Finite Elements and Topology Optimization

GAMES 201 Lecture 6

Yuanming Hu

MIT CSAIL

July 13, 2020

Table of Contents

1. FEM Overview
2. Discretizing Poisson's equation
3. Discretizing linear elasticity
4. Topology optimization

Finite element method

Finite element method (FEM) belongs to the family of Galerkin methods. In FEM, continuous PDEs are converted to discrete (linear) systems.

Typical steps:

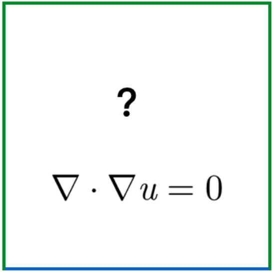
1. Convert strong-form PDEs to **weak forms**, using a test function w.
2. Integrate by parts to redistribute gradient operators.
3. Use the divergence theorem to simplify equations and enforce Neumann boundary conditions (BCs).
4. Discretization (build the stiffness matrix and right-hand side).
5. Solve the (discrete) linear system.

Understanding FEM is important for many other discretization methods, including the Material Point Method (MPM) later in this course.

2D Poisson's equation

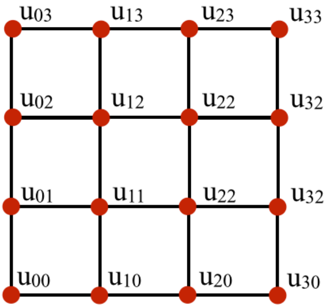
$$u(\mathbf{x}) = f(\mathbf{x}), \mathbf{x} \in \partial\Omega$$

Dirichlet boundary



Neumann boundary

$$\nabla u(\mathbf{x}) \cdot \mathbf{n} = g(\mathbf{x}), \mathbf{x} \in \partial\Omega$$



Discretized degrees of freedoms

Application: pressure projection in fluid simulations.

Table of Contents

1. FEM Overview
2. Discretizing Poisson's equation
3. Discretizing linear elasticity
4. Topology optimization

Weak formulation

Finite Elements and Topology Optimization

Yuanming Hu

FEM Overview

Discretizing Poisson's equation

Discretizing linear elasticity

Topology optimization

Arbitrary 2D test function $w(\mathbf{x})$:

$$\nabla \cdot \nabla u = 0 \iff \forall w, \iint_\Omega w(\nabla \cdot \nabla u) \mathrm{d}A = 0$$

Intuitively:

1. $\Rightarrow$: trivial
2. $\Leftarrow$: if $\nabla \cdot \nabla u(\mathbf{x}) \neq 0$, we can always construct a test function $w(\mathbf{x})$ s.t. ...

Getting rid of second-order derivatives (I)

We want to get rid of $\nabla \cdot \nabla$ in

$$\nabla \cdot \nabla u = 0.$$

Recall integration by parts, or derivative of products:

$$\nabla w \cdot \nabla u + w \nabla \cdot \nabla u \quad = \quad \nabla \cdot (w \nabla u) \tag{1}$$

Since $\nabla \cdot \nabla u = 0$, we have

$$\nabla w \cdot \nabla u \quad = \quad \nabla \cdot (w \nabla u) \tag{2}$$

To summarize,

$$\nabla \cdot \nabla u = 0 \iff \forall w, \iint_\Omega \nabla w \cdot \nabla u \mathrm{d}A = \iint_\Omega \nabla \cdot (w \nabla u) \mathrm{d}A. \tag{3}$$

Getting rid of second-order derivatives (II)

$$\iint_\Omega \nabla w \cdot \nabla u \mathrm{d}A = \iint_\Omega \nabla \cdot (w \nabla u) \mathrm{d}A \tag{4}$$

Divergence theorem applied to the RHS:

$$\iint_\Omega \nabla w \cdot \nabla u \mathrm{d}A = \oint_{\partial\Omega} w \nabla u \cdot \mathrm{d}\mathbf{n} \tag{5}$$

Discretization (I) Basis functions

$$u(\mathbf{x}) = \sum_j u_j \phi_j(\mathbf{x}) \quad (6)$$

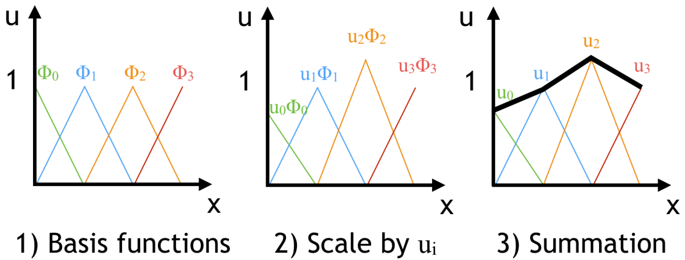


Figure: Compusing 1D piece-wise linear functions using 1D Basis functions.

