Tool Kit—Phase I Computational Model

- **Motivation**: Exploit existing homogeneity of codes:

```
Read (initial) state
for NUM_STEPS
  for NUM_UPDATES & maybe until convergence
    U (Grid Function(s)) -> Grid Function(s)
  end for
end for
Write (final) state
```

- **Expedite:**
  - standardization/collaboration
  - development speed/correctness
  - efficient use of computational resources
  - uniform/automatic access to analysis resources
  - access to Berger & Oliger mesh refinement

- **Important Question:**

  Can it be expected that this simple structure can persist across the development of more complex problem formulations and more powerful solution methods?

  Need to define expected models of programming.

1-d RPNL example: (Robert Marsa, UT Austin)

$$\phi_{tt} = \phi_{xx} \quad \text{on} \quad 0 \le x \le 1,\ \ t > 0 \qquad \phi(0,t) = \phi(1,t) = 0$$

```
output C

parameter int N := 256
parameter float xmin := 0.0
parameter float xmax := 1.0

coordinates t,x
uniform grid g1 [0:N] xmin:xmax

float Phi on g1 at -1,0,1
float Pi on g1 at -1,0,1

operator D_LF(f,x) := (<0>f[1] - <0>f[-1])/(2*dx)
operator D_LF(f,t) := (<1>f[0] - <-1>f[0])/(2*dt)
operator D_FW(f,x) :=
    (-3*<1>f[0] + 4*<1>f[1] - <1>f[2])/(2*dx)
operator D_BW(f,x) :=
    (3*<1>f[0] - 4*<1>f[-1] + <1>f[-2])/(2*dx)

residual Phi[0:0] := D_FW(Phi,x)
residual Phi[1:N-2] := D_LF(Phi,t) - D_LF(Pi,x)
residual Phi[N-1:N-1] := D_BW(Phi,x)

residual Pi[0:0] := D_LF(Pi,t)
residual Pi[1:N-2] := D_LF(Pi,t) - D_LF(Phi,x)
residual Pi[N-1:N-1] := D_LF(Pi,t)

update Phi
update Pi
```

Existing 3D Components (Sryacuse/Texas):

- F90 Data Structure

- BH initial data generator, apparent horizon locator

- f77 / C (linear storage) GF3 .hdf readers/writers

- GF3 .hdf reader for Explorer

- Prototype GF3 direct-to-server mechanism (DTM)
  (Nair, HPCF, UT Austin)