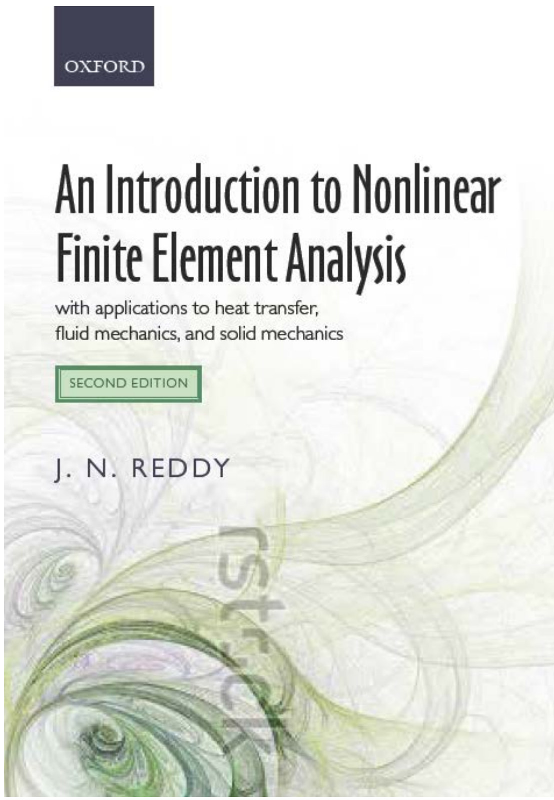
MEEN 673: Nonlinear Finite Element Analysis

1D Nonlinear Finite Element Analysis

Read: Chapter 4



CONTENTS

- Types of nonlinearities
- Finite element formulation of 1-D problem (Sec. 4.2 and Sec. 4.3)
- Solution of nonlinear equations (Sec. 4.4)
- Calculation of tangent matrix coefficients
- Computer implementation (Sec. 4.5)
- Numerical examples (Sec. 4.5)

TYPES OF NONLINEARITIES

"Load" vs. "deflection" (or "cause" vs. "effect") is nonlinear because of

- the source of the nonlinearity is geometry or description of motion (e.g., structures undergoing large displacements, strains, or rotations; convective terms of the Navier-Stokes equations).
- the source of the nonlinearity is in the material description (e.g., the elastic material parameters are strain-dependent or temperature dependent; the viscosity of the fluid is strain-rate dependent).

MODEL 1-D PROBLEM

Model Equation

$$-\frac{d}{dx}\left(a(x,u)\frac{du}{dx}\right)+b(x,u)\frac{du}{dx}+c(x,u)u=f(x),\ \ 0<x<L$$

where $(u_x = du\,/\,dx)$

$$a=a(x,u,u_x),\ b=b(x,u,u_x),\ \ c=c(x,u,u_x)$$

$$\text{Approximate solution:}\quad u(x)\approx u_h(x)=\sum_{j=1}^{n}u_j\psi_j(x)$$

Weak Form

$$0=\int_{x_a}^{x_b}\left(a\frac{dw_i}{dx}\frac{du_h}{dx}+bw_i\frac{du_h}{dx}+cw_iu_h-w_if\right)dx-w_i(x_a)Q_{\rm a}-w_i(x_b)\cdot Q_b$$

$$=\int_{x_a}^{x_b}\left(a\frac{dw_i}{dx}\frac{du_h}{dx}+bw_i\frac{du_h}{dx}+cw_iu_h\right)dx-\left[\int_{x_b}^{x_a}w_if\,dx+w_i(x_a)Q_{\rm a}+w_i(x_b)\cdot Q_b\right]$$

FINITE ELEMENT MODEL

$$u_h^e(x) = \sum_{j=1}^{n} u_j^e \psi_j^e(x), \qquad w_i = \psi_i^e(x)$$

$$0 = \int_{x_a}^{x_b} \left(a \frac{dw_i}{dx}\frac{du_h}{dx} + bw_i \frac{du_h}{dx} + cw_i u_h \right) dx - \left[\int_{x_b}^{x_a} w_i f \, dx + w_i(x_a) Q_a + w_i(x_b) \cdot Q_b \right]$$

$$= \sum_{j=1}^{n} u_j \int_{x_a}^{x_b} \left(a \frac{d\psi_i}{dx}\frac{d\psi_j}{dx} + b\psi_i \frac{d\psi_j}{dx} + c\psi_i \psi_j \right) dx - \left[\int_{x_b}^{x_a} \psi_i f \, dx + \psi_i(x_a) Q_1^e + \psi_i(x_b) Q_n^e \right]$$

$$\sum_{j=1}^{n} K_{ij}^e(u_k) u_j^e = F_i^e \quad \Rightarrow \quad \mathbf{K}^e(\mathbf{u}^e)\mathbf{u}^e = \mathbf{F}^e$$

$$K_{ij}^e = \int_{x_a}^{x_b} \left(a_e \frac{d\psi_i^e}{dx}\frac{d\psi_j^e}{dx} + b_e \psi_i^e \frac{d\psi_j^e}{dx} + c_e \psi_i^e \psi_j^e \right) dx$$

$$F_i^e = \int_{x_a}^{x_b} f_e \, \psi_i \, dx + \psi_i(x_a) Q_1^e + \psi_i(x_b) Q_n^e$$

SOLUTION OF NONLINEAR EQUATIONS

Direct Iteration

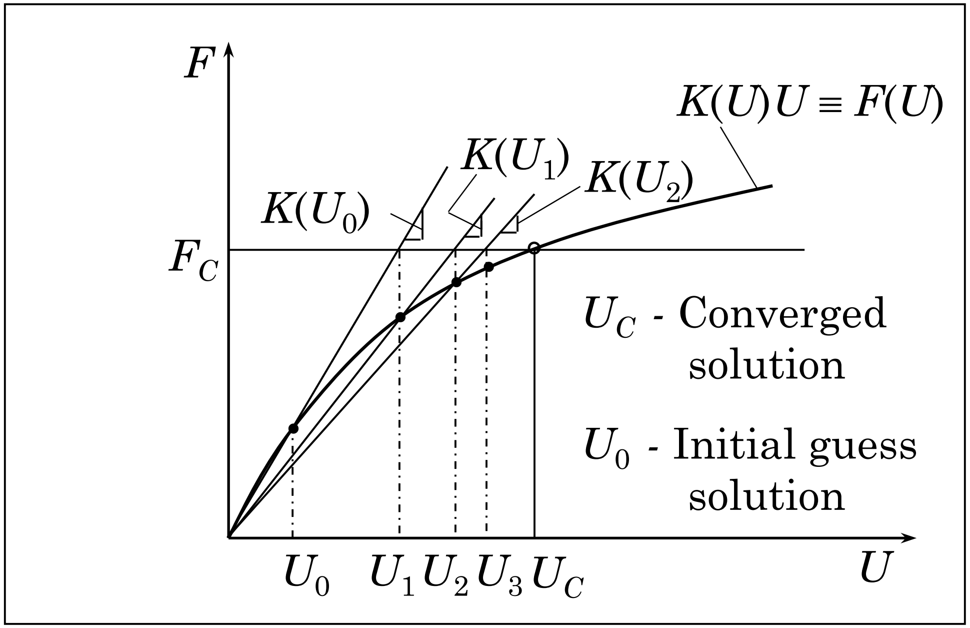
Non-Linear Finite Element Model

$$\mathbf{K}^e(\mathbf{u}^e)\mathbf{u}^e = \mathbf{F}^e \Rightarrow \text{assembled} \quad \mathbf{K}(\mathbf{U})\mathbf{U} = \mathbf{F}$$

