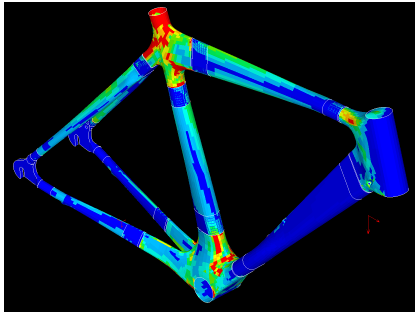
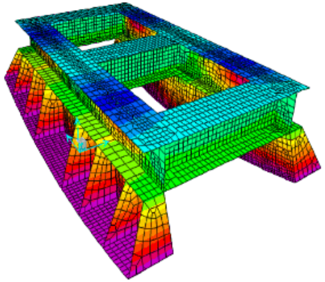
EPFL



(Credit: Form Lab)



(Credit: Adesol)

Introduction

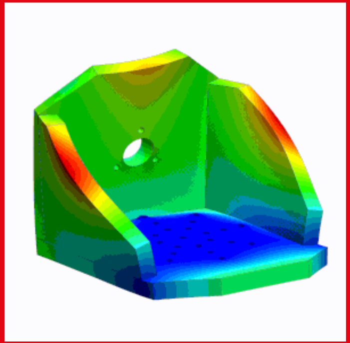
Informations générales

ME-473

Dynamic finite element analysis of structures

Stefano Burzio

Welcome!



(Credit: Sentek)

- Who are the instructors?
- How is the learning process structured?
- What reading materials are recommended?
- How will my learning be assessed?
- What does the curriculum include?
- Why is dynamic FEA important?

ME-473 Teaching staff

Lecturer

Stefano Burzio

Teaching assistant

Timothée Daniel Salamon
Ph.D. student at LFMI

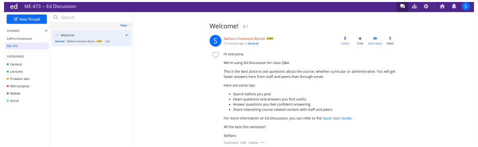
- Who am I?

Logistic

Lecture	Tuesday 15h - 17h	CM 1 104
Exercise	Tuesday 17h - 18h	CM 1 104
Office hours	Friday 11h - 12h	ME A2 390

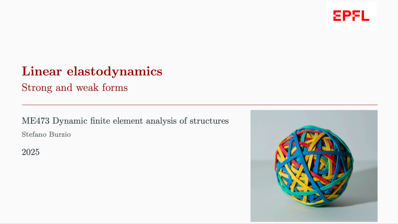
Ed discussion forum: (Link)

A general student forum, where both lecturers and fellow students can answer your questions.

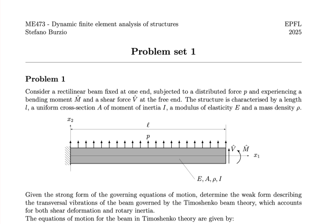


Moodle page: (Link)

An official platform for announcements and sharing materials, including lecture notes, exercises, practice problems, and more.



Slides
(posted Tuesday morning)



ME473 - Dynamic finite element analysis of structures EPFL
Stefano Burzio 2025

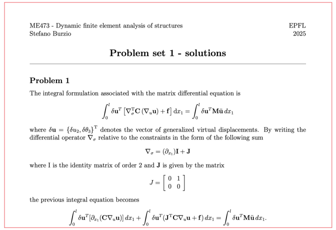
Problem set 1

Problem 1

Consider a rectilinear beam fixed at one end, subjected to a distributed force p and experiencing a bending moment $\bar{M}$ and a shear force $\bar{V}$ at the free end. The structure is characterised by a length l, a uniform cross-section A of moment of inertia I, a modulus of elasticity E and a mass density ρ.

Given the strong form of the governing equations of motion, determine the weak form describing the transversal vibrations of the beam governed by the Timoshenko beam theory, which accounts for both shear deformation and rotary inertia.
The equations of motion for the beam in Timoshenko theory are given by:

Problem sets
(posted Tuesday morning)



ME473 - Dynamic finite element analysis of structures EPFL
Stefano Burzio 2025

Problem set 1 - solutions

Problem 1

The integral formulation associated with the matrix differential equation is

$$\int_0^l \delta \mathbf{u}^T \left[\nabla_x^T \mathbf{C}(\nabla_x \mathbf{u}) + \mathbf{f}\right] dx_1 = \int_0^l \delta \mathbf{u}^T \mathbf{M}\ddot{\mathbf{u}}\, dx_1$$

where $\delta\mathbf{u} = \{\delta u_2, \delta\theta_3\}^T$ denotes the vector of generalized virtual displacements. By writing the differential operator ∇_x relative to the constraints in the form of the following sum

