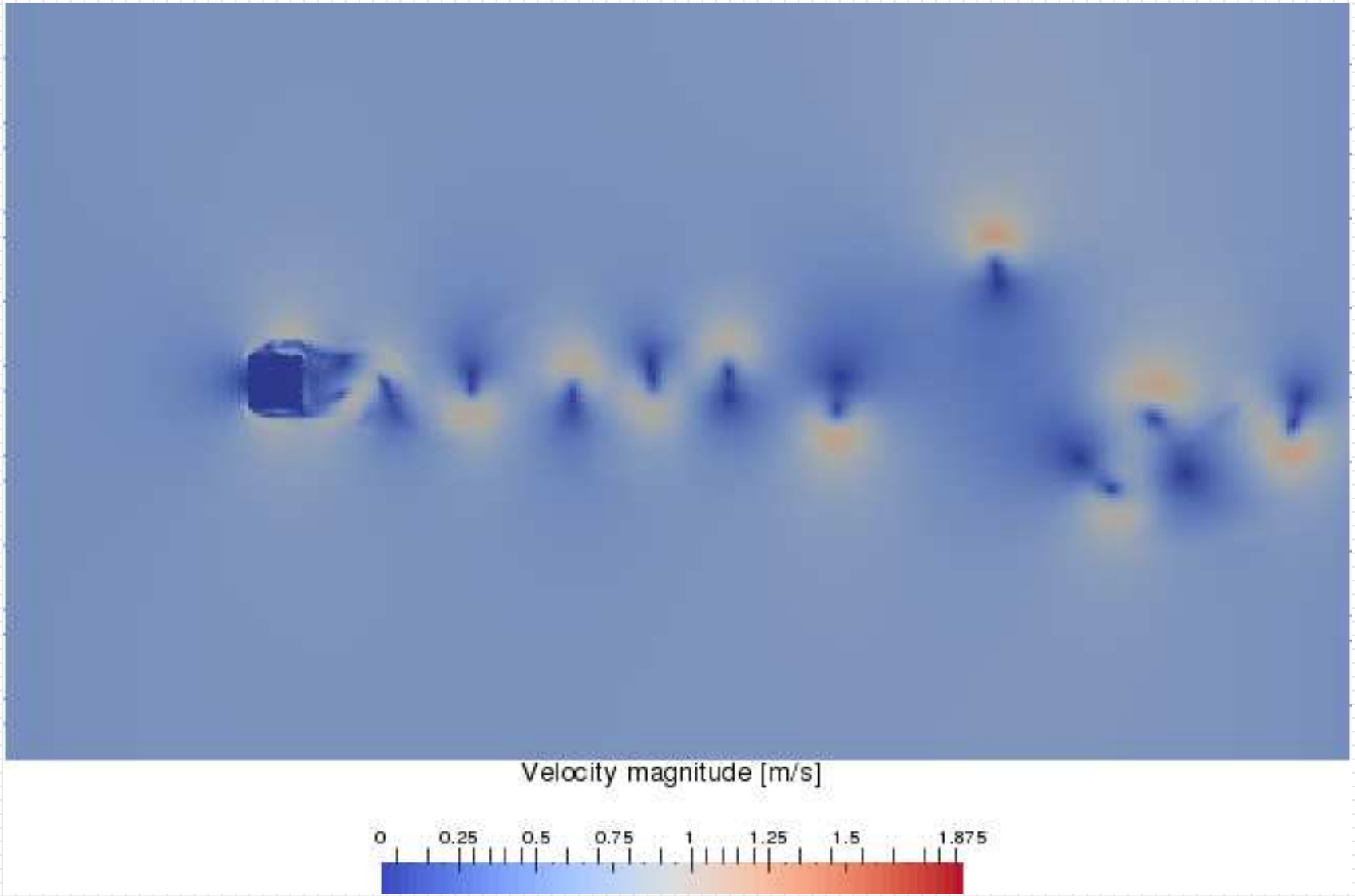
libmpdata++ 2.0: immersed boundary teaser



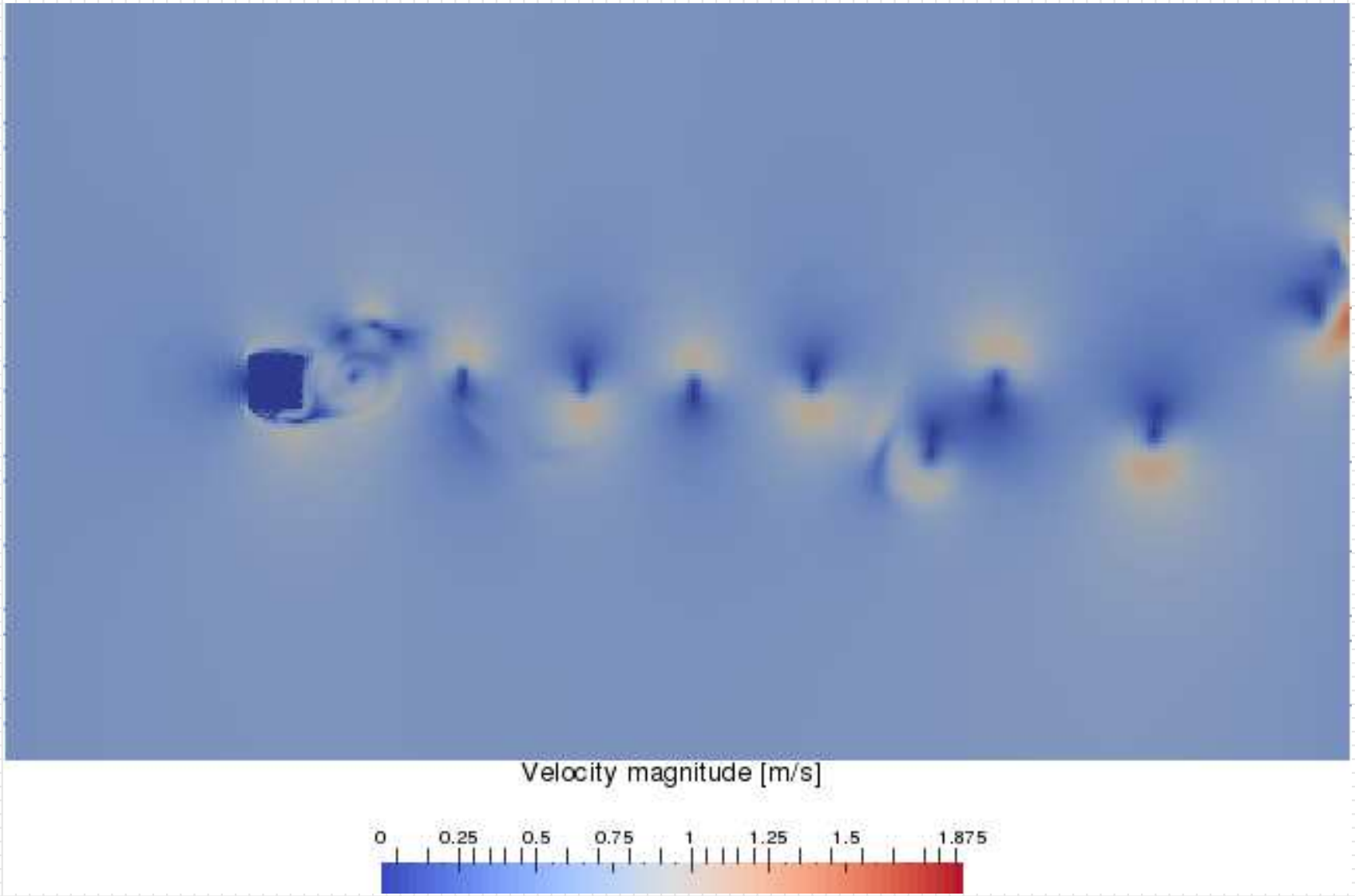
libmpdata++ 2.0: immersed boundary teaser



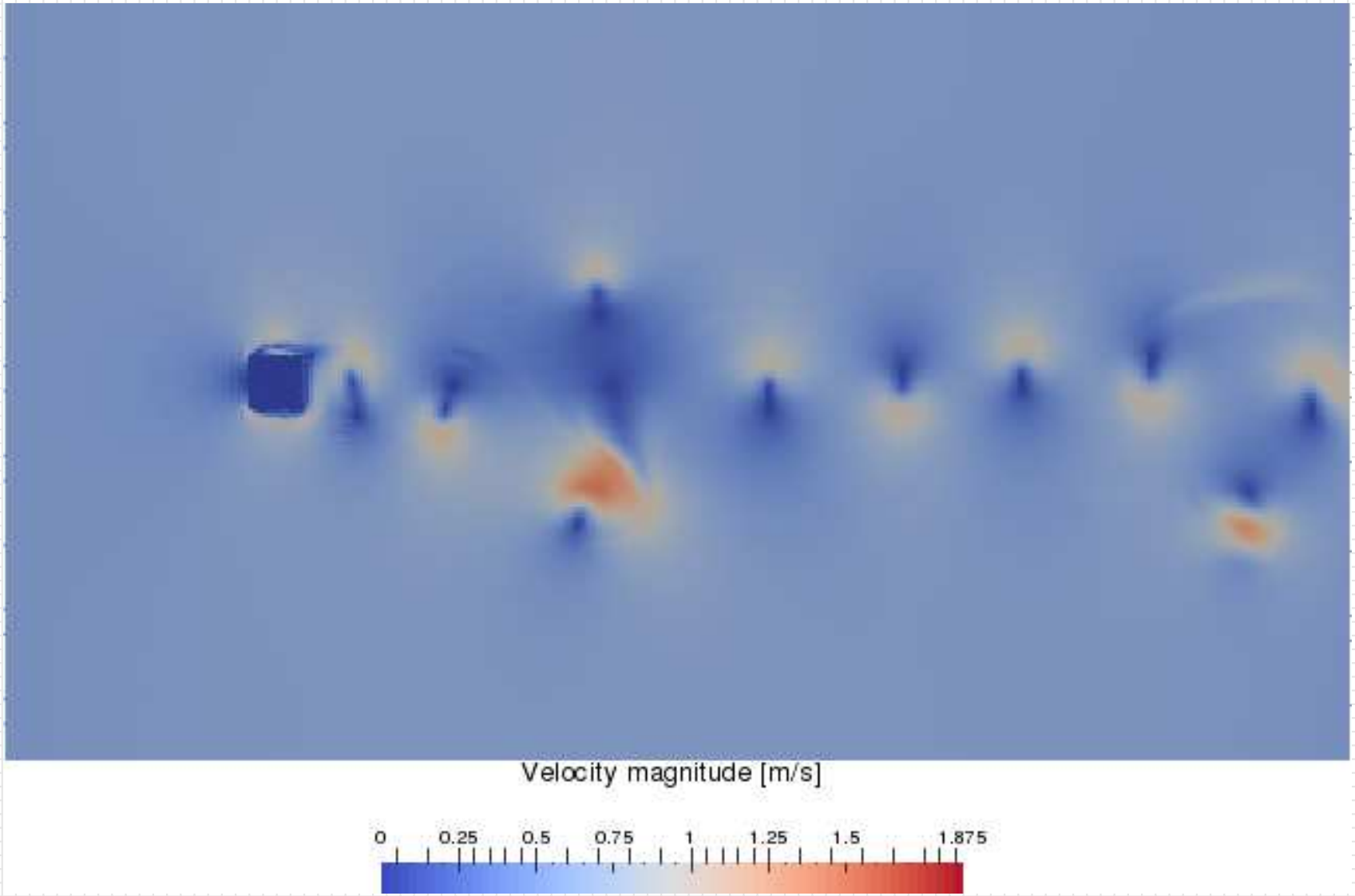
libmpdata++ 2.0: immersed boundary teaser



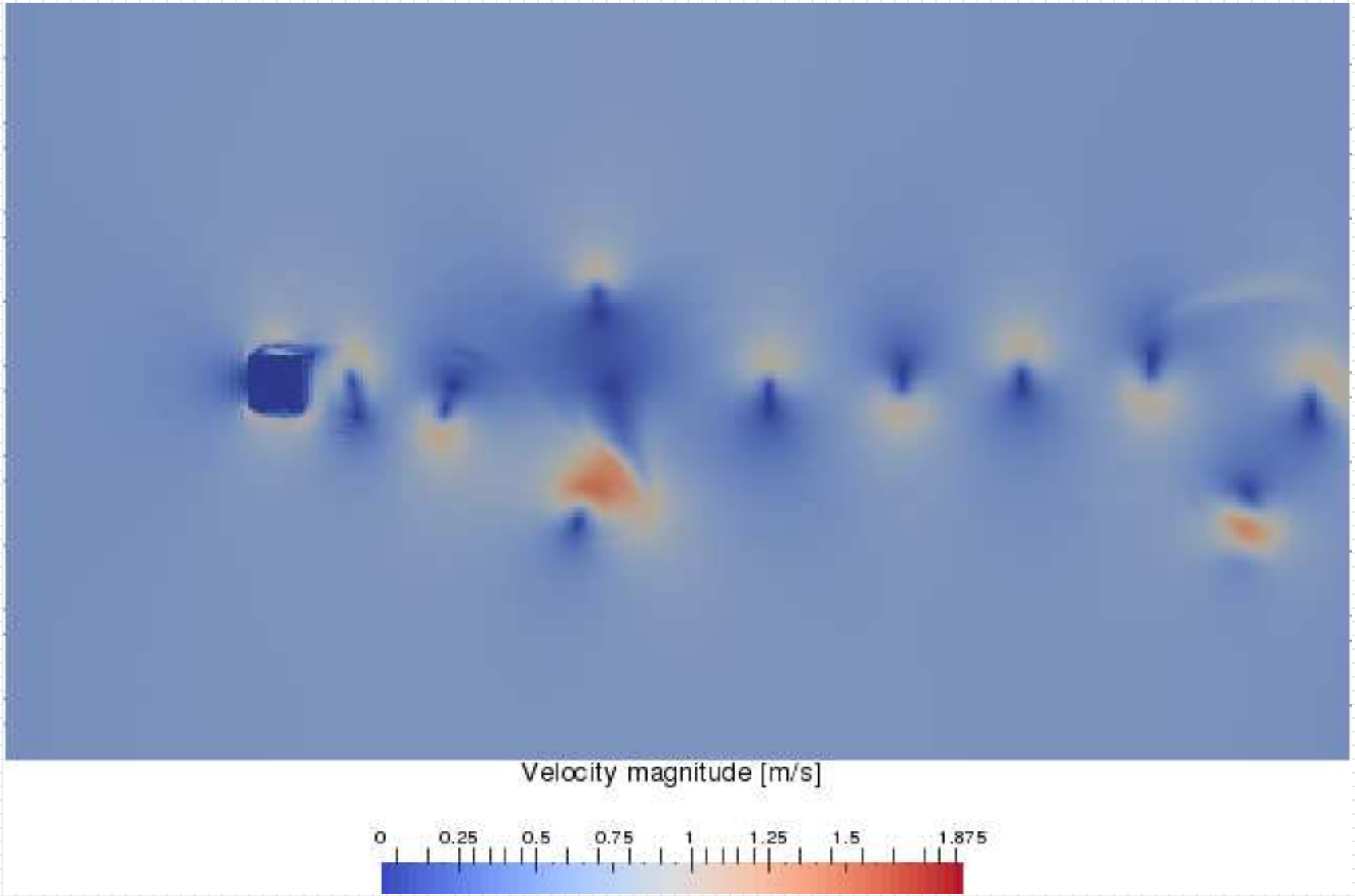
libmpdata++ 2.0: immersed boundary teaser