Direct Iteration Method

When the solution $\mathbf{U}^r$ at r^{th} iteration is known, solve for $\mathbf{U}^{r+1}$

$$\mathbf{K}(\mathbf{U}^r)\mathbf{U}^{r+1} = \mathbf{F}$$



SOLUTION OF NONLINEAR EQUATIONS (continued)

Direct Iteration Method

Solution $\mathbf{U}^r$ at r^{th} iteration is known and solve for $\mathbf{U}^{r+1}$

$$\mathbf{K}(\mathbf{U}^r)\mathbf{U}^{r+1} = \mathbf{F}$$

Convergence Criterion

$$\varepsilon = \sqrt{\frac{\sum_{I=1}^{NEQ}\left(U_I^r - U_I^{r+1}\right)^2}{\sum_{I=1}^{NEQ}\left(U_I^{r+1}\right)^2}} \leq \text{specified tolerance}$$

SOLUTION OF NONLINEAR EQUATIONS

Newton Iteration

Objective:

$$\text{Residual,}\quad \mathbf{R}(\mathbf{U}) \equiv \mathbf{K}(\mathbf{U})\mathbf{U} - \mathbf{F} \Rightarrow \mathbf{0}$$

Taylor's series:

$$\mathbf{R}(\mathbf{U}^{r+1}) = \mathbf{R}(\mathbf{U}^{r}) + (\mathbf{U}^{r+1} - \mathbf{U}^{r})\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]^{r} + \frac{1}{2!}(\mathbf{U}^{r+1} - \mathbf{U}^{r})^{2}\left[\frac{\partial^{2} \mathrm{R}}{\partial \mathbf{U}^{2}}\right]^{r} + \cdots$$

$$\approx \mathbf{R}(\mathbf{U}^{r}) + (\mathbf{U}^{r+1} - \mathbf{U}^{r})\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]^{r} + O(\delta\mathbf{U})^{2},\quad \delta\mathbf{U} = \mathbf{U}^{r+1} - \mathbf{U}^{r}$$

Requiring the residual $\mathbf{R}^{r+1}$ to be zero at the $r+1^{\mathbf{st}}$ iteration, we have

$$\boxed{\mathbf{K}^{\mathbf{tan}}(\mathbf{U}^{r})\delta\mathbf{U} = -\mathbf{R}^{r} = \mathbf{F}^{r} - \mathbf{K}(\mathbf{U}^{r})\mathbf{U}^{r}}$$

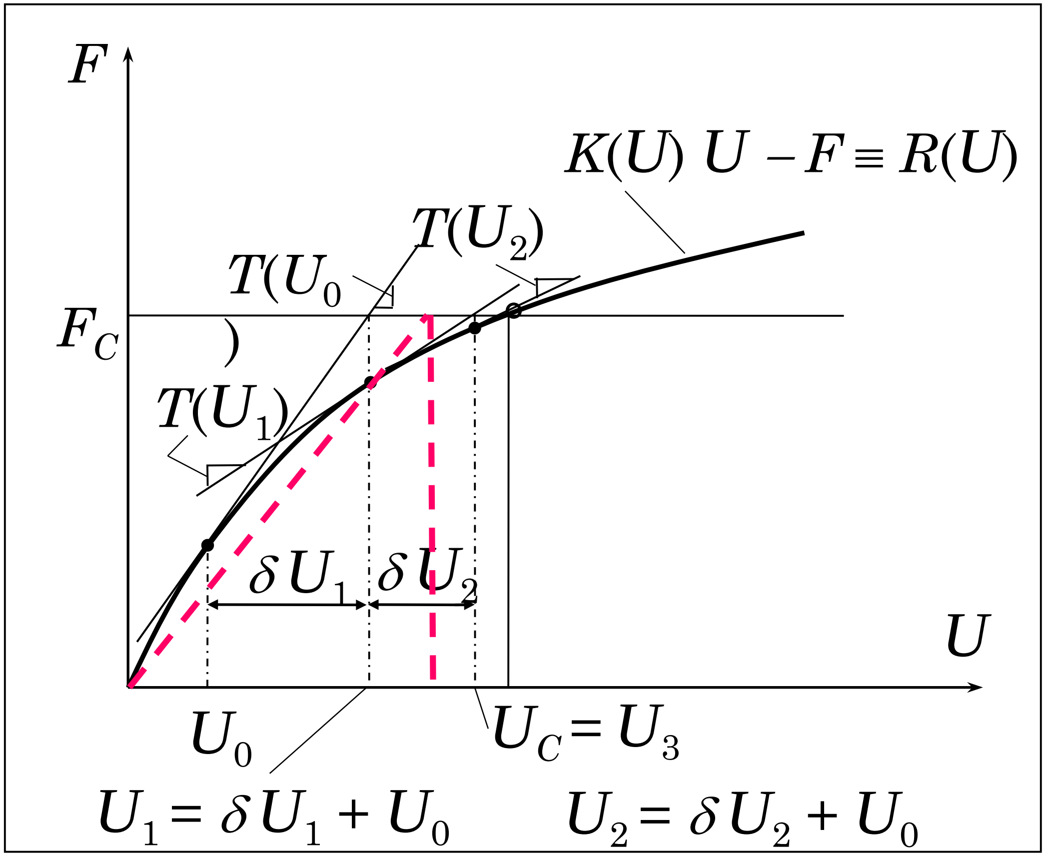
The tangent matrix at the element level is

$$\left(K_{ij}^{e}\right)^{\tan} = \frac{\partial R_{i}^{e}}{\partial u_{j}^{e}} = \frac{\partial}{\partial u_{j}^{e}}\left(\sum_{p=1}^{n} K_{ip}^{e} u_{p}^{e} - F_{i}^{e}\right)$$

SOLUTION OF NONLINEAR EQUATIONS
Newton-Raphson Iteration (continued)

$$\left(K_{ij}^e\right)^{\tan} = \frac{\partial R_i^e}{\partial u_j^e} = \frac{\partial}{\partial u_j^e}\left(\sum_{p=1}^{n} K_{ip}^e u_p^e - F_i^e\right) = K_{ij}^e + \sum_{p=1}^{n} \frac{\partial K_{ip}^e}{\partial u_j^e} u_p^e \equiv T_{ij}^e$$

$$\mathbf{T}(\mathbf{U}^r)\delta\mathbf{U} = \mathbf{F}^r - \mathbf{K}(\mathbf{U}^r)\mathbf{U}^r, \quad \mathbf{U}^{r+1} = \mathbf{U}^r + \delta\mathbf{U}$$



