Introduction to supercomputing:

numerical experimentsi
on shared and distributed memory cluster

Igor KONSHIN[1,2]

[1]Marchuk Institute of Numerical Mathematics of RAS
[2]Sechenov University

September 22–25, 2020

Plan

- Login to cluster
- User setup
- Run // tests
- Plots with timing and speedup

Instructions:

http://cluster2.inm.ras.ru/~hpc19/0_INDEX_0.html

http://dodo.inm.ras.ru/~konshin/tr/2020-Sechenov-China-exp.pdf

Login to cluster

```
ssh hpc19@dodo.inm.ras.ru
Password: *****

To user setup:                              xsetup nickname
To request the computing resources:         xqs
To list the parallel tests available:       xlist
To run parallel test (on nodes):            xrun test
To construct the plots (on h2):             xplot
To view plots:    http://cluster2.inm.ras.ru/~hpc19/nickname/*.png
To view sources:  http://cluster2.inm.ras.ru/~hpc19/sources/*.c
To see this help info:                      xinfo
hpc19@h2:~/public_html>
```

¡ Now you are at the host computer ***h2*** *in account* ***hpc19*** *!*

User setup: **xsetup**

```
hpc19@h2:~/public_html> xsetup nickname
Setup for user: nickname
hpc19@h2:~/public_html/nickname>
```

Here *nickname* is your personal identification name or surname

*¡ Now you are at **your** own personal local directory !*

Computing resources request: **xqs**

```
hpc19@h2:~/public_html/nickname> xqs
qsub: waiting for job 52193.h2.hpc to start
qsub: job 52193.h2.hpc ready

hpc19@cl1n036:~/public_html/nickname>
```

*¡ Now you are at your local directory at your **personal cluster node** !*

Linear algebra: daxpy, norm, mvm

List of parallel tests: **xlist**

```
hpc19@cl1n036:~/public_html/nickname> xlist
omp_daxpy    omp_norm    omp_mvm
mpi_daxpy    mpi_norm    mpi_mvm-bcast    mpi_mvm-isend
t1a t2a t2b t2c t2d t3b t3c t3d t4b t4c t4d t5h t5i t5j t6k t6l
```

Types of tests:

- omp_* — tests on OpenMP (shared memory)
- mpi_* — tests on MPI (distributed memory)
- t1a ... t6l — INMOST tests (distributed memory)
- t3d t4d t5h t5i t5j t6k t6l — INMOST tests with *.jpg output

List of examples:

- *_daxpy: $\alpha \cdot \vec{x} + \vec{y}$
- *_norm: $\|\vec{x}\|$
- *_mvm*: $y = A \cdot x$

*¡ *-bcast and *-isend are different implementations of the same example !*

Parallel run: **xrun**

```
hpc19@cl1n036:~/public_html/nickname> xrun omp_daxpy

#OMP daxpy: N=1048576 M=3 np=12 ::: p time(sec) perf(MFLOPS) S=t(1)/t(p)

     1      0.021050       149.437          1.000
     2      0.014708       213.873          1.431
     3      0.007322       429.634          2.875
     4      0.005641       557.627          3.732
     5      0.008197       383.745          2.568
     6      0.006517       482.684          3.230
     8      0.010810       291.000          1.947
    10      0.003337       942.588          6.308
    12      0.004306       730.547          4.889
```

¡ // run of daxpy on shared memory with OpenMP !

Construct the plots: **xplot**

Then, go back to h2 (exit) and run xplot
to construct the plots for timing and speedup

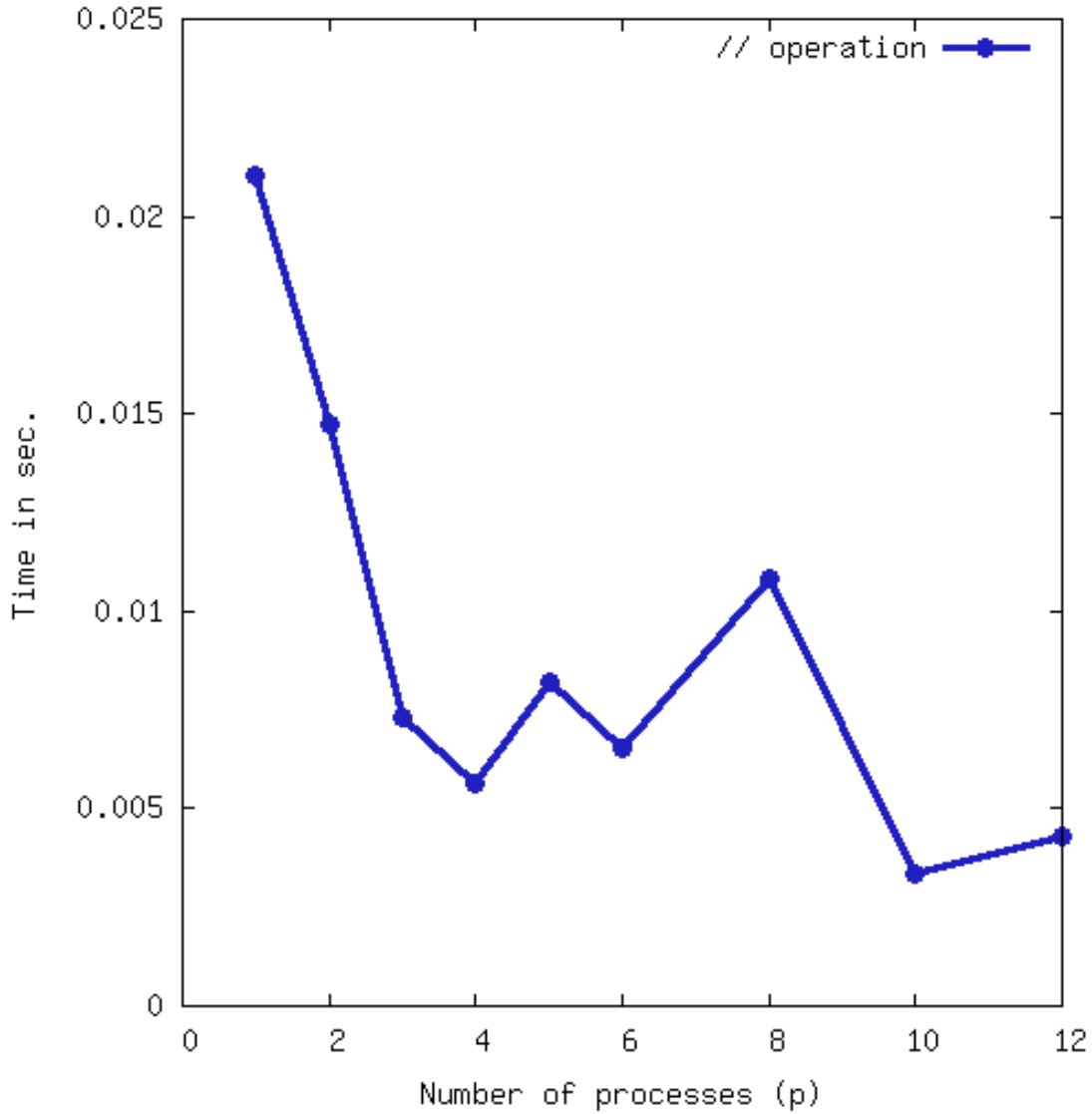
```
hpc19@cl1n036:~/public_html/nickname> exit
qsub: job 52193.h2.hpc completed

hpc19@h2:~/public_html/nickname> xplot
-rw-r-r- 1 hpc19 users 4796 Jan 14 01:12 s.png
-rw-r-r- 1 hpc19 users 4695 Jan 14 01:12 t.png
See rezulting "*.png" plots in:
http://cluster2.inm.ras.ru/~hpc19/nickname
where "s.png" is for Speedup and "t.png" is for Time plots
```

