

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

Sylwester Arabas  
Faculty of Physics, University of Warsaw, Poland

seminar presented at the  
Graduate School for Computational Studies, Hyogo University

Kōbe, Japan, September 7, 2015

let me introduce myself

## alma mater (MSc/2008, PhD/2013, postdoc)

	est.	staff	students	
			BSc/MSc	PhD
University of Warsaw	1816	6000	55000	3000
Faculty of Physics	1816	300	1000	150
Institute of Geophysics	1948	30	20	20
Atmospheric Physics Division	1949	10	10	10

## collaboration with Shima-san

- 2010: first contact thanks to Enomoto-san
- 2010: 4 weeks at JAMSTEC/Yokohama: RICO-5DM project
- 2013: Arabas & Shima paper in J. Atmos. Sci.
- 2014: 1 week at Hyogo Univ., our group starts using rokko comp.
- 2011-2015: >10 related posters & talks → super-droplet advocate

## alma mater (MSc/2008, PhD/2013, postdoc)

	est.	staff	students	
			BSc/MSc	PhD
University of Warsaw	1816	6000	55000	3000
Faculty of Physics	1816	300	1000	150
Institute of Geophysics	1948	30	20	20
Atmospheric Physics Division	1949	10	10	10

## collaboration with Shima-san

- 2010: first contact thanks to Enomoto-san
- 2010: 4 weeks at JAMSTEC/Yokohama: RICO-SDM project
- 2013: Arabas & Shima paper in J. Atmos. Sci.
- 2014: 1 week at Hyogo Univ., our group starts using rokko comp.
- 2011-2015: >10 related posters & talks  $\rightsquigarrow$  super-droplet advocate

## alma mater (MSc/2008, PhD/2013, postdoc)

	est.	staff	students	
			BSc/MSc	PhD
University of Warsaw	1816	6000	55000	3000
Faculty of Physics	1816	300	1000	150
Institute of Geophysics	1948	30	20	20
Atmospheric Physics Division	1949	10	10	10

## collaboration with Shima-san

2010: first contact thanks to Enomoto-san

2010: 4 weeks at JAMSTEC/Yokohama: RICO-SDM project

2013: Arabas & Shima paper in J. Atmos. Sci.

2014: 1 week at Hyogo Univ., our group starts using rokko comp.

2011-2015: >10 related posters & talks  $\rightsquigarrow$  super-droplet advocate

## alma mater (MSc/2008, PhD/2013, postdoc)

	est.	staff	students	
			BSc/MSc	PhD
University of Warsaw	1816	6000	55000	3000
Faculty of Physics	1816	300	1000	150
Institute of Geophysics	1948	30	20	20
Atmospheric Physics Division	1949	10	10	10

## collaboration with Shima-san

2010: first contact thanks to Enomoto-san

2010: 4 weeks at JAMSTEC/Yokohama: RICO-SDM project

2013: Arabas & Shima paper in J. Atmos. Sci.

2014: 1 week at Hyogo Univ., our group starts using rokko comp.

2011-2015: >10 related posters & talks  $\rightsquigarrow$  super-droplet advocate

## alma mater (MSc/2008, PhD/2013, postdoc)

	est.	staff	students	
			BSc/MSc	PhD
University of Warsaw	1816	6000	55000	3000
Faculty of Physics	1816	300	1000	150
Institute of Geophysics	1948	30	20	20
Atmospheric Physics Division	1949	10	10	10

## collaboration with Shima-san

2010: first contact thanks to Enomoto-san

2010: 4 weeks at JAMSTEC/Yokohama: RICO-SDM project

2013: Arabas & Shima paper in J. Atmos. Sci.

2014: 1 week at Hyogo Univ., our group starts using rokko comp.

2011-2015: >10 related posters & talks  $\rightsquigarrow$  super-droplet advocate

## alma mater (MSc/2008, PhD/2013, postdoc)

	est.	staff	students	
			BSc/MSc	PhD
University of Warsaw	1816	6000	55000	3000
Faculty of Physics	1816	300	1000	150
Institute of Geophysics	1948	30	20	20
Atmospheric Physics Division	1949	10	10	10

## collaboration with Shima-san

2010: first contact thanks to Enomoto-san

2010: 4 weeks at JAMSTEC/Yokohama: RICO-SDM project

2013: Arabas & Shima paper in J. Atmos. Sci.

2014: 1 week at Hyogo Univ., our group starts using rokko comp.

2011-2015: >10 related posters & talks  $\rightsquigarrow$  super-droplet advocate

## alma mater (MSc/2008, PhD/2013, postdoc)

	est.	staff	students	
			BSc/MSc	PhD
University of Warsaw	1816	6000	55000	3000
Faculty of Physics	1816	300	1000	150
Institute of Geophysics	1948	30	20	20
Atmospheric Physics Division	1949	10	10	10

## collaboration with Shima-san

2010: first contact thanks to Enomoto-san

2010: 4 weeks at JAMSTEC/Yokohama: RICO-SDM project

2013: Arabas & Shima paper in J. Atmos. Sci.

2014: 1 week at Hyogo Univ., our group starts using rokko comp.

2011-2015: >10 related posters & talks  $\rightsquigarrow$  super-droplet advocate

## 2015 super-droplet workshop at the University of Warsaw



let me introduce our team

The team of the University of Warsaw, Poland



Anna  
Zimniak

prof. Hanna  
Pawowska

Anna  
Jaruga

Piotr  
Dzielan

Sylwester  
Arabas

Maciek  
Waruszcwski



the team @ the University of Warsaw, Poland



Anna  
Zimniak

prof. Hanna  
Pawowska

Anna  
Jaruga

Piotr  
Dziekan

Sylwester  
Arabas

Maciek  
Waruszewski





Anna  
Zimniak

prof. Hanna  
Pawowska

Anna  
Jaruga

Piotr  
Dziekan

Sylwester  
Arabas

Maciek  
Waruszewski





Anna  
Zimniak

prof. Hanna  
Pawowska

Anna  
Jaruga

Piotr  
Dziekan

Sylwester  
Arabas

Maciek  
Waruszewski



prof. Wojciech Grabowski



Dorota Jarecka



prof. Piotr Smolarkiewicz

the team @ the University of Warsaw, Poland



Anna  
Zimniak

prof. Hanna  
Pawowska

Anna  
Jaruga

Piotr  
Dziekan

Sylwester  
Arabas

Maciek  
Waruszewski

@ NCAR, Boulder, Colorado, USA



prof. Wojciech Grabowski



Dorota Jarecka

@ ECMWF, Reading, UK



prof. Piotr Smolarkiewicz

the team @ the University of Warsaw, Poland



Anna  
Zimniak

prof. Hanna  
Pawowska

Anna  
Jaruga

Piotr  
Dziekan

Sylwester  
Arabas

Maciek  
Waruszewski

@ NCAR, Boulder, Colorado, USA



prof. Wojciech Grabowski



Dorota Jarecka



prof. Piotr Smolarkiewicz

the team @ the University of Warsaw, Poland



Anna  
Zimniak

prof. Hanna  
Pawowska

Anna  
Jaruga

Piotr  
Dziekan

Sylwester  
Arabas

Maciek  
Waruszewski

@ NCAR, Boulder, Colorado, USA



prof. Wojciech Grabowski



Dorota Jarecka

@ ECMWF, Reading, UK



prof. Piotr Smolarkiewicz

the team @ the University of Warsaw, Poland



Anna  
Zimniak

prof. Hanna  
Pawowska

Anna  
Jaruga

Piotr  
Dziekan

Sylwester  
Arabas

Maciek  
Waruszewski

@ NCAR, Boulder, Colorado, USA



prof. Wojciech Grabowski



Dorota Jarecka

@ ECMWF, Reading, UK



prof. Piotr Smolarkiewicz

# Plan of the talk

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

# Plan of the talk

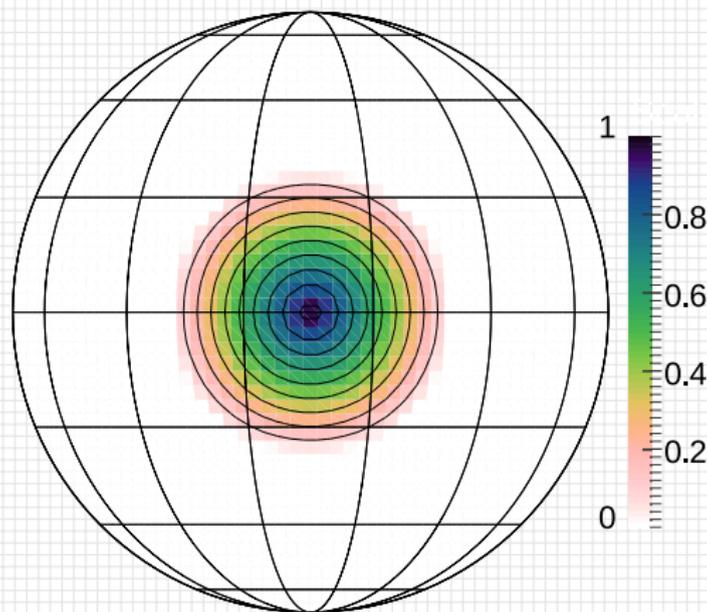
- 1 what's libmpdata++
- 2 libmpdata++: a hello-world program
- 3 libmpdata++ 1.0: summary of features
- 4 libmpdata++ 2.0: new features under development
- 5 closing remarks

the generalised transport equation

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

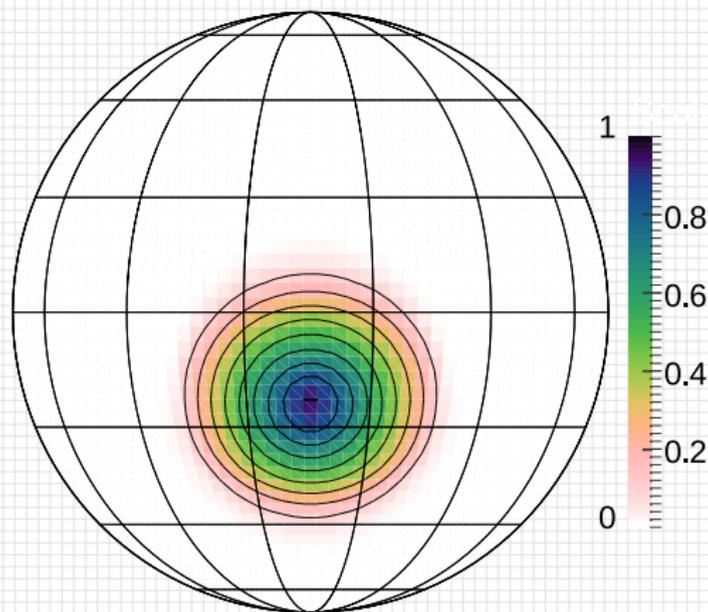
# the generalised transport equation

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



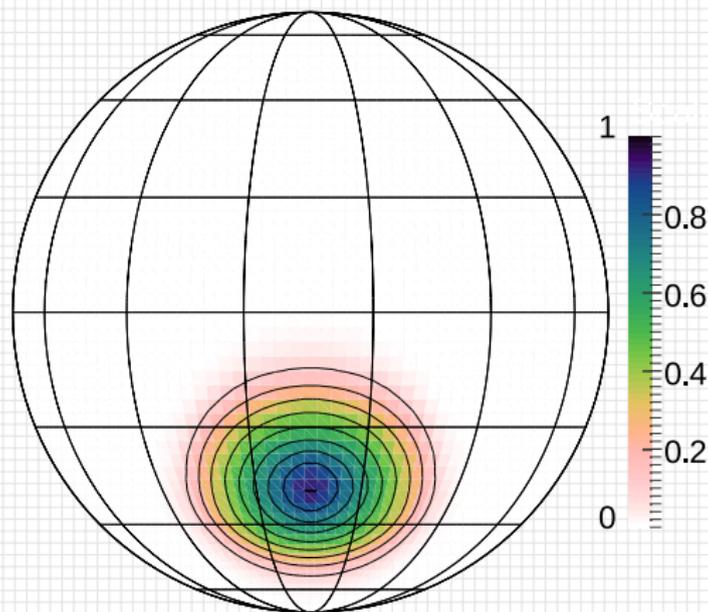
# the generalised transport equation

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



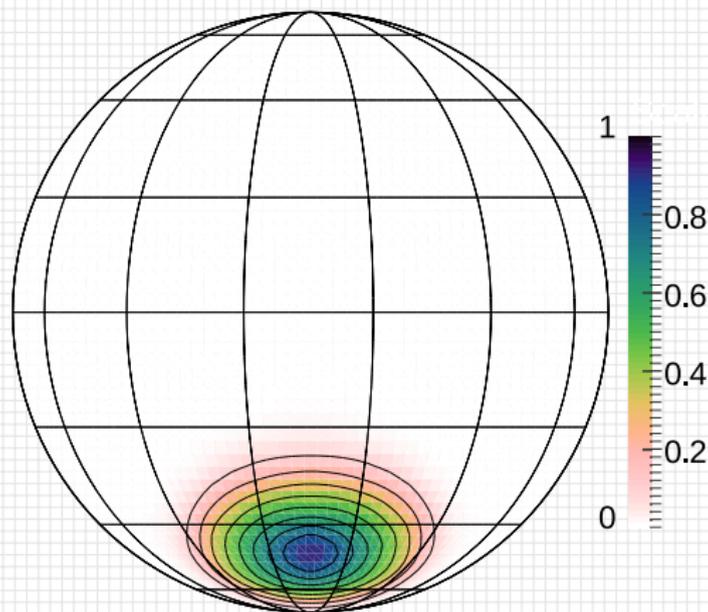
# the generalised transport equation

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



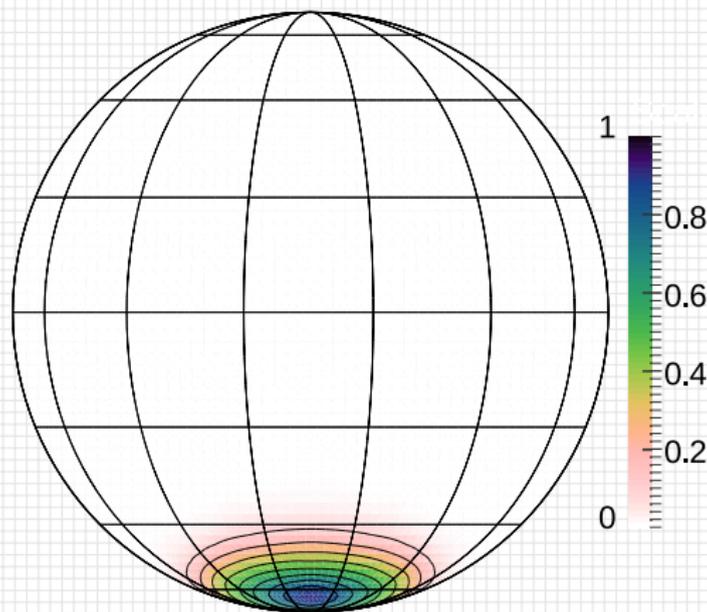
# the generalised transport equation

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



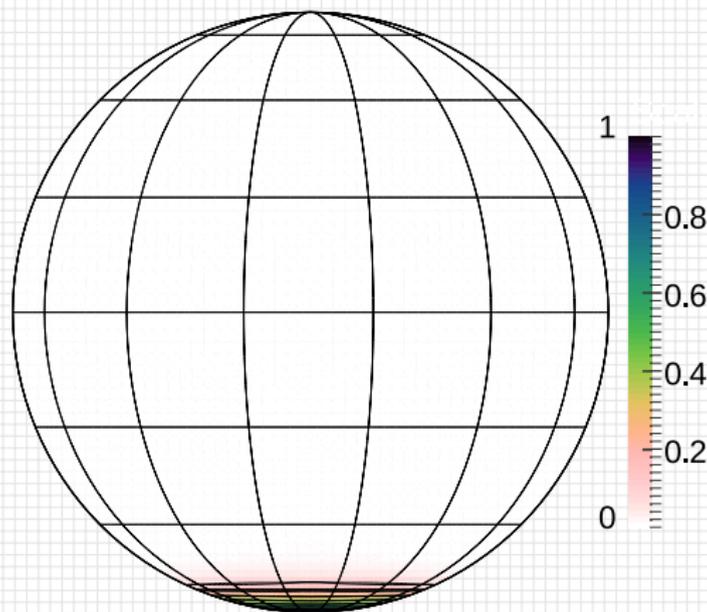
# the generalised transport equation

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



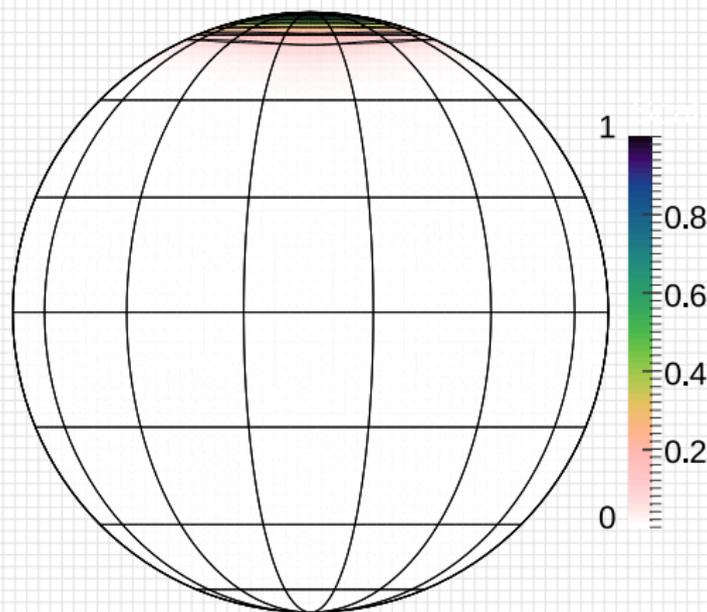
# the generalised transport equation

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



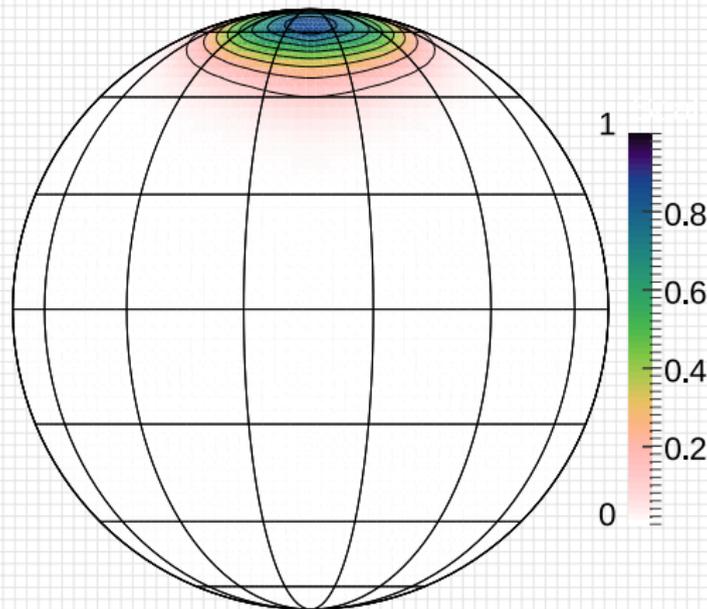
# the generalised transport equation

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



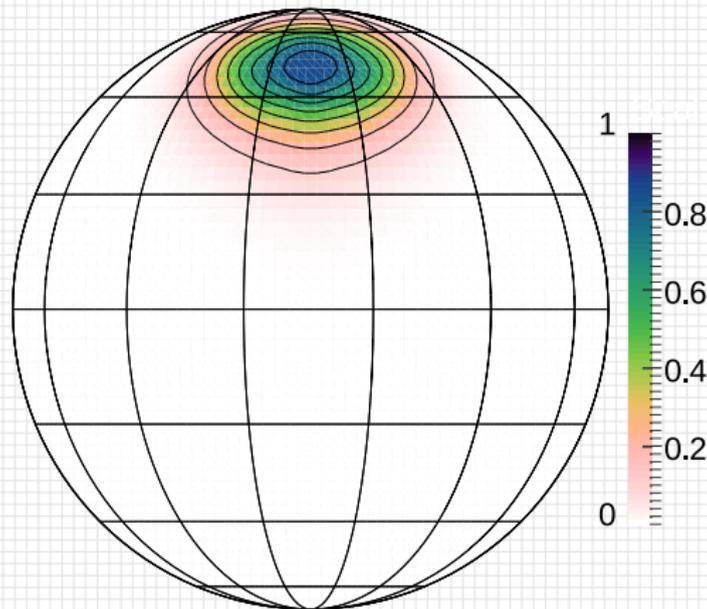
# the generalised transport equation

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



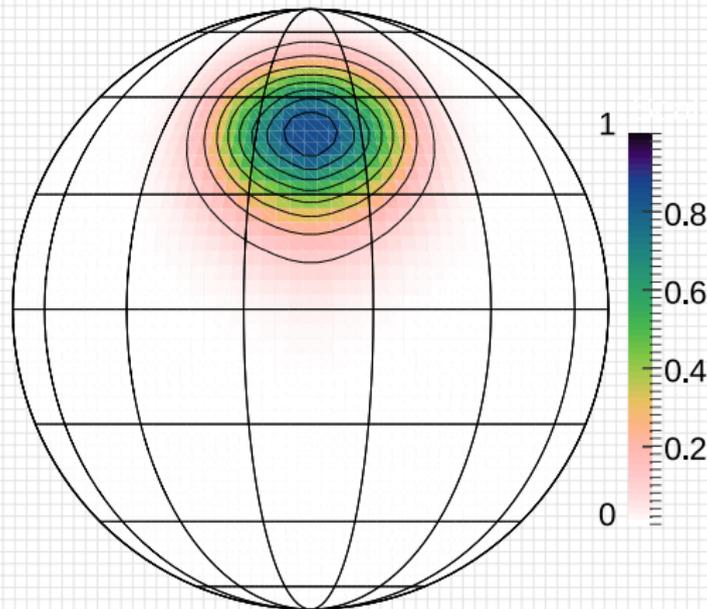
# the generalised transport equation

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



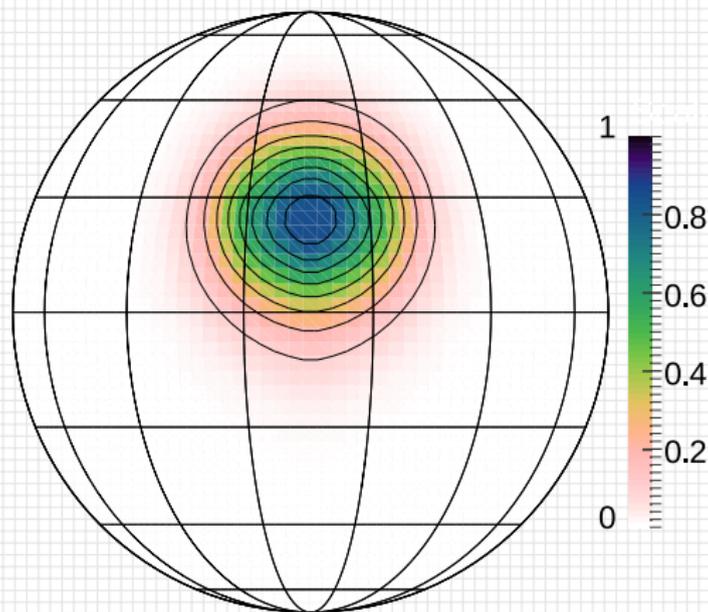
# the generalised transport equation

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



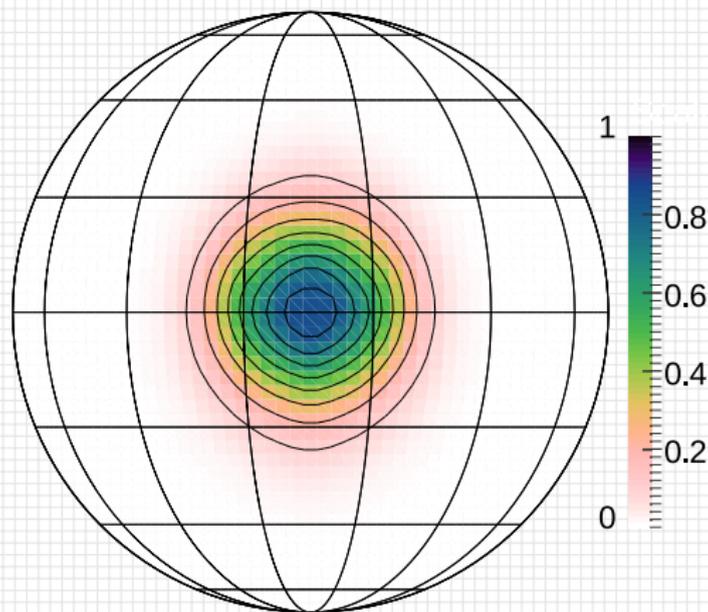
# the generalised transport equation

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



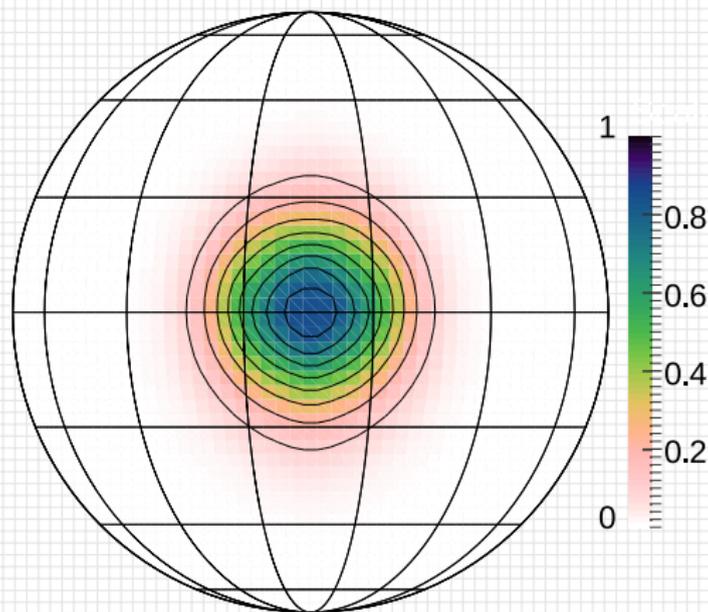
# the generalised transport equation

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



# 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

**libmpdata++**: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

**libmpdata++**: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

**libmpdata++**: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

libmpdata++: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

libmpdata++: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

libmpdata++: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

libmpdata++: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

libmpdata++: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

**libmpdata++**: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

**libmpdata++**: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

**libmpdata++**: a new C++11 / Blitz++ based 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: ~ 700 research papers
- Google Books: ~ 200 mentions in books

original single-file Fortran 77 implementation used till today

- unspecified license, no versioning
- e-mail distribution, copy-paste-modify reuse
- no unit/regression tests

**libmpdata++**: a new C++11 / Blitz++ based 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, free/libre/open code
- **result correctness**  
~> multifaceted peer-reviewed automated tests, free/libre/open code
- **result reproducibility**  
~> atomic versions, no legal nor tech. obstacles, free/libre/open code

researcher = developer

- **ease of extending**  
~> concise OOP syntax, separation of concerns, free/libre/open code  
automated tests, continuous integration

## priority: researchers' productivity

researcher = user

- **ease of obtaining and using**

↪ public repository, documentation, examples, free/libre/open code

- **result correctness**

↪ multifaceted peer-reviewed automated tests, free/libre/open code

- **result reproducibility**

↪ atomic versions, no legal nor tech. obstacles, free/libre/open code

researcher = developer

- **ease of extending**

↪ concise OOP syntax, separation of concerns, free/libre/open code  
automated tests, continuous integration

## priority: researchers' productivity

researcher = user

- **ease of obtaining and using**

↪ public repository, documentation, examples, free/libre/open code

- **result correctness**

↪ multifaceted peer-reviewed automated tests, free/libre/open code

- **result reproducibility**

↪ atomic versions, no legal nor tech. obstacles, free/libre/open code

researcher = developer

- **ease of extending**

↪ concise OOP syntax, separation of concerns, free/libre/open code  
automated tests, continuous integration

## priority: researchers' productivity

researcher = user

- **ease of obtaining and using**

↪ public repository, documentation, examples, free/libre/open code

- **result correctness**

↪ multifaceted peer-reviewed automated tests, free/libre/open code

- **result reproducibility**

↪ atomic versions, no legal nor tech. obstacles, free/libre/open code

researcher = developer

- **ease of extending**

↪ concise OOP syntax, separation of concerns, free/libre/open code  
automated tests, continuous integration

## priority: researchers' productivity

researcher = user

- **ease of obtaining and using**

↪ public repository, documentation, examples, free/libre/open code

- **result correctness**

↪ multifaceted peer-reviewed automated tests, free/libre/open code

- **result reproducibility**

↪ atomic versions, no legal nor tech. obstacles, free/libre/open code

researcher = developer

- **ease of extending**

↪ concise OOP syntax, separation of concerns, free/libre/open code  
automated tests, continuous integration

## priority: researchers' productivity

researcher = user

- **ease of obtaining and using**

↪ public repository, documentation, examples, free/libre/open code

- **result correctness**

↪ multifaceted peer-reviewed automated tests, free/libre/open code

- **result reproducibility**

↪ atomic versions, no legal nor tech. obstacles, free/libre/open code

researcher = developer

- **ease of extending**

↪ concise OOP syntax, separation of concerns, free/libre/open code  
automated tests, continuous integration

# Plan of the talk

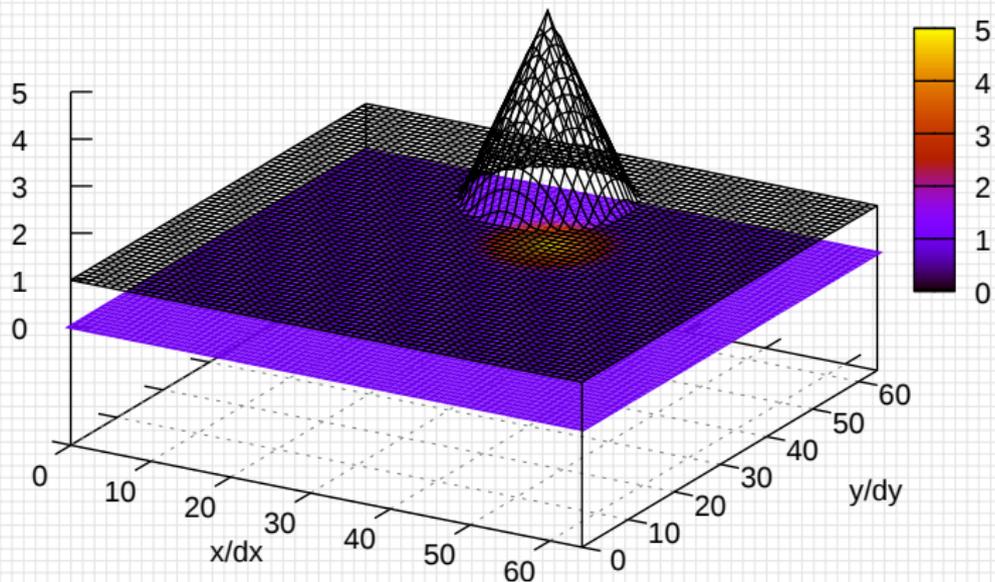
- 1 what's libmpdata++
- 2 libmpdata++: a hello-world program
- 3 libmpdata++ 1.0: summary of features
- 4 libmpdata++ 2.0: new features under development
- 5 closing remarks

# Plan of the talk

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

# libmpdata++: rotating cone test

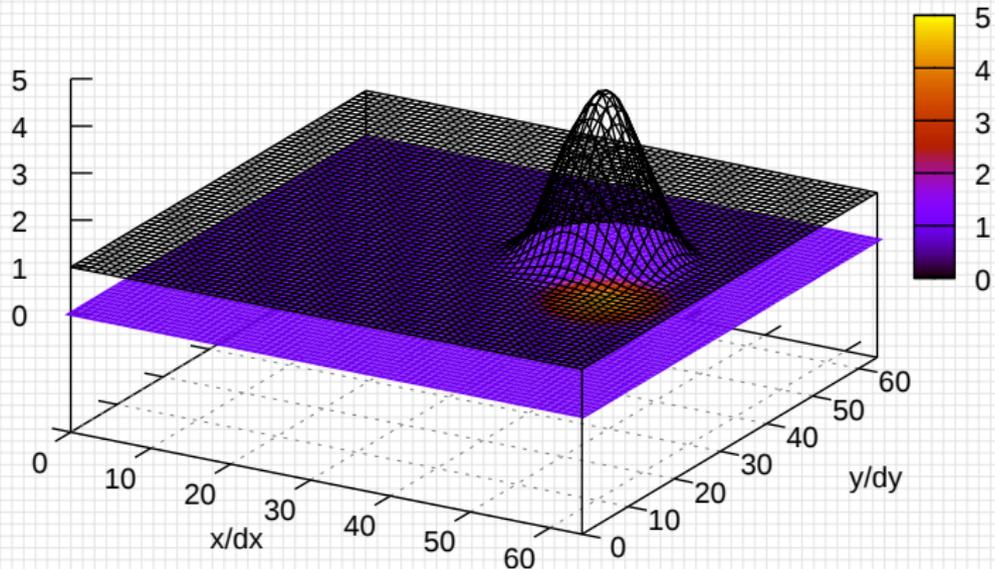
(t/dt=0)



