



ML/DL/HPC ecosystem for solving applied tasks TUTORIAL

**D.V. Podgainy, A. Rahmonova, S. Shadmehri,
O.I. Streltsova, M.I. Zuev**

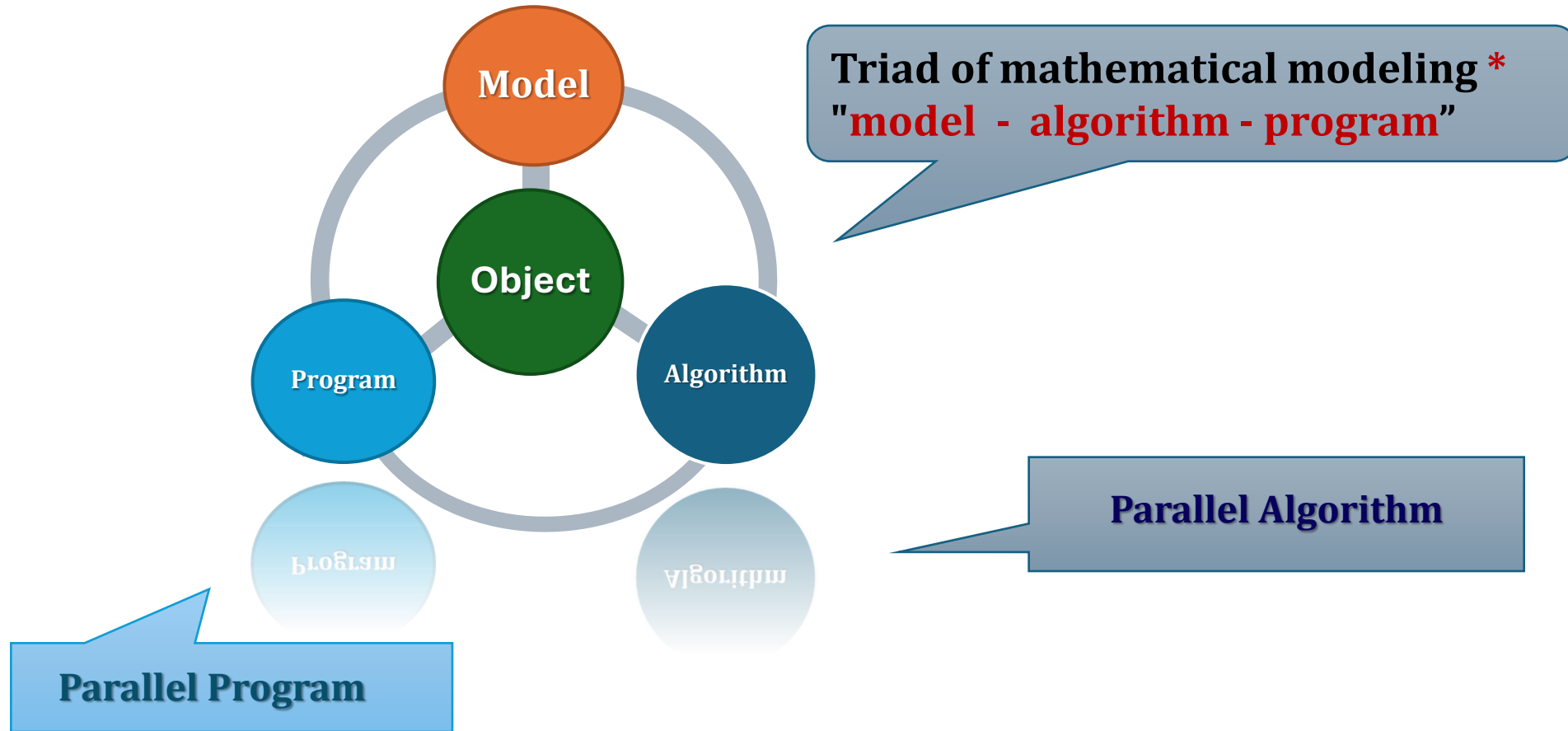
Heterogeneous Computation Team HybriLIT
Laboratory of Information Technologies, JINR, Dubna, Russia

The work was performed with the support of
the Russian Science Foundation within the
framework of project No. 22-71-10022