*¡ Two ***.png** files with plots were created on **h2** host !*

¡ These files can be seen in regular browser via internet !

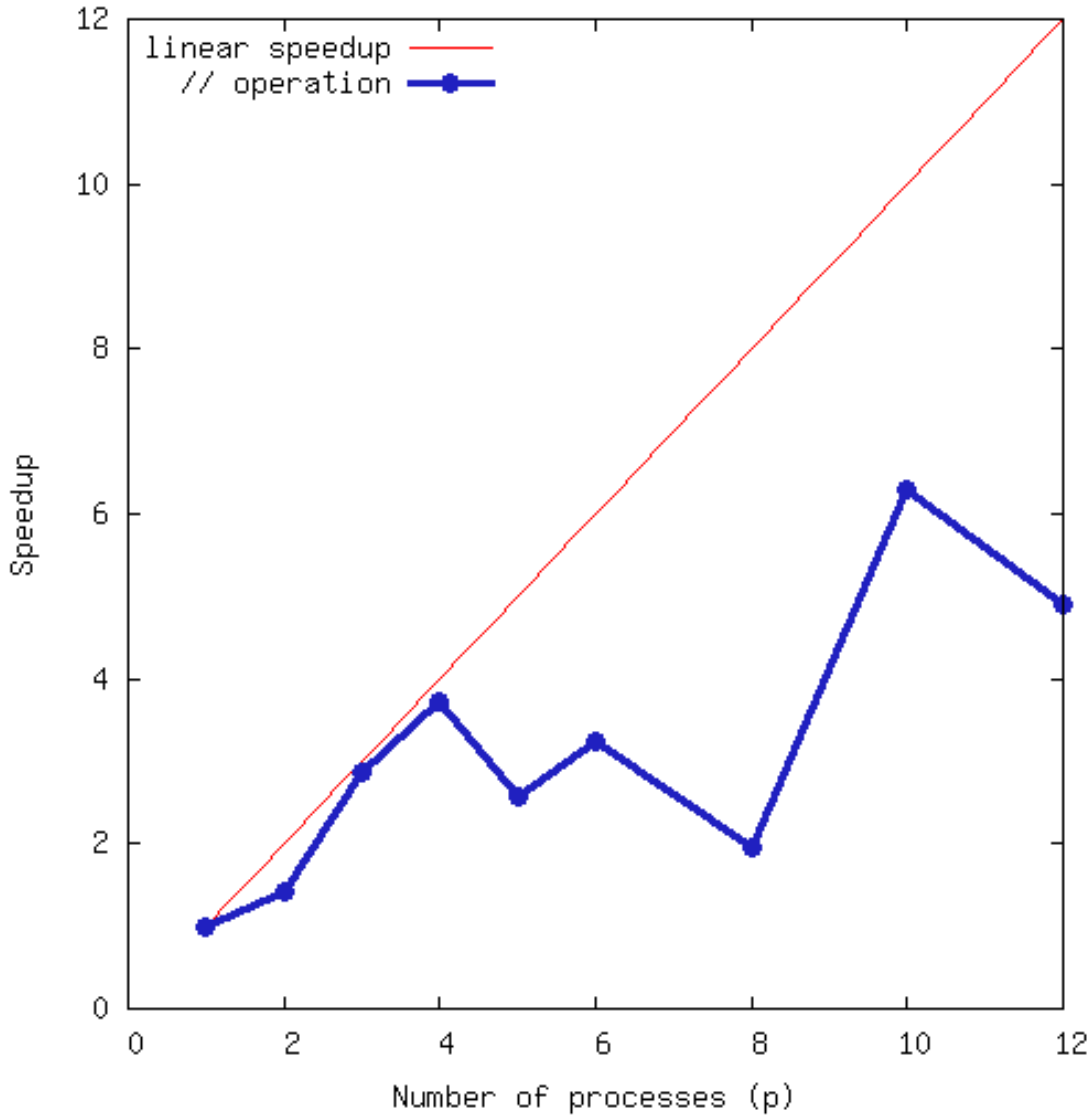
¡ Do not forget to press Ctrl+<R> (Reload) in time to update !

OMP daxpy: $\alpha \cdot \vec{x} + \vec{y}$



¡ *Time in sec.:* *http://cluster2.inm.ras.ru/~hpc19/nickname/t.png* !

