The VM2D Code: Open-Source Software Implementation of Meshless Lagrangian Vortex Methods for 2D Incompressible Flow Simulation

Kseniia S. Kuzmina, **Ilia K. Marchevsky**, **Evgeniya P. Ryatina**

Bauman Moscow State Technical University
Applied mathematics dep.

Ivannikov Institute for System Programming
of the Russian Academy of Sciences

Xi'an
October 17^{th}, 2018

Motivation

Vortex methods are an efficient tool for incompressible flow simulations and solving coupled hydroelastic problems:

- aerospace systems: wings, launch systems;
- aircraft trails (at flight level and near aerodromes);
- main and tail rotors of helicopters;
- parachute systems;
- problems of industrial aerodynamics.

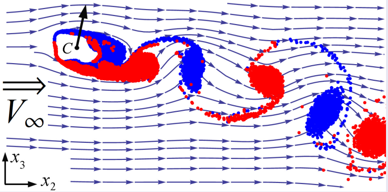
The main feature:

Vortex methods are **meshless** Lagrangian methods

- arbitrary motion of the body;
- there is no need to deform or reconstruct mesh;
- it is ‘convenient’ to simulate external flows;
- computational sources are ‘concentrated’ at the region with non-zero vorticity.

Continuity equation &
Navier – Stokes equation:

$$\nabla \cdot \vec{V} = 0, \qquad \frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \nabla)\vec{V} = \nu \Delta \vec{V} - \frac{\nabla p}{\rho}.$$



Boundary conditions:

Perturbation decay on infinity: $\vec{V}(\vec{r}, t) \to \vec{V}_\infty, \quad p(\vec{r}, t) \to p_\infty, \quad |\vec{r}| \to \infty.$

No-slip condition: $\vec{V}(\vec{r}, t) = \vec{0}$ or $\vec{V}(\vec{r}, t) = \vec{V}_K(\vec{r}, t), \quad \vec{r} \in K.$

Initial conditions:

Given velocity field: $\vec{V}(\vec{r}, 0) = \vec{V}_0(\vec{r}).$

$\vec{V}$ — velocity field; $\vec{V}_\infty$, p_∞ — velocity and pressure on infinity;
p — pressure; ν — kinematic viscosity coefficient.

Vortex methods. Main ideas

Vorticity $\vec{\Omega}(\vec{r}, t) = \mathrm{rot}\,\vec{V}(\vec{r}, t)$ — main computational variable

- Navier – Stokes equation in Helmholtz form (2D case):

$$\frac{\partial\vec{\Omega}}{\partial t} + \mathrm{rot}\big(\vec{\Omega} \times (\vec{V} + \vec{W})\big) = 0, \quad \vec{V} = \vec{V}(\vec{\Omega}).$$

- This equation can be considered as the transfer equation for vorticity $\vec{\Omega}$, which moves with velocity $\vec{V} + \vec{W}$,

$$\vec{W}(\vec{r}, t) = -\nu\frac{\nabla\Omega}{\Omega} \text{ — diffusive velocity}^{1,2}, \quad \Omega = \vec{\Omega}\cdot\vec{k}.$$

- There is no vorticity generation in the flow domain.

- New vorticity is generated only on the surface line of the airfoil; $\gamma(\vec{r})$, $\vec{r} \in K$ — unknown intensity of the vortex sheet at the surface line of the airfoil.

1. Dynnikova G.Ya. Vortex motion in two-dimensional viscous fluid flows // Fluid Dynamics, 2003. Vol. 38. № 5. Pp. 670—678.
2. Ogami Y., Akamatsu T. Viscous flow simulation using the discrete vortex model — the diffusion velocity method // Computers and Fluids, 1991. V. 19, No. 3–4. Pp. 433–441.

