Vvflow CFD Suite

Гувернюк С.В., Дынников Я.А., Дынникова Г.Я.,
Малахова Т.В., Сыроватский Д.А., Андронов П.Р.

НИИ механики МГУ, Москва

2 декабря 2018

Что умеет vvflow

Уравнения Навье-Стокса 2D

Жидкость

- вязкая
- несжимаемая

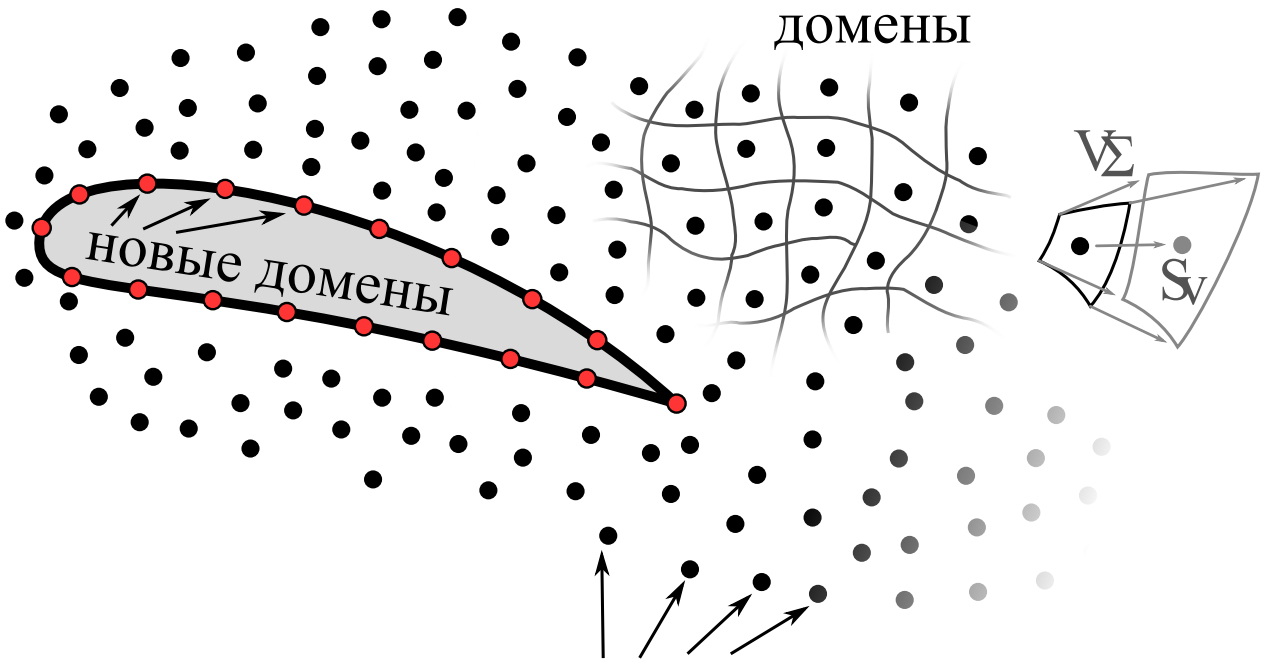
Тела

- много, произвольной формы
- недеформируемые, подвижные
- упруго связаны друг с другом
- соударяющиеся

Результат

- Картина течения
- Движение тел
- Силы (гидродинамические, реакции опоры)
- Эпюры (давления, трения)

