PARALLEL CODE: FLOW − Physics.F : propogate

Add existing levels of Adams−Bashforth update:

```
field%rhs = field%current + C1*field%force + C2*field%force2
```

Copy forces for next AB update:

```
field%force2 = field%force
```

Calculate forces and rhs's:

Calculate Coriolis force −> mass_flux%force

Calculate div(rho u) −> density%force

Calculate advection terms in temperature equation −> temperature%force

Calculate div(kappa grad T) :

*gamma*Ck/rho_bar kappa (del^2 T)* −> temperature%rhs
*gamma*Ck/rho_bar d/dz(kappa) d/dz(T)* −> temperature%force

Calculate div (u) and viscous heating terms

u_i_spatial −> u_i_spectral
d/dx(u) + d/dy(v) + d/dz(w) −> div_u
temperature%force = temperature%force + visc_heating

Calculate diffusion terms

mass_flux%rhs = mass_flux%rhs + del^2(u_i) + 1/3 grad(div_u)

Calculate compressional heating T div(u) −> temperature%force

Calculate nonlinear terms in configuration space: Cache! (E−registers)

nl1 −− rho u u + pressure + scaling(Bx^2 − B^2/2)*
nl2 −− rho u v + + scaling(Bx*By)*
nl3 −− rho u w + + scaling(Bx*Bz)*
nl4 −− rho v v + pressure + scaling(By^2 − B^2/2)*
nl5 −− rho v w + + scaling(By*Bz)*
nl6 −− rho w w + pressure + scaling(Bz^2 − B^2/2)*
*nl8 −− −(u*By − v*Bx)*
*nl9 −− v*Bz − w*By*
*nl10 −− w*Bx − u*Bz*

Calculate advection terms −> mass_flux%force

Calculate magnetic diffusion and Ohmic heating :

magnetic diffusion −> tor/poloidal%rhs
temperature%force = temperature%force + ohmic_heating

Add buoyancy; apply forcing −> mass_flux%force

Update temperature, density and momenta

```
field%current = field%rhs + C0*field%force
```

Calculate advection of magnetic fields −> tor/poloidal%force

Update magnetic fields:

```
tor/poloidal%current = tor/poloidal%rhs + cc0*tor/poloidal%force
```

Implicit solve for the temperature −> temperature%current

Fix boundary conditions and create configuration space fields