libmpdata++ 2.0: immersed boundary teaser



libmpdata++ 2.0: immersed boundary teaser



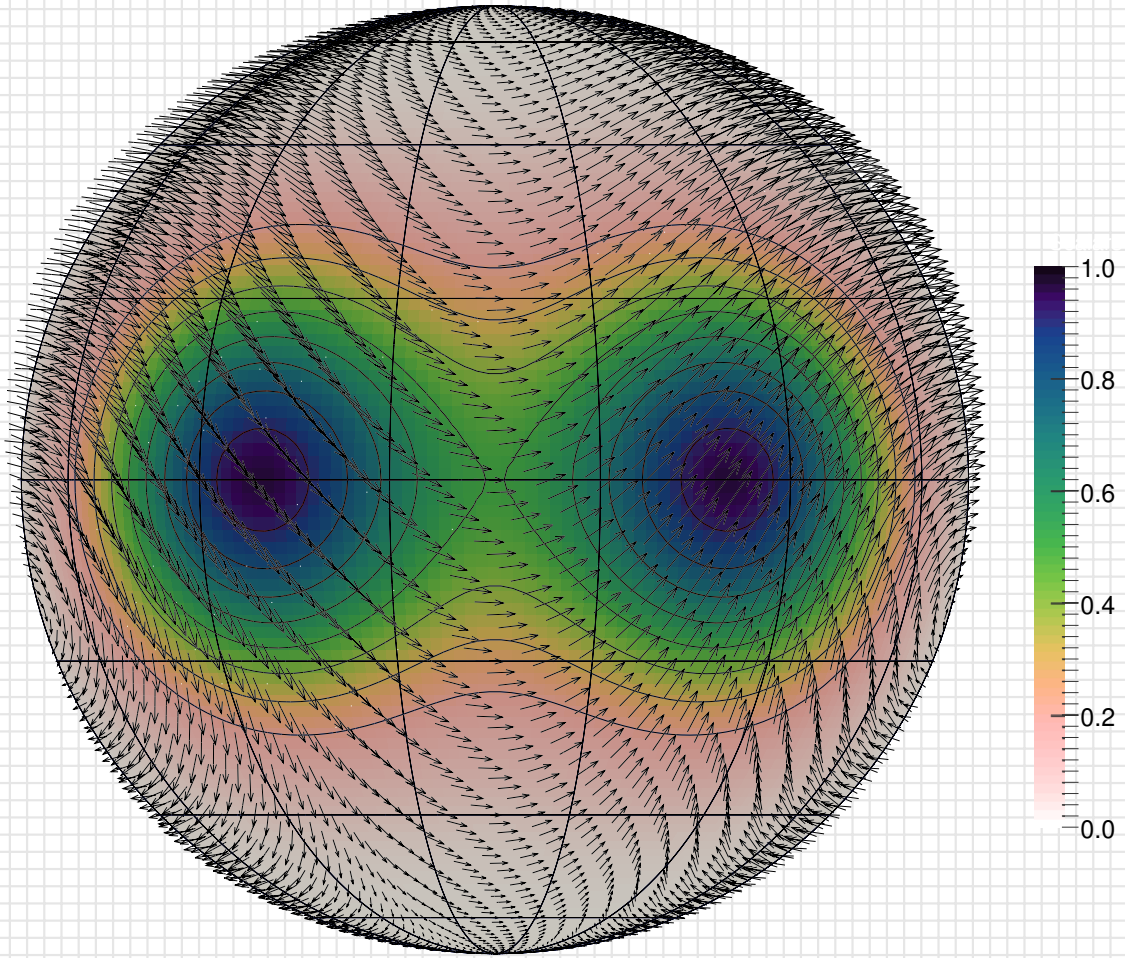
libmpdata++ 2.0: summary of features under development

- higher-order operators
  (for DNS/iLES simulations)
- implicit treatment of absorbers
  (for immersed-boundary method)
- adaptive timestepping
- distributed-memory parallelisation
  (using Boost.MPI & HDF5/MPI-IO)

libmpdata++ 2.0: summary of features under development

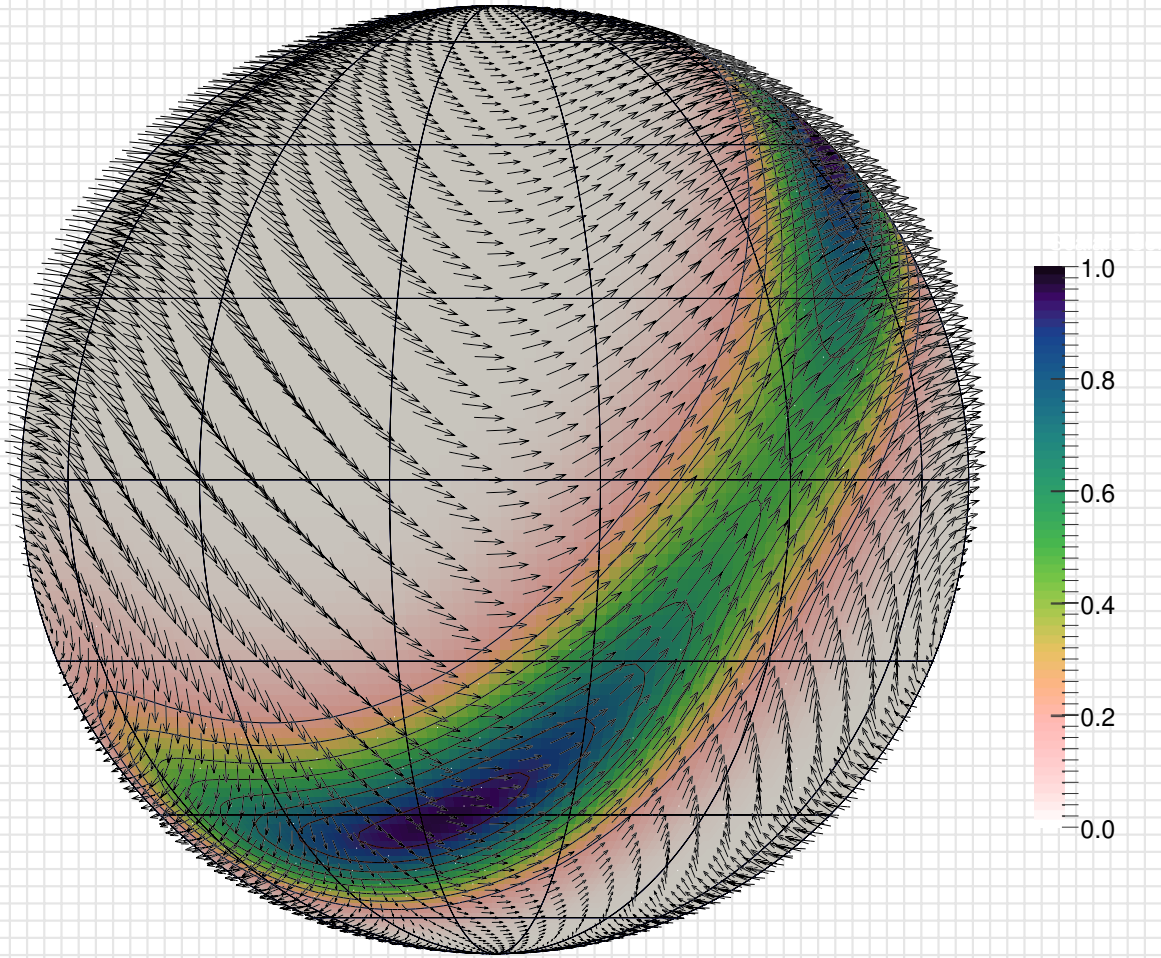
- higher-order operators
  (for DNS/iLES simulations)
- implicit treatment of absorbers
  (for immersed-boundary method)
- adaptive timestepping
- distributed-memory parallelisation
  (using Boost.MPI & HDF5/MPI-IO)

libmpdata++ 2.0: adaptive timestepping teaser



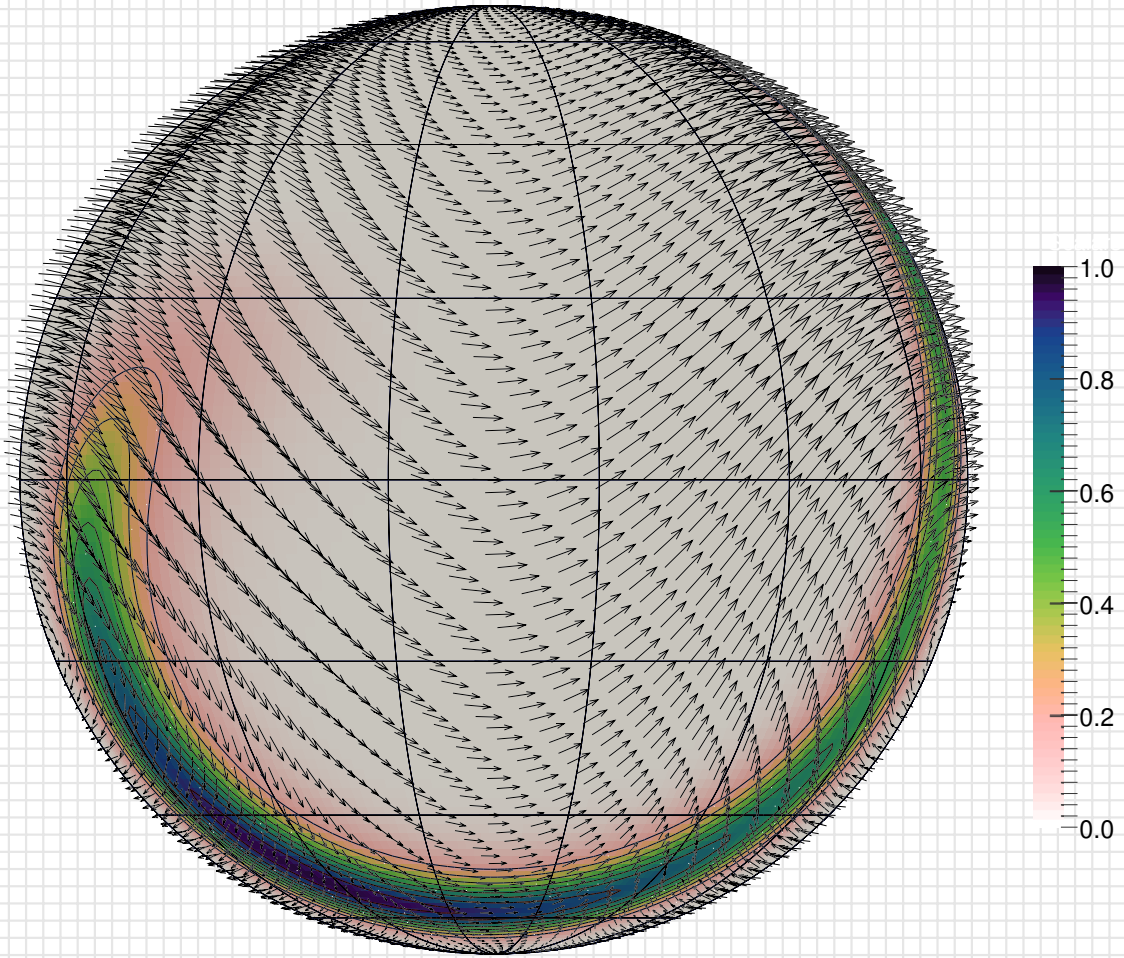
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



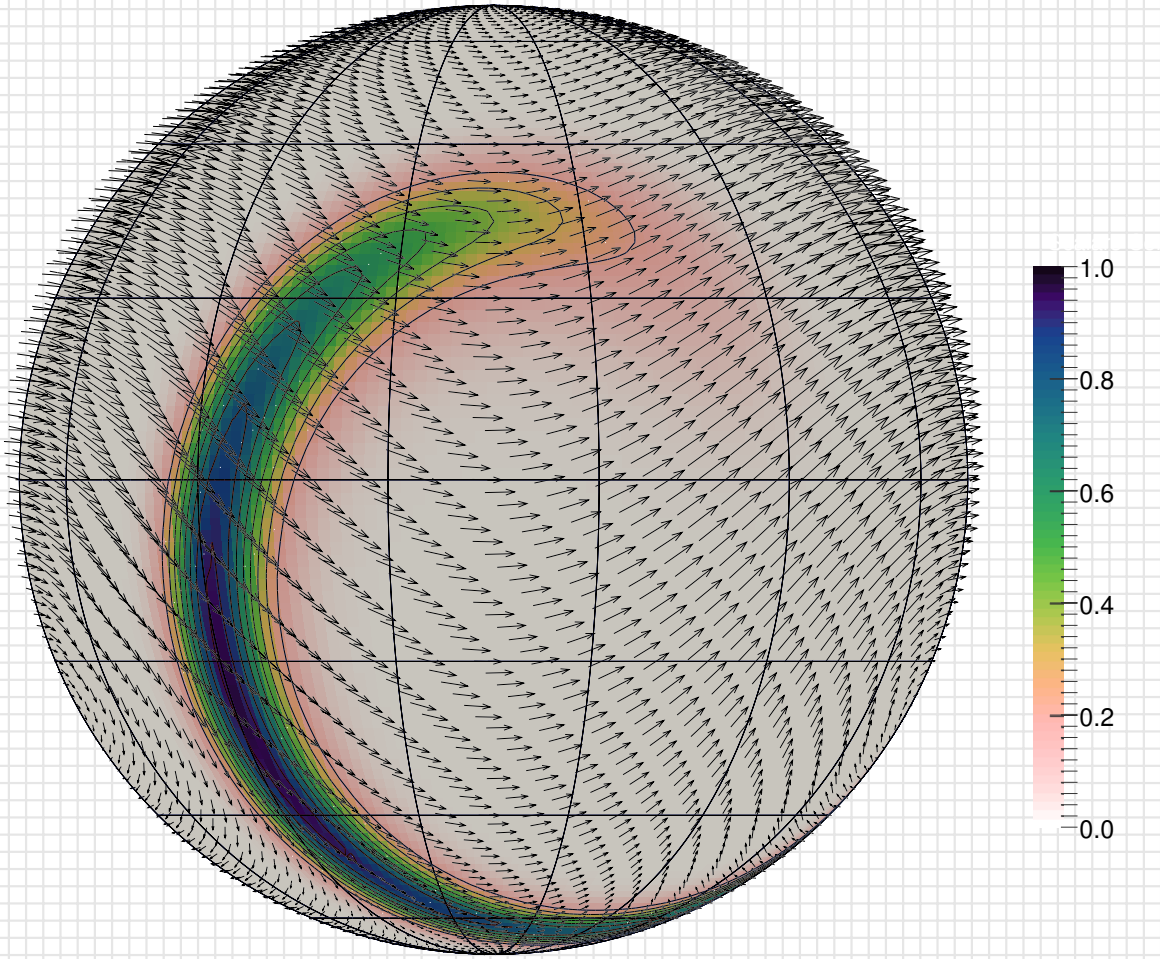
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



