4.3) Finite Difference Methods.

A) Shooting Method

B) Matrix Method

C) Newton Raphson.

A) Shooting Method

Tradition approach for solving F-S equations (Boundary value problem)

Boundary Value Problem : BC on both ends.

IVP : all B.Cs on one side /end.

In the shooting method

1. Temporarily drop B.C at one end, replace with additional B.C at initial end, example $S(0) = S_g$ (guess)

2. Integrate using forward Euler or R-K from 0 to some η_c

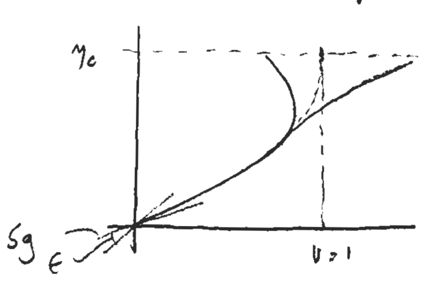
$$F_{i+1} = F_i + \Delta\eta U_i$$
$$U_{i+1} = U_i + \Delta\eta S_i$$
$$S_{i+1} = S_i + \Delta\eta f(F_i, U_i, S_i; \beta_u)$$

with $F_1 = 0 = U_1$, $S_i = S_g$

3. If/is $U_N = 1$?, if not adjust S_g, and repeat from 1.

- Problem with shooting method
  - iterative (not efficient)
  - A well posed B.VP can easily result in a very ill-posed I.V problem

Ex. when $\beta_u < 0$ or $S(0) \approx 0$



B) Matrix Method (Modern)

- Solve B.V w/o artificially changing B.Cs
- Assemble a matrix of algebraic equations + B.Cs, and solve simultaneously (all at once) to get solution

Define residuals for each unknown

$$R_F = \int_{\eta}^{\eta+\Delta\eta} (F' - U)\, d\eta = 0$$

$$R_u = \int_{\eta}^{\eta+\Delta\eta} (U' - S)\, d\eta = 0$$

$$R_S = \int_{\eta}^{\eta+\Delta\eta} (S' + \cdots)\, d\eta = 0$$

$$\Rightarrow \quad F_{i+1} - F_i - \frac{\Delta\eta}{2}(U_{i+1} + U_i) = 0$$

$$U_{i+1} - U_i - \frac{\Delta\eta}{2}(S_{i+1} + S_i) = 0$$

$$S_{i+1} - S_i - \frac{\Delta\eta}{2}\left(\frac{\beta_u+1}{2}\right)(F_{i+1}S_{i+1} + F_iS_i) + \cdots = 0$$

Note: Implicit explicit scheme has no effect on matrix method, accuracy is important

We have B.C: $F_1 = 0$, $U_1 = 0$, $U_N = 1$

and $3N \times 3N$ non-linear equation system for F, U, and S with parameter β_u. We can solve this using multidimension N-Raphson method

c) Newton-Raphson Method

Applicable to any well-posed linear or non linear system of equations.

Scalar Case: find x s.t. $f(x) = 0$, $f(x)$ is given

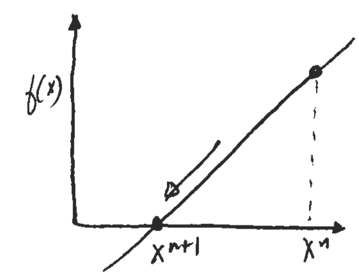
Given some guess X^n, $f(x^n) \neq 0$

we can update x^n such that $f(x^n + \delta x^n) = 0$

or $f(x^n + \delta x^n) \simeq f(x^n) + \left.\frac{df}{dx}\right|_{x^n} \delta x^n + H.O.T = 0$

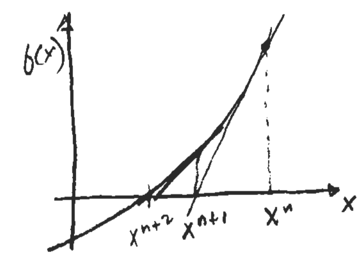
$$\Rightarrow \quad \delta x^n = -f(x^n) \Big/ \left.\frac{df}{dx}\right|_{x^n}$$

new guess: $x^{n+1} = x^n + \delta x^n$

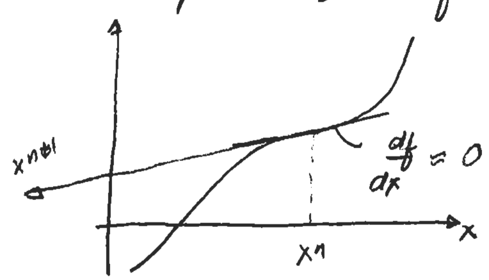


If $f(x)$ is linear in x, converges in one iteration

For non-linear $f(x)$, (local) convergence is generally quadratic: $\delta x^{n+1} \sim (\delta x^n)^2$ (for a close initial guess)