HYBRILIT WORKSHOP
2025

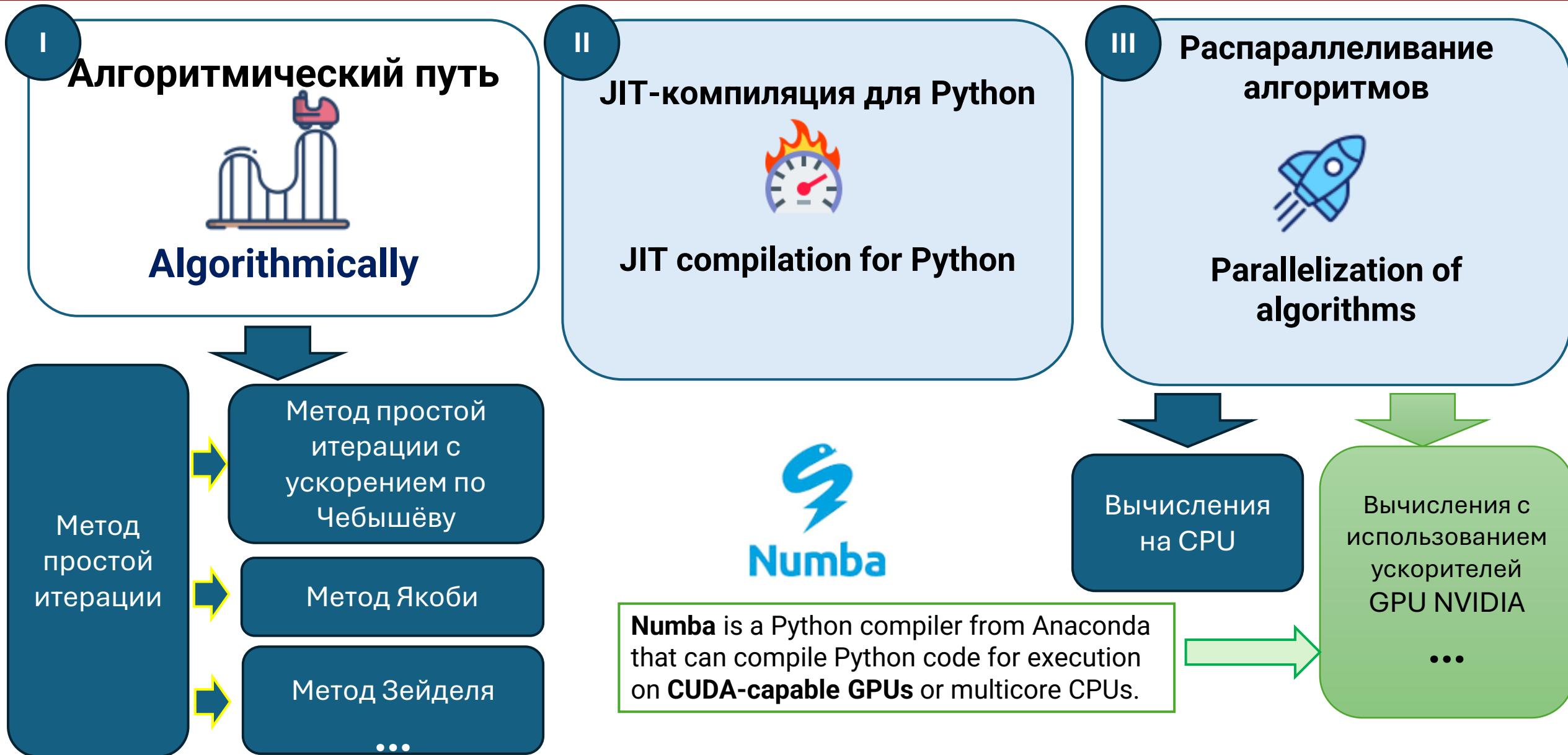


Goal: Efficient parallelization of complex numerical problems



* Самарский А.А., Михайлов А.П. Математическое моделирование. Идеи, методы, примеры. М.: Наука, 2001.