OMP daxpy: $\alpha \cdot \vec{x} + \vec{y}$

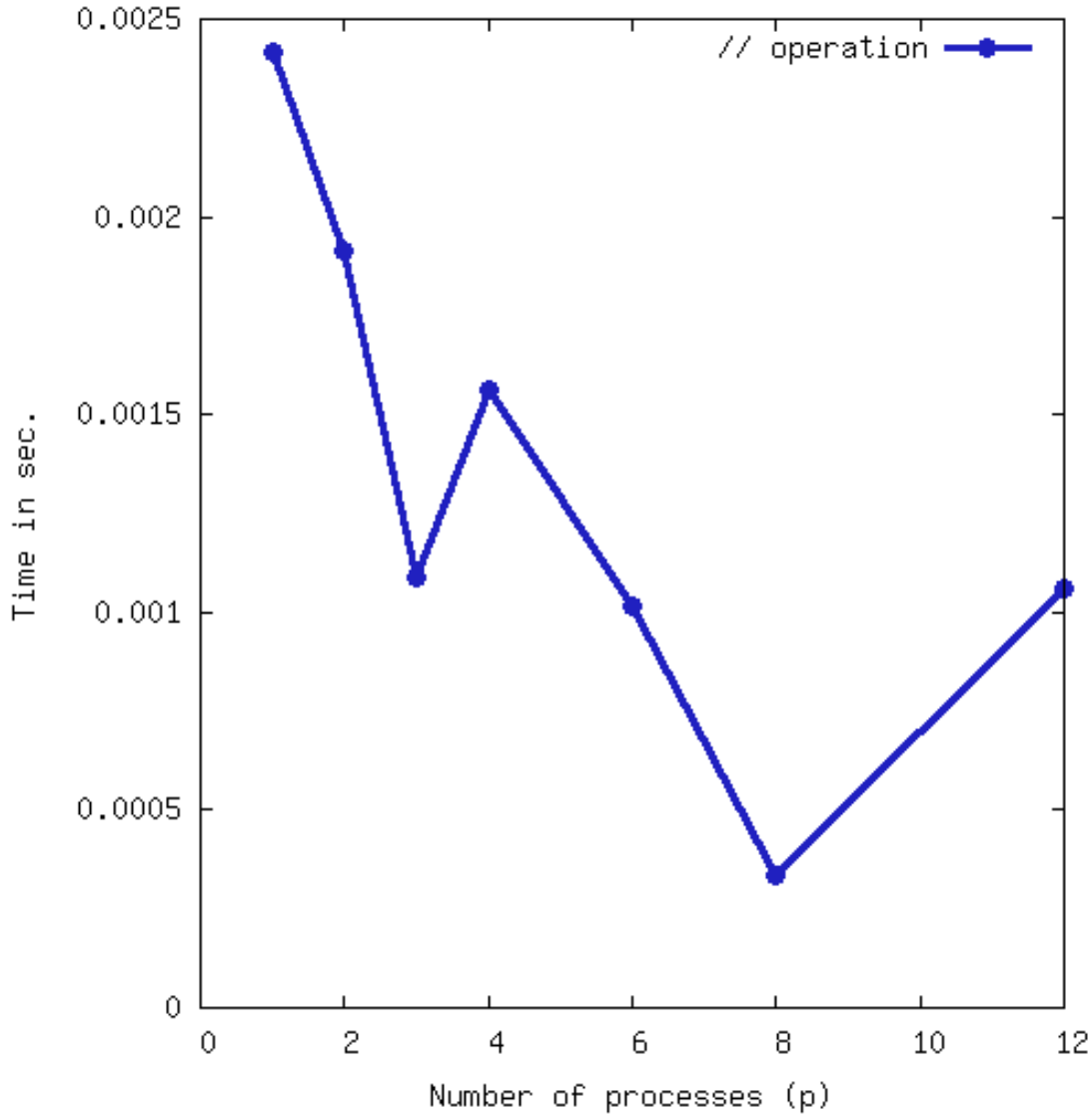


¡ Speedup: *http://cluster2.inm.ras.ru/~hpc19/nickname/s.png* !

The same test daxpy on MPI

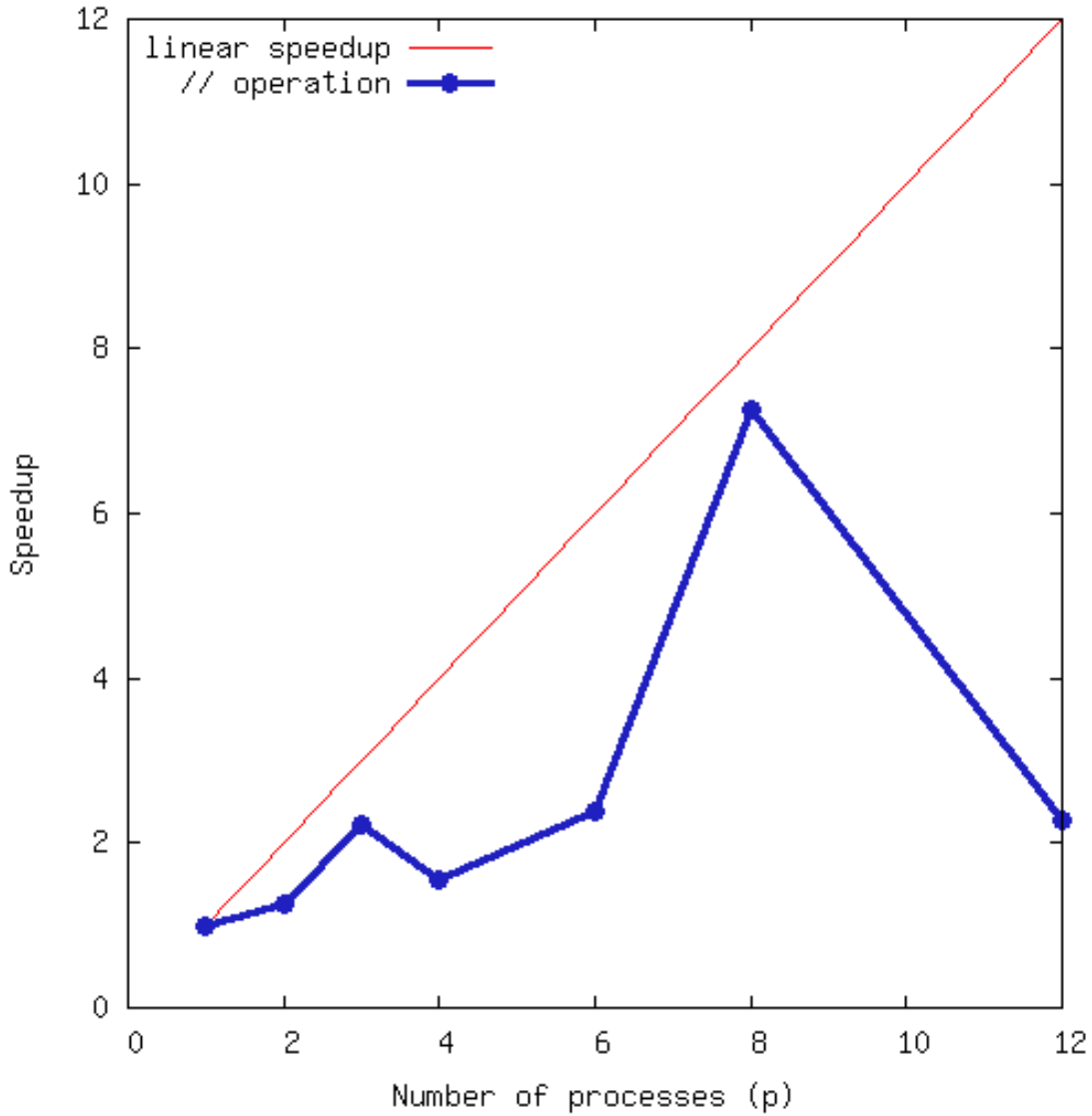
```
hpc19@h2:~/public_html/nickname> xqs
hpc19@cl1n036:~/public_html/nickname> xrun mpi_daxpy
hpc19@cl1n036:~/public_html/nickname> exit
hpc19@h2:~/public_html/nickname> xplot

See results in:
    http://cluster2.inm.ras.ru/~hpc19/nickname/*.png
```