N-R method has problems if $\frac{df}{dx} \simeq 0 \rightarrow \delta x \rightarrow \infty$



In practice δx^n must be examined before updating x^n, δx^n may be under relaxed

$$x^{n+1} = x^n + r\,\delta x^n$$

where $r = 1$ if δx is reasonable
$r < 1$ otherwise

Multidimensional N-R:

Given M equations in M unknowns.

$$f_1(x_1, x_2 \cdots\cdots x_M)$$
$$f_2(x_1, x_2, \cdots\cdots x_M)$$
$$\vdots$$
$$f_M(x_1, x_2, \cdots\cdots x_M).$$

find $\vec{x}$ such that $\vec{f}(\vec{x}) = \vec{0}$

Given some guess $\vec{x}^n$, $\vec{f}(\vec{x}^n) \neq \vec{0}$

we seek $\delta\vec{x}^n$ s.t

$$\vec{f}(\vec{x}^n + \delta\vec{x}^n) \simeq \vec{f}(\vec{x}^n) + \left[\frac{\partial\vec{f}}{\partial\vec{x}}\right]\Bigg/_{\vec{x}^n} \delta\vec{x}^n = \vec{0}$$

$J = \left[\frac{\partial\vec{f}}{\partial\vec{x}}\right]$ — $M \times M$ Jacobian matrix

$$\therefore \quad [J]\{\delta x\} = -\vec{f}(\vec{x})$$

$M \times M$ linear system

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots \\ \vdots & \ddots & \\ \frac{\partial f_M}{\partial x_1} & \cdots & \frac{\partial f_M}{\partial x_M} \end{bmatrix} \longleftarrow \text{i,j entry} = \frac{\partial f_i}{\partial x_j}\Bigg/_{\vec{x}^n}$$

$$\therefore \quad \delta\vec{x}^n = -[J]^{-1}\vec{f}(\vec{x}^n)$$

new guess (or solution) $\vec{x}^{n+1} = \vec{x}^n + \delta\vec{x}^n$

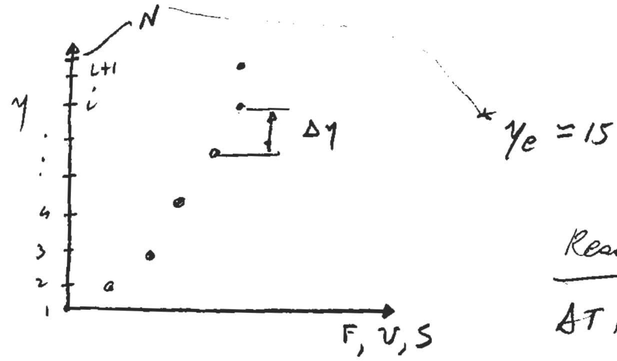
Problem if $\left[\frac{\partial\vec{f}}{\partial\vec{x}}\right]$ is singular, or ill-conditioned. $\rightarrow \delta\vec{x} \sim \infty$

Application of N-R to F-S equations

$$\frac{\partial F}{\partial \eta} - U = 0$$

$$\frac{\partial U}{\partial \eta} - S = 0$$

$$\frac{\partial S}{\partial \eta} + \frac{1+\beta_u}{2} FS + \beta_u (1-U^2) = 0$$



Reading

ATP 549-558

$\eta = 0$: $F = 0$, $U = 0$, $U(\eta = \eta_e) = 1$

Discrete system is obtained using Trapezoidal scheme

$$F_{i+1} - F_i - \frac{\Delta\eta}{2}(U_{i+1} + U_i) \equiv R_{F_i}(F_i, U_i, F_{i+1}, U_{i+1}) = 0$$

$$U_{i+1} - U_i - \frac{\Delta\eta}{2}(S_{i+1} - S_i) \equiv R_{U_i}(U_i, S_i, U_{i+1}, S_{i+1}) = 0$$

$$S_{i+1} - S_i + \left(\frac{1+\beta_u}{2}\right)\frac{\Delta\eta}{2}(F_{i+1}S_{i+1} + F_i S_i) + \beta_u \Delta\eta\left(1 - 1/2\,(U_{i+1}^2 + U_i^2)\right)$$
$$\equiv R_{S_i}(F_i, U_i, S_i, F_{i+1}, U_{i+1}, S_{i+1}, \beta_u) = 0$$

