NEWTON-RAPHSON ITERATION FOR SOLVING A SYSTEM OF NONLINEAR EQS.

- Recall Newton's method for a single nonlinear eqn. $f(u) = 0$

Near the root u^*, $\underbrace{f(u^*)}_{0} \cong f(u) + f'(u)\cdot(u^* - u)$ (Taylor Series)

so $(u^* - u) \cong -\dfrac{f(u)}{f'(u)}$ OR $u^* \cong u - \dfrac{f(u)}{f'(u)}$

ITERATIVE SOLUTION:

start with initial guess u^0

m = ITERATION #

ITERATE : $u^{m+1} = u^m + \delta u^m$, $\quad \delta u^m \rightarrow -f(u^m)/f'(u^m)$

convergence : $|\delta u^m| \le \varepsilon$ or $|f(u^{m+1})| \le \varepsilon$

- EXTEND NEWTON'S METHOD FOR A SYSTEM OF NONLINEAR EQS.

$$\underline{f}(\underline{u}) = \underline{0} \quad \text{OR} \quad f_j(u_1, u_2, \ldots, u_J) = 0\ ; \quad j = 1, 2, \ldots J$$

Again, use Taylor-Series (multi-variable) near root $\underline{u}^*$,

$$0 = f_j(\underline{u}^*) = f_j(\underline{u}) + \left.\frac{\partial f_j}{\partial u_1}\right|_{\underline{u}} (u_1^* - u_1) + \cdots\cdots + \left.\frac{\partial f_j}{\partial u_J}\right|_{\underline{u}} (u_J^* - u_J)$$

i.e.

$$0 = f_j(\underline{u}) + \sum_{k=1}^{J} \left.\frac{\partial f_j}{\partial u_k}\right|_{\underline{u}} \delta u_k, \quad \delta u_k \rightarrow (u_k^* - u_k) \qquad \text{(A)}$$

.... BUT WE CANNOT SOLVE THIS EQUATION INDEPENDENTLY FOR δu_k's. HOWEVER WHEN WE WRITE (A) FOR EACH j, GET A SYSTEM OF J <u>LINEAR</u> EQS. TO SOLVE FOR THE δu_k's !!

$$\sum_{k=1}^{J} \left.\frac{\partial f_j}{\partial u_k}\right|_{\underline{u}} \delta u_k = -f_j(\underline{u}), \quad j = 1, 2, \ldots J$$

IN MATRIX FORM,

JACOBIAN MATRIX [F]

$$\underbrace{\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_J} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_J} \\ \vdots & & & \\ \frac{\partial f_J}{\partial u_1} & \frac{\partial f_J}{\partial u_2} & \cdots & \frac{\partial f_J}{\partial u_J} \end{bmatrix}}_{\text{evaluated at } \underline{u}} \begin{Bmatrix} \delta u_1 \\ \delta u_2 \\ \vdots \\ \delta u_J \end{Bmatrix} = \begin{Bmatrix} -f_1(\underline{u}) \\ -f_2(\underline{u}) \\ \vdots \\ -f_J(\underline{u}) \end{Bmatrix}$$

NEWTON-RAPHSON ITERATION:

initial guess : $\underline{u}^0$ m = ITERATION #

ITERATE : in each iteration, compute $\underline{f}(\underline{u}^m)$ - RHS vector;

- compute Jacobian matrix $[F^m]$, whose (j,k) element is $\dfrac{\partial f_j}{\partial u_k}(\underline{u}^m)$,
  - solve $[F^m]\,\delta\underline{u}^m = -\underline{f}(\underline{u}^m)$
  - update $\underline{u}^{m+1} = \underline{u}^m + \delta\underline{u}^m$