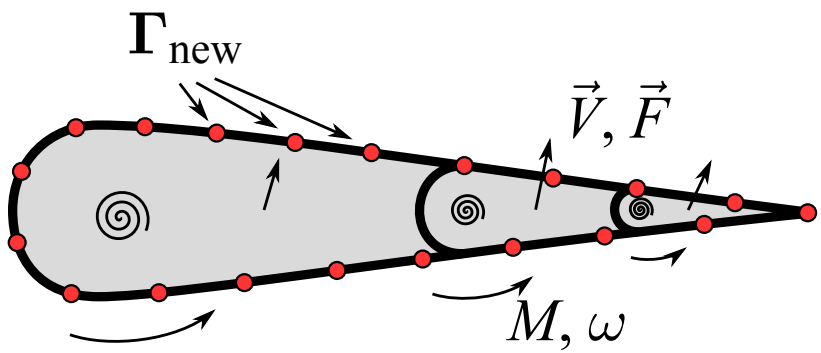
Метод вязких вихревых доменов



$$\frac{\partial \vec{V}}{\partial t} = \vec{V} \times^{\rightarrow} + \nu \nabla^2 V - \frac{1}{\rho} \nabla \left(p + \frac{V^2}{2} \right), \nabla \cdot \vec{V} = 0$$

$$\frac{\partial}{\partial t} = -\nabla \cdot \left(\vec{V}_\Sigma \right), \boxed{\vec{V}_\Sigma = \vec{V} - \nu \overset{\nabla}{—}} \Rightarrow \int_{S_v} ds = \text{const}$$

Система линейных алгебраических уравнений



Неизвестные:

γ_i — циркуляции новых доменов

$\vec{F}_{\text{hydro}}, M_{\text{hydro}}$ — г.-д. силы

$\vec{V}, \omega$ — скорости тел

$\vec{F}_O, M_O$ — реакции опоры

Уравнения:

Условие прилипания:

$$\vec{n}_s \left(\sum \vec{K} \times \Delta\mathbf{\Gamma}_s + \sum \vec{K} \times \mathbf{\Gamma}_{\text{free}} + \sum \vec{K} \times 2\omega\Delta\vec{l}_s \right) = \vec{V}_s \vec{n}_s$$

Выражения гидродинамических сил:

$$\vec{F}_{\text{hydro}} = \frac{1}{\Delta t} \sum_{s=1}^{N_s} [\Delta\mathbf{\Gamma}_s \times \vec{r}_s] - \frac{1}{2} \sum_{s=1}^{N_s} [\vec{e}_z \times \Delta\vec{l}_s] \cdot V_{\text{old}}^2$$

$$M_{\text{hydro}} = \frac{1}{2\Delta t} \sum_{s=1}^{N_s} \Delta\mathbf{\Gamma}_s \cdot \vec{r}_*^{\,2} - \frac{1}{2} \sum_{s=1}^{N_s} (\vec{r}_* \cdot \Delta\vec{l}_s) \cdot V_{\text{old}}^2$$

Второй закон Ньютона:

$$\vec{F}_{\text{hydro}} + \vec{F}_O + \vec{F}_{O\ \text{child}} = m\dot{\vec{V}}$$

$$M_{\text{hydro}} + M_O + M_{O\ \text{child}} + [\Delta\vec{R}_O \times \vec{F}_{O\ \text{child}}] = J\dot{\omega}$$

Кинематика: $\vec{V} = \vec{V}_{\text{root}} + \omega_{\text{root}} [\vec{e}_z \times \Delta\vec{R}_O]$