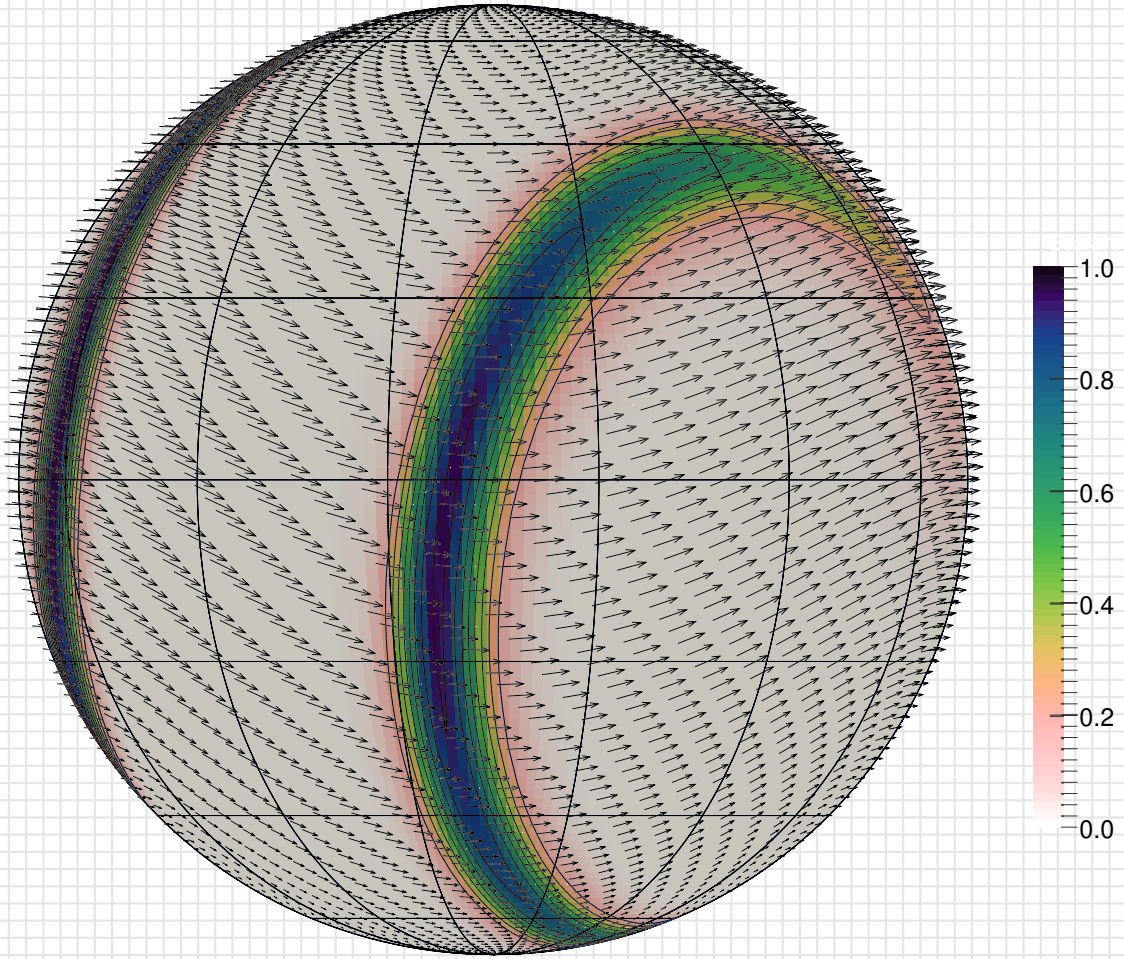
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



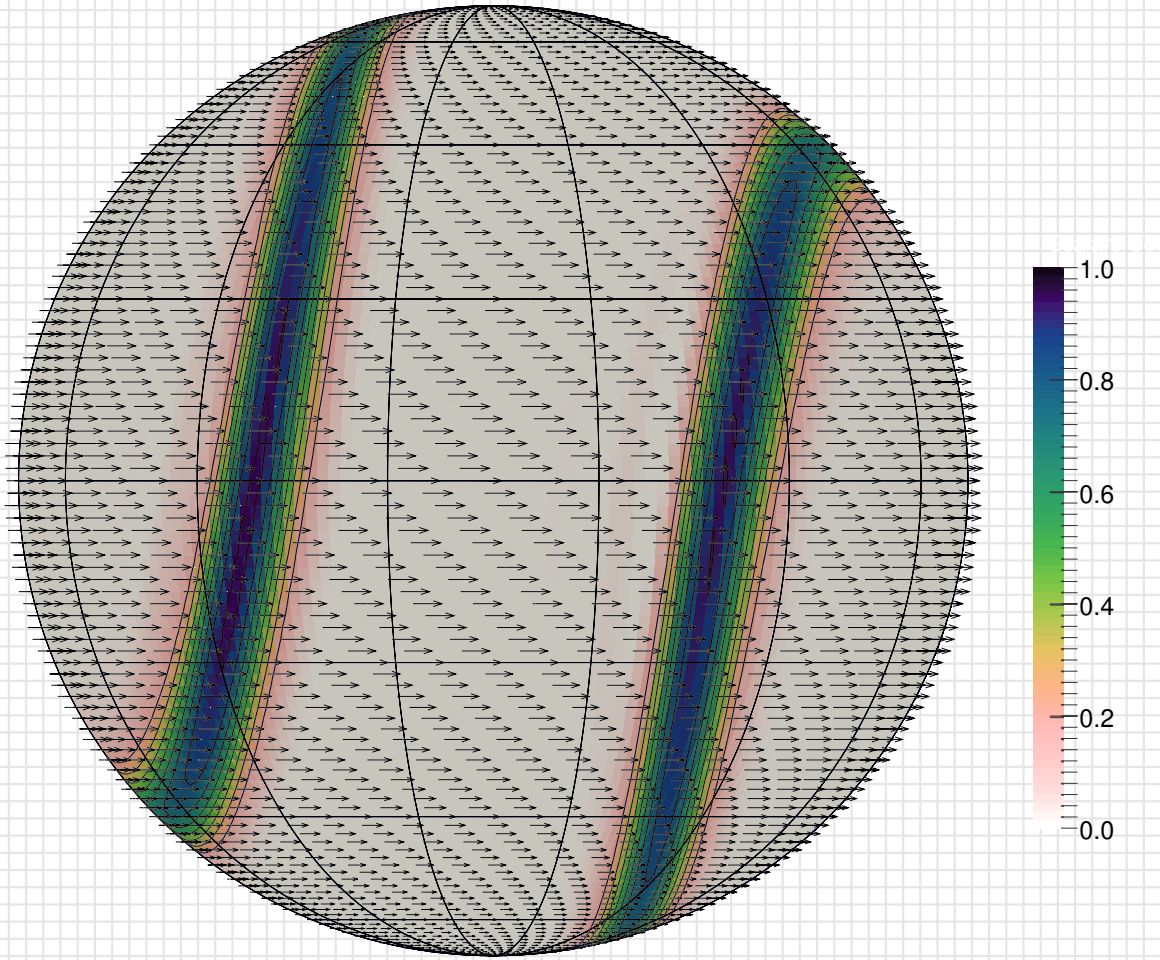
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



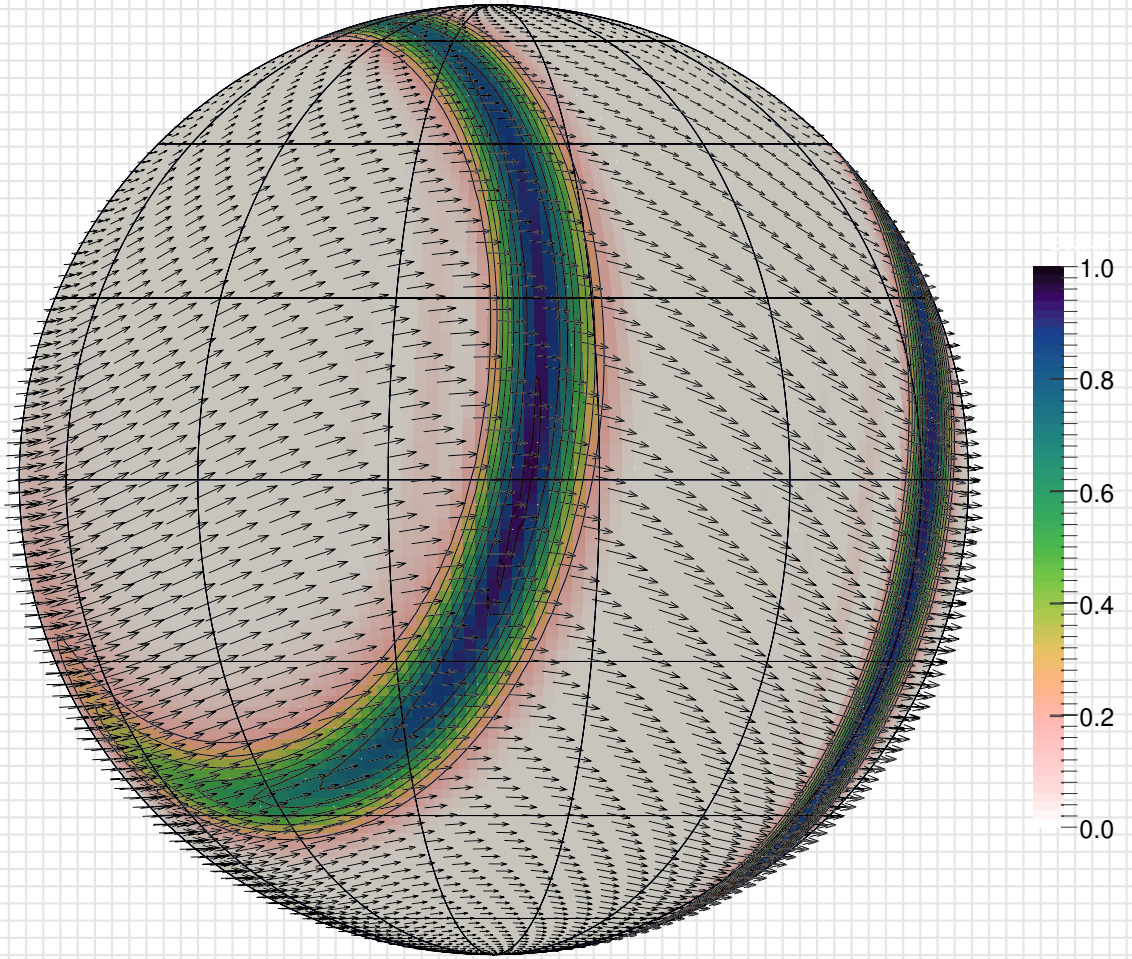
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



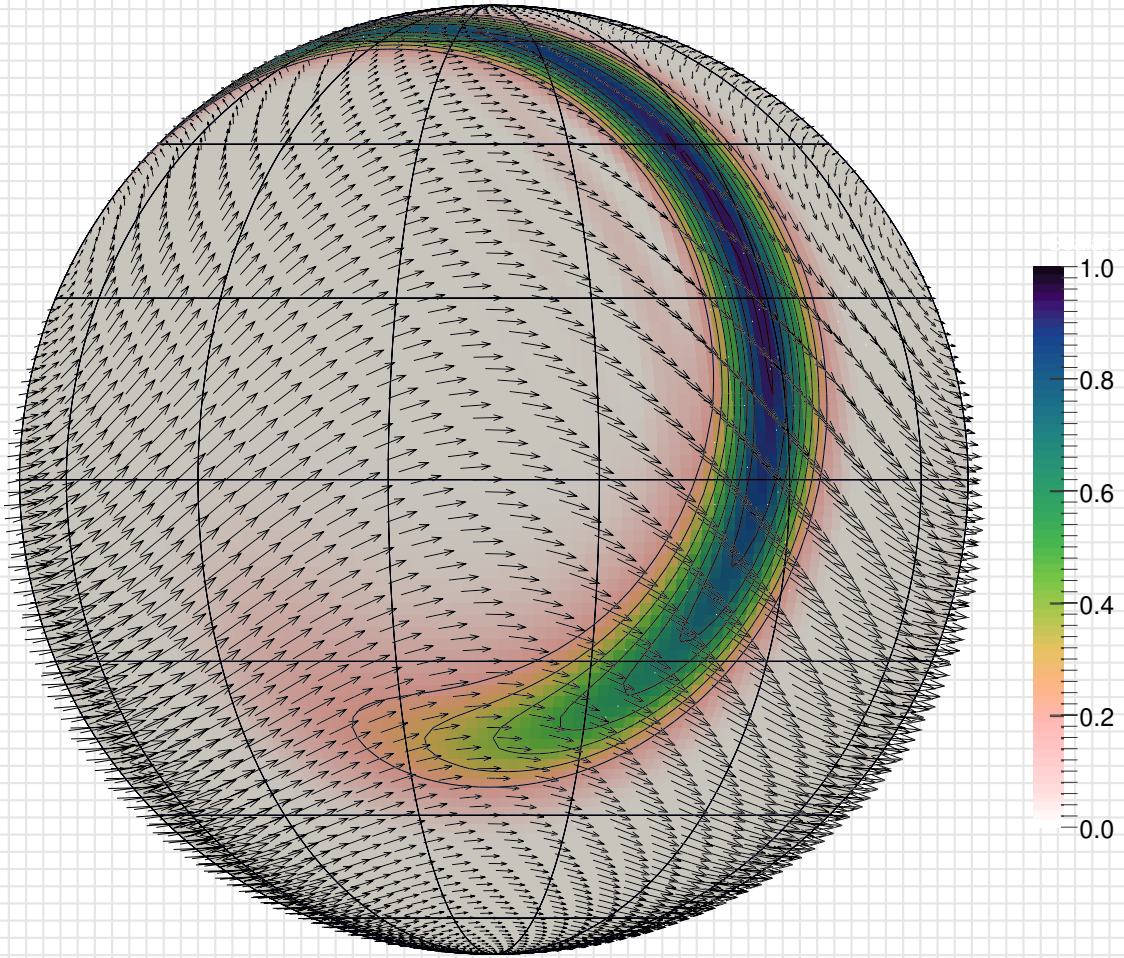
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



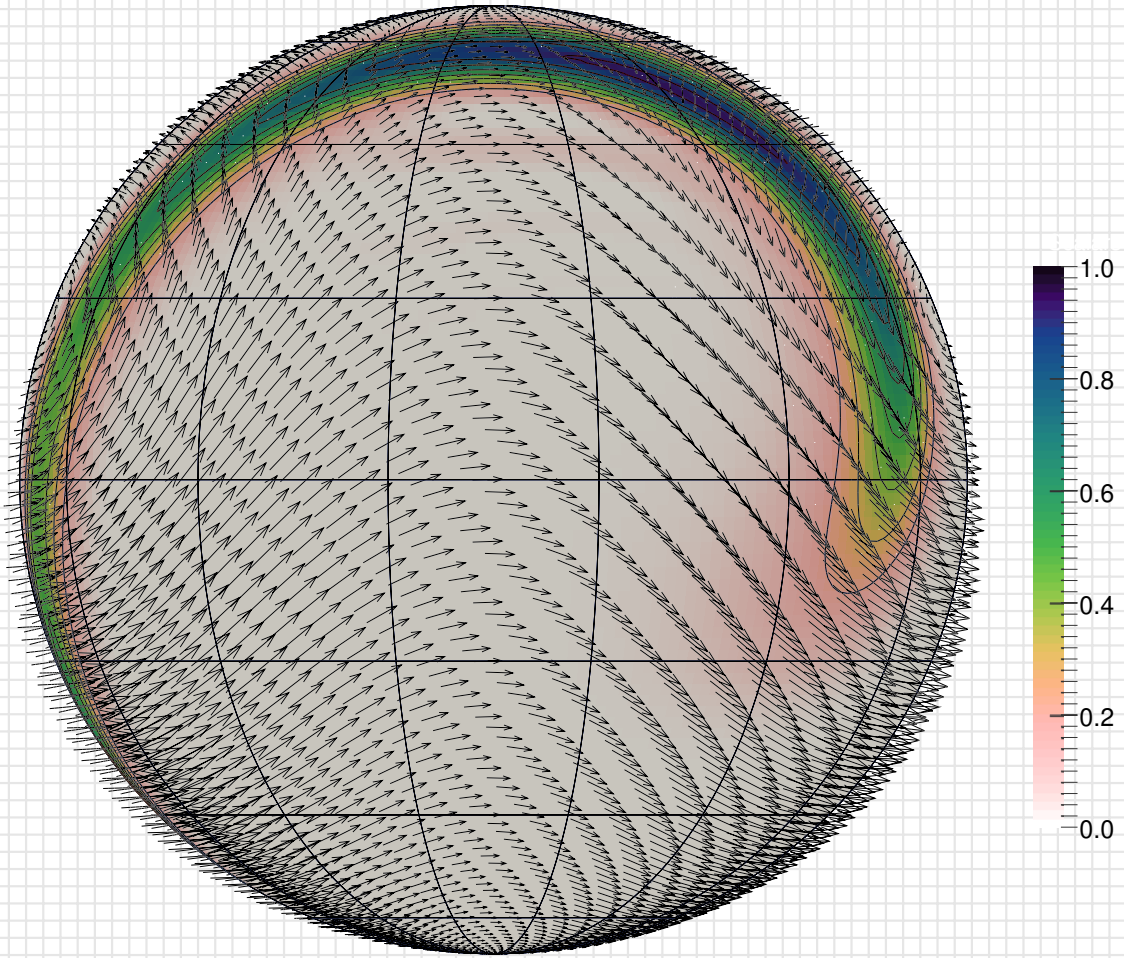
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