Vortex sheets and source sheet

Airfoil influence is simulated by the attached vortex and source sheets and the free vortex sheet on the airfoil surface line:

- Intensity of the attached vortex sheet

$$\gamma_{\mathrm{att}}(\vec{r},\,t) = \vec{V}_K(\vec{r},\,t)\cdot\vec{\tau}(\vec{r},\,t), \quad \vec{r}\in K.$$

- Intensity of the attached source sheet

$$q_{\mathrm{att}}(\vec{r},\,t) = \vec{V}_K(\vec{r},\,t)\cdot\vec{n}(\vec{r},\,t), \quad \vec{r}\in K.$$

- Intensity of the free vortex sheet $\gamma(\vec{r},\,t)$ can be found from boundary condition on the airfoil surface line.

$\vec{\tau}(\vec{r},\,t)$ и $\vec{n}(\vec{r},\,t)$ — unit tangent and normal vectors.

3. Zhukovsky N.E. On attached vortices. 1908.

Velocity field reconstruction

Generalized Biot – Savart law

$$\vec{V}(\vec{r},t) = \vec{V}_\infty + \frac{1}{2\pi}\int_{S(t)} \frac{\vec{\Omega}(\vec{\xi},t)\times(\vec{r}-\vec{\xi})}{|\vec{r}-\vec{\xi}|^2}dS + \frac{1}{2\pi}\oint_{K(t)} \frac{\vec{\gamma}(\vec{\xi},t)\times(\vec{r}-\vec{\xi})}{|\vec{r}-\vec{\xi}|^2}dl_K +$$

$$+\frac{1}{2\pi}\oint_{K(t)} \frac{\vec{\gamma}_{\mathrm{att}}(\vec{\xi},t)\times(\vec{r}-\vec{\xi})}{|\vec{r}-\vec{\xi}|^2}dl_K + \frac{1}{2\pi}\oint_{K(t)} \frac{q_{\mathrm{att}}(\vec{\xi},t)(\vec{r}-\vec{\xi})}{|\vec{r}-\vec{\xi}|^2}dl_K,$$

$$\vec{\gamma}_{\mathrm{att}} = \gamma_{\mathrm{att}}\vec{k}, \qquad \vec{\gamma} = \gamma\vec{k}, \qquad \vec{\Omega} = \Omega\vec{k}, \qquad \vec{n}(\vec{r},\,t)\times\vec{\tau}(\vec{r},\,t) = \vec{k}.$$

The limit value of the floe velocity at the airfoil surface line

$$\vec{V}_-(\vec{r},\,t) = \vec{V}(\vec{r},\,t) - \frac{\gamma(\vec{r},\,t)-\gamma_{\mathrm{att}}(\vec{r},\,t)}{2}\vec{\tau}(\vec{r},\,t) + \frac{q_{\mathrm{att}}(\vec{r},\,t)}{2}\vec{n}(\vec{r},\,t)$$

- Cauchy – Lagrange integral for potential flows:

$$\frac{\partial \Phi}{\partial t} + \frac{p}{\rho} + \frac{\vec{V}^2}{2} = \mathrm{const};$$

- generalized Cauchy – Lagrange integral for non-potential flows [5];
- the expressions for computation of the integral loads acting the airfoil [6].

4. Dynnikova G.Ya. An analog of the Bernoulli and Cauchy-Lagrange integrals for a time-dependent vortex flow of an ideal incompressible fluid // Fluid Dynamics, 2000. V. 35, No. 1. Pp. 24–32.

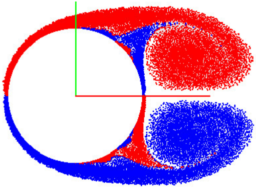
5. Dynnikova G.Ya. The Integral Formula for Pressure Field in the Nonstationary Barotropic Flows of Viscous Fluid // Journal of Mathematical Fluid Mechanics, 2014. V. 16, No. 1. Pp. 145–162

Vortex wake simulation

The vorticity distribution in the flow domain is simulated by the large number of the separate vortex elements (VE).