U_C - Converged solution

U_0 - Initial guess solution

COMPUTATION OF TANGENT MATRIX COEFFICIENTS

$$\sum_{j=1}^{n} K_{ij}^{e}(u_k)u_j^e = F_i^e, \quad K_{ij}^e = \int_{x_a}^{x_b}\left(a_e \frac{d\psi_i^e}{dx}\frac{d\psi_j^e}{dx} + b_e \psi_i^e \frac{d\psi_j^e}{dx} + c_e \psi_i^e \psi_j^e\right) dx$$

$$F_i^e = \int_{x_a}^{x_b} f_e \, \psi_i \; dx + \psi_i(x_a) Q_1^e + \psi_i(x_b) Q_n^e$$

$$\left(K_{ij}^e\right)^{\tan} = \frac{\partial R_i^e}{\partial u_j^e} = \frac{\partial}{\partial u_j^e}\left(\sum_{p=1}^{n} K_{ip}^e u_p^e - F_i^e\right) = K_{ij}^e + \sum_{p=1}^{n} \frac{\partial K_{ip}^e}{\partial u_j^e} u_p^e \equiv T_{ij}^e$$

EXERCISE:

Given: $a_e(x,u) = a_0^e(x) + a_u^e u(x) + a_{ux}^e \dfrac{du}{dx}$

Compute: T_{ij}^e

Computation of Tangent Matrix Coefficients

An Example:

$$-\frac{d}{dx}\left(u\frac{du}{dx}\right) = f_0, \quad 0 < x < 1;$$

$$\left[u\frac{du}{dx}\right]_{x=0} = \hat{Q}, \quad u(1) = \hat{u}$$

$$K_{ij}^e = \int_{x_a}^{x_b} \left[\sum_{k=1}^{n} u_k^e \psi_k^e\right] \frac{d\psi_i^e}{dx}\frac{d\psi_j^e}{dx} dx$$

$$= \sum_{k=1}^{n} u_k^e \int_{x_a}^{x_b} \psi_k^e \frac{d\psi_i^e}{dx}\frac{d\psi_j^e}{dx} dx$$

Computation of Tangent Matrix Coefficients

$$T_{ij}^e = K_{ij}^e + \sum_{m=1}^{n} \frac{\partial K_{im}^e}{\partial u_j^e} u_m^e = K_{ij}^e + \sum_{m=1}^{n} \frac{\partial}{\partial u_j^e}\left(\int_{x_a}^{x_b} u_h \frac{d\psi_i^e}{dx}\frac{d\psi_m^e}{dx}\,dx\right) u_m^e$$

$$= K_{ij}^e + \int_{x_a}^{x_b} \frac{\partial u_h}{\partial u_j^e}\frac{d\psi_i^e}{dx}\left(\sum_{m=1}^{n} u_m^e \frac{d\psi_m^e}{dx}\right) dx = K_{ij}^e + \int_{x_a}^{x_b} \frac{du_h}{dx}\frac{d\psi_i^e}{dx}\psi_j^e\,dx$$

$$\equiv K_{ij}^e + \hat{K}_{ij}^e$$

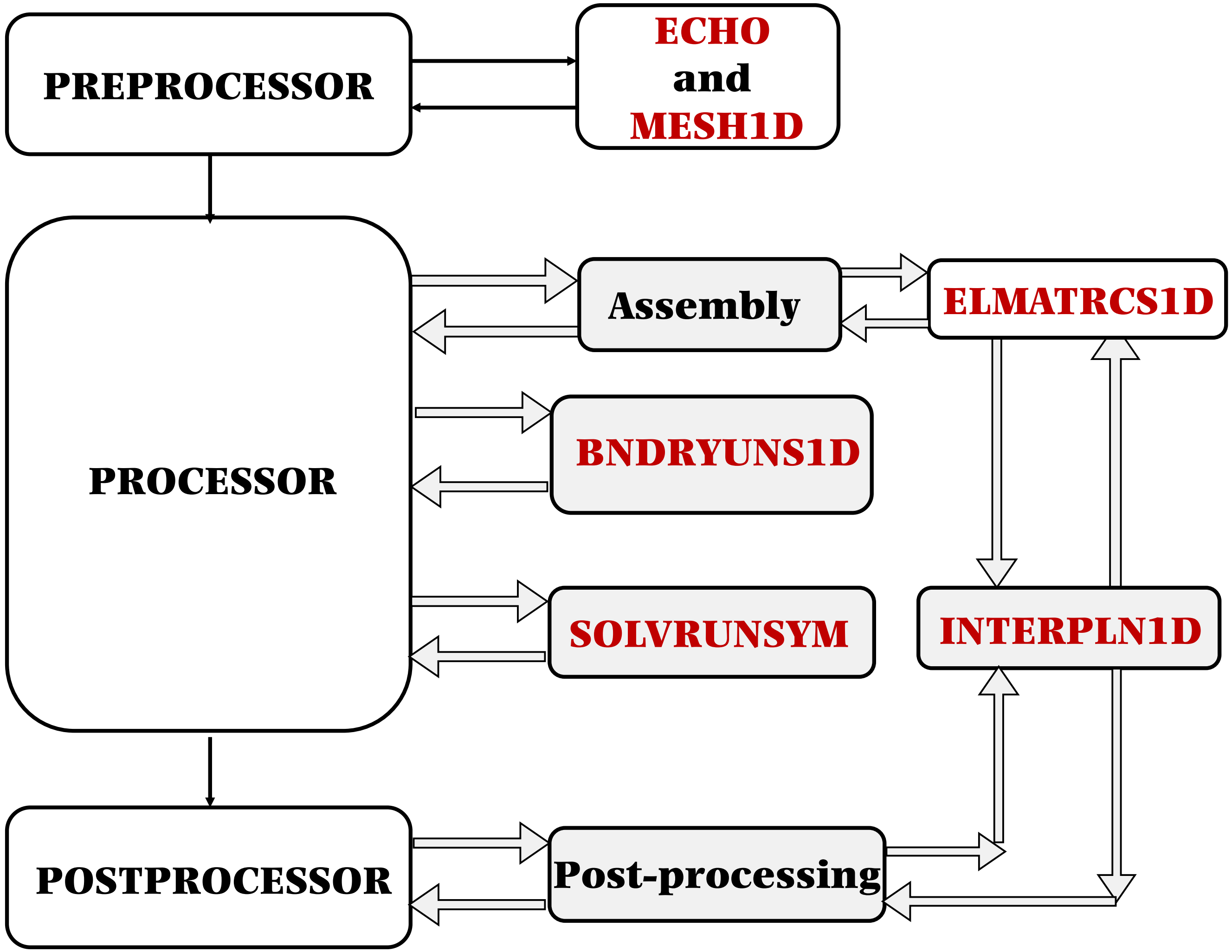
where (linearize) $\hat{K}_{ij}^e = \int_{x_a}^{x_b} \frac{d\bar{u}_h}{dx}\frac{d\psi_i^e}{dx}\psi_j^e\,dx$