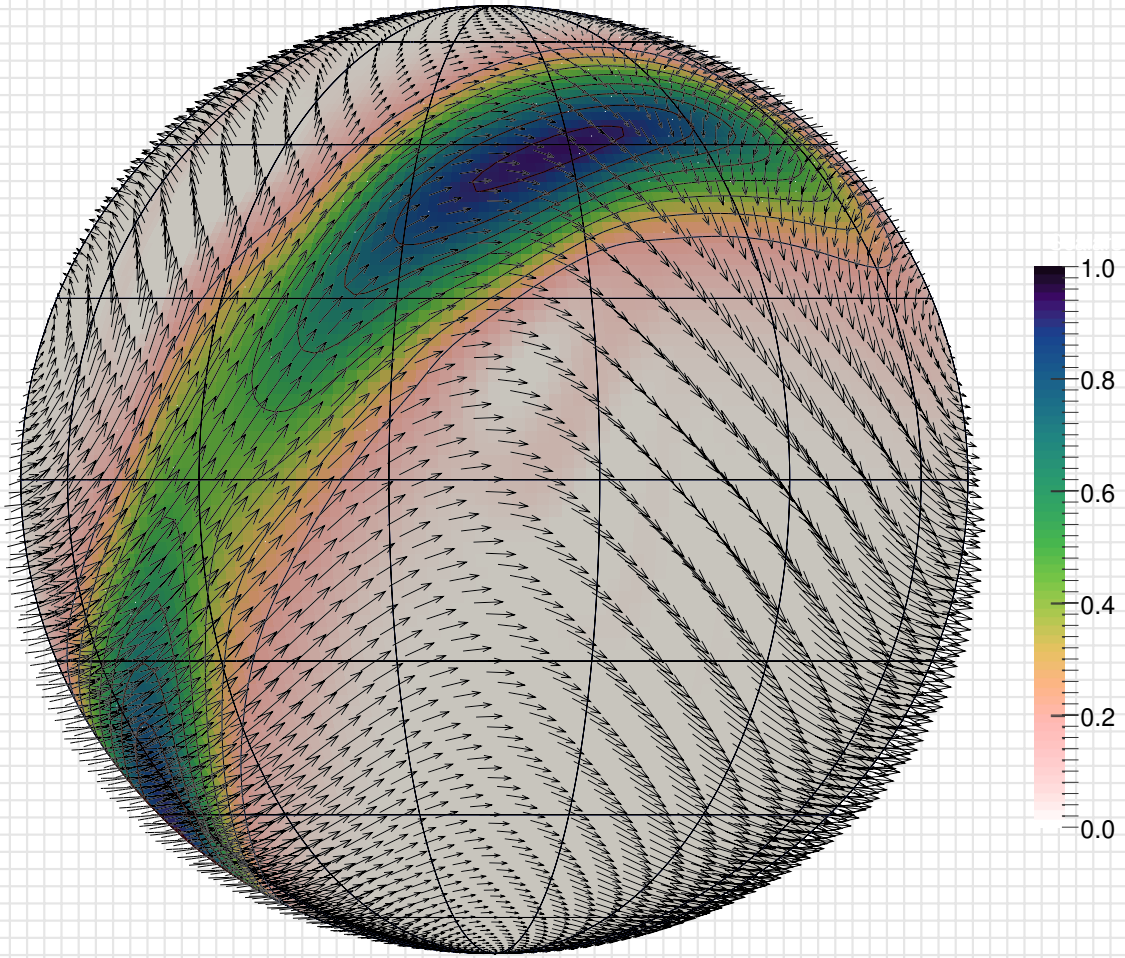
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



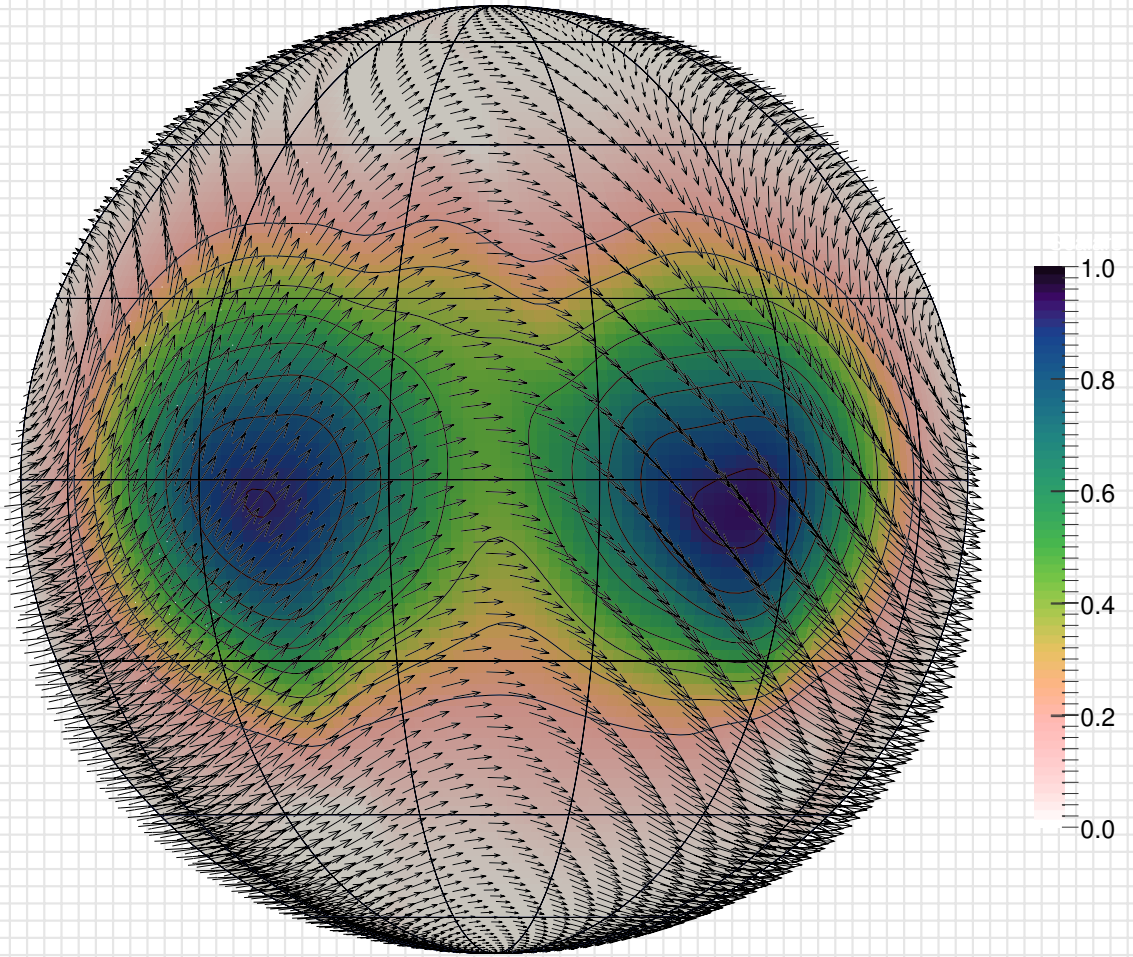
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



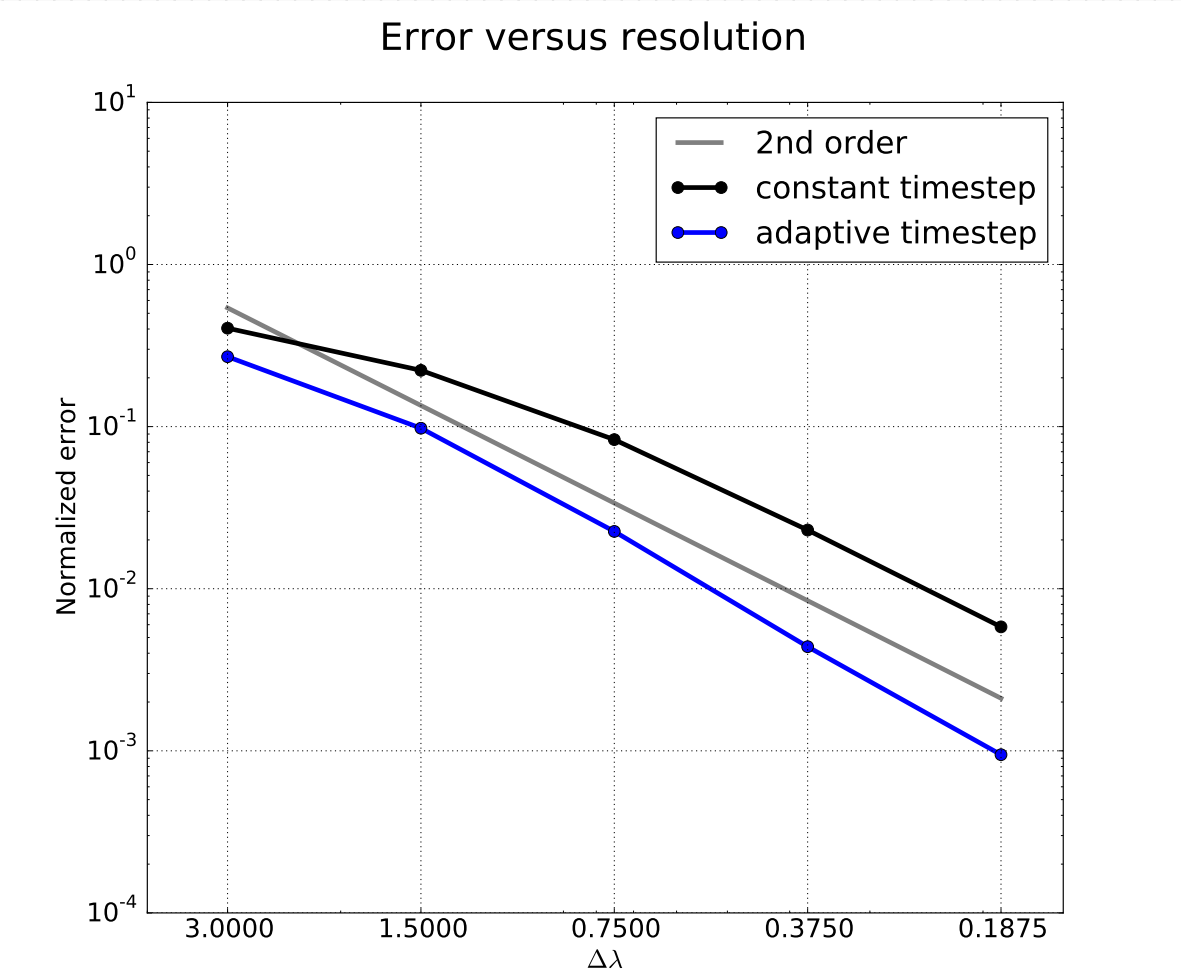
- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: adaptive timestepping teaser



- advection test by Nair and Lauritzen, 2010

libmpdata++ 2.0: summary of features under development

- higher-order operators
  (for DNS/iLES simulations)
- implicit treatment of absorbers
  (for immersed-boundary method)
- adaptive timestepping
- distributed-memory parallelisation
  (using Boost.MPI & HDF5/MPI-IO)

libmpdata++ 2.0: summary of features under development

- higher-order operators
  (for DNS/iLES simulations)
- implicit treatment of absorbers
  (for immersed-boundary method)
- adaptive timestepping
- distributed-memory parallelisation
  (using Boost.MPI & HDF5/MPI-IO)

libmpdata++ 2.0: summary of features under development

- higher-order operators
  (for DNS/iLES simulations)
- implicit treatment of absorbers
  (for immersed-boundary method)
- adaptive timestepping
- distributed-memory parallelisation
  (using Boost.MPI & HDF5/MPI-IO)

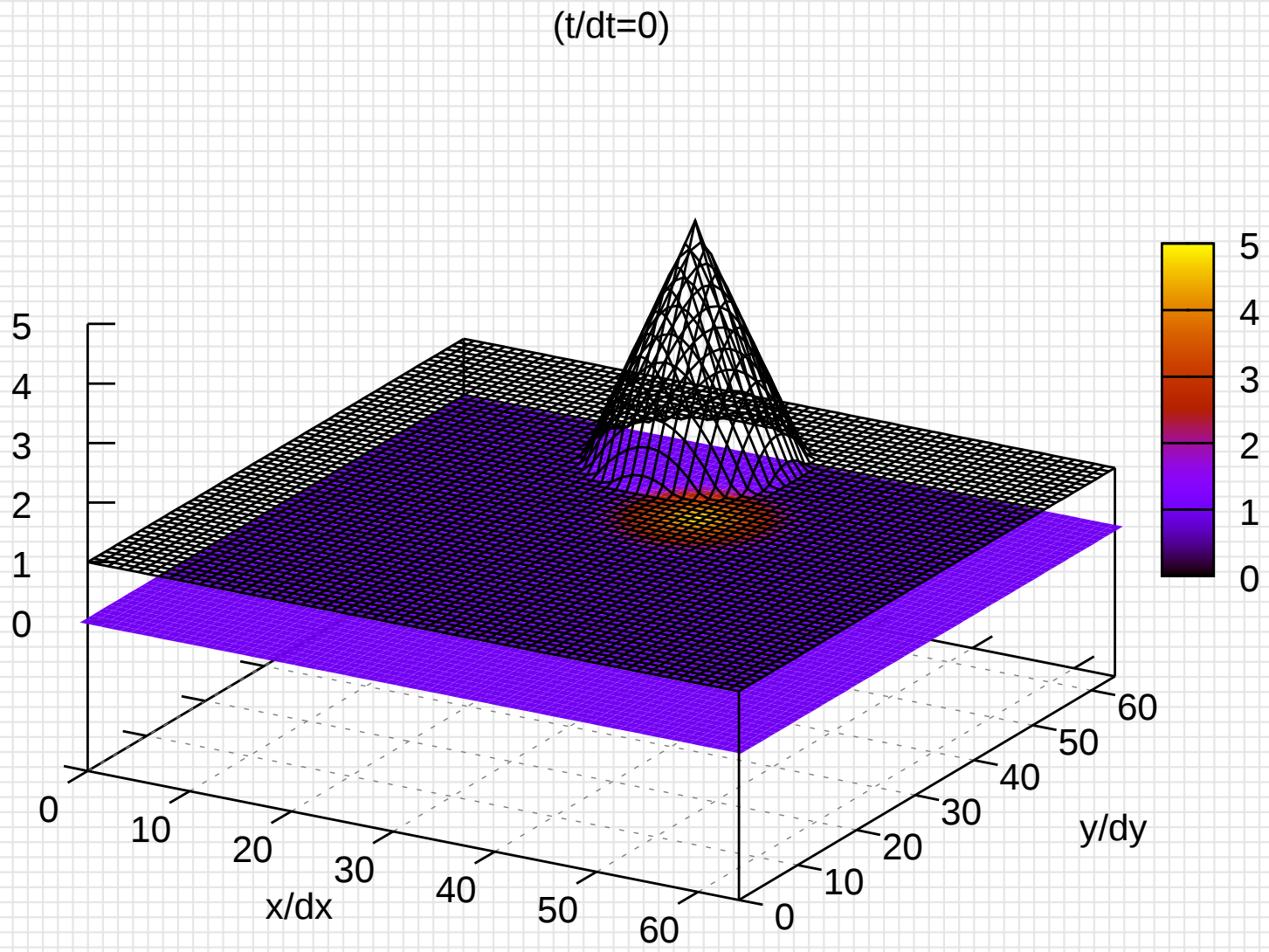
coded by
Maciej Waruszewski

Plan of the talk

1. what's libmpdata++
2. libmpdata++ 1.0: summary of features
3. libmpdata++ 2.0: new features under development
4. libmpdata++: a hello-world program
5. closing remarks

