2. The Finite Element Method

Anders Logg
logg@tti-c.org

Toyota Technological Institute at Chicago

Sixth Winter School in Computational Mathematics
Geilo, March 5-10 2006

Basic example: Poisson's equation

- Strong form: Find $u \in \mathcal{C}^2(\overline{\Omega})$ with $u = 0$ on $\partial\Omega$ such that

$$-\Delta u = f \quad \text{in } \Omega$$

- Weak form: Find $u \in H^1_0(\Omega)$ such that

$$\int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}x = \int_\Omega v \, f \, \mathrm{d}x \quad \text{for all } v \in H^1_0(\Omega)$$

- Canonical form: Find $u \in V$ such that

$$a(v, u) = L(v) \quad \text{for all } v \in \hat{V}$$

with $a : \hat{V} \times V \to \mathbb{R}$ a *bilinear form* and $L : \hat{V} \to \mathbb{R}$ a *linear form* (functional)

Obtaining the discrete system

Let $\hat{V}_h \subset \hat{V}$ and $V_h \subset V$ be discrete function spaces. Then

$$a(v, U) = L(v) \quad \text{for all } v \in \hat{V}_h$$

is a discrete linear system for the approximate solution $U \in V_h$

With $\hat{V} = \mathrm{span}\{\hat{\phi}_i\}_{i=1}^N$ and $V_h = \mathrm{span}\{\phi_i\}_{i=1}^N$ we obtain the linear system

$$AU = b$$

for the degrees of freedom (U_i) of $U = \sum_{i=1}^N U_i \phi_i$ where

$$A_{ij} = a(\hat{\phi}_i, \phi_j)$$
$$b_i = L(\hat{\phi}_i)$$

Outline

- Chapter 3 in lecture notes

The Ciarlet definition

A finite element is a triple

$$(K, \mathcal{P}_K, \mathcal{N}_K)$$

- K is a bounded closed subset of $\mathbb{R}^d$ with nonempty interior and piecewise smooth boundary
- $\mathcal{P}_K$ is a function space on K of dimension $n_K < \infty$
- $\mathcal{N}_K = \{\nu_1^K, \nu_2^K, \ldots, \nu_{n_K}^K\}$ is a basis for $\mathcal{P}_K'$ (the bounded linear functionals on $\mathcal{P}_K$)

The nodes

In the simplest case the nodes are given by evaluation of function values at a set of points $\{x_i^K\}_{i=1}^{n_K}$:

$$\nu_i^K(v) = v(x_i^K), \quad i = 1, 2, \ldots, n_K$$

Other types of nodes:

- directional derivatives at points
- moments of function values or derivatives

The nodal basis

A special basis $\{\phi_i^K\}_{i=1}^{n_K}$ for $\mathcal{P}_K$ that satisfies

$$\nu_i^K(\phi_j^K) = \delta_{ij}, \quad i, j = 1, 2, \ldots, n_K$$

Implies that

$$v = \sum_{i=1}^{n_K} \nu_i^K(v) \phi_i^K$$

for any $v \in \mathcal{P}_K$

Local and global nodes

- Want to define a global function space V_h
- Piece together local function spaces $\{\mathcal{P}_K\}_{K \in \mathcal{T}}$
- Local nodes: $\mathcal{N}_K = \{\nu_i^K\}_{i=1}^{n_K}$
- Global nodes: $\mathcal{N} = \{\nu_i\}_{i=1}^{N}$

Mapping local nodes to global nodes

For each cell $K \in \mathcal{T}$ we define a *local-to-global mapping*

$$\iota_K : [1, n_K] \rightarrow N$$

that specifies how the local nodes $\mathcal{N}_K$ are mapped to the corresponding global nodes $\mathcal{N}$

Evaluation of global nodes:

$$\nu_{\iota_K(i)}(v) = \nu_i^K(v|_K), \quad i = 1, 2, \ldots, n_K$$

for any $v \in V_h$

Definition of V_h

Define V_h as the set of functions on Ω satisfying

$$v|_K \in \mathcal{P}_K \quad \forall K \in \mathcal{T},$$

We also require that for each pair of cells $(K, K') \in \mathcal{T} \times \mathcal{T}$ and local node numbers $(i, i') \in [1, n_K] \times [1, n_{K'}]$ such that