We have the linearized tangent matrix

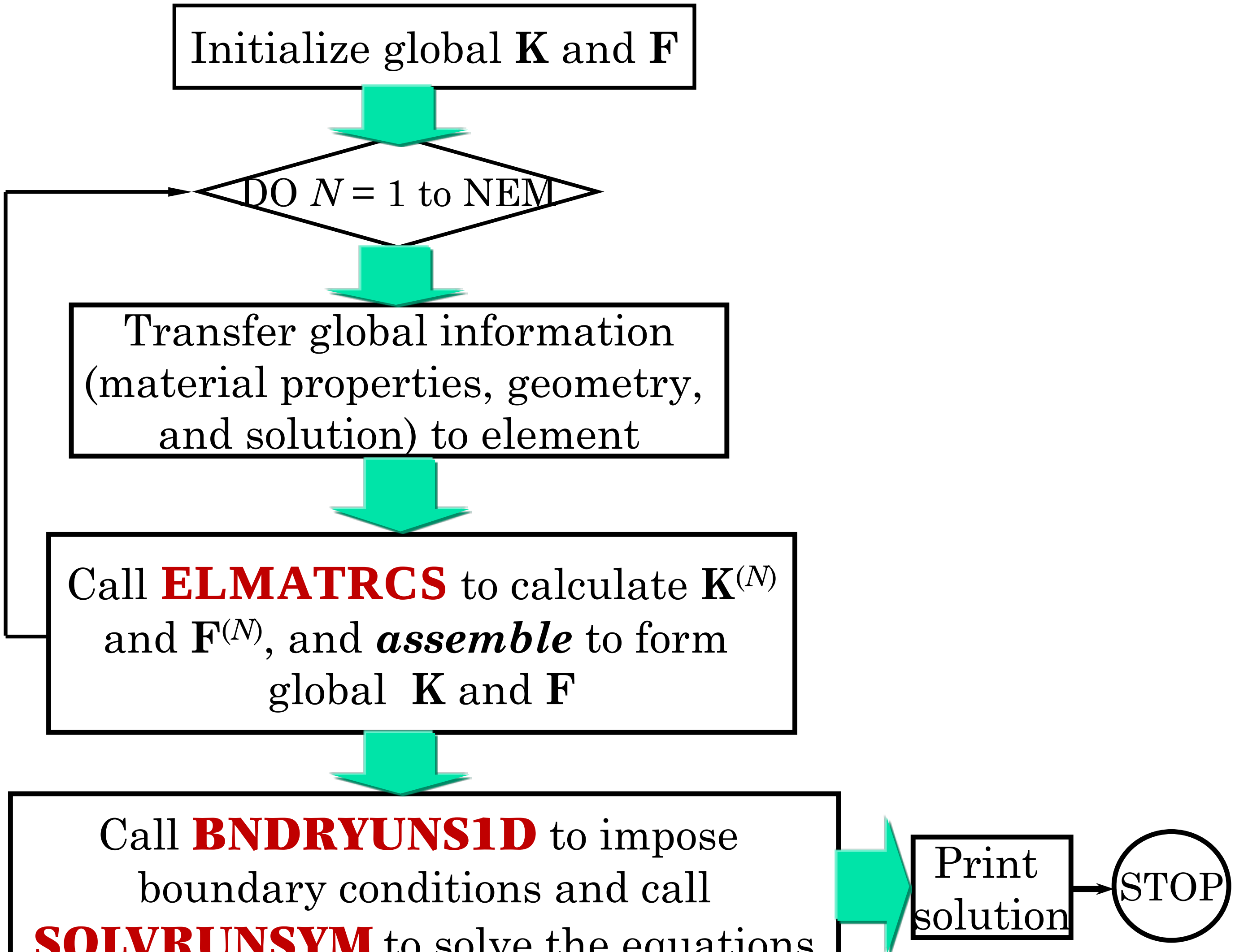
$$T_{ij}^e = \int_{x_a}^{x_b} \bar{u}_h \frac{d\psi_i^e}{dx}\frac{d\psi_j^e}{dx}\,dx + \int_{x_a}^{x_b} \frac{d\bar{u}_h}{dx}\frac{d\psi_i^e}{dx}\psi_j^e\,dx$$

$$\bar{u}_h^e(x) = \sum_{m=1}^{n} \bar{u}_m^e \psi_m^e, \quad \frac{d\bar{u}_h^e}{dx} = \sum_{m=1}^{n} \bar{u}_m^e \frac{d\psi_m^e}{dx}$$