$$\nabla_x = (\partial_{x_1})\mathbf{I} + \mathbf{J}$$

where $\mathbf{I}$ is the identity matrix of order 2 and $\mathbf{J}$ is given by the matrix

$$J = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

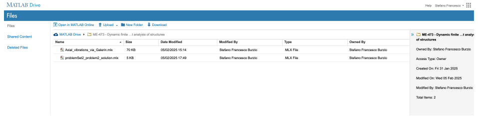
the previous integral equation becomes

$$\int_0^l \delta \mathbf{u}^T [\partial_{x_1}(\mathbf{C}\nabla_x \mathbf{u})]\, dx_1 + \int_0^l \delta \mathbf{u}^T (\mathbf{J}^T \mathbf{C}\nabla_x \mathbf{u} + \mathbf{f})\, dx_1 = \int_0^l \delta \mathbf{u}^T \mathbf{M}\ddot{\mathbf{u}}\, dx_1.$$

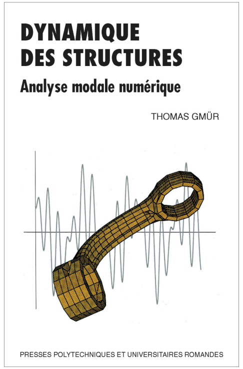
Problem solutions
(posted Tuesday evening)

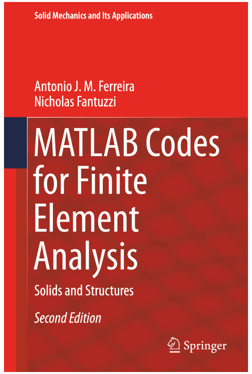
Matlab drive (Link)

Read-only online MATLAB code repository.
Log in with your Switch-edu account



Recommended books

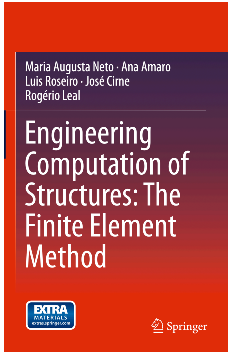


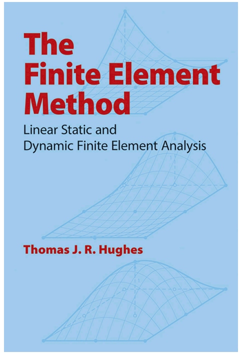


Main references:

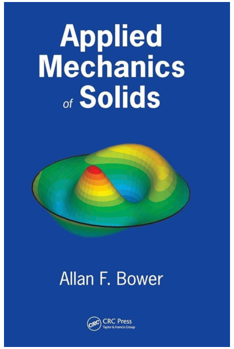
- Thomas Gmür
  Dynamique des structures (PPUR)
  EPFL library: (07 534 GMU 2012)
- Antonio Ferreira and Nicholas Fantuzzi
  MATLAB Codes for Finite Element Analysis
  Available online (link)