$$\Omega(\vec{r}) = \sum_{i=1}^{n} \Gamma_i \delta(\vec{r} - \vec{r}_i),$$

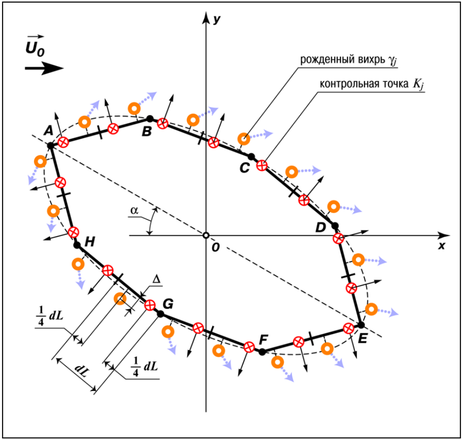
Γ_i — VE circulations, $\vec{r}_i$ — VE positions.



Vortex elements motion

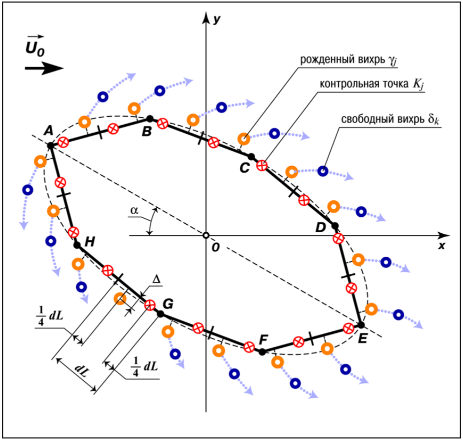
Motion equations: $$\frac{D\vec{\Omega}}{Dt} = 0 \quad \Leftrightarrow \quad \begin{cases} \Gamma_i = \text{const}, \\ \dfrac{d\vec{r}_i}{dt} = \vec{V}(\vec{r}_i) + \vec{W}(\vec{r}_i), \quad i = 1, \dots, n. \end{cases}$$

$$\vec{V}(\vec{r}_i) = \sum_{\substack{j=1 \\ j \neq i}}^{n} \underbrace{\frac{\Gamma_j}{2\pi} \frac{\vec{k} \times (\vec{r}_i - \vec{r}_j)}{|\vec{r}_i - \vec{r}_j|^2}}_{\vec{v}_{ij}} + \vec{V}_\gamma + \vec{V}_\gamma^{\text{att}} + \vec{V}_q^{\text{att}} + \vec{V}_\infty.$$

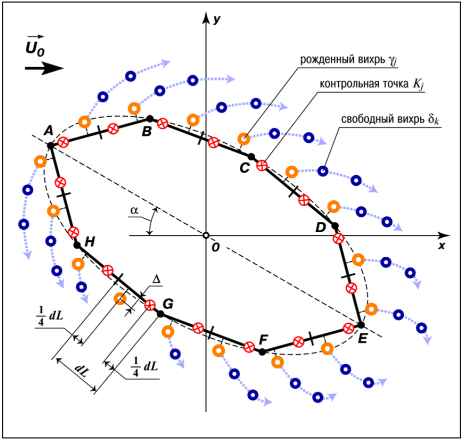


$\vec{U}_0$
y
x
0
α
A
B
C
D
E
F
G
H
рожденный вихрь γ_j
контрольная точка K_j
Δ
$\frac{1}{4}\,dL$
dL
$\frac{1}{4}\,dL$

Vortex elements motion



Vortex elements motion



VM2D code

- C++ language (GCC, Intel C++ Compiler, MSVC)
- Linux / Windows
- Parallel technologies: OpenMP, MPI, NVidia CUDA

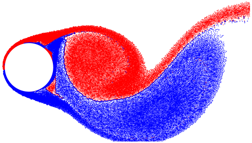
```
##  ## ##   ##  ####  #####
##  ## ### ### ##  ## ##  ##
##  ## ## # ##    ## ##  ##
 ####  ##   ##   ## ##  ##
  ##   ##   ## ###### #####
```