B.Cs : $F_1 \equiv R_{BC_1} = 0$, $U_1 \equiv R_{BC_2} = 0$, $U_N - 1 = R_{BC_3} = 0$

In addition, we need additional equations to drive either β_u or equivalently H as global variables

1. $R_\beta(\beta_u) = \beta_u - \beta_{u\,spec} = 0$

or 2. $R_H(U_1, U_2, \cdots U_N) = H - H_{spec} = 0$

$H = \frac{\delta_1^*}{\theta_1}$ $\Rightarrow R_H = \delta_1^* - \theta_1 H_{spec} = 0$

$$= \int_0^{\eta_e}(1-U)\,d\eta - H_{spec}\int_0^{\eta_e}(1-U)\,U\,d\eta = 0$$

$$= \sum_{i=1}^{N-1}\left(1 - \frac{U_{i+1}+U_i}{2}\right)\Delta\eta - (H_{spec})\sum_{1}^{N-1}\left(1 - \frac{U_{i+1}+U_i}{2}\right)\left(\frac{U_{i+1}+U_i}{2}\right)\Delta\eta = 0$$

Set up N-R system to solve for

δF_i, δU_i, δS_i, and $\delta \beta_u$ or δH

giving $3N+1$ unknowns.

Main task in applying/using N-R is linearizing the equation and assembling the Jacobian matrix. Lets examine one residual equation . . .

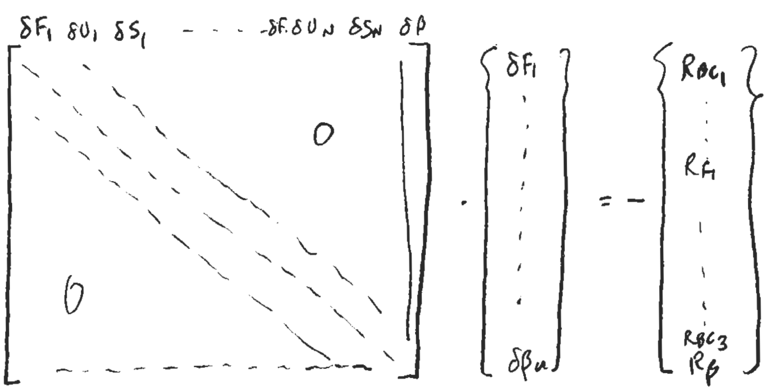
$$R_{F_i}^{n+1} = R_{F_i}\left(F_{i+1}^{n+1},\ U_{i+1}^{n+1}\ \cdots\right) = R_{F_i}\left(F_{i+1}^{n} + \delta F_{i+1}^{n},\ U_{i+1}^{n} + \delta U_{i+1}^{n}\ \cdots\right)$$

$$\simeq R_{F_i}^{n} + \left(\frac{\partial R_{F_i}}{\partial F_{i+1}}\right)^{n} \delta F_{i+1}^{n} + \left(\frac{\partial R_{F_i}}{\partial U_{i+1}}\right)^{n} \delta U_{i+1}^{n} + \cdots = 0$$

$$\Rightarrow \underbrace{\left(\frac{\partial R_{F_i}}{\partial F_{i+1}}\right)^{n}}_{\text{Jacobian Coeff}} \delta F_{i+1}^{n} + \underbrace{\left(\frac{\partial R_{F_i}}{\partial U_{i+1}}\right)^{n}}_{\text{Jacobian Coeff}} \underbrace{\delta U_{n+1}^{n}}_{\text{unknowns}} + \cdots = \underbrace{-R_{F_i}^{n}}_{\text{RHS}}$$

Coefficient Example: $\dfrac{\partial R_{F_i}}{\partial F_{i+1}} = 1$, $\dfrac{\partial R_{F_i}}{\partial U_{i+1}} = -\dfrac{\Delta \eta}{2}$, etc.