FLOW CHART OF FEM1DUNSYM CODE

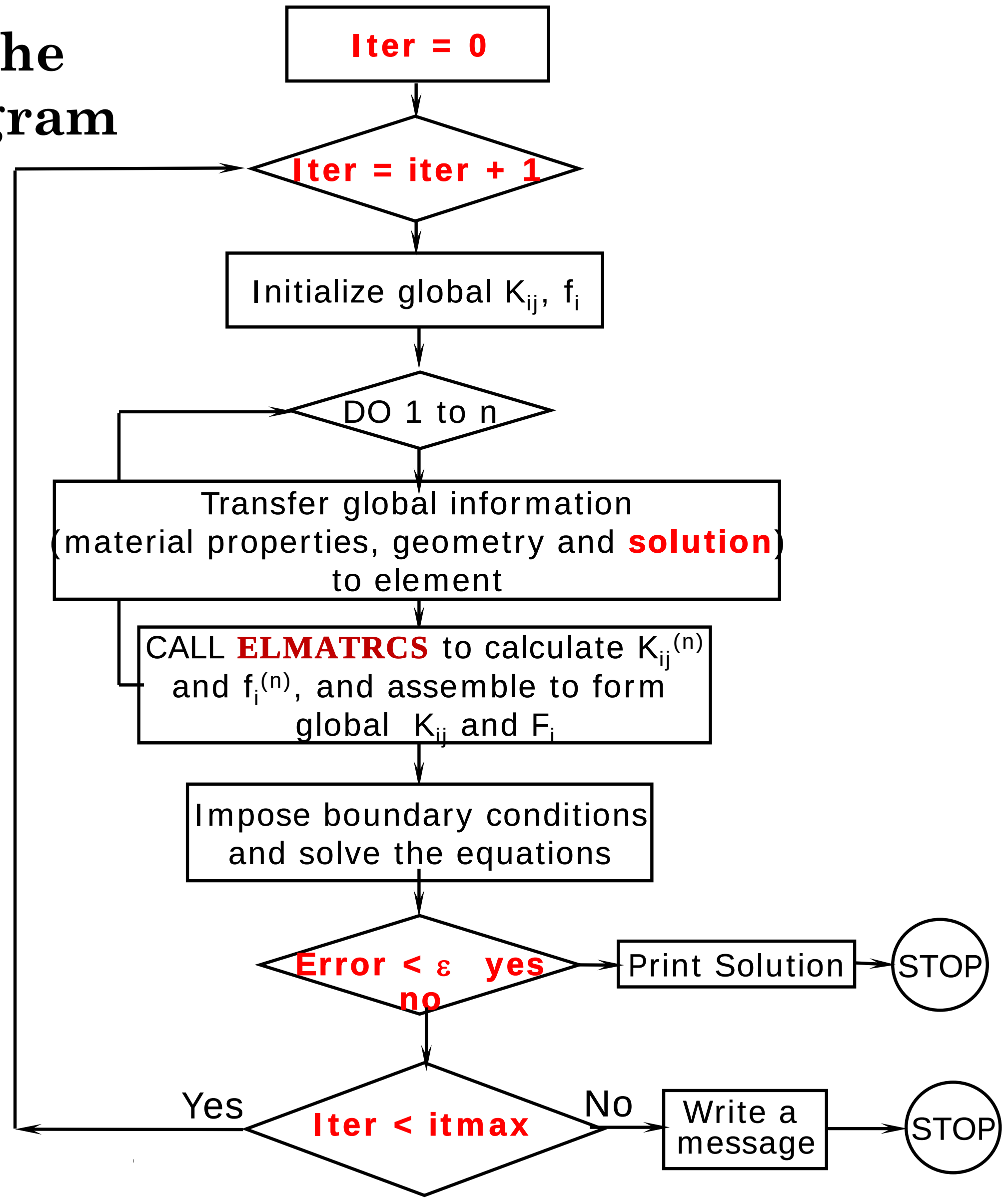


Flow Chart of a PROCESSOR Unit for *Linear Analysis*



GENERAL LOGIC IN A COMPUTER PROGRAM for Nonlinear Analysis

Logic in the MAIN program



FE PROGRAM FEM1DUNSYM (Sec. 4.5)

Variables used in the program

NPE - nodes per element, n

ELX(i) - Global coordinate of the ith node of element e, x_i^e

ELK(i, j) - Element coefficient, K_{ij}^e

ELF(i) - Element coefficient, f_i^e

$AX0, AX1$ – Coefficients in the definition of $a(x)$: $a(x) = AX0 + AX1 * x$

SFL(i) – Element shape (or approximation) funtion, ψ_i^e

DSFL(i) – Derivative of the ith shape function with respect to the local (normalized) coordinate ξ: $\dfrac{d\psi_i}{d\xi}$

GDSFL(i) – Derivative of the ith shape function with respect to x:

$$\psi_i = \psi_i(x(\xi)); \quad \Rightarrow \frac{d\psi_i}{d\xi} = \frac{d\psi_i}{dx}\frac{dx}{d\xi} = J\frac{d\psi_i}{dx}$$

$$\frac{d\psi_i}{dx} = (J)^{-1}\frac{d\psi_i}{d\xi} \quad \Leftrightarrow \quad \text{GDSF}(I) = \left(\text{GJ}\right)^{-1} \cdot \text{DSF}(I)$$

FINITE ELEMENT PROGRAM FEM1D-2

Numerical Integration

$$x = \sum_{j=1}^{n} x_j^e \psi_j^e(\xi) = \sum_{j=1}^{n} ELX(j) * SFL(j),$$

$$\int_{x_a}^{x_b} \left(a_e \frac{d\psi_i}{dx}\frac{d\psi_j}{dx} + b_e \psi_i \frac{d\psi_j}{dx} + c_e \psi_i \psi_j \right) dx$$

$$= \int_{-1}^{1} \left(c_e \psi_i(\xi)\psi_j(\xi) + b_e \psi_i \frac{1}{J_e}\frac{d\psi_j}{d\xi} + a_e \frac{1}{J_e}\frac{d\psi_i}{d\xi}\frac{1}{J_e}\frac{d\psi_j}{d\xi} \right) J_e \, d\xi$$

$$\approx \sum_{I=1}^{NGP} \hat{F}_{ij}(\xi_I) W_I = \sum_{I=1}^{NGP} \hat{F}_{ij} \, \mathrm{GAUSPT}(I, NGP) * GAUSWT(J, NGP)$$

$$\hat{F}_{ij}(\xi) = \left(c_e \psi_i(\xi)\psi_j(\xi) + b_e \psi_i \frac{1}{J_e}\frac{d\psi_j}{d\xi} + a_e \frac{1}{J_e}\frac{d\psi_i}{d\xi}\frac{1}{J_e}\frac{d\psi_j}{d\xi} \right) J_e,$$

$$\frac{d\psi_i}{dx} = \frac{d\xi}{dx}\frac{d\psi_i}{d\xi} = \frac{1}{J_e}\frac{d\psi_i}{d\xi}, \quad dx = J_e d\xi \quad \text{or} \quad J_e = \frac{dx}{d\xi} = 0.5 h_e$$

$GAUSPT(I, j)$ – Ith Gauss point, ξ_I, for the j-point Gauss rule

$GAUSWT(I, j)$ – Ith Gauss weight, W_I, for the j-point Gauss rule

FINITE ELEMENT PROGRAM FEM1D-3

The following statement should be inside a do-loop on number of Gauss points:

$$K_{ij} = \int_{x_a}^{x_b} \left[a(u,x)\frac{d\psi_i}{dx}\frac{d\psi_j}{dx} + b(u,x)\psi_i\frac{d\psi_j}{dx} + c(x,u)\psi_i\psi_j \right] dx$$

$$\approx \sum_{NI=1}^{NGP} \hat{F}_{ij}(\xi_{NI})W_{NI}$$

$$= \big[A \cdot GDSF(I) \cdot GDSF(J) + B \cdot SF(I) \cdot GDSF(J)$$

$$+ C \cdot SF(I) \cdot DSF(J) \big] \cdot \text{CNST}$$

$$\text{CNST} = \text{GJ} * \text{GAUSWT}(NI)$$

Define A, B, and C as given in the problem (you may assume a general form that is useful for a number of problems):

$$a(x,u) = ax0 + ax1 * x + axu * u + axdu * \frac{du}{dx} + \cdots$$

SUBROUTINE ELMATRCS1D-1

(see pages 196-198 of the text book)

```
      SUBROUTINE ELMATRCS1D (IEL,NPE,NONLIN,F0)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GAUSPT(5,5),GAUSWT(5,5)
      COMMON /SHP/ SFL(4),GDSFL(4)
      COMMON /STF/ ELK(3,3),ELF(3),ELX(3),AX0,AX1,
                   BX0,BX1,CX0,CX1,FX0,FX1,FX2
C
      DATA GAUSPT/5*0.0D0,–0.57735027D0,0.57735027D0,3*0.0D0,
     1  –0.77459667D0,0.0D0,0.77459667D0,2*0.0D0,–0.86113631D0,
     2  –0.33998104D0,0.33998104D0,0.86113631D0,0.0D0,
     3  –0.906180D0,–0.538469D0,0.0D0,0.538469D0,0.906180D0/

      DATA GAUSWT /2.0D0,4*0.0D0,2*1.0D0,3*0.0D0,0.55555555D0,
     1   0.88888888D0,0.55555555D0,2*0.0D0,0.34785485D0,
     2   2*0.65214515D0,0.34785485D0,0.0D0,0.2369227D0,
     3   0.478629D0,0.568889D0,0.478629D0,0.236927D0/
C
```

