Preparation of input files, understanding of solver settings, element types, boundary conditions, material models, problem types

Outline

- Preparation of input files
- Understanding of oofem modules
- Available analyses
- Element and material libraries

OOFEM input

- Input file is text file with specific structure (described in OOFEM Input manual)
- Input file can be prepared manually by any text editor, generated by special purpose code, preprocessor or converter
- Input file consists of several sections
  - Ordering of sections is compulsory
  - Each section can consist of one or more records (corresponding to lines)
- Input file can contain comments (lines beginning with '#' character)
- Long Input records (lines) can be splitted using continuation character at the end of each line '\'
- Input file supports @include directive
- Content is not case sensitive

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

OOFEM Input – syntax of records

General record syntax

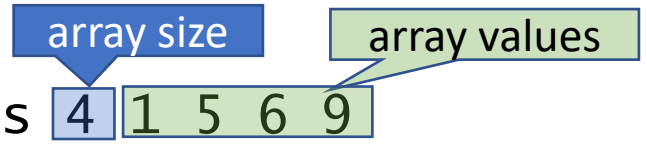
```
Record_kwd [label] [Attr1_kwd #(type) ] … [AttrX_kwd #(type)]
```

Order of attributes is optional

Supported attribute types

- `in – integer number`
- `rn – real number`
- `ia – integer array`
- `ra – real array`

```
val 2
pi 3.141592
nodes 4 1 5 6 9
coords 3 1.0 4.2 8.1
```



More types exist, consult oofem input manual

OOFEM Input – syntax of records

Example

Record format:

```
Particle #(in) color #(in) mass #(rn) coords #(ra) [name #(s)]
```

Valid input records

```
Particle 2 color 5 mass 0. 1 8 coords 3 0. 0 1. 0 2. 0 name ”P136”
Particle 2 mass 0. 1 8 coords 3 0. 0 1. 0 2. 0 color 5
```

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Output file record

String containing system path to output file

Examples:

```
/home/user/work/oofem/job1/results.oofem.out
C:\Users\user\Documents\job1\results.out
```

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Job description record

String containing job description

Examples:

```
Linear analysis of cantilever beam
```

```
Three point bending analysis, l=300mm, E=30 GPa
```

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Analysis record

Describes type of analysis to be performed

Example: Record format for linear static analysis

```
LinearStatic nsteps #(in) [sparselinsolverparams
#(...)]\ [sparselinsolverparams #(...)]

EigenValueDynamic nroot #(in) rtolv #(rn) \
[eigensolverparams #(...)]
```

Example of valid records:

```
LinearStatic nsteps 7

EigenValueDynamic nroot 3 rtolv 1.e-6
```

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Domain record

Describes type of domain (determines the default DOFs)

Domain record format

```
domain *domainType
```

Examples of valid records

```
domain 3d
domain 2dPlaneStress
domain 2dIncompFlow
```

Note: no longer relevant, DOFs assigned dynamically according to element needs

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Output manager record

Controls output. It can filter output to specific solution steps, and within these selected steps, allows also to filter output only to specific DOF managers and elements.

Record format

```
OutputManager [tstep all] [tstep step #(in)] [tsteps out #(rl)] \
  [dofman all] [dofman output #(rl)] [dofman except #(rl)] \
  [element all] [element output #(rl)] [element except #(rl)
```

Examples of valid records

```
Outputmanager tstep_all dofman_all element_all
Outputmanager tstep_step 10 dofman_all
Outputmanager tstep_all dofman_output {(1 50) (200 300)} \
element_except {(1 100)}
```

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Component sizes record

Describes the number of domain components.

Record format

```
ndofman #(in) nelem #(in) ncrosssect #(in) nmat #(in) nbc #(in) \
nic #(in) nltf #(in) nset #(in) [nbarrier #(in)]
```

- *ndofman* is number of DOF managers (nodes),
- *nelem* number of nodes,
- *ncrosssect* number of cross sections,
- *nmat* number of material models,
- *nbc* number of boundary conditions,
- *nic* number of initial conditions,
- *nltf* number of (time) functions,
- *nset* number of sets
- *nbarrier* number of nonlocal bariers.