MPI daxpy: $\alpha \cdot \vec{x} + \vec{y}$



¡ *Time in sec.:* *http://cluster2.inm.ras.ru/~hpc19/nickname/t.png* !

MPI daxpy: $\alpha \cdot \vec{x} + \vec{y}$



¡ Speedup: *http://cluster2.inm.ras.ru/~hpc19/nickname/s.png* !

The other tests...

```
hpc19@cl1n036:~/public_html/nickname> xlist
omp_daxpy     omp_norm     omp_mvm
mpi_daxpy     mpi_norm     mpi_mvm-bcast     mpi_mvm-isend
```

*¡ To save the plot rename *.png file by command:* ***mv s.png s-new.png*** *!*

There are some advanced commans...

Some additional commands:

xget test - get test source (on h2 or nodes)
xcc test - compile the local test source (on h2)
xrun - run the local parallel test (on nodes)

*¡ Add "**-v1**" to the name of the test — it will be much more interesting :) !*

Mathematical physics: INMOST

INMOST tests

```
hpc19@cl1n036:~/public_html/nickname> xlist
.....
t3d t4d t5h t5i t5j t6k t6l
```

¡ Pictures: *http://cluster2.inm.ras.ru/~hpc19/nickname/**JPG**/*.jpg* !

¡ ***exit*** *and* ***xplot*** *are NOT required,*
while ***Ctrl+<R>*** *is requred !*

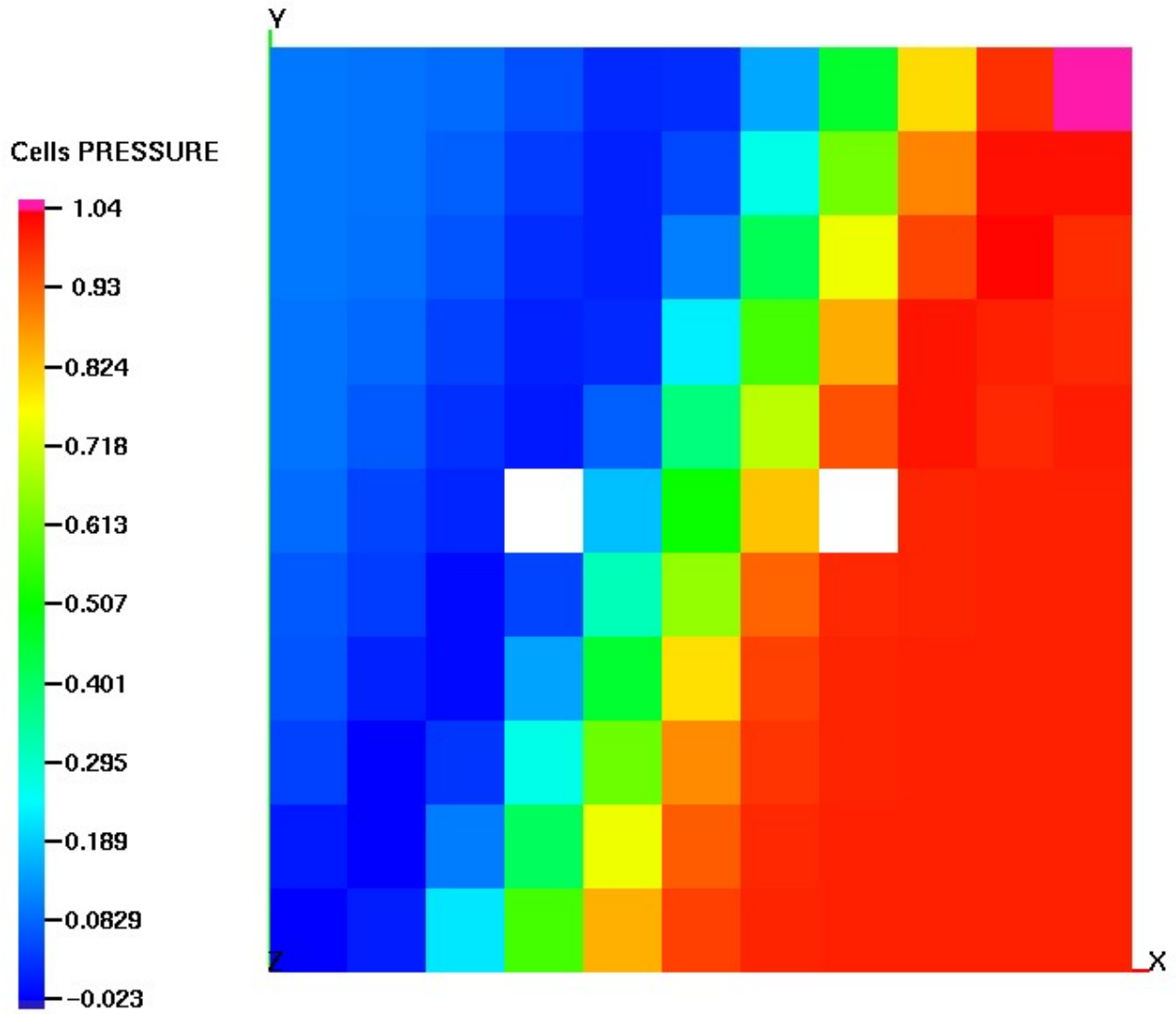
INMOST tests: description

- t3d — anisotropic diffusion with two wells with a mimetic scheme (reference operator method)
- t4d — two-well anisotropic diffusion with a non-linear finite-volume scheme
- t5h — oil in the reservoir with anisotropic permeability that starts the flow in a zigzag
- t5i — oil on several layers of SPE10 test
- t5j — oil in the real Norn field
- t6k — incompressible fluid in the cavity
- t6l — incompressible fluid in a stepped region (L-shaped domain)

¡ INMOST tests were prepared by K.M. Terekhov !