SUBROUTINE ELMATRCS1D-2

```
      NGP=IEL+1
      EL=ELX(IEL+1)-ELX(1)
      DO 10 I=1,NPE
        ELF(I)=0.0
        DO 10 J=1,NPE
        IF(NONLIN.GT.1)THEN
           TANG(I,J)=0.0
        ENDIF
10      ELK(I,J) = 0.0
        DO 50 NI=1,NGP
        XI=GAUSS(NI,NGP)
        CALL INTERPLN1D (ELX,GJ,IEL,NPE,XI)
        CNST=GJ*WT(NI,NGP)
        X=0.0
        U=0.0
        DU=0.0
        DO 20 I=1,NPE
        X=X+SFL(I)*ELX(I)
20      CONTINUE
        AX=AX0+AX1*X
        BX=BX0+BX1*X
        CX=CX0+CX1*X
        FX=FX0+FX1*X+FX2*X*X
```

$$\xi_{NI} = GAUSPT(NI, NGP)$$

$$CNST = J * w_{NI} = GJ * GAUSWT(NI)$$

$$x = \sum_{i=1}^{n} x_i^e \psi_i^e(\xi) = \sum_{i=1}^{n} ELX(i) * SFL(i)$$

$$a(x) = AX0 + AX1 * x + AXU * u$$

$$f(x) = FX0 + FX1 * x + FX2 * x * x$$

SUBROUTINE TO CALCULATE ELEMENT COEFFICIENTS-3

```
      IF(NONLIN.GT.0)THEN
        U=0.0
        DU=0.0
        DO 20 I=1,NPE
        U=U+SFL(I)*ELU(I)
        DU=DU+GDSFL(I)*ELU(I)
 20     CONTINUE
        AX=AX0+AX1*X+AU1*U+AUX1*DU+AU2*U*U+AUX2*DU*DU
        BX=BX0+BX1*X+BU1*U+BUX1*DU+BU2*U*U+BUX2*DU*DU
        CX=CX0+CX1*X+CU1*U+CUX1*DU+CU2*U*U+CUX2*DU*DU
        IF(NONLIN.GT.1)THEN
          AXT1=(AU1+2.0*AU2*U)*DU
          AXT2=(AUX1+2.0*AUX2*DU)*DU
          BXT1=(BU1+2.0*BU2*U)*DU
          BXT2=(BUX1+2.0*BUX2*DU)*DU
          CXT1=(CU1+2.0*CU2*U)*U
          CXT2=(CUX1+2.0*CUX2*DU)*U
        ENDIF
      ENDIF
```

$$u=\sum_{i=1}^{n} u_i^e \psi_i^e(\xi)=\sum_{i=1}^{n} ELU(i)*SFL(i)$$

Define parts of a, b, and c that depend on u and du/dx

$$a(x,u)=AX0+AX1*x+AU1*u+AUX1*\frac{du}{dx}+AU2*(u^2)+AUX2*\left(\frac{du}{dx}\right)^2$$

SUBROUTINE TO CALCULATE ELEMENT COEFFICIENTS-4

```
      DO 40 I=1,NPE
      ELF(I)=ELF(I)+FX*SFL(I)*CNST
         DO 40 J=1,NPE
            S00=SFL(I)*SFL(J)*CNST
            S10=GDSFL(I)*SFL(J)*CNST
            S11=GDSFL(I)*GDSFL(J)*CNST
            ELK(I,J)=ELK(I,J)+AX*S11+BX*S01+CX*S00
           IF(NONLIN.GT.1)THEN
              TANG(I,J)=TANG(I,J)+AXT1*S10+AXT2*S11+BXT1*S00
     *                 +BXT2*S01+CXT1*S00+CXT2*S01
           ENDIF
40    CONTINUE
50    CONTINUE
```

$$\int_{x_a}^{x_b} f(x)\psi_i \, dx =$$

$$\int_{-1}^{1} f(\xi)\psi_i(\xi) J \, d\xi = \sum_{NI=1}^{NGP} FX * SFL(i) * CNST$$

$$TANG(i,j) \equiv \sum_{k=1}^{n} \frac{\partial K_{ik}}{\partial u_j} u_k$$

$$\int_{x_a}^{x_b} \left[a(x,u)\frac{d\psi_i}{dx}\frac{d\psi_j}{dx} + b(x,u)\psi_i \frac{d\psi_j}{dx} + c(x,u)\psi_i\psi_j \right] dx$$

$$= \sum_{NI=1}^{NGP} [AX * S11 + BX * S01 + CX * S00]$$

SUBROUTINE TO CALCULATE ELEMENT COEFFICIENTS-5

```
C
C The residual vector and tangent coefficient matrix are calculated
C

      IF(NONLIN.GT.0 .AND. ITYPE.GT.1)THEN
        DO 60 I=1,NPE
          DO 60 J=1,NPE
60        ELF(I)=ELF(I)-ELK(I,J)*ELU(J)
        DO 80 I=1,NPE
          DO 80 J=1,NPE
80        ELK(I,J)=ELK(I,J)+TANG(I,J)
        ENDIF
        RETURN
        END
```

$$-R_i \equiv F_i - \sum_{j=1}^{n} K_{ij} u_j$$

$$T_{ij} \equiv K_{ij} + TANG(i,j)$$

MAIN PROGRAM
UPDATING AND SAVING SOLUTIONS

```
      CALL SLVUNSYM(GLK,MXNEQ,MXFBW,NEQ,NHBW)
      IF(NONLIN.EQ.0 .OR. NCOUNT.EQ.1)THEN
         WRITE(IT,395)F0
         WRITE(IT,350)(GLK(I,NBW),I=1,NEQ)
         IF(NONLIN.EQ.0)THEN
            STOP
         ENDIF
      ENDIF
C   Previous iteration solution is saved and current solution is updated
      DO 210 I=1,NEQ
      GP2(I)=GP1(I)
      IF(ITYPE.LE.1)THEN
        GP1(I)=GLK(I,NBW)
      ELSE
        GP1(I)=GP1(I)+GLK(I,NBW)
      ENDIF
  210 CONTINUE
```

MAIN PROGRAM: ERROR CHECK

```
C     Test for the convergence of the solution
         DNORM=0.0
         DINORM=0.0
         DO 220 IE=1,NEQ
         DNORM=DNORM+GP1(IE)*GP1(IE)
220      DINORM=DINORM+(GP1(IE)-GP2(IE))*(GP1(IE)-GP2(IE))
         TOLR=DSQRT(DINORM/DNORM)
         IF(TOLR.GT.EPS)THEN
            WRITE(IT,440)ITER,TOLR
            WRITE(IT,350)(GP1(I),I=1,NEQ)
            GOTO 80
        ELSE
            WRITE(IT,400)NL,F0
            WRITE(IT,420)ITER,TOLR
            WRITE(IT,350)(GP1(I),I=1,NEQ)
        ENDIF
  270 CONTINUE
        STOP
```

IMPOSOITION OF BOUNDARY CONDITIONS

NSPV – Number of specified primary variables of the problem.

