REGK Summary

-load in all the variables, equilibria, arrays

- calculate the initial timestep based on CFL as well as the initial hyper coefficients

-begin timeloop

- repeat, divergent, noinc all set to false to begin with

- if neither repeat or divergent are true,

- set p\_count = 0

- if first(first timestep? Do a bunch of ‘convol’ calls with fields)

- set first false

- do first calculation (funcne\_i, funcAkpar\_i (inc gs?), funcg2,func gm)

- call semi\_imp\_op () to updated SI\_oper

- Begin predictor step (eqs 48-51), get all the “star” arrays

- call a bunch of convols for the g\_stars

- call PHI\_POT (returns phi\_k from ne\_k)

- call a bunch of convols for other fields (not g)

- call funcAKpar\_i again with the star values

- begin corrector “p-loop”

- equations 52-55

- calculate the relative error **(THIS IS WHERE SI\_oper is used!!)**

- compare to “epsilon” , if smaller, exit the p-loop

- if the p\_iter has hit pmax, turn on “repeat=true” so that the timestep and hypercoeffs are recalculated from the beginning

-update variables with the new values from the p-loop!

- **Next, do 3D stuff (NOT NECESSARY FOR 2D)**

- Call a bunch of diagnostics

- necessary to ensure the timestep is small enough based on the calculated flows

- calc the CFL condition again (CFL\_frac\*CFL\_flow = x, why “x”?)

- Save datafiles if at required time iteration

- Calc new timestep based on the new CFL condition (called “x” here for some reason)

- update the hyper coeffs

That’s it!

\*\* see appendicies about the operator splitting and numerical scheme. Seems to be custom semi-implicit (Crank-nicholson like) scheme

\*\* Convol just takes the first function in kspace and returns partial\_x and partial\_y in real space