- Linear algebra: Eigen (eigen.tuxfamily.org)
- Open source code: https://github.com/vortexmethods/VM2D/
- Doxygen-documentation: http://vortexmethods.github.io/VM2D/ (in Russian)
- Modular structure allows to increase the capabilities

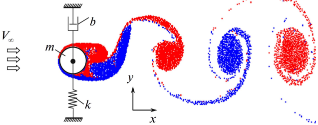
Versions

- VM2Dv.1.0 (December, 1st, 2017, ISPRASOpen 2017)
- VM2Dv.1.1 (April, 2nd, 2018, PaCT 2018)
  + Moving airfoils
  + NVidia CUDA
- VM2Dv.1.2 (June, 14th, 2018, ECCM-ECFD 2018)
  + Hydroenastic problems
  + Rotating airfoils
- VM2Dv.1.3 (September, 26th, 2018, VoenMech 2018)
  + Pressure and velocity computation in given points
- VM2Dv.1.4 (October, 16th, 2018, ICVFM 2018)
  + Extended set of the functions which use NVidia CUDA

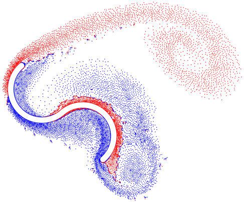
Some capabilities of the VM2D



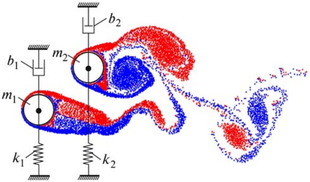
Flow around immovable airfoil



Coupled hydroelastic problems



Rotating airfoils



System of airfoils

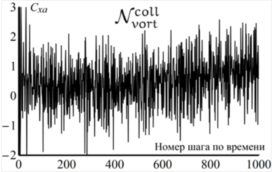
Main computational blocks

- **Vorticity generation**
  - Boundary integral equation solving
- **Vortex elements velocities computation**
  - Convective velocities
  - Diffusive velocities
- **Loads computation**
  - Hydrodynamic force and moment
  - Viscous stresses
  - Velocity and pressure computation at the given points
- **Vortex wake evolution**
  - ODE system solving
  - No-throw control
  - Vortex wake resrtucturing

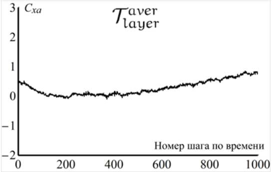
Authors developments

- **New approach to satisfy boundary condition:**
  - Fredholm integral equation of the 2-nd kind instead of 1-st kind;
  - bounded kernel instead of singular one
  - numerical schemes with high order of accuracy

7. Kempka S.N., Glass M.W., Peery J.S., Strickland J.H. Accuracy considerations for implementing velocity boundary conditions in vorticity formulations // SANDIA REPORT. SAND96-0583, UC-700, 1996. 52 p.
8. Kuzmina K., Marchevskii I., Moreva V., Ryatina E. Numerical scheme of the second order of accuracy for vortex methods for incompressible flow simulation around airfoils // Russian Aeronautics, 2017, V. 60, № 3.
9. Kuzmina K., Marchevskii I., Moreva V. Vortex Sheet Intensity Computation in Incompressible Flow Simulation Around an Airfoil by Using Vortex Methods // Mathematical Models and Computer Simulations, 2018, Vol. 10, No. 3, pp. 276–287.
10. Kuzmina K., Marchevsky I., Ryatina E. Exact analytical formulae for linearly distributed vortex and source sheets influence computation in 2D vortex methods // IOP Conf. Series: Journal of Physics: Conf. Series, 2017. V. 918, 012013. 9 p.