Example of valid record

```
ndofman 10 nelem 5 nic 0 nbc 2 nltf 2 ncrosssect 1 nmat 1 nset 1
```

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Node record(s)

Describes the properties of individual DOF managers (nodes)

Example of Record format

```
*DofManagerType (num#)(in) [load #(ra)] [DofIDMask #(ia)] \
[doftype #(ia) masterMask #(ia)]
```

Example of supported DofManagerType:

```
Node coords #(ra) [lcs #(ra)]
```

Example of valid records

```
Node 2 coords 3 0. 0. 1.
Node 3 coords 3 1. 1. 2. lcs 6 0. 1. 0.  0. 0. 1.
```

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Element record(s)

Describes the type and properties of individual elements

Example of Record format

```
*ElementType (num#)(in) [mat #(in) crossSect #(in)] nodes #(ia) [activityltf #(in)]
```

Example of supported element types

Planestress2d [NIP #(in)]

Beam2d [dofstocondense #(ia)]

Example of valid records

```
Planestress 1 nodes 4 1 2 5 6 activityltf 2
Beam2d 2 nodes 2 3 4 mat 1 crosssect 2
Beam2d 3 nodes 2 7 8 dofstocondense 1 3
```

Notes:

- material and crosssections can be specified using sets instead of mat and crosssect attributes
- Refer to Element library manual for a full list of element types

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Cross section record(s)

Describe type and parameters of individual cross section models

Example of Record format

```
*CrossSectType (num#)(in) [material #(in) set #(in)]
```

where CrossSectType can be, for example

```
SimpleCS [thick #(rn)] [width #(rn)] [area #(rn)] [iy #(rn)]\
  [iz #(rn)] [ik #(rn)] [shearareay #(rn)] [shearareaz #(rn)]\
  beamshearcoeff #(rn)
```

Example of valid records

```
Simplecs 1 thick 0.1
SimpleCS 1 area 1.e3 Iy 0.0026244 beamShearCoeff 1.e18 material 1 set 1
```

Refer to oofem Input manual for a list of supported cross section models

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Material model record(s)

Describe type and parameters of individual constitutive models

Example of Record format

```
*MaterialType num #(in) d #(rn)
```

where MaterialType can, for example, be

IsoLE E #(rn) n #(rn) tAlpha #(rn)

Example of valid records

```
IsoLE 1 d 1. E 25.e6 n 0.2 tAlpha 1.2e-5
```

Refer to oofem Material manual for list of available material models

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Boundary conditions record(s)

Describe type and attributes of applied boundary conditions

Example of Record format

```
*EntType (num#)(in) loadTimeFunction #(in) [set #(in)] [valType #(in)]
[dofs #(ia)] [isImposedTimeFunction #(in)]
```

where `EntType` can, for example, be

```
BoundaryCondition prescribedvalue #(rn) [d #(rn)]
NodalLoad components #(ra) [cstype #(in)]
```

Example of valid records

```
BoundaryCondition 2 loadTimeFunction 1 dofs 1 5 values 1 0 set 5
NodalLoad 6 loadTimeFunction 1 dofs 3 1 3 5 Components 3 7.0 0.0 0.0 set 5
```

Refer to oofem Input manual for complete list of available boundary conditions

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

(Time) functions record(s)

Describe (time) functions used to describe (time) variation of certain inputs.

Example of Record format

```
ConstantFunction f(t) #(rn)
PiecewiseLinFunction [nPoints #(in) t #(ra) f(t) #(ra)]\
[datafile #("string")]
```

Example of valid records

```
ConstantFunction 1 f(t) 1.
PiecewiseLinFunction 2 npoints 3 t 3 0 10 100 f(t) 3 0. 1.
1.
```

Refer to oofem Input manual for complete list of available boundary conditions

OOFEM Input – sections

1. **output file**
2. **job description**
3. **analysis record**
4. export module record(s)
5. **domain record**
6. **output manager record**
7. **components size record**
8. **node record(s)**
9. **element record(s)**
10. **cross section record(s)**
11. **material type record(s)**
12. nonlocal barriers record(s)
13. **load, boundary conditions record(s)**
14. initial conditions record(s)
15. **time functions record(s)**
16. optional xfem manager and associated record(s)
17. **set record(s)**

Set record(s)

Sets specify regions of the geometry as a combination of volumes, surfaces, edges, and nodes. The main usage of sets are to connect regions of elements to a given cross section or apply a boundary condition, though sets can be used for many other things as well.