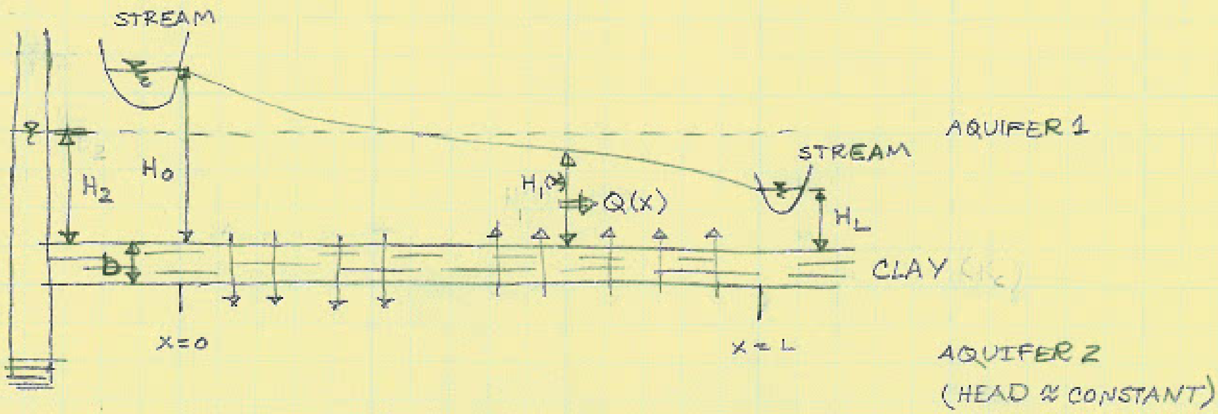
convergence : $\max(\text{abs}(\delta\underline{u}^m)) \le \varepsilon$ or $\max(\text{abs}(\underline{f}(\underline{u}^m))) \le \varepsilon$

SO, TO SOLVE A SYSTEM OF NONLINEAR EQS., WE USE AN ITERATIVE APPROACH INVOLVING SOLUTION OF A LINEAR SYSTEM OF EQS. FOR $\delta\underline{u}^m$ IN EACH ITERATION.

```
%Newton's method for an example system of 2 nonlinear equations
%equation 1:    u(1)*u(2)-10=0
%equation 2:    u(1)/u(2)-2=0
%
%initial guess
u(1)=1;u(2)=1;
%iteration loop
delta=99;
iter=0;
while delta>1e-6
   iter=iter+1;
%compute the right-hand side vector
f1=u(1)*u(2)-10;
f2=u(1)/u(2)-2;
f=[f1;f2];
%compute Jacobian Matrix
F(1,1)=u(2);
F(1,2)=u(1);
F(2,1)=1/u(2);
F(2,2)=-u(1)/u(2)/u(2);
%solve for delu (increment) using backslash
delu=F\(-f);
%update u
u=u+delu';
%print and check for convergence
[iter f']
delta=max(abs(f));
end
```

```
ans =

   1  -9  -1

ans =
  iteration #          f1                 f2
  2.00000000000000  20.00000000000000  -0.80000000000000

ans =

  3.00000000000000                  0   1.60000000000000

ans =

  4.00000000000000  -0.49382716049383   0.29090909090909

ans =
                        ~10^-2
  5.00000000000000  -0.03190834466555   0.02388922335942

ans =
                        ~10^-4
  6.00000000000000  -0.00032166727969   0.00017789036909

                                          QUADRATIC
ans =                                     CONVERGENCE
                        ~10^-8
  7.00000000000000  -0.00000001718713   0.00000001077106
```

SIMPLE EXAMPLE OF A NONLINEAR BVP : "UNCONFINED" GROUNDWATER FLOW



At any x, $Q(x) = \underbrace{-K\frac{dH_1}{dx}(x)}_{\substack{\text{flux}\\ \text{(Darcy eqn.)}}} \cdot \underbrace{H_1(x)}_{\text{thickness}}$ per unit width $\perp$ to page

K = hydraulic conductivity

mass balance : $\frac{dQ}{dx}$ = upward flux from AQUIFER 2 $= K_{clay}\frac{H_2(x) - H_1(x)}{b}$.