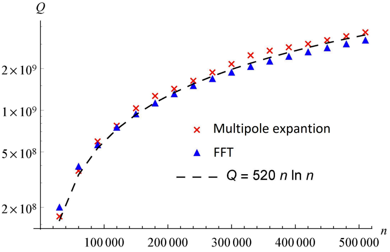


‘Old’ schemes



‘New’ schemes

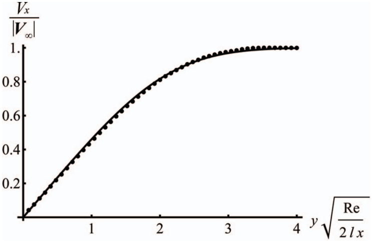
- **Quantitative estimations of the algorithm blocks complexities**
  - taking into account specifity of the problem statement;
  - estimation of parallelization opportunity.
- **Quantitative estimations of the complexity and accuracy of fast methods**
  - method with multipole expantion;
  - method with FFT.
- **Algorithms adaptation to perform computations on GPUs**

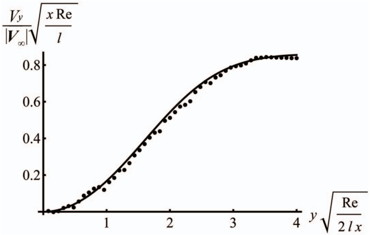


Direct computation: $6n^2$ operations

11. Kuzmina K.S., Marchevsky I.K., Moreva V.S. Parallel implementation of vortex element method on CPUs and GPUs // Procedia Computer Science. 2015. Vol. 66. P. 73-82.

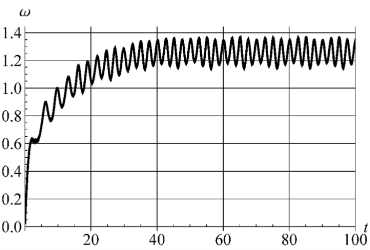
Examples: Blasius problem



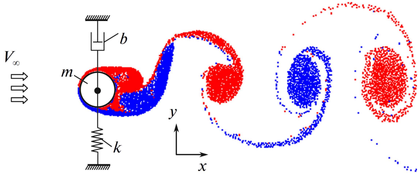


Examples: flow around Savonius rotor

- Rotor with small moment of inertia
- Coupling strategy with monolithic approach



Example: resonance of the circular airfoil



Constrain is linear viscoelastic (Kelvin — Voigt-type)

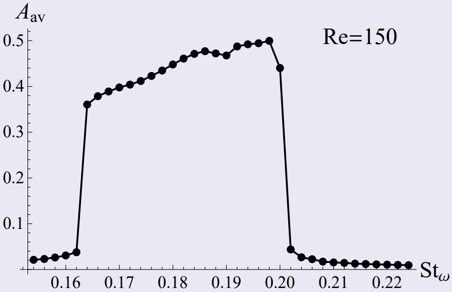
$$m\ddot{y}_* + b\dot{y}_* + ky_* = F_y.$$

y_* — deviation from the equilibrium position, F_y — lift force,

m — airfoil's mass, b — damping factor, k — constraint's elasticity.

Example: resonance of the circular airfoil

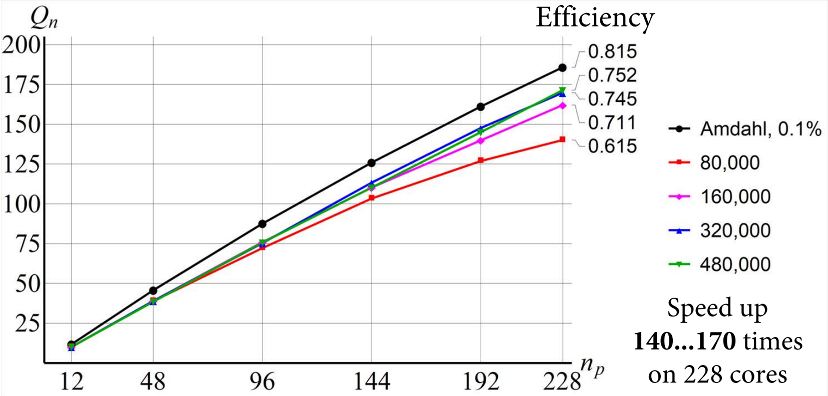
$$\rho_0 = 1.0, \qquad V_\infty = 3.0,$$
$$D = 1.0, \qquad \mathrm{Re} = 150,$$
$$b = 0.731, \qquad \rho/\rho_0 \approx 50.$$