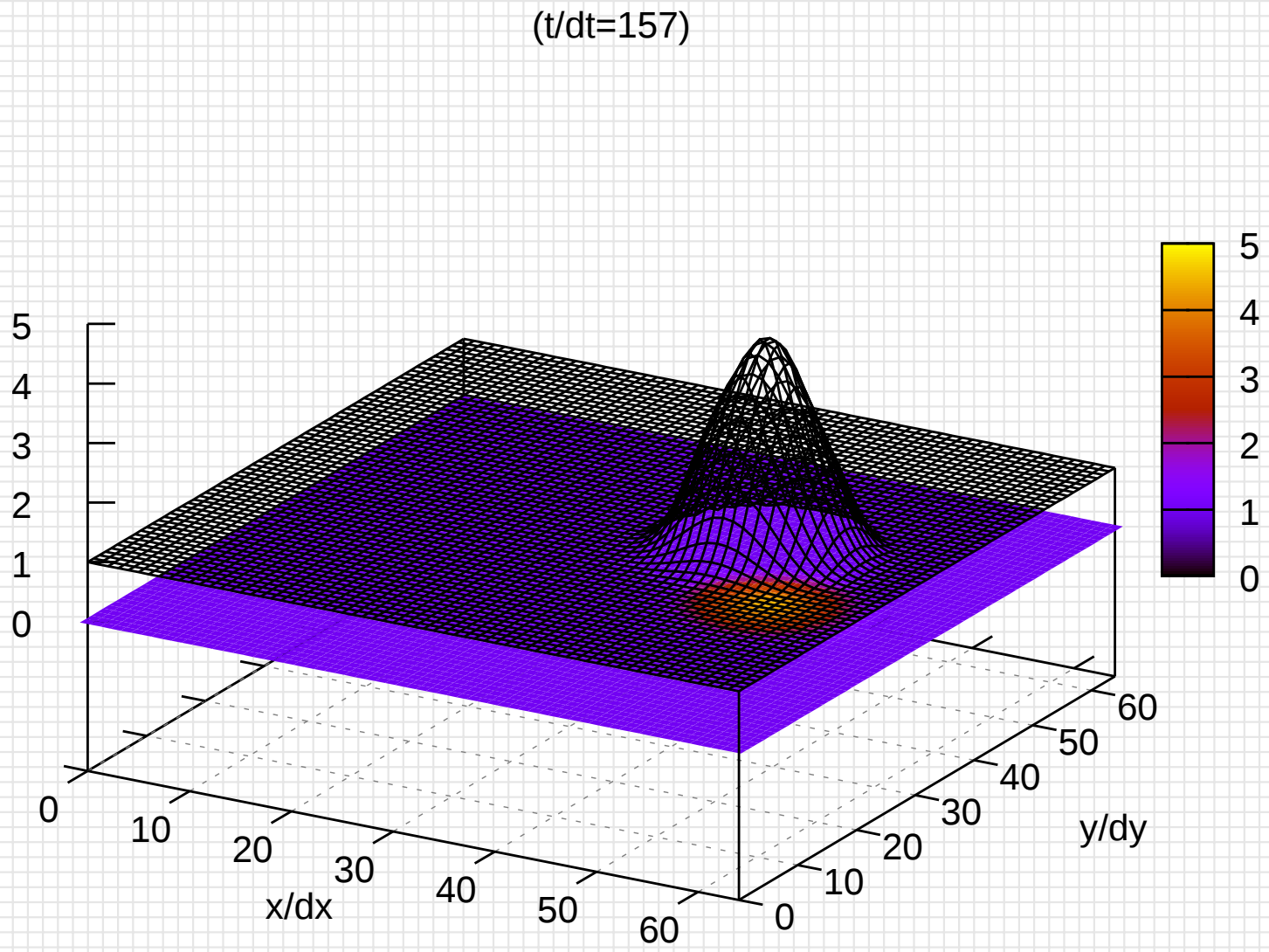
Plan of the talk

libmpdata++: rotating cone test



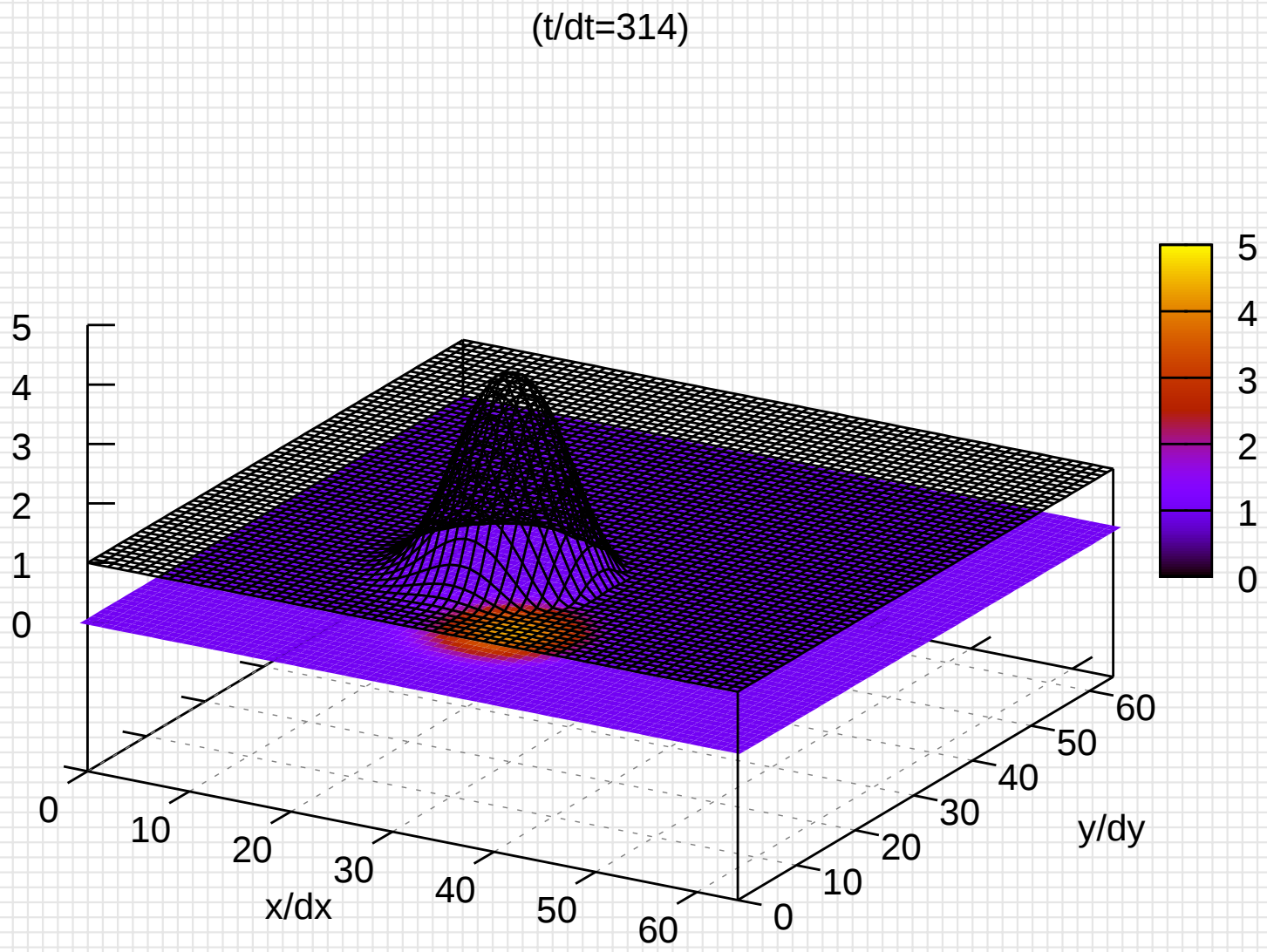
64 LOC using libmpdata++

libmpdata++: rotating cone test



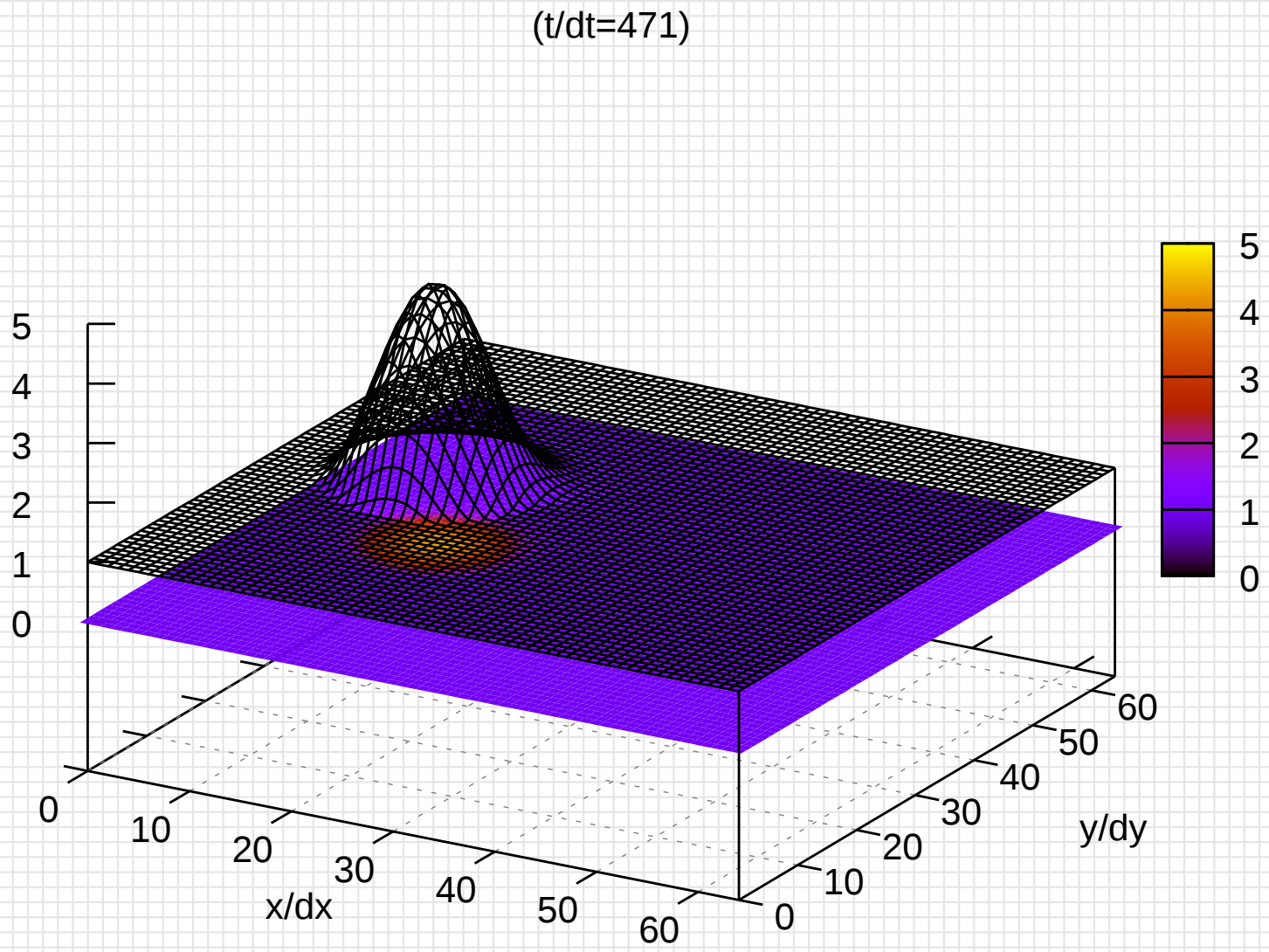
64 LOC using libmpdata++

libmpdata++: rotating cone test



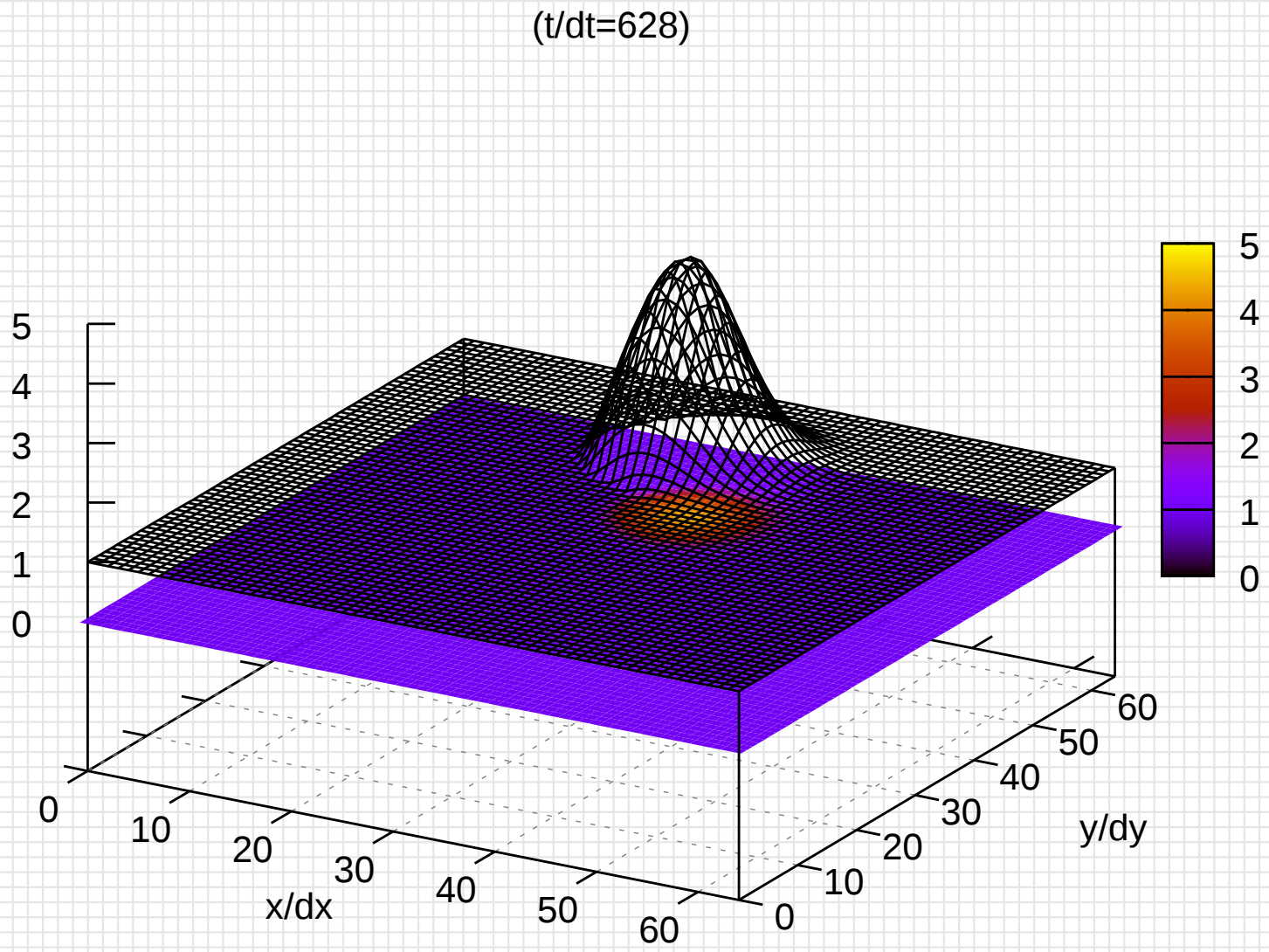
64 LOC using libmpdata++

libmpdata++: rotating cone test



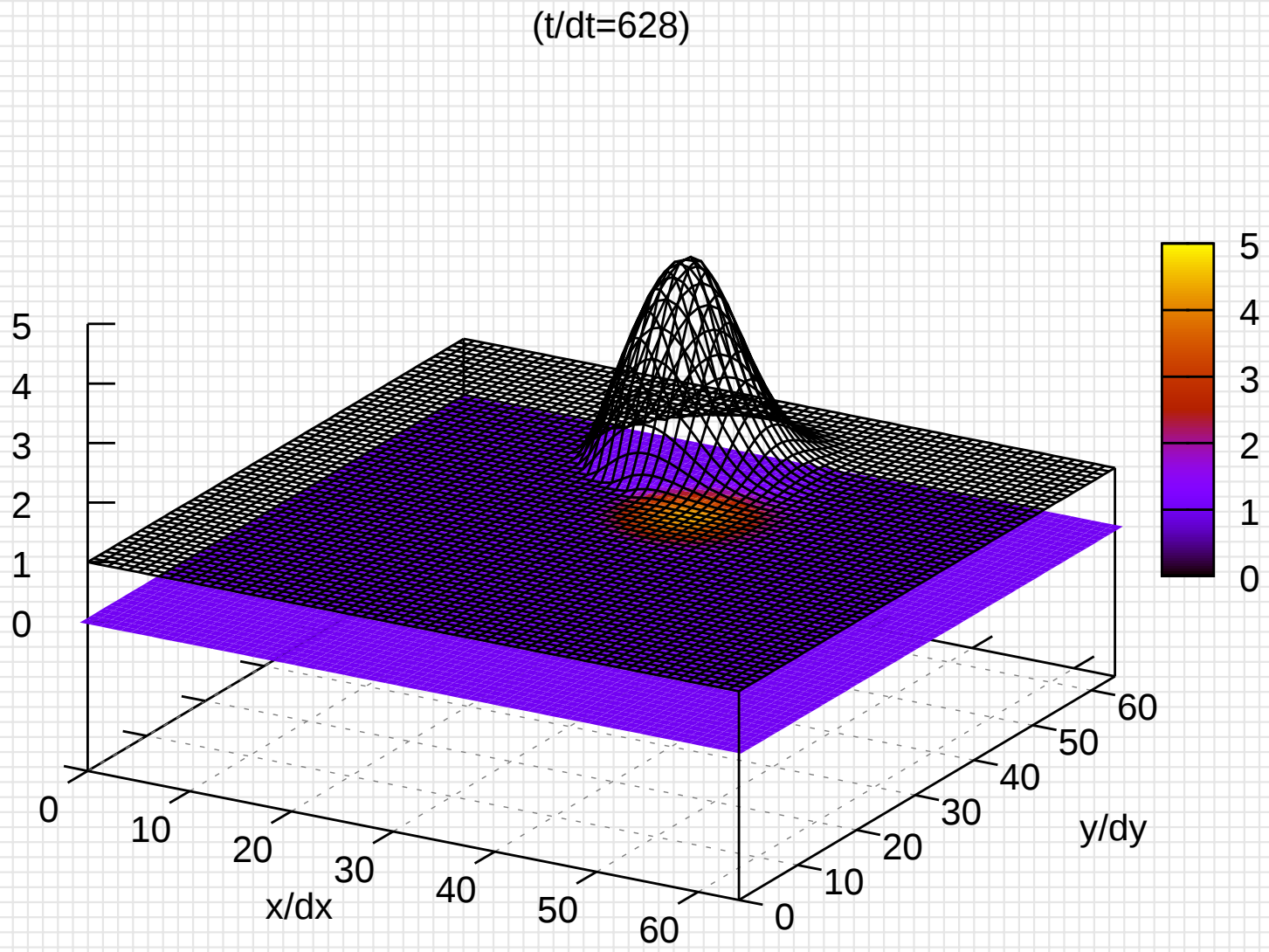
64 LOC using libmpdata++

libmpdata++: rotating cone test



64 LOC using libmpdata++

libmpdata++: rotating cone test



64 LOC using libmpdata++

```cpp
#include <libmpdata++/solvers/mpdata.hpp>
#include <libmpdata++/concurr/serial.hpp>
#include <libmpdata++/output/gnuplot.hpp>

int main()
{
  namespace lmpdt = libmpdataxx;
  const int nx=64, ny=64, nt = 628;

  // compile-time parameters
  struct ct_params_t : lmpdt::ct_params_default_t
  {
    using real_t = double;
    enum { n_dims = 2 };
    enum { n_eqns = 1 };
  };

  // solver choice
  using run_t = lmpdt::output::gnuplot< lmpdt::solvers::mpdata< ct_params_t >>;

  // runtime parameters
  typename run_t::rt_params_t p;
  p.grid_size = {nx+1, ny+1};
  p.outfreq = nt/4;
  p.gnuplot_output = "out_%s_%d.svg";
  p.gnuplot_with = "lines";
  p.gnuplot_cbrange = p.gnuplot_zrange = "[0:5]";

  // sharedmem concurency and boundary condition choice
  lmpdt::concurr::serial<
    run_t,
    lmpdt::bcond::open, lmpdt::bcond::open, // x-left, x-right
    lmpdt::bcond::open, lmpdt::bcond::open  // y-left, y-right
  > run(p);
```

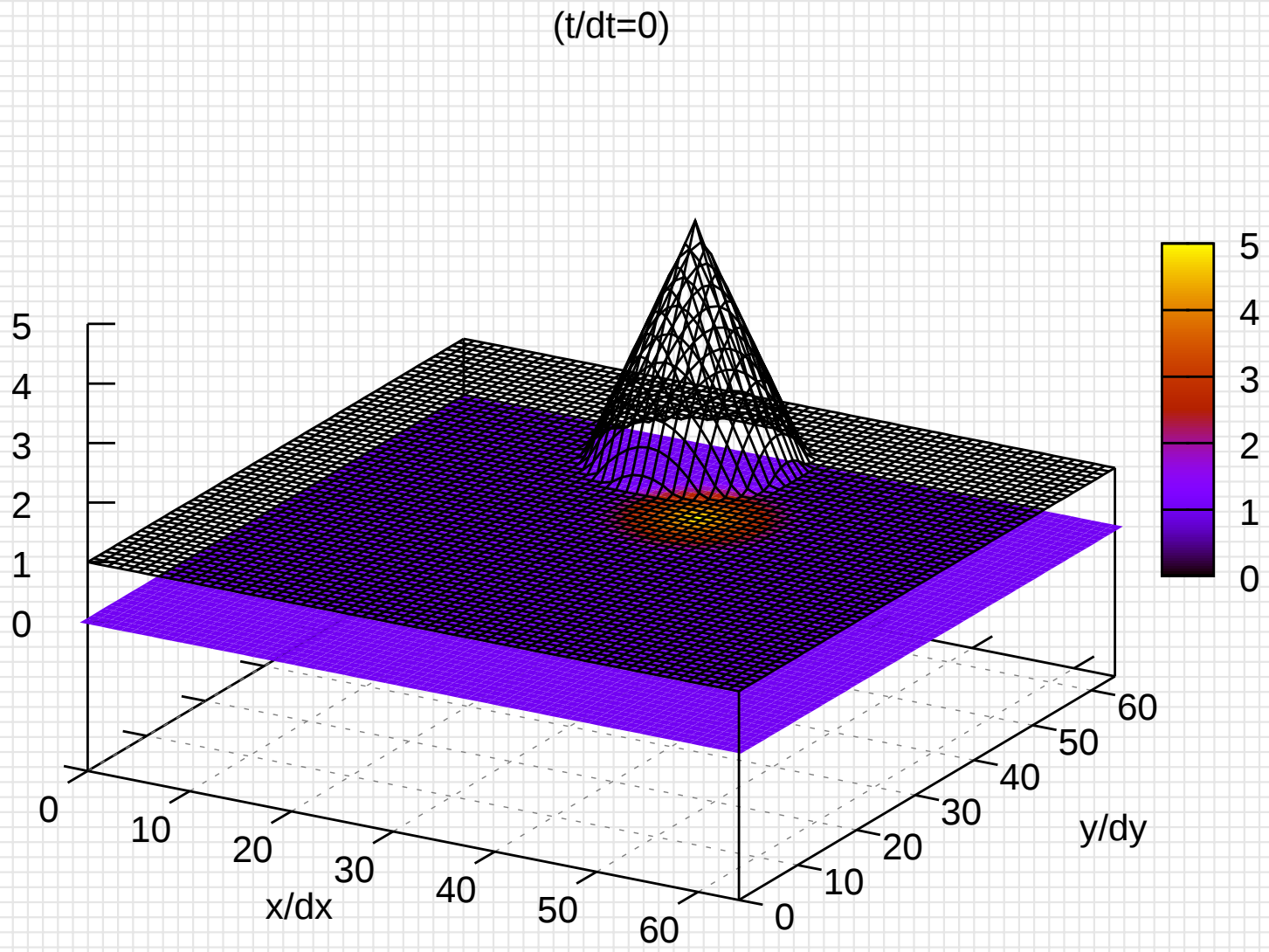
```
35
36   // initial condition
37   {
38     using namespace blitz::tensor;
39     auto psi = run.advectee();
40
41     const double
42       dt = .1, dx = 1, dy = 1, omega = .1,
43       h = 4., h0 = 1, r = .15 * nx * dx,