64 LOC using libmpdata++

# libmpdata++: rotating cone test

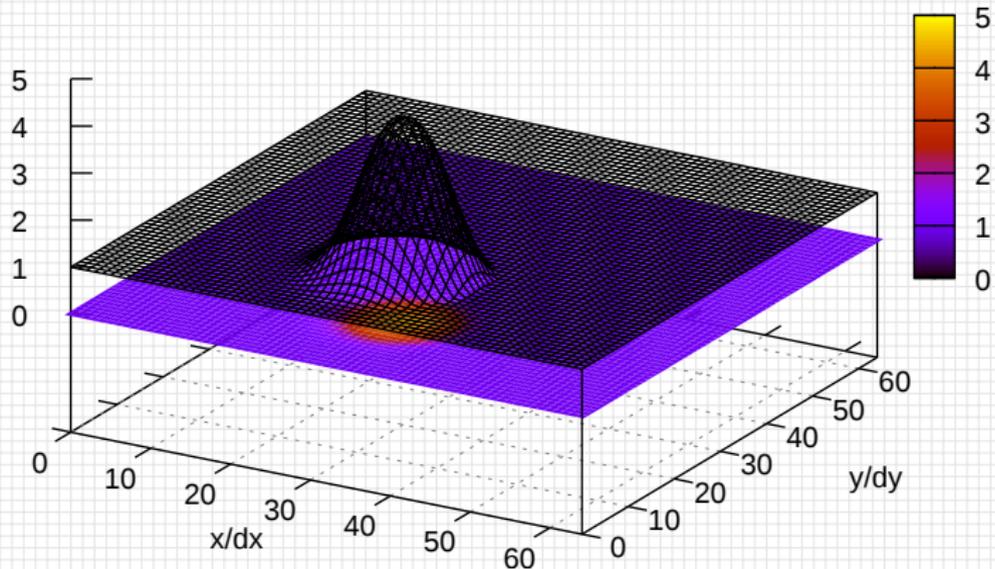
( $t/dt=157$ )



64 LOC using libmpdata++

# libmpdata++: rotating cone test

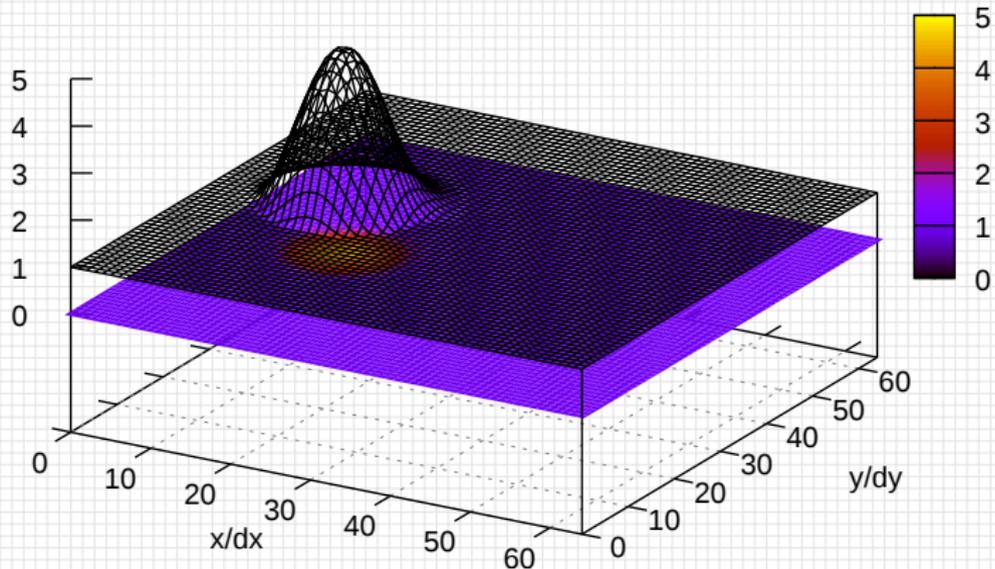
(t/dt=314)



64 LOC using libmpdata++

# libmpdata++: rotating cone test

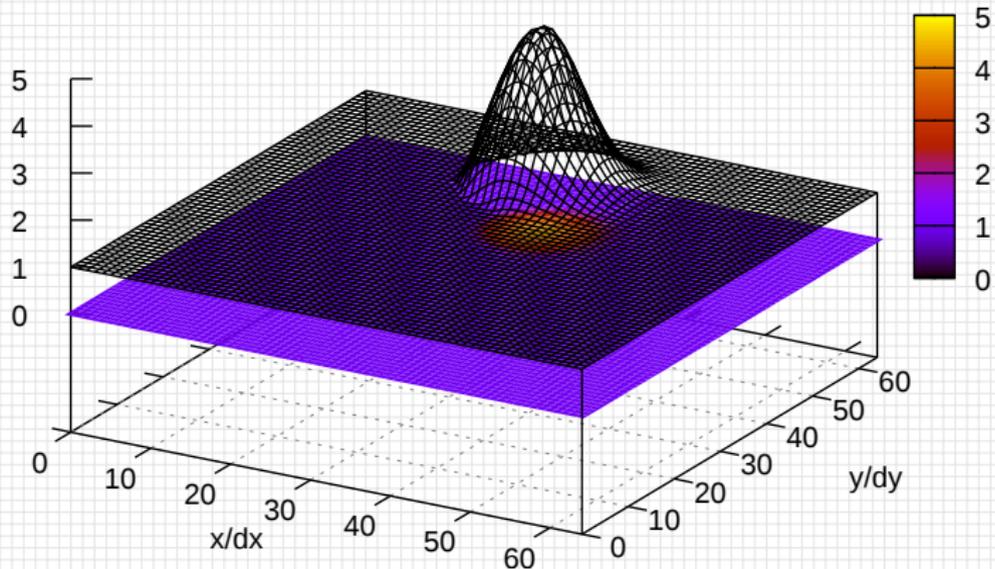
(t/dt=471)



64 LOC using libmpdata++

# libmpdata++: rotating cone test

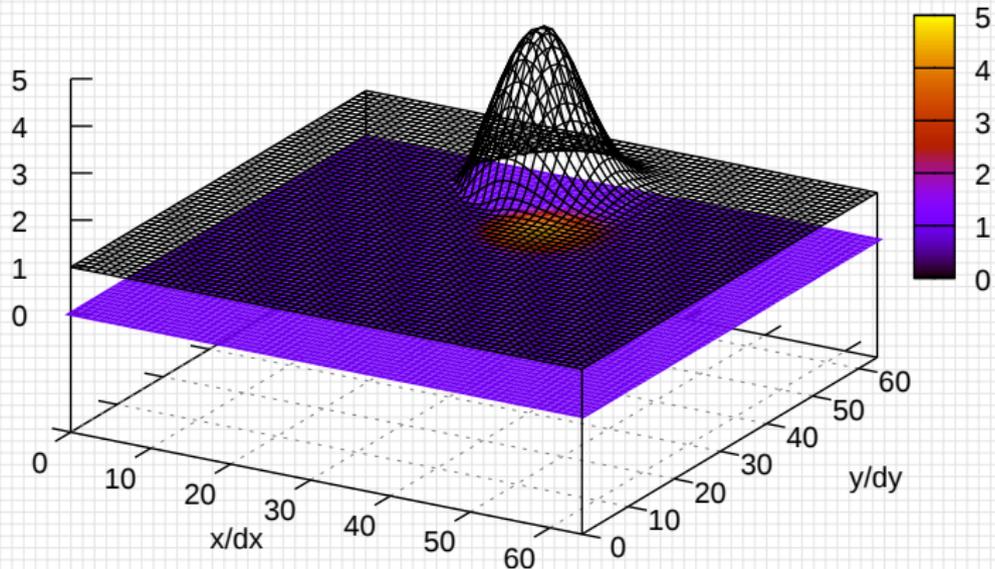
(t/dt=628)



64 LOC using libmpdata++

# libmpdata++: rotating cone test

(t/dt=628)



**64 LOC using libmpdata++**

```

1 #include <libmpdata++/solvers/mpdata.hpp>
2 #include <libmpdata++/concurr/serial.hpp>
3 #include <libmpdata++/output/gnuplot.hpp>
4
5 int main()
6 {
7     namespace lmpdt = libmpdataxx;
8     const int nx=64, ny=64, nt = 628;
9
10    // compile-time parameters
11    struct ct_params_t : lmpdt::ct_params_default_t
12    {
13        using real_t = double;
14        enum { n_dims = 2 };
15        enum { n_eqns = 1 };
16    };
17
18    // solver choice
19    using run_t = lmpdt::output::gnuplot< lmpdt::solvers::mpdata< ct_params_t >>;
20
21    // runtime parameters
22    typename run_t::rt_params_t p;
23    p.grid_size = {nx+1, ny+1};
24    p.outfreq = nt/4;
25    p.gnuplot_output = "out_%s_%d.svg";
26    p.gnuplot_with = "lines";
27    p.gnuplot_cbrange = p.gnuplot_zrange = "[0:5]";
28
29    // sharedmem concurency and boundary condition choice
30    lmpdt::concurr::serial<
31        run_t,
32        lmpdt::bcond::open, lmpdt::bcond::open, // x-left, x-right
33        lmpdt::bcond::open, lmpdt::bcond::open // y-left, y-right
34    > 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

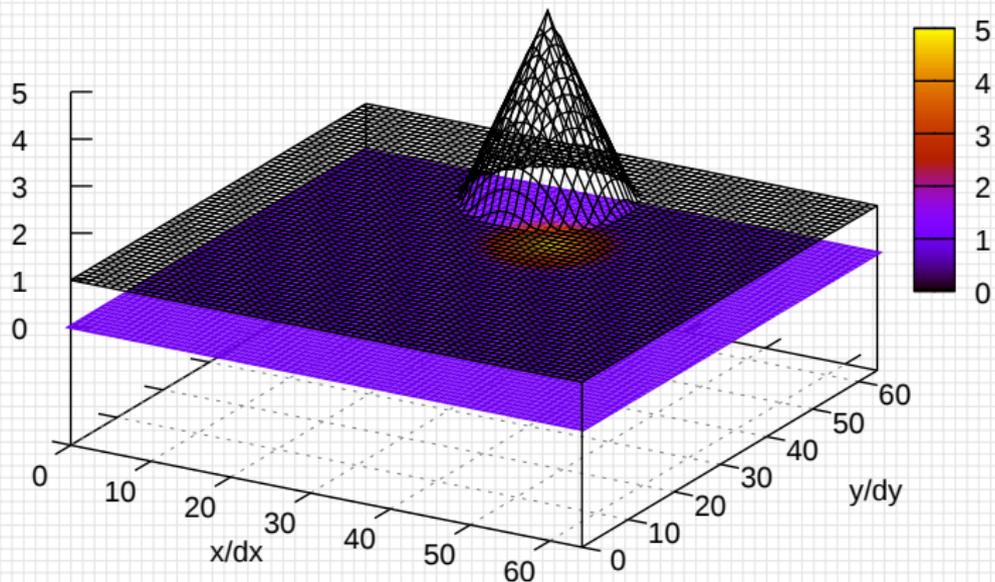
```

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

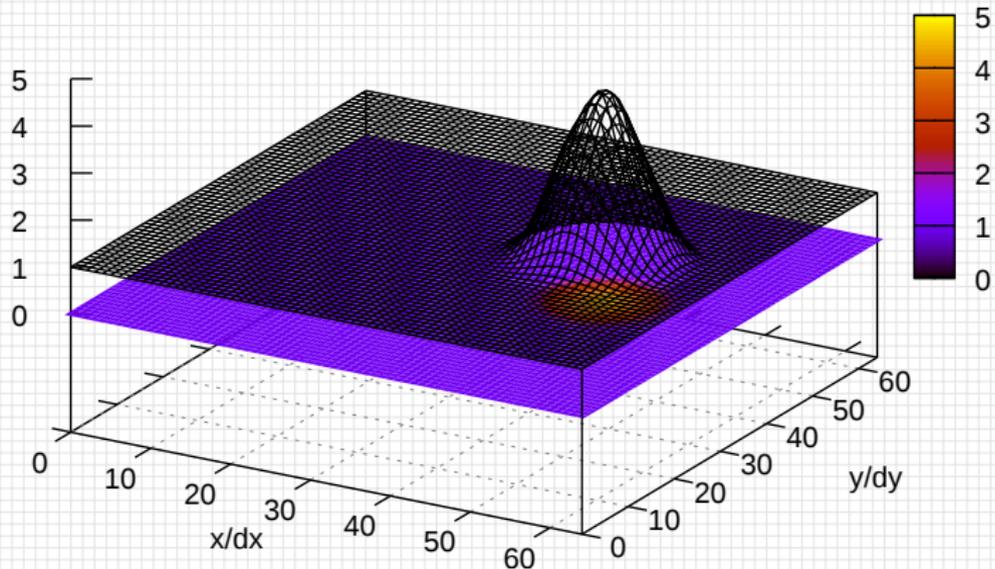
(t/dt=0)



64 LOC using libmpdata++

# libmpdata++: rotating cone test

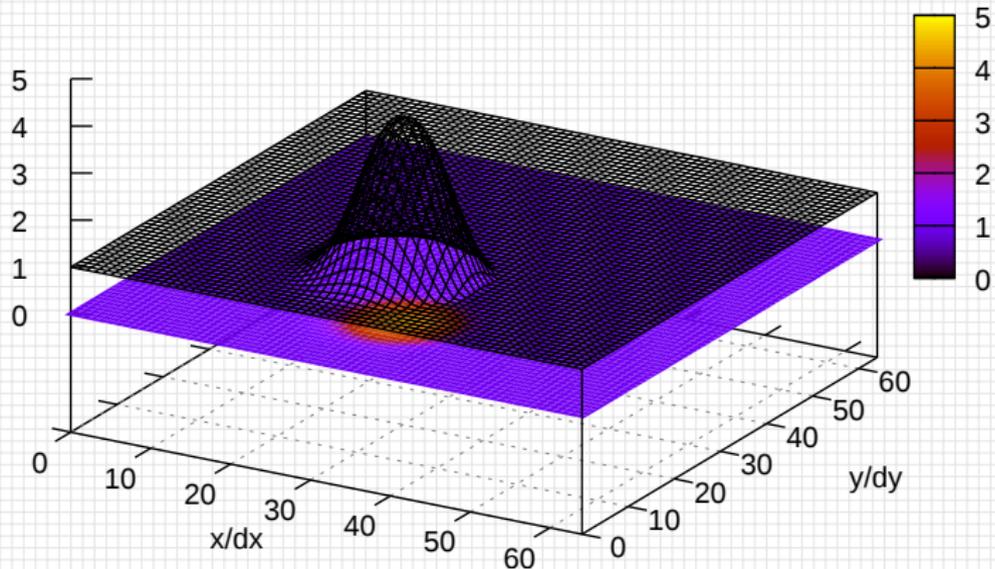
( $t/dt=157$ )



64 LOC using libmpdata++

# libmpdata++: rotating cone test

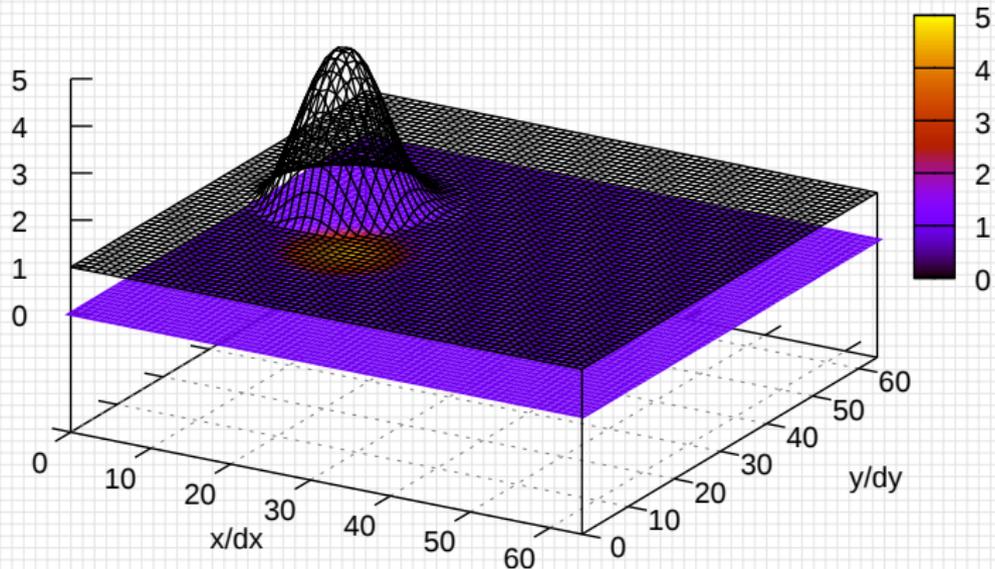
(t/dt=314)



64 LOC using libmpdata++

# libmpdata++: rotating cone test

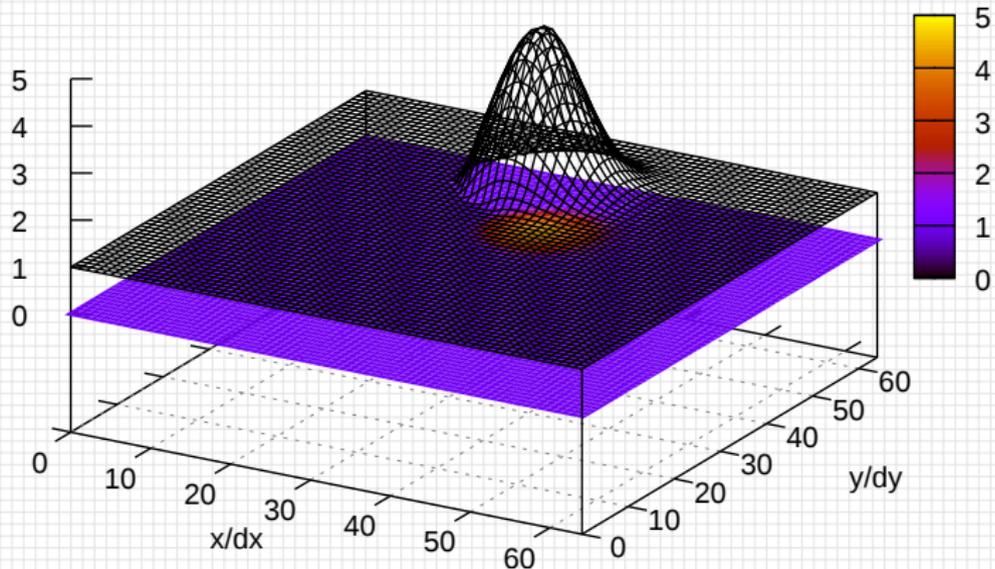
(t/dt=471)



64 LOC using libmpdata++

# libmpdata++: rotating cone test

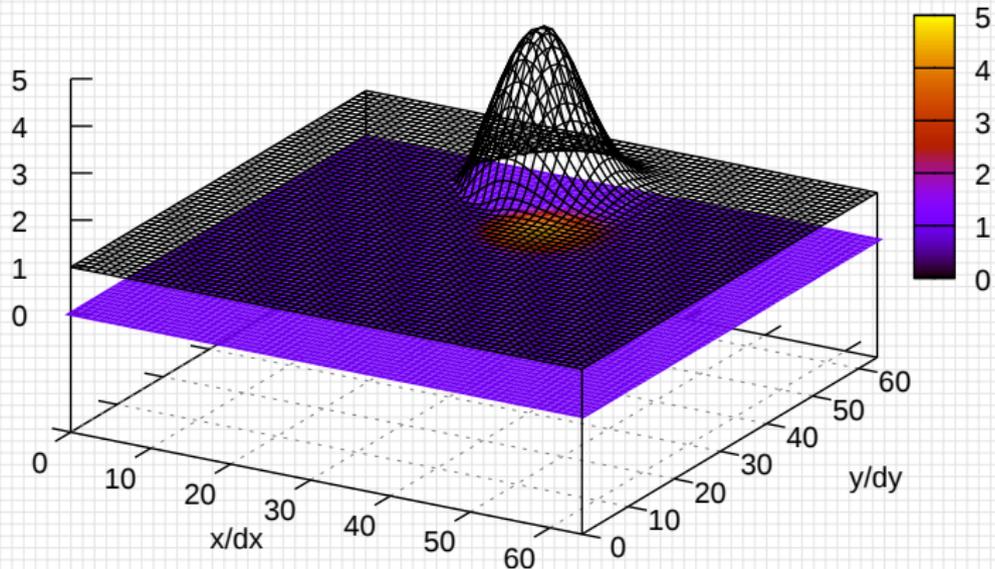
(t/dt=628)



64 LOC using libmpdata++

# libmpdata++: rotating cone test

(t/dt=628)



**64 LOC using libmpdata++**

with multi-threading  $\rightsquigarrow$  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  90%
21031 slayoo    20   0  R  73.7  0.1   4    0:01.68 hello_worl
```

# MPI + threads $\rightsquigarrow$ also 64 LOC!!! (recompilation only)

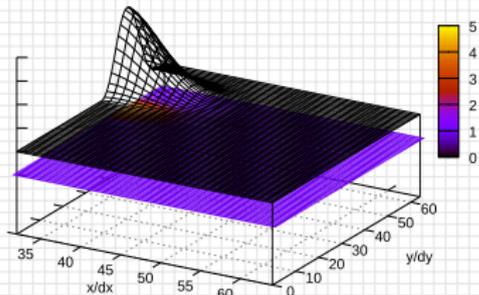
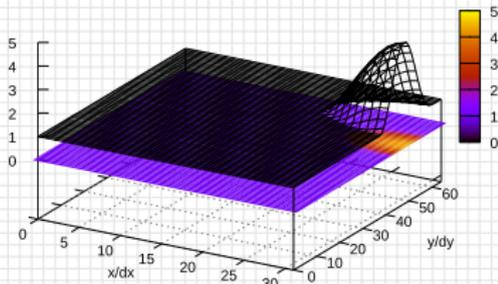
```
$ cmake . -DCMAKE_CXX_COMPILER=mpic++  
$ make  
$ OMP_NUM_THREADS=2 mpirun -np 2 ./hello_world
```

```
$ 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++: a hello-world program
- 3 libmpdata++ 1.0: summary of features
- 4 libmpdata++ 2.0: new features under development
- 5 closing remarks

# Plan of the talk

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

# libmpdata++ 1.0: from hello-world to real-world problems

- support for integration in 1D, 2D & 3D

- support for multiple transported fields

- numerous MPDATA options implemented:

- `MPDATA::MPC` (MPC coupling)
- `MPDATA::MPC2` (MPC coupling)
- `MPDATA::MPC3` (MPC coupling)
- `MPDATA::MPC4` (MPC coupling)
- `MPDATA::MPC5` (MPC coupling)
- `MPDATA::MPC6` (MPC coupling)
- `MPDATA::MPC7` (MPC coupling)
- `MPDATA::MPC8` (MPC coupling)
- `MPDATA::MPC9` (MPC coupling)
- `MPDATA::MPC10` (MPC coupling)
- `MPDATA::MPC11` (MPC coupling)
- `MPDATA::MPC12` (MPC coupling)
- `MPDATA::MPC13` (MPC coupling)
- `MPDATA::MPC14` (MPC coupling)
- `MPDATA::MPC15` (MPC coupling)
- `MPDATA::MPC16` (MPC coupling)
- `MPDATA::MPC17` (MPC coupling)
- `MPDATA::MPC18` (MPC coupling)
- `MPDATA::MPC19` (MPC coupling)
- `MPDATA::MPC20` (MPC coupling)
- `MPDATA::MPC21` (MPC coupling)
- `MPDATA::MPC22` (MPC coupling)
- `MPDATA::MPC23` (MPC coupling)
- `MPDATA::MPC24` (MPC coupling)
- `MPDATA::MPC25` (MPC coupling)
- `MPDATA::MPC26` (MPC coupling)
- `MPDATA::MPC27` (MPC coupling)
- `MPDATA::MPC28` (MPC coupling)
- `MPDATA::MPC29` (MPC coupling)
- `MPDATA::MPC30` (MPC coupling)
- `MPDATA::MPC31` (MPC coupling)
- `MPDATA::MPC32` (MPC coupling)
- `MPDATA::MPC33` (MPC coupling)
- `MPDATA::MPC34` (MPC coupling)
- `MPDATA::MPC35` (MPC coupling)
- `MPDATA::MPC36` (MPC coupling)
- `MPDATA::MPC37` (MPC coupling)
- `MPDATA::MPC38` (MPC coupling)
- `MPDATA::MPC39` (MPC coupling)
- `MPDATA::MPC40` (MPC coupling)
- `MPDATA::MPC41` (MPC coupling)
- `MPDATA::MPC42` (MPC coupling)
- `MPDATA::MPC43` (MPC coupling)
- `MPDATA::MPC44` (MPC coupling)
- `MPDATA::MPC45` (MPC coupling)
- `MPDATA::MPC46` (MPC coupling)
- `MPDATA::MPC47` (MPC coupling)
- `MPDATA::MPC48` (MPC coupling)
- `MPDATA::MPC49` (MPC coupling)
- `MPDATA::MPC50` (MPC coupling)
- `MPDATA::MPC51` (MPC coupling)
- `MPDATA::MPC52` (MPC coupling)
- `MPDATA::MPC53` (MPC coupling)
- `MPDATA::MPC54` (MPC coupling)
- `MPDATA::MPC55` (MPC coupling)
- `MPDATA::MPC56` (MPC coupling)
- `MPDATA::MPC57` (MPC coupling)
- `MPDATA::MPC58` (MPC coupling)
- `MPDATA::MPC59` (MPC coupling)
- `MPDATA::MPC60` (MPC coupling)
- `MPDATA::MPC61` (MPC coupling)
- `MPDATA::MPC62` (MPC coupling)
- `MPDATA::MPC63` (MPC coupling)
- `MPDATA::MPC64` (MPC coupling)
- `MPDATA::MPC65` (MPC coupling)
- `MPDATA::MPC66` (MPC coupling)
- `MPDATA::MPC67` (MPC coupling)
- `MPDATA::MPC68` (MPC coupling)
- `MPDATA::MPC69` (MPC coupling)
- `MPDATA::MPC70` (MPC coupling)
- `MPDATA::MPC71` (MPC coupling)
- `MPDATA::MPC72` (MPC coupling)
- `MPDATA::MPC73` (MPC coupling)
- `MPDATA::MPC74` (MPC coupling)
- `MPDATA::MPC75` (MPC coupling)
- `MPDATA::MPC76` (MPC coupling)
- `MPDATA::MPC77` (MPC coupling)
- `MPDATA::MPC78` (MPC coupling)
- `MPDATA::MPC79` (MPC coupling)
- `MPDATA::MPC80` (MPC coupling)
- `MPDATA::MPC81` (MPC coupling)
- `MPDATA::MPC82` (MPC coupling)
- `MPDATA::MPC83` (MPC coupling)
- `MPDATA::MPC84` (MPC coupling)
- `MPDATA::MPC85` (MPC coupling)
- `MPDATA::MPC86` (MPC coupling)
- `MPDATA::MPC87` (MPC coupling)
- `MPDATA::MPC88` (MPC coupling)
- `MPDATA::MPC89` (MPC coupling)
- `MPDATA::MPC90` (MPC coupling)
- `MPDATA::MPC91` (MPC coupling)
- `MPDATA::MPC92` (MPC coupling)
- `MPDATA::MPC93` (MPC coupling)
- `MPDATA::MPC94` (MPC coupling)
- `MPDATA::MPC95` (MPC coupling)
- `MPDATA::MPC96` (MPC coupling)
- `MPDATA::MPC97` (MPC coupling)
- `MPDATA::MPC98` (MPC coupling)
- `MPDATA::MPC99` (MPC coupling)
- `MPDATA::MPC100` (MPC coupling)

- coordinate transformations

- open, cyclic, polar & rigid boundary conditions

- source-term handling

- shallow-water and Boussinesq dynamics

# libmpdata++ 1.0: from hello-world to real-world problems

- support for integration in 1D, 2D & 3D
- support for multiple transported fields
- numerous MPDATA options implemented:

```
MPDATA_OPTIONS(
    // 1D
    MPDATA_OPTION_1D,
    // 2D
    MPDATA_OPTION_2D,
    // 3D
    MPDATA_OPTION_3D,
    // coordinate transformations
    MPDATA_OPTION_COORD_TRANSFORMATIONS,
    // open, cyclic, polar & rigid boundary conditions
    MPDATA_OPTION_BOUNDARY_CONDITIONS,
    // source-term handling
    MPDATA_OPTION_SOURCE_TERM_HANDLING,
    // shallow-water and Boussinesq dynamics
    MPDATA_OPTION_SHALLOW_WATER_AND_BOUSSINESQ_DYNAMICS
);
```

- coordinate transformations
- open, cyclic, polar & rigid boundary conditions
- source-term handling
- shallow-water and Boussinesq dynamics

# libmpdata++ 1.0: from hello-world to real-world problems

- 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: from hello-world to real-world problems

- 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: from hello-world to real-world problems

- 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: from hello-world to real-world problems

- 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: from hello-world to real-world problems

- 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: from hello-world to real-world problems

- 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: from hello-world to real-world problems

- 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: from hello-world to real-world problems

- 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: from hello-world to real-world problems

- 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

Geosci. Model Dev., 8, 1005–1032, 2015  
www.geosci-model-dev.net/8/1005/2015/  
doi:10.5194/gmd-8-1005-2015  
© Author(s) 2015. CC Attribution 3.0 License.



Geoscientific  
Model Development



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

A. Jaruga<sup>1</sup>, S. Arabas<sup>1</sup>, D. Jarecka<sup>1,2</sup>, H. Pawłowska<sup>1</sup>, P. K. Smolarkiewicz<sup>3</sup>, and M. Waruszewski<sup>1</sup>

<sup>1</sup>Institute of Geophysics, Faculty of Physics, University of Warsaw, Warsaw, Poland

<sup>2</sup>National Center for Atmospheric Research, Boulder, CO, USA

<sup>3</sup>European Centre for Medium-Range Weather Forecasts, Reading, UK

Jaruga et al. 2015

Geosci. Model Dev., 8, 1005–1032, 2015

[www.geosci-model-dev.net/8/1005/2015/](http://www.geosci-model-dev.net/8/1005/2015/)

doi:10.5194/gmd-8-1005-2015

© Author(s) 2015. CC Attribution 3.0 License.



Geoscientific  
Model Development



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

A. Jaruga<sup>1</sup>, S. Arabas<sup>1</sup>, D. Jarecka<sup>1,2</sup>, H. Pawłowska<sup>1</sup>, P. K. Smolarkiewicz<sup>3</sup>, and M. Waruszewski<sup>1</sup>

<sup>1</sup>Institute of Geophysics, Faculty of Physics, University of Warsaw, Warsaw, Poland

<sup>2</sup>National Center for Atmospheric Research, Boulder, CO, USA

<sup>3</sup>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"*

Jaruga et al. 2015

Geosci. Model Dev., 8, 1005–1032, 2015

[www.geosci-model-dev.net/8/1005/2015/](http://www.geosci-model-dev.net/8/1005/2015/)

doi:10.5194/gmd-8-1005-2015

© Author(s) 2015. CC Attribution 3.0 License.



Geoscientific  
Model Development



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

A. Jaruga<sup>1</sup>, S. Arabas<sup>1</sup>, D. Jarecka<sup>1,2</sup>, H. Pawłowska<sup>1</sup>, P. K. Smolarkiewicz<sup>3</sup>, and M. Waruszewski<sup>1</sup>

<sup>1</sup>Institute of Geophysics, Faculty of Physics, University of Warsaw, Warsaw, Poland

<sup>2</sup>National Center for Atmospheric Research, Boulder, CO, USA

<sup>3</sup>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"*

Jaruga et al. 2015

Geosci. Model Dev., 8, 1005–1032, 2015

[www.geosci-model-dev.net/8/1005/2015/](http://www.geosci-model-dev.net/8/1005/2015/)

doi:10.5194/gmd-8-1005-2015

© Author(s) 2015. CC Attribution 3.0 License.