where $H_2(x) = H_2$, constant

i.e. $\frac{d}{dx}\left(-KH_1\frac{dH_1}{dx}\right) = K_{clay}\frac{H_2 - H_1}{b}$

Non-dimensionalize,

$\xi = x/L$, $u = H_1/H_2$, $u_0 = H_0/H_2$, $u_L = H_L/H_2$

$\frac{d}{d\xi}\left(-u\frac{du}{d\xi}\right) = \underbrace{\left(\frac{K_{clay}L^2}{K b H_2}\right)}_{\text{call this } \alpha}(1-u)$

So, $\frac{d}{d\xi}\left(-\underset{D(u)}{u}\frac{du}{d\xi}\right) = \alpha\underset{r(u)}{(1-u)}$ Nonlinear diffusion eqn.

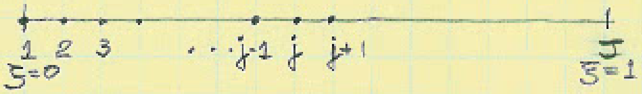
B.C. $u = u_0$ @ $\xi = 0$ $(x=0)$, $u = u_L$ @ $\xi = 1$. $(x=L)$

first recognize that $u\frac{du}{d\xi} = \frac{d}{d\xi}\left(\frac{u^2}{2}\right)$.

SO, WE CAN WRITE :

$$\boxed{\frac{d^2}{d\xi^2}(u^2) + 2\alpha(1-u) = 0, \quad 0 \le \xi \le 1. \quad u(0) = u_0,\ u(1) = u_L}$$

TO SOLVE THIS NONLINEAR BVP, DISCRETIZE (FINITE-DIFF.) TO GET A SYSTEM OF J EQS. IN J UNKNOWNS.



$$f_1 = u_1 - u_0 \qquad = 0$$

$$f_j = \frac{u_{j+1}^2 - 2u_j^2 + u_{j-1}^2}{\Delta\xi^2} + 2\alpha(1-u_j) \qquad = 0 \quad , \; j = 2, 3, \ldots J-1$$

$$f_J = u_J - u_L \qquad = 0$$

SYSTEM OF J EQS. IN J UNKNOWNS (NONLINEAR) - SOLVE BY NEWTON-RAPHSON

- INITIAL GUESS, THEN ITERATE $\underbrace{[F^m]}_{\text{Jacobian}} \delta \underline{u}^m = -\underbrace{\underline{f}}_{\text{function}}(\underline{u}^m)$. ($m$ = iteration #) (NEED TO WRITE AS $\underline{f}(\underline{u}) = 0$).
- LET US WRITE OUT THIS LINEAR SYSTEM OF EQS. FOR $\delta\underline{u}^m$ -

NOTE THAT $F(j,k) = \frac{\partial f_j}{\partial u_k}(\underline{u}^m)$.

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & & & & 0 \\ \frac{+2u_1}{\Delta\xi^2} & \frac{-4u_2}{\Delta\xi^2} - 2\alpha & \frac{+2u_3}{\Delta\xi^2} & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & \frac{+2u_{j-1}}{\Delta\xi^2} & \frac{-4u_j}{\Delta\xi^2} - 2\alpha & \frac{+2u_{j+1}}{\Delta\xi^2} & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & & \frac{+2u_{J-2}}{\Delta\xi^2} & \frac{-4u_{J-1}}{\Delta\xi^2} - 2\alpha & \frac{+2u_J}{\Delta\xi^2} \\ 0 & \cdots & & & & 0 & 0 & 1 \end{bmatrix}}_{\text{ALL AT } \underline{u}^m} \begin{Bmatrix} \delta u_1^m \\ \delta u_2^m \\ \vdots \\ \vdots \\ \vdots \\ \delta u_{J-1}^m \\ \delta u_J^m \end{Bmatrix} = \begin{Bmatrix} u_0 - u_1^m \\ -f_2^m \\ \vdots \\ -f_j^m \\ \vdots \\ -f_{J-1}^m \\ u_L - u_J^m \end{Bmatrix}$$

$$-f_j^m = \underbrace{\frac{-u_{j+1}^2 + 2u_j^2 - u_{j-1}^2}{\Delta\xi^2} - 2\alpha(1-u_j)}_{\text{with } \underline{u}^m}$$