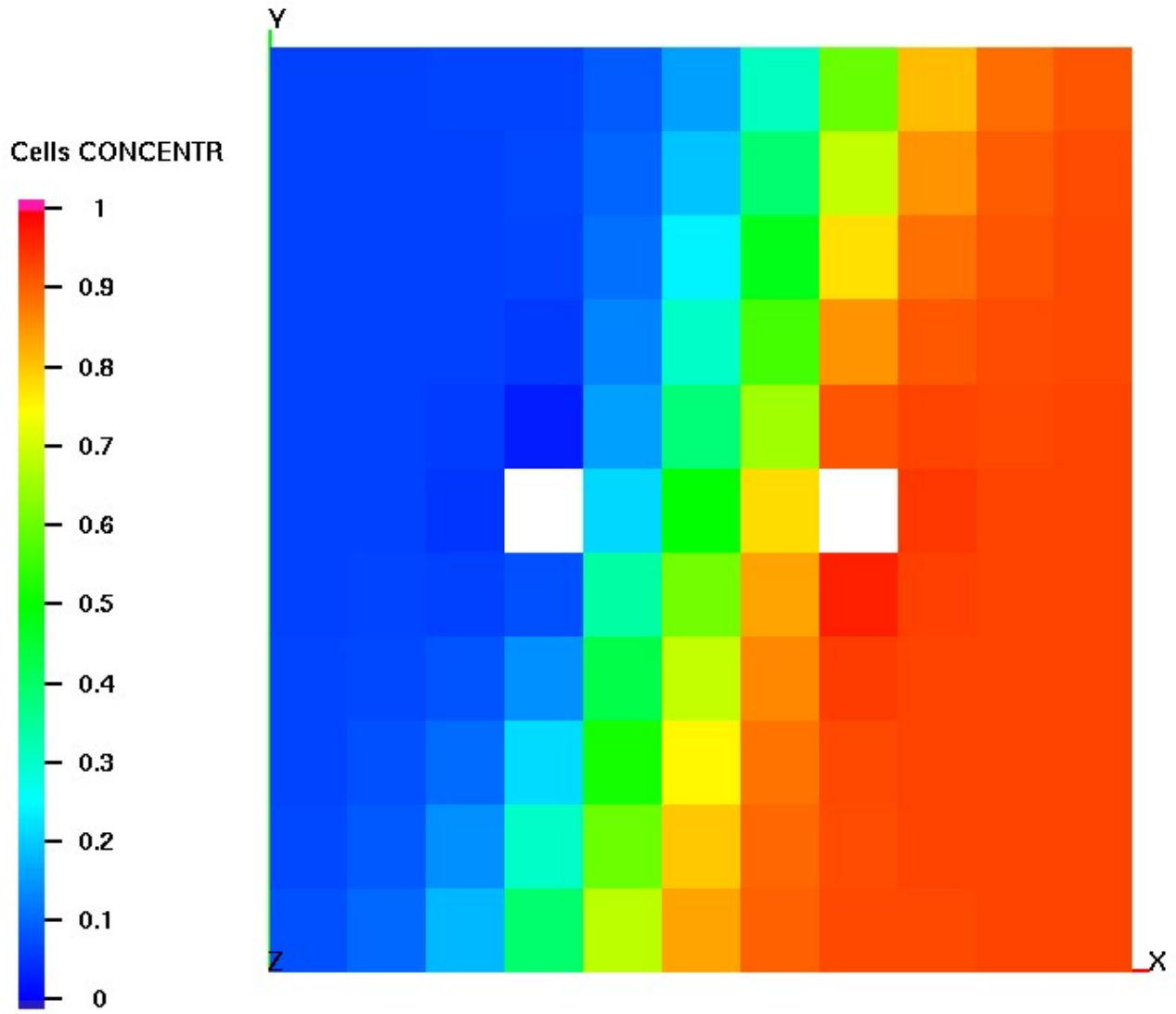
t3d: two-well anisotropic diffusion

with a mimetic scheme (reference operator method)



