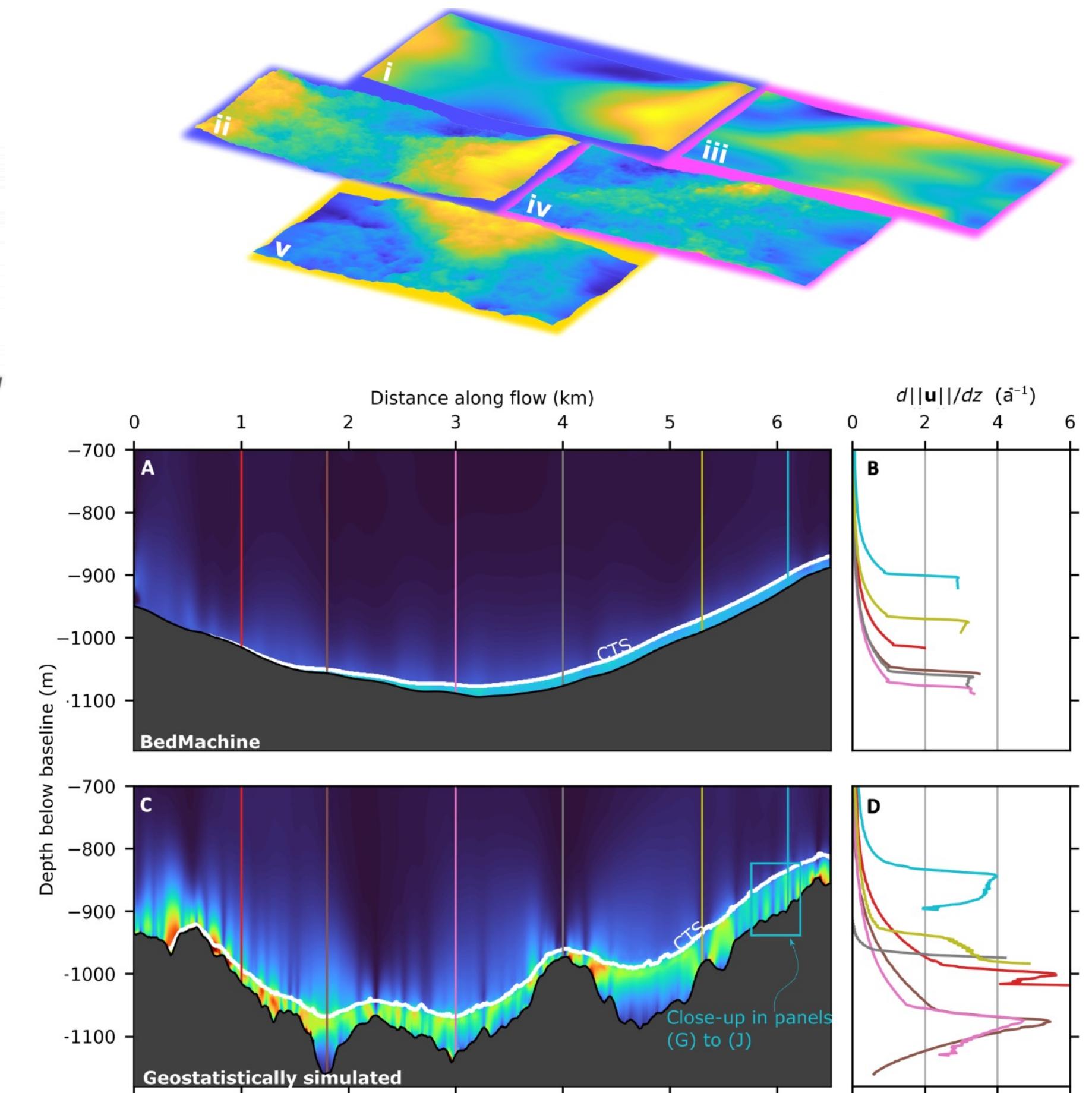
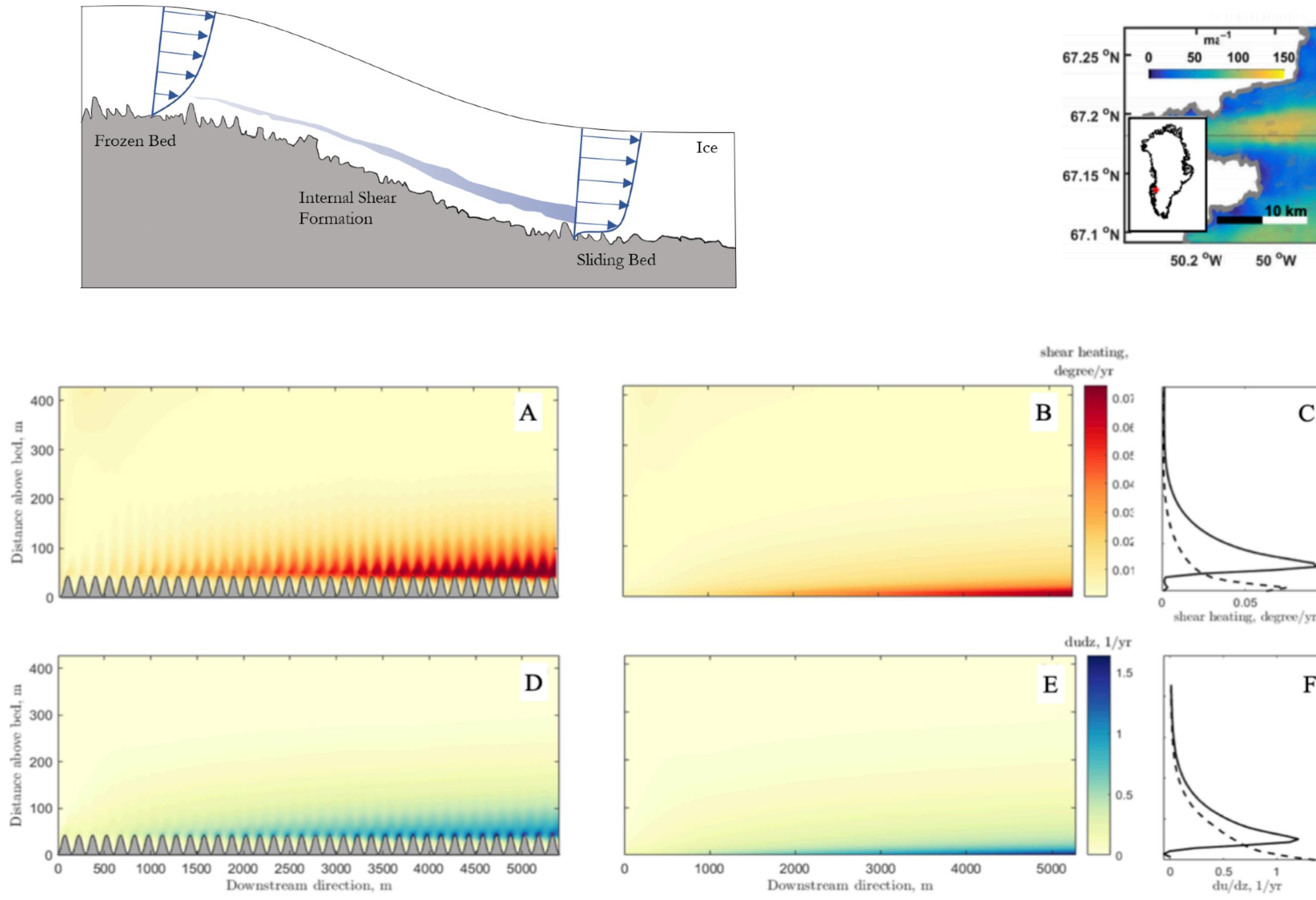