Закон гука: $M_O = k_\alpha \Delta\alpha$

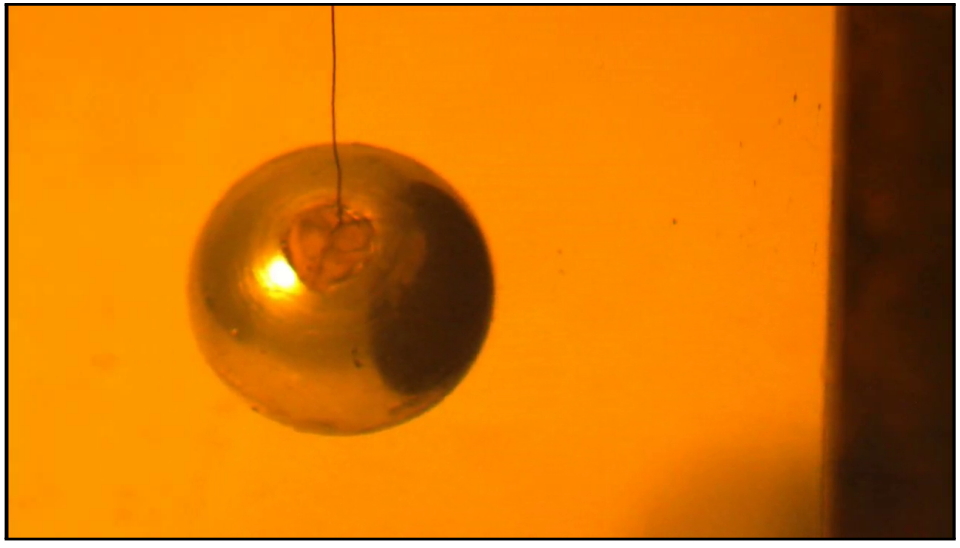
Планы развития (2015)

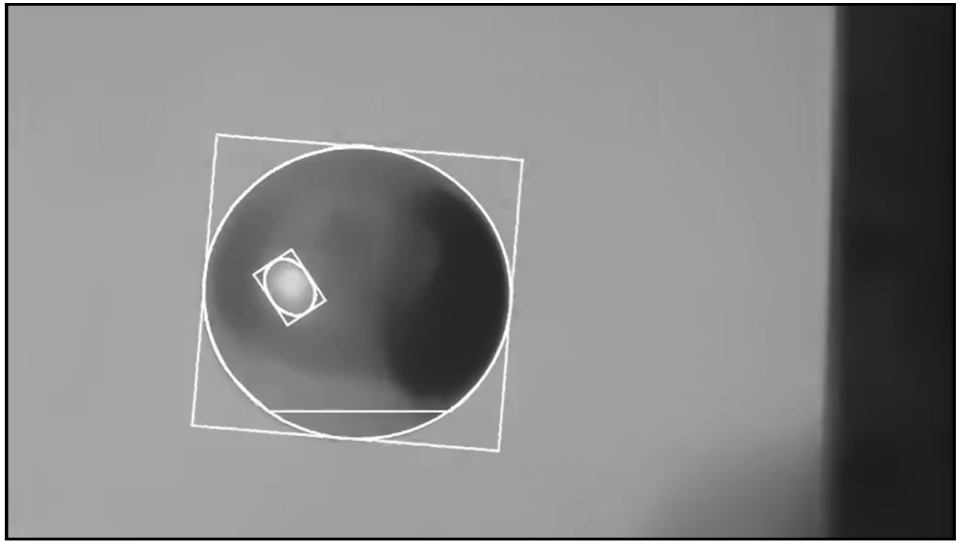
- [±] Оптимизация кода, эффективность распараллеливания
- [−] Графический интерфейс
- [+] Соударение тел

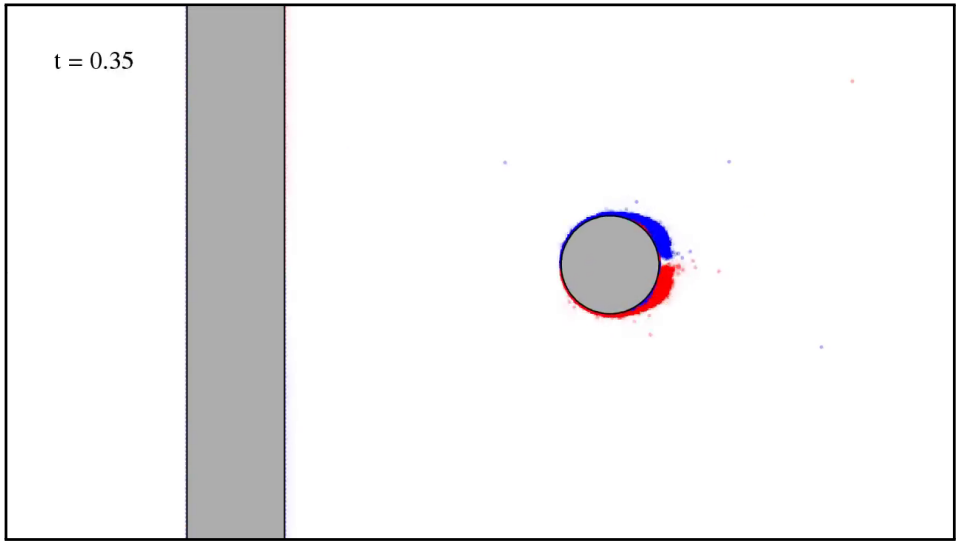
Планы развития (2015)