ISPV(I,J) – Array containing the information about the global node number and the local degree of freedom that is specified.

ISPV(I,1) – For the Ith boundary condition, the global node number at which the BC is specified.

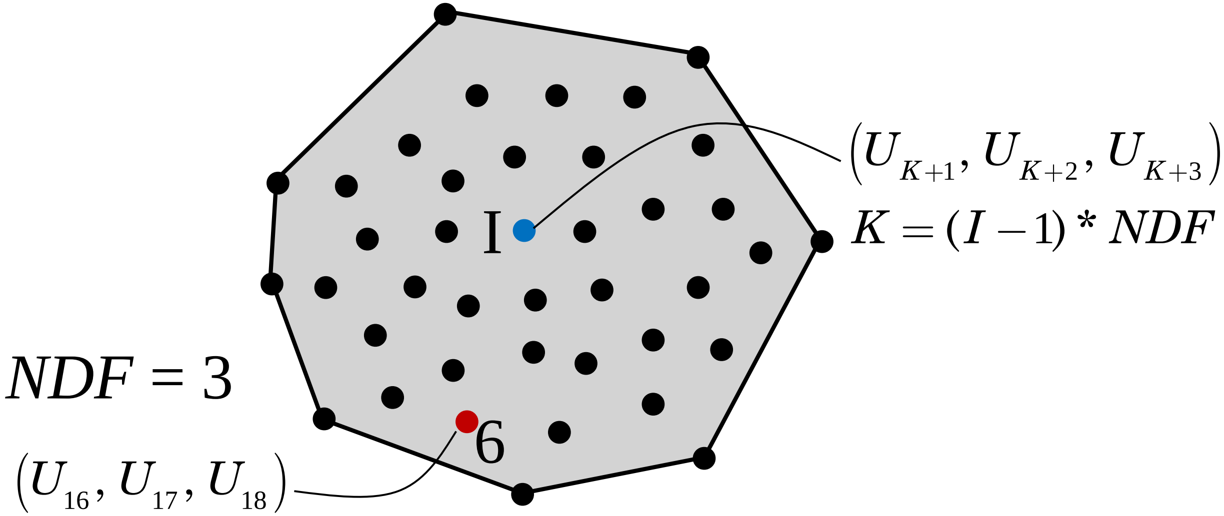
ISPV(I,2) – For the Ith boundary condition, the local degree of freedom that is specified.

VSPV(I) – The specified value of the deg. of freedom.

Similar meaning for NSSV, ISSV, and VSSV for specified secondary variables

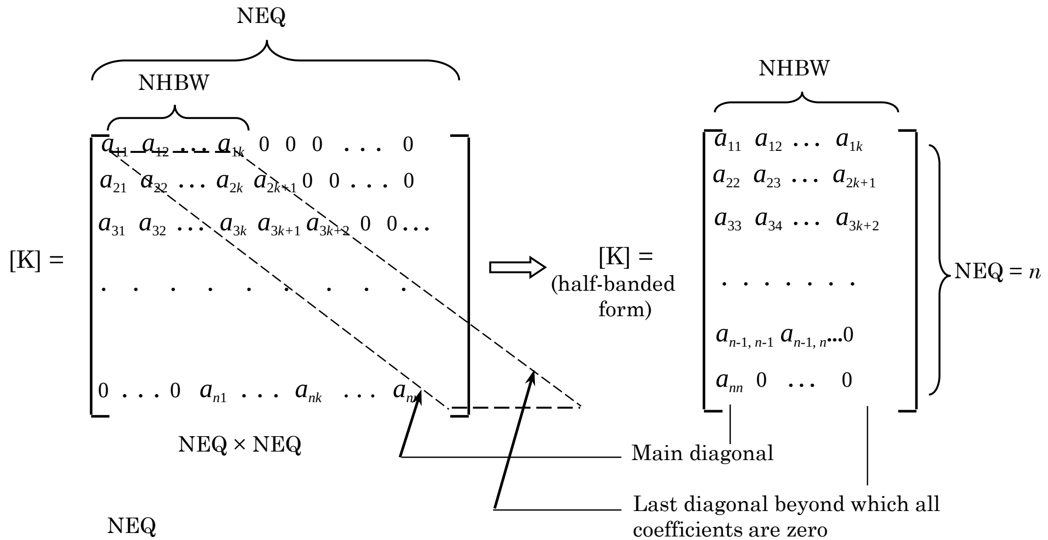
IMPOSOITION OF BOUNDARY CONDITIONS

NDF= number of primary degrees of freedom at a node

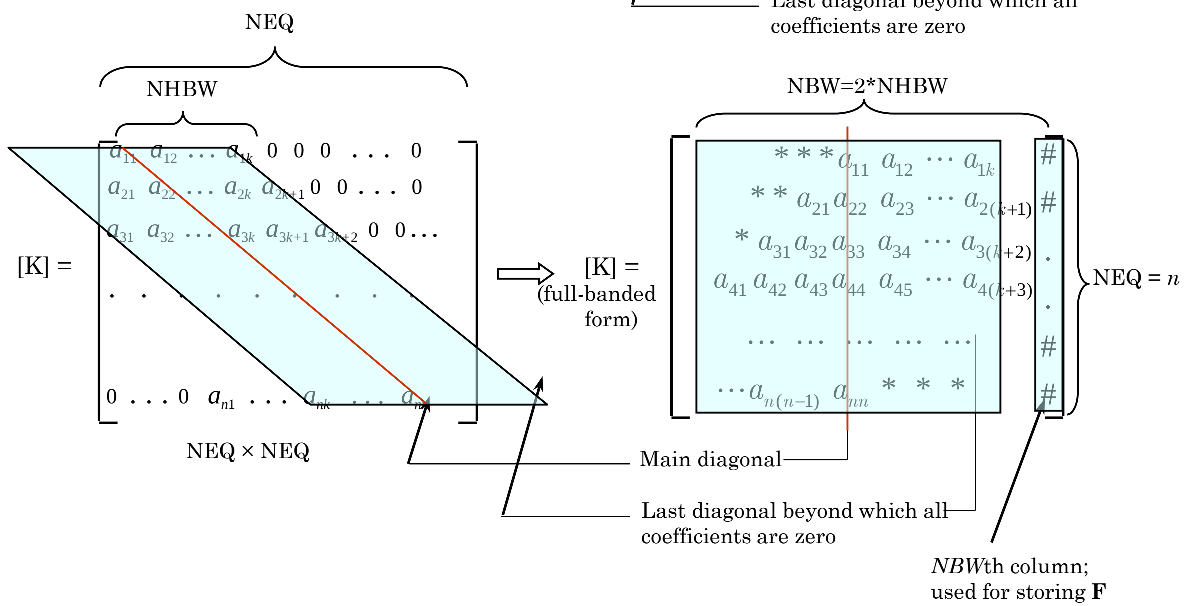


FORMATS OF ASSEMBLED EQUATIONS

BANDED *SYMMETRIC* SYSTEM (used in FEM1D)