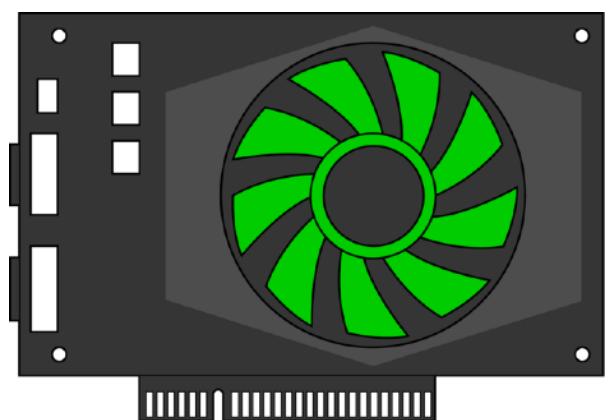
Understanding ice motion and localisation of flow

- Physical processes (THM)
- Influence on surface flow speed (sensitivity, inversions)

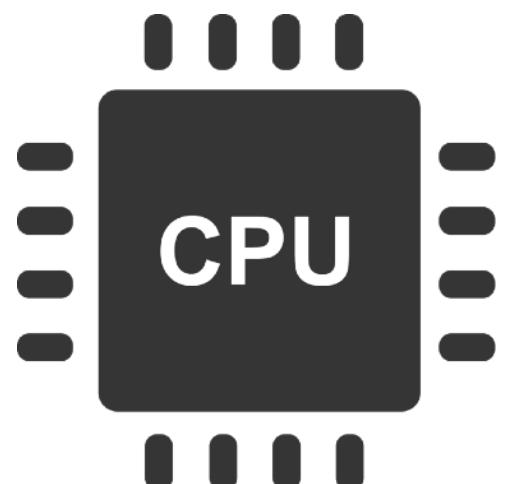


Challenges

- Inherently 3D problems (forward and inverse)
- Multi-physics (mechanical, thermal, hydrological)
- High resolution needed to capture flow localisation
- Requires a supercomputing approach
- Majority of top500 supercomputers are GPU-powered

