Continuum structures

What does this mean to you?

Kim Lau Nielsen

Department of Civil and Mechanical Engineering

Section of Solid Mechanics
Technical University of Denmark

Overview of Day 5

8-10+: 3-part lecture:
- Intro to continuum modelling
- Element matrix for Q4 elements
- Introduction to Fortran code

10-11.00: Computer exercises:
- Get started with Code::Blocks

11-12:00: Introduction to Fortran (by Asger Nedergaard)
- In building 421 / 72 at 11:00-12:00

13-17: Working on DAY 5 exercises

But, before we start ….!

Please hand-in your solution to ”Assignment 1”

General FEM procedure

0) Discretize domain into a finite # of elements
(many ways – the more elements the better)

1) Node and element numbering

2) Calculate $[k^e]$ and assemble $[K]$

3) Insert BC

4) Solve for $\{D\}$

5) Compute ε, σ, N …
(interpolation of solutions between nodes) $\longrightarrow u$

6) Plot and evaluate results

Plane elasticity

2D linear elasticity under the assumption of small displacement gradients and strains

$$\varepsilon_{11} = \frac{\partial u}{\partial x}, \quad \varepsilon_{22} = \frac{\partial v}{\partial y} \quad \text{and} \quad \varepsilon_{12} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

The strain and stress vectors are written as

$$\{\varepsilon\} = \left\{ \begin{array}{c} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{array} \right\} \quad \text{and} \quad \{\sigma\} = \left\{ \begin{array}{c} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{array} \right\}$$

Hooke's law (constitutive relation)

$$\{\sigma\} = [C]\{\varepsilon\}$$

Originates from Hookes Generalized Law

Isotropic materials and plane stress

$$[C]_{plane\ stress} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix}$$

Isotropic materials and plane strain

$$[C]_{plane\ strain} = \frac{E}{(1-2\nu)(1+\nu)}\begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & (1-2\nu)/2 \end{bmatrix}$$

ν is Poisson's ratio and E is the Young's modulus.

Plane elasticity

Stress-strain relation

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix} = [C] \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{Bmatrix} \quad \text{with} \quad [C]_{plane\ stress} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix}$$

Generalized Hookes law for <u>2D plane stress</u>

$$\sigma_{11} = \frac{E}{1-\nu^2}[\epsilon_{11} + \nu\epsilon_{22}] - \cancel{\frac{E}{1-\nu}\hat{\alpha}\Delta T}$$

$$\sigma_{22} = \frac{E}{1-\nu^2}[\epsilon_{22} + \nu\epsilon_{11}] - \cancel{\frac{E}{1-\nu}\hat{\alpha}\Delta T}$$

$$\sigma_{12} = \frac{E}{1+\nu}\epsilon_{12}$$

$$\cancel{\sigma_{3i} = 0}$$

Plane elasticity

Small strain (small displacement) assumption (in 2D plane stress):

$$\varepsilon_{11} = \frac{\partial u}{\partial x}, \quad \varepsilon_{22} = \frac{\partial v}{\partial y} \quad \text{and} \quad \varepsilon_{12} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

In a vector/matrix notation this can be written as:

$$\{\varepsilon\} = \left\{ \begin{array}{c} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{array} \right\} = \left\{ \begin{array}{c} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{array} \right\} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \left\{ \begin{array}{c} u \\ v \end{array} \right\}$$

The displacement field is the unknown in a FE model

2D plane element (Q4)

- Element displacement vector

$$\{d\}^T = \{u_1, v_1, u_2, v_2, u_3, v_3, u_4, v_4\}^T$$

- Internal element displacements

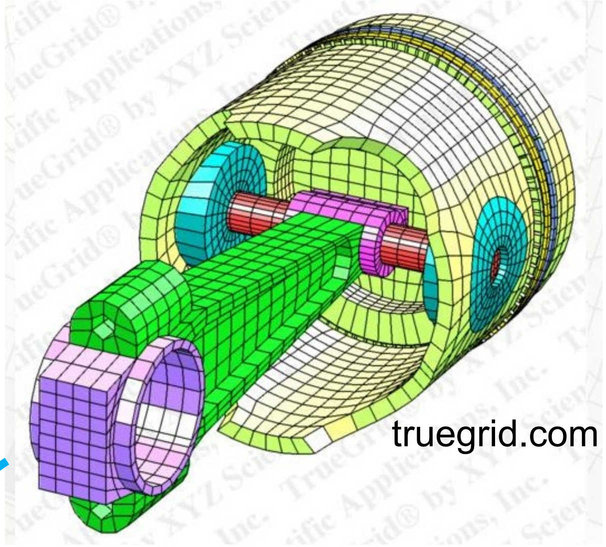
$$u(x,y) = \sum_{i=1}^{n} N_i(x,y)\; u_i$$