Example of Record format

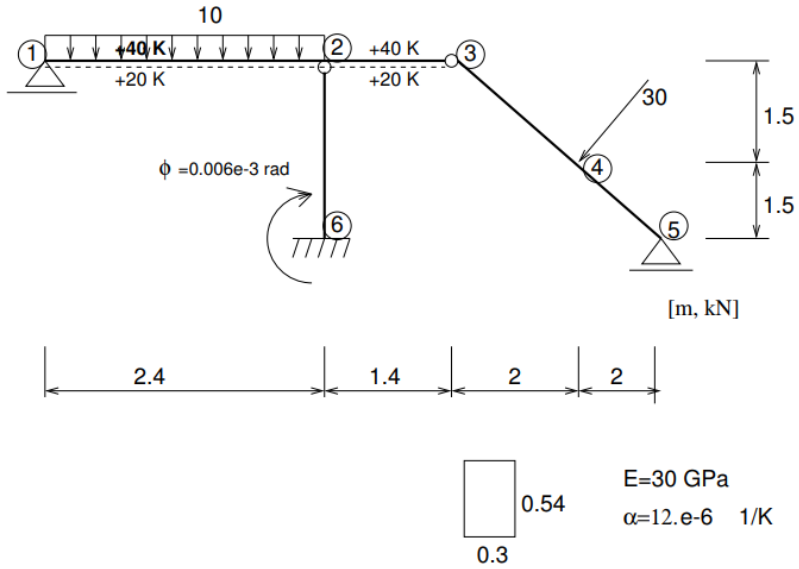
```
Set (num#)(in) [elements #(ia)] [elementranges #(rl)] \
 [allElements] [nodes #(ia)] [noderanges #(rl)] [allNodes] \
 [elementboundaries #(ia)] [elementedges #(ia)]
```

Example of valid records

```
Set 1 elementranges {(1 20)}
Set 2 nodes 2 1 2
Set 3 elementedges 4 1 1 3 1
```

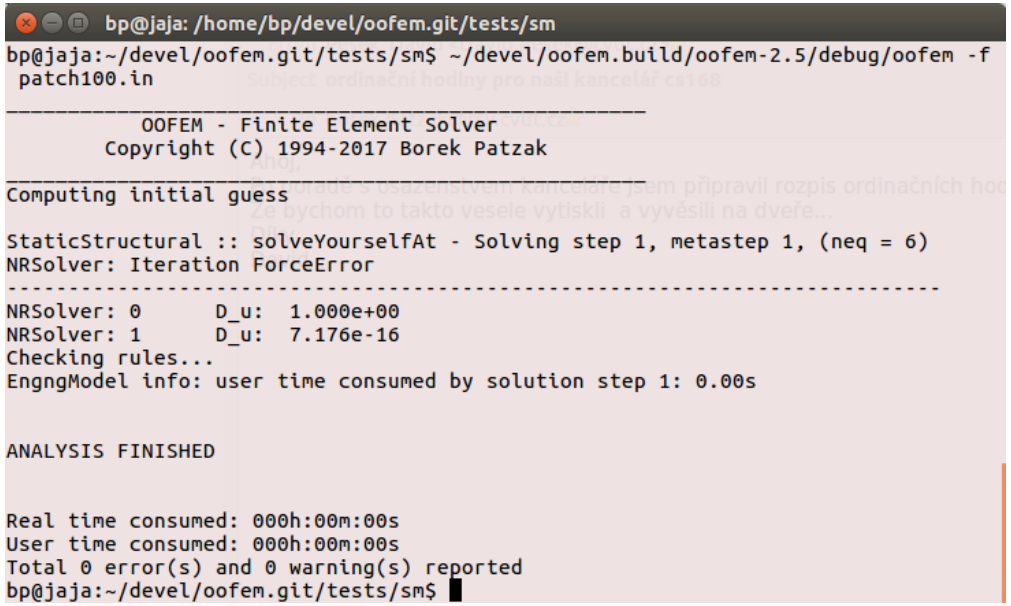
Example of input file

```
beam2d_1.out
Simple Beam Structure - linear analysis
#only momentum influence to the displacements is taken into account
#beamShearCoeff is artificially enlarged.
StaticStructural nsteps 3 nmodules 0
domain 2dBeam
OutputManager tstep_all dofman_all element_all
ndofman 6 nelem 5 ncrosssect 1 nmat 1 nbc 6 nic 0 nltf 3 nset 7
node 1 coords 3 0. 0. 0.
node 2 coords 3 2.4 0. 0.
node 3 coords 3 3.8 0. 0.
node 4 coords 3 5.8 0. 1.5
node 5 coords 3 7.8 0. 3.0
node 6 coords 3 2.4 0. 3.0
Beam2d 1 nodes 2 1 2
Beam2d 2 nodes 2 2 3 DofsToCondense 1 6
Beam2d 3 nodes 2 3 4 DofsToCondense 1 3
Beam2d 4 nodes 2 4 5
Beam2d 5 nodes 2 6 2 DofsToCondense 1 6
SimpleCS 1 area 1.e8 Iy 0.0039366 beamShearCoeff 1.e18 thick 0.54 material 1 set 1
IsoLE 1 d 1. E 30.e6 n 0.2 tAlpha 1.2e-5
BoundaryCondition 1 loadTimeFunction 1 dofs 1 3 values 1 0.0 set 4
BoundaryCondition 2 loadTimeFunction 1 dofs 1 5 values 1 0.0 set 5
BoundaryCondition 3 loadTimeFunction 2 dofs 3 1 3 5 values 3 0.0 0.0 -0.006e-3 set 6
ConstantEdgeLoad 4 loadTimeFunction 1 Components 3 0.0 10.0 0.0 loadType 3 set 3
NodalLoad 5 loadTimeFunction 1 dofs 3 1 3 5 Components 3 -18.0 24.0 0.0 set 2
StructTemperatureLoad 6 loadTimeFunction 3 Components 2 30.0 -20.0 set 7
PeakFunction 1 t 1.0 f(t) 1.
PeakFunction 2 t 2.0 f(t) 1.
PeakFunction 3 t 3.0 f(t) 1.
Set 1 elementranges {(1 5)}
Set 2 nodes 1 4
Set 3 elementedges 2 1 1
Set 4 nodes 2 1 5
Set 5 nodes 1 3
Set 6 nodes 1 6
Set 7 elements 2 1 2
```