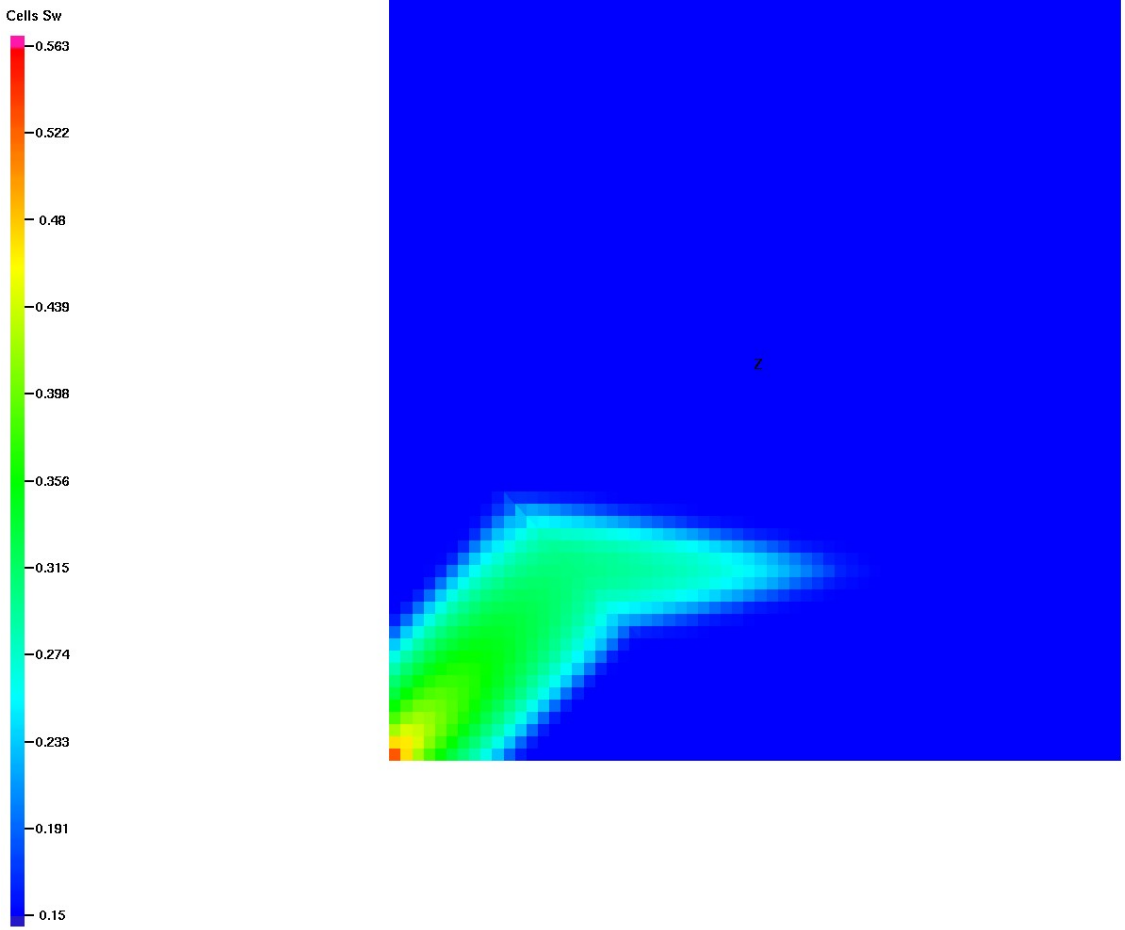
t4d: two-well anisotropic diffusion

with a non-linear finite-volume scheme

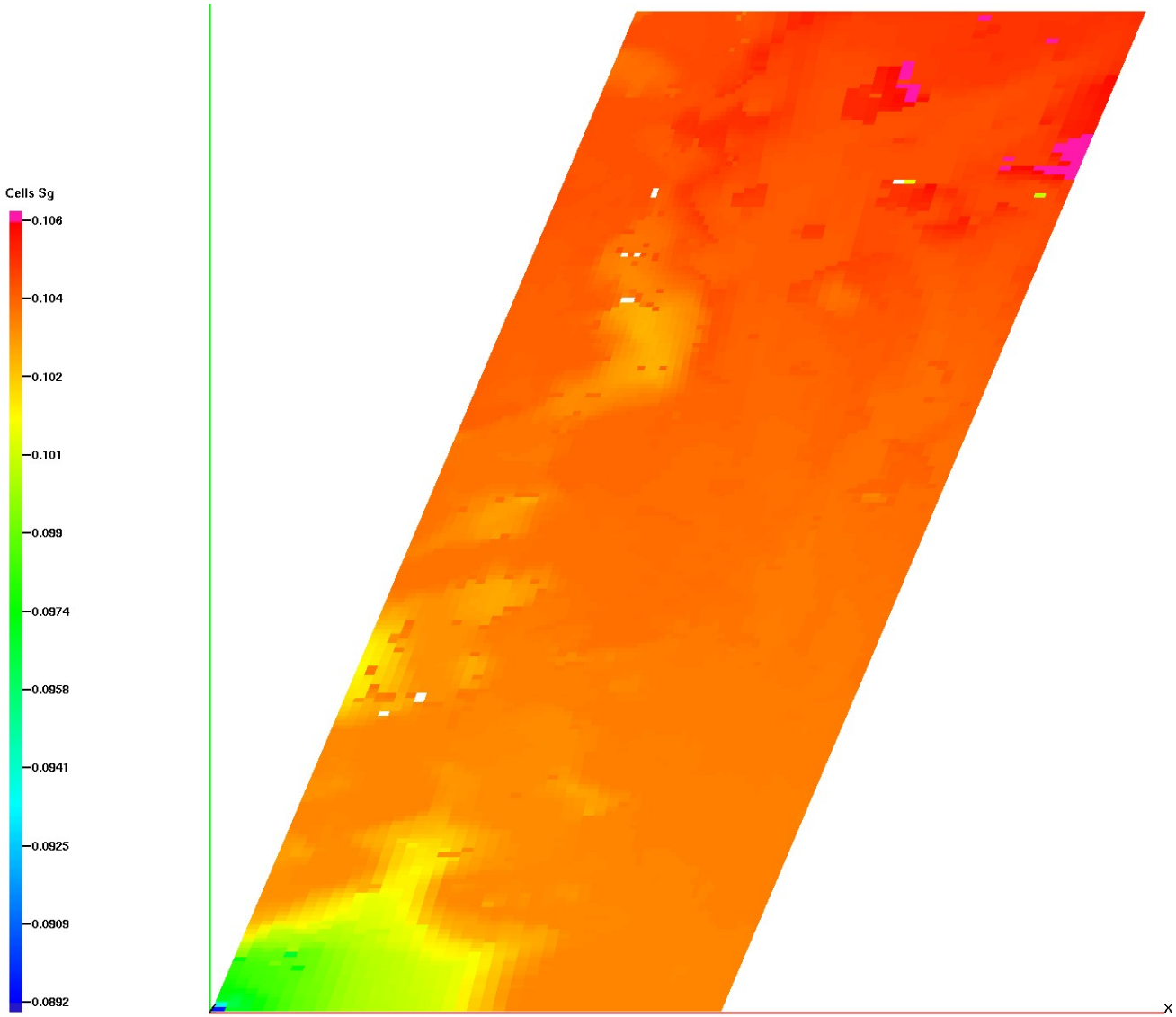


