

Comments on Programming

the following are some good practices, but do not replace a good programming class

- use functions/subroutines to structure your program

```
call initialConditions(u,v,phi)
do while (t < tend)
  call calcTimeStep(u,v,dt)
  call predict(u,v)
  call poisson(u,v,phi)
  call correct(u,v,phi)
  t = t+dt
end
```

- use variables for mesh sizes: M (and N if needed)
- preallocate all vectors and arrays

```
M = 40;
phi = zeros(M+2,M+2);
u = zeros(M+1,M+2);
v = zeros(M+2,M+1);
```

Comments on Programming

the following are some good practices, but do not replace a good programming class

- in Matlab, arrays are matrices, where the first index is row# and the second column#
- can use Matlab functions `transpose()` and `flipud()` for plotting
- run times can be long, so to give peace of mind that your code is still running properly, output some diagnostics after a fixed number of time steps, for example, min/max velocities, etc.
- use Matlab's build in **debugger** to locate bugs!!!!
- test individual pieces of your code = subroutines/functions separately