Running the oofem

- OOFEM is console application, has to be run from command line



Useful solver command line options

Option	Description
-v	Prints oofem version
-f path	Path to input file
-r step	Restarts the analysis from saved step context
-rn	Forces profile optimization
-l loglevel	Sets the log level (Errors=0, Warnings=1, Relevant=2, Info=3, Debug=4)
-qo path	Redirects standart output to given file
-qe path	Redirects standard error to given file
-c	Forces saving context for every solution step

Understanding output file

```
############################################################# www.oofem.org ###
      ####    ####   ######  ######  #    #
     #    #  #    #  #       #       ##  ##
    #    #  #    #  #####   #####   # ## #
   #    #  #    #  #       #       #    #
  #    #  #    #  #       #       #    #   OOFEM ver. 2.5
  ####    ####   #       ######  #    #   Copyright (C) 1994-2017 Borek Patzak
#############################################################################


Starting analysis on: Fri Aug 24 10:58:09 2018

Homework www sm40 no. 1
```

Initial section with solver version, job name,

starting date and time

```
==============================================================
Output for time 1.00000000e+00
==============================================================
```

Header for each solution (time) step and domain

Understanding output file

```
Output for domain   1
```

```
DofManager output:
------------------
Node           1 (       1):
  dof 1   d -1.37172495e-03
  dof 3   d  0.00000000e+00
  dof 5   d -2.38787802e-05
Node           2 (       2):
  dof 1   d -1.37172495e-03
  dof 3   d  2.03123137e-14
  dof 5   d -1.01549259e-06
```

```
Element output:
---------------
beam element 1 (       1) :
  local displacements  -1.3717e-03 0.0000e+00 -2.3879e-05 -1.3717e-03 2.0312
  local end forces     0.0000e+00 -8.9375e+00 0.0000e+00 0.0000e+00 -1.5062e
  GP  1.1  :  strains  0.0000e+00 0.0000e+00 0.0000e+00 3.6323e-05 0.0000e+0
              stresses 0.0000e+00 0.0000e+00 0.0000e+00 4.2897e+00 0.0000e+0
  GP  1.2  :  strains  0.0000e+00 0.0000e+00 0.0000e+00 2.0106e-05 0.0000e+0
              stresses 0.0000e+00 0.0000e+00 0.0000e+00 2.3745e+00 0.0000e+0
  GP  1.3  :  strains  0.0000e+00 0.0000e+00 0.0000e+00 -1.0531e-06 0.0000e+
              stresses 0.0000e+00 0.0000e+00 0.0000e+00 -1.2437e-01 0.0000e+
  GP  1.4  :  strains  0.0000e+00 0.0000e+00 0.0000e+00 -1.7270e-05 0.0000e+
              stresses 0.0000e+00 0.0000e+00 0.0000e+00 -2.0396e+00 0.0000e+
```