CAN DO THIS LINEAR SYSTEM SOLUTION FOR $\delta\underline{u}^m$ - USING YOUR THOMAS ALGORITHM FUNCTION

NONLINEAR DIFFUSION EQUATION

Consider $\frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(D(u) \frac{\partial u}{\partial x} \right) = 0 , \quad 0 \le x \le L .$ (A)

Let $D(u) = \alpha u^p$ for our discussion. This is a nonlinear diffusion equation because the diffusivity depends on u.

HOW DO WE SOLVE THIS EQUATION NUMERICALLY USING AN IMPLICIT METHOD (EULER BACKWARD OR CRANK-NICHOLSON)?

- We expect that a finite-difference approximation to (A) will generate a nonlinear system of algebraic eqs. to solve for the u_j^{n+1}'s in each time-step – We can use Newton-Raphson to solve this system of nonlinear eqs.
- First, let us write the diffusion term in a slightly more convenient form:

$$\frac{\partial}{\partial x}\left(\alpha u^p \frac{\partial u}{\partial x}\right) = \frac{\alpha}{p+1} \cdot \frac{\partial^2}{\partial x^2}\left(u^{p+1}\right).$$

CHAIN RULE:

$$\frac{d}{dx}(u^{p+1}) = \frac{d}{du}(u^{p+1})\frac{du}{dx} = (p+1)\, u^p \frac{du}{dx}.$$

- SO WE HAVE

$$\frac{\partial u}{\partial t} - \frac{\alpha}{p+1} \frac{\partial^2 (u^{p+1})}{\partial x^2} = 0 , \quad 0 \le x \le L .$$

- AS BEFORE, WE WILL USE A FINITE-DIFFERENCE APPROX. AT INTERIOR NODES, $j = 2 \ldots J-1$. WITH EULER-BACKWARD, THESE ARE:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} - \frac{\alpha}{p+1} \frac{(u_{j-1}^{n+1})^{p+1} - 2\,(u_j^{n+1})^{p+1} + (u_{j+1}^{n+1})^{p+1}}{\Delta x^2} = 0 \quad (B)$$

(note: "n" and "n+1" are superscripts indicating time level; p+1 is an exponent. u_j^n is KNOWN AT TIME LEVEL "n". u_{j-1}^{n+1}, u_j^{n+1}, u_{j+1}^{n+1} ARE UNKNOWN)

- LET US ALSO WRITE THE USUAL B.C. EQS. AT $j=1$ AND J:

$$\underline{j=1}: \quad c_1 U_1^{n+1} + c_2 \frac{U_2^{n+1} - U_1^{n+1}}{\Delta x} = \boxed{B_1^{n+1}} \longrightarrow \text{GIVEN FROM B.C. SPECIFIED.}$$

$$\underline{j=J}: \quad c_3 U_J^{n+1} + c_4 \frac{U_J^{n+1} - U_{J-1}^{n+1}}{\Delta x} = \boxed{B_J^{n+1}}$$

- TO SOLVE THE J ALGEBRAIC EQS. USING NEWTON ITERATION, WE FIRST NEED TO PUT THEM IN "STANDARD FORM"

$$\underline{f}(\underline{u}^{n+1}) = \underline{0} \quad \text{OR} \quad f_j(u_1^{n+1}, \ldots, u_J^{n+1}) = 0 \; ; \; j = 1, \ldots J$$