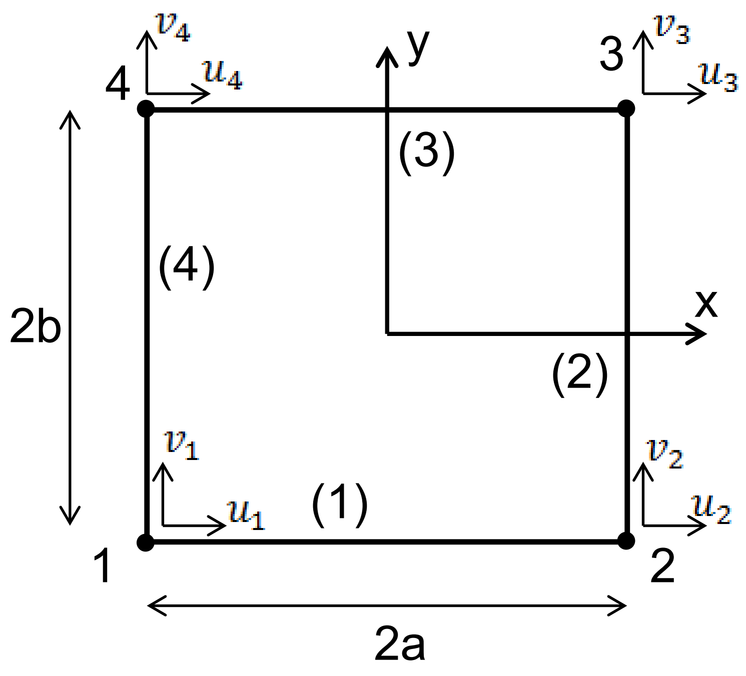
$$v(x,y) = \sum_{i=1}^{n} N_i(x,y)\; v_i,$$

- Shape functions

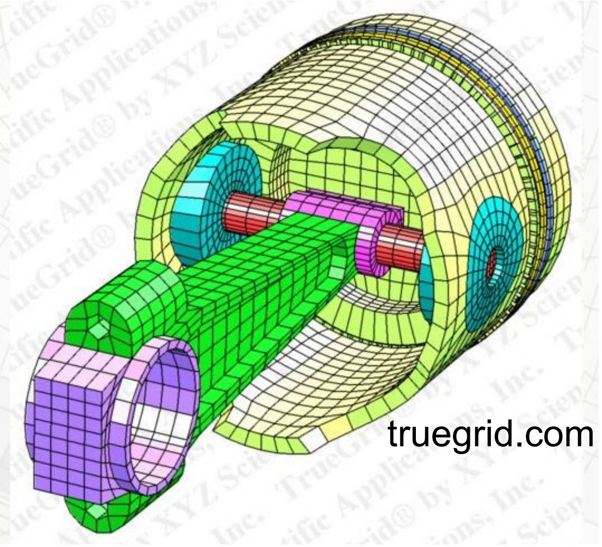
$$N_1 = \frac{(a-x)(b-y)}{4ab} \quad , \quad N_2 = \frac{(a+x)(b-y)}{4ab}$$

$$N_3 = \frac{(a+x)(b+y)}{4ab} \quad , \quad N_4 = \frac{(a-x)(b+y)}{4ab}$$





2D plane element (Q4)