$$\iota_K(i) = \iota_{K'}(i')$$

we have

$$\nu_i^K(v|_K) = \nu_{i'}^{K'}(v|_{K'})$$

Definition

A Lagrange finite element is a triple

$$(K, \mathcal{P}_K, \mathcal{N}_K)$$

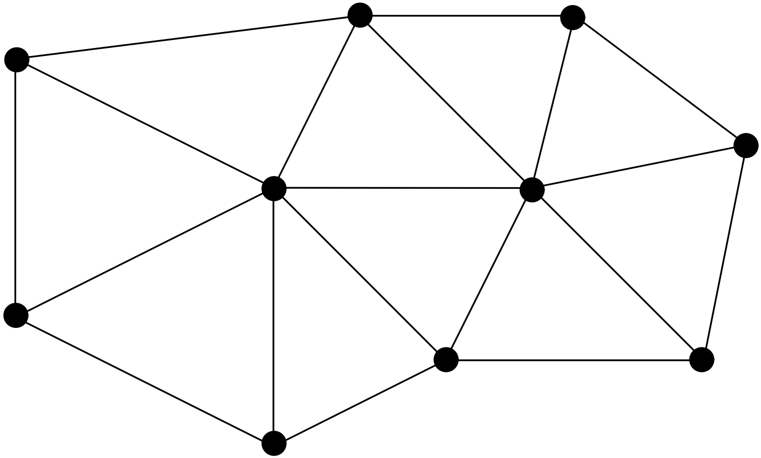
where

- K is a simplex (line, triangle, tetrahedron) in $\mathbb{R}^d$
- $\mathcal{P}_K$ is the space $P_q(K)$ of scalar polynomials of degree $\leq q$
- $\mathcal{N}_K$ is point evaluation (at a set of specific points)

The linear Lagrange element

- K is a line, triangle or tetrahedron
- $\mathcal{P}_K$ is the first-degree polynomials on K
- $\mathcal{N}$ is point evaluation at the vertices

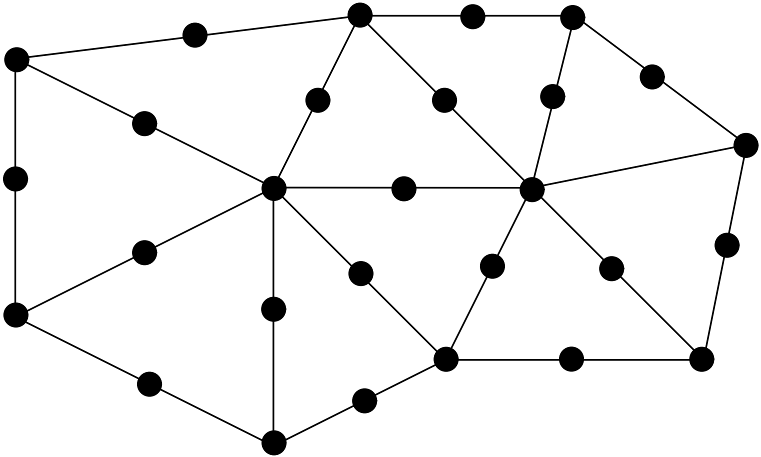
Linear Lagrange elements



The quadratic Lagrange element

- K is a line, triangle or tetrahedron
- $\mathcal{P}_K$ is the second-degree polynomials on K
- $\mathcal{N}$ is point evaluation at the vertices and edge midpoints

Quadratic Lagrange elements



Simplify the description

- Introduce a reference finite element $(K_0, \mathcal{P}_0, \mathcal{N}_0)$
- Generate each $(K, \mathcal{P}_K, \mathcal{N}_K)$ from $(K_0, \mathcal{P}_0, \mathcal{N}_0)$

The mapping $F_K : K_0 \rightarrow K$

Introduce a mapping

$$F_K : K_0 \rightarrow K$$

from the reference cell K_0 to K

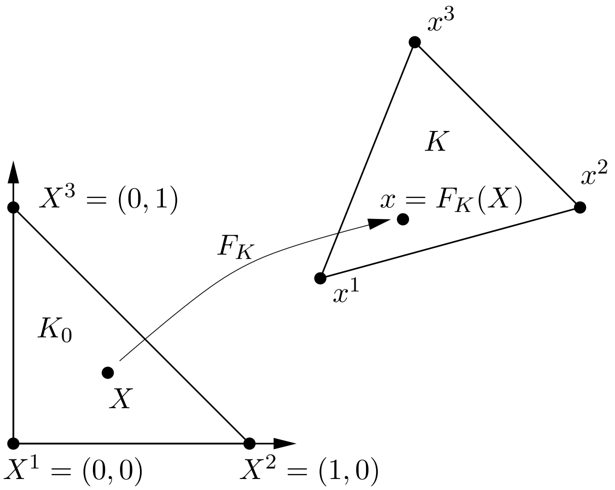
Notation:

$$x = F_K(X) \in K \text{ for } X \in K_0$$

Types of mappings:

- Affine: $F_K(X) = A_K X + b_K$
- Isoparametric: $(F_K)_i \in \mathcal{P}_0$ for $1 \leq i \leq d$

Generating K from K_0



Generating $\mathcal{P}_K$ from $\mathcal{P}_0$

Generate the function space $\mathcal{P}_K$ by

$$\mathcal{P}_K = \{v = v_0 \circ F_K^{-1} : v_0 \in \mathcal{P}_0\}$$

Each function $v = v(x)$ is given by

$$v(x) = v_0(F_K^{-1}(x)) = v_0 \circ F_K^{-1}(x)$$

for some $v_0 \in \mathcal{P}_0$

Generating $\mathcal{N}_K$ from $\mathcal{N}_0$

Generate the nodes $\mathcal{N}_K$ by

$$\mathcal{N}_K = \{\nu_i^K : \nu_i^K(v) = \nu_i^0(v \circ F_K), \quad i = 1, 2, \ldots, n_0\}$$

Each node ν_i^K is given by

$$\nu_i^K(v) = \nu_i^0(v \circ F_K)$$

for some $\nu_i^0 \in \mathcal{N}_0$

Generating $(K, \mathcal{P}_K, \mathcal{N}_K)$ from $(K_0, \mathcal{P}_0, \mathcal{N}_0)$

Generate $(K, \mathcal{P}_K, \mathcal{N}_K)$ by

$$\begin{aligned} K &= F_K(K_0) \\ \mathcal{P}_K &= \{v = v_0 \circ F_K^{-1} : v_0 \in \mathcal{P}_0\} \\ \mathcal{N}_K &= \{\nu_i^K : \nu_i^K(v) = \nu_i^0(v \circ F_K), \quad i = 1, 2, \ldots, n_0\} \end{aligned}$$

The finite element $(K, \mathcal{P}_K, \mathcal{N}_K)$ is generated by

- the reference finite element $(K_0, \mathcal{P}_0, \mathcal{N}_0)$
- the mapping $F_K : K_0 \to K$

Generating the nodal basis

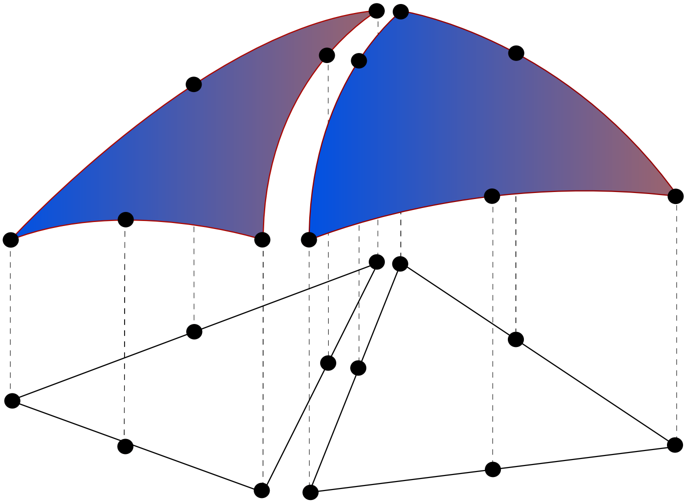
- Let $\{\Phi_i\}_{i=1}^{n_0}$ be a nodal basis for $\mathcal{P}_0$
- Let $\phi_i^K = \Phi_i \circ F_K^{-1}$ for $i = 1, 2, \ldots, n_0$

It follows that

$$\nu_i^K(\phi_j^K) = \nu_i^0(\phi_j^K \circ F_K) = \nu_i^0(\Phi_j) = \delta_{ij}$$