BANDED *UNSYMMETRIC* SYSTEM



Example 1: 1-D Heat flow

$$-\frac{d}{dx}\left[k(T)\frac{dT}{dx}\right] = 0,\ \ 0 < x < L$$

$$T(0) = 500^{\circ}\text{K},\ \ T(L) = 300^{\circ}\text{K}$$

$$k(T) = k_0\left(1 + k_1 T\right),\ \ k_0 = 0.2\ \text{W/(m}^{\circ}\text{K)},\ k_1 = 2\times 10^{-3}\ {}^{\circ}K^{-1}$$

x	DI/NI Linear	Direct iteration 8L	Direct iteration 4Q	Newton iteration 8L	Newton iteration 4Q
0.0000	500.00	500.00	500.00	500.00	500.00
0.0225	475.00	477.24	477.24	477.24	477.24
0.0450	450.00	453.94	453.94	453.94	453.94
0.0675	425.00	430.06	430.06	430.05	430.05
0.0900	400.00	405.54	405.54	405.54	405.54
0.1125	375.00	380.35	380.35	380.34	380.34
0.1350	350.00	354.40	354.40	354.40	354.40
0.1575	325.00	327.65	327.65	327.65	327.65
0.1800	300.00	300.00	300.00	300.00	300.00

Example 2: Large deformation of a bar

Principle of virtual displacements

$$0 = \int_{V^e} \delta\varepsilon_{xx}^e \sigma_{xx}^e \, dA dx - \int_{x_a}^{x_b} \delta u \, f(x) dx - \delta u^e(x_a) P_a^e - \delta u^e(x_b) P_b^e$$

$$\varepsilon_{xx} = \frac{du}{dx} + \frac{1}{2}\left(\frac{du}{dx}\right)^2, \quad \delta\varepsilon_{xx} = \frac{d\delta u}{dx} + \frac{du}{dx}\frac{d\delta u}{dx}$$

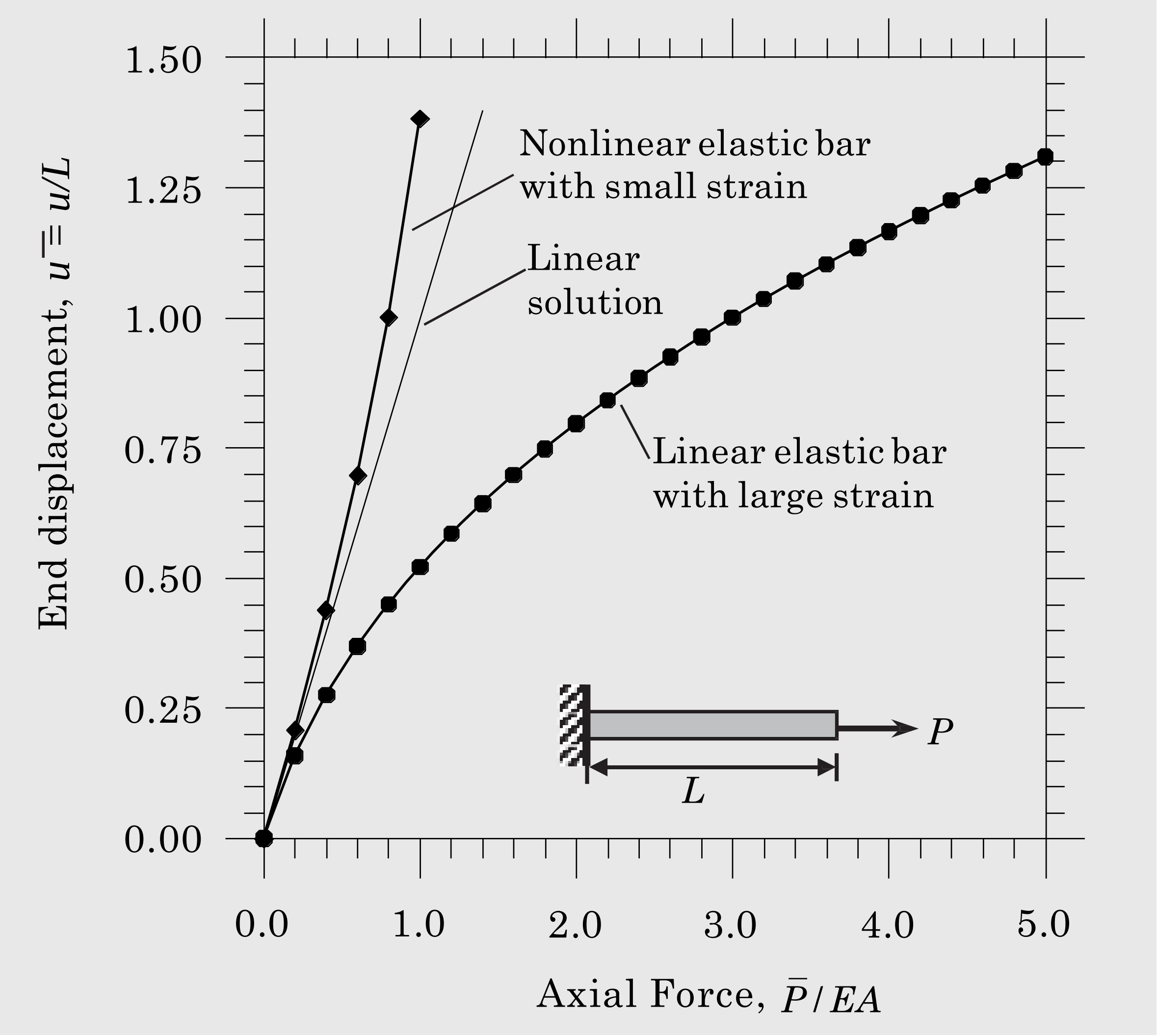
$$0 = \int_{x_a}^{x_b} \left(\frac{d\delta u}{dx} + \frac{du}{dx}\frac{d\delta u}{dx}\right) N_{xx} \, dx - \int_{x_a}^{x_b} \delta u \, f(x) dx$$

$$- \delta u(x_a) P_a - \delta u(x_b) P_b$$

$$P_a = -\left[\left(1 + \frac{du}{dx}\right) N_{xx}\right]_{x_a}, \quad P_b = \left[\left(1 + \frac{du}{dx}\right) N_{xx}\right]_{x_b}$$

Example 2: Large deformation of a bar (continued)

$$E = E_0(1 - \alpha\varepsilon_{xx}), \quad \varepsilon_{xx} = \frac{du}{dx}$$



SUMMARY

- Finite element formulation of a 1-D model nonlinear problem
- Solution of nonlinear equations (Picard and Newton)
- Calculation of tangent matrix coefficients
- General logic in a computer program
- Numerical examples
  - Nonlinear (material) nonlinear analysis of a 1D heat transfer problem
  - Nonlinear (geometric and material) analysis of a bar problem