- Internal element displacement on vector form

$$\{\mathbf{u}\} = \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & \dots & 0 \\ 0 & N_1 & 0 & \dots & N_n \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ \vdots \\ v_n \end{Bmatrix}$$

$$= [N]\{d\}$$

Recall that;

$$\{\varepsilon\} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix}$$

- Relation between strain and displacements

$$\{\varepsilon\} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} [N]\{d\} \longrightarrow \{\varepsilon\} = \underbrace{\frac{1}{4ab} \begin{bmatrix} -(b-y) & 0 & (b-y) & \dots & 0 \\ 0 & -(a-x) & 0 & \dots & (a-x) \\ -(a-x) & -(b-y) & -(a+x) & -\dots & (b+y) \end{bmatrix}}_{[\bar{B}]=[B_0]} \{d\}$$

or in short; $\{\varepsilon\} = [\bar{B}]\{d\}$ (and the stresses are; $\{\sigma\} = [C][\bar{B}]\{d\}$)

2D plane element (Q4) – Developing the FE method

- Virtual work principle ($\delta\Omega = \delta W$)

$$\int_V \{\delta\varepsilon\}^T \{\sigma\}\, dV = \int_S \{\delta\mathbf{u}\}^T \{F\}\, dS + \int_V \{\delta\mathbf{u}\}^T \{\Phi\}\, dV + \sum_i \{\delta\mathbf{u}\}_i^T \{p\}_i$$

$$\boxed{\text{which has to hold for any kinematically admissible displacement variations } \{\delta\mathbf{u}\}, \text{ associated strain variations } \{\delta\varepsilon\} \text{ and nodal displacement variations } \{\delta D\}.}$$

Important to realize

Where;

$\delta\Omega$ is variation in internal energy
δW is variation in external work
V is structural volume
S is surface area
$\{F\}$ is surface traction
$\{\Phi\}$ is body force
$\{\delta\mathbf{u}\}$ is virtual displacement variation
$\{\sigma\}$ is the stress vector
$\{\delta\varepsilon\}$ is the virtual strain variation vector
$\{p\}_i$ is a concentrated (nodal) load vector

2D plane element (Q4) – Developing the FE method

- Virtual work principle ($\delta\Omega = \delta W$)

$$\underbrace{\int_V \{\delta\varepsilon\}^T \{\sigma\}\, dV}_{\text{Internal (elastic) energy}} = \underbrace{\int_S \{\delta\mathbf{u}\}^T \{F\}\, dS}_{\text{Work of surface traction}} + \underbrace{\int_V \{\delta\mathbf{u}\}^T \{\Phi\}\, dV}_{\text{Work of volume forces}} + \underbrace{\sum_i \{\delta\mathbf{u}\}_i^T \{p\}_i}_{\text{Work of concen-trated loads}}$$

- Variation in displacements and strains;

$$\{\delta\mathbf{u}\} = [N]\{\delta d\} \quad , \quad \{\delta\varepsilon\} \equiv [\overline{B}]\{\delta d\}$$

- Neglect surface tractions, volume forces, and consider only <u>one single element</u>;

$$\sum_e \int_{V^e} \{\delta d\}^T [B_0]^T \{\sigma\}\, dV = \{\delta D\}^T \{P\}$$

2D plane element (Q4) – Stiffness matrix

- Neglect surface tractions, volume forces, and consider only one single element;

$$\sum_e \int_{V^e} \{\delta d\}^T [B_0]^T \{\sigma\}\, dV = \{\delta D\}^T \{R_{ext}\}$$

- If this should hold for any admissible displacement field, then;

$$\sum_e \int_{V^e} [B_0]^T \{\sigma\}\, dV = \{R_{ext}\}$$, or; $$\sum_e \int_{V^e} [B_0]^T [C][B_0]\{d\}\, dV = \{R_{ext}\}$$

- For the global system we then have;

$$\sum_e \underbrace{\int_{V^e} [B_0]^T [C][B_0]\, dV}_{[k^e]} \{D\} = \{R_{ext}\} \quad \longrightarrow \quad [K]\,\{D\} = \{R_{ext}\}$$

- If we include surface and volumen forces;

$$\{R_{ext}\} = \{P\} + \sum \{r^e\}$$, with; $$\{r^e\} = \int_{S^e} [N]^T \{F\}\, dS + \int_{V^e} [N]^T \{\Phi\}\, dV$$