Geoscientific  
Model Development



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

A. Jaruga<sup>1</sup>, S. Arabas<sup>1</sup>, D. Jarecka<sup>1,2</sup>, H. Pawłowska<sup>1</sup>, P. K. Smolarkiewicz<sup>3</sup>, and M. Waruszewski<sup>1</sup>

<sup>1</sup>Institute of Geophysics, Faculty of Physics, University of Warsaw, Warsaw, Poland

<sup>2</sup>National Center for Atmospheric Research, Boulder, CO, USA

<sup>3</sup>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"*

Jaruga et al. 2015

Geosci. Model Dev., 8, 1005–1032, 2015

[www.geosci-model-dev.net/8/1005/2015/](http://www.geosci-model-dev.net/8/1005/2015/)

doi:10.5194/gmd-8-1005-2015

© Author(s) 2015. CC Attribution 3.0 License.



Geoscientific  
Model Development



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

A. Jaruga<sup>1</sup>, S. Arabas<sup>1</sup>, D. Jarecka<sup>1,2</sup>, H. Pawłowska<sup>1</sup>, P. K. Smolarkiewicz<sup>3</sup>, and M. Waruszewski<sup>1</sup>

<sup>1</sup>Institute of Geophysics, Faculty of Physics, University of Warsaw, Warsaw, Poland

<sup>2</sup>National Center for Atmospheric Research, Boulder, CO, USA

<sup>3</sup>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”*

Jaruga et al. 2015

Geosci. Model Dev., 8, 1005–1032, 2015

[www.geosci-model-dev.net/8/1005/2015/](http://www.geosci-model-dev.net/8/1005/2015/)

doi:10.5194/gmd-8-1005-2015

© Author(s) 2015. CC Attribution 3.0 License.



Geoscientific  
Model Development



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

A. Jaruga<sup>1</sup>, S. Arabas<sup>1</sup>, D. Jarecka<sup>1,2</sup>, H. Pawłowska<sup>1</sup>, P. K. Smolarkiewicz<sup>3</sup>, and M. Waruszewski<sup>1</sup>

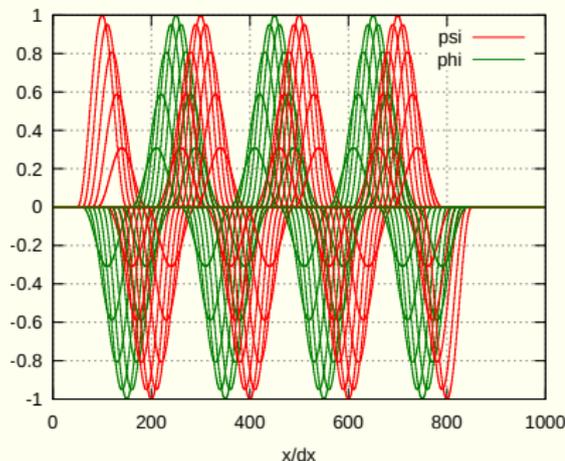
<sup>1</sup>Institute of Geophysics, Faculty of Physics, University of Warsaw, Warsaw, Poland

<sup>2</sup>National Center for Atmospheric Research, Boulder, CO, USA

<sup>3</sup>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”*



**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:

$$\begin{aligned}\psi_i^{n+1} &= \psi_i^* + 0.5 \Delta t \omega \phi_i^{n+1}, \\ \phi_i^{n+1} &= \phi_i^* - 0.5 \Delta t \omega \psi_i^{n+1},\end{aligned}\tag{17}$$

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

template <class ct_params_t>
struct coupled_harmonic : 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

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

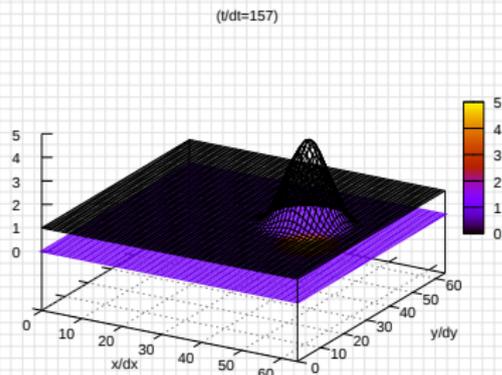
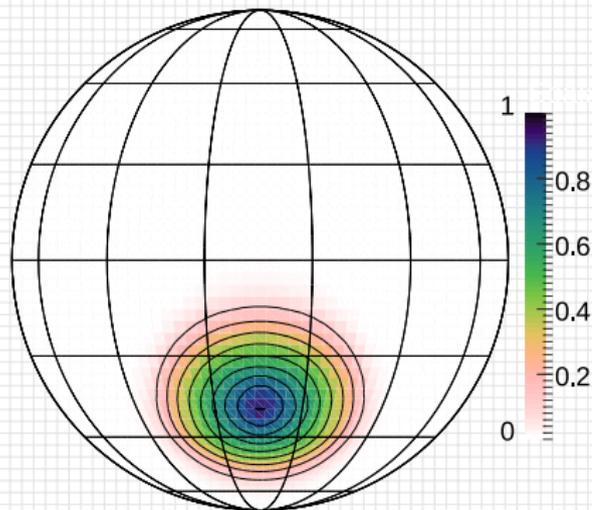
homogeneous advection

# libmpdata++ 1.0: solver/algorithm hierarchy

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

homogeneous advection

user/test  
code

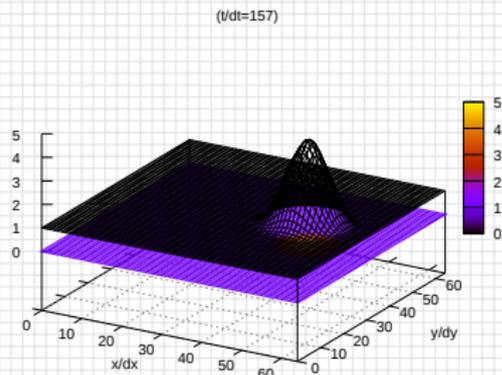
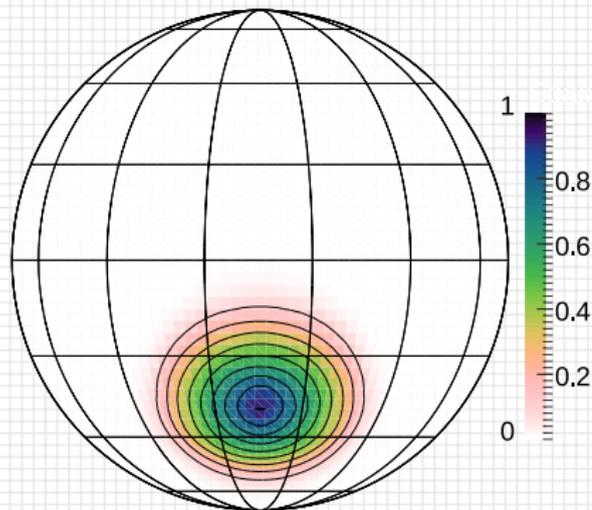


# libmpdata++ 1.0: solver/algorithm hierarchy

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

homogeneous advection

user/test  
code



# libmpdata++ 1.0: solver/algorithm hierarchy

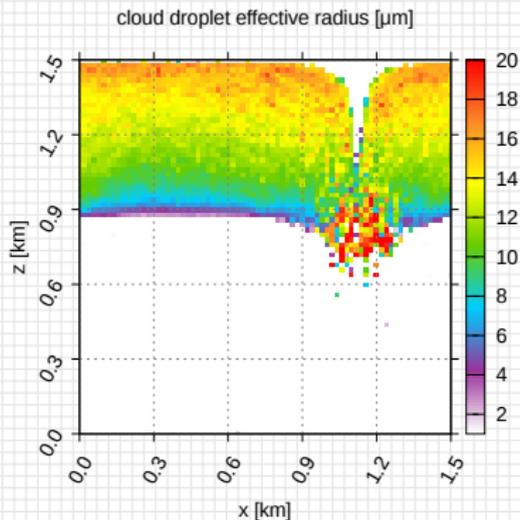
$$\partial_t(G\psi) + \nabla \cdot (G\bar{u}\psi) = 0$$

homogeneous advection

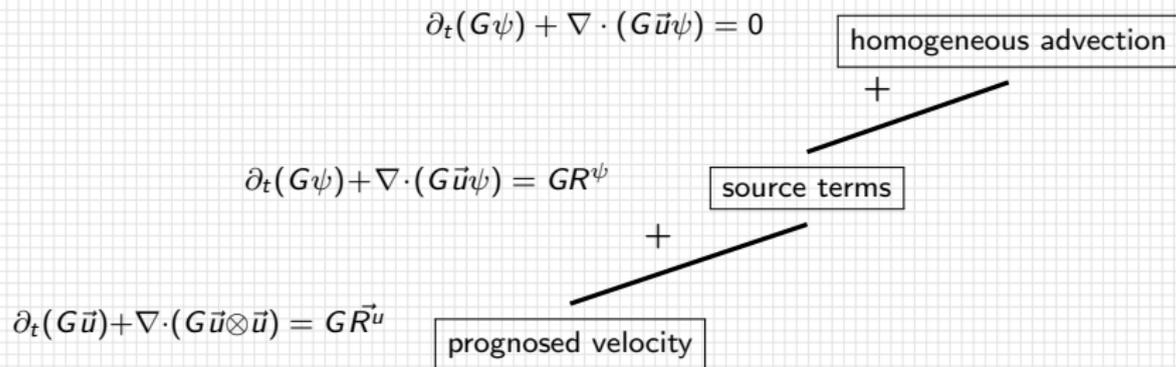
+

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

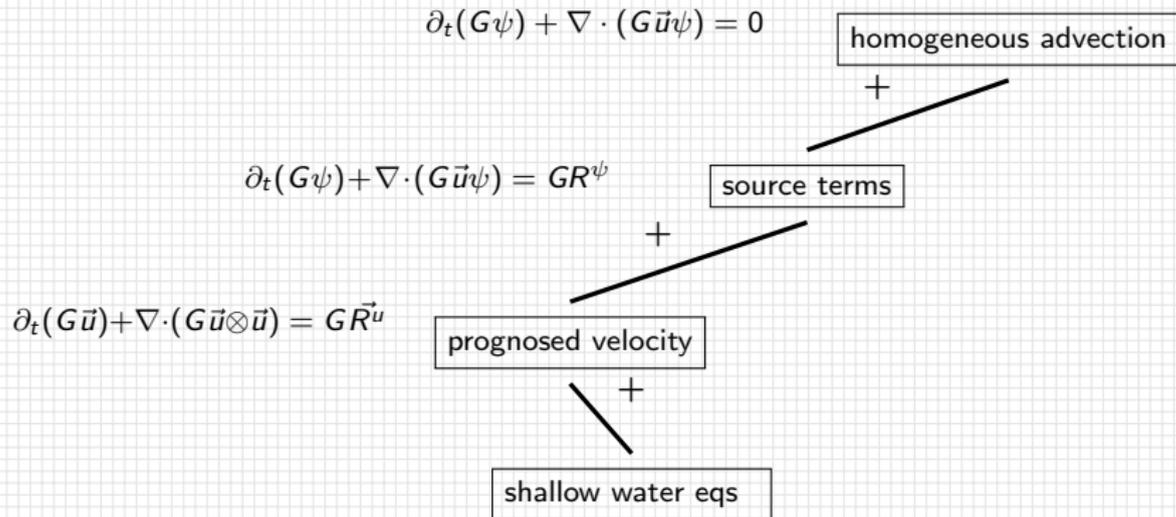
source terms



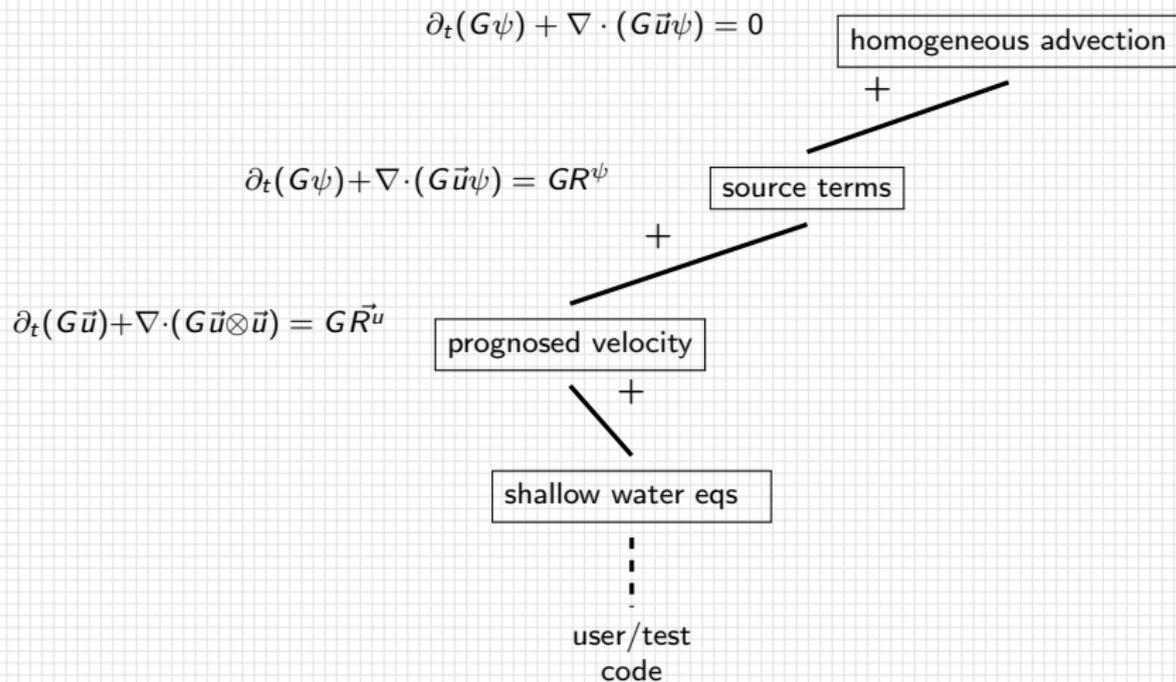
# libmpdata++ 1.0: solver/algorithm hierarchy



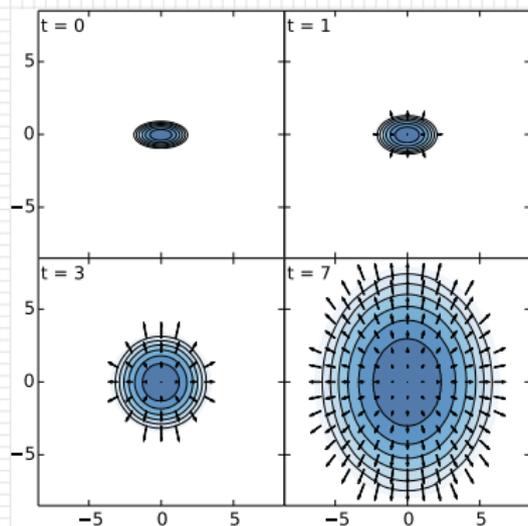
# libmpdata++ 1.0: solver/algorithm hierarchy



# libmpdata++ 1.0: solver/algorithm hierarchy

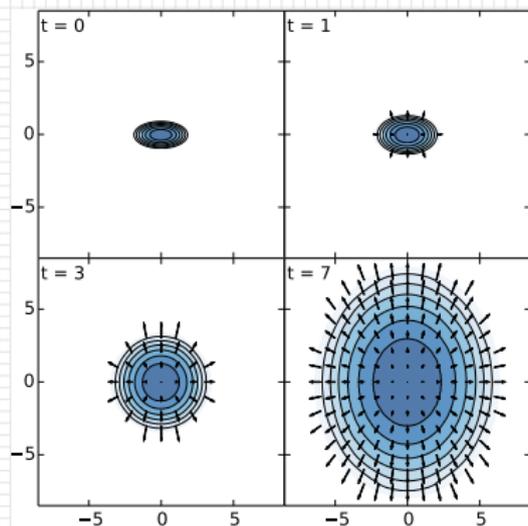


# libmpdata++: 3D shallow-water system example



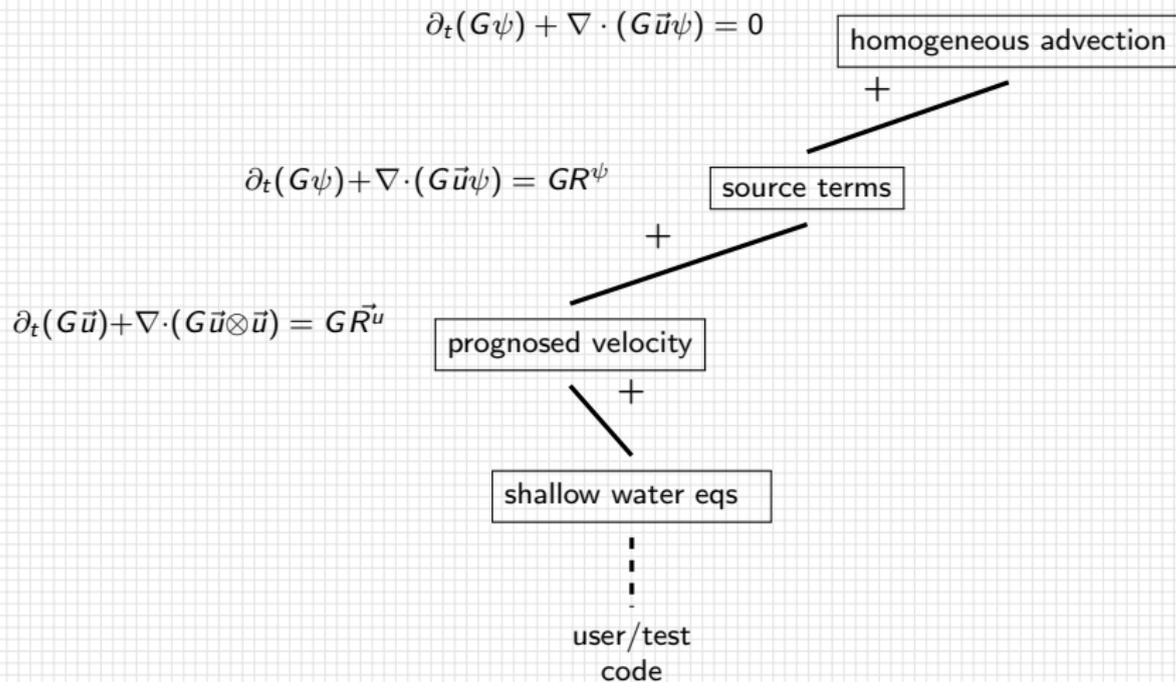
- inspired by 2D experiment of Schär and Smolarkiewicz, 1996
- example and original analytic solution by [Dorota Jarecka / NCAR](#) (Jarecka, Jaruga & Smolarkiewicz 2015, J. Comp. Phys. 289)

# libmpdata++: 3D shallow-water system example

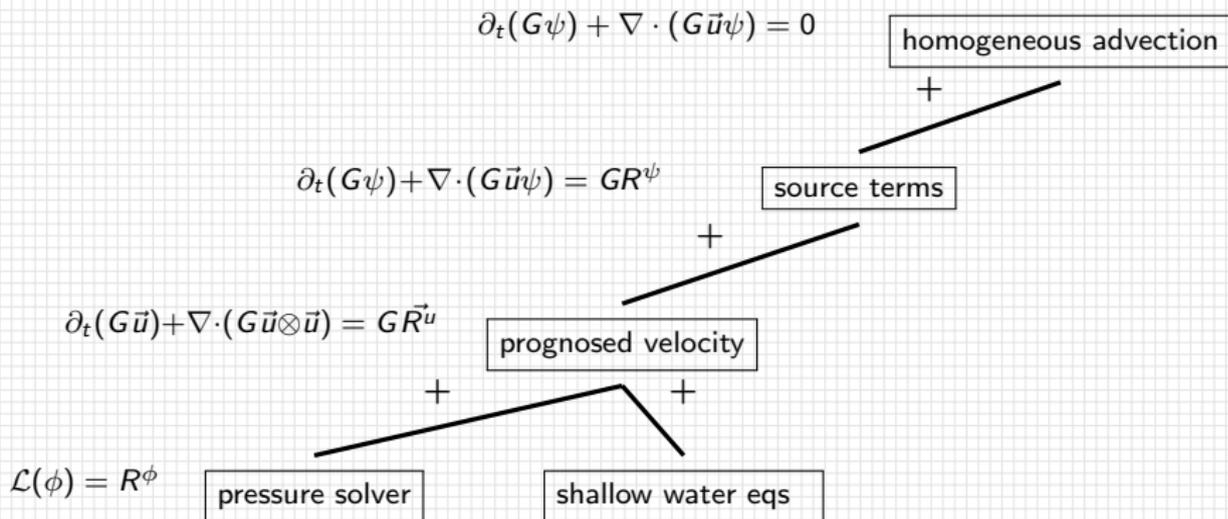


- inspired by 2D experiment of Schär and Smolarkiewicz, 1996
- example and original analytic solution by [Dorota Jarecka / NCAR](#) (Jarecka, Jaruga & Smolarkiewicz 2015, J. Comp. Phys. 289)

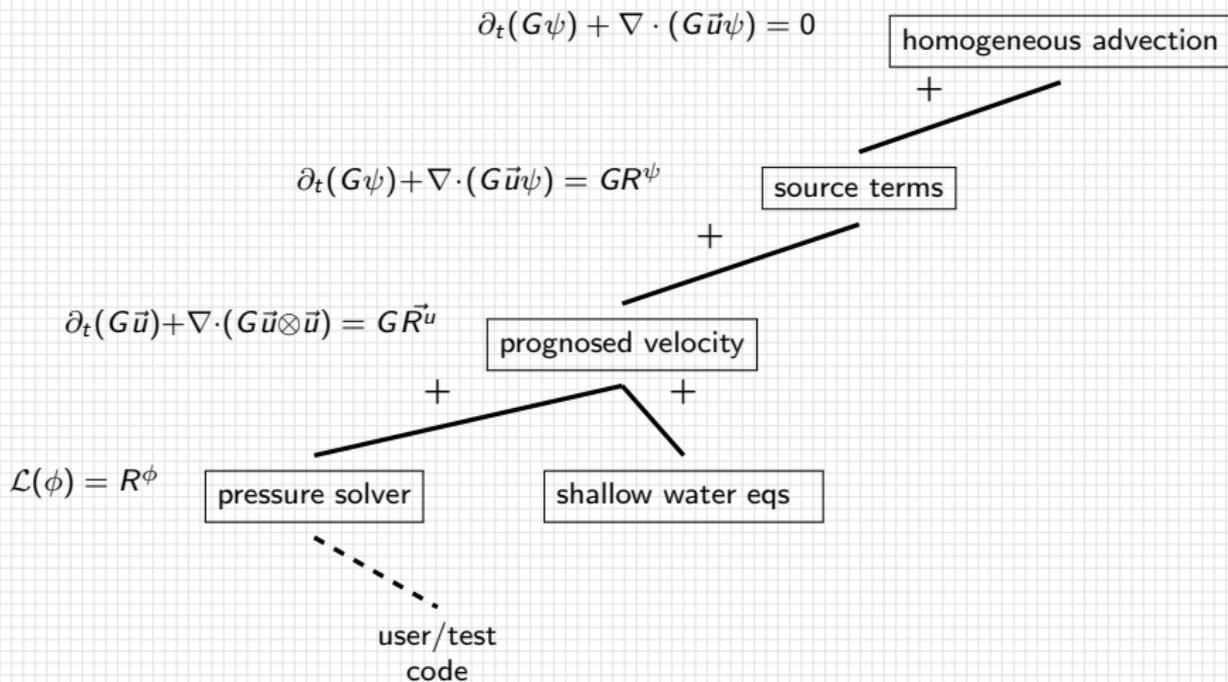
# libmpdata++ 1.0: solver/algorithm hierarchy



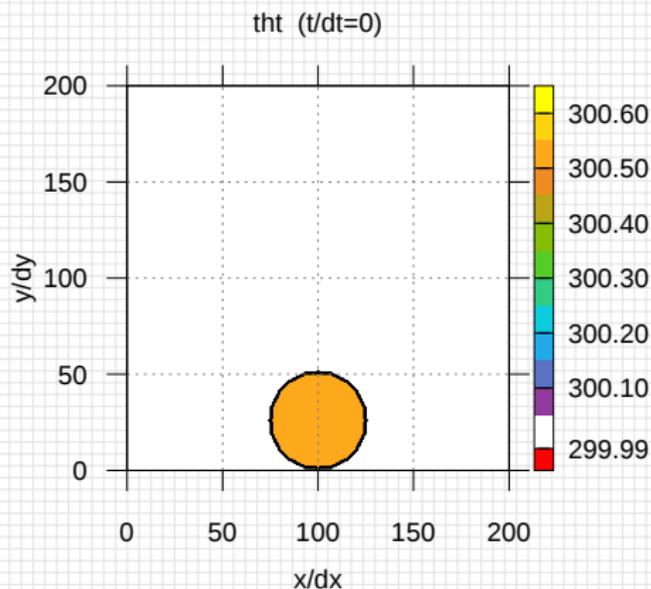
# libmpdata++ 1.0: solver/algorithm hierarchy



# libmpdata++ 1.0: solver/algorithm hierarchy



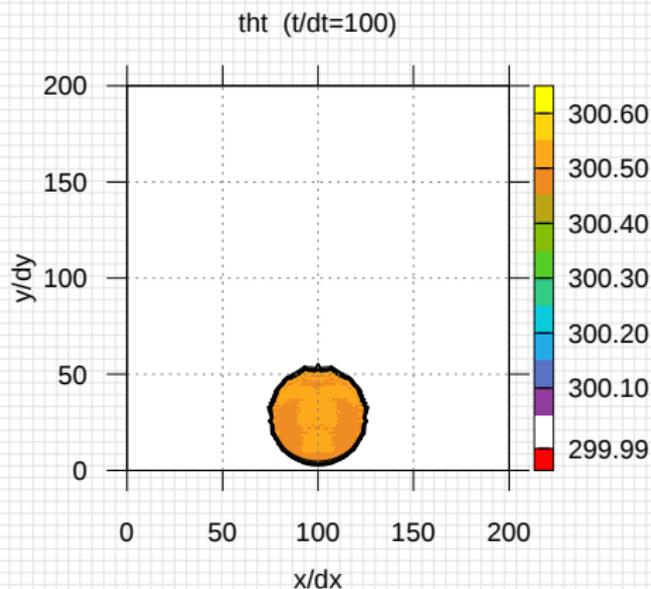
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

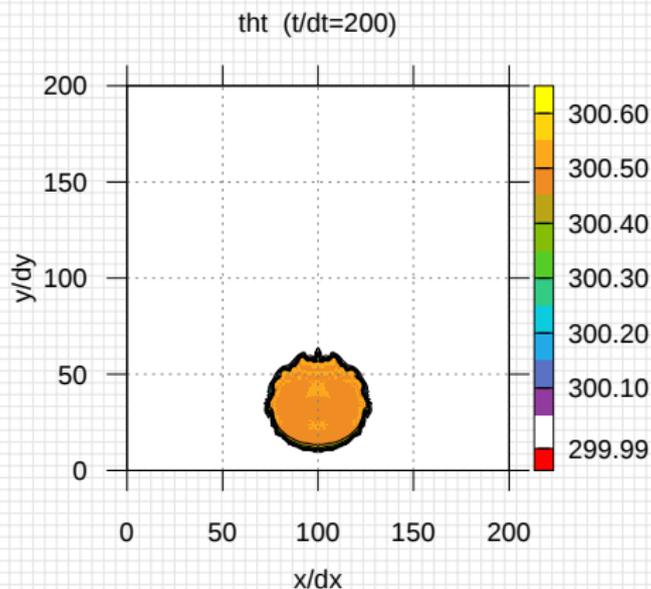
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

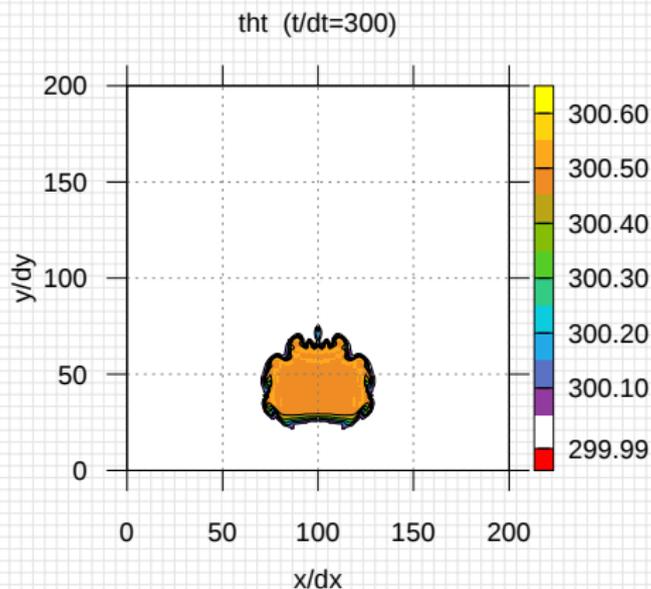
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

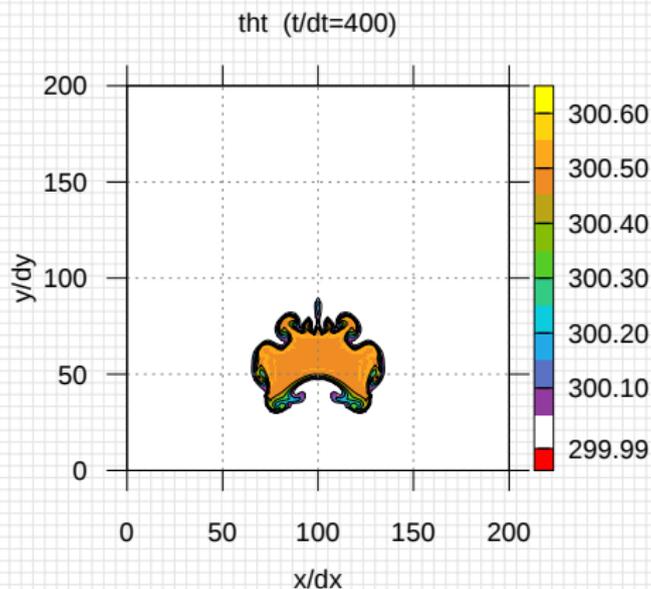
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

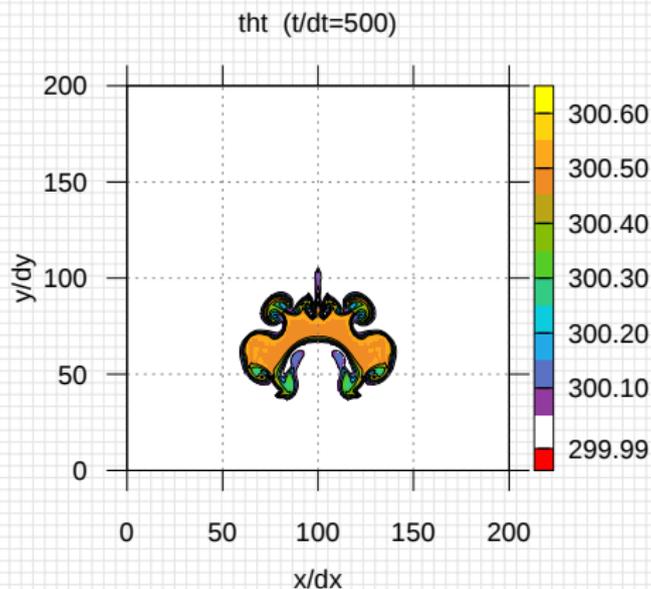
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

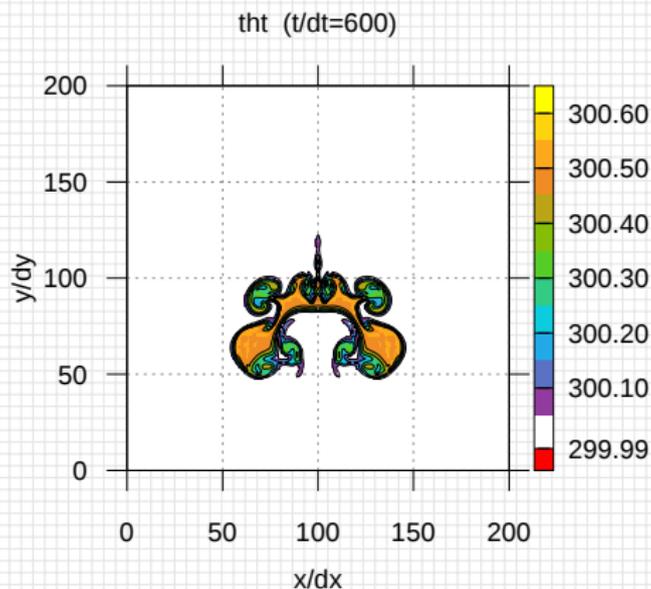
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