Amplitudes for different Sh_ω

MPI + OpenMP

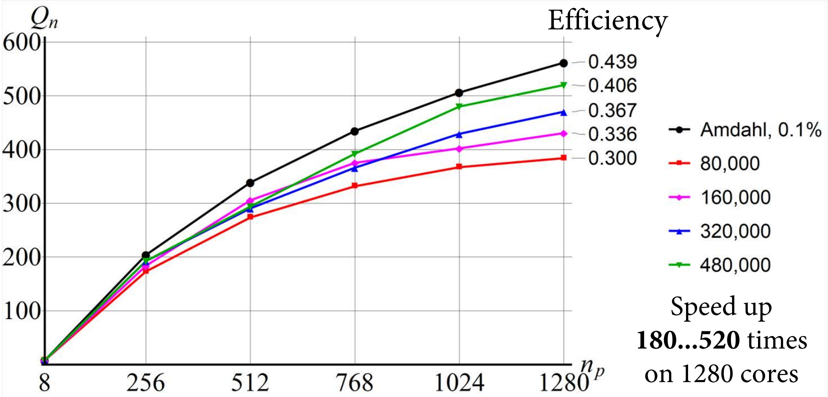
ISPRAN cluster BL2x220c G7, Infiniband QDR
19 nodes: 2 x Intel Xeon X5670 (6 cores), 2.93 GHz



Efficiency

	Absolute	0.1% sequential code
228 cores	$\mathbf{0.62 \ldots 0.75}$	$\mathbf{0.75 \ldots 0.88}$
96 cores	$\mathbf{0.75 \ldots 0.79}$	$\mathbf{0.82 \ldots 0.92}$

National research center 'Kurchatov institute', Infiniband QDR
160 nodes: 2 x Intel Xeon E5343 (4 cores), 2.33 GHz



Efficiency

	Absolute	0.1% sequential code
1280 cores	$\mathbf{0.30\ldots0.41}$	$\mathbf{0.68\ldots0.93}$
256 cores	$\mathbf{0.68\ldots0.76}$	$\mathbf{0.84\ldots0.94}$

- Tests were performed on two GPUs:

	GeForce GTX 970	Tesla K40c
CUDA Cores	1664	2880
Memory	4 Gb (3.5 Gb)	12 Gb

Optimal block size - 128

- Comparison with ISPRAS cluster BL220:

Number of core	1	12	48	G970	96	144	192	228	K40
Prob. '80 000'	1	10.2	39.1	**58.7**	72.3	103.4	127.0	140.1	**158.9**
Prob. '480 000'	1	10.2	38.6	**66.1**	75.8	110.3	145.1	171.5	**162.6**

- Comparison with cluster NRC 'Kurchatov Institute':

Number of cores	1	8	G970	256	512	768	1024	K40	1280
Prob. '80 000'	1	7.2	**127.9**	173.3	273.7	331.7	367.2	**467.6**	468.6
Prob. '480 000'	1	7.3	**190.8**	192.8	294.0	392.3	463.0	**469.0**	565.9

2 MPI nodes (Ethernet) + GeForce970 $\rightarrow$ x 1.6 times
3 MPI nodes (Ethernet) + GeForce970 $\rightarrow$ x 2.2 times

Conclusions

- The VM2D code is developed that allows to simulate 2D viscous incompressible flow using vortex methods.
- The VM2D has the open source code.
- The VM2D structure allows to modify code:
  - expand the range of problems;
  - implement new methods and algorithms.
- It is possible to perform calculations in parallel mode using both CPU and GPU.
- The analysis of the parallelization efficiency show that GPU is the most promising direction of performance improvement.