The Jacobian matrix will take the form.



Note that the Jacobian has a sparse tridiagonal structure, except the row and column due to R_β. This can eliminated by solving system in 2 steps . . .

Step ① - The last row containing the global variable eqn is put aside, and the column ($\delta\beta_u$) is placed on the RHS.

$$
\begin{array}{c}
\begin{bmatrix} \ddots & & 0 \\ & \ddots & \\ 0 & & \ddots \end{bmatrix} \\
\delta F_1\ \delta U_1\ \delta S_1 \cdots\cdots \delta F_N, \delta U_N\ \delta S_N
\end{array}
\begin{Bmatrix} \delta F_1 \\ \delta U_1 \\ \delta S_1 \\ \vdots \\ \delta F_N \\ \delta U_N \\ \delta S_N \end{Bmatrix}
= - \begin{Bmatrix} R_{BC_1} \\ \vdots \\ R_{BC_3} \end{Bmatrix}
- \delta\beta_u \begin{Bmatrix} \frac{\partial R_{BC_1}}{\partial \beta_u} \\ \vdots \\ \frac{\partial R_i}{\partial \beta_u} \\ \vdots \\ \frac{\partial R_{BC}}{\partial \beta_u} \end{Bmatrix}
$$

Solve using block matrix solver

$$
\begin{Bmatrix} \delta F_1 \\ \vdots \\ \delta S_N \end{Bmatrix}
= - \begin{Bmatrix} r_{F_1} \\ r_{u_1} \\ r_{S_1} \\ \vdots \\ r_{S_N} \end{Bmatrix}
- \delta\beta_u \begin{Bmatrix} s_{F_1} \\ s_{u_1} \\ \vdots \\ s_{S_N} \end{Bmatrix} \qquad (*)
$$

Step ② We need to solve for $\delta\beta_u$ so that $\delta F, \delta U, \delta S$ can be completely determined. Taking the residual equation for $\delta\beta_u$

$$
\begin{bmatrix} \frac{\partial R_\beta}{\partial F_1} & \frac{\partial R_\beta}{\partial U_1} & \cdots & \frac{\partial R_\beta}{\partial S_N} \end{bmatrix}
\begin{Bmatrix} \delta F_1 \\ \vdots \\ \delta S_N \end{Bmatrix}
+ \begin{bmatrix} \frac{\partial R_\beta}{\partial \beta_u} \end{bmatrix} \delta\beta_u = - R_\beta
$$

Substituting *

$$
\begin{bmatrix} \frac{\partial R_\beta}{\partial F_1} & \cdots & \frac{\partial R_\beta}{\partial S_N} \end{bmatrix} \cdot
\begin{Bmatrix} s_{F_1} \\ \vdots \\ s_{S_N} \end{Bmatrix} \delta\beta_u
+ \frac{\partial R_\beta}{\partial \beta_u} \cdot \delta\beta_u = - R_\beta
+ \begin{bmatrix} \frac{\partial R_\beta}{\partial F_1} & \cdots & \frac{\partial R_\beta}{\partial S_N} \end{bmatrix} \cdot
\begin{Bmatrix} r_{F_1} \\ \vdots \\ r_{S_N} \end{Bmatrix}
$$

This can be solved for $\delta\beta_u$, which is substituted in (*) to get the final update for $\delta F, \delta U, \delta S$.

F-S solution steps (algorithm)

1. Init F_i, U_i, S_i
2. Setup system
   a) fill matrix (Jacobian)
   b) fill RHS
3. Solve system for $\delta F_i, \delta U_i, \delta S_i$
4. Update
5. Check for convergence → goto to 2

If H is specified & β_u calculated, add row and column for unknown β_u. $R_\beta = H - H_{spec} = 0$

Express R_β as

$$R_\beta = \delta_1^* - \theta_1 H_{spec} = 0$$

$$= \int_0^{\eta_e} (1-U)\, \delta\eta - H_{spec} \int_0^{\eta_e} (1-U)\, U\, \delta\eta = 0$$

$$= \sum_1^{N-1} \left(1 - \frac{U_{i+1} + U_i}{2}\right) \Delta\eta - H_{spec} \sum_1^{N-1} \left(1 - \frac{U_{i+1} + U_i}{2}\right)\left(\frac{U_{i+1} + U_i}{2}\right) \Delta\eta = 0$$