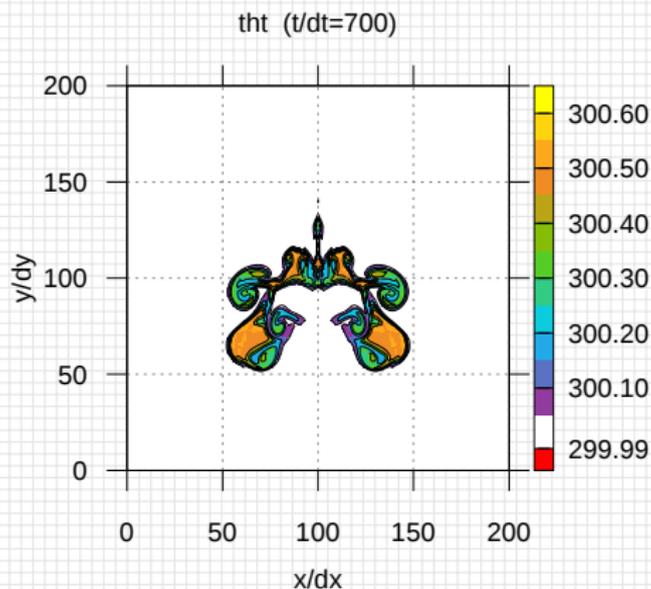
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

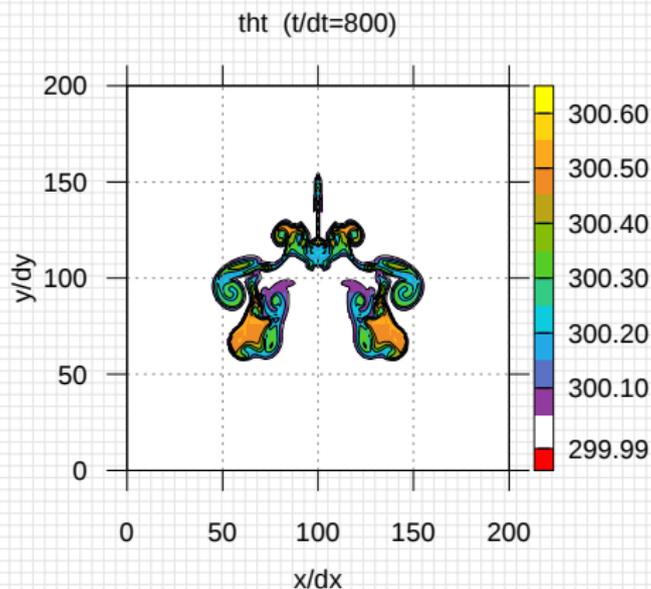
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

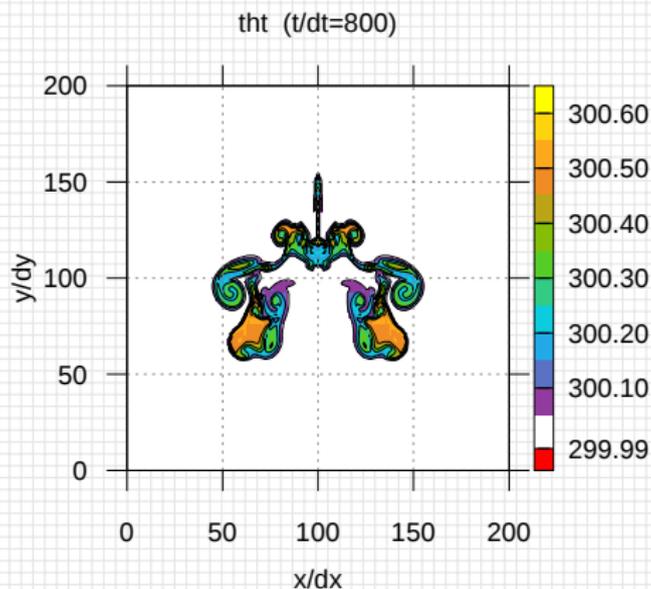
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

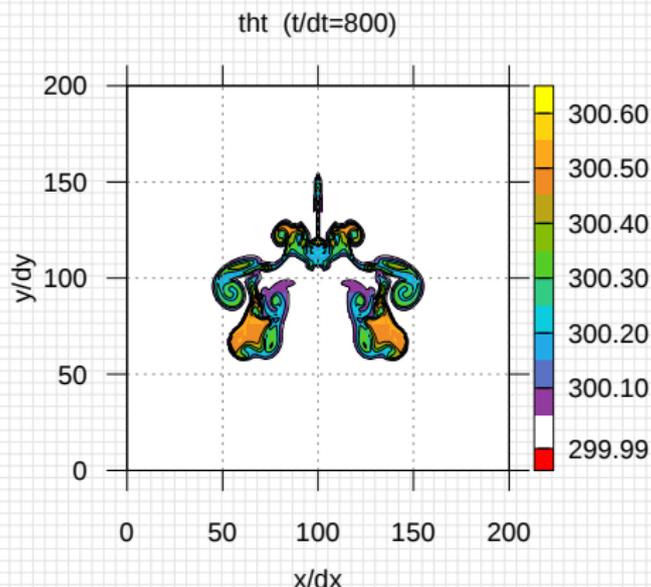
# libmpdata++: 2D Boussinesq convection example



- 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](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

# libmpdata++: 2D Boussinesq convection example

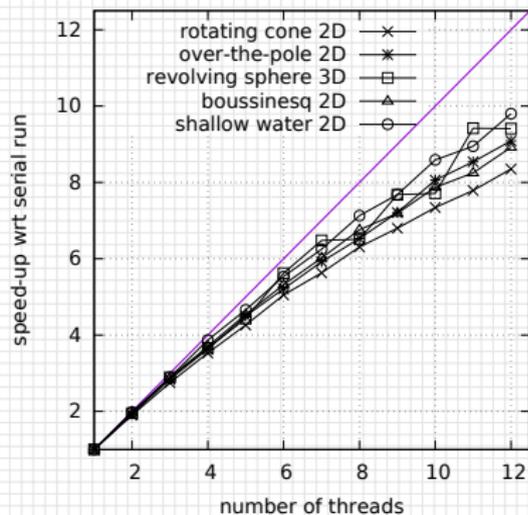


- reproduced experiment of Smolarkiewicz and Pudykiewicz, 1992
- **<200 lines of code** with libmpdata++

[https://github.com/igfuw/libmpdataxx/tree/master/tests/paper\\_2015\\_GMD/8\\_boussinesq\\_2d](https://github.com/igfuw/libmpdataxx/tree/master/tests/paper_2015_GMD/8_boussinesq_2d)

# libmpdata++: performance & scalability

# libmpdata++: performance & scalability

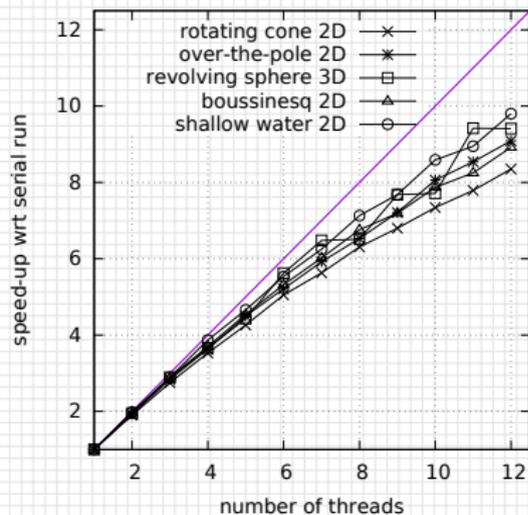


(3D, homogeneous  
advection, serial)

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

Analysis performed by Maciek Waruszewski, computational time granted by the Center for Cooperative Work on Computational Science of the University of Hyogo

# libmpdata++: performance & scalability

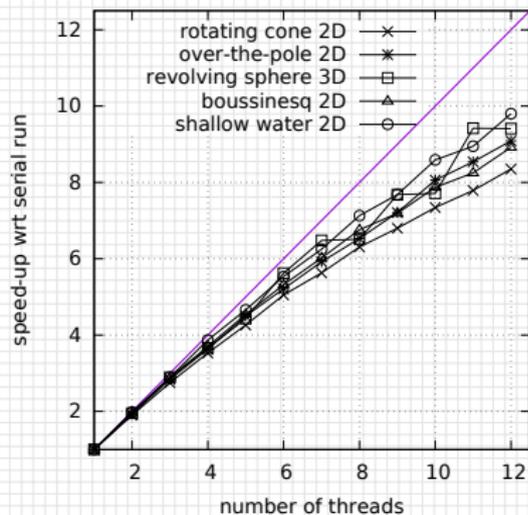


(3D, homogeneous advection, serial)

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

Analysis performed by Maciek Waruszewski, computational time granted by the Center for Cooperative Work on Computational Science of the University of Hyogo

# libmpdata++: performance & scalability



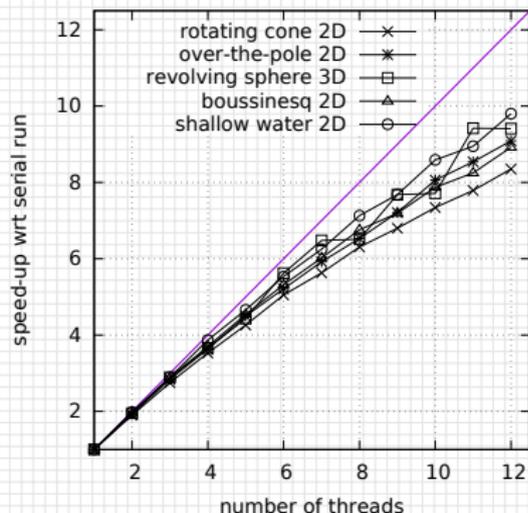
libmpdata++ / F77  
CPU-time ratios

(3D, homogeneous  
advection, serial)

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

Analysis performed by Maciek Waruszewski, computational time granted by the Center for Cooperative Work on Computational Science of the University of Hyogo

# libmpdata++: performance & scalability



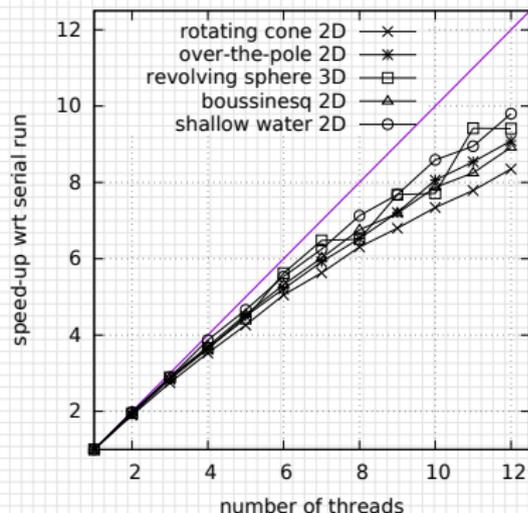
libmpdata++ / F77  
CPU-time ratios

(3D, homogeneous  
advection, serial)

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

Analysis performed by Maciek Waruszewski, computational time granted by the Center for Cooperative Work on Computational Science of the University of Hyogo

# libmpdata++: performance & scalability



libmpdata++ / F77  
CPU-time ratios

(3D, homogeneous  
advection, serial)

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

Analysis performed by Maciek Waruszewski, computational time granted by the Center for Cooperative Work on Computational Science of the University of Hyogo

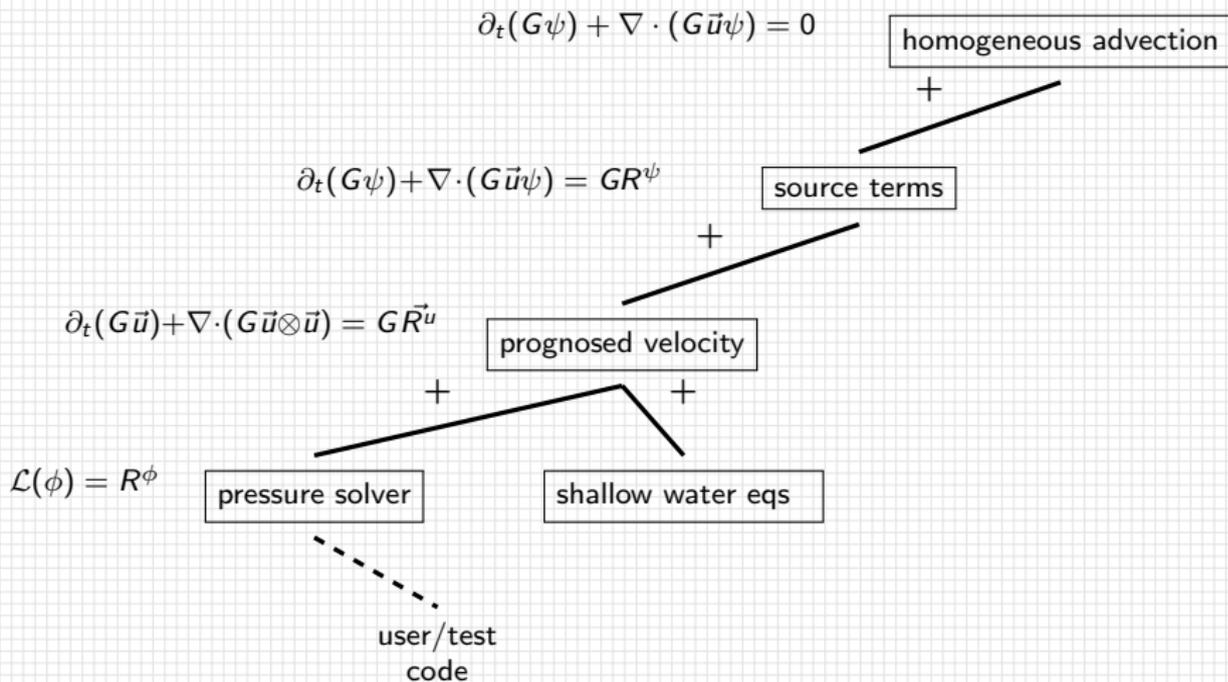
# Plan of the talk

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

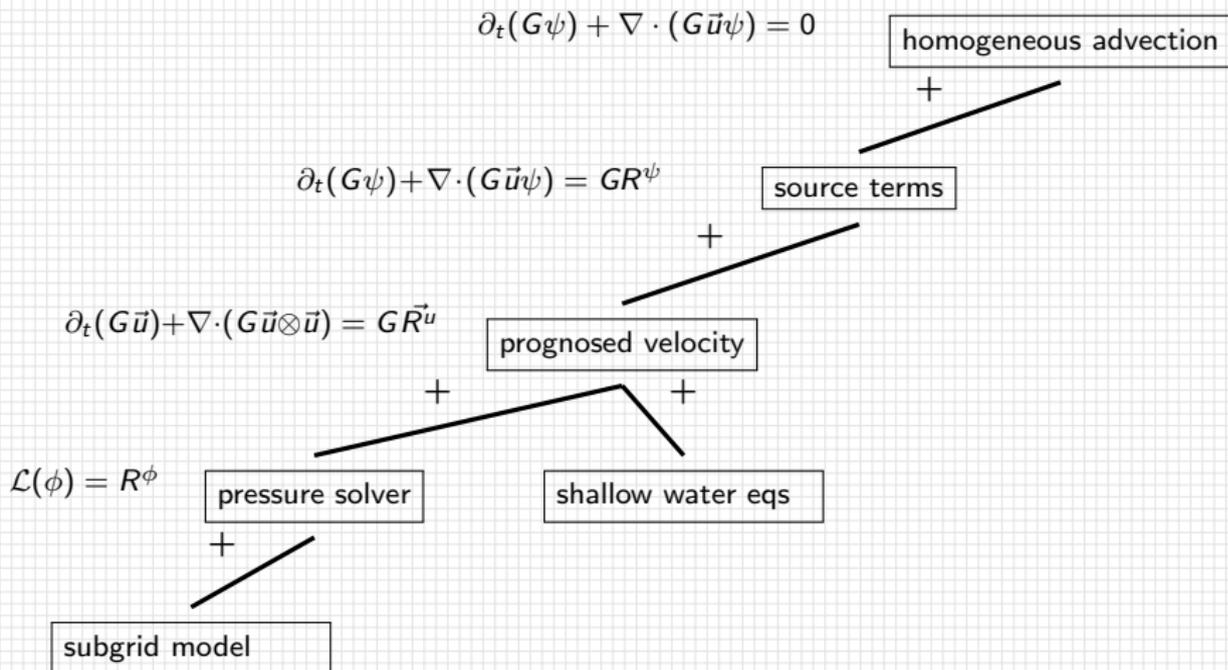
# Plan of the talk

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

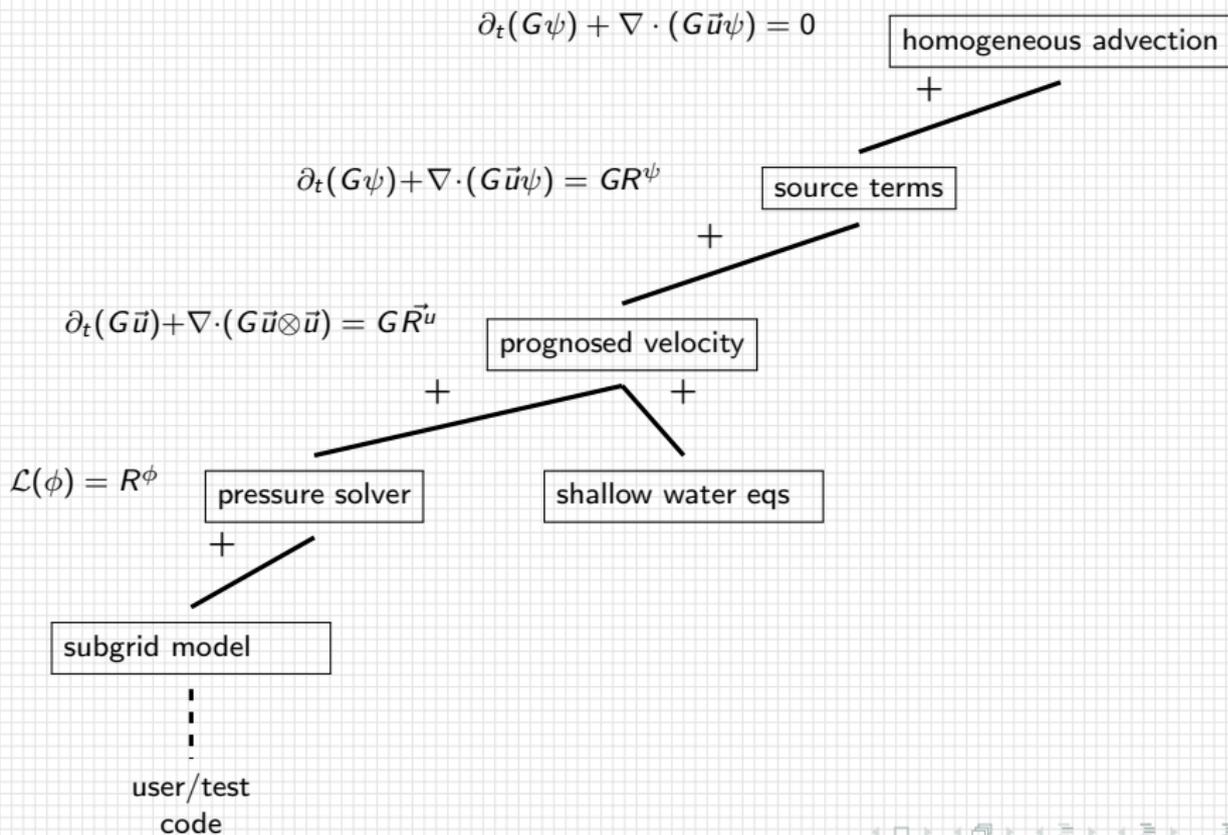
# libmpdata++ 1.0: solver/algorithm hierarchy



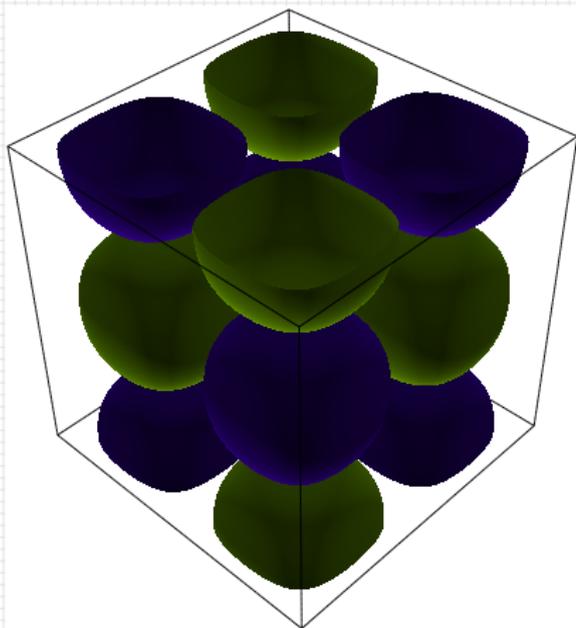
# libmpdata++ 2.0: solver/algorithm hierarchy



# libmpdata++ 2.0: solver/algorithm hierarchy



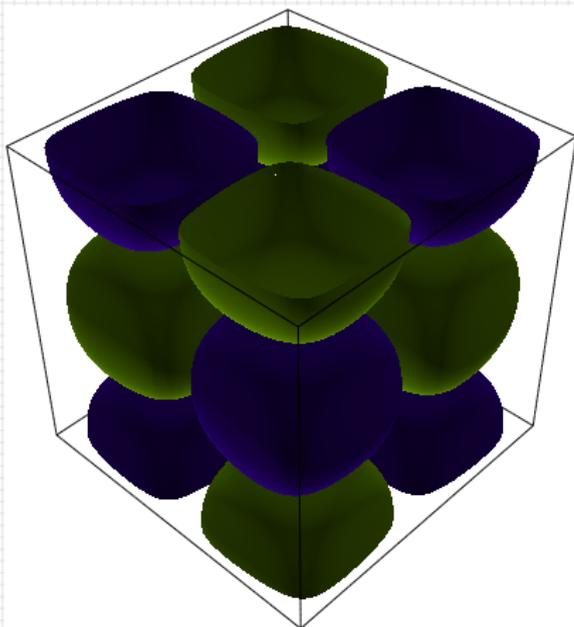
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

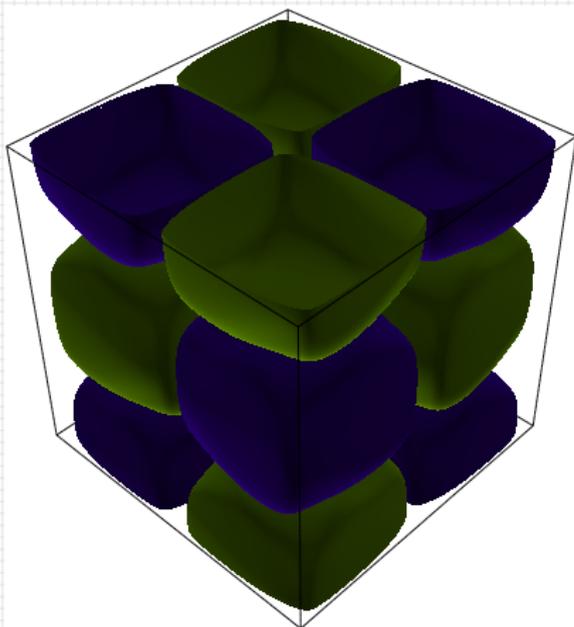
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

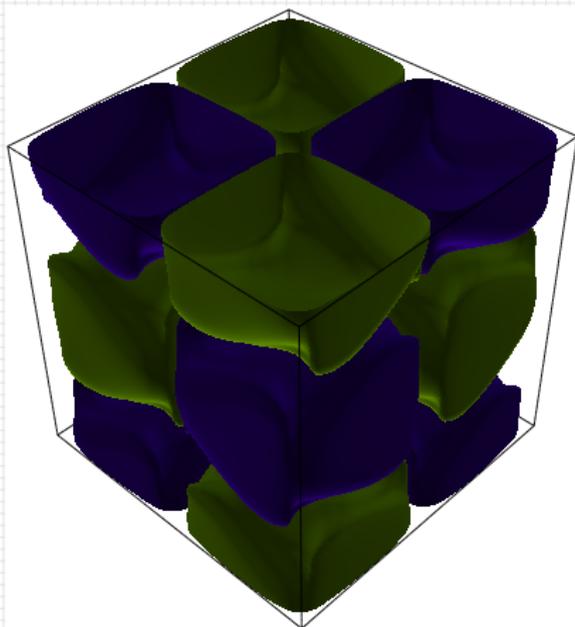
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

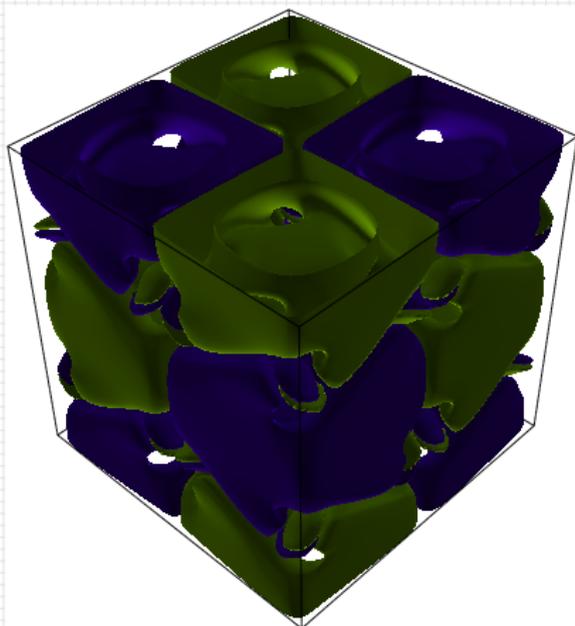
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

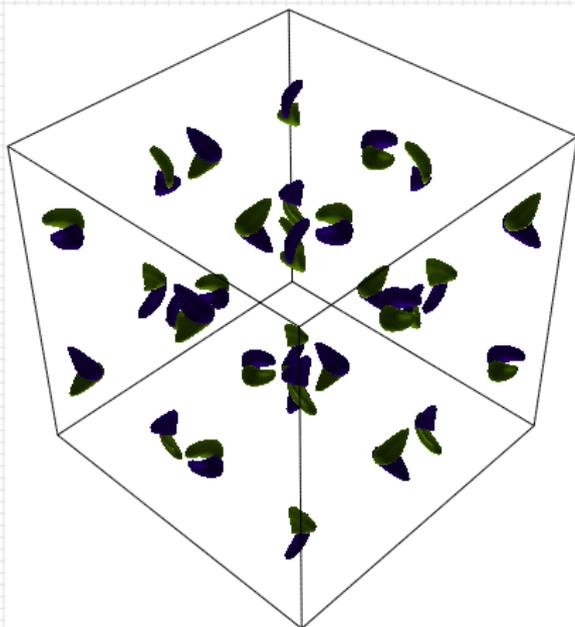
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

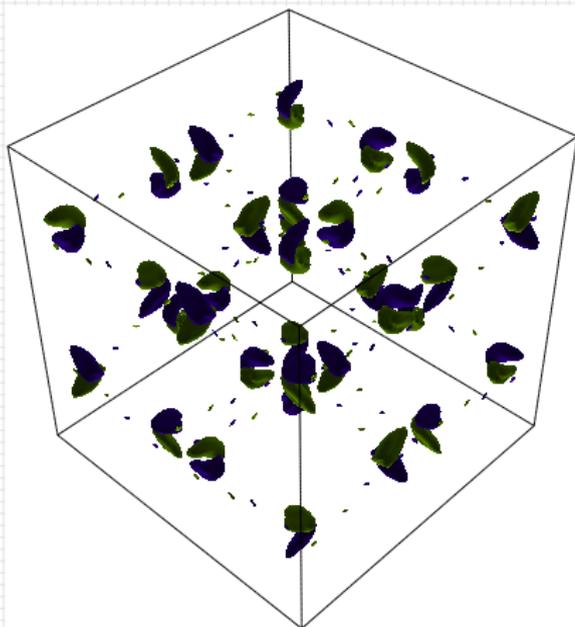
## libmpdata++ 2.0: Taylor-Green vortex example



- 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>

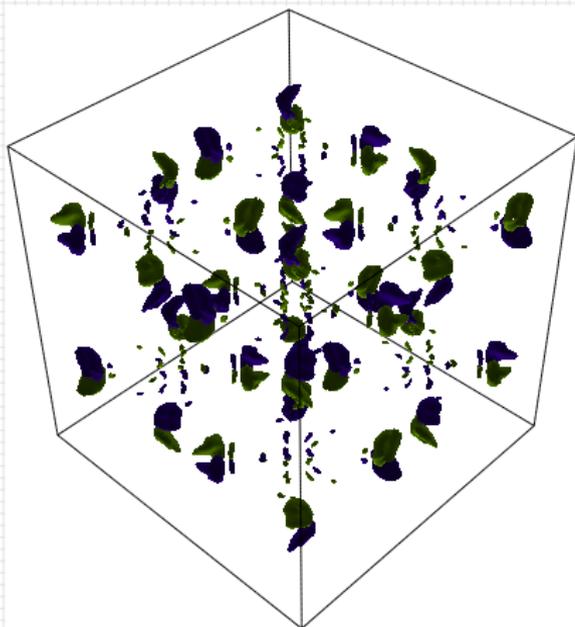
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

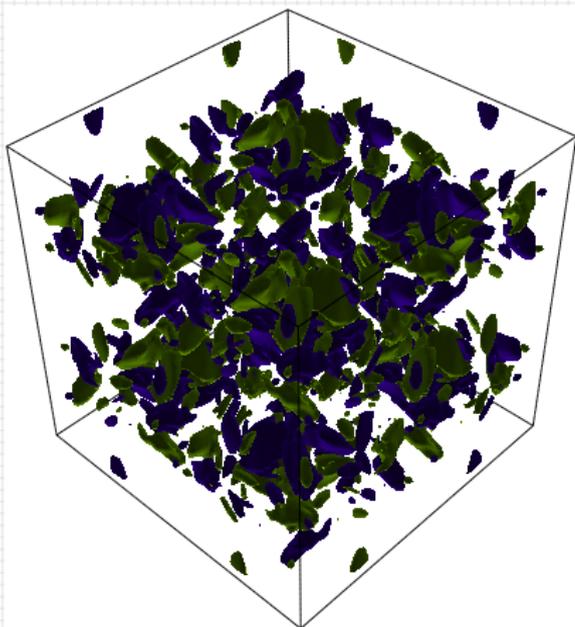
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

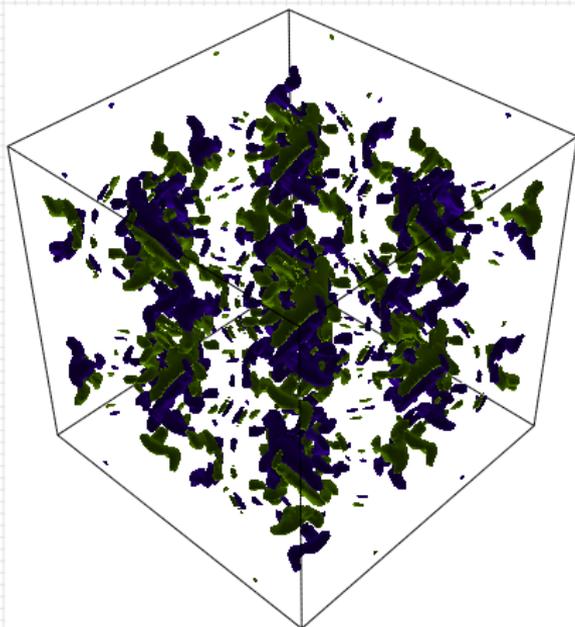
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

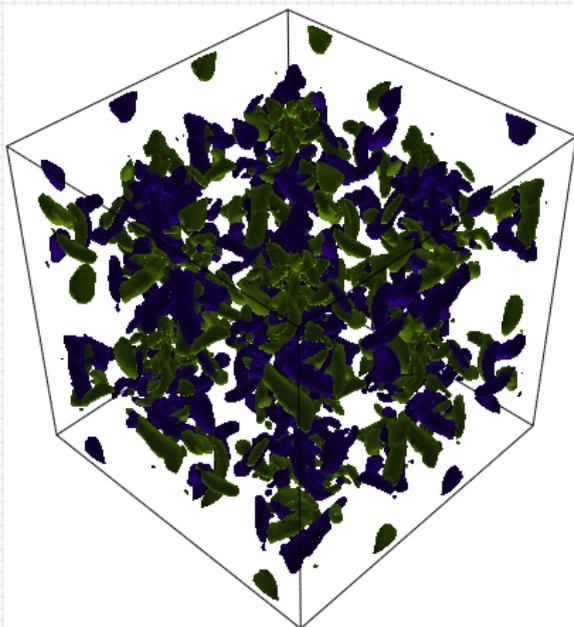
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