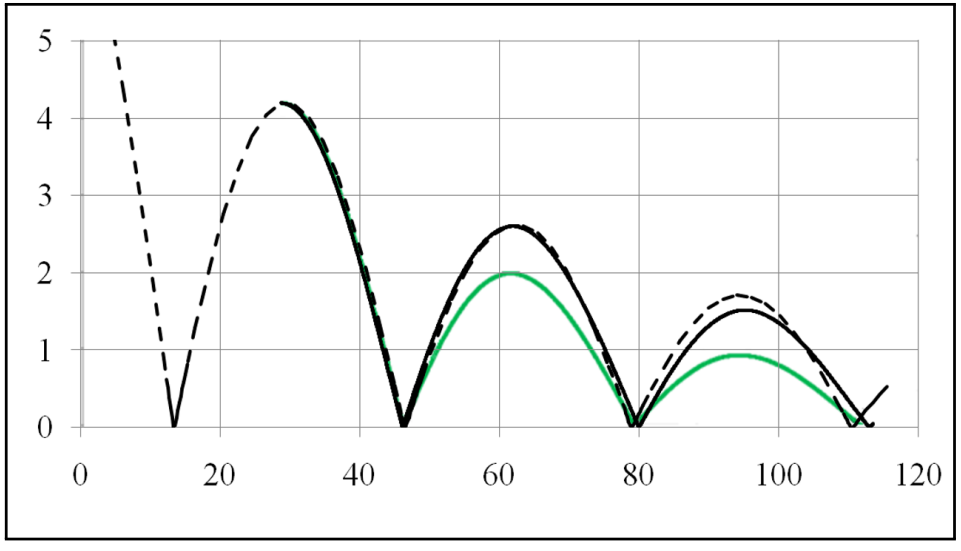
- [±] Оптимизация кода, эффективность распараллеливания
- [−] Графический интерфейс
- [+] **Соударение тел**

Сопряженная задача с соударением тел









Дюраль в воде, $\rho_b/\rho_0 = 2.87, k = 0.7$

Динамика тел при соударении

Гипотеза 1:

$$V(t_c + 0) = -k \cdot V(t_c - 0)$$

Гипотеза 2:

$$p_c : V(t_c + 0) = 0, \quad p_c \neq p_b$$

$$p(t_c + 0) = p(t_c - 0) + (1 + k) \cdot p_c$$

Поперечные автоколебания цилиндра, сильно загромождающего поток вязкой жидкости в плоском канале / С. В. Гувернюк, Г. Я. Дынникова, Я. А. Дынников, П. Р. Андронов // Материалы XXIII Международной конференции «Нелинейные задачи теории гидродинамической устойчивости и турбулентность» 25 февраля – 04 марта 2018 г. – МАКС Пресс Москва, 2018. – С. 108–117.