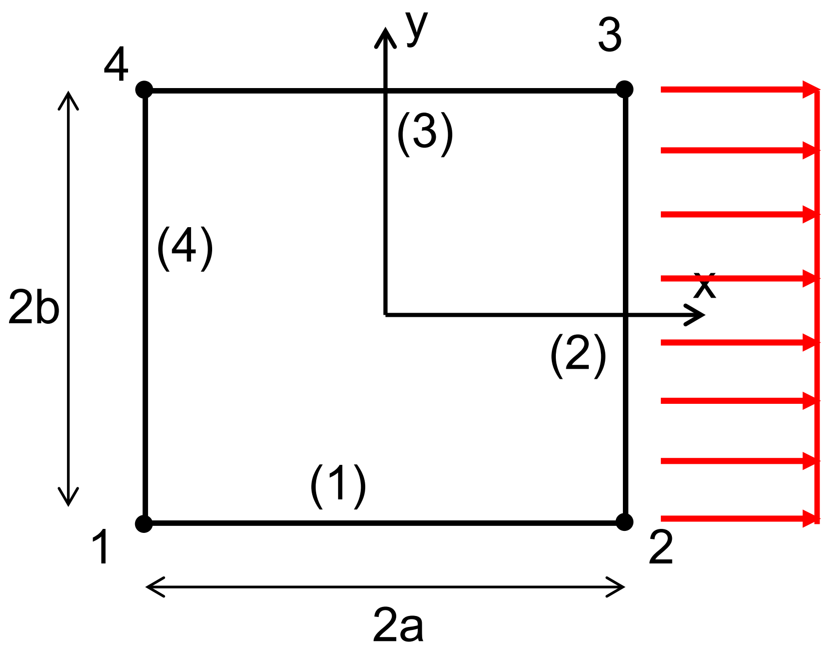
2D plane element (Q4) – Surface traction

- Exampel of distributed surface traction

$$\{r^e\} = \int_{S^e} [N]^T \{F\}\, dS$$

$$\downarrow$$

$$\{r_e\} = t \int_{-b}^{b} [N]^T|_{x=a} \{F\}\,\mathrm{d}y$$



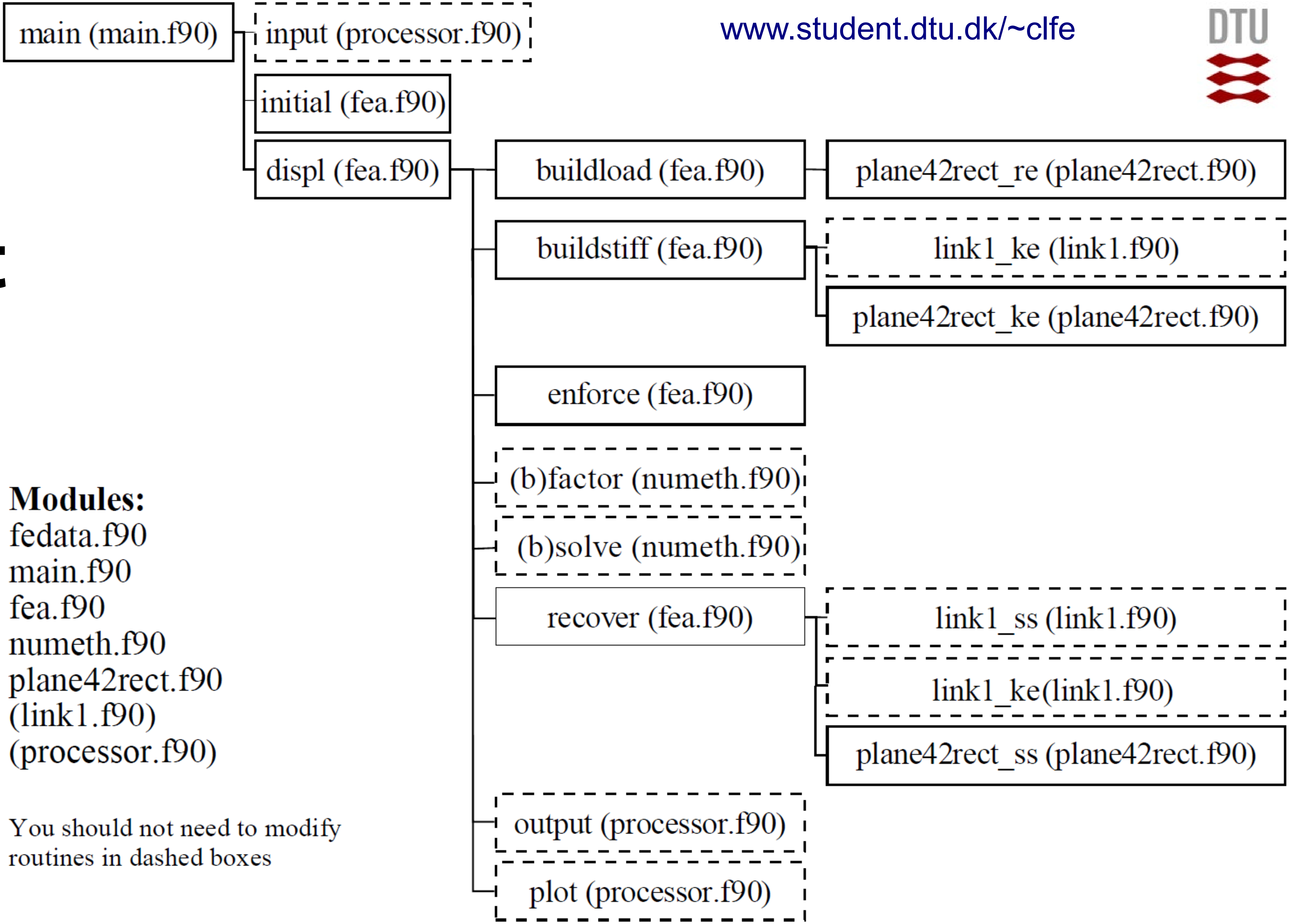
with;

