. describe type of flow they simulate  
. the turbulence models they use  
. spatial and temporal discretisation algorithms they employ  
. practical application of the code  
  
PyFR is an open source project written to solve advection-diffusion time problems including compressible flow, unsteady flow and separation problems and both 2D and 3D. It uses Euler and Navier Stokes as the governing flow equations.

Les and dns – much finer scales

Large Eddy Simulation - direct numerical simulation

Divide domain into cells, assume solution is constant

Create polynomial approximation of solution – continuous in each element, can increase order of polynomial to make more accurate, but more computational cost

Solution is discontinuous across elements, flux reconstruction scheme is continuous

Polynomial approximation of elements, discontinuous across elements, high order and continuous in flux reconstruction scheme

Flux reconstruction allows partial temporal locality - most of data to make computation is in element, only need flux on neighbouring elements to compute equations. We need high flops per bite

No rans model

Explicit and implicit schemes

One code can run on multiple types of system

Python layer for high level functions

2 kernels – one for matrix (BLAS),

pointwise kernel – computes flux at solution points, written in templating syntax, at runtime converted into hardware specific accelerated functions

Low pressure turbines, large scale simulations as easily scalable, just add more gpus (due to high level of parallelism)

BDF2 dual time step (second-order backward difference formula)

Good for unstructured fluids

Works on multiple hardwares!