- Let $\mu = \frac{\alpha \Delta t}{\Delta x^2}$

$$f_1(\underline{u}^{n+1}) = \left(c_1 - \frac{c_2}{\Delta x}\right) u_1^{n+1} + \frac{c_2}{\Delta x} u_2^{n+1} - B_1^{n+1} = 0$$

$$f_j(\underline{u}^{n+1}) = -\frac{\mu}{p+1}\left(u_{j-1}^{n+1}\right)^{p+1} + \left(u_j^{n+1} + \frac{2\mu}{p+1}\left(u_j^{n+1}\right)^{p+1}\right) - \frac{\mu}{p+1}\left(u_{j+1}^{n+1}\right)^{p+1} - u_j^{n} = 0\,; \quad j = 2 \ldots J-1$$

* THIS TERM WILL NOT CHANGE DURING ITERATION

$$f_J(\underline{u}^{n+1}) = -\frac{c_4}{\Delta x} u_{J-1}^{n+1} + \left(c_3 + \frac{c_4}{\Delta x}\right) u_J^{n+1} - B_J^{n+1} = 0$$

What is the Jacobian matrix associated with these eqs.? Let $u_j^{n+1,m}$ denote the m^{th} (current) iteration estimate of u_j^{n+1}; $[F]^m$ the corresponding Jacobian matrix.

$$\begin{bmatrix} c_1 - \frac{c_2}{\Delta x} & \frac{c_2}{\Delta x} & 0 & \cdots & \cdots & 0 \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ 0 & 0 & \cdots & 0 & -\frac{c_4}{\Delta x} & c_3 + \frac{c_4}{\Delta x} \end{bmatrix}$$

$$\frac{\partial f_j^{n+1,m}}{\partial u_j^{n+1,m}} = 1 + 2\mu\left(u_j^{n+1}\right)^p$$

$$\frac{\partial f_j^{n+1,m}}{\partial u_{j+1}^{n+1,m}} = -\mu\left(u_{j+1}^{n+1,m}\right)^p$$

$$\frac{\partial f_j^{n+1,m}}{\partial u_{j-1}^{n+1,m}} = -\mu\left(u_{j-1}^{n+1,m}\right)^p$$

Note that the Jacobian matrix is tridiagonal.

It is best constructed in the form of the a, b, c vectors.

STEPS INVOLVED IN CALCULATING SOLUTION:

START FROM INITIAL CONDITION AT $t = 0$

TIME LOOP

IN EACH TIME STEP, NEED TO SOLVE FOR u_j^{n+1}'s

INITIAL GUESS: $u_j^{n+1,0} = u_j^n$ (often good)

NEWTON-RAPHSON ITERATION LOOP (while max(abs(f)) $> \epsilon$)

i) EVALUATE $\underline{f}(\underline{u}^{n+1,m})$ (m = iteration #, $\underline{u}^{n+1,m}$ = current estimate)

ii) EVALUATE $[F]^m$ IN THE FORM OF a, b, c VECTORS

iii) SOLVE $[F]^m \delta\underline{u}^m = -\underline{f}(\underline{u}^{n+1,m})$ (delu = thomas(a, b, c, -f))

iv) UPDATE $\underline{u}^{n+1,m+1} = \underline{u}^{n+1,m} + \delta\underline{u}^m$

good place to check for convergence max(abs(f)) $\geq \epsilon$

NOW HAVE CORRECT $\underline{u}^{n+1}$

BEFORE GOING TO NEXT TIME-STEP, NEED TO SET THIS $\underline{u}^{n+1} = \underline{u}^n$ FOR NEXT TIME-STEP ("uold" = "u")

PLOT IF NEEDED

IN PRACTICE, YOU WILL NEED TWO VECTORS ($\underline{u}^n$ = "uold") AND ($\underline{u}^{n+1,m}$ = "u")

YOU CAN OVERWRITE THIS VECTOR IN EACH ITERATION