44       x0 = .5 * nx * dx, y0 = .75 * ny * dy,
45       xc = .5 * nx * dx, yc = .50 * ny * dy;
46
47     // cone shape cut at h0
48     psi = blitz::pow(i * dx - x0, 2) +
49           blitz::pow(j * dy - y0, 2);
50
51     psi = h0 + where(
52       psi - pow(r, 2) <= 0,                 // if
53       h - blitz::sqrt(psi / pow(r/h,2)),    // then
54       0.                                    // else
55     );
56
57     // constant-angular-velocity rotational field
58     run.advector(0) =  omega * (j * dy - yc) * dt/dx;
59     run.advector(1) = -omega * (i * dx - xc) * dt/dy;
60   }
61
62   // time stepping
63   run.advance(nt);
64 }
```

```
35
36   // initial condition
37   {
38     using namespace blitz::tensor;
39     auto psi = run.advectee();
40
41     const double
42       dt = .1, dx = 1, dy = 1, omega = .1,
43       h = 4., h0 = 1, r = .15 * nx * dx,
44       x0 = .5 * nx * dx, y0 = .75 * ny * dy,
45       xc = .5 * nx * dx, yc = .50 * ny * dy;
46
47     // cone shape cut at h0
48     psi = blitz::pow(i * dx - x0, 2) +
49           blitz::pow(j * dy - y0, 2);
50
51     psi = h0 + where(
52       psi - pow(r, 2) <= 0,                 // if
53       h - blitz::sqrt(psi / pow(r/h,2)),    // then
54       0.                                    // else
55     );
56
57     // constant-angular-velocity rotational field
58     run.advector(0) =  omega * (j * dy - yc) * dt/dx;
59     run.advector(1) = -omega * (i * dx - xc) * dt/dy;
60   }
61
62   // time stepping
63   run.advance(nt);
64 }
```

CMakeLists.txt

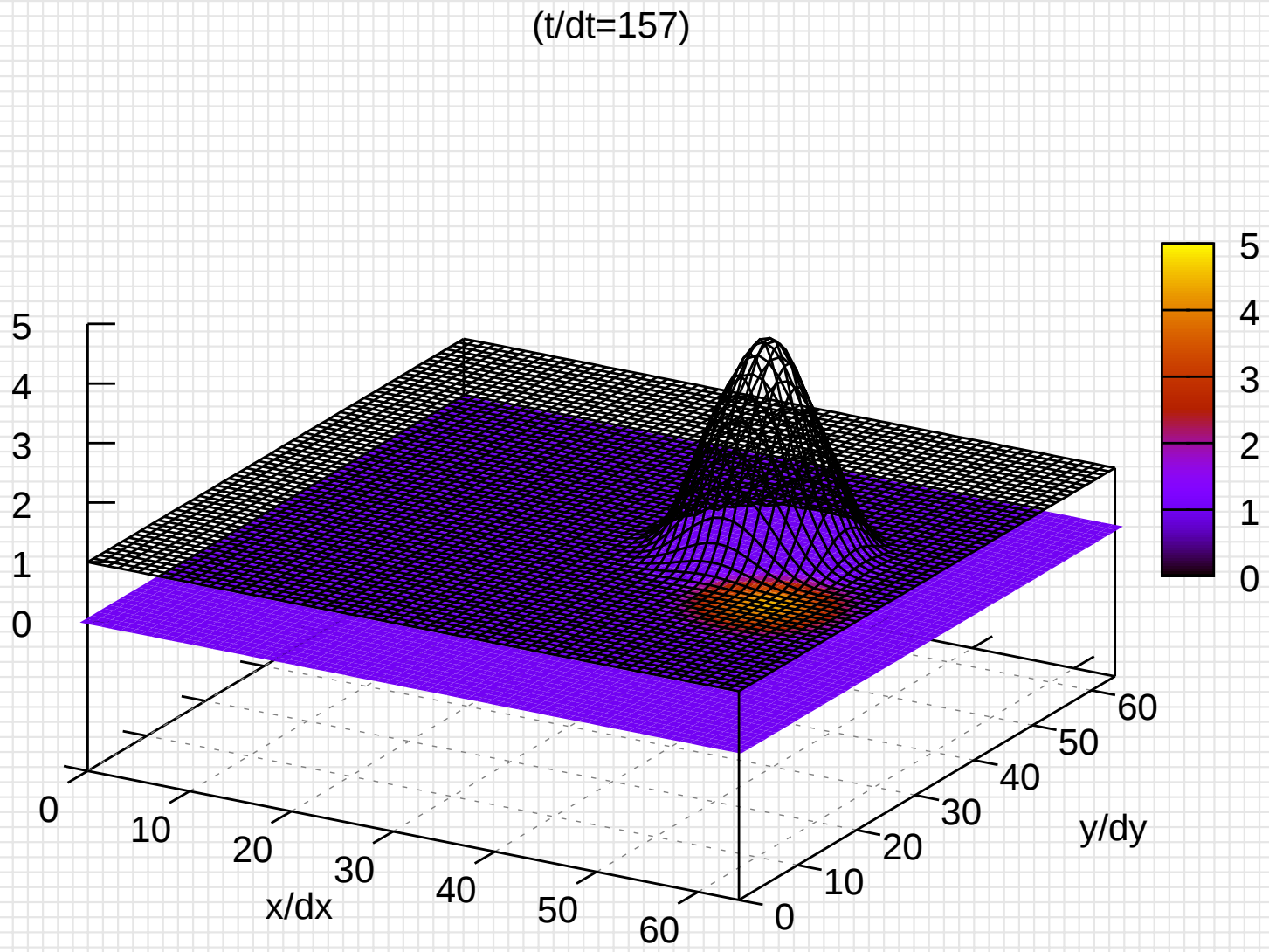
```
1 cmake_minimum_required(VERSION 3.0)
2 project(hello_world CXX)
3 find_package(libmpdata++)
4 set(CMAKE_CXX_FLAGS ${libmpdataxx_CXX_FLAGS_RELEASE})
5 add_executable(hello_world hello_world.cpp)
6 target_link_libraries(hello_world ${libmpdataxx_LIBRARIES})
```