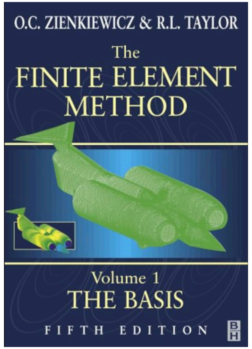
English

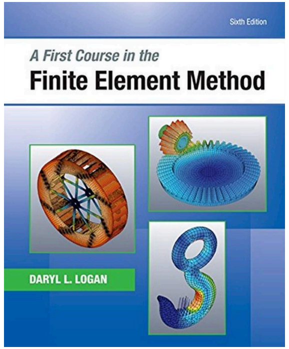




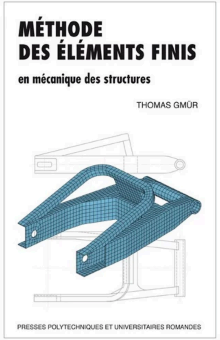
All-time classics







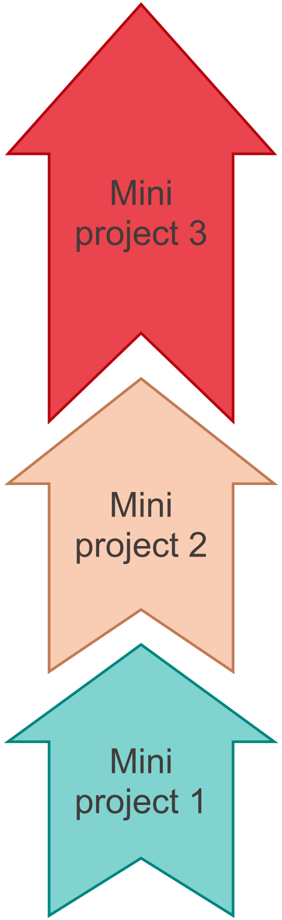
ME-372



Syllabus

Week	Module	Lecture topic	Problem set	Mini projects
1	Linear elastodynamics	Strong and weak forms	1	Groups formation
2		Galerkin method	2	
3		Finite element method	3	Project 1 statement
4		Solid elements	4	
5		Systematisation of the procedure		**Project 1 submission**
6	Special structural elements	Trusses	5	Project 2 statement
7		Frames	6	
8		Kirchoff plates		**Project 2 submission**
9		Riessen-Mindlin plates	7	Project 3 statement
10	Rotating and dissipative structures	Strong and weak forms	8	
11		Discrete form	9	
12	Algorithms for large eigenvalue problems	Lancoz, subspace algorithms	10	
13		Free, dissipative and rotating systems		**Project 3 submission**
14		Wilson and Newmark schemes		Project 3 presentations

Mini projects submission guidelines



- **Objective**: put course content into practice
- **Groups**: 3 to 5 students (register groups on Moodle)
- **Group assessment**:
  - work divided equally among the group
  - evaluation criteria on Moodle
- **Deliverables**:
  - PDF report (explaining your analysis and detailing the code)
  - Simulation file (MATLAB, ANSYS, Abaqus)
- **Deadlines**:

Project	Statement	Submission
1	Tuesday March 4	**Friday March 21**
2	Tuesday March 25	**Friday April 11**
3	Tuesday April 15	**Friday May 23**

- Presentations of mini-project 3 on Tuesday May 27

Main learning objectives

1. **Understand the fundamentals of dynamic FEA**

   Develop a strong theoretical foundation in the principles of dynamic analysis, including *modal analysis* and *frequency response*.

2. **Analyse the dynamic behaviour of structural components**

   Study the dynamic response of specific structures such as *rods*, *trusses*, *beams*, *plates*, and *shells* under various loading and boundary conditions.

3. **Develop proficiency in finite element formulation for dynamic problems**

   Learn to derive and implement finite element equations for time-dependent structural dynamics problems, including damping and rotational effects.

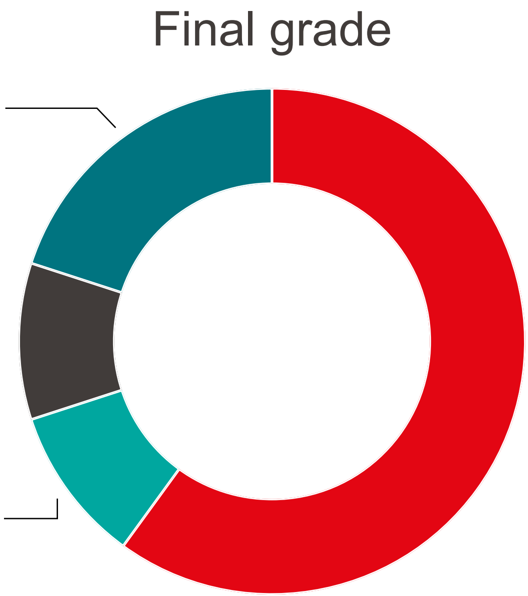
4. **Apply computational methods for solving dynamic systems**

   Utilise numerical techniques such as direct integration, *modal superposition*, and *time-stepping methods* to solve complex dynamic problems.

5. **Implement advanced applications in engineering design**

   Apply dynamic FEA to real-world engineering problems, such as vibration analysis, impact simulation, and fatigue prediction, using commercial FEA software.