$$\{F\} = \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} = \begin{Bmatrix} F \\ 0 \end{Bmatrix}$$

This yield;

$$\{r_e\} = t \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ b & 0 \\ 0 & b \\ b & 0 \\ 0 & b \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} F \\ 0 \end{Bmatrix} = tbF \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Fortran flowchart



www.student.dtu.dk/~clfe

Modules:
fedata.f90
main.f90
fea.f90
numeth.f90
plane42rect.f90
(link1.f90)
(processor.f90)

You should not need to modify routines in dashed boxes

Input file (truss)

```
/TITLE, TRUSS GROUND STRUCTURE
! GSTRUCT Version: 1.8
! File: example1
! User: sigmund
! Course: MEK 41525
! Date: 220902
! Time: 16:02
! Element type: LINK1
! No. of Nodes:        3
! No. of Elements:     3
/PREP7
! Define element type: ET, type #, name
ET, 1, LINK1
! Define material property: MP, property, card #, value
MP,EX,1,  1.00000000
! Define real constant (area) property: R, card #, value
R,1,  1.00000000
! Define nodal coordinate: N, node #, x coord, y coord, z coord
N,        1,  0.00000000    ,  0.00000000    , 0.
N,        2,  0.00000000    ,  1.00000000    , 0.
N,        3,  1.00000000    ,  0.00000000    , 0.
N,        4,  1.00000000    ,  1.00000000    , 0.
! Define element connectivity: EN, element #, nodal list
EN,        1,        1,        3
EN,        2,        1,        2
EN,        3,        2,        3
EN,        4,        3,        4
EN,        5,        2,        4
! Define boundary support conditions: D, node #, dof label,
value
D,        1,UX,     0.000000000
D,        1,UY,     0.000000000
D,        2,UX,     0.000000000
! Define load conditions: F, node #, dof label, value
F,        3,FY,    -.1000000000
F,        4,FX,     .1000000000
FINISH
```