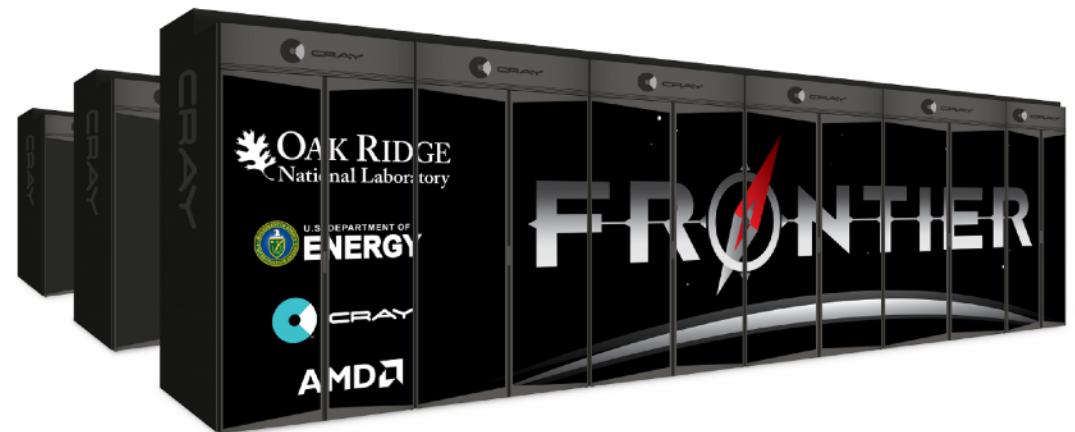
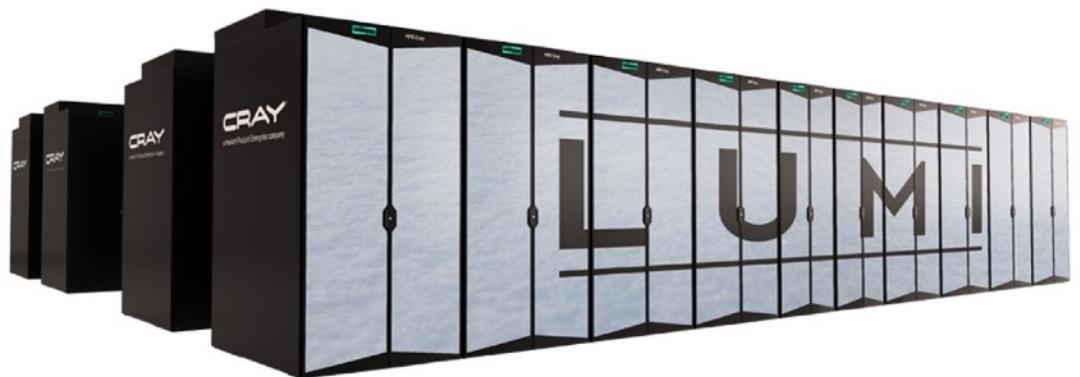