In this course we focus on linear basis functions, which means the fields are exactly *linear/bilinear/trilinear* interpolated versions of the degrees of freedoms u_i. (Higher order basis functions are rarely used in graphics.)

Discretization (I) Basis functions

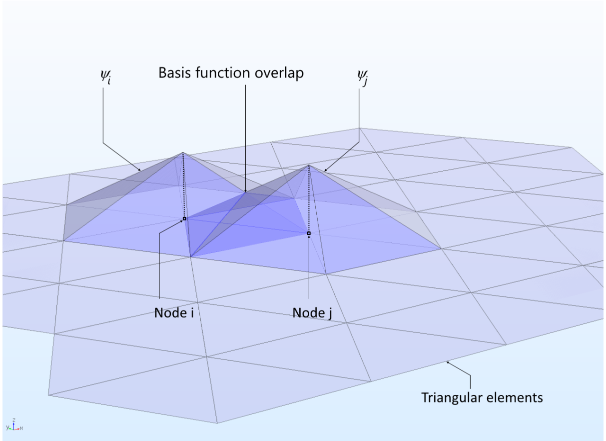


Figure: 2D basis functions on a triangular mesh. Source: https://www.comsol.com/multiphysics/finite-element-method

Discretization (I) Basis functions

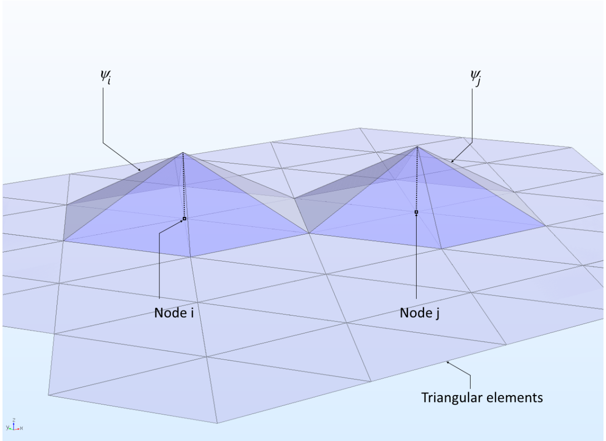


Figure: 2D basis functions on a triangular mesh. Source: https://www.comsol.com/multiphysics/finite-element-method

Discretization (I)

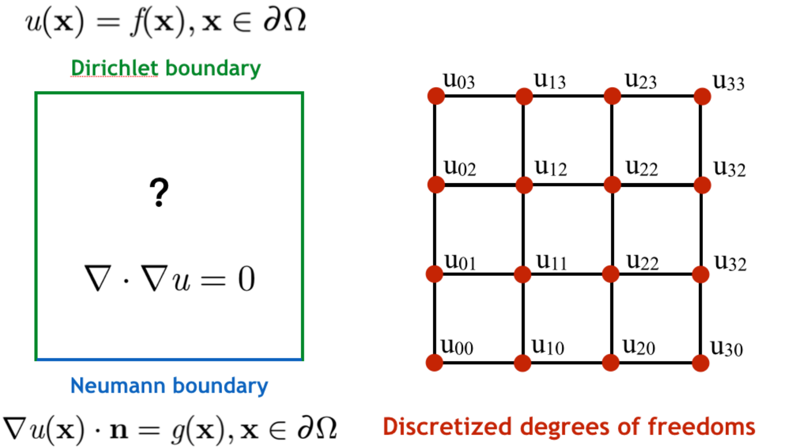


Figure: We use rectangular (quadrilateral) finite elements. Use your imagination to visualize the basis functions :-)

Discretization (II)

Now we represent $u(\mathbf{x})$ as

$$u(\mathbf{x}) = \sum_j u_j \phi_j(\mathbf{x}), \tag{7}$$

Recall that we wan to solve for u s.t.

$$\iint_\Omega \nabla w \cdot \nabla u \mathrm{d}A = \oint_{\partial\Omega} w \nabla u \cdot \mathrm{d}\mathbf{n}, \tag{8}$$

Simple substitution gives

$$\forall w, \iint_\Omega \nabla w \cdot \nabla \left(\sum_j u_j \phi_j \right) \mathrm{d}A = \oint_{\partial\Omega} w \nabla u \cdot \mathrm{d}\mathbf{n}. \tag{9}$$