Header for each domain

Dof Manager (node) output

For each node, value of primary unknown for each DOF is reported

Element output

Format depends on particular element, but generally internal variables at each integration point are reported

Understanding output file

```
      R E A C T I O N S  O U T P U T:
      ------------------------------


      Node         1 iDof  3 reaction -8.9375e+00    [bc-id: 1]
      Node         3 iDof  5 reaction  0.0000e+00    [bc-id: 2]
      Node         5 iDof  3 reaction -1.8750e+01    [bc-id: 1]
      Node         6 iDof  1 reaction  1.8000e+01    [bc-id: 3]
      Node         6 iDof  3 reaction -2.0312e+01    [bc-id: 3]
      Node         6 iDof  5 reaction -5.3999e+01    [bc-id: 3]

User time consumed by solution step 1: 0.004 [s]
```

For structural analyses the reaction table is reported.

Finally solution time for each step reported

```
Finishing analysis on: Fri Aug 24 10:58:09 2018

Real time consumed: 000h:00m:00s
User time consumed: 000h:00m:00s
```

At the end the total solution time for all steps is reported

Outline

- Preparation of input files
- **Understanding of oofem modules**
- **Available analyses**
- Element and material libraries

OOFEM Modules

- OOFEMlib – OOFEM core module
- SM – Structural Module
- TM – Transport Module
- FM – Fluid mechanics Module
- Interface Modules to external software

Structural Module (SM)

Available analyses

- Linear static [*linearstatic*]
  - Multiple load cases
  - Adaptive (Zienkiewicz-Zhu aposteriory)
  - Parallel
- Incremental linear static [*IncrLinearStatic*]
  - Sequence of linear analyses
  - Changes in static system
  - Addition/removal of elements

Structural Module (SM)

Available analyses / Cont.

- StaticStructural (Nonlinear static) [*StaticStructural, NonLinearStatic*]
  - Direct & indirect control, Meta-steps
  - Adaptive
  - Parallel
- Linear stability [*LinearStability*]
- Eigen value dynamic [*EigenValueDynamic*]
  - Subspace iteration, Inverse iteration
  - Parallel (SLEPc)
- Linear dynamic (implicit & explicit) [*DEIDynamic*, *DIIDynamic*]
- Nonlinear dynamic (explicit, implicit, explicit-parallel) [*NlDEIDynamic*, *NonLinearDynamic*]

Structural Module (SM)

Element library

- Trusses (1D, 2D,3D) [*truss1d, truss2d, truss3d*]
- Timoshenko beams (2D, 3D) [*beam2d, beam3d*]
- 2D elements
  - plane stress, plane strain (Linear & quadratic triangles, quads)
    [*planestress2d, qplanestress2d, trplanestress2d, qtrplstr, trplanestrrot, trplanestressrotallman, quad1planestrain, trplanestrain*]
  - Plates & Shells (triangular Mindlin-based elements, DKT, MITC)
    [*dktplate, qdktplate, cctplate, tr_shell01, tr_shell02, quad1mindlin, mitc4shell]*
  - Axisymmetric elements (linear triangle& quad)
    [*Axisymm3d, Q4axisymm, L4axisymm*]

Structural Module (SM)

Element library / cont.

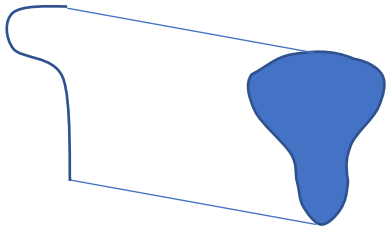
- 3D solid elements [*lspace, lspacebb, qspace, ltrspace, qtrspace, lwedge, qwedge*]
  - Linear & quadratic tetrahedrons & bricks
- Special elements
  - Interface elements in 1D and 2D
  - Spring element, lumped mass element
  - Subsoil elements [*quad1plateSubsoil, tria1platesubsoil*]
  - **IGA** elements
    - BSpline & Nurbs plane stress and 3D elements