libmpdata++: rotating cone test



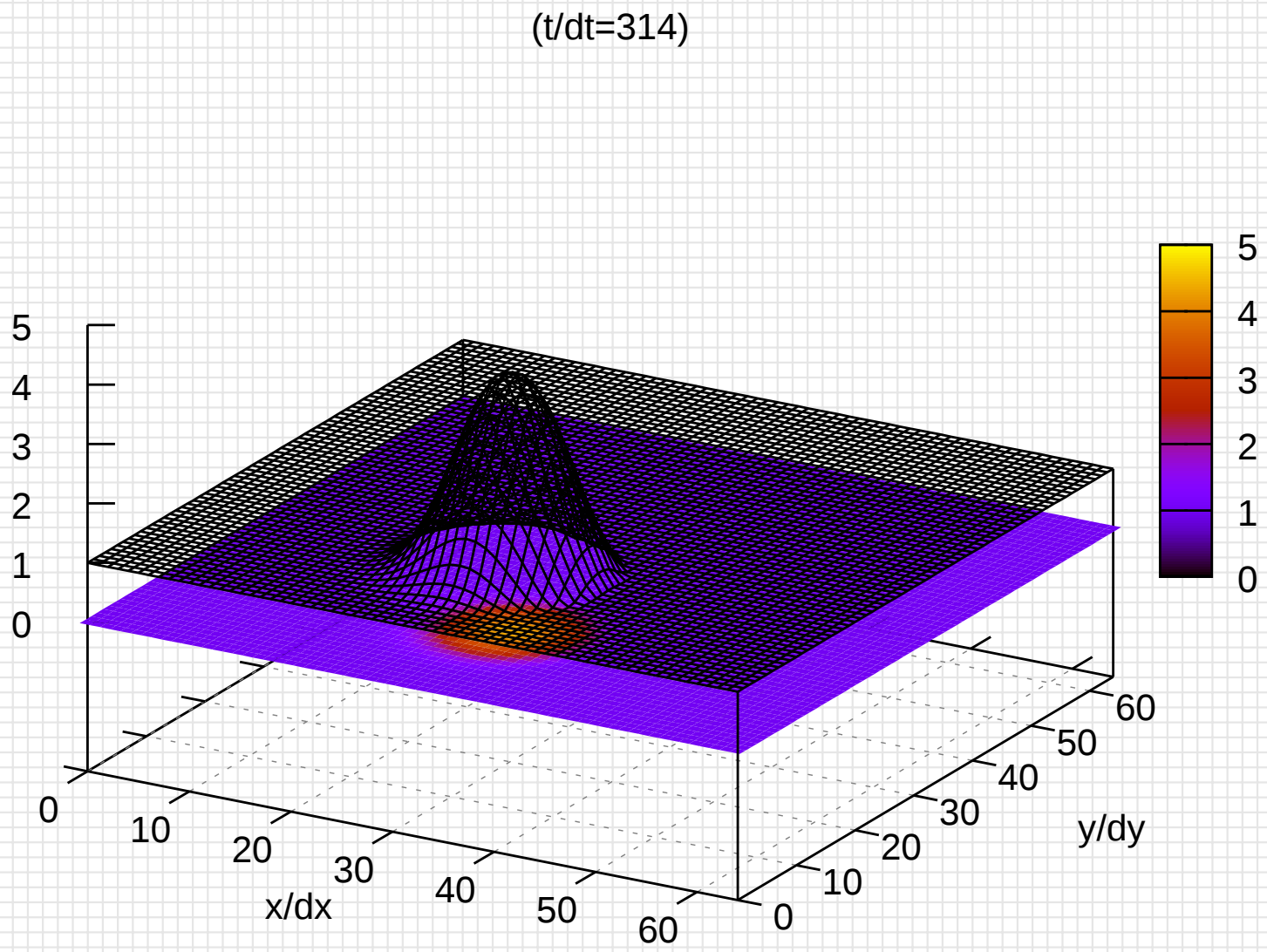
64 LOC using libmpdata++

libmpdata++: rotating cone test



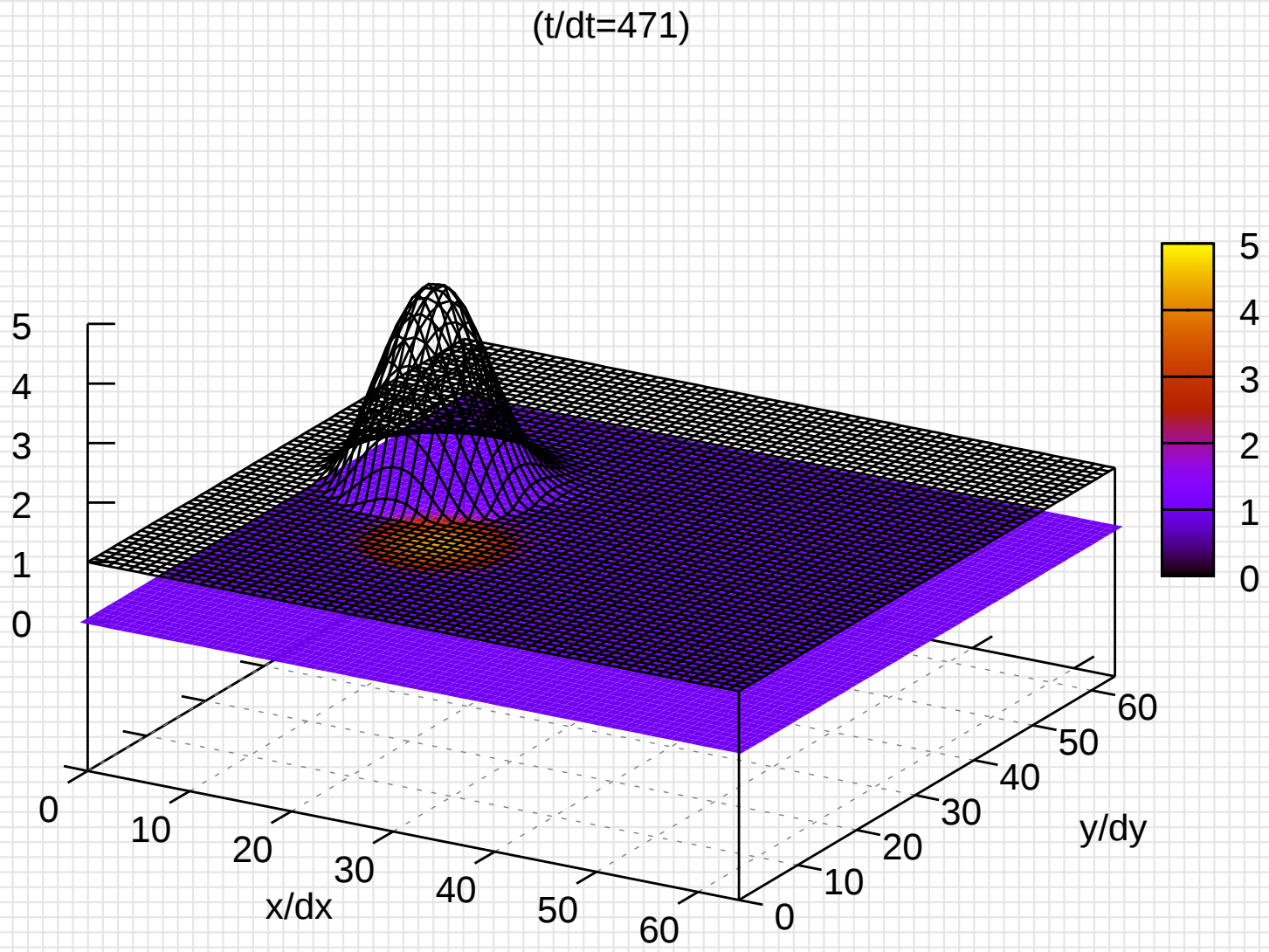
64 LOC using libmpdata++

libmpdata++: rotating cone test



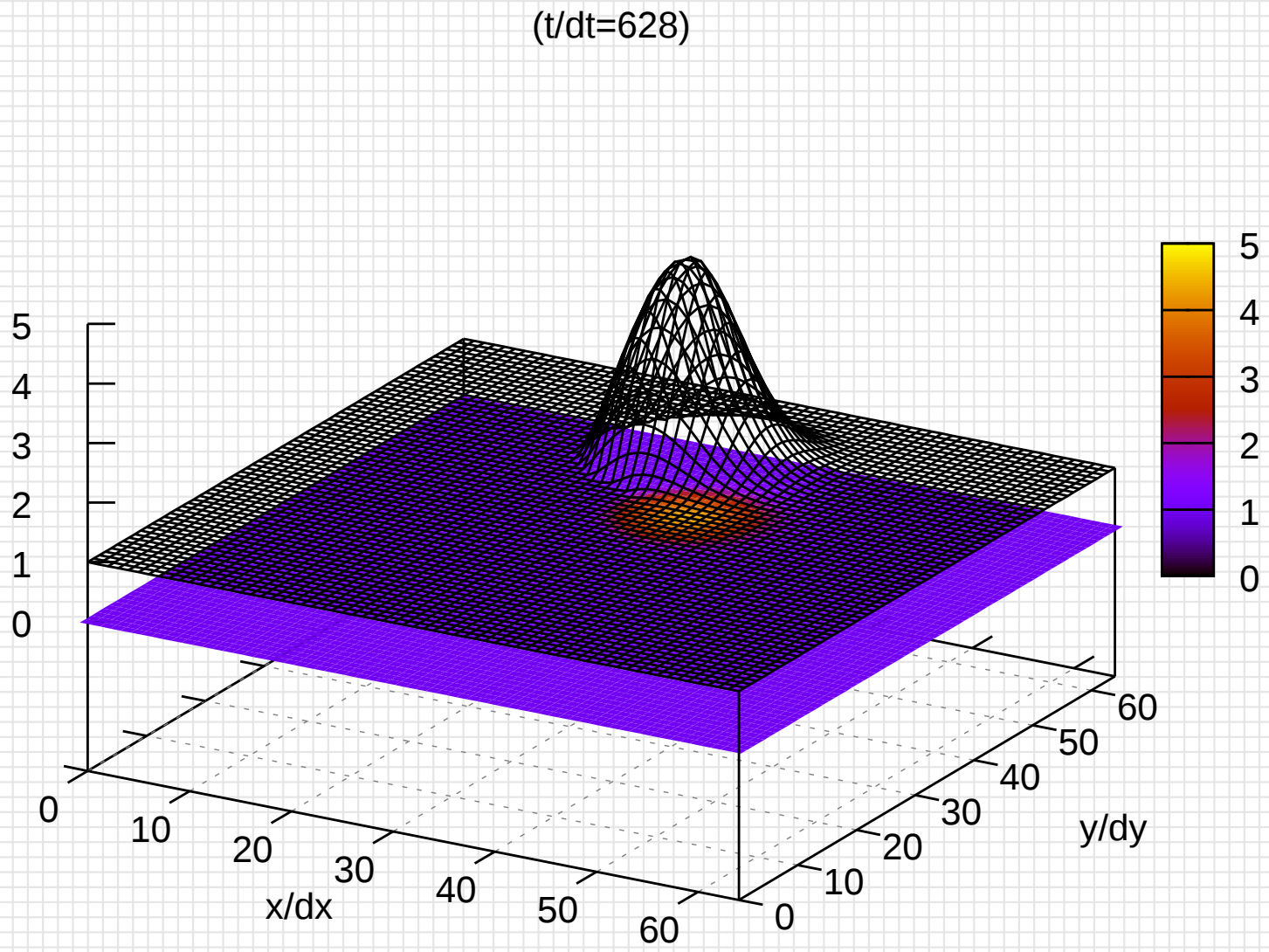
64 LOC using libmpdata++

libmpdata++: rotating cone test



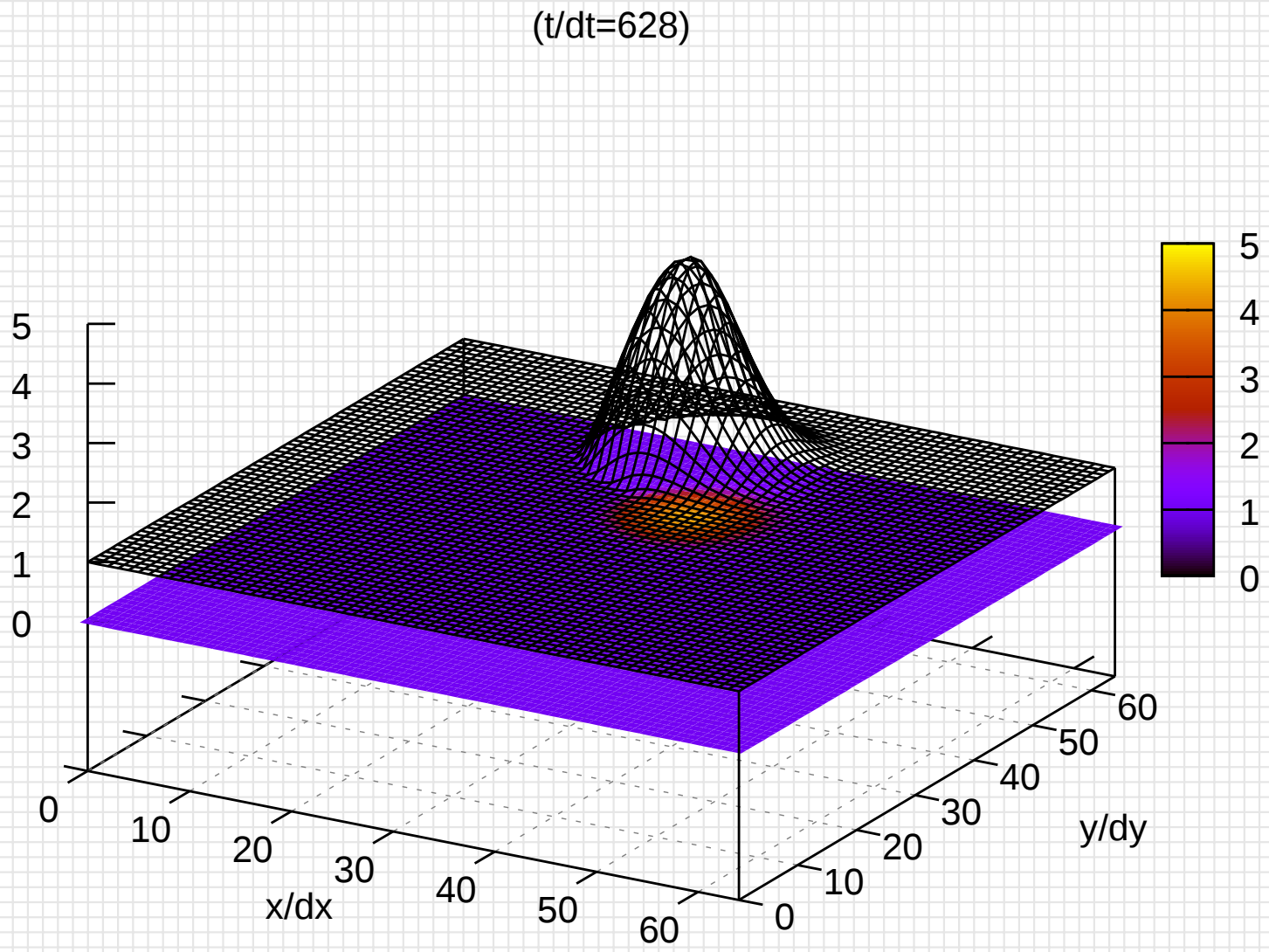
64 LOC using libmpdata++

libmpdata++: rotating cone test



64 LOC using libmpdata++

libmpdata++: rotating cone test



64 LOC using libmpdata++

side note: concurrency nomenclature

problem decomposition method

- domain decomposition (data parallelism)
- ...

process interaction method (hardware related)

- shared-memory (e.g., multi-core CPU)
  $\rightsquigarrow$ synchronisation-based concurrency, e.g., OpenMP
- distributed-memory (e.g., multi-node cluster)
  $\rightsquigarrow$ communication-based concurrency, e.g., MPI
- hybrid, e.g., OpenMP + MPI

side note: concurrency nomenclature

problem decomposition method

- domain decomposition (data parallelism)
- . . .

process interaction method (hardware related)

- shared-memory (e.g., multi-core CPU)
  $\rightsquigarrow$ synchronisation-based concurrency, e.g., OpenMP
- distributed-memory (e.g., multi-node cluster)
  $\rightsquigarrow$ communication-based concurrency, e.g., MPI
- hybrid, e.g., OpenMP + MPI

side note: concurrency nomenclature

problem decomposition method

- domain decomposition (data parallelism)
- ...

process interaction method (hardware related)

- shared-memory (e.g., multi-core CPU)
  $\rightsquigarrow$ synchronisation-based concurrency, e.g., OpenMP
- distributed-memory (e.g., multi-node cluster)
  $\rightsquigarrow$ communication-based concurrency, e.g., MPI
- hybrid, e.g., OpenMP + MPI

side note: concurrency nomenclature

problem decomposition method

- domain decomposition (data parallelism)
- ...

process interaction method (hardware related)

- shared-memory (e.g., multi-core CPU)
  $\rightsquigarrow$ synchronisation-based concurrency, e.g., OpenMP
- distributed-memory (e.g., multi-node cluster)
  $\rightsquigarrow$ communication-based concurrency, e.g., MPI
- hybrid, e.g., OpenMP + MPI

side note: concurrency nomenclature

problem decomposition method

- domain decomposition (data parallelism)
- ...

process interaction method (hardware related)