It's sufficient to use only the basis function ϕ_i as test functions w:

$$\forall i, \iint_\Omega \nabla \phi_i \cdot \nabla \left(\sum_j u_j \phi_j \right) \mathrm{d}A = \oint_{\partial\Omega} \phi_i \nabla u \cdot \mathrm{d}\mathbf{n}. \tag{10}$$

Discretization (III)

$$\forall i, \iint_\Omega \nabla \phi_i \cdot \nabla \left(\sum_j u_j \phi_j \right) \mathrm{d}A = \oint_{\partial\Omega} \phi_i \nabla u \cdot \mathrm{d}\mathbf{n}. \tag{11}$$

Extract $\sum_j u_j$ out of $\iint_\Omega$:

$$\forall i, \sum_j \left(\iint_\Omega \nabla \phi_i \cdot \nabla \phi_j \mathrm{d}A \right) u_j = \oint_{\partial\Omega} \phi_i \nabla u \cdot \mathrm{d}\mathbf{n} \tag{12}$$

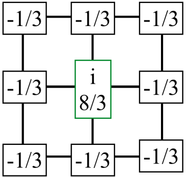
In matrix form...

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{13}$$

$\mathbf{K}$: "stiffness" matrix; $\mathbf{u}$: degree of freedoms/solution vector; $\mathbf{f}$: load vector

Discretization (IV)

Now we need to compute $\mathbf{K}_{ij} = \iint_\Omega \nabla\phi_i \cdot \nabla\phi_j \mathrm{d}A$. Here we are using a simply basis function so it's not hard to compute analytically. (In more difficult cases people use Gaussian quadrature).



9-point Laplacian stencil $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Discretization (IV)

Recall the 5-point Laplacian stencil we obtained using finite difference in previous lectures:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Boundary Conditions

Recall that our linear system is

$$\forall i, \sum_j \left(\iint_\Omega \nabla \phi_i \cdot \nabla \phi_j \mathrm{d}A \right) u_j = \oint_{\partial\Omega} \phi_i \nabla u \cdot \mathrm{d}\mathbf{n} \tag{14}$$

Two types of boundary conditions

1. Dirichlet boundary conditions $u(\mathbf{x}) = f(\mathbf{x}), \mathbf{x} \in \partial\Omega$.
   Easy: directly set corresponding $u_i = f(\mathbf{x}_i)$.
2. Neumann boundary conditions $\nabla u(\mathbf{x}) \cdot \mathbf{n} = g(\mathbf{x}), \mathbf{x} \in \partial\Omega$
   Plug g into the RHS of the equation, which yields non-zero entries in $\mathbf{f}$.

Table of Contents

1. FEM Overview
2. Discretizing Poisson's equation
3. Discretizing linear elasticity
4. Topology optimization

Linear elasticity FEM

Cauchy momentum equation:

$$\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = \frac{1}{\rho}\nabla \cdot \boldsymbol{\sigma} + \mathbf{g}$$

$\mathbf{v}$: velocity
ρ: density
$\boldsymbol{\sigma}$: Cauchy stress tensor (symmetric 2/3D "matrix")
$\mathbf{g}$: body force (e.g., gravity)
Quasistatic state ($\mathbf{v} = 0$), constant density, no gravity:

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}$$

Degree of freedom: **displacement** $\mathbf{u}$. Note that $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{u})$ (more on this later).
Infinitesimal deformation: Lagrangian/Eulerian classification does not make sense.

Index notation

Spacial axis $x, y, z, \dots$ are uniformly represented as $\mathbf{x}_\alpha, \mathbf{x}_\beta, \mathbf{x}_\gamma, \dots$
Comma "," means spatial derivatives. For example, $\boldsymbol{\sigma}_{\alpha\beta,\gamma} = \frac{\partial \boldsymbol{\sigma}_{\alpha\beta}}{\partial \mathbf{x}_\gamma}$.

Vector notation v.s. index notation:

$$\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = \frac{1}{\rho}\nabla \cdot \boldsymbol{\sigma} + \mathbf{g} \iff \frac{\mathrm{D}\mathbf{v}_\alpha}{\mathrm{D}t} = \frac{1}{\rho}\sum_\beta \boldsymbol{\sigma}_{\alpha\beta,\beta} + \mathbf{g}_\alpha.$$

($\boldsymbol{\sigma}_{\alpha\beta,\beta}$ stands for $\frac{\partial \boldsymbol{\sigma}_{\alpha\beta}}{\partial \mathbf{x}_\beta}$.)
(In this lecture we do **not** use implicit summation for clarity.)

Discretize Cauchy momentum equation using FEM

More difficult compared to Poisson's problem: **a)** scalar v.s. vector; **b)** direct v.s. extra linear mapping.