Goal: Accelerating computations for complex numerical problems



Решение задачи Дирихле для уравнения Пуассона итерационными методами

I. Метод простой итерации (метод Якоби) с постоянным параметром τ :

$$\frac{y^{s+1} - y^s}{\tau} = \Delta y^s + f(x), \quad x \in \omega_h, \quad y^s|_{\gamma_h} = \mu(x), \\ y^s = y_0, \quad s = 0, 1, \dots$$

где s - номер итерации.

Отметим, что в случае квадратной сетки, т.е. при $h_1 = h_2 = h$ оптимальным значение параметра τ (при котором для достижения заданной точности затрачивается минимум итераций) определяется выражением:

$$\tau_0 = \frac{h^2}{4},$$

а итерационная схема имеет вид [1, с.456]:

$$y_{i_1, i_2}^{s+1} = \frac{1}{4} (y_{i_1-1, i_2-1}^s + y_{i_1+1, i_2}^s + y_{i_1, i_2-1}^s + y_{i_1, i_2+1}^s) + \frac{h^2}{4} f_{i_1, i_2}, \quad i_\alpha = 1, \dots, N_\alpha - 2, \quad y^s|_{\gamma_h} = \mu(x).$$

Оценка асимптотического порядка числа итераций для достижения точности ϵ :

$$n(\epsilon) \geq \frac{2 \ln(1/\epsilon)}{\pi^2 h^2}.$$

```
import plotly.graph_objects as go

import pandas as pd

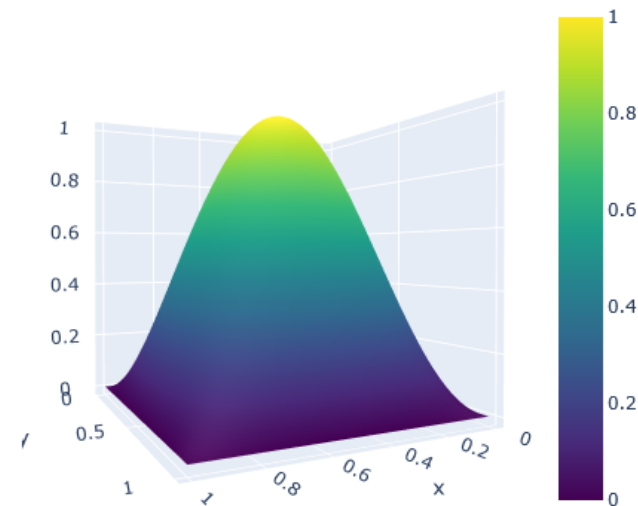
# Generate data
x1 = np.linspace(0.0, 1.0, num=N1, endpoint=True)
x2 = np.linspace(0.0, 1.0, num=N2, endpoint=True)

fig = go.Figure(data=[go.Surface(x=x1, y=x2, z=y1, colorscale='Viridis')])
#fig = go.Figure(x=xGrid, y=yGrid, z=z_data, colorscale='Viridis')

fig.update_layout(title=dict(text='Poisson equation'), autosize=False,
                    width=500, height=500,
                    margin=dict(l=65, r=50, b=65, t=90))

fig.show()
```

Poisson equation



I.2. Метод верхней релаксации

Метод верхней релаксации определяется следующим образом [2]:

$$\begin{cases} y_{i_1, i_2}^{s+1} = (1 - \omega) y_{i_1, i_2}^s + \omega \left(\frac{2}{h_1^2} + \frac{2}{h_2^2} \right)^{-1} \left(\frac{y_{i_1-1, i_2}^{s+1} + y_{i_1+1, i_2}^s}{h_1^2} + \frac{y_{i_1, i_2-1}^{s+1} + y_{i_1, i_2+1}^s}{h_2^2} - f_{i_1, i_2} \right), & i_\alpha = 1, \dots, N_\alpha - 2, x \in \omega_h, \\ y|_{\gamma_h} = \mu(x), \end{cases}$$

где $\omega \in (1, 2)$.

Вычисления начинаются с $i_1 = 1, i_2 = 1$ и продолжаются либо по строкам, либо по столбцам. Найденное приближение y_{i_1, i_2}^{s+1} размещается на месте y_{i_1, i_2}^s сразу при проходе в цикле.

