MTH 655 CLASS SUMMARY (almost)

- min G(U)
  - U solves a nonlinear PDE:
    - F(U)=0
    - discretize:
      - F_h(U_h)=0
    - apply Newton's method
      - J s =- F_h
    - solve linear system
      - » iterative solver (multigrid,gmres)
      - » parallel preconditioner
      - » use domain decomposition (multilevel)
- **http://www.math.oregonstate.edu/~mpesz/teaching/655**

LAB 1

```
function [x,it] = newton(x,tol)
error = abs(f(x)); it = 0;
while abs(error) > tol
    fx = f(x);
    dfx = df (x);
    x = x - fx/dfx;
    it = it + 1;
    error = abs(f(x));
end
```

no need for another abs(f(x))

use relative and absolute conv. criteria [Kelley]

what if dfx =0 ?

LAB 1

1. solve sin(x)=1 *finite set of numbers*
2. rates of convergence:
    1. c) is Newton's method from 3)
    2. d) is secant method from 3)
3. Newton's algorithm for x^2-1=0

quadratic convergence only close to exact root !!!!!

6. EXTRA: stability of numerical derivatives
    - most found *critical h \approx 10^{-7}*

LAB 2

- use cygwin and Unix to operate on files
- compile and run
- correct

```
norm = 0.D0
do i = 1,n
  norm = norm + (abs(x(i)))**2
enddo
norm = sqrt(norm)
```

x(i)*x(i) suffices !!!

POWERS EXTREMELY EXPENSIVE

LAB 2

1. compute norms L_1,L_2,L_infty
2. “sum” of harmonic series
   - appears to converge

sadly, appears to converge to
15 (single precision)
23 (double precision)

   - how big a vector can you allocate ?

machine dependent

3. rewrite NEWTON’s algorithm in FORTRAN
   - difference between single and double precision

LAB 3-4

- ssh and scp to app.science.oregonstate.edu
- link and compile a sample program
  - use LAPACK library
    - call DGESV(n, 1, a, n, pivot, f, n, ok)
  - use modules, Makefiles
  - correct the code

replace by DPTSV (requires formatting)

```
do i = 1, n
a(i,i) = 2D0
if (i.gt.1) a(i,i-1) = -1D0
if (i.lt.n) a(i,i+1) = -1D0
enddo
```

```
do i = 1, n
a(i,i) = 2D0
enddo
do i=1,n-1
a(i,i+1) =
-1D0
a(i+1,i) =
-1D0
enddo
```

LAB 3-4

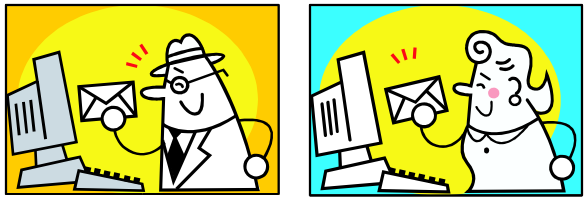
1. Implement example 2 or 3 from class
   - compute residual, Jacobian
   - call appropriate solver (DGESV ?)
2. Try a modification of Newton
   - remember to save Jacobian (factorization !)
3. Change convergence criteria –
   - tolerance etc.

LAB 5-6: iterative solvers

1. Solve Au=0
   - stationary methods
   - cgs, gmres
2. FORTRAN (Jacobi)
   1. y=1.D0/a(I,i)*(….)
3. Block Jacobi: slowly convergent
4. Newton:
   - use gmres instead of A\b
   - should influence number of Newton iterations
   - ***for ex. if tolerance is 0.5***
5. Multigrid

LAB 7-8

- implementation using MPI



- ssh and scp to cluster
- set up environment
- use the queue'ing system
  - qsub, qstat

LAB 7-8 mypi.f

- compile and run http://www.math.oregonstate.edu/~mpesz/t
- modify myn in code to be 10->100->1000.

error decreases by a factor of 100, 100 (method isO(h^2))

- Test your code on 2, 4, 8 processors.

results are the same, of course ….

LAB 7-8: handshake

- if (p.lt.nproc) then call MPI_Send(u(myn),1,MPI_DOUBLE_PRECISION, rank +1,tag,MPI_COMM_WORLD,ierr); endif

- if (p.gt.1) then call MPI_Send(u(1), 1, MPI_DOUBLE_PRECISION, rank - 1, tag,MPI_COMM_WORLD,ierr); endif

- if (p.lt.nproc) then call MPI_RECV (uright, 1, MPI_DOUBLE_PRECISION, rank, tag,MPI_COMM_WORLD,status,ierr); endif

- if (p.ge.1) then call MPI_RECV (uleft, 1, MPI_DOUBLE_PRECISION, rank - 1, tag,MPI_COMM_WORLD,status,ierr); endif

- http://www.math.oregonstate.edu/~mpesz/t

LAB 7-8 myjacobi.f

- implement
  - Jacobi in parallel,
  - same as overlapping domain decomposition
  - uses mechanisms from
    - MYPI: global reduce
    - handshake: local communication
  - http://www.math.oregonstate.edu/~mpesz/teaching/
  - http://www.math.oregonstate.edu/~mpesz/teaching/

LAB 9

1. confirm that a descent method and a decrease in residual do not lead to a success (min of x^2)
   - ex1: oscillates
   - ex2: converges to 1
2. Newton's method for minimization
   - same issues as usual: lack of global convergence
   - fast convergence only really close to the root
3. Globalized Newton with Armijo's rule and line search with backtracking
   - algorithm blows up for arctan(x)=0
   - correction: first two steps use lambda=1/4,1/8

What we have NOT done

- parallel speedup studies
- real domain decomposition examples
- calling algebraic multigrid as a linear solver
- comprehensive example:
  - find min G(U) where F(U)=0
    - coefficient identification
    - uncertainty quantification
    - inverse problems (using forward models)
    - data assimilation (Gauss-Newton)
    - nonlinear LSQ (Leverberg-Marquardt)
    - optimal control
    - shape optimization
- SEMINAR today, 12:00 in Gilkey 113

Comprehensive example

- min G(U)
  - U solves nonlinear PDE:
    - F(U)=0
    - discretize:
      - F_h(U_h)=0
    - apply Newton's method
      - J s =- F_h
    - solve linear system
      - » iterative solver (gmres)
      - » parallel preconditioner
      - » use domain decomposition

Thank you !