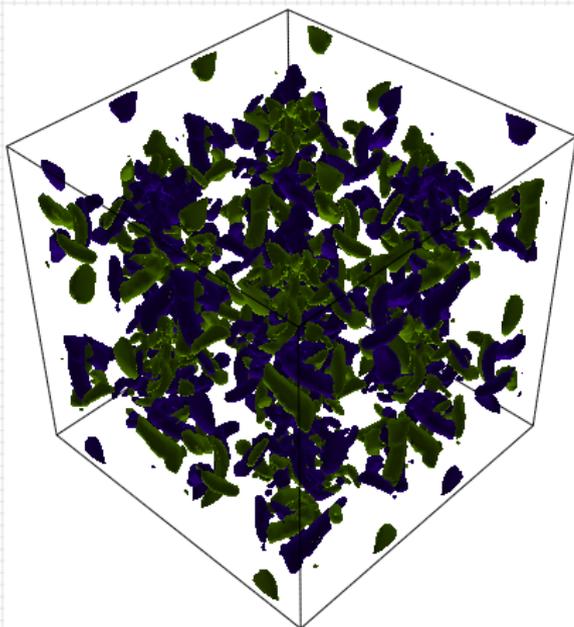
# libmpdata++ 2.0: Taylor-Green vortex example



- 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>

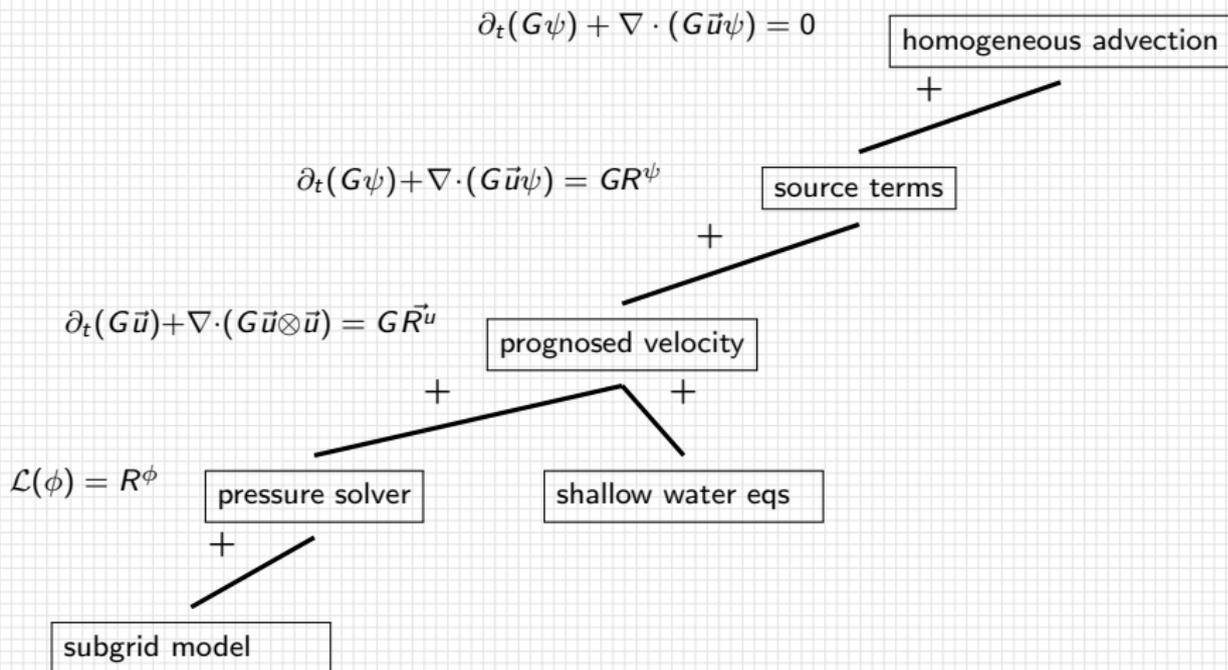
# libmpdata++ 2.0: Taylor-Green vortex example



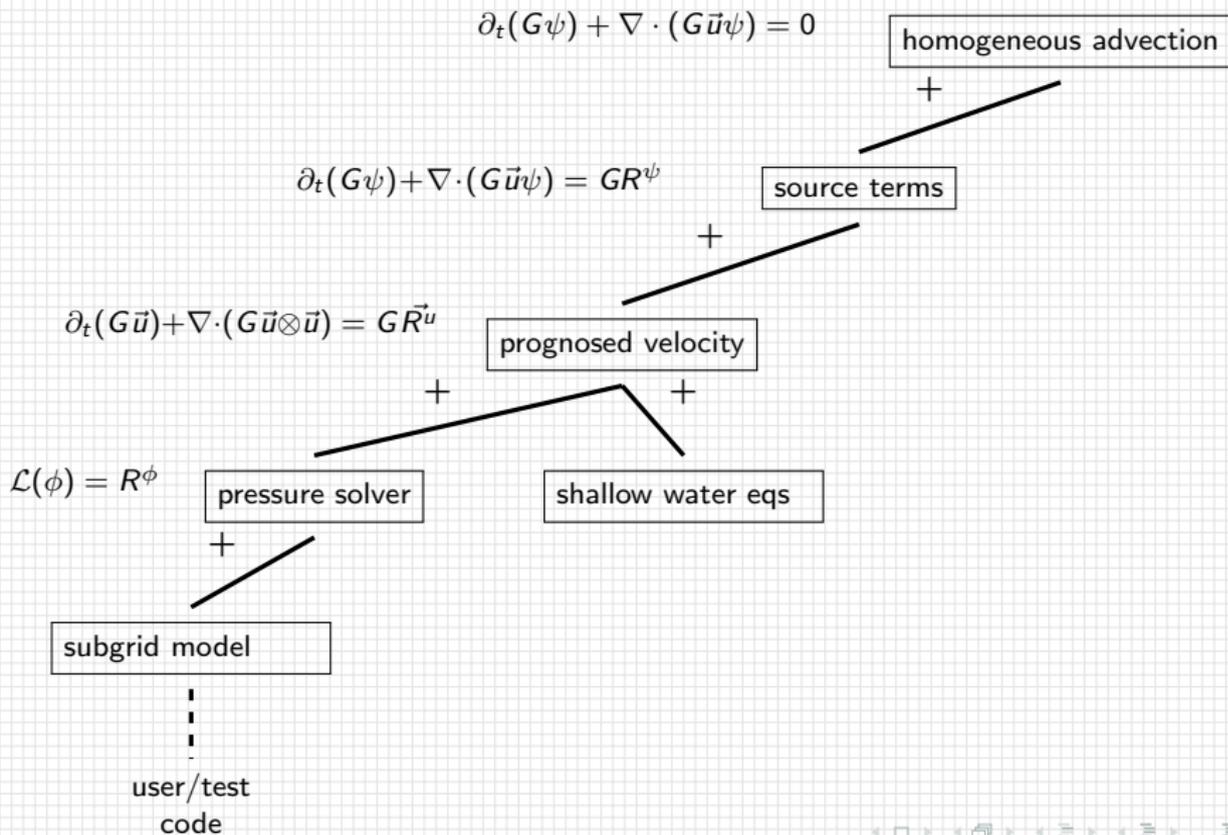
- inspired by spectral calculations of Brachet et al., 1983
- **<100 lines of code** with libmpdata++

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

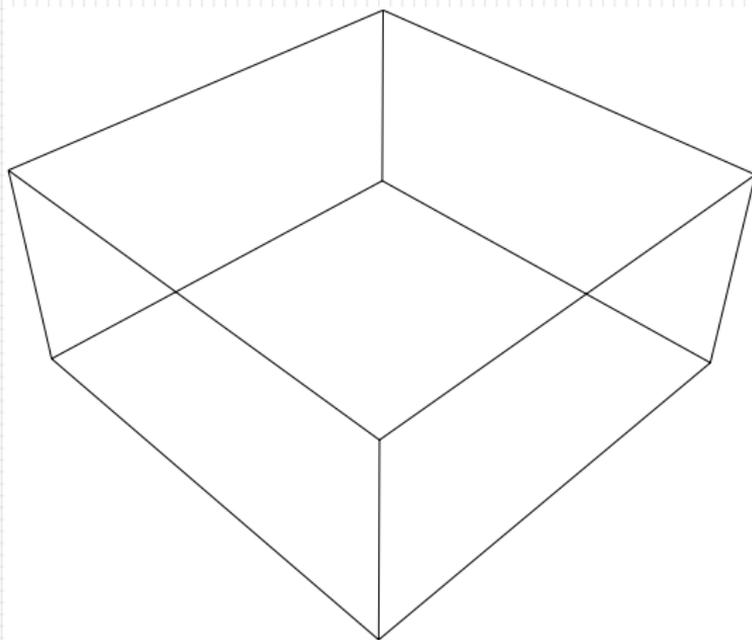
# libmpdata++ 2.0: solver/algorithm hierarchy



# libmpdata++ 2.0: solver/algorithm hierarchy

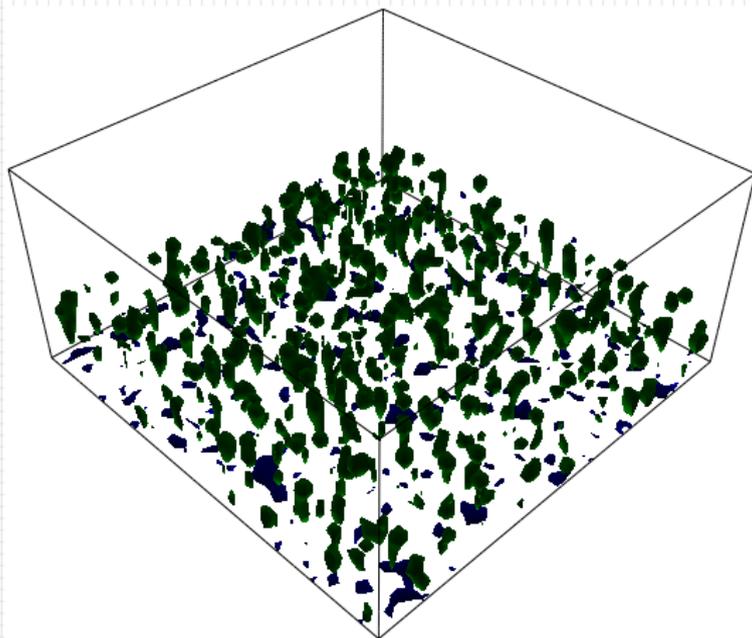


## libmpdata++ 2.0: convective boundary layer example



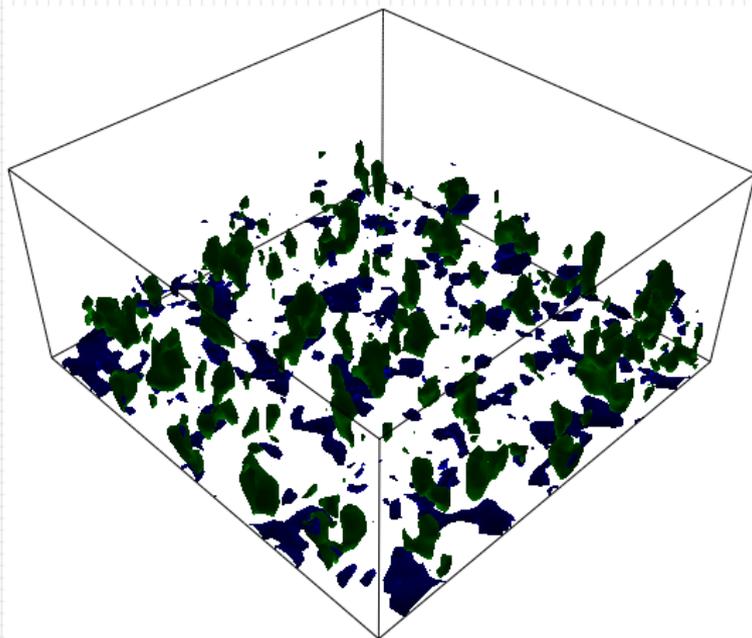
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



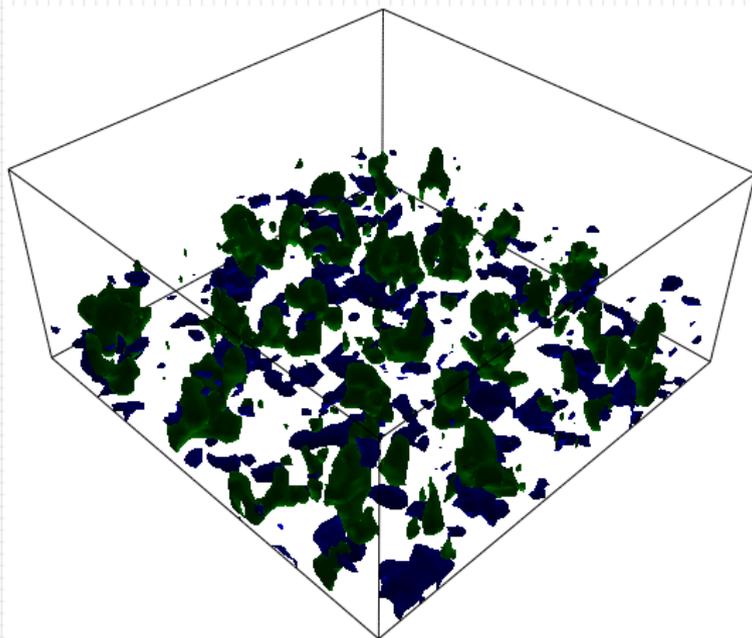
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



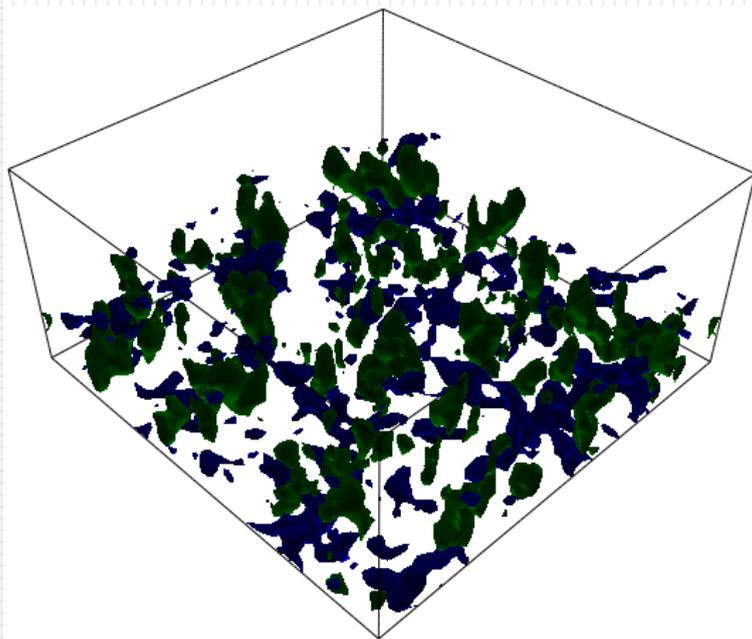
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



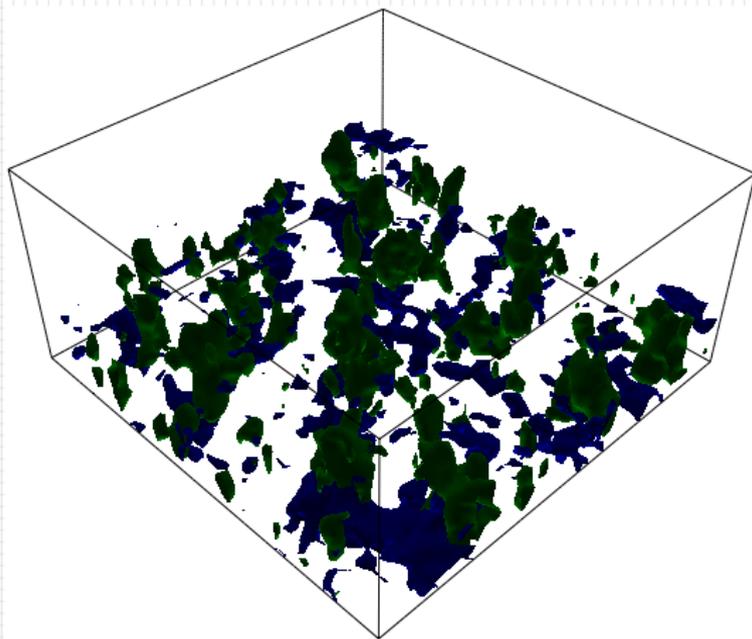
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



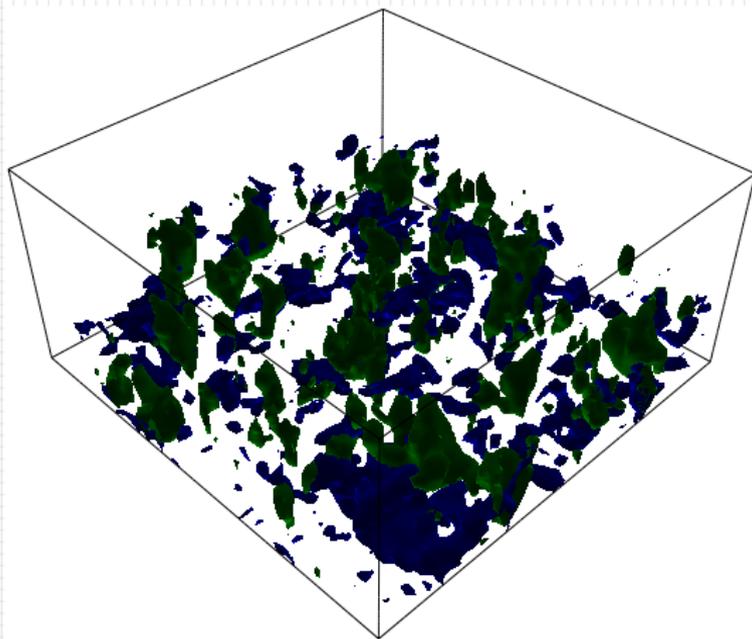
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



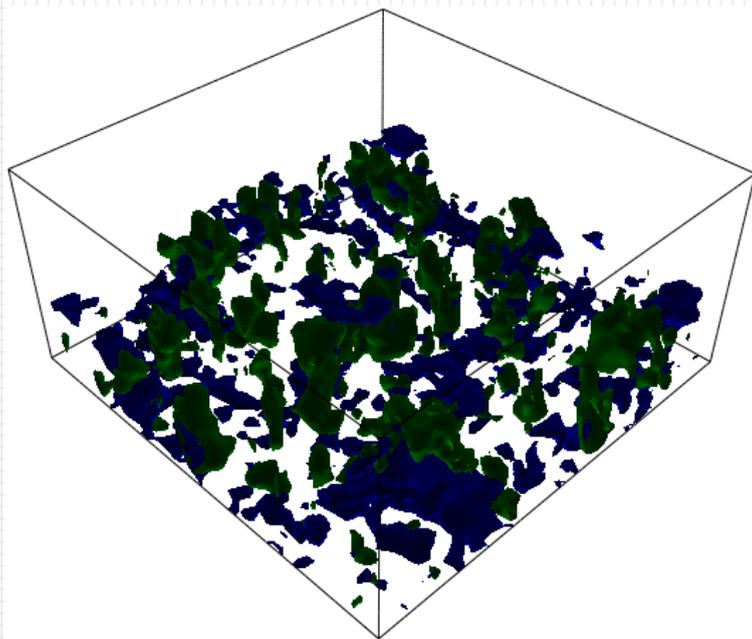
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



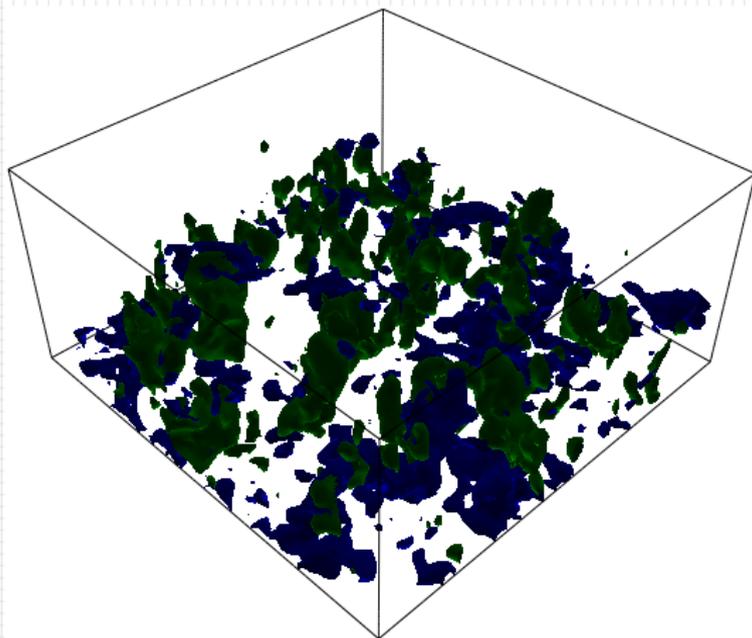
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



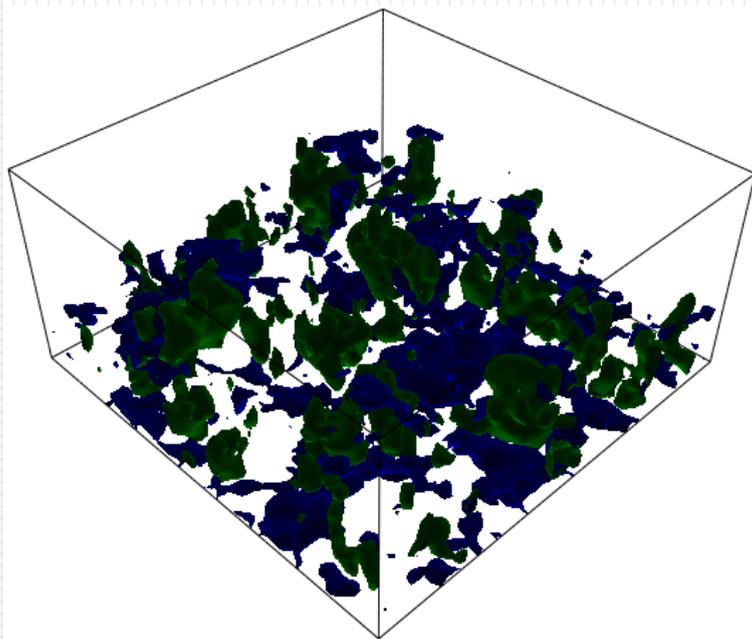
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



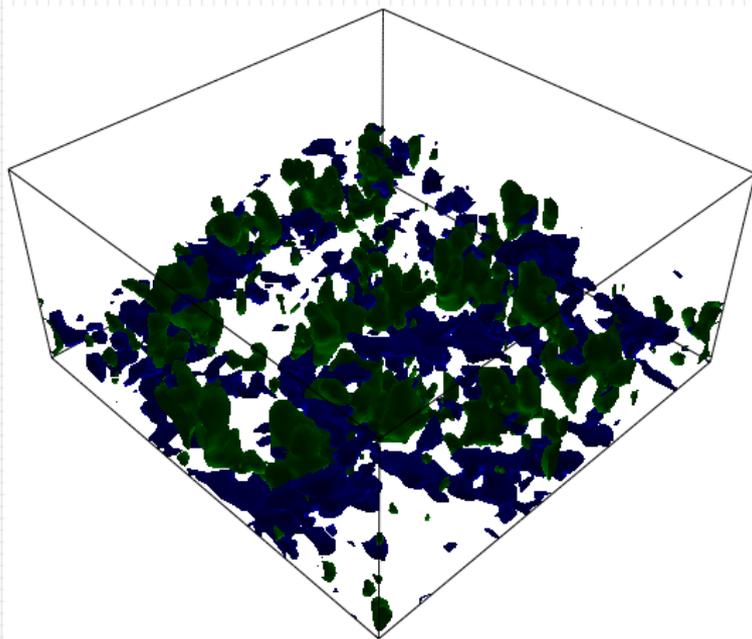
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



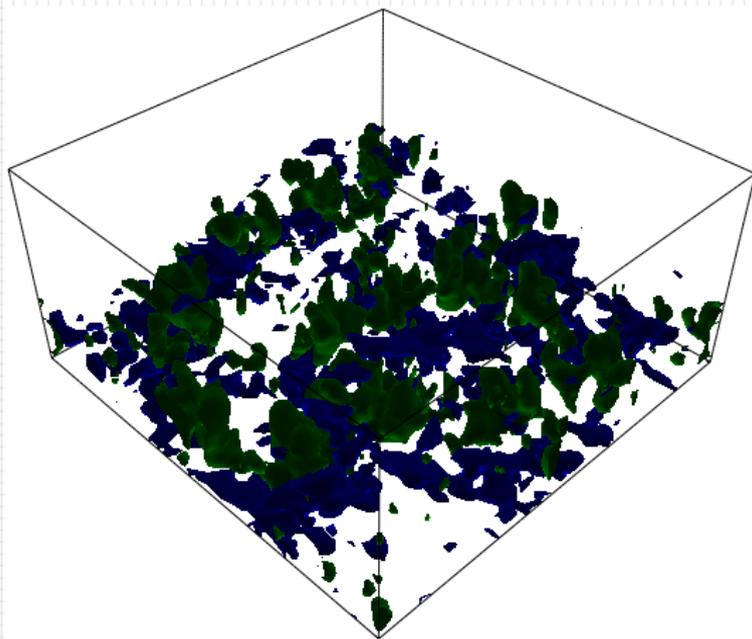
- iLES setup following Margolin et al., 1999
- <250 lines of code with libmpdata++

# libmpdata++ 2.0: convective boundary layer example



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

# libmpdata++ 2.0: convective boundary layer example



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

# libmpdata++ 2.0: summary of features under development

- **Support for 32-bit architectures**  
(`libmpdata++` will be able to compile on 32-bit systems)
- **Support for 64-bit floating point numbers**  
(`libmpdata++` will be able to compile on 64-bit systems)
- **Support for 64-bit integers**  
(`libmpdata++` will be able to compile on 64-bit systems)
- **Support for 64-bit pointers**  
(`libmpdata++` will be able to compile on 64-bit systems)



coded by  
Maciej Waruszewski

# libmpdata++ 2.0: summary of features under development

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

coded by  
Maciej Waruszewski

# libmpdata++ 2.0: summary of features under development

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

coded by  
Maciej Waruszewski

# libmpdata++ 2.0: summary of features under development

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

coded by  
Maciej Waruszewski

# libmpdata++ 2.0: summary of features under development

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

coded by  
Maciej Waruszewski

# libmpdata++ 2.0: summary of features under development

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

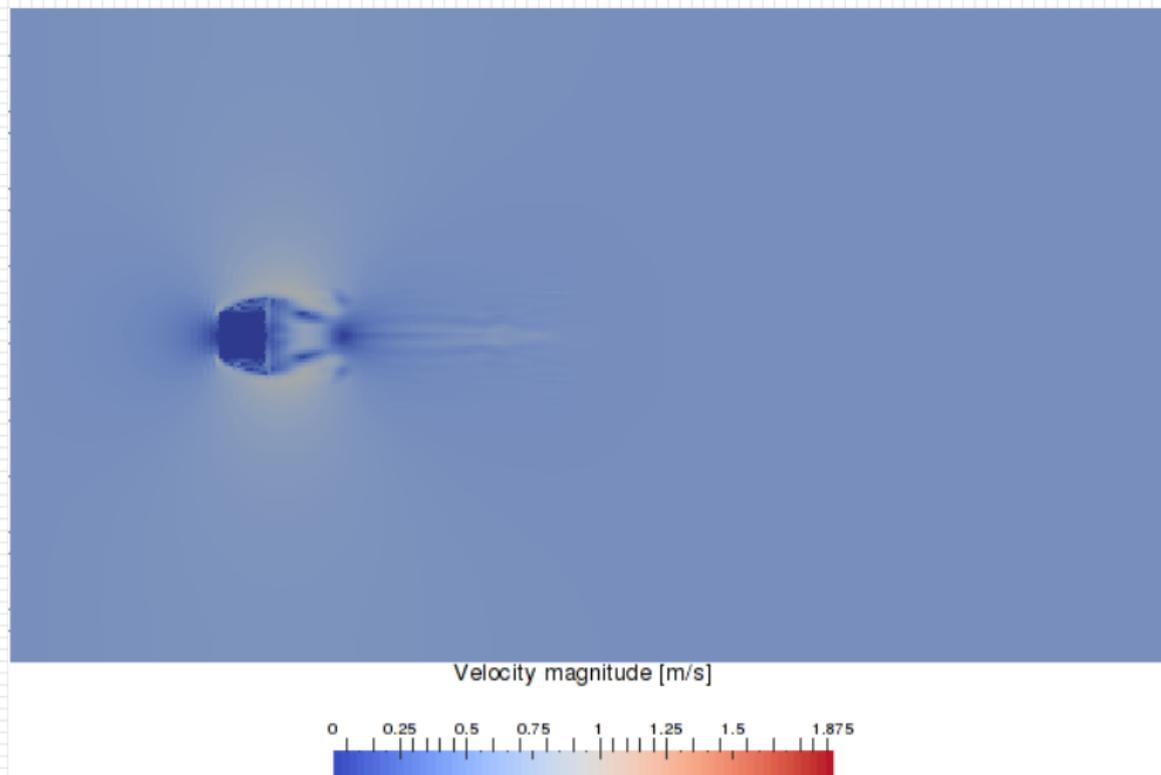
coded by  
Maciej Waruszewski

# libmpdata++ 2.0: summary of features under development

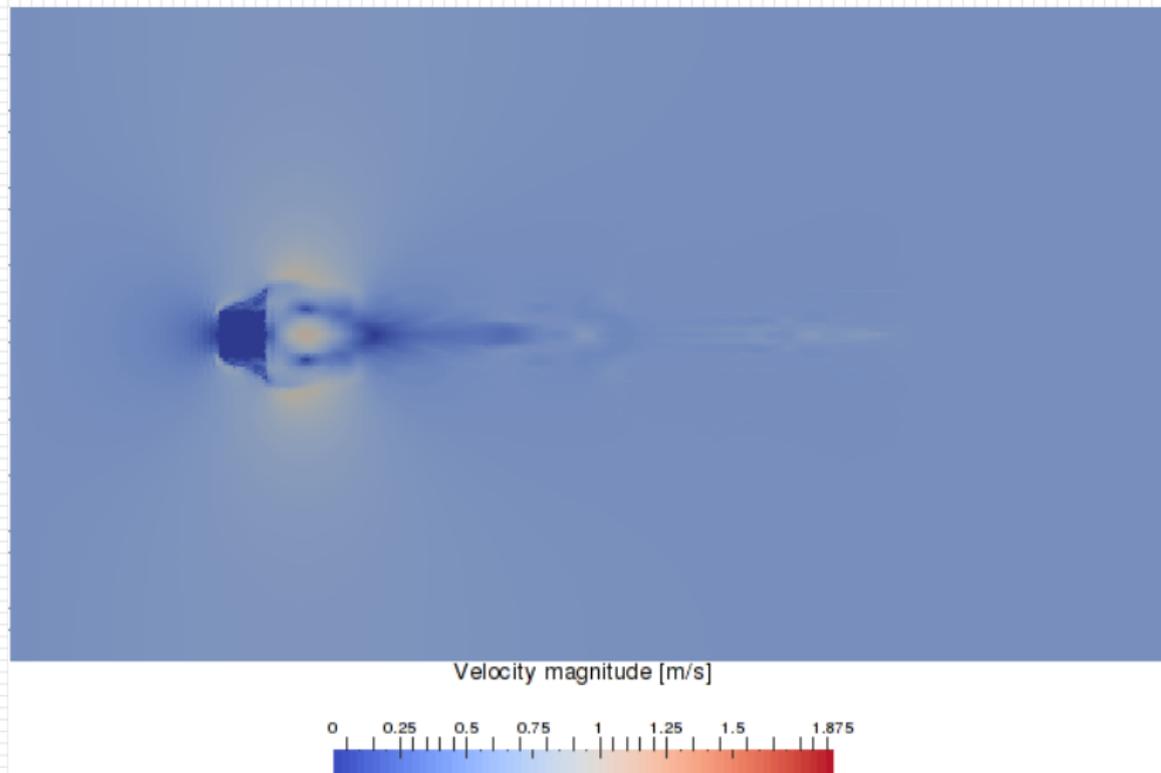
- higher-order operators  
(for DNS/iLES simulations)
- adaptive timestepping
- implicit treatment of absorbers  
(for immersed-boundary method)
- distributed-memory parallelisation  
(using Boost.MPI & HDF5/MPI-IO)

coded by  
Maciej Waruszewski

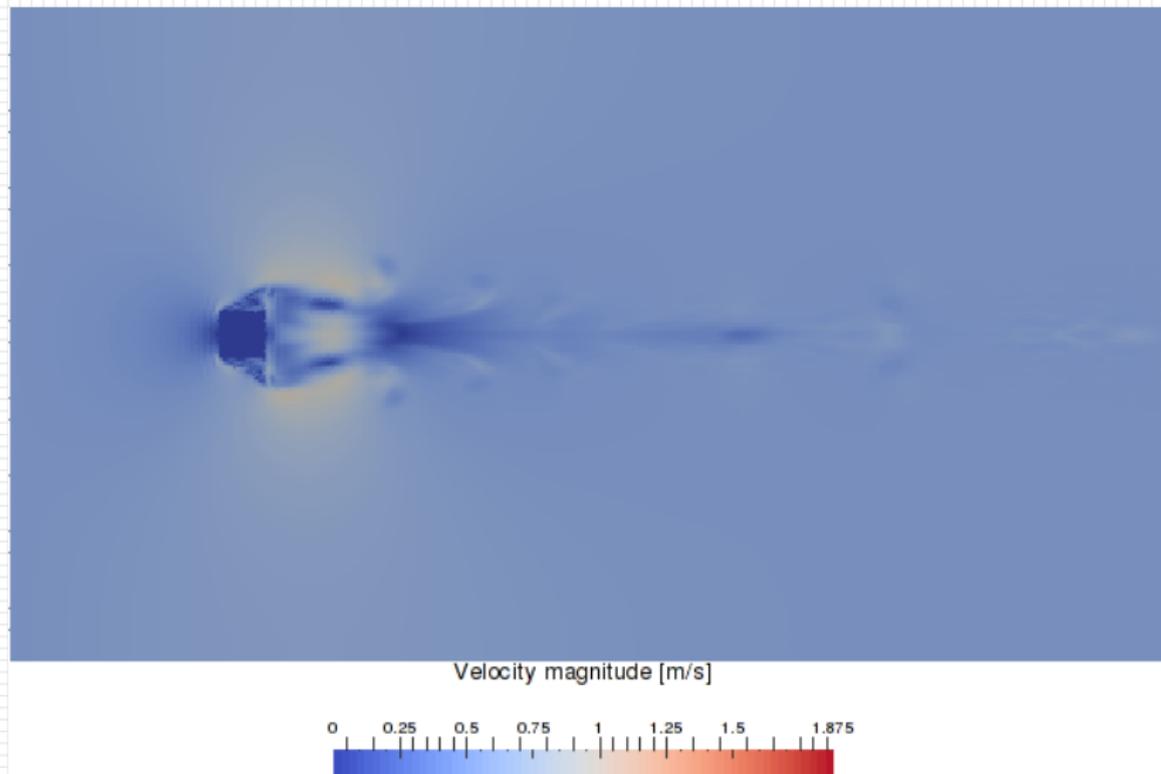
# libmpdata++ 2.0: immersed boundary teaser



