

Firedrake: automated simulation right from the equations.

Scientists and engineers express simulation challenges in concise, readable, mathematics. Traditionally, this was just the first step in a long and arduous software engineering effort to produce correct, efficient and parallel simulations. With Firedrake, a scientist or engineer can type a differential equation in weak form and have a highly efficient parallel simulation generated and executed automatically.

A mathematical problem statement...

Find $u \in P^4(\Omega)$ such that:

$$\int_{\Omega} uv + \nabla u \cdot \nabla v - (1 + 8\pi^2) \cos(2\pi x) \cos(2\pi y) \, dx = 0 \quad \forall v \in P^4(\Omega)$$

...can be expressed as a few lines of typed mathematics ...

```
from firedrake import *
Omega = Mesh("meshfile.msh")
P4 = FunctionSpace(Omega, "CG", 4)
v = TestFunction(P4)
u = Function(P4)
f = Function(P4)
f.interpolate(Expression("(1+8*pi*pi)*cos(x[0]*pi*2)*cos(x[1]*pi*2)"))
solve(u*v + (dot(grad(u), grad(v)) - f*v)*dx == 0, u)
```

...and the solve call instructs Firedrake to automatically generate the simulation code and execute it.

Key features

- Support for a limitless range of differential equations.
- A wide range of finite element spaces, including structure-preserving $H(\text{Div})$ and $H(\text{Curl})$ element families.
- Specialised support for high aspect ratio domains for critical simulation fields such as ocean and atmosphere flows.
- Automated support for MPI and hybrid MPI/OpenMP parallelism. A subset of Firedrake functionality is available for GPU.
- Sophisticated inner loop optimisations and automated exploitation of the vector features of latest generation CPUs.

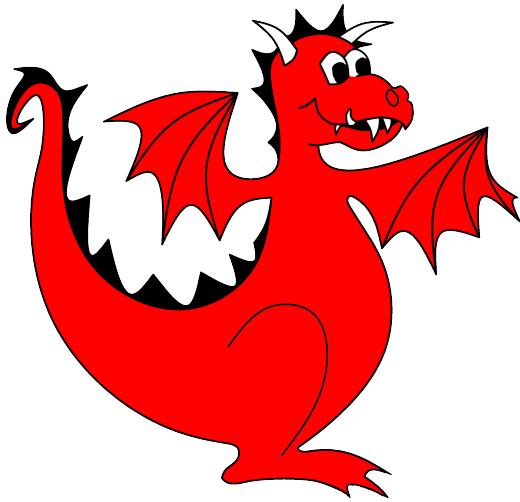
- High level access to the complete suite of world class linear and non-linear solvers provided by the PETSc package.
- High level, maths based, interactive programming environment which maximised the productivity of the scientist or engineer and minimises debugging.

Defeating multidisciplinary complexity by separating concerns.

Advancing simulation science requires complex scenarios to be simulated using advanced numerics on sophisticated parallel hardware. The code generation approach in Firedrake enables computer scientists to advance the compiler and parallelisation technology, independent of the numerics. Numericists to develop sophisticated discretisations independent of their implementation, and application specialists to deploy these numerics efficiently and effectively. Firedrake is the glue that enables experts to advance their own field, while directly deploying the outcome across disciplines.

Some branding graphics

The Firedrake logo:



The banner from our website (<http://firedrakeproject.org>):



Firedrake