t5h: oil in the reservoir with anisotropic permeability

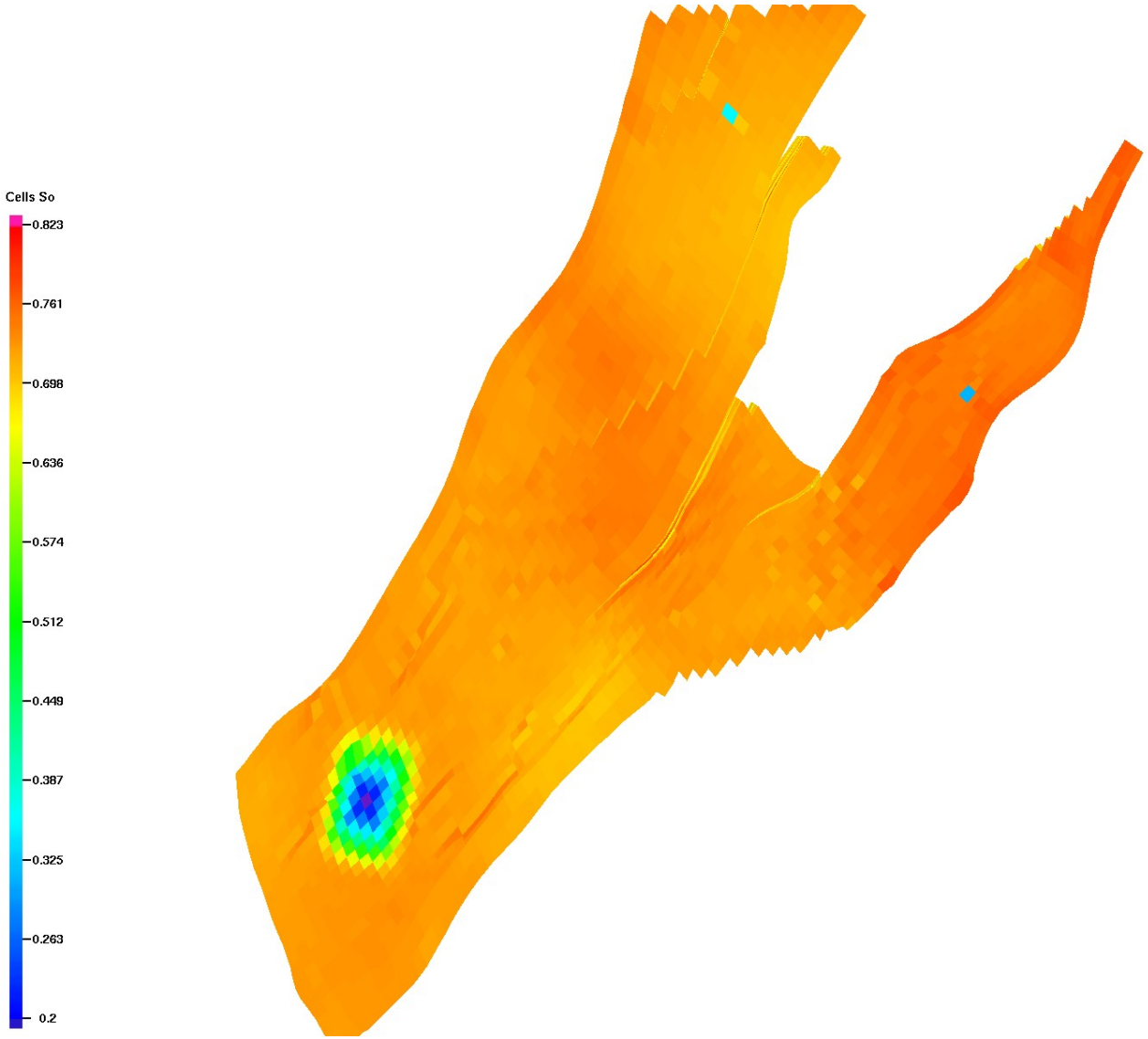
that starts the zigzag flow



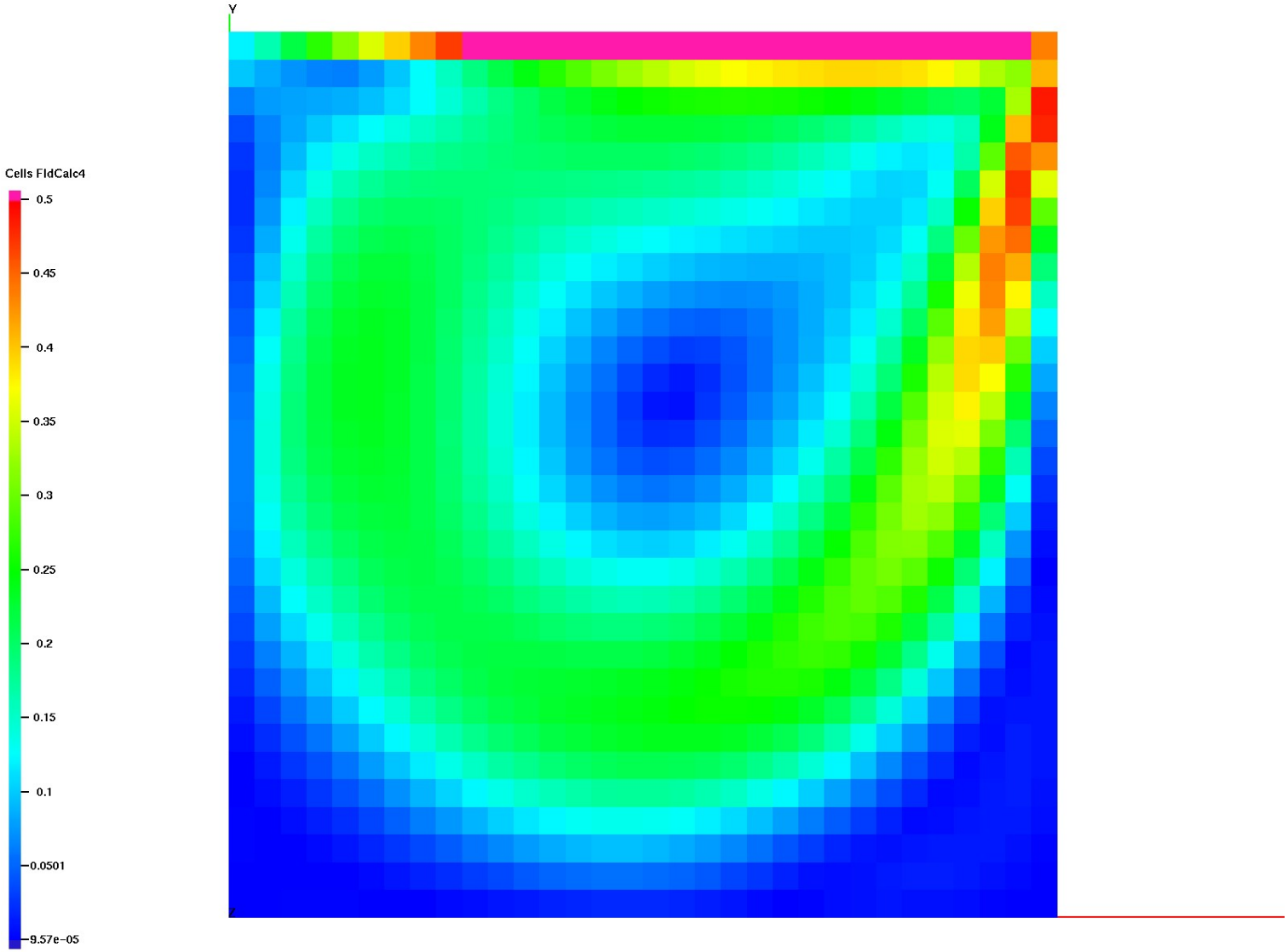
t5i: oil in several layers of SPE10 test