Input file (continuum)

```
/TITLE, CONTINUUM GROUND STRUCTURE
! GSTRUCT Version: 1.8
! File: continuum1
! User: sigmund
! Course: MEK 41525
! Date: 220902
! Time: 16:24
! Element type: PLANE42
! No. of Nodes:          9
! No. of Elements:       4
/PREP7
! Define element type: ET, type #, name
ET,1,PLANE42
! Define material property: MP, property, card #, value
MP,EX,1,  1.00000000
MP,PRXY,1, 0.300000000
MP,DENS,1,  1.00000000
! Define nodal coordinate: N, node #, x-coord, y-coord, z-coord
N,        1,  0.00000000    ,  0.00000000    , 0.
N,        2,  0.00000000    ,  1.00000000    , 0.
N,        3,  0.00000000    ,  2.00000000    , 0.
N,        4,  1.00000000    ,  0.00000000    , 0.
N,        5,  1.00000000    ,  1.00000000    , 0.
N,        6,  1.00000000    ,  2.00000000    , 0.
N,        7,  2.00000000    ,  0.00000000    , 0.
N,        8,  2.00000000    ,  1.00000000    , 0.
N,        9,  2.00000000    ,  2.00000000    , 0.
! Define element connectivity: EN, element #, nodal list
EN,       1,       1,       4,       5,       2
EN,       2,       2,       5,       6,       3
EN,       3,       4,       7,       8,       5
EN,       4,       5,       8,       9,       6
! Define boundary support conditions: D, node #, dof label, value
D,        1,UX,     0.000000000
D,        1,UY,     0.000000000
D,        2,UX,     0.000000000
D,        2,UY,     0.000000000
! Define nodal load conditions: F, node #, dof label, value
F,        8,FX,     .01000000000
F,        9,FX,     .01000000000
! Define surface load conditions: SFE, element #, face #, PRES, 0, value
FINISH
```

Matlab to Fortran conversion

	MATLAB	FORTRAN90
Editor	Built-in	Built-in
Compilation	Run-time compilation	Compile by `Ctrl+Shift+b` and run by `Ctrl+F5`
Cases	Case sensitive ($K \neq k$)	Case in-sensitive ($K = k$)
Variables	Scalar variables need not be defined	All variables must be defined: integer :: n real(8) :: length real(8) :: a, f(17), k(13,13) real(8), dimension(17) :: f logical :: banded
Dynamic allocation	NA	real(8), dimension(:,:), allocatable :: k ⋮ allocate (k(neqn, neqn))
Modules	NA	May contain "private" and "public" subroutines
Types	We did not use it	see under "FEM specific remarks"
Subroutine calls	[areas] = bisect(areas,amin, ...); function [areas] = bisect(aold,amin, ...) ⋮	call bisect(areas, amin, ...) subroutine bisect(aold, amin, ...) real(8), intent(in) :: amin, ... real(8), dimension(:), intent(inout) :: aold ⋮

Matlab to Fortran conversion

	MATLAB	FORTRAN90
do loops	*for* $i = 1 : n$ ... *end*	*do* $i = 1, n$ ... *end do*

if statements	*if* (...)
	...
	end
		if (...) *then*
		...
		end if

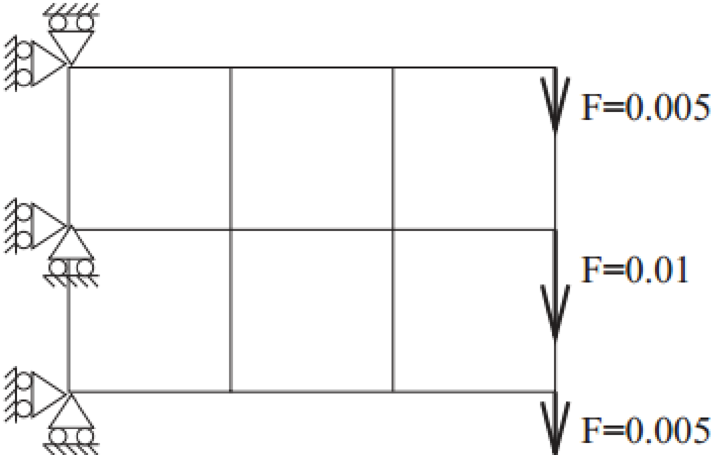
Indexing	K(edof,edof) = K(edof,edof) + k	k(edof,edof) = k(edof,edof) + ke
Vector multiplication	a = b'*c	a = dot_product(b,c)
Matrix multiplication	a = b*c	a = matmul(b,c)
printing to screen	remove ";"	print*,'aout=',aout
print matrix	K	do i = 1,neqn print "(24(f4.2,tr1))",k(i,1:neqn) end do
# of colums in matrix	size(K,2)	size(k,2)
Integer division	i=1, j=3 => i/j = 0.3333	i=1, j=3 => i/j = 0 (NB! Integer part of result!)