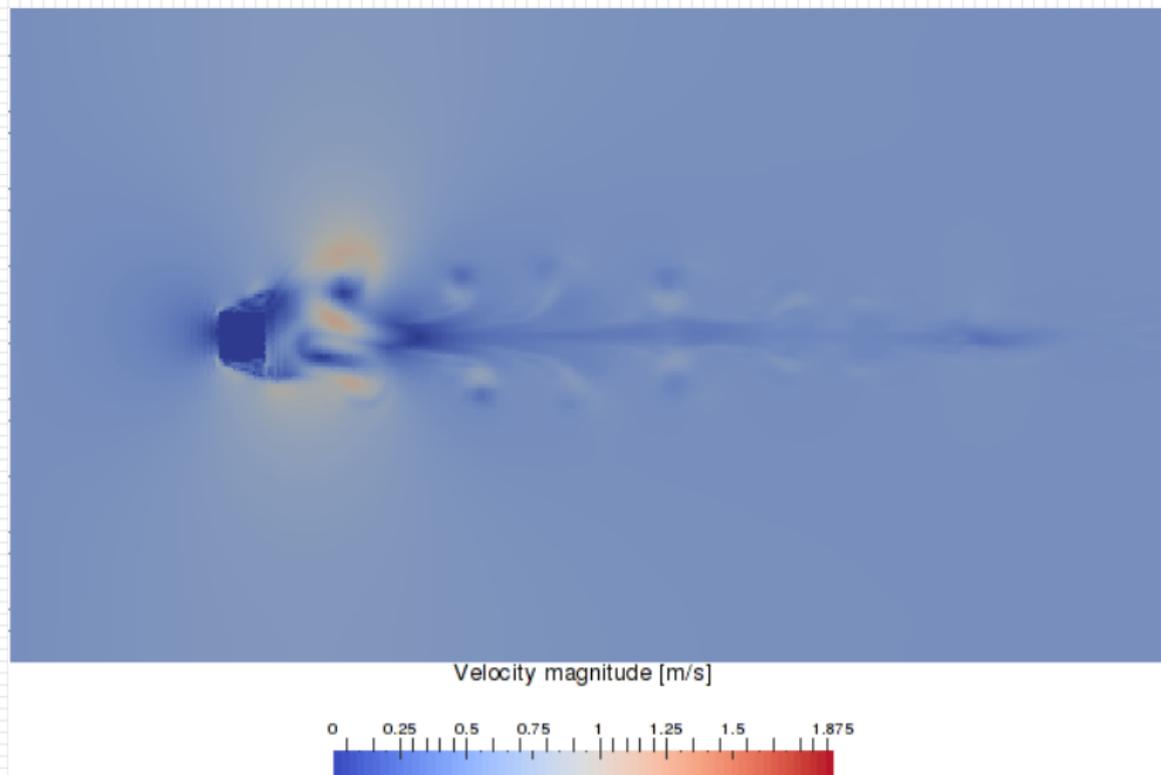
# libmpdata++ 2.0: immersed boundary teaser



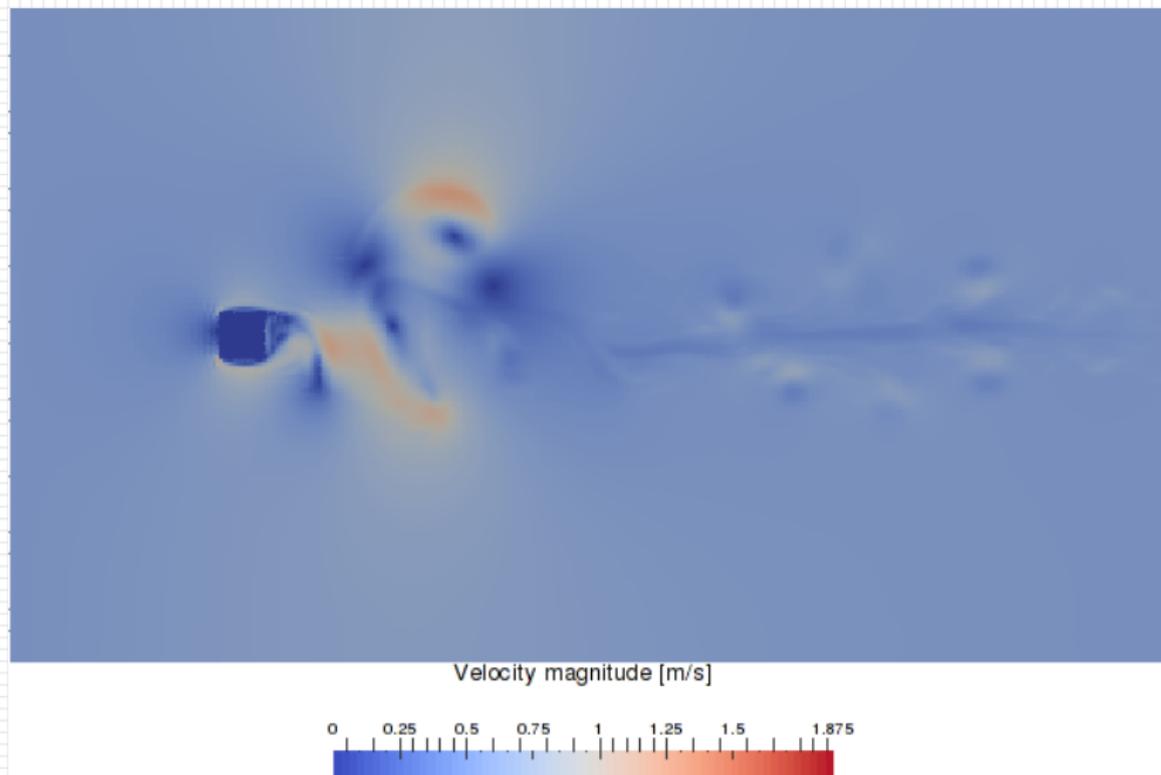
# libmpdata++ 2.0: immersed boundary teaser



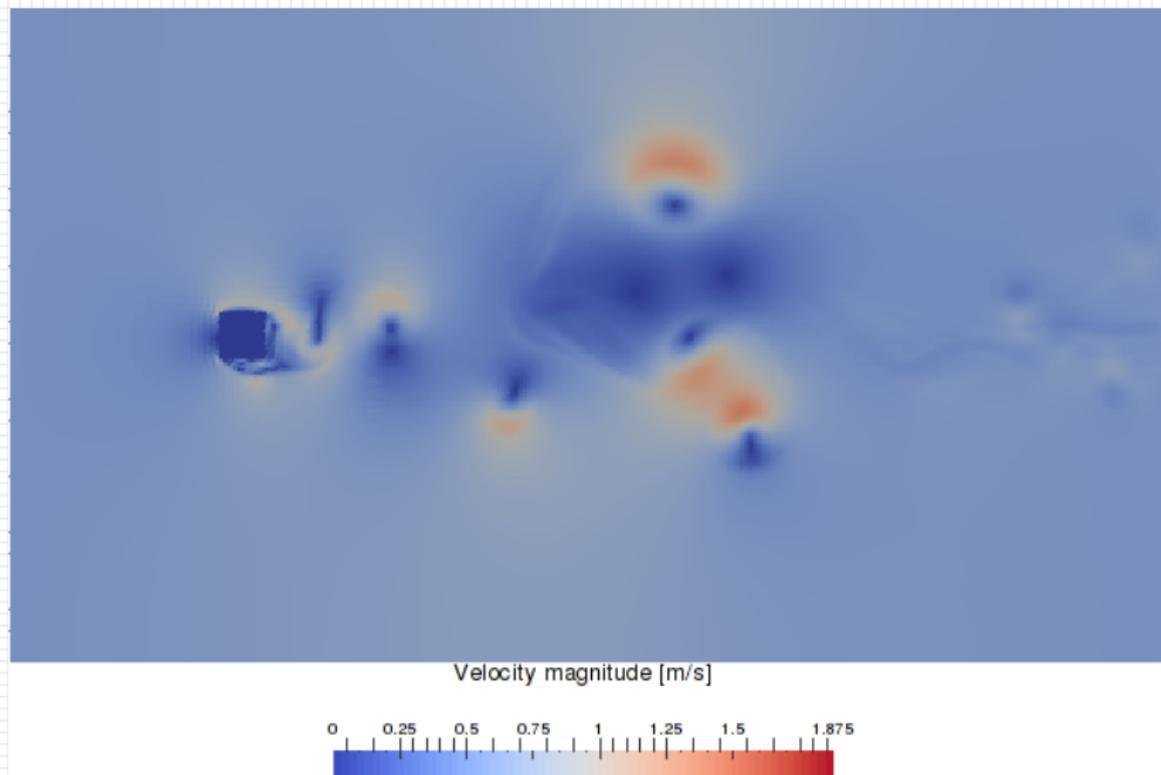
# libmpdata++ 2.0: immersed boundary teaser



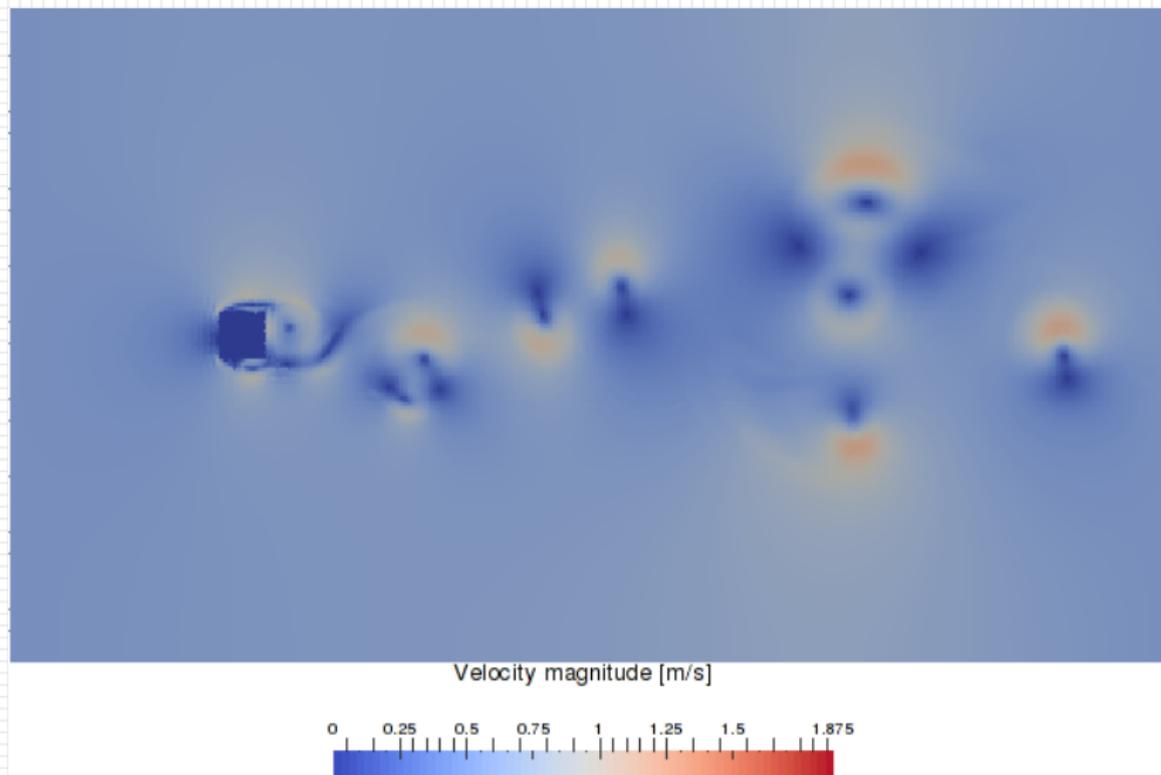
# libmpdata++ 2.0: immersed boundary teaser



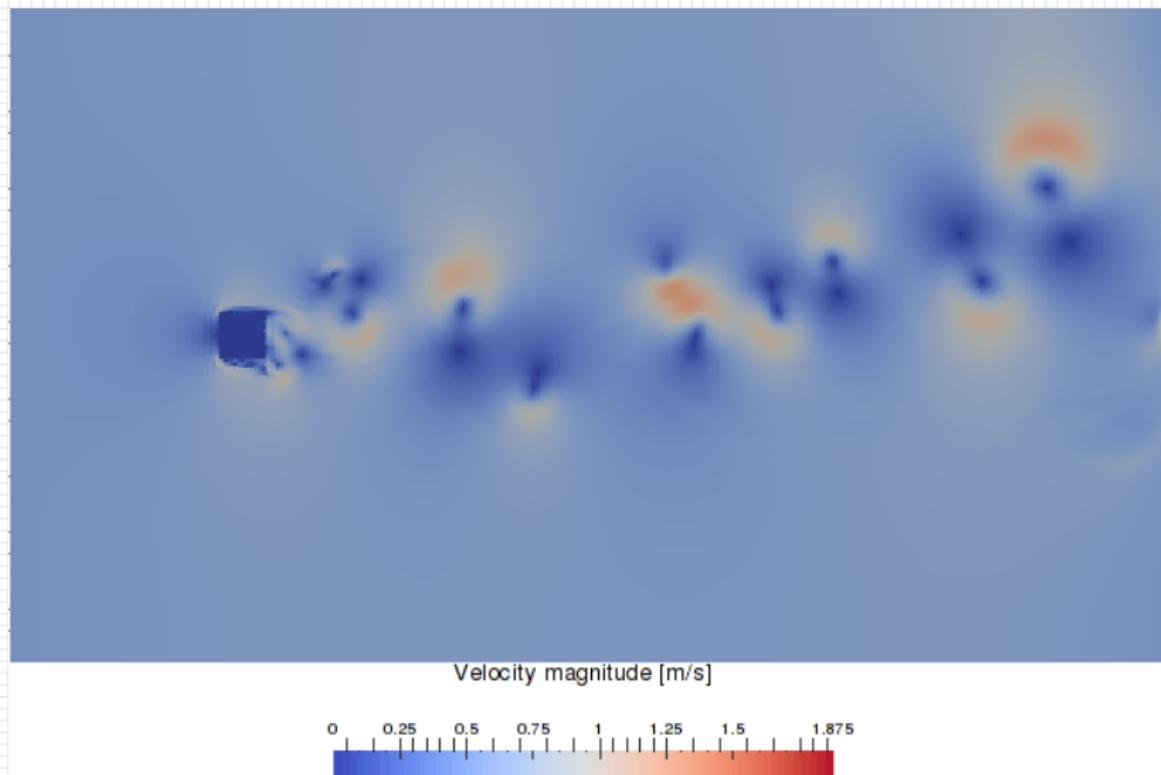
# libmpdata++ 2.0: immersed boundary teaser



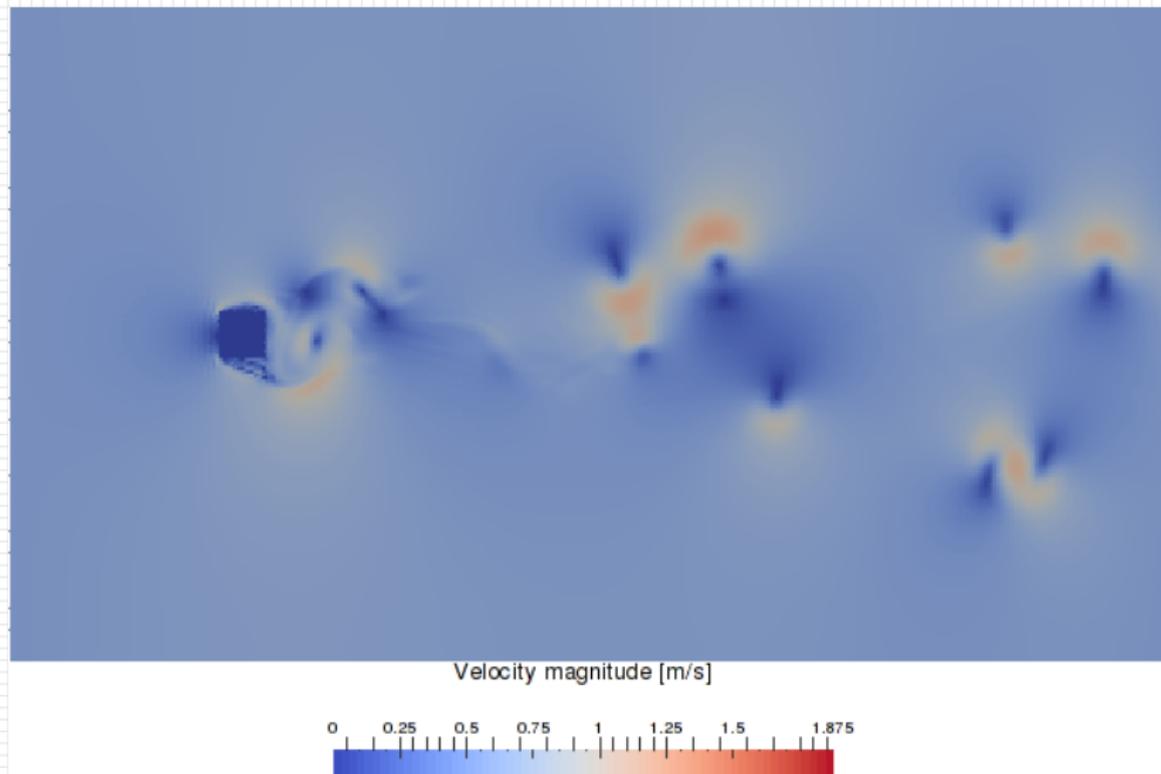
# libmpdata++ 2.0: immersed boundary teaser



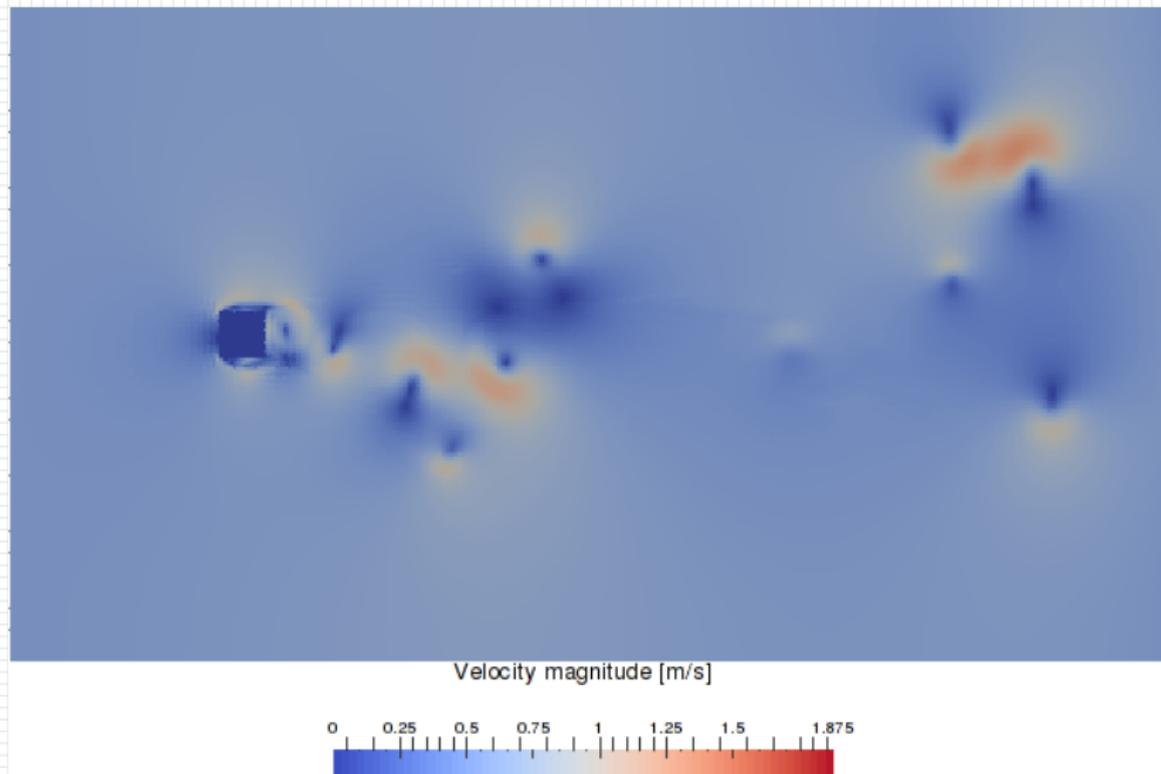
# libmpdata++ 2.0: immersed boundary teaser



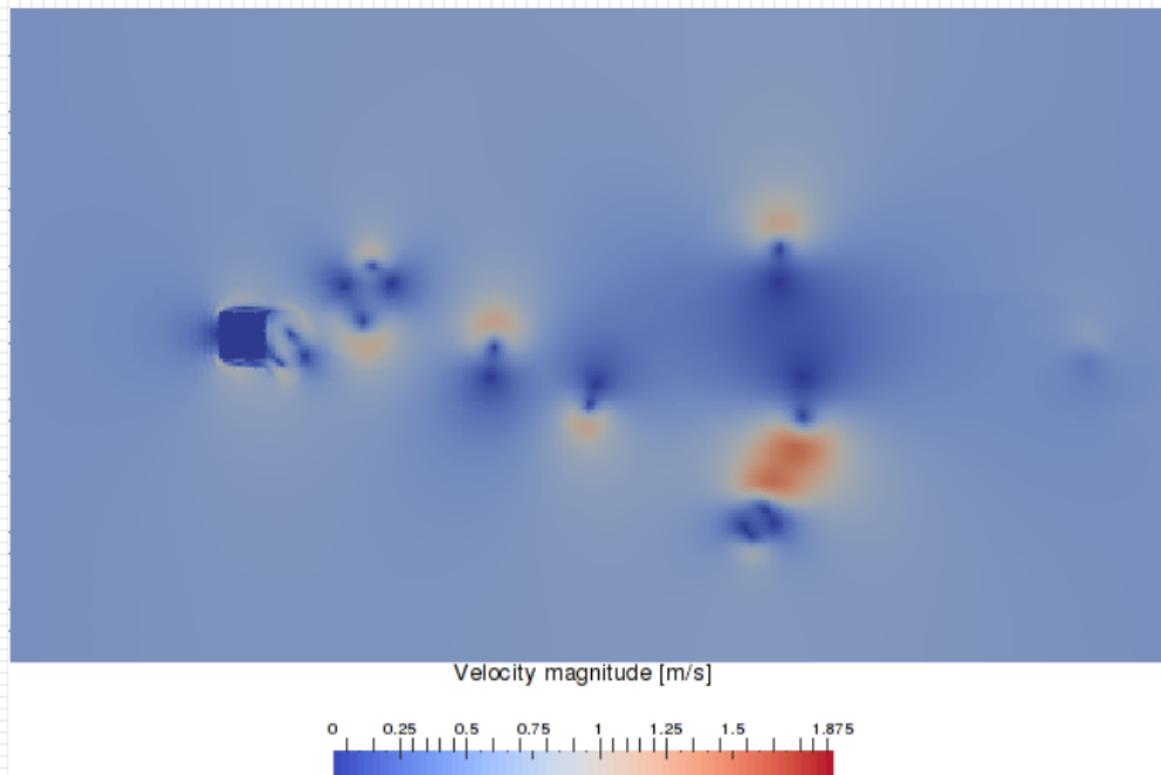
# libmpdata++ 2.0: immersed boundary teaser



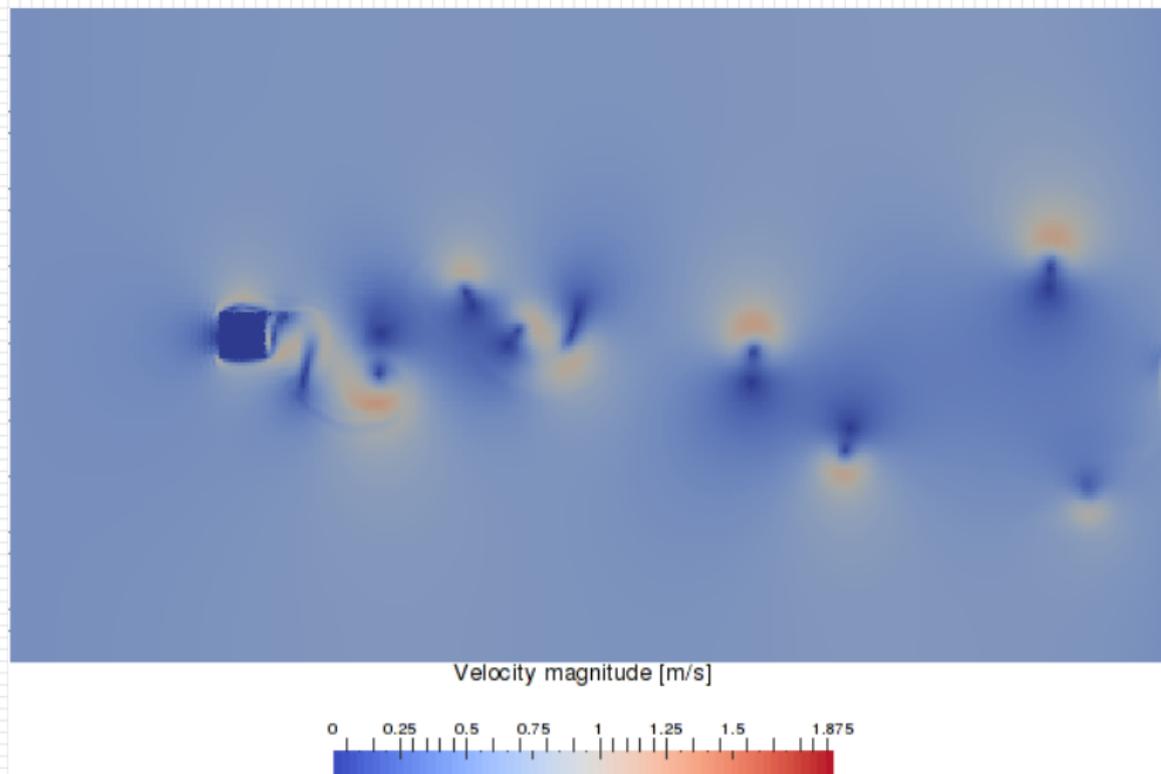
# libmpdata++ 2.0: immersed boundary teaser



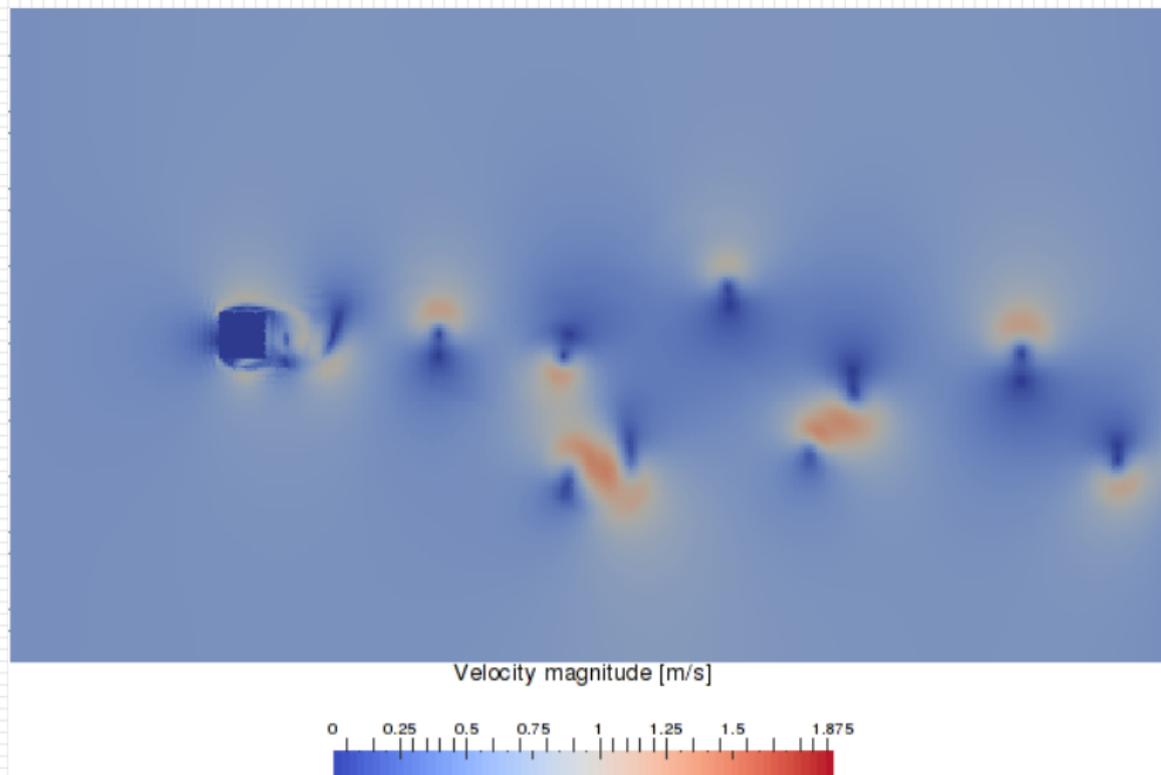
# libmpdata++ 2.0: immersed boundary teaser



