

Firedrake

Automated simulation right from the equations

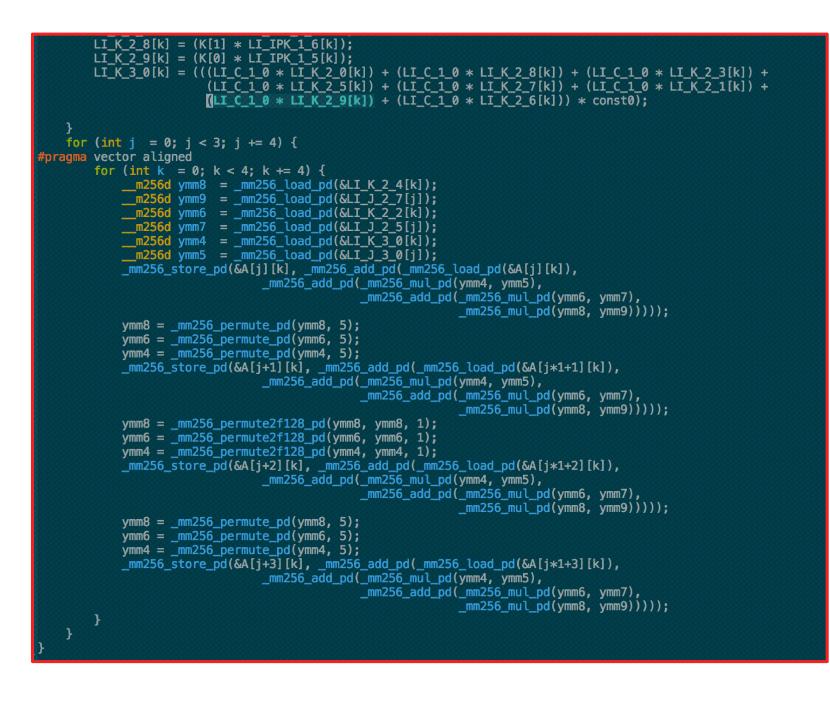
```
\phi^{n+1/2} = \phi^n - \frac{\Delta t}{2} p^n
            p^{n+1} = p^n + \frac{\int_{\Omega} \nabla \phi^{n+1/2} \cdot \nabla v \, dx}{\int_{\Omega} v \, dx} \quad \forall v \in V
             \phi^{n+1} = \phi^{n+1/2} - \frac{\Delta t}{2} p^{n+1}
where
           \nabla \phi \cdot n = 0 on \Gamma_N
                     p = \sin(10\pi t) on \Gamma_D
```

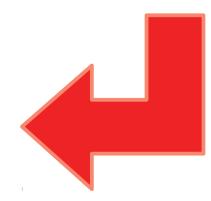
Finite element simulations are specified in Python using high-level

mathematical syntax.

Firedrake allows mathematicians, scientists and engineers to work at the level of the equations.

```
from firedrake import *
mesh = Mesh("wave_tank.msh")
V = FunctionSpace(mesh, 'Lagrange', 1)
p = Function(V, name="p")
phi = Function(V, name="phi")
u = TrialFunction(V)
v = TestFunction(V)
p_{in} = Constant(0.0)
bc = DirichletBC(V, p_in, 1)
T = 10.
dt = 0.001
t = 0
while t <= T:
    p_in.assign(sin(2*pi*5*t))
    phi -= dt / 2 * p
    p += assemble(dt * inner(grad(v), grad(phi))*dx) \
         / assemble(v*dx)
    bc.apply(p)
    phi -= dt / 2 * p
    t += dt
```

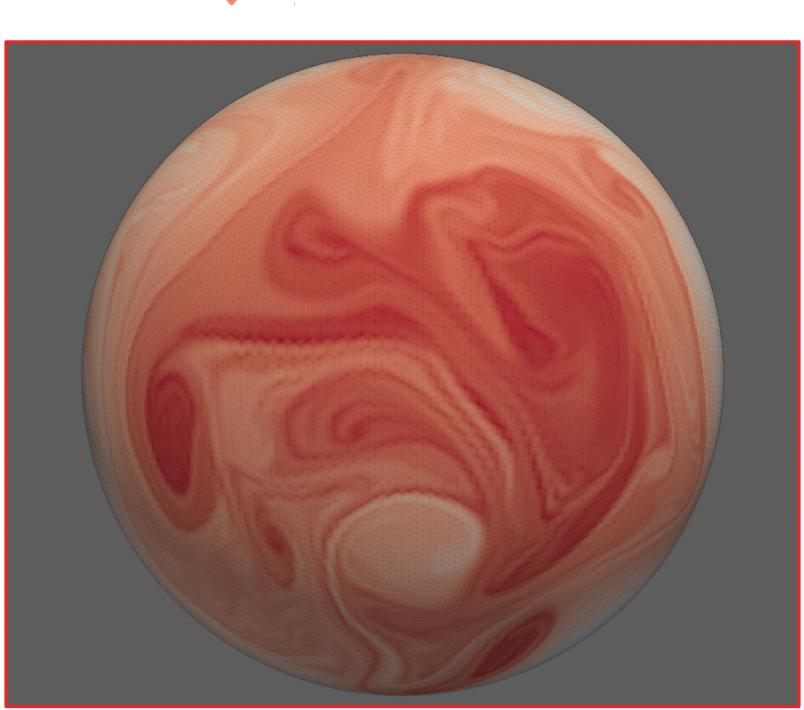




High performance vectorised parallel code is generated, compiled and executed automatically on the fly as the simulation runs.



- Limitless flexibility in specifying equations.
- A huge range of sophisticated finite element spaces.
- World-class linear and nonlinear solvers built in.
- Seamless scaling from laptop to supercomputer.
- Specialist support for high aspect ratio domains.





prism.ac.uk firedrakeproject.org