Structural Module (SM)

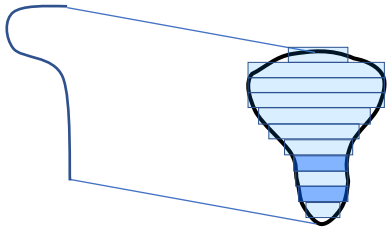
Cross sections

Defines the (kinematic) assumptions for the cross section of the beam/plate/shell

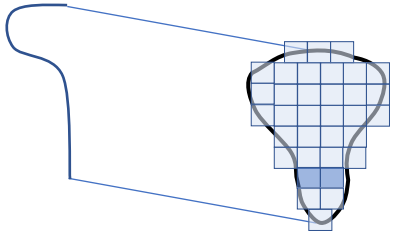
- Integral model [*simplecs*] (single material, cross section characterized by integral quantities: A, Iy, Iz, Ik, …)
- Layered model [*layeredcs*] : Cross section consisting of several layers (of different material) with given width and thickness
- Fibered model [*fiberedcs*] : Cross section consisting of fibers (of different material) with given dimensions



$$N = EA\epsilon_s; M = EI_y\chi$$



$$\epsilon_x^i = \epsilon_s + \chi z$$
$$\sigma_x^i = E^i\varepsilon_x^i$$
$$N = \Sigma\sigma_x^i A^i; M = \Sigma\sigma_x^i A^i z^i$$

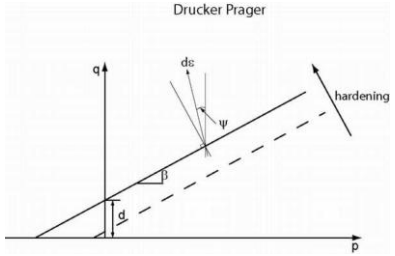


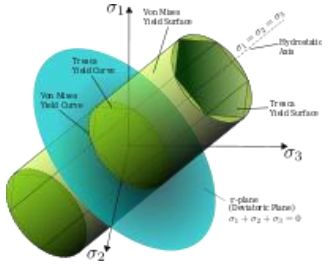
$$\epsilon_x^i = \epsilon_s + \chi z$$
$$\sigma_x^i = E^i\varepsilon_x^i$$
$$N = \Sigma\sigma_x^i A^i; M = \Sigma\sigma_x^i A^i z^i$$

Structural Module (SM)

Material library

- Elastic materials
  - Isotropic, orthotropic, hyper elastic (large strains)
    [*isole, orthole, anisole, hypermat*]
- Plasticity based materials
  - Drucker-Prager plasticity [*DruckerPrager*]
  - Von Mises plasticity (also large strains)
    [*misesmat, MisesMatNl, MisesMatGrad*]
  - Composite plasticity model for masonry





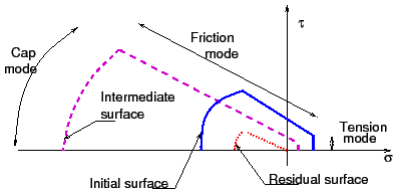
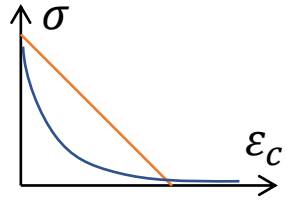


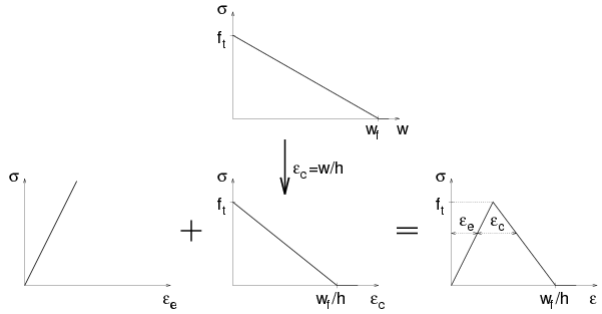
Image source: Wikipedia, http://en.wikipedia.org/wiki/Von_Mises_yield_criterion

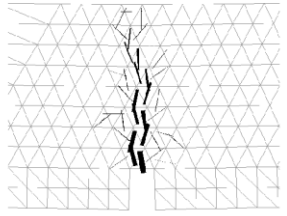
Structural Module (SM)

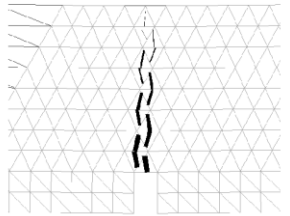
Material library / Cont.