Thus, $\{\phi_i^K\}_{i=1}^{n_K}$ is a nodal basis for $\mathcal{P}_K$

Generating the global function space



The test and trial spaces

The test space:

$$\hat{V}_h = \mathrm{span}\{\hat{\phi}_i\}_{i=1}^N$$

The trial space:

$$V_h = \mathrm{span}\{\phi_i\}_{i=1}^N$$

Note:

$$|\hat{V}_h| = |V_h|$$

The discrete variational problem

Find $U \in V_h$ such that

$$a(U; v) = L(v) \quad \forall v \in \hat{V}_h$$

Semilinear form (linear in second argument):

$$a : V_h \times \hat{V}_h \to \mathbb{R}$$

Linear form (functional):

$$L : \hat{V}_h \to \mathbb{R}$$

The system of discrete equations

The discrete variational problem corresponds to a system of discrete equations:

$$F(U) = 0$$

for the vector (U_i) of degrees of freedom of the solution

$$U = \sum_{i=1}^{N} U_i \phi_i \in V_h$$

where

$$F_i(U) = a(U; \hat{\phi}_i) - L(\hat{\phi}_i), \quad i = 1, 2, \ldots, N$$

Linearization

The Jacobian A of F is given by

$$
\begin{aligned}
A_{ij} = \frac{\partial F_i(U)}{\partial U_j} &= \frac{\partial}{\partial U_j} a(U; \hat{\phi}_i) = a'(U; \hat{\phi}_i) \frac{\partial U}{\partial U_j} \\
&= a'(U; \hat{\phi}_i) \phi_j = a'(U; \hat{\phi}_i, \phi_j)
\end{aligned}
$$

Note that

$$
a'(U; \cdot, \cdot) : \hat{V}_h \times V_h \to \mathbb{R}
$$

is a bilinear form

A nonlinear Poisson equation

Differential equation:

$$\begin{aligned} -\nabla \cdot ((1+u)\nabla u) &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

Variational problem: Find $u \in V$ such that

$$\int_\Omega \nabla v \cdot ((1+u)\nabla u) \, \mathrm{d}x = \int_\Omega v f \, \mathrm{d}x$$

for all $v \in \hat{V}$

The discrete variational problem

Find $U \in V_h$ such that

$$a(U; v) = L(v) \quad \forall v \in \hat{V}_h$$

where

$$a(U; v) = \int_\Omega \nabla v \cdot ((1 + U)\nabla U) \,\mathrm{d}x$$
$$L(v) = \int_\Omega v \, f \,\mathrm{d}x$$

Linearization

Compute (Fréchet) derivative of the semilinear form:

$$a'(U; v, w) = \int_\Omega \nabla v \cdot (w \nabla U) \,\mathrm{d}x + \int_\Omega \nabla v \cdot ((1 + U) \nabla w) \,\mathrm{d}x$$

for any $w \in V_h$

The Jacobian:

$$A_{ij} = a'(U; \hat{\phi}_i, \phi_j) = \int_\Omega \nabla \hat{\phi}_i \cdot (\phi_j \nabla U) \,\mathrm{d}x + \int_\Omega \nabla \hat{\phi}_i \cdot ((1 + U) \nabla \phi_j) \,\mathrm{d}x$$

The discrete variational problem

Find $U \in V_h$ such that

$$a(v, U) = L(v) \quad \forall v \in \hat{V}_h$$

Bilinear form:

$$a : \hat{V}_h \times V_h \rightarrow \mathbb{R}$$

Linear form (functional):

$$L : \hat{V}_h \rightarrow \mathbb{R}$$

Note the relation to the semilinear form a:

$$a(v, U) = a'(U; v, U) = a'(U; v)\, U$$

The system of discrete equations

The discrete variational problem corresponds to a system of discrete equations:

$$AU = b$$

for the vector (U_i) of degrees of freedom of the solution where

$$A_{ij} = a(\hat{\phi}_i, \phi_j),$$
$$b_i = L(\hat{\phi}_i).$$

Note the relation to the nonlinear Jacobian F':

$$A_{ij} = a(\hat{\phi}_i, \phi_j) = a'(U; \hat{\phi}_i, \phi_j)$$

Reduction to multilinear forms

- $a(U;\cdot)$ is a linear form for any fixed $U \in V_h$
- $a'(U;\cdot,\cdot)$ is a bilinear form for any fixed $U \in V_h$

We thus need to consider the following multilinear forms:

$$\begin{aligned} a(U;\cdot) &: \hat{V}_h \rightarrow \mathbb{R} \\ L &: \hat{V}_h \rightarrow \mathbb{R} \\ a'(U,\cdot,\cdot) &: \hat{V}_h \times V_h \rightarrow \mathbb{R} \end{aligned}$$