1000 GB/s



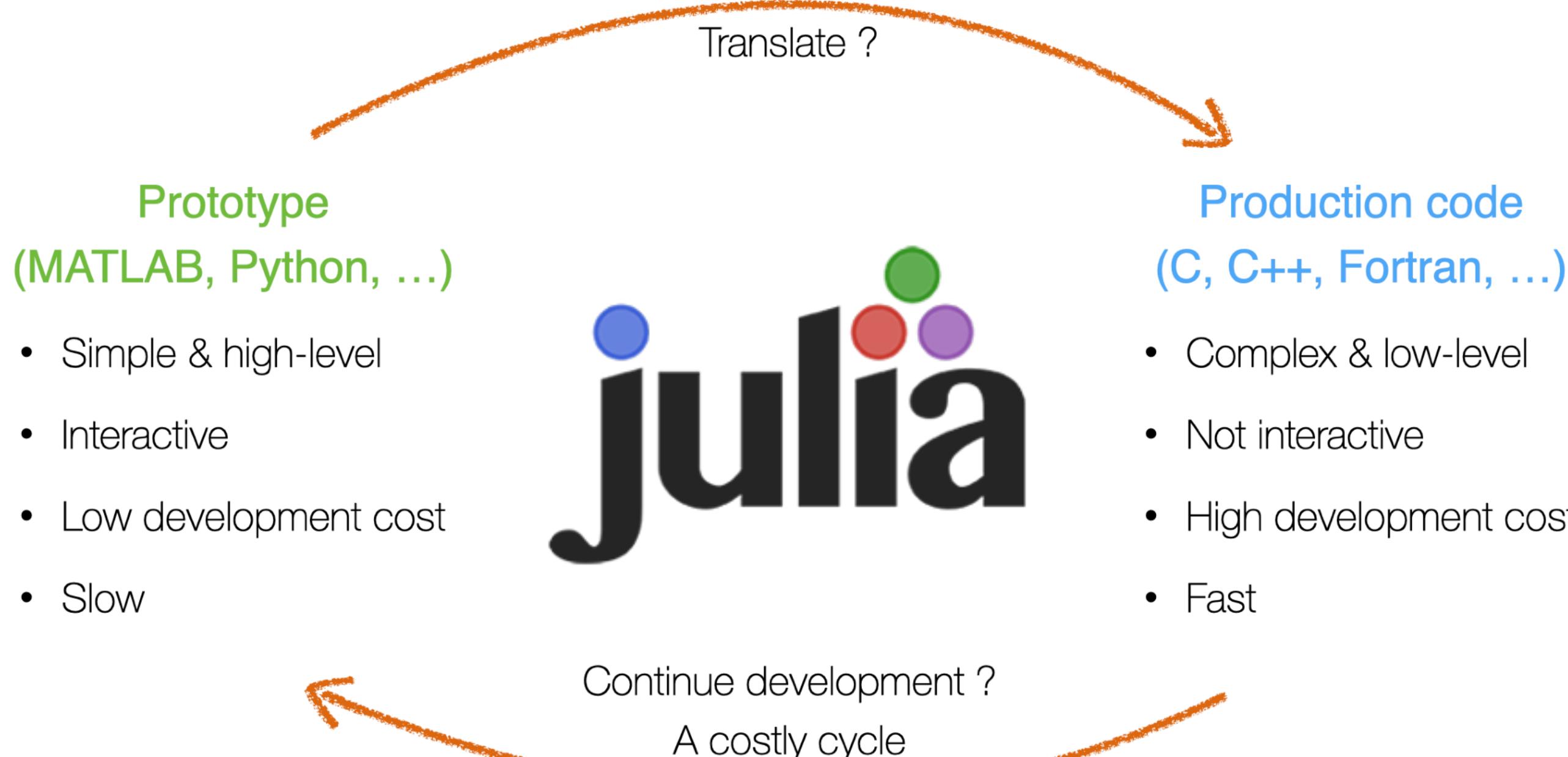
100 GB/s

Memory bandwidth



The 3 P challenge

- Performance fast: (JAOT) compiled language
- Portability LLVM-based compiler
- Productivity simple & high-level interactive



```
1 using ImplicitGlobalGrid
2 using ParallelStencil
3 using ParallelStencil.FiniteDifferences3D
4 @init_parallel_stencil(CUDA, Float64, 3)
5
6 @parallel function step!(T2,T,Ci, lam, dt, dx, dy, dz)
7     @inn(T2) = @inn(T) + dt*((
8         lam*@inn(Ci)*(@d2_xi(T)/dx^2 +
9             @d2_yi(T)/dy^2 +
10            @d2_zi(T)/dz^2 ) )
11    return
12 end
13
14 function diffusion3D()
15     # Physics
16     lam      = 1.0          # Thermal conductivity
17     c0       = 2.0          # Heat capacity
18     lx=ly=lz = 1.0         # Domain length x|y|z
19
20     # Numerics
21     nx=ny=nz = 512         # Nb gridpoints x|y|z
22     nt       = 100          # Nb time steps
23     me,      = init_global_grid(nx, ny, nz)
24     dx       = lx/(nx_g()-1) # Space step in x
25     dy       = ly/(ny_g()-1) # Space step in y
26     dz       = lz/(nz_g()-1) # Space step in z
27
28     # Initial conditions
29     T = @ones(nx,ny,nz).*1.7 # Temperature
30     T2 = copy(T)           # Temperature (2nd)
31     Ci = @ones(nx,ny,nz)./c0 # 1/Heat capacity
32
33     # Time loop
34     dt = min(dx^2,dy^2,dz^2)/lam/maximum(Ci)/6.1
35     for it = 1:nt
36         @hide_communication (16, 2, 2) begin
37             @parallel step!(T2,T,Ci, lam, dt, dx, dy, dz)
38             update_halo!(T2)
39         end
40         T, T2 = T2, T
41     end
42
43     finalize_global_grid()
44 end
45
46 diffusion3D()
```