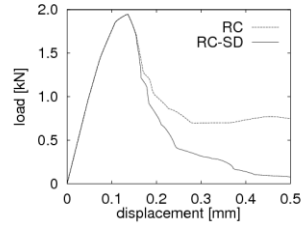
- Material models for tensile failure
  - Rotating crack model
    - Linear, exponential softening [*concrete3*]
    - RCSD – with transition to scalar damage model to prevent stress locking [*rcsd, rcsde*]
  - Fixed crack model [*ConcreteFCM*]
  - Regularization
    - Crack band approach
    - Nonlocal variant [*rcsdnl*]









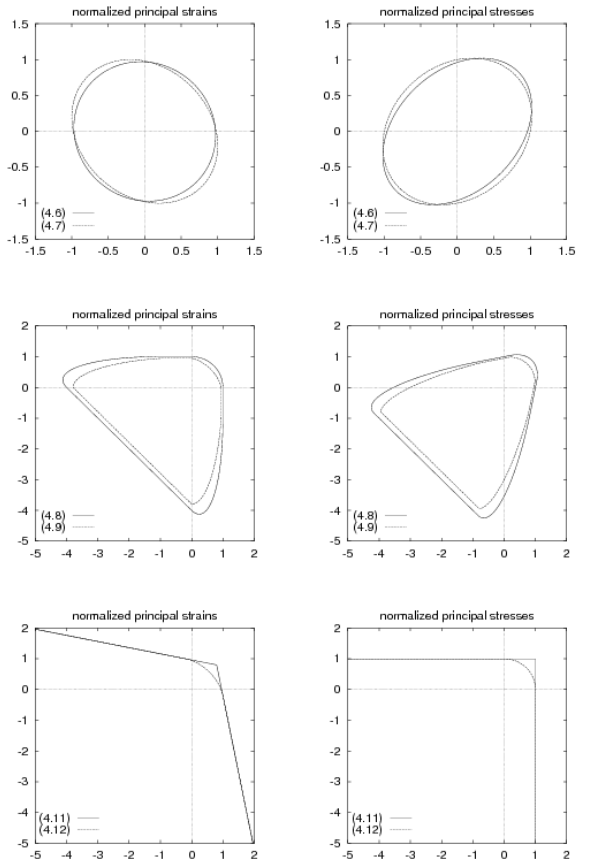


Structural Module (SM)

Material library / Cont.

- Isotropic damage model (local crack band & nonlocal) [*idm1, idm1nl*]
  - Equivalent strain definitions:
    - Mazars
    - Rankine
    - Energy-based
    - Modified Mises (Vree et al. 1995)
- MDM – anisotropic damage model (local & nonlocal) [*mdm*]

$$\boldsymbol{\sigma} = \boldsymbol{D} : \boldsymbol{\varepsilon} = (1 - \omega)\boldsymbol{D}_e : \boldsymbol{\varepsilon}$$



Loading surfaces for various definitions of equivalent strain

Structural Module (SM)

Material library / Cont.

- Material models for concrete
  - Mazars model (local & nonlocal) [*mazarsmodel, mazarsmodelnl*]
  - Creep models (CEB-FIP, B3) [*CebFip78,doublepowerlaw, EC2CreepMat,b3mat, b3solidmat, mps, MPSDamMat*]
  - Microplane model M4 [microplane_m4]
  - DPM – damage-plastic model [*concreteDPM*]