Оптимальное значение параметра ω (при $N_1 = N_2 = N$):

$$\omega = \omega_0 = \frac{2}{1 + \sin \frac{\pi}{N}}.$$

[2] AlroWiki Алгоритм верхней релаксации численного решения двумерного уравнения Пуассона

```
[66]: import deepxde as dde
import numpy as np
```

```
Using backend: pytorch
Other supported backends: tensorflow.compat.v1, tensorflow, jax, paddle.
paddle supports more examples now and is recommended.
```

```
[67]: geom = dde.geometry.Rectangle([0, 0], [1, 1])
```

```
[76]: # Define sine function
if dde.backend.backend_name == "pytorch":
    sin = dde.backend.pytorch.sin
else:
    from deepxde.backend import tf

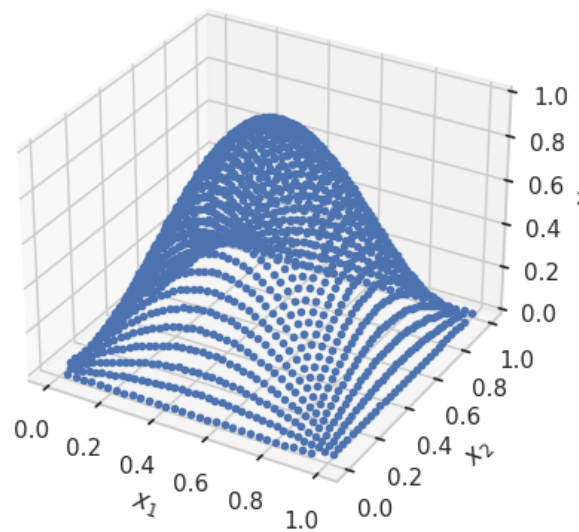
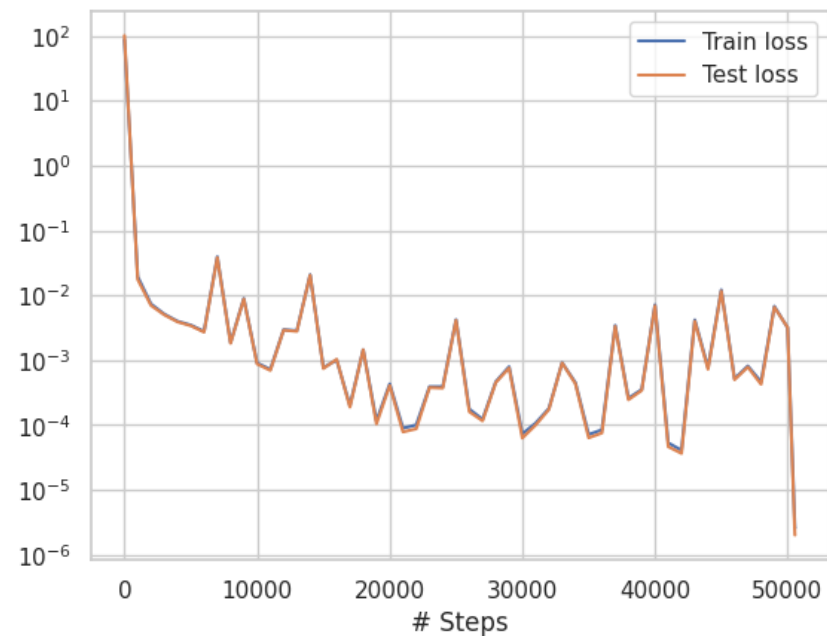
    sin = tf.sin
```

```
[77]: def fun_f(x,y):
# s = fun_s(x,y)
return ( - 2.0*np.pi**2 * np.sin(np.pi*x)*np.sin(np.pi*y) )
```

```
[78]: def pde(x, y):
dy_xx = dde.grad.hessian(y, x, i=0, j=0)
dy_yy = dde.grad.hessian(y, x, i=1, j=1)
f = ( - 2.0*np.pi**2 * sin(np.pi*x[:, 0:1])*sin(np.pi*x[:, 1:2]) )
return -dy_xx - dy_yy + f
```

```
[79]: def boundary(_, on_boundary):
return on_boundary
```

```
Saving loss history to /home/amp001/morhology_test/loss.dat ...
Saving training data to /home/amp001/morhology_test/train.dat ...
Saving test data to /home/amp001/morhology_test/test.dat ...
```



Спасибо за внимание!