Acknowledgements

- The research is supported by
  - the Ministry of Education and Science of the Russian Federation (project no. 9.2422.2017/PCh),
  - the Grant of the President of the Russian Federation for Young Ph.D (project no. MK-743.2018.8)
- The authors express their gratitude to
  - Ivannikov Institute for System Programming of the Russian Academy of Sciences,
  - National Research Center 'Kurchatov Institute'

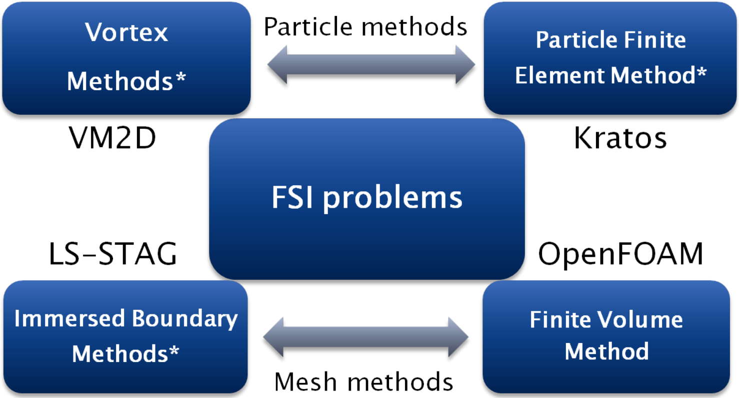
  for providing the computational resources.

Authors

Marchevsky Ilia — iliamarchevsky@mail.ru
PhD, associate Professor at Applied Mathematics dep. in BMSTU,
Senior research scientist in Ivannikov Institute for System Programming of the RAS

Kuzmina Kseniia — kuz-ksen-serg@yandex.ru
PhD student, assistant lecturer at Applied Mathematics dep. in BMSTU,
Research assistant in Ivannikov Institute for System Programming of the RAS

Ryatina Evgeniya — evgeniya.ryatina@yandex.ru
Student at Applied Mathematics dep. in BMSTU,
Intern researcher in Ivannikov Institute for System Programming of the RAS



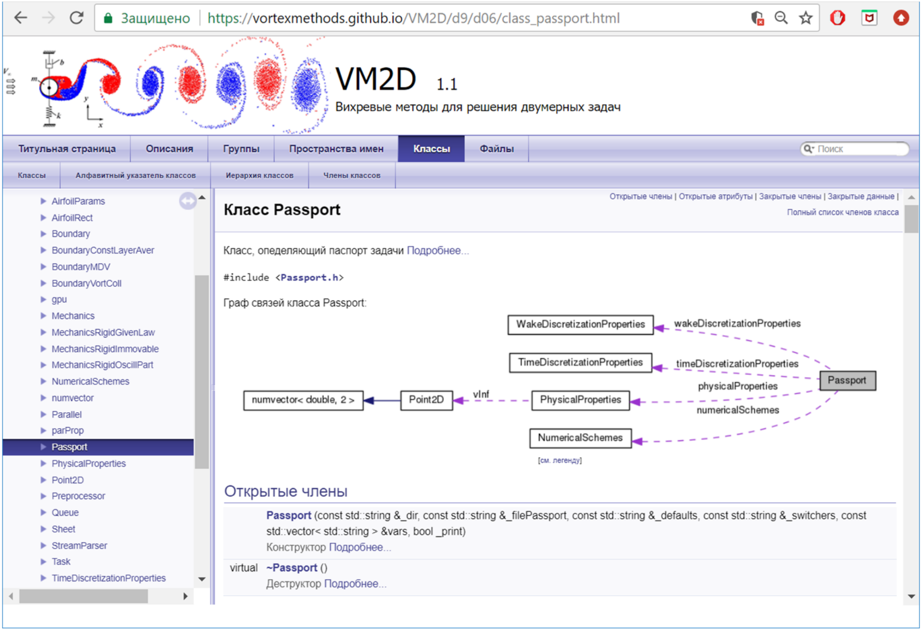
* — investigations on these methods are carried out at the Applied mathematics dep. in BMSTU.