Планы развития (2015)

- [±] **Оптимизация кода**, ~~эффективность распараллеливания~~
- [−] Графический интерфейс
- [+] Соударение тел

Vvflow CFD Suite

Прикладные программы (C++):

- `vvcompose` — препроцессор;
- `vvflow` — проведение расчёта;
- `vvxtract` — экспорт результатов;
- `vvplot` — визуализация данных;

Тестирование кода:

- `googletest` — C++ юнит тесты;
- `pytest` — интеграционные тесты;

Continuous integration, continuos delivery:

- Сборка проекта в `docker`;
- `bitbucket pipelines` — автосборка;
- `packagecloud.io` — deb repository;

Тестирование кода: pytest

/opt/vvflow/pytest/refframe/test_refframe.py • - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help

test_refframe.py

```
#!/usr/bin/env python3

def test(env):
    # ref_o
    env.vvcompose('compose_ref_o.lua', 'ref_o.h5')
    env.vvflow('ref_o.h5')
    env.vvplot('results_ref_o/000001.h5', 'ref_o.tar',
        args='-x -1.5,1.5 -B -S --ref-S o --tar')
    sf_o = env.run(['tar', '-xOf', 'ref_o.tar', 'map_streamfunction']).stdout
    sf_o = struct.unpack('f'*int(len(sf_o)/4), sf_o)

    # ref_b
    env.vvcompose('compose_ref_b.lua', 'ref_b.h5')
    env.vvflow('ref_b.h5')
    ret = env.vvplot('results_ref_b/000001.h5', 'ref_b.tar',
        args='-x -1.5,1.5 -B -S --ref-S b --tar')
    sf_b = env.run(['tar', '-xOf', 'ref_b.tar', 'map_streamfunction']).stdout
    sf_b = struct.unpack('f'*int(len(sf_b)/4), sf_b)

    #compare
    assert len(sf_o) == len(sf_b)
    spread_o = numpy.percentile(sf_o, 95) - numpy.percentile(sf_o, 5)
    spread_b = numpy.percentile(sf_o, 95) - numpy.percentile(sf_o, 5)
    abserror = numpy.absolute(numpy.subtract(sf_o, sf_b))
    e = 2 * numpy.percentile(abserror, 80) / (spread_o + spread_b)

    logging.info('abserror(80%) = {}'.format(numpy.percentile(abserror, 80)))
    logging.info('e = {}'.format(e))
    assert e < 0.001

```

Line 30, Column 1 Spaces: 4 Python

Планы развития (2015)

- [±] Оптимизация кода, ~~эффективность распараллеливания~~
- [−] ~~Графический интерфейс~~
- [+] **Консольный интерфейс**
- [+] Соударение тел

Вместо GUI - скрипты Lua

/opt/vvflow/pytest/vvplot/compose_plate.lua - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help

compose_plate.lua

```lua
#!/usr/bin/env vvcompose

S.inf_vx = "0"
S.re = 600
S.dt = 0.005
S.dt_save = 0.005
S.finish = 0.005

plate = gen_plate{R1=0.04, R2=0.04, L=1, N=400}
plate.label = "plate"
plate:move_r({-0.5, 0})
plate.holder_pos = {0, 0, 0}
S.body_list:insert(plate)

N = 500
for i = 1, N do
    rad = 0.2 * i/N * math.sin(i)
    phi = i/N * 10*math.pi
    x = 0.25 + rad*math.cos(phi)
    y = 0.25 + rad*math.sin(phi)
    S.vort_list:append({ x,  y, 1/N})
    S.vort_list:append({-x, -y, 1/N})
end

S.caption = "plate"
S:save(arg[1])
```