Exercises 5.0

The handout Fortran code (available on DTU Learn) works on Windows platforms and we suggest using Code::Blocks along with the build-in GNU Compiler. The Code::Blocks platform is pre-installed on the computers in the databar. For Windows-users a “Getting started with Code::Blocks” instruction can be found in the same zip-file holding the code. Other platform/compiler may also be used on your Laptop but here you must be able to set the system up yourselves.

To get started with today’s exercises:

- Download and extract the `"Day5basisCode.zip"` file from the course homepage to a folder on your X-drive.
- Unzip the file and find the pdf-file “Getting started with Code::Blocks” and follow the instructions herein.
- The file “FEM_fortran.cbp” holds a predefined project which you need to load into Code:Blocks.



Exercises 5.1

Implement distributed (traction) loads in the Fortran program and verify with comparisons to an analytical solution.

The input file statement for surface tractions (pressure) is

SFE,elem#,face#,PRES,0,15.	apply pressure of 15 across face face# of element elem#

The "input" subroutine converts this line to a row in the `loads` matrix e.g. (2,elem#,face#,15), where the first number is the code (2) for surface loads.

For modeling of surface tractions, element face numbers are 1, 2, 3 and 4, corresponding to the lower, right, upper and left edges. Note that the commercial FE-software ANSYS only can model surface tractions as pressure loads, i.e. perpendicular to the surface. Since we try to follow the ANSYS input format, our input format does the same. The pressure loads are always positive into the cell.

Exercises 5.2

Compute element strains and stresses (e.g. ϵ_{11}, σ_{22}, etc.) for the simple test case in Figure 5.2 and compare with analytical values. **Hint:** Compute strains and stresses (in the centroid ($x = y = 0$)) in the subroutine "plane42rect_ss". Call the plotting routine "plotmatlabeval" with a vector containing the stresses (from the "displ" subroutine).
The "plane42rect_ss" routine is called with the command-line

```
call plane42rect_ss(xe, de, young, nu, estress, estrain)
```

where the input variables are element nodal coordinates, Young's modulus and Poisson's ratio and the outputs are the element stress and strain vectors.

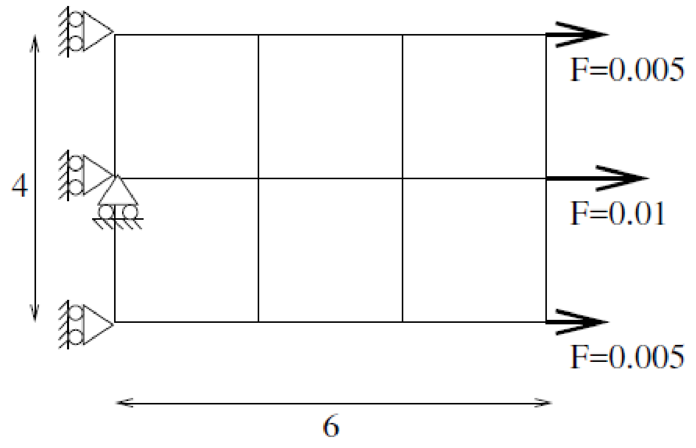


Figure 5.2: 6-element cantilever beam.

*Exercises 5.3

Implement volume loads and demonstrate this by an example. **Hint:** Write a subroutine very similar to the "plane42rect_re" subroutine. Use also that

$$[\text{Nvol}] = \int_{a}^{-a} \int_{b}^{-b} [N]^T \, dy \, dx = \begin{bmatrix} ab & 0 & ab & 0 & ab & 0 & ab & 0 \\ 0 & ab & 0 & ab & 0 & ab & 0 & ab \end{bmatrix}. \qquad (5.24)$$

The input-file syntax for volume (acceleration) loads is

MP,DENS,1,7.1	define material specific density of 7.1 (kg/m^3)
ACEL,ax,ay,0	apply acceleration in x-direction (ax) or y-direction (ay)

The acceleration values are saved in the vector `accel`={ax ay}.

Exercises

- Feedback on “Assignment 1” will be available at lecture start next week

- REMEMBER:

“How to get going in FORTRAN” by Asger Nedergaard
Build 421, Aud. 72 at 11:00-12:00

Please participate to get some tips and tricks…