All Modules: Boundary conditions

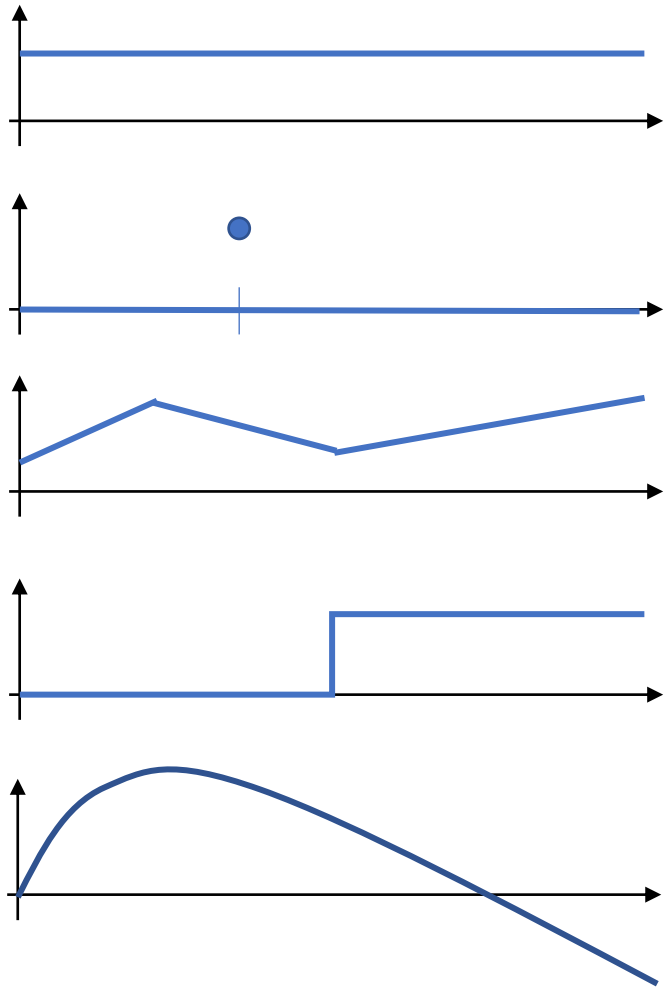
1. Dirichlet BC (prescribed value of unknown on the boundary)
   - *BoundaryCondition #(in) loadTimeFunction #(in) [dofs #(ia)] values #(ia) [set #(in)]*
   - *LinearConstraintBC* ($\Sigma w_i d_i = c$)

2. *Neumann BC (prescribed derivatives, fluxes on the boundary)*
   - *Nodal fluxes, nodal loads: Nodalload*
   - Volume flux, dead weight*: DeadWeight*
   - Edge fluxes, edge loads*: ConstantEdgeLoad, LinearEdgeLoad*
   - Surface fluxes, surface loads*: ConstantSurfaceLoad*
3. Structural temperature loading/eigen strain loading*: StructTemperatureLoad, StructEigenstrainLoad*
4. Mixed BC*: MixedGradientPressure, MixedGradientPressureWeaklyPeriodic, MixedGradientPressureNeumann, MixedGradientPressureDirichlet*

All Modules: (time) functions

Describe (time) variation

- Constant function [*ConstantFunction*]
- Peak function [*PeakFunction*]
- Piece wise linear functio [*PiecewiseLinFunction*]
- Heaviside function [*HeavisideLTF*]
- User defined LTF [*UsrDefLTF*]



Transport Module (TM)

- Analyses
  - Stationary transport problem [*StationaryProblem*]
  - Transient transport problem (linear& nonlinear) [*NonStationaryProblem*]
- Heat transfer/mass transport elements
  - 2D linear triangles and quads [*tr1ht, tr1mt,, quad1ht, quad1mt*]
  - Axisymmetric triangles and quads [*traxisum1ht, qadaxisym1th*]
  - 3D linear brick [*tetra1ht, brict1ht,qbrick1ht*]
- Elements for coupled heat & mass transfer
  - 2D linear quad [*tr1hmt, quad1hmt]*
  - 3D linear brick [*brick1hmt, qbrick1hmt]*

CFD Module (FM)

- Analyses
  - Stokes problem
  - Transient incompressible flow [*CBS, SUPG*]
    - Free surface, immiscible two fluid flow (VOF, LS)
- Fluid module elements
  - Stokes problem elements [*Tr121Stokes, Tet21Stokes, Hexa21Stokes, Tr1BubbleStokes, Tet1BubbleStokes*]
  - 2D incompressible CBS linear triangle [*Tr1CBS*]
  - 2D incompressible SUPG elements [*Tr1SUPG, Tr21SUPG, Tr1SUPGAxi, TET1SUPG*]
    - Tr11, Tr21 elements
    - Py1 element in 3D

Coupled problems

Staggered Problem [*StaggeredProblem*]

- Represent a sequence of subproblems, where the results of i-th problem can depend on results from previous problems.
- Examples: thermo-mechanical, thermo-hydro-mechanical analysis
- Syntax: *StaggeredProblem nsteps #(in) deltaT #(rn) prob1 #(s) prob2 #(s)*