# TMA4220 - PDEs /w FEM

## Programming Project

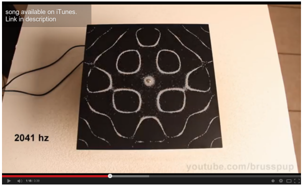
G. A. Hasle  T. Baerland  E. Ingebrigtsen

Finite Element Methods on Free Vibrations

## The Free Vibration Equation

$$\rho \frac{\partial^2 u}{\partial t^2} = \nabla \sigma(u) \tag{1}$$

- Is it possible to model




with Finite Element Methods?

# Introduction

- We want a variational formulation of (1)
- Usual procedure: Multiply (1) with a test function $v$ and integrate over the domain to obtain

$$\rho \int_{\Omega} \ddot{u} \cdot v \mathrm{d}\Omega = - \int_{\Omega} \epsilon(v) \cdot C\epsilon(u) \, \mathrm{d}\Omega \tag{2}$$

# Introduction

- Semi-discretization on (2) yields

$$M\ddot{\boldsymbol{u}} = -A\boldsymbol{u} \tag{3}$$

$$A = [A_{ij}] = \int_{\Omega} \varepsilon\left(\varphi_i\right)^T C\varepsilon\left(\varphi_j\right) \mathrm{d}\Omega$$

$$M = [M_{ij}] = \int_{\Omega} \rho\varphi_i^T \varphi_j \mathrm{d}\Omega$$

- Assuming $\boldsymbol{u} = \boldsymbol{u}e^{i\omega t}$ on (3) yields the generalized eigenvalue problem

$$\omega^2 M\boldsymbol{u} = A\boldsymbol{u} \tag{4}$$

# Building $A$ and $M$

$$\varphi_{\hat{i},1} = \begin{bmatrix} \varphi_{\hat{i}} \\ 0 \\ 0 \end{bmatrix}, \quad \varphi_{\hat{i},2} = \begin{bmatrix} 0 \\ \varphi_{\hat{i}} \\ 0 \end{bmatrix}, \quad \varphi_{\hat{i},3} = \begin{bmatrix} 0 \\ 0 \\ \varphi_{\hat{i}} \end{bmatrix}.$$

$$\varphi_{\hat{i}}\Big|_{T_k} = c_0 + c_1 x + c_2 y + c_3 z$$

$$\varepsilon\left(\varphi_{\hat{i},d}\right) = \begin{bmatrix} c_1 \\ 0 \\ 0 \\ c_2 \\ c_3 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ c_2 \\ 0 \\ c_1 \\ 0 \\ c_3 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ c_3 \\ 0 \\ c_1 \\ c_2 \end{bmatrix}$$

# Building $A$ and $M$

- Stiffness matrix $A$:

$$\varepsilon\left(\varphi_i\right)^T C\varepsilon\left(\varphi_j\right)\mathrm{area}(T_k)$$

- Mass matrix $M$:

$$\rho \cdot \det(B) M^P_{\alpha,\beta}$$

# Implementation

- Solve $\omega^2 M\boldsymbol{u} = A\boldsymbol{u}$ for the $k$ lowest eigenvalues using MATLAB's `eigs`
- Use the different eivenvectors $\boldsymbol{u_i}$ to make animations
- $\boldsymbol{x} = \boldsymbol{x_0} + \alpha \boldsymbol{u_i} \sin(t)$
- Collection of MATLAB scripts/functions
- Open to different geometries (chess pawn, gun, plate) and materials
- ParaView for the post processing

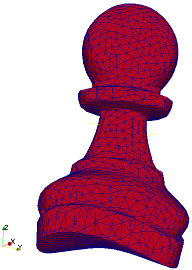
# Animations

- Animation of a gun(!). Mode 7,9,10 and 20
- Animation of a chess pawn. Mode 7,10,14 and 15

# Boundary condition

Can we have boundary conditions?

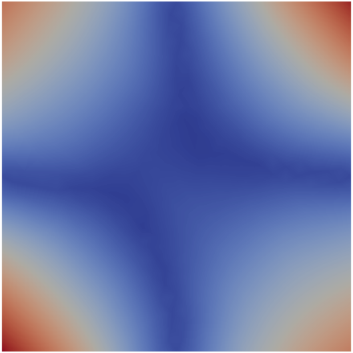
- Physical interpretation
- Free vibration
- Can use Dirichlet
- Animation of a chess pawn. Homogeneous boundary conditions




¿¿

# Back to the plate

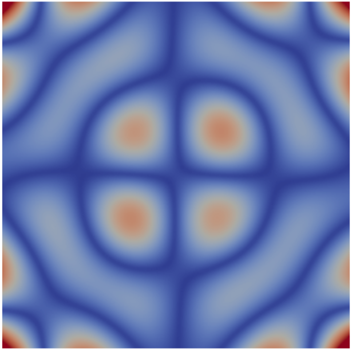
- An example of our results (3443 Hz):



- Needs finer mesh

# Back to the plate (cont.)

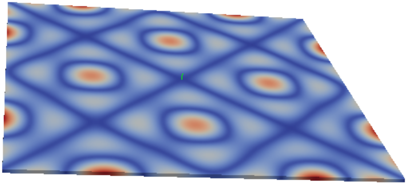
- Finer mesh gives (3031.6 Hz):



- Picks up frequencies closer together.

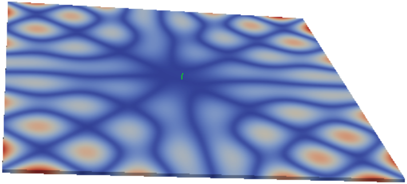
# Even more plates

- 7207Hz



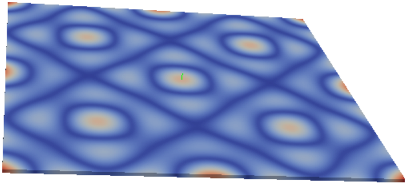
# Even more plates

- 7222Hz



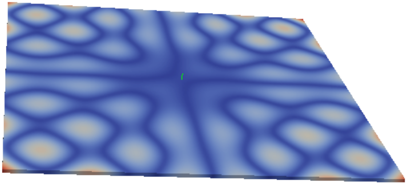
# Even more plates

- 7261Hz



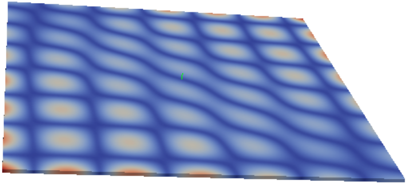
# Even more plates

- 7282Hz



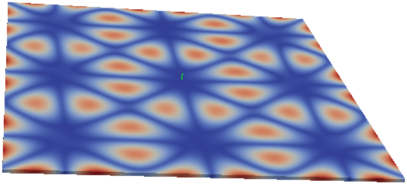
# Even more plates

- 7797Hz



# Even more plates

- 8049HZ



# Even more plates

- 8121Hz