Assessment method

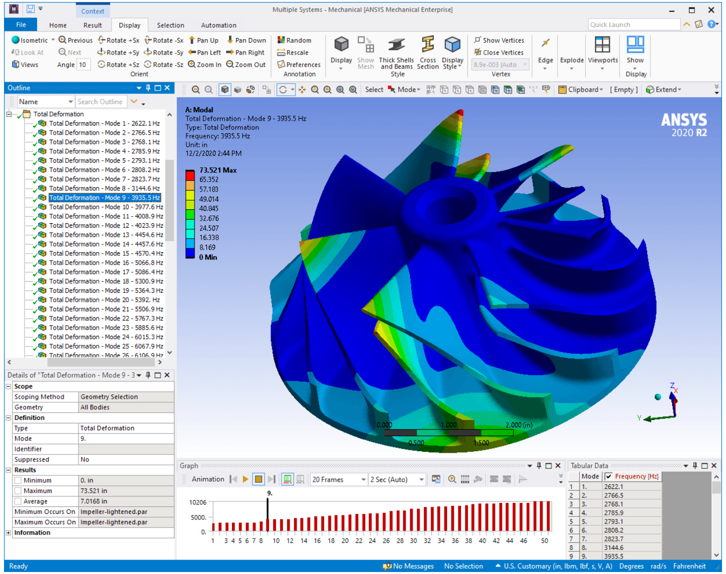


The finite element analysis dichotomy

Makers	Users

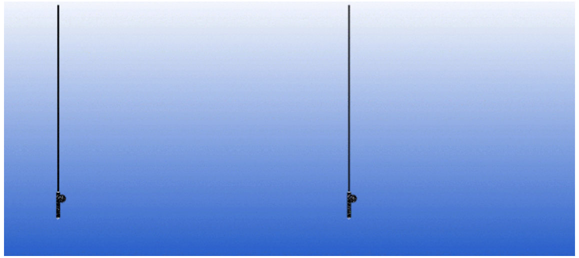
```
static public void verify()
{
    cgiSolverBlazeClass solver = new cgiSolverBlazeClass();
    solver.createStructure();

    StringBuilder sb = new StringBuilder();
    solver.getDefaultTestPath(ref sb);
    string sInputFileName = sb.ToString() + "\\tests\\newModels\\Verify-Example4-Sparse.r3a";

    // the following should be called after createStructure()
    cgiSolverBlazeClass.ListMessageDelegate ListMsg = new cgiSolverBlazeClass.ListMessageDelegate(Callback);
    cgiSolverBlazeClass.StatusMessageDelegate StatusMsg = new cgiSolverBlazeClass.StatusMessageDelegate(StatusCallback);
    cgiSolverBlazeClass.SparseSolverProgressDelegate SparseMsg = new cgiSolverBlazeClass.SparseSolverProgressDelegate(MlkProgress);
    solver.setListMessageFunction(ListMsg);
    solver.setStatusMessageFunction(StatusMsg);
    solver.setSparseSolverStatusFunction(SparseMsg);

    solver.setModelType((int)cgiModelEnum.kModel_Frame2D);

    // LENGTH=ft;   DIMENSION=in; FORCE=kip; FORCE_LINE=kip/ft; MOMENT=kip-ft; FORCE_SURFACE=lb/ft^2;
    // DISPLACEMENT_TRANS=in; DISPLACEMENT_ROTATE=rad; MODULUS=kip/in^2; WEIGHT_DENSITY=lb/ft^3; STRESS=lb/in^2
    // SPRING_TRANS_1D=lb/in; SPRING_ROTATE_1D=lb-in/rad; SPRING_TRANS_2D=kip/in^2; SPRING_TRANS_3D=kip/in^3
    solver.setStandardEnglishUnits();

    // define materials
    List<cgiMaterialCli> listMat = new List<cgiMaterialCli>();
    cgiMaterialCli mat = new cgiMaterialCli();
    mat.setId(1);
    mat.setProperties("Default222", 29000, 0.3, 450);
    listMat.Add(mat);
    solver.setMaterials(listMat);

    // define sections
    List<cgiSectionCli> listSect = new List<cgiSectionCli>();
    cgiSectionCli sect1 = new cgiSectionCli();
    sect1.setId(1);
    sect1.setProperties("W27X84", 24.8, 12.282, 12.7488, 2850, 106, 2.81);
    listSect.Add(sect1);
    cgiSectionCli sect2 = new cgiSectionCli();
    sect2.setId(2);
    sect2.setProperties("W27X84", 24.8, 12.282, 12.7488, 2850, 106, 2.81);
    listSect.Add(sect2);
```

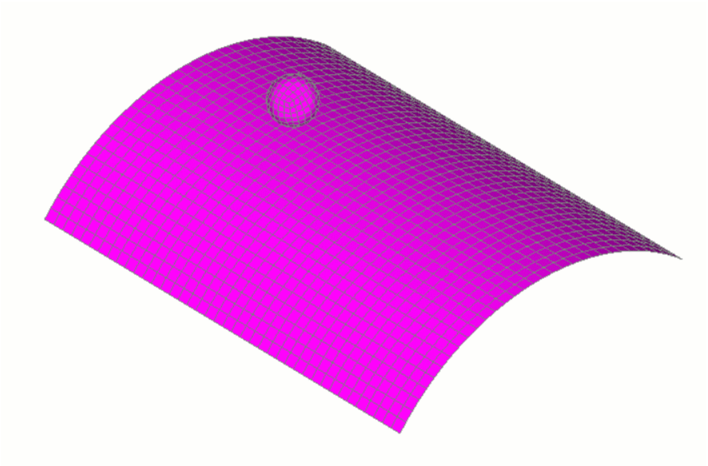
(Credit: Cg-inc)



(Credit: Padt)

Dynamic or not dynamic





Static analysis in a nutshell:

- No matter how you apply the load, the solver assumes it happens infinitely slowly, meaning the loading method has no impact on structural behaviour.
- The load is not changing in time.
- A constant load doesn't mean a linear response—buckling, yielding, and other nonlinear effects can occur, but the analysis remains static.