Weak form with test function: $\mathbf{w}(\mathbf{x}) : \mathcal{R}^2 \to \mathcal{R}^2$:

$$\sum_\beta \boldsymbol{\sigma}_{\alpha\beta,\beta} \mathbf{w}_\alpha = 0,$$

Integration by parts:

$$\sum_\beta \boldsymbol{\sigma}_{\alpha\beta,\beta} \mathbf{w}_\alpha + \sum_\beta \boldsymbol{\sigma}_{\alpha\beta} \mathbf{w}_{\alpha,\beta} = \sum_\beta (\boldsymbol{\sigma}_{\alpha\beta} \mathbf{w}_\alpha)_{,\beta} \Rightarrow \sum_\beta \boldsymbol{\sigma}_{\alpha\beta} \mathbf{w}_{\alpha,\beta} = \sum_\beta (\boldsymbol{\sigma}_{\alpha\beta} \mathbf{w}_\alpha)_{,\beta}.$$

Divergence theorem:

$$\forall \alpha \forall \mathbf{w}, \iint_\Omega \sum_\beta \boldsymbol{\sigma}_{\alpha\beta} \mathbf{w}_{\alpha,\beta} \mathrm{d}A = \oint_{\partial\Omega} \sum_\beta (\boldsymbol{\sigma}_{\alpha\beta} \mathbf{w}_\alpha) \mathrm{d}\mathbf{n}_\beta. \tag{15}$$

Discretization

$$\forall\alpha\forall\mathbf{w}, \iint_{\Omega}\sum_{\beta}\boldsymbol{\sigma}_{\alpha\beta}\mathbf{w}_{\alpha,\beta}\mathrm{d}A = \oint_{\partial\Omega}\sum_{\beta}(\boldsymbol{\sigma}_{\alpha\beta}\mathbf{w}_{\alpha})\mathrm{d}\mathbf{n}_{\beta}. \tag{16}$$

Replace $\mathbf{w}$ and $\mathbf{u}$ with their discrete versions:

$$\mathbf{w}_{\alpha}(\mathbf{x}) = \sum_{i}\mathbf{w}_{i\alpha}\phi_{i\alpha}(\mathbf{x}), \quad \mathbf{u}_{\alpha}(\mathbf{x}) = \sum_{j}\mathbf{u}_{j\alpha}\phi_{j\alpha}(\mathbf{x}) \tag{17}$$

$$\forall\alpha\forall i, \iint_{\Omega}\sum_{\beta}\left[\boldsymbol{\sigma}\left(\mathbf{u}(\mathbf{x})\right)\right]_{\alpha\beta}\phi_{i\alpha}(\mathbf{x})\mathrm{d}A = \oint_{\partial\Omega}\sum_{\beta}(\boldsymbol{\sigma}_{\alpha\beta}\phi_{i\alpha})\mathrm{d}\mathbf{n}_{\beta}. \tag{18}$$

Relating $\boldsymbol{\sigma}$ to $\mathbf{u}$

From infinitesimal strain theory:

Strain tensor:

$$\mathbf{e} = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)$$

Cauchy stress tensor:

$$\boldsymbol{\sigma} = \lambda \mathrm{tr}(\mathbf{e})\mathbf{I} + 2\mu \mathbf{e}$$

... or in index notation:

$$\mathbf{e}_{\alpha\beta} = \frac{1}{2}\left(\mathbf{u}_{\alpha,\beta} + \mathbf{u}_{\beta,\alpha}\right)$$

$$\boldsymbol{\sigma}_{\alpha\beta} = \lambda \delta_{\alpha\beta} \sum_{\alpha} \mathbf{e}_{\alpha\alpha} + 2\mu \mathbf{e}_{\alpha\beta}$$

$$\delta_{\alpha\beta} = \begin{cases} 1, & \text{if } \alpha = \beta \\ 0, & \text{if } \alpha \neq \beta \end{cases}$$

In one word: here $\boldsymbol{\sigma}$ is a **linear** function of $\mathbf{u}$.

Building the linear system

$\boldsymbol{\sigma}$ is a linear function of $\mathbf{u}$.

$$\forall\alpha\forall i, \iint_\Omega \sum_\beta \left[\boldsymbol{\sigma}\left(\mathbf{u}(\mathbf{x})\right)\right]_{\alpha\beta} \phi_{i\alpha}(\mathbf{x})\mathrm{d}A = \oint_{\partial\Omega} \sum_\beta (\boldsymbol{\sigma}_{\alpha\beta}\phi_{i\alpha})\mathrm{d}\mathbf{n}_\beta. \quad (19)$$

Again,

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

Stencil size: away from the boundary, how many non-zero entries are there per row in the $\mathbf{K}$ matrix? $3^2 \times 2 = 18$ in 2D; $3^3 \times 3 = 81$ in 3D.