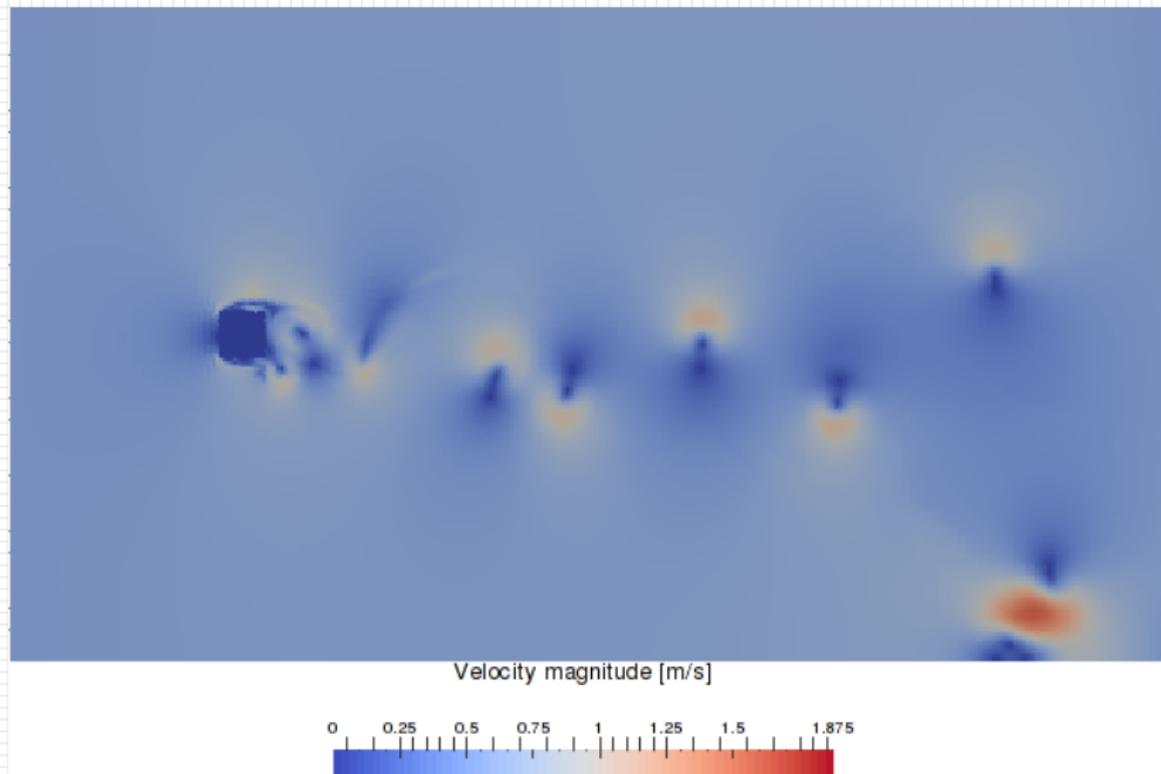
# libmpdata++ 2.0: immersed boundary teaser



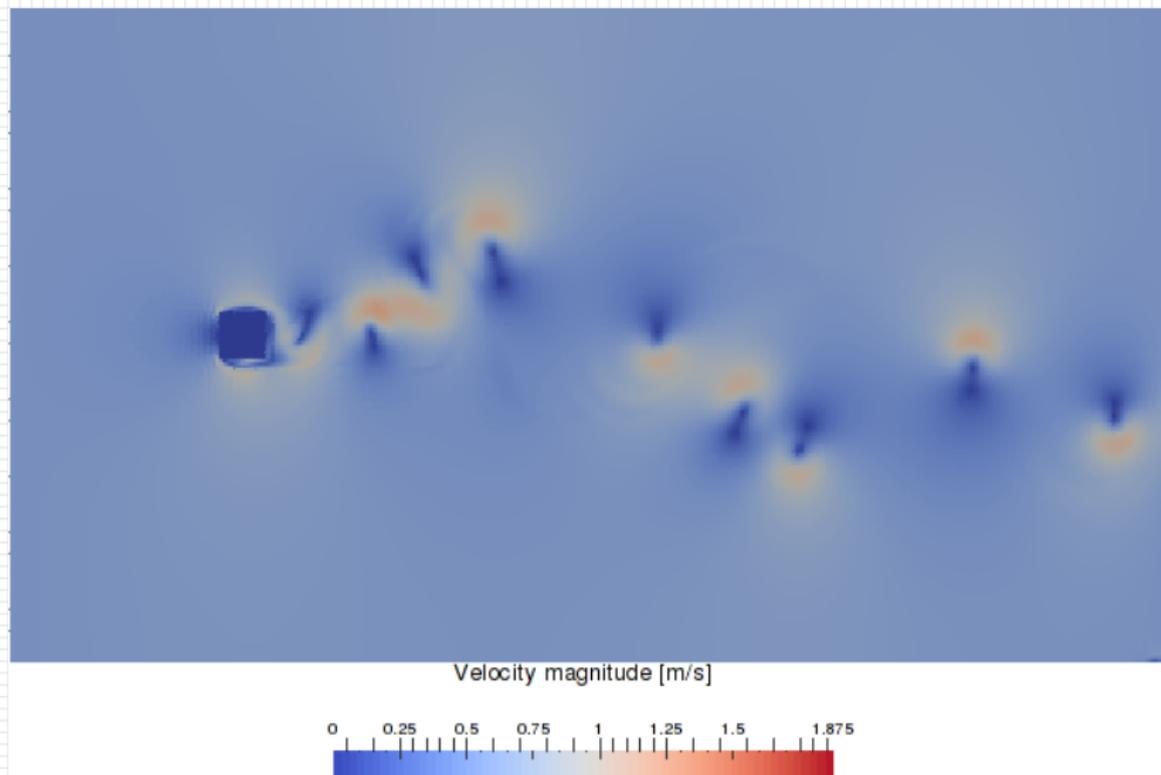
# libmpdata++ 2.0: immersed boundary teaser



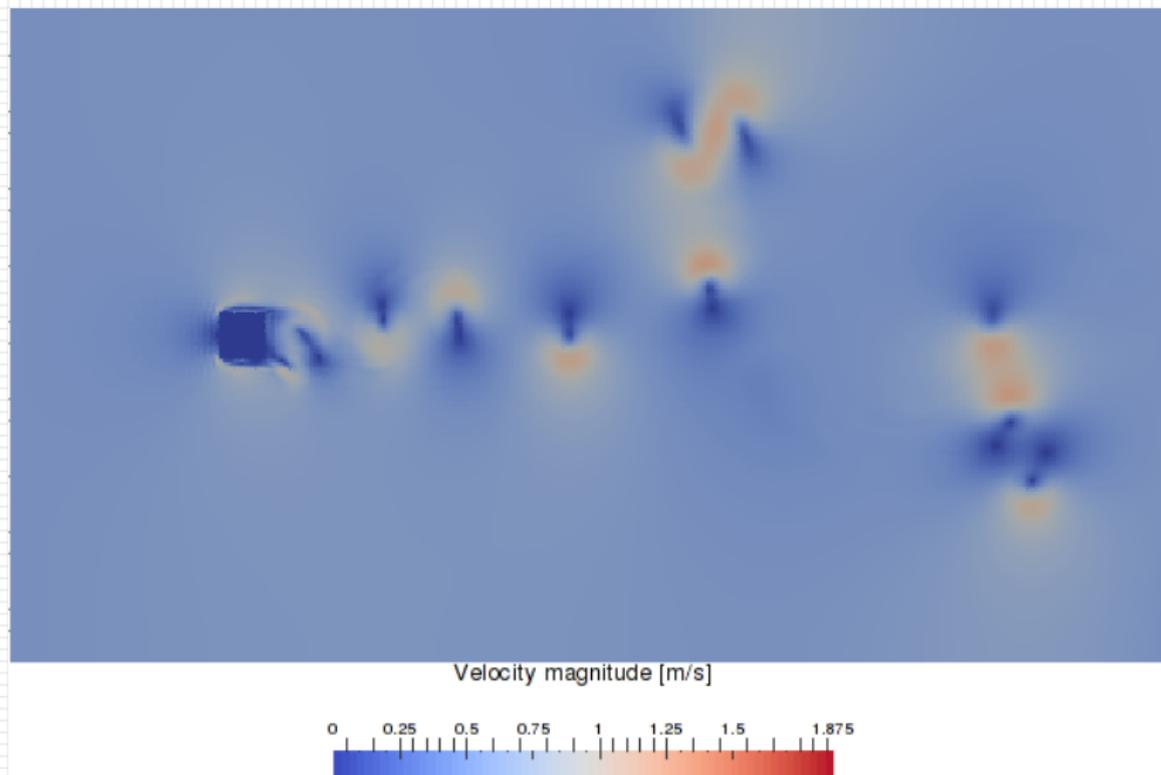
# libmpdata++ 2.0: immersed boundary teaser



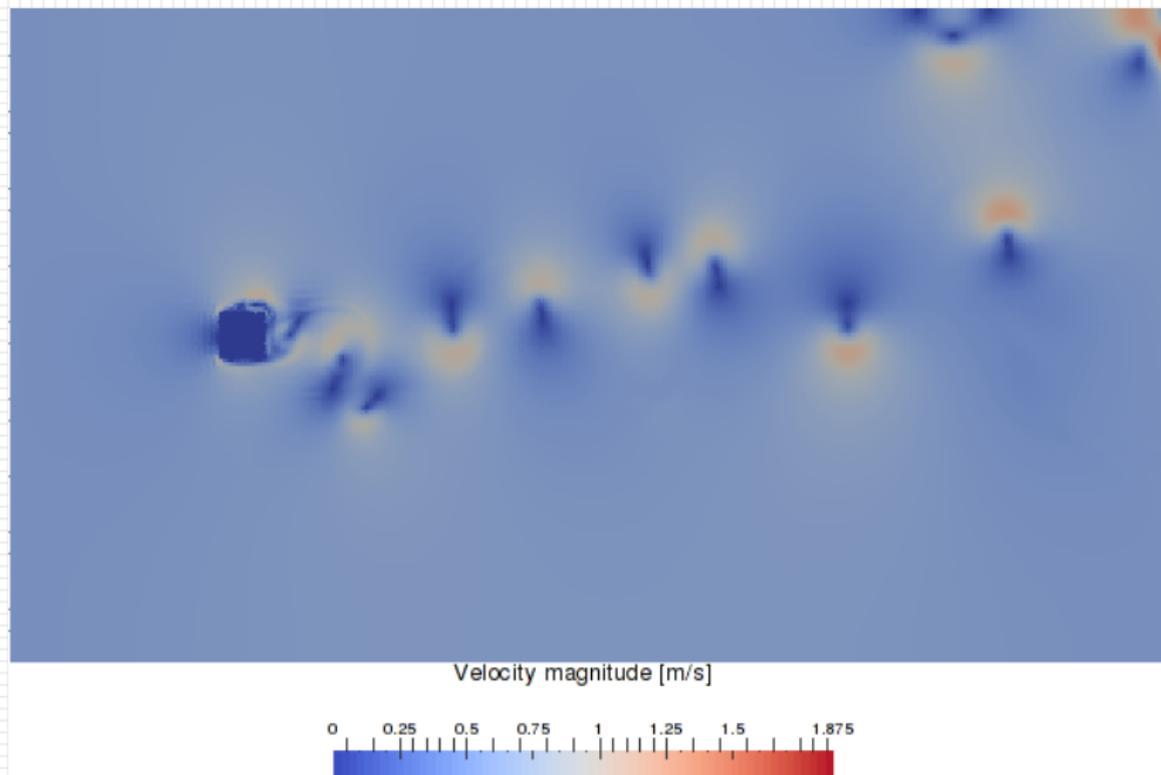
# libmpdata++ 2.0: immersed boundary teaser



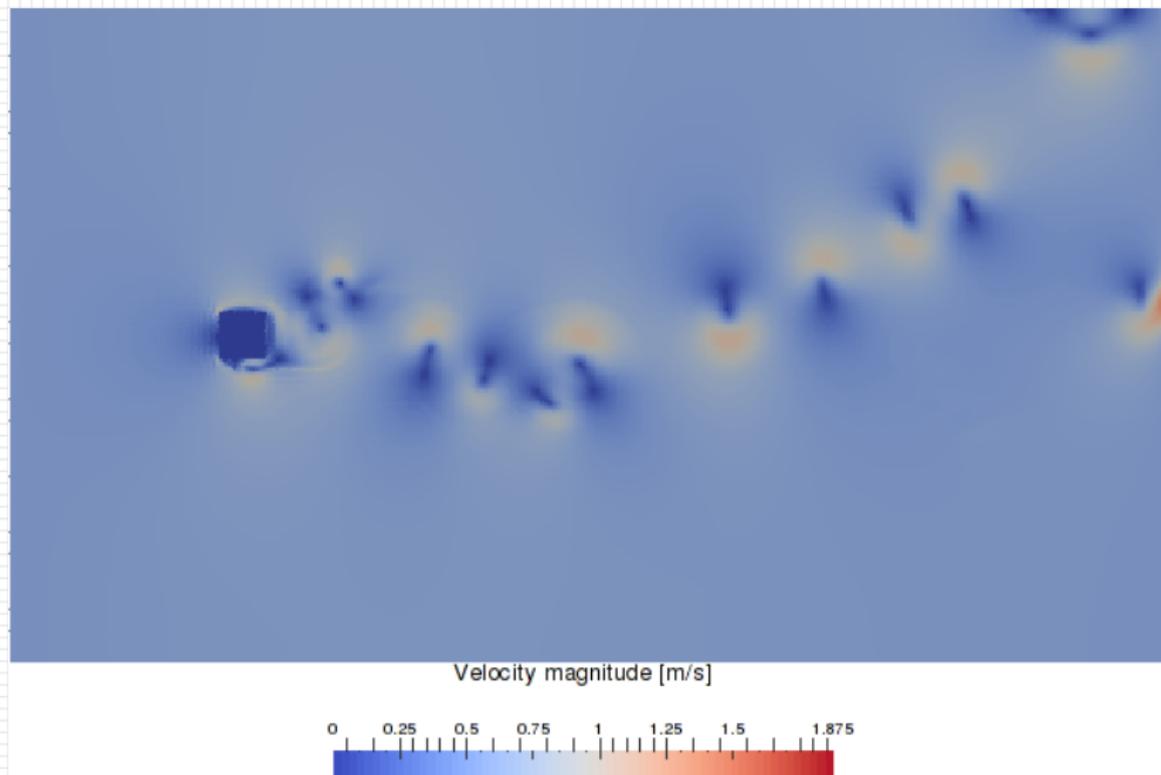
# libmpdata++ 2.0: immersed boundary teaser



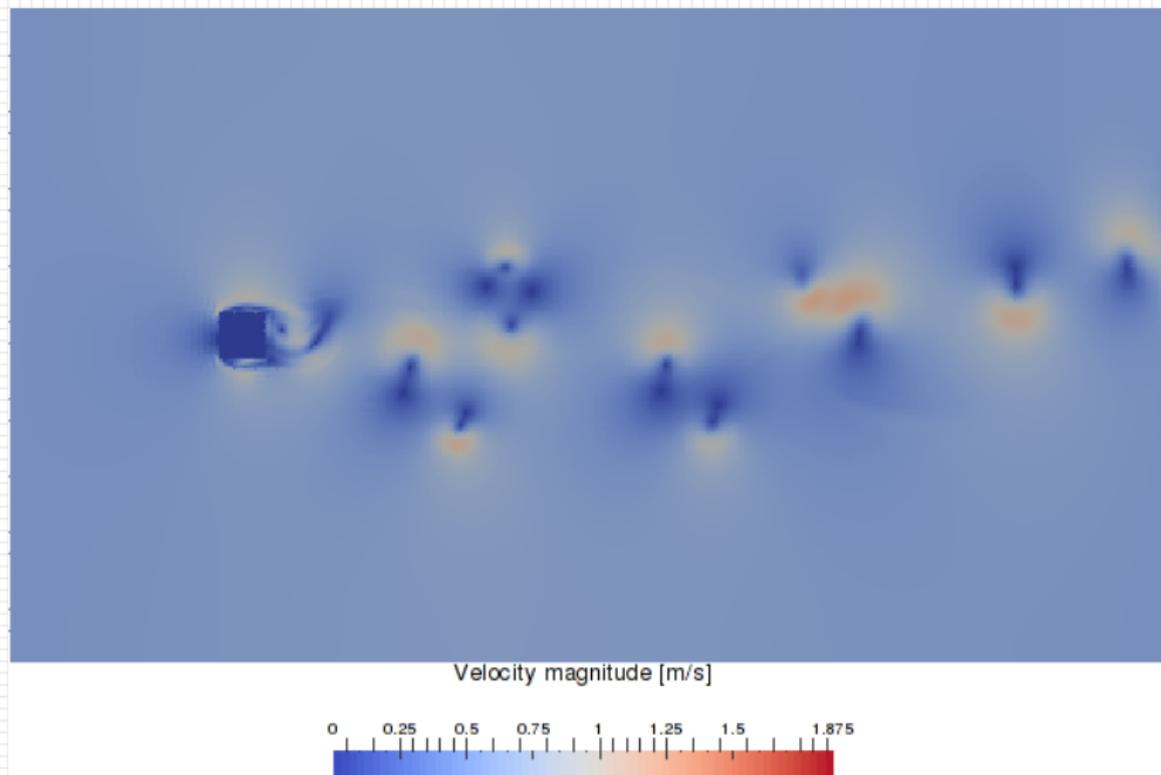
# libmpdata++ 2.0: immersed boundary teaser



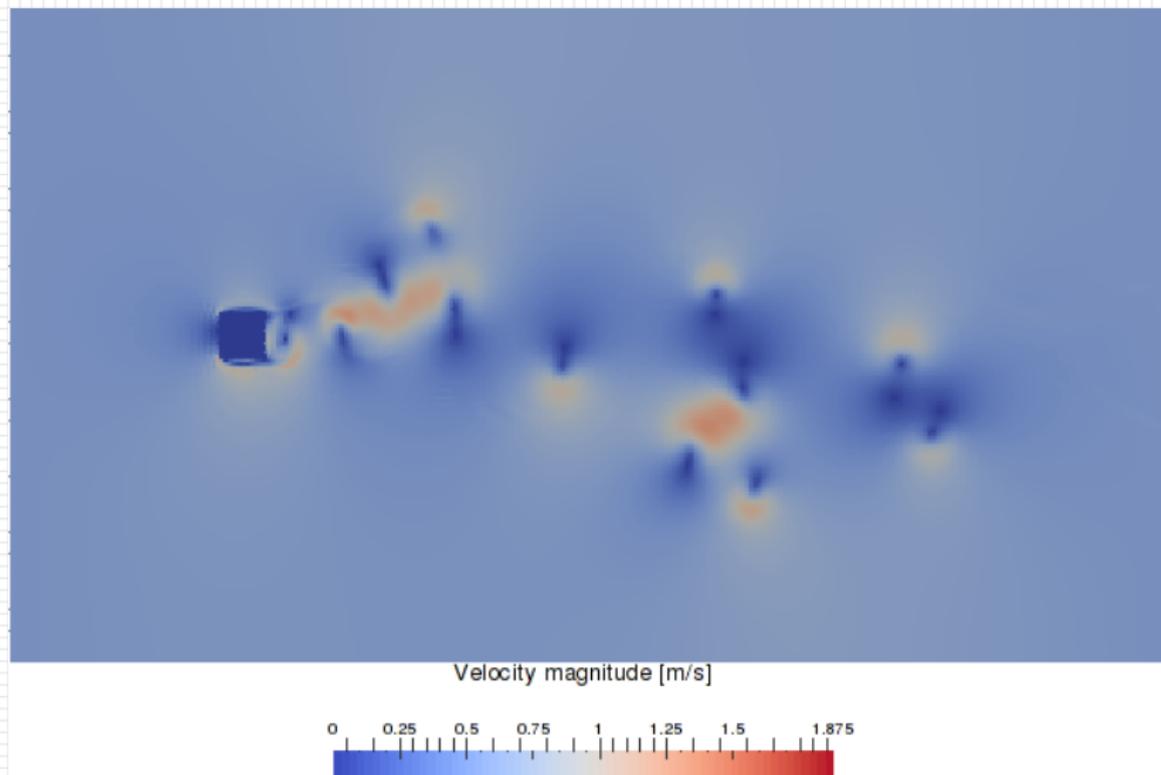
# libmpdata++ 2.0: immersed boundary teaser



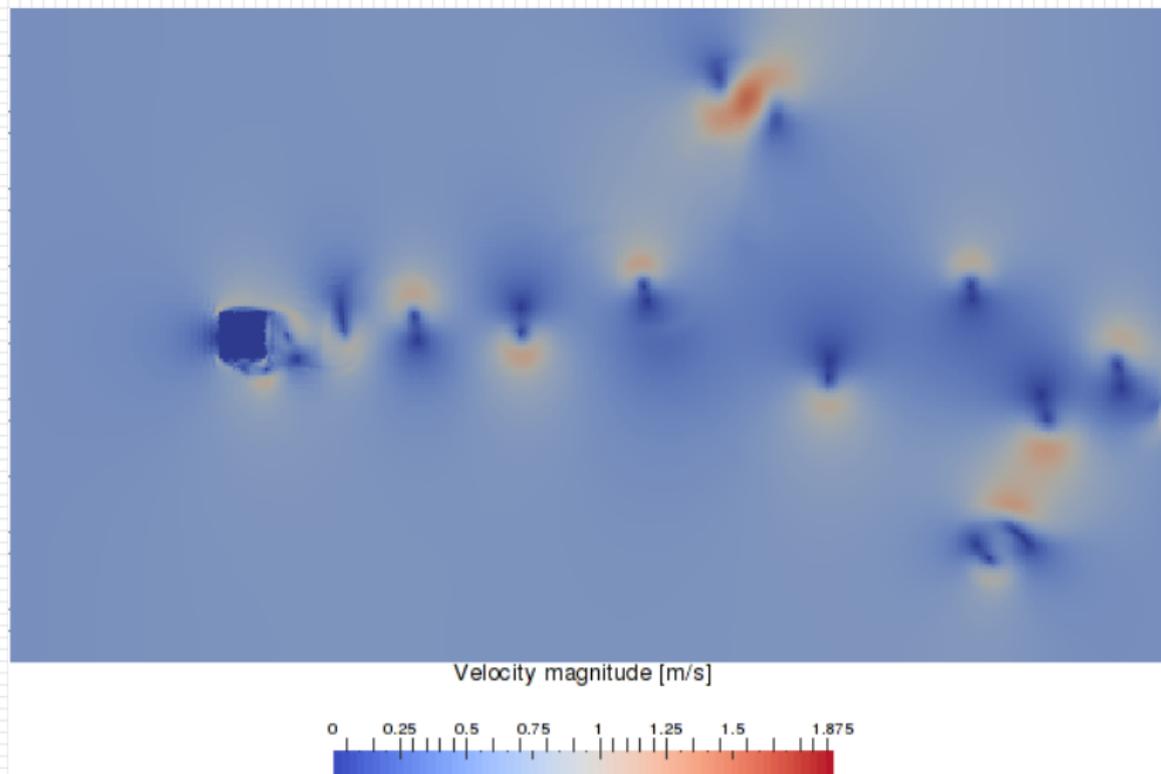
# libmpdata++ 2.0: immersed boundary teaser



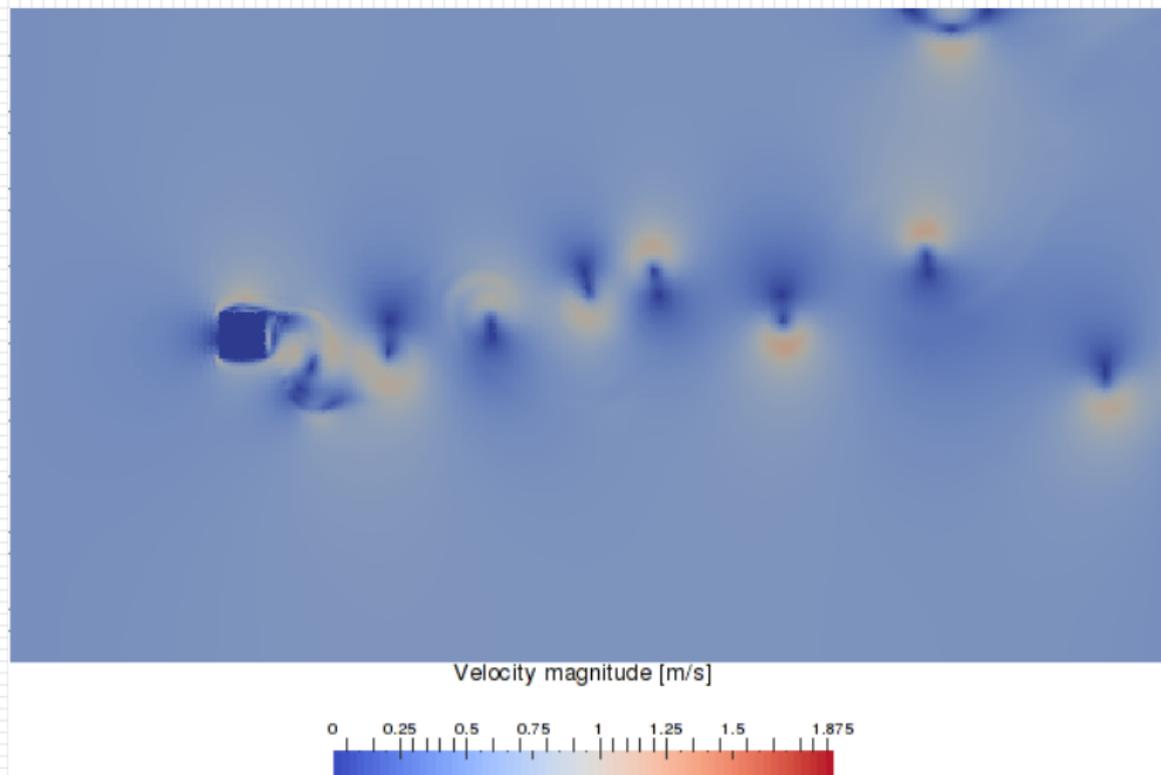
# libmpdata++ 2.0: immersed boundary teaser



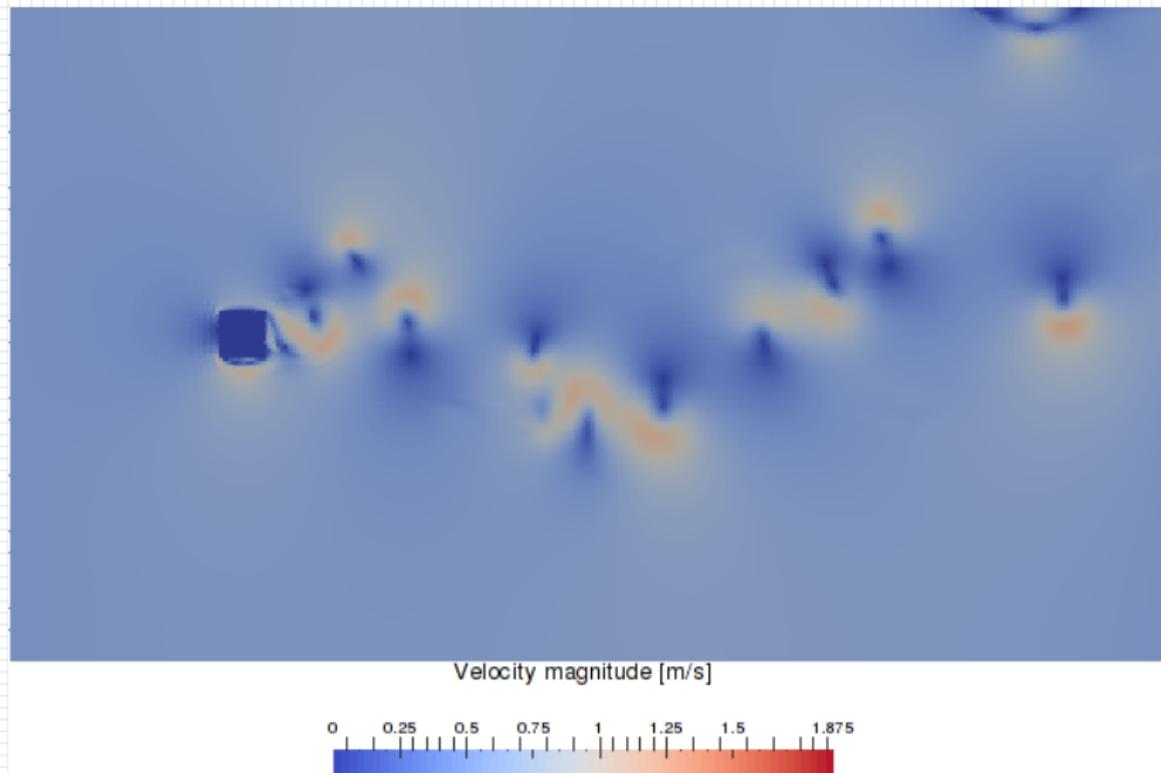
# libmpdata++ 2.0: immersed boundary teaser



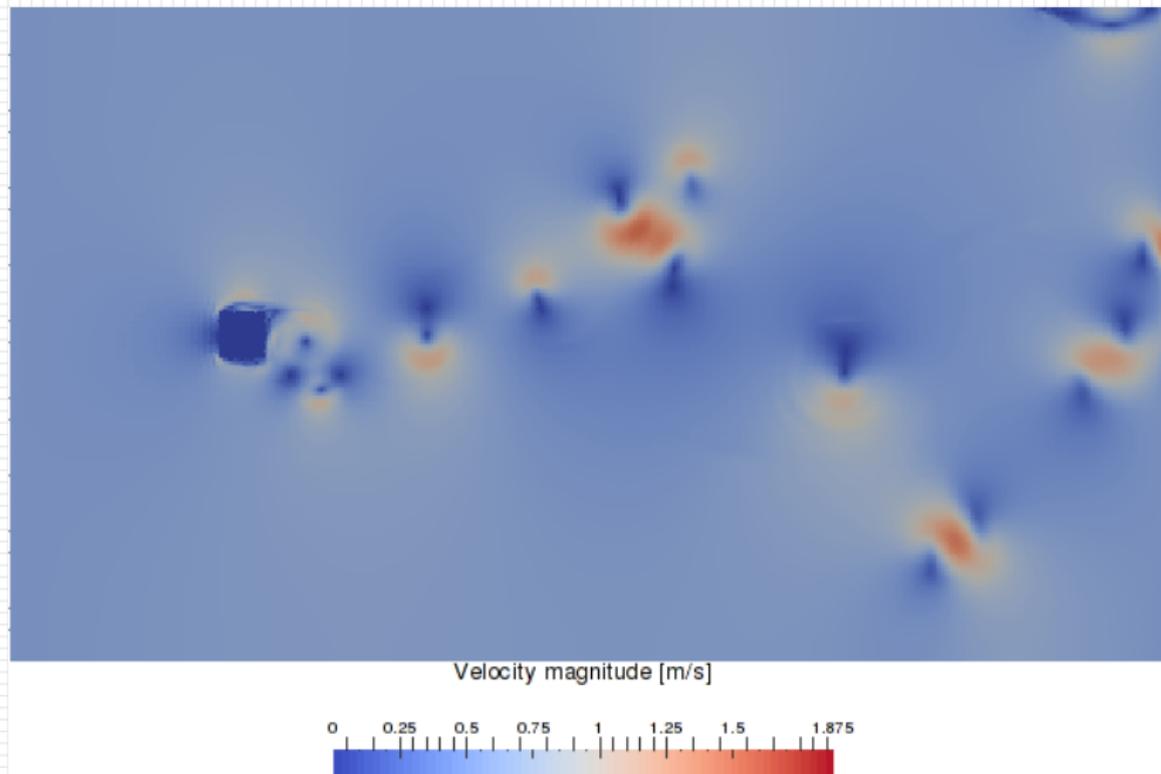
# libmpdata++ 2.0: immersed boundary teaser