The general multilinear form

Consider general multilinear forms of arity $r > 0$:

$$a : V_h^1 \times V_h^2 \times \cdots \times V_h^r \rightarrow \mathbb{R}$$

defined on the product space

$$V_h^1 \times V_h^2 \times \cdots \times V_h^r$$

of a given set $\{V_h^j\}_{j=1}^r$ of discrete function spaces on a triangulation $\mathcal{T}$ of a domain $\Omega \subset \mathbb{R}^d$

Some notation

The basis for V_h^j:

$$V_h^j = \text{span}\{\phi_i^j\}_{i=1}^{N^j}$$

Multiindex i of length $|i| = r$:

$$i = (i_1, i_2, \ldots, i_r)$$

The index set $\mathcal{I}$:

$$\begin{aligned}\mathcal{I} &= \prod_{j=1}^{r} [1, |V_h^j|] = [1, |V_h^1|] \times [1, |V_h^2|] \times \cdots \times [1, |V_h^r|] \\ &= \{(1, 1, \ldots, 1), (1, 1, \ldots, 2), \ldots, (N^1, N^2, \ldots, N^r)\}\end{aligned}$$

The global tensor A

The multilinear form a defines a tensor A:

$$A_i = a(\phi^1_{i_1}, \phi^2_{i_2}, \ldots, \phi^r_{i_r}) \quad \forall i \in \mathcal{I}$$

- Rank of A is r
- Dimension of A is $(|V_h^1|, |V_h^2|, \ldots, |V_h^r|) = (N^1, N^2, \ldots, N^r)$
- A is typically sparse

Linear and bilinear forms

- Linear forms:
  - $r = 1$
  - A (or b) is a vector (the "load vector")
- Bilinear forms:
  - $r = 2$
  - A is a matrix (the "stiffness matrix")
- Higher-arity forms:
  - $r > 2$?

A trilinear form

The weighted Poisson's equation:

$$-\nabla \cdot (w \nabla u) = f$$

Discrete variational problem:

$$\int_\Omega w \nabla v \cdot \nabla U \, \mathrm{d}x = \int_\Omega v \, f \, \mathrm{d}x \quad \forall v \in \hat{V}_h$$

A trilinear form

- The trilinear form:

$$a : V_h^1 \times V_h^2 \times V_h^3 \to \mathbb{R}$$

where

$$a(v, U, w) = \int_\Omega w \nabla v \cdot \nabla U \, \mathrm{d}x$$

- The rank three tensor:

$$A_i = \int_\Omega \phi_{i_3}^3 \nabla \phi_{i_1}^1 \cdot \nabla \phi_{i_2}^2 \, \mathrm{d}x$$

The action of a trilinear form

- Expansion of w:

$$w = \sum_{i=1}^{N^3} w_i \phi_i^3$$

- Tensor contraction:

$$A : w = \left(\sum_{i_3=1}^{N^3} A_{i_1 i_2 i_3} w_{i_3} \right)_{i_1 i_2}$$

- Linear system:

$$(A : w)U = b$$

Computing the global tensor A

Need to compute the entries

$$A_i = a(\phi^1_{i_1}, \phi^2_{i_2}, \ldots, \phi^r_{i_r})$$

for all $i \in \mathcal{I}$

- Total number of entries is $N^1 N^2 \cdots N^r$
- Most of the entries are zero (A is sparse)

The naive algorithm

for $i \in \mathcal{I}$

$$A_i = a(\phi^1_{i_1}, \phi^2_{i_2}, \dots, \phi^r_{i_r})$$

end for

- Not efficient
- Each cell visited multiple times
- Need to recompute local data
- Need to explicitly handle sparsity

Assemble contributions

Write the multilinear form as a sum:

$$a = \sum_{K \in \mathcal{T}} a_K$$

It follows that

$$A_i = \sum_{K \in \mathcal{T}} a_K(\phi^1_{i_1}, \phi^2_{i_2}, \ldots, \phi^r_{i_r})$$

For Poisson, we have

$$a_K(v, U) = \int_K \nabla v \cdot \nabla U \, \mathrm{d}x$$

Some more notation

- $\iota_K^j : [1, n_K^j] \to [1, N^j]$ the local-to-global for V_h^j
- $\iota_K : \mathcal{I}_K \to \mathcal{I}$ the collective local-to-global mapping:

$$\iota_K(i) = (\iota_K^1(i_1), \iota_K^2(i_2), \ldots, \iota_K^3(i_3)) \quad \forall i \in \mathcal{I}_K$$

where $\mathcal{I}_K$ is the index set

$$\mathcal{I}_K = \prod_{j=1}^{r} [1, |\mathcal{P}_K^j|] = \{(1, 1, \ldots, 1), \ldots, (n_K^1, n_K^2, \ldots, n_K^r)\}$$