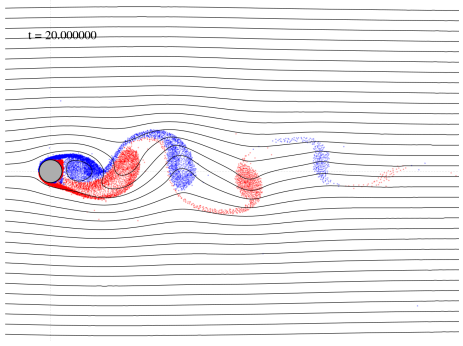
compose.lua

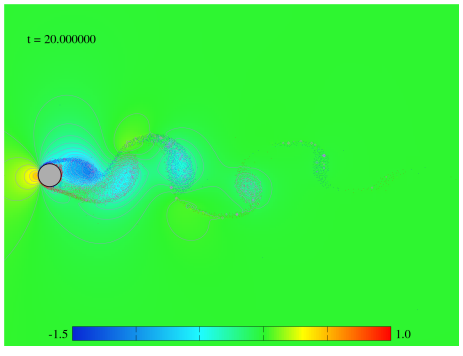
```lua
#!/usr/bin/env vvcompose

A_120 = math.asin(120/1450)

cyl = gen_cylinder{R=0.015, N=600}
cyl.label = "cyl"
cyl.holder_pos.r.y = 1.450
cyl:move_o(A_120)
cyl.holder_pos.o = 0
cyl.delta_pos.o = A_120
cyl.collision_min.o = 0
cyl.density = 7.4 -- steel
cyl.bounce = 0.7
cyl.spring_const.o = 0
S.body_list:insert(cyl)

wall = gen_plate{R1=0.015, R2=0.015, L=0.35, N=600}
wall.label = "wall"
wall:move_d(90)
wall:move_r({-0.015, -0.35/2})
S.body_list:insert(wall)

S.re = 1e6
S.dt = 0.001
S.dt_save = 0.01
S.gravity.y = -9.8
S.caption = "bounce_k0.7"
S.finish = 10

S:save("bounce.h5")
```

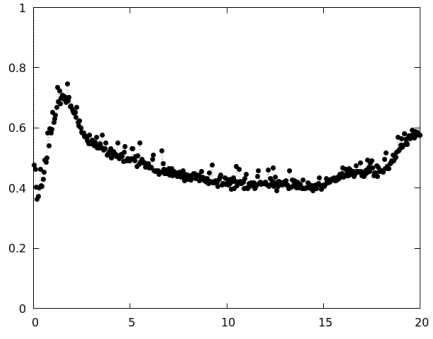
Line 27, Column 1 Tab Size: 4 Lua

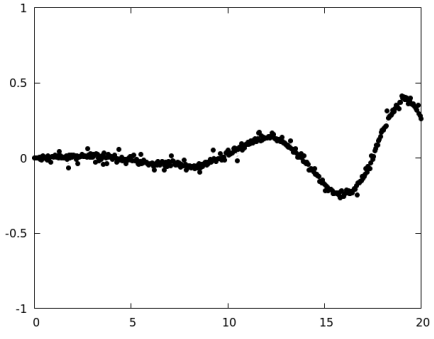
Пример запуска

```
rosik@rlap: ~
rosik@rlap:~$ sudo docker run -it --name ex vvflow/vvflow
[sudo] password for rosik:
root@f99afac2b99e:~# cd example/
root@f99afac2b99e:~/example# make
vvcompose cylinder.lua
-- Running CFD simulation up to t=20
-- It may take about a minute or two
vvflow --progress ./re600_n350.h5
t=20.05         N=6633
vvplot results_re600_n350/000400.h5 bvs.png -x -2,18 -BVS
vvplot results_re600_n350/000400.h5 bvg.png -x -2,18 -B --V 0 --G 4
vvplot results_re600_n350/000400.h5 bvp.png -x -2,18 -B --V 0 -P --res-hi 256
vvxtract stepdata_re600_n350.h5 time body00/force_holder | gpquick -p -u 1:2 -y 0 1 > cx.png
vvxtract stepdata_re600_n350.h5 time body00/force_holder | gpquick -p -u 1:3 -y -1 1 > cy.png
root@f99afac2b99e:~/example# exit
rosik@rlap:~$ sudo docker cp ex:/root/example ./
rosik@rlap:~$
```









Спасибо за внимание