Comparison with other methods and codes

	Vortex Method	OpenFOAM	Kratos	LS-STAG
Time	+	±	+	−
Accuracy	±	+	−	+
Body motion	+	±	±	+
Parallelization	+	+	−*	−*
Auto time step	−	+	−	±
Turbulent flows	−	+	−	+
3D case	−*	+	+	−*

* — this property is available for some special cases or 'under construction'

6. Kraposhin M., Kuzmina K., Marchevsky I., Puzikova V. Study of OpenFOAM Efficiency for Solving Fluid-Structure Interaction Problems // Selected papers of the 11th Workshop, J. Nóbrega, H. Jasak (Eds.). Springer, 2018.


Защищено https://vortexmethods.github.io/VM2D/d9/d06/class_passport.html
VM2D 1.1
Вихревые методы для решения двумерных задач
Титульная страница | Описания | Группы | Пространства имен | Классы | Файлы
Поиск
Классы | Алфавитный указатель классов | Иерархия классов | Члены классов
AirfoilParams
AirfoilRect
Boundary
BoundaryConstLayerAver
BoundaryMDV
BoundaryVortColl
gpu
Mechanics
MechanicsRigidGivenLaw
MechanicsRigidImmovable
MechanicsRigidOscillPart
NumericalSchemes
numvector
Parallel
parProp
Passport
PhysicalProperties
Point2D
Preprocessor
Queue
Sheet
StreamParser
Task
TimeDiscretizationProperties
Открытые члены | Открытые атрибуты | Закрытые члены | Закрытые данные |
Полный список членов класса
Класс Passport
Класс, опеделяющий паспорт задачи Подробнее...
#include <Passport.h>
Граф связей класса Passport:
WakeDiscretizationProperties
wakeDiscretizationProperties
TimeDiscretizationProperties
timeDiscretizationProperties
Passport
physicalProperties
numvector< double, 2 >
Point2D
vInf
PhysicalProperties
numericalSchemes
NumericalSchemes
[см. легенду]
Открытые члены
Passport (const std::string &_dir, const std::string &_filePassport, const std::string &_defaults, const std::string &_switchers, const std::vector< std::string > &vars, bool _print)
Конструктор Подробнее...
virtual ~Passport ()
Деструктор Подробнее...

Abstract classes

Airfoil — airfoil geometry

- **Rectilinear** — airfoil approximation by rectilinear panels
- Curvilinear — airfoil approximation with taking into account the curvilinearity of the airfoil surface line

Boundary — method of the BC satisfaction (IE solving)

- MDV — 'classical' method of discrete vortices
- VortColl — collocation method for IE of the 2-nd kind
- **ConstLayerAver** — piecewise-constant vortex sheet intensity
- LinearLayerAver — discontinuous piecewise-constant vortex sheet intensity approximation
- FEMLayerAver — FEM-type vortex sheet intensity approximation

Mechanics — hydroelastic problems models

- **RigidImmovable** — immovable airfoil moving with given law
- **RigidGivenLaw** — immovable airfoil
- **RigidOscillPartitioned** — viscoelastic constrain; partitioned coupling strategy (applicable for airfoils with density bigger than flow density)
- **RigidOscillStronglyCoupled** — viscoelastic constrain; monolithic coupling strategy (applicable for airfoils with any density)
- **RigidRotateStronglyCoupled** — rotating airfoil, coupling by monolithic approach (applicable for airfoils with any density)

Velocity — method of VE velocities computation

- **BiotSavart** — direct computation (each VE interact with all other VE)
- **BarnesHut** — fast method using multipole expansion method
- **Fourier** — fast method using FFT