- $\mathcal{T}_i \subset \mathcal{T}$ subset of cells in $\mathcal{T}$ on which all of the basis functions $\{\phi_{i_j}^j\}_{j=1}^r$ are supported

Assemble contributions

$$\begin{aligned} A_i &= \sum_{K \in \mathcal{T}} a_K(\phi^1_{i_1}, \phi^2_{i_2}, \ldots, \phi^r_{i_r}) \\ &= \sum_{K \in \mathcal{T}_i} a_K(\phi^1_{i_1}, \phi^2_{i_2}, \ldots, \phi^r_{i_r}) \\ &= \sum_{K \in \mathcal{T}_i} a_K(\phi^{K,1}_{(\iota^1_K)^{-1}(i_1)}, \phi^{K,2}_{(\iota^2_K)^{-1}(i_2)}, \ldots, \phi^{K,r}_{(\iota^r_K)^{-1}(i_r)}) \end{aligned}$$

Improved assembly algorithm (I)

$A = 0$
for $K \in \mathcal{T}$
  for $i \in \mathcal{I}_K$
    $A_{\iota_K(i)} = A_{\iota_K(i)} + a_K(\phi_{i_1}^{K,1}, \phi_{i_2}^{K,2}, \ldots, \phi_{i_r}^{K,r})$
  end for
end for

The element tensor

Define the *element tensor* A^K by

$$A_i^K = a_K(\phi_{i_1}^{K,1}, \phi_{i_2}^{K,2}, \ldots, \phi_{i_r}^{K,r}) \quad \forall i \in \mathcal{I}_K$$

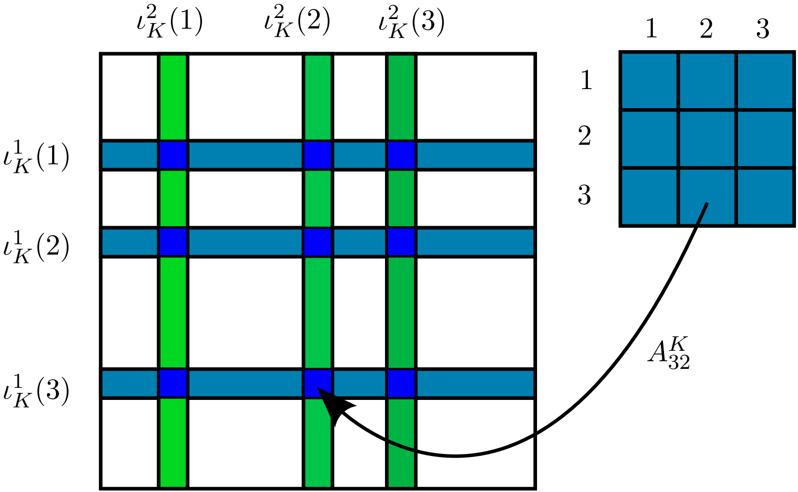
- Rank of A^K is r
- Dimension of A^K is $(|\mathcal{P}_K^1|, |\mathcal{P}_K^2|, \ldots, |\mathcal{P}_K^r|) = (n_K^1, n_K^2, \ldots, n_K^r)$
- A^K is typically dense

Improved assembly algorithm (II)

$A = 0$
for $K \in \mathcal{T}$
    Compute the element tensor A^K
    Add A^K to A according to $\{\iota_K\}_{K \in \mathcal{T}}$
end for

- Optimize computation of A^K (FFC)
- Optimize addition of A^K into A (PETSc)
- Separation of concerns
- Increased granularity

Adding the element tensor A^K



Summary

We need to

- automate the tabulation of nodal basis functions on the reference finite element $(K_0, \mathcal{P}_0, \mathcal{N}_0)$
- automate the computation of the element tensor A^K
- automate the assembly of the global tensor A

Automated by FIAT, FFC and DOLFIN

Upcoming lectures

0. Automating the Finite Element Method
1. Survey of Current Finite Element Software
2. The Finite Element Method
3. Automating Basis Functions and Assembly
4. Automating and Optimizing the Computation of the Element Tensor
5. FEniCS and the Automation of CMM
6. FEniCS Demo Session