- shared-memory (e.g., multi-core CPU)
  $\rightsquigarrow$ synchronisation-based concurrency, e.g., OpenMP
- distributed-memory (e.g., multi-node cluster)
  $\rightsquigarrow$ communication-based concurrency, e.g., MPI
- hybrid, e.g., OpenMP + MPI

side note: concurrency nomenclature

problem decomposition method

- domain decomposition (data parallelism)
- ...

process interaction method (hardware related)

- shared-memory (e.g., multi-core CPU)
  $\rightsquigarrow$ synchronisation-based concurrency, e.g., OpenMP
- distributed-memory (e.g., multi-node cluster)
  $\rightsquigarrow$ communication-based concurrency, e.g., MPI
- hybrid, e.g., OpenMP + MPI

side note: concurrency nomenclature

problem decomposition method

- domain decomposition (data parallelism)
- ...

process interaction method (hardware related)

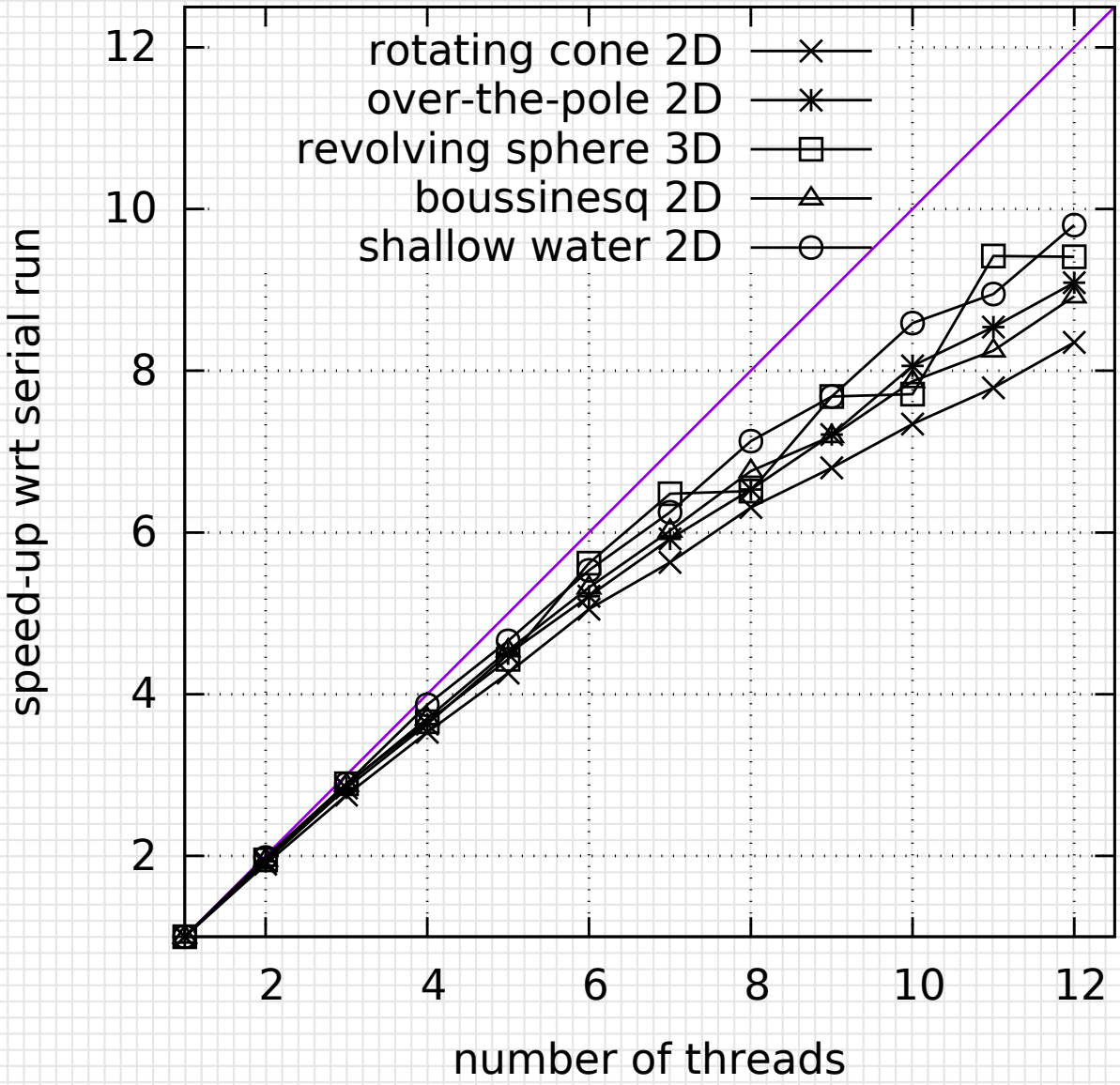
- shared-memory (e.g., multi-core CPU)
  $\rightsquigarrow$ synchronisation-based concurrency, e.g., OpenMP
- distributed-memory (e.g., multi-node cluster)
  $\rightsquigarrow$ communication-based concurrency, e.g., MPI
- hybrid, e.g., OpenMP + MPI

with multi-threading ⇝ also 64 LOC!

```
2c2
< #include <libmpdata++/concurr/serial.hpp>
---
> #include <libmpdata++/concurr/threads.hpp>
30c30
<   lmpdt::concurr::serial<
---
>   lmpdt::concurr::threads<
```

```
$ top
...
  PID USER      PR  NI    S  %CPU %MEM nTH     TIME+ COMMAND
21031 slayoo    20   0    R  73.7  0.1   4   0:01.68 hello_worl  90%
...
```

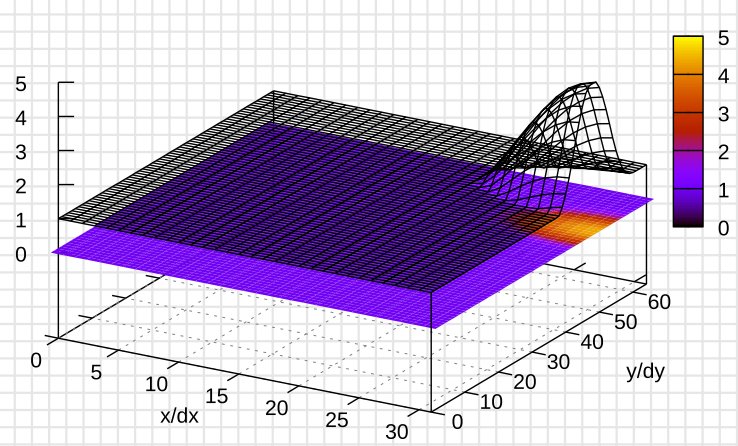
libmpdata++: multi-threading performance scaling

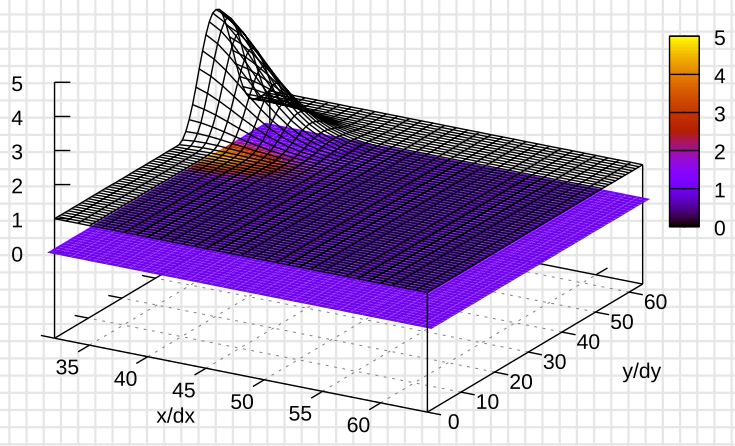


MPI + threads ⇝ also 64 LOC!!! (recompilation only)

```
$ cmake . -DCMAKE_CXX_COMPILER=mpic++
$ make
$ OMP_NUM_THREADS=2 mpirun -np 2 ./hello_world
```

```
$ top
...
  PID USER      PR  NI    S  %CPU %MEM nTH     TIME+ COMMAND
19640 slayoo    20   0    R  65.5  0.3   2   0:00.92 hello_worl  98%
19641 slayoo    20   0    R  64.0  0.3   2   0:00.91 hello_worl  99%
...
```





Plan of the talk

1. what's libmpdata++
2. libmpdata++ 1.0: summary of features
3. libmpdata++ 2.0: new features under development
4. libmpdata++: a hello-world program
5. closing remarks

Plan of the talk

libmpdata++ is free/libre open-source software
distributed under the terms of GNU GPL v3

you are all more than welcome to:
use, study, extend and redistribute the code

libmpdata++ is free/libre open-source software
distributed under the terms of GNU GPL v3

you are all more than welcome to:
use, study, extend and redistribute the code

NATIONAL SCIENCE CENTRE
POLAND

"HARMONIA" international co-operation project

- Title: **Aerosol processing by clouds: development of a multifaceted object-oriented numerical simulation framework**
- Leader: **prof. Hanna Pawłowska**
- Duration: 3 years (2013-2016)
- Key goal: Development of a new free & open-source LES-type tool equipped with Lagrangian aerosol-cloud microphysics
- Partners:

NATIONAL SCIENCE CENTRE
POLAND

"HARMONIA" international co-operation project

- Title: **Aerosol processing by clouds: development of a multifaceted object-oriented numerical simulation framework**
- Leader: **prof. Hanna Pawłowska**
- Duration: 3 years (2013-2016)
- Key goal: Development of a new free & open-source LES-type tool equipped with Lagrangian aerosol-cloud microphysics
- Partners:

NATIONAL SCIENCE CENTRE
POLAND

"HARMONIA" international co-operation project

- Title: **Aerosol processing by clouds: development of a multifaceted object-oriented numerical simulation framework**
- Leader: **prof. Hanna Pawłowska**
- Duration: 3 years (2013-2016)
- Key goal: Development of a new free & open-source LES-type tool equipped with Lagrangian aerosol-cloud microphysics
- Partners:

Piotr Smolarkiewicz

NATIONAL SCIENCE CENTRE
POLAND

"HARMONIA" international co-operation project

- Title: **Aerosol processing by clouds: development of a multifaceted object-oriented numerical simulation framework**
- Leader: **prof. Hanna Pawłowska**
- Duration: 3 years (2013-2016)
- Key goal: Development of a new free & open-source LES-type tool equipped with Lagrangian aerosol-cloud microphysics
- Partners:

Piotr Smolarkiewicz Wojciech Grabowski

NATIONAL SCIENCE CENTRE
POLAND

"HARMONIA" international co-operation project

- Title: **Aerosol processing by clouds: development of a multifaceted object-oriented numerical simulation framework**
- Leader: **prof. Hanna Pawłowska**
- Duration: 3 years (2013-2016)
- Key goal: Development of a new free & open-source LES-type tool equipped with Lagrangian aerosol-cloud microphysics
- Partners:

Piotr Smolarkiewicz Wojciech Grabowski

NATIONAL SCIENCE CENTRE
POLAND

"HARMONIA" international co-operation project

- Title: **Aerosol processing by clouds: development of a multifaceted object-oriented numerical simulation framework**
- Leader: **prof. Hanna Pawłowska**
- Duration: 3 years (2013-2016)
- Key goal: Development of a new free & open-source LES-type tool equipped with Lagrangian aerosol-cloud microphysics
- Partners:

Piotr Smolarkiewicz Wojciech Grabowski

ECMWF

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

libmpdata++

a prospective dynamical core for LES

Thank you for your attention

Thank you for your attention

- libmpdata++ paper: Jaruga et al. 2015, Geosci. Model Dev. doi:10.5194/gmd-8-1005-2015
- spreading drop paper: Jarecka et al. 2015, J. Comp. Phys. doi:10.1016/j.jcp.2015.02.003
- our group's website: http://foss.igf.fuw.edu.pl/

acknowledgements:

- Development of libmpdata++ have been supported by Poland's National Science Centre (decision no. 2012/06/M/ST10/00434)
- and have benefited from computational time granted by the Center for Cooperative Work on Computational Science, Univ. Hyogo, Japan

Thank you for your attention

- libmpdata++ paper: Jaruga et al. 2015, Geosci. Model Dev. doi:10.5194/gmd-8-1005-2015
- spreading drop paper: Jarecka et al. 2015, J. Comp. Phys. doi:10.1016/j.jcp.2015.02.003
- our group's website: `http://foss.igf.fuw.edu.pl/`

acknowledgements:

- Development of libmpdata++ have been supported by Poland's National Science Centre (decision no. 2012/06/M/ST10/00434)
- and have benefited from computational time granted by the Center for Cooperative Work on Computational Science, Univ. Hyogo, Japan

Thank you for your attention

- libmpdata++ paper: Jaruga et al. 2015, Geosci. Model Dev. doi:10.5194/gmd-8-1005-2015
- spreading drop paper: Jarecka et al. 2015, J. Comp. Phys. doi:10.1016/j.jcp.2015.02.003
- our group's website: `http://foss.igf.fuw.edu.pl/`

acknowledgements:

- Development of libmpdata++ have been supported by Poland's National Science Centre (decision no. 2012/06/M/ST10/00434)
- and have benefited from computational time granted by the Center for Cooperative Work on Computational Science, Univ. Hyogo, Japan

Thank you for your attention

- libmpdata++ paper: Jaruga et al. 2015, Geosci. Model Dev. doi:10.5194/gmd-8-1005-2015
- spreading drop paper: Jarecka et al. 2015, J. Comp. Phys. doi:10.1016/j.jcp.2015.02.003
- our group's website: `http://foss.igf.fuw.edu.pl/`

acknowledgements:

- Development of libmpdata++ have been supported by Poland's National Science Centre (decision no. 2012/06/M/ST10/00434)
- and have benefited from computational time granted by the Center for Cooperative Work on Computational Science, Univ. Hyogo, Japan

C++ vs. FORTRAN: language-choice tradeoffs

Formula translation in Blitz++, NumPy and modern Fortran: A case study of the language choice tradeoffs

Sylwester Arabas[1], Dorota Jarecka[1], Anna Jaruga[1], Maciej Fijałkowski[2]

[1]Institute of Geophysics, Faculty of Physics, University of Warsaw
[2]PyPy Team

Journal	Scientific Programming
DOI	10.3233/SPR-140379
Online Date	Monday, March 24, 2014

libmpdata++ vs. F77-MPDATA: CPU-time ratios

(3D, homogeneous advection, serial)

grid	59^3	$(2\times59)^3$	$(4\times59)^3$	$(6\times59)^3$
ratio	4.8	2.0	1.4	0.9

C++ vs. FORTRAN: language-choice tradeoffs

Formula translation in Blitz++, NumPy and modern Fortran: A case study of the language choice tradeoffs

Sylwester Arabas[1], Dorota Jarecka[1], Anna Jaruga[1], Maciej Fijałkowski[2]

[1]Institute of Geophysics, Faculty of Physics, University of Warsaw
[2]PyPy Team

Journal	Scientific Programming
DOI	10.3233/SPR-140379
Online Date	Monday, March 24, 2014

libmpdata++ vs. F77-MPDATA: CPU-time ratios

(3D, homogeneous advection, serial)

grid	59^3	$(2\times59)^3$	$(4\times59)^3$	$(6\times59)^3$
ratio	4.8	2.0	1.4	0.9

libmpdata++: some design choices

legal

- license: GPL
- repo: github.com/igfuw/

library components

- solvers/algorithms:
  - ...
- boundary conditions:
  - ...
- output handlers:
  - HDF5/XDMF (MPI-IO)
  - gnuplot
- shared-mem concurrency:
  - OpenMP
  - Boost.Thread
  - C++11 threads
- distributed-mem concurr.:
  - MPI

dependencies

- C++ compiler (C++11 & OpenMP)
- Blitz++
- Boost (ptr_container, timer, thread, preprocessor, filesystem, format, property_tree, MPI)
- CMake, CTest
- MPI
- HDF5
- gnuplot-iostream

API

- header-only library
- template-based component selection
- inheritance-based component extensions
- user exposed to Blitz++ API