What does $\mathbf{K}$ look like?

The 8×8 $\mathbf{K}_e$ matrix

```
k=[ 1/2-nu/6   1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
   -1/4+nu/12 -1/8-nu/8  nu/6       1/8-3*nu/8];
[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
  k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
  k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
  k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
  k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
```

After some transforms (e.g. strain-displacement matrix $\mathbf{B}$, stress-train matrix $\mathbf{E}$). (Source: *A 99 line topology optimization code written in Matlab*[1])

[1] O. Sigmund (2001). "A 99 line topology optimization code written in Matlab". In: *Structural and multidisciplinary optimization* 21.2, pp. 120–127.

Table of Contents

1. FEM Overview
2. Discretizing Poisson's equation
3. Discretizing linear elasticity
4. Topology optimization

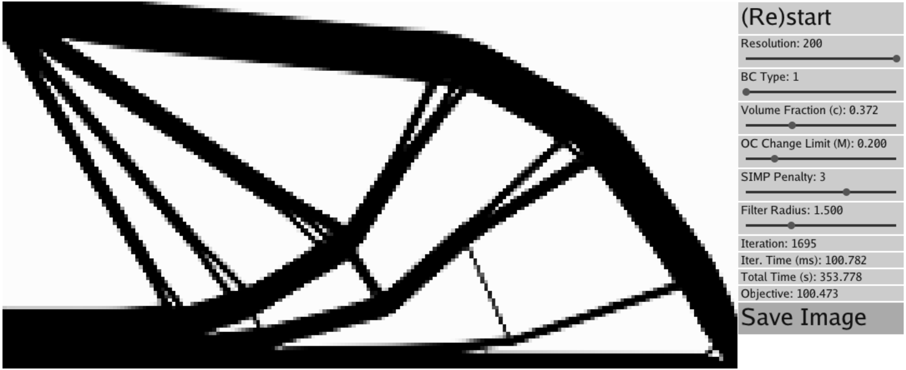
Topology optimization

The minimal compliance topology optimization problem can be formulated as:

$$
\begin{aligned}
\min \quad L(\boldsymbol{\rho}) &= \mathbf{u}^T\mathbf{K}(\boldsymbol{\rho})\mathbf{u} && (20)\\
s.t. \quad \mathbf{K}(\boldsymbol{\rho})\mathbf{u} &= \mathbf{f} && (21)\\
\sum_e \boldsymbol{\rho}_e &\leq cV, && (22)\\
\boldsymbol{\rho}_e &\in [\boldsymbol{\rho}_{\textbf{min}}, 1] && (23)
\end{aligned}
$$

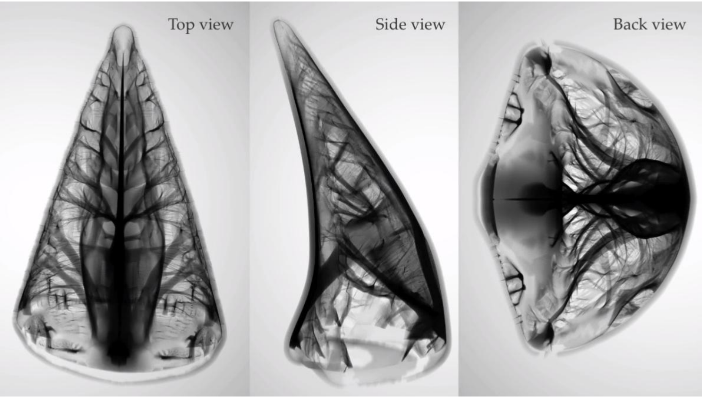
L : measure of deformation energy, or the loss function
c : volume fraction (e.g., 0.3)
$\boldsymbol{\rho}_e$: material occupancy ($0 =$ empty, $1 =$ filled) of cell e.
V: total volume

Topology optimization (Demo)



Check out the supplementary material for more details. Keywords: Solid Isotropic Material with Penalization (SIMP), Optimility Criterion (OC)

Narrow-band TopOpt on a sparsely populated grid



Optimizing $1,040,875,347$ FEM voxels[2]. [Bilibili]

[2] H. Liu et al. (2018). "Narrow-band topology optimization on a sparsely populated grid". In: *ACM Transactions on Graphics (TOG)* 37.6, pp. 1–14.