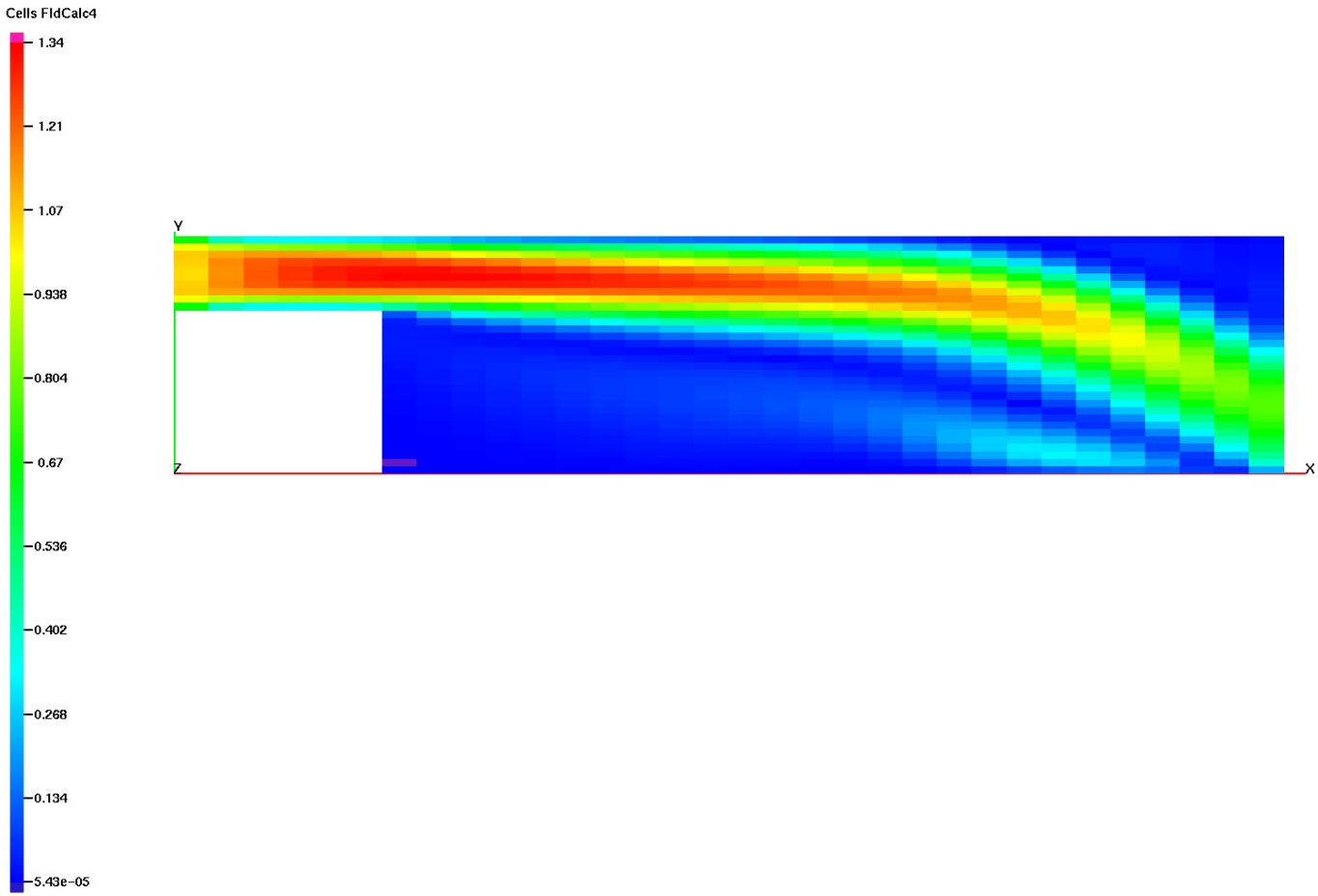
t5j: oil in the real Norn field



t6k: incompressible fluid in a cavity



t6l: incompressible fluid in a stepped region (L-shaped domain)



INMOST

http://inmost.org — homepage

Documentation:

https://github.com/INMOST-DEV/INMOST/wiki — wiki

http://www.inmost.org/Doxygen/html/annotated.html — doxygen

INMOST: download and compile

Wiki instructions:
https://github.com/INMOST-DEV/INMOST/wiki/0400-Compilation-Linux

```
$ git clone https://github.com/INMOST-DEV/INMOST
$ mkdir INMOST/build
$ cd INMOST/build
$ cmake -DUSE_MPI=ON -DCOMPILE_EXAMPLES=ON ..
$ make
```

INMOST: examples

Wiki instructions:

https://github.com/INMOST-DEV/INMOST/wiki/1800-Examples

```
$ cd Examples
### Parallel Grid Generation
$ mpiexec -np 4 GridGen/GridGen 4 32 32 32 grid.pvtk

### Parallel Finite Volume Discretization
$ mpiexec -np 4 FVDiscr/FVDiscr grid.pvtk A.mtx b.rhs

### Solve the Matrix Stored in MTX Format
$ mpiexec -np 4 MatSolve/MatSolve -m A.mtx -b b.rhs

### If you have ParaView installed:
$  paraview --data=grid.pvtk
   P_OWNER_PROCESSOR - partitioning to processors
$  paraview --data=result.pvtk
   Solution - the solution to the problem
```