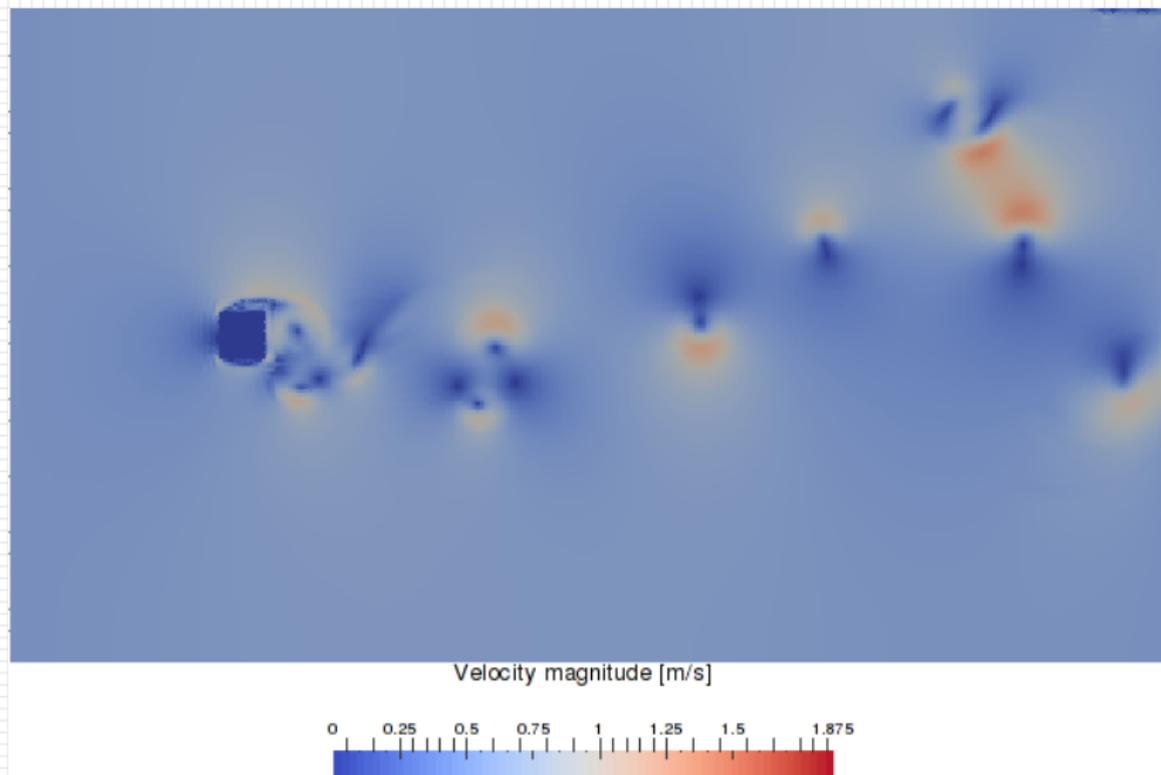
# libmpdata++ 2.0: immersed boundary teaser



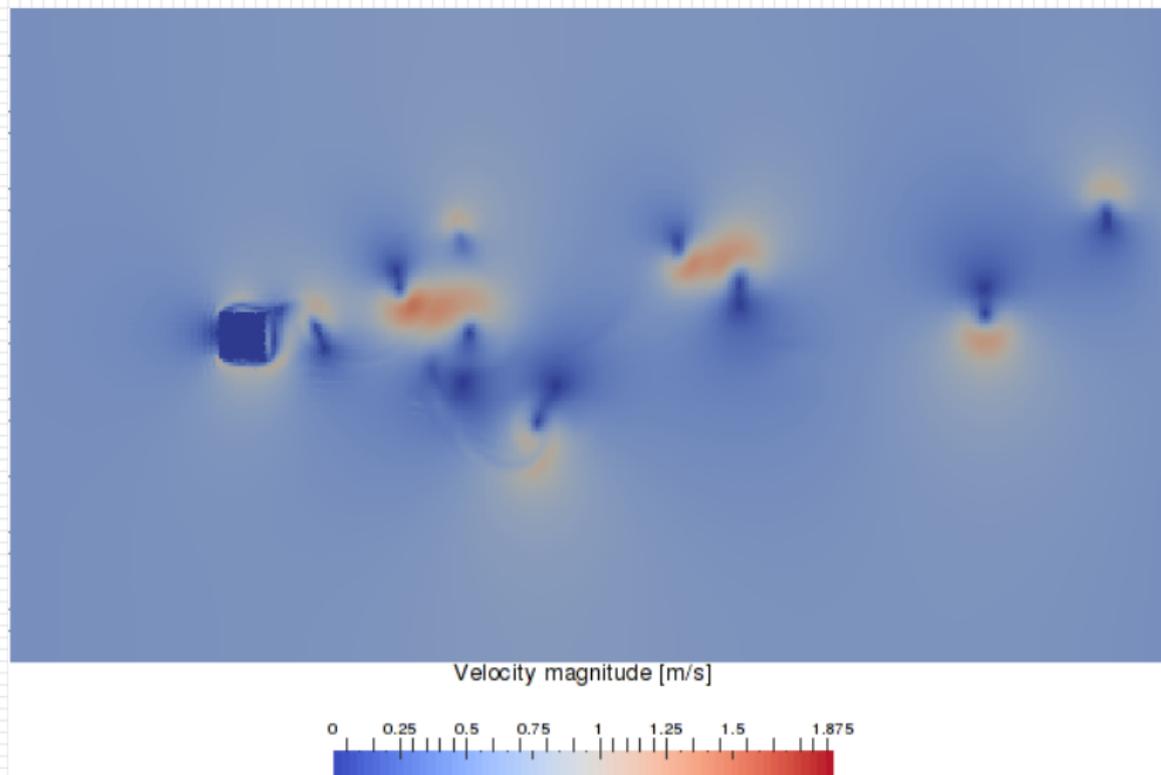
# libmpdata++ 2.0: immersed boundary teaser



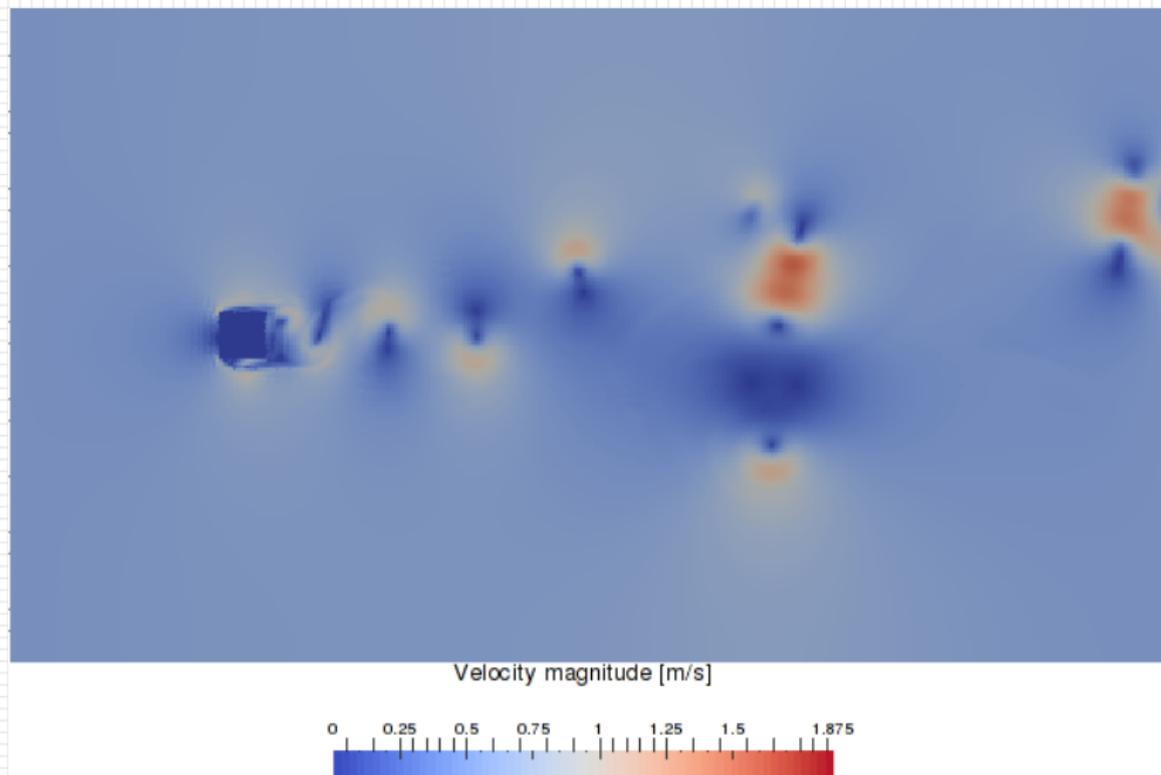
# libmpdata++ 2.0: immersed boundary teaser



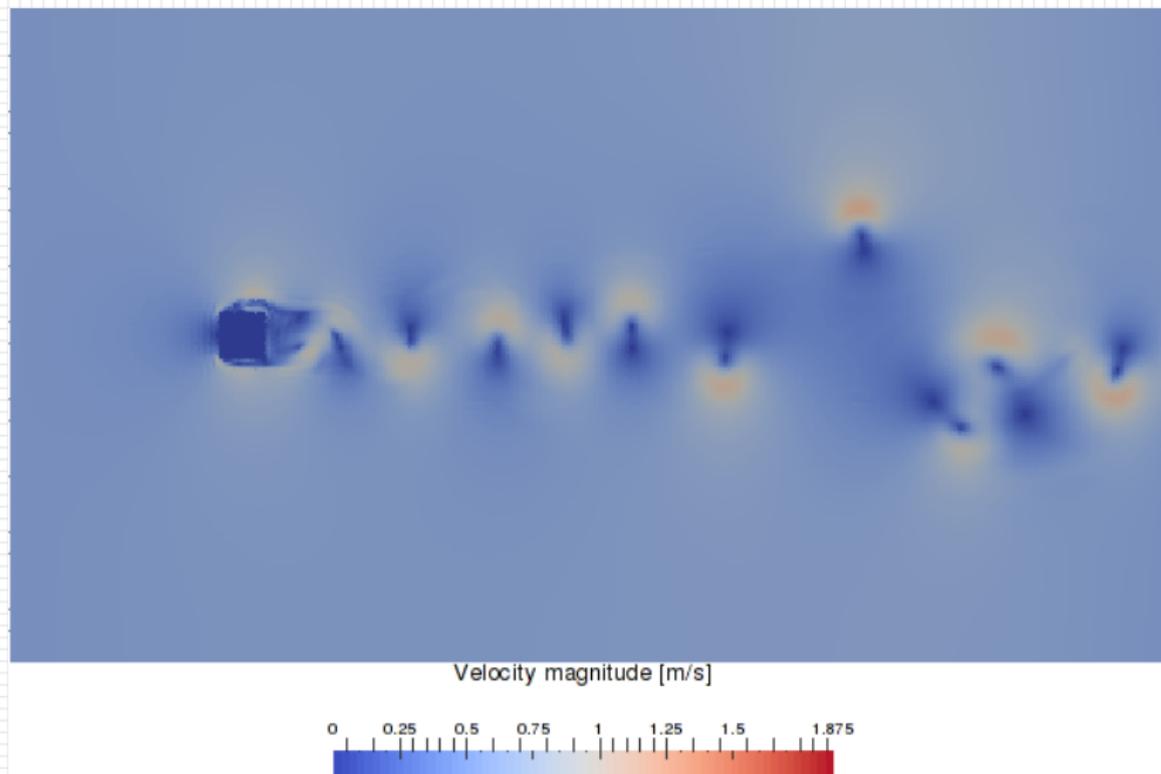
# libmpdata++ 2.0: immersed boundary teaser



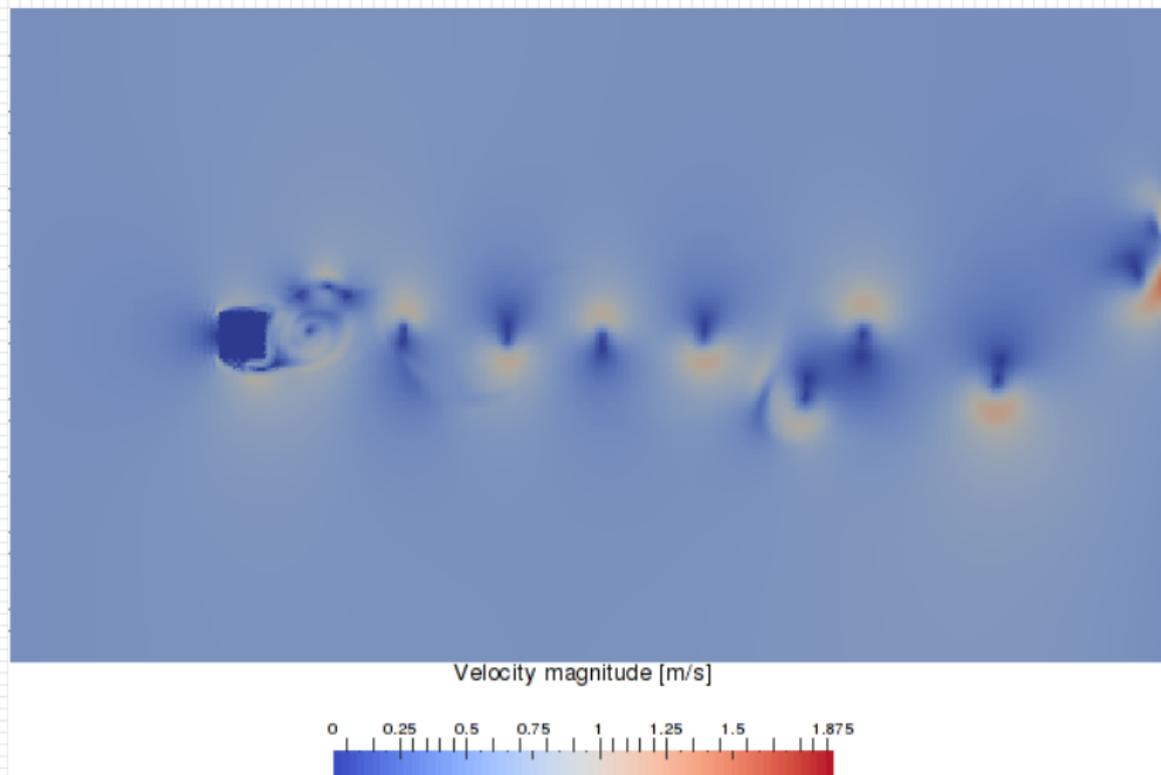
# libmpdata++ 2.0: immersed boundary teaser



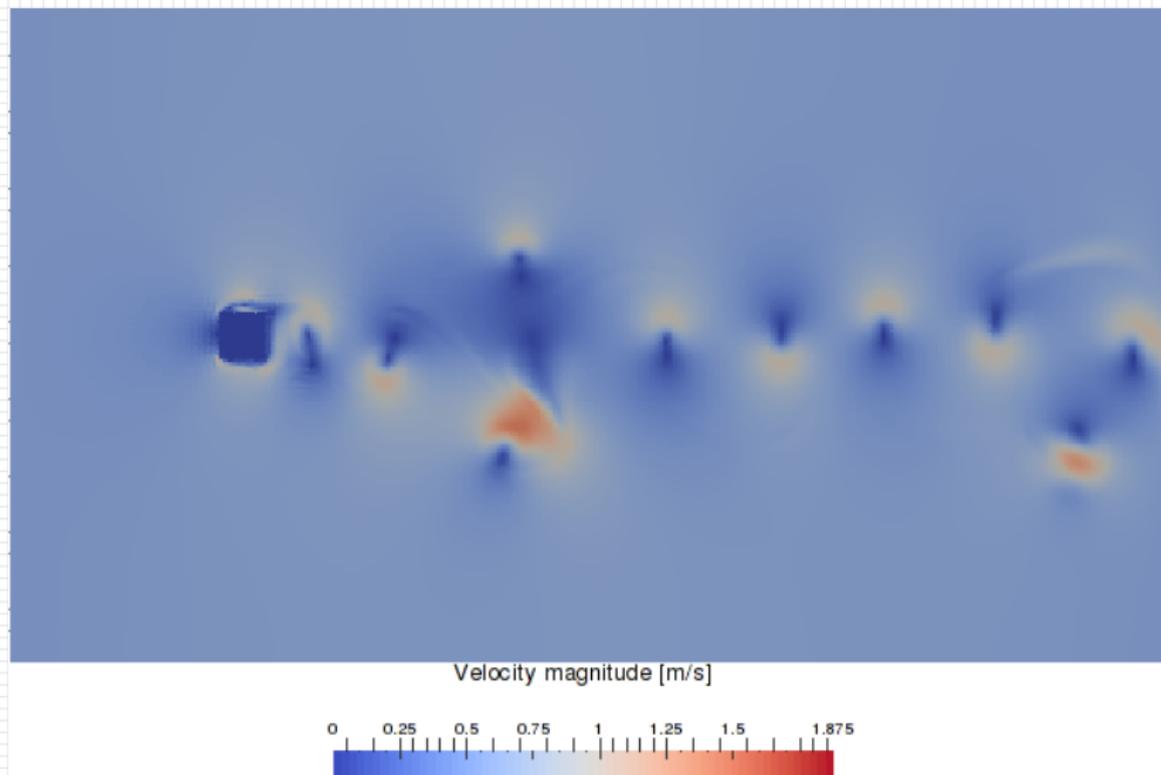
# libmpdata++ 2.0: immersed boundary teaser



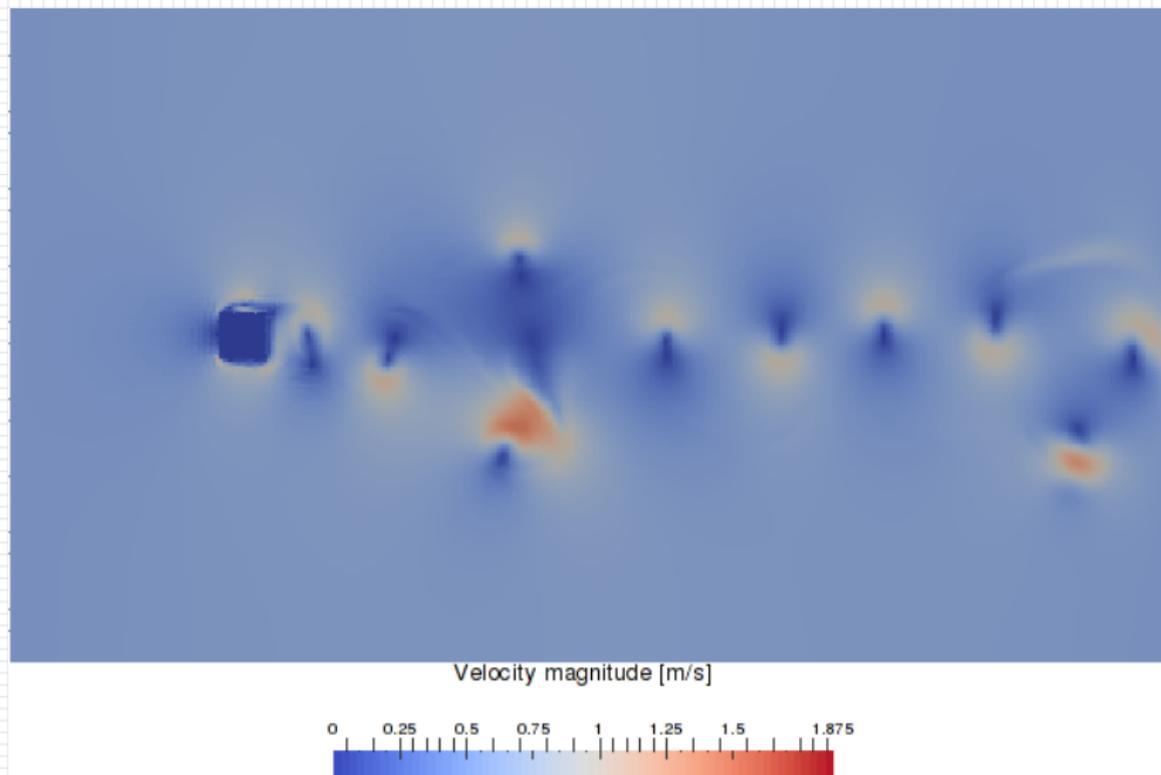
# libmpdata++ 2.0: immersed boundary teaser



# libmpdata++ 2.0: immersed boundary teaser



# libmpdata++ 2.0: immersed boundary teaser



# Plan of the talk

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

# Plan of the talk

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

# libmpdata++: some design choices

## legal

- public domain
- <https://github.com/mtetra/libmpdata++>

## library components

- `libmpdata++`
- `libmpdata++-api`
- `libmpdata++-api-headers`
- `libmpdata++-api-headers-headers`
- `libmpdata++-api-headers-headers-headers`
- `libmpdata++-api-headers-headers-headers-headers`

## architectural components

- `libmpdata++-api-headers-headers-headers-headers-headers`
- `libmpdata++-api-headers-headers-headers-headers-headers-headers`
- `libmpdata++-api-headers-headers-headers-headers-headers-headers-headers`

## dependencies

- `libmpdata++`
- `libmpdata++-api`
- `libmpdata++-api-headers`
- `libmpdata++-api-headers-headers`
- `libmpdata++-api-headers-headers-headers`
- `libmpdata++-api-headers-headers-headers-headers`

## API

- header-only library
- header-based component headers
- header-based component headers headers
- header-based component headers headers headers

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://github.com/igfuw/)

## library components

- solvers/algorithms:
  - Cvxopt
- boundary conditions:
  - Dirichlet
  - Neumann
- output handlers:
  - HDF5/ADAM (MPI-IO)
  - graphs
- shared-mem concurrency:
  - OpenMP
  - Boost.Thread
  - C++11 threads
- distributed-mem concurr.:
  - MPI

## dependencies

- C++11
- Boost
- Eigen
- GMP
- Intel MKL
- MPI
- HDF5

## API

- header-only library
- template-based component mixing
- interfaces over component boundaries
- user exposed to C++ API

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://github.com/igfuw/)

## library components

- solvers/algorithms:
  - Cvxopt
- boundary conditions:
  - Dirichlet
  - Neumann
- output handlers:
  - HDF5/ADAMS/MPFIO
  - graphs
- shared-mem concurrency:
  - OpenMP
  - Boost.Thread
  - C++11 threads
- distributed-mem concurr.:
  - MPI

## dependencies

- C++11
- Boost
- Eigen
- CMake/CXZ
- MPI
- HDF5

## API

- header-only library
- template-based component interfaces
- interfaces based on constant expressions
- user exposed to C++11 API

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://github.com/igfuw/)

## library components

- solvers/algorithms:
  - Cvxopt
  - IPOPT
- boundary conditions:
  - Dirichlet
  - Neumann
- output handlers:
  - VTK5/VTK4, VTK4-IO
  - gnuplot
- shared-mem concurrency:
  - OpenMP
  - Boost.Thread
  - C++11 threads
- distributed-mem concurr.: MPI

## dependencies

- C++11
- Boost
- Boost.Math
- Boost.Serialization
- Boost.Thread
- CMake, CXX
- MPI
- VTK5

## API

- header-only library
- template-based component interfaces
- interfaces based component registration
- user exposed to C++-like API

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://github.com/igfuw/)

## library components

- solvers/algorithms:
  - `lsqr`
  - `lsqr`
  - `lsqr`
- boundary conditions:
  - `bc`
- output handlers:
  - `CSV/JSON/HTML`
  - `graph`
- shared-mem concurrency:
  - `OpenMP`
  - `Boost.Thread`
  - `OpenCL` (work in progress)
- distributed-mem concurr.:
  - `mpi`

## dependencies

- `Eigen3`
- `Boost`
- `Boost`
- `Boost`
- `OpenCL` (optional)
- `mpi` (optional)
- `OpenMP` (optional)

## API

- header-only library
- templated-based component interfaces
- interface-based component registration
- user extended to `libmpdata++` API

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://github.com/igfuw/)

## library components

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

## dependencies

- Eigen3
- Boost
- MPI
- MPIIO
- OpenMP
- gnuplot
- HDF5

## API

- header-only library
- template-based component interfaces
- interfaces to other component libraries
- user exposed to MPI++ API

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://github.com/igfuw/)

## library components

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

## dependencies

- Eigen
- Boost
- MPI
- MPIIO
- OpenMP
- gnuplot
- HDF5

## API

- header-only library
- template-based component interfaces
- inheritance-based component structure
- user exposed to C++ API

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://github.com/igfuw/)

## library components

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

## dependencies

- Eigen
- Boost
- MPI
- MPIIO
- OpenMP
- gnuplot
- MPI
- HDF5

## API

- header-only library
- template-based component interfaces
- interfaces to other component libraries
- user exposed to REX++ API

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://github.com/igfuw/)

## library components

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

## dependencies

- Eigen
- Boost
- Boost.Fiber
- MPI
- XDMF

## API

- header-only library
- templated-based component interfaces
- inheritance-based component abstraction
- user-exposed to HLL++ API

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://github.com/igfuw/)

## library components

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

## dependencies

- Eigen
- Boost
- MPI
- MPIIO
- HDF5

## API

- header-only library
- templated-based component interfaces
- inheritance-based component relationships
- user-exposed to C++ API

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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

```
libmpdata++ depends on:
- Boost
- Eigen
- GMP
- HDF5
- MPI
- OpenMP
- XDMF
- gnuplot
- libtiff
- libz
- libjpeg
- libpng
- libcurl
- libssh2
- libssl
- libcrypto
- liblzma
- libb2
- liblz4
- libzstd
- liblz4
- libzstd
```

## API

```
libmpdata++ is a:
- header-only library
- templated-based component
- interface-based component
- user-exposed to C++ API
```

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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

```
libmpdata++ depends on:
- Boost
- Eigen
- GMP
- HDF5
- MPI
- OpenMP
- gnuplot
- libxdmf
- libxdmf2
- libxdmf3
- libxdmf4
- libxdmf5
- libxdmf6
- libxdmf7
- libxdmf8
- libxdmf9
- libxdmf10
- libxdmf11
- libxdmf12
- libxdmf13
- libxdmf14
- libxdmf15
- libxdmf16
- libxdmf17
- libxdmf18
- libxdmf19
- libxdmf20
- libxdmf21
- libxdmf22
- libxdmf23
- libxdmf24
- libxdmf25
- libxdmf26
- libxdmf27
- libxdmf28
- libxdmf29
- libxdmf30
- libxdmf31
- libxdmf32
- libxdmf33
- libxdmf34
- libxdmf35
- libxdmf36
- libxdmf37
- libxdmf38
- libxdmf39
- libxdmf40
- libxdmf41
- libxdmf42
- libxdmf43
- libxdmf44
- libxdmf45
- libxdmf46
- libxdmf47
- libxdmf48
- libxdmf49
- libxdmf50
- libxdmf51
- libxdmf52
- libxdmf53
- libxdmf54
- libxdmf55
- libxdmf56
- libxdmf57
- libxdmf58
- libxdmf59
- libxdmf60
- libxdmf61
- libxdmf62
- libxdmf63
- libxdmf64
- libxdmf65
- libxdmf66
- libxdmf67
- libxdmf68
- libxdmf69
- libxdmf70
- libxdmf71
- libxdmf72
- libxdmf73
- libxdmf74
- libxdmf75
- libxdmf76
- libxdmf77
- libxdmf78
- libxdmf79
- libxdmf80
- libxdmf81
- libxdmf82
- libxdmf83
- libxdmf84
- libxdmf85
- libxdmf86
- libxdmf87
- libxdmf88
- libxdmf89
- libxdmf90
- libxdmf91
- libxdmf92
- libxdmf93
- libxdmf94
- libxdmf95
- libxdmf96
- libxdmf97
- libxdmf98
- libxdmf99
- libxdmf100
```

## API

```
libmpdata++ is a:
- header-only library
- multiplatform, compiled, multi-arch
- independent from component dependencies
- user-exposed to MPI++ API
```

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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

```
libmpdata++ depends on:
- boost
- boost::asio
- boost::filesystem
- boost::program_options
- boost::thread
- boost::unit_test
- boost::unit_test_framework
- boost::unit_test_test_suite
- boost::unit_test_test_suite2
- boost::unit_test_test_suite3
- boost::unit_test_test_suite4
- boost::unit_test_test_suite5
- boost::unit_test_test_suite6
- boost::unit_test_test_suite7
- boost::unit_test_test_suite8
- boost::unit_test_test_suite9
- boost::unit_test_test_suite10
- boost::unit_test_test_suite11
- boost::unit_test_test_suite12
- boost::unit_test_test_suite13
- boost::unit_test_test_suite14
- boost::unit_test_test_suite15
- boost::unit_test_test_suite16
- boost::unit_test_test_suite17
- boost::unit_test_test_suite18
- boost::unit_test_test_suite19
- boost::unit_test_test_suite20
- boost::unit_test_test_suite21
- boost::unit_test_test_suite22
- boost::unit_test_test_suite23
- boost::unit_test_test_suite24
- boost::unit_test_test_suite25
- boost::unit_test_test_suite26
- boost::unit_test_test_suite27
- boost::unit_test_test_suite28
- boost::unit_test_test_suite29
- boost::unit_test_test_suite30
- boost::unit_test_test_suite31
- boost::unit_test_test_suite32
- boost::unit_test_test_suite33
- boost::unit_test_test_suite34
- boost::unit_test_test_suite35
- boost::unit_test_test_suite36
- boost::unit_test_test_suite37
- boost::unit_test_test_suite38
- boost::unit_test_test_suite39
- boost::unit_test_test_suite40
- boost::unit_test_test_suite41
- boost::unit_test_test_suite42
- boost::unit_test_test_suite43
- boost::unit_test_test_suite44
- boost::unit_test_test_suite45
- boost::unit_test_test_suite46
- boost::unit_test_test_suite47
- boost::unit_test_test_suite48
- boost::unit_test_test_suite49
- boost::unit_test_test_suite50
- boost::unit_test_test_suite51
- boost::unit_test_test_suite52
- boost::unit_test_test_suite53
- boost::unit_test_test_suite54
- boost::unit_test_test_suite55
- boost::unit_test_test_suite56
- boost::unit_test_test_suite57
- boost::unit_test_test_suite58
- boost::unit_test_test_suite59
- boost::unit_test_test_suite60
- boost::unit_test_test_suite61
- boost::unit_test_test_suite62
- boost::unit_test_test_suite63
- boost::unit_test_test_suite64
- boost::unit_test_test_suite65
- boost::unit_test_test_suite66
- boost::unit_test_test_suite67
- boost::unit_test_test_suite68
- boost::unit_test_test_suite69
- boost::unit_test_test_suite70
- boost::unit_test_test_suite71
- boost::unit_test_test_suite72
- boost::unit_test_test_suite73
- boost::unit_test_test_suite74
- boost::unit_test_test_suite75
- boost::unit_test_test_suite76
- boost::unit_test_test_suite77
- boost::unit_test_test_suite78
- boost::unit_test_test_suite79
- boost::unit_test_test_suite80
- boost::unit_test_test_suite81
- boost::unit_test_test_suite82
- boost::unit_test_test_suite83
- boost::unit_test_test_suite84
- boost::unit_test_test_suite85
- boost::unit_test_test_suite86
- boost::unit_test_test_suite87
- boost::unit_test_test_suite88
- boost::unit_test_test_suite89
- boost::unit_test_test_suite90
- boost::unit_test_test_suite91
- boost::unit_test_test_suite92
- boost::unit_test_test_suite93
- boost::unit_test_test_suite94
- boost::unit_test_test_suite95
- boost::unit_test_test_suite96
- boost::unit_test_test_suite97
- boost::unit_test_test_suite98
- boost::unit_test_test_suite99
- boost::unit_test_test_suite100
```

## API

```
libmpdata++ is a header-only library that provides:
- a Boost-like-based component interface
- a Boost-like-based component implementation
- a user-defined Boost-like API
```











# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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

```
libmpdata++ depends on boost, hdf5, gnuplot, openmpi,
```

## API

```
libmpdata++ exports boost, hdf5, gnuplot, openmpi,
```

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr.container, timer, thread, preprocessor, filesystem, format, property.tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

# libmpdata++: some design choices

## legal

- license: GPL
- repo: [github.com/igfuw/](https://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++11
- Blitz++
- Boost (ptr\_container, timer, thread, preprocessor, filesystem, format, property\_tree, MPI)
- CMake, CTest
- MPI
- HDF5

## API

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

**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**

# ご清聴ありがとうございました。

# ご清聴ありがとうございました。

- libmpdata++ paper: Jaruga et al. 2015, Geosci. Model Dev. doi:10.5194/gmd-8-1005-2015

- 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
- My visit to Kobe has been supported by University of Hyogo

# ご清聴ありがとうございました。

- libmpdata++ paper: Jaruga et al. 2015, Geosci. Model Dev. doi:10.5194/gmd-8-1005-2015
- 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
- My visit to Kōbe has been supported by University of Hyogo

# ご清聴ありがとうございました。

- libmpdata++ paper: Jaruga et al. 2015, Geosci. Model Dev. doi:10.5194/gmd-8-1005-2015
- 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
- My visit to Kōbe has been supported by University of Hyogo

# ご清聴ありがとうございました。

- libmpdata++ paper: Jaruga et al. 2015, Geosci. Model Dev. doi:10.5194/gmd-8-1005-2015
- 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
- My visit to Kōbe has been supported by University of Hyogo

# ご清聴ありがとうございました。

- libmpdata++ paper: Jaruga et al. 2015, Geosci. Model Dev. doi:10.5194/gmd-8-1005-2015
- 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
- My visit to Kōbe has been supported by University of Hyogo