Automating 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

Outline

Would you implement a Poisson solver like this?

```
...
for (dof_handler.begin_active(); cell! = dof_handler.end(); ++cell)
{
  for (unsigned int i = 0; i < dofs_per_cell; ++i)
    for (unsigned int j = 0; j < dofs_per_cell; ++j)
      for (unsigned int q_point = 0; q_point < n_q_points; ++q_point)
        cell_matrix(i, j) += (fe_values.shape_grad (i, q_point) *
                              fe_values.shape_grad (j, q_point) *
                              fe_values.JxW(q_point));

  for (unsigned int i = 0; i < dofs_per_cell; ++i)
    for (unsigned int q_point = 0; q_point < n_q_points; ++q_point)
      cell_rhs(i) += (fe_values.shape_value (i, q_point) *
                      <value of right-hand side f> *
                      fe_values.JxW(q_point));
}
```

Or would you implement it like this?

```
for (int i = 1; i <= nbf; i++)
  for (int j = 1; j <= nbf; j++)
    elmat.A(i, j) += (fe.dN(i, 1) * fe.dN(j, 1) +
                      fe.dN(i, 2) * fe.dN(j, 2) +
                      fe.dN(i, 3) * fe.dN(j, 3)) *
                      detJxW;

for (int i = 1; i <= nbf; i++)
  elmat.b(i) += fe.N(i)*<value of f>*detJxW;
```

Or maybe like this?

```
a = dot(grad(v), grad(U))*dx
L = v*f*dx
```

Or perhaps like this?

$$
\begin{aligned}
-\Delta u &= f \quad && \text{in } \Omega \subset \mathbb{R}^d \\
u &= 0 \quad && \text{on } \partial\Omega
\end{aligned}
$$

Differential equation

Poisson's equation:

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \subset \mathbb{R}^d \\ u &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

- Heat transfer: u is the temperature
- Electrostatics: u is the potential
- Appears in some form in most differential equations

Variational problem

Multiply with a test function $v \in \hat{V}$ and integrate by parts:

$$\int_\Omega -v \, \Delta u \, \mathrm{d}x = \int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}x - \int_{\partial\Omega} v \, \partial_n u \, \mathrm{d}s = \int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}x$$

Find $u \in V$ such that

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

Suitable test and trial spaces:

$$\hat{V} = V = H_0^1(\Omega)$$

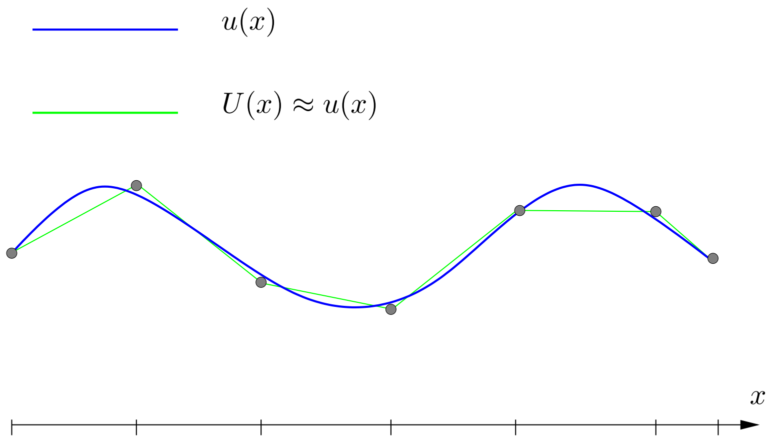
Discrete variational problem

Find $U \in V_h$ such that

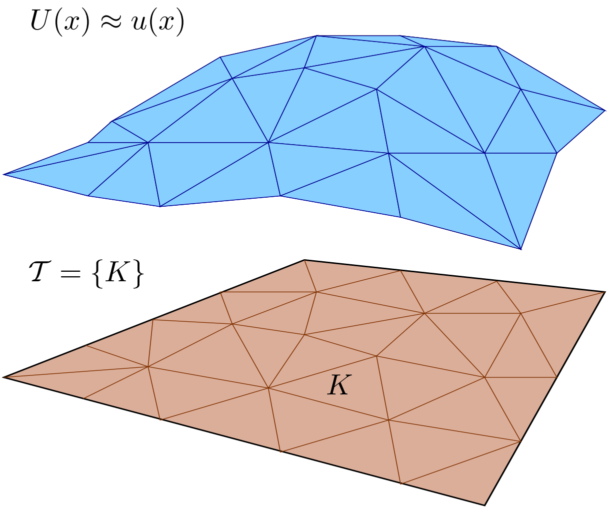
$$\int_\Omega \nabla v \cdot \nabla U \,\mathrm{d}x = \int_\Omega v\, f \,\mathrm{d}x \quad \forall v \in \hat{V}_h$$

- $\hat{V}_h \subset \hat{V}$ and $V_h \subset V$ is a pair of *discrete* function spaces
- Galerkin's method: $R(U) = -\Delta U - f \perp \hat{V}_h$
- FEM: Galerkin with piecewise polynomial spaces

Finite element function spaces



Finite element function spaces



The discrete system

Introduce bases for $\hat{V}_h$ and V_h:

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

Find $U \in V_h$ such that

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

Find $(U_i) \in \mathbb{R}^N$ such that

$$\int_\Omega \nabla \hat{\phi}_i \cdot \nabla (\sum_{j=1}^N U_j \phi_j) \, \mathrm{d}x = \int_\Omega \hat{\phi}_i \, f \, \mathrm{d}x, \quad i = 1, 2, \ldots, N$$

The discrete system

Find $(U_i) \in \mathbb{R}^N$ such that

$$\sum_{j=1}^{N} U_j \int_\Omega \nabla \hat{\phi}_i \cdot \nabla \phi_j \, \mathrm{d}x = \int_\Omega \hat{\phi}_i \, f \, \mathrm{d}x, \quad i = 1, 2, \ldots, N$$

Find $(U_i) \in \mathbb{R}^N$ such that

$$AU = b$$

where

$$A_{ij} = \int_\Omega \nabla \hat{\phi}_i \cdot \nabla \phi_j \, \mathrm{d}x$$
$$b_i = \int_\Omega \hat{\phi}_i \, f \, \mathrm{d}x$$

Canonical form

Find $U \in V_h$ such that

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

Find $(U_i) \in \mathbb{R}^N$ such that

$$AU = b$$

where

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

Implementation

```
element = FiniteElement(...)

v = BasisFunction(element)
U = BasisFunction(element)
f = Function(element)

a = dot(grad(v), grad(U))*dx
L = v*f*dx
```

Automating the finite element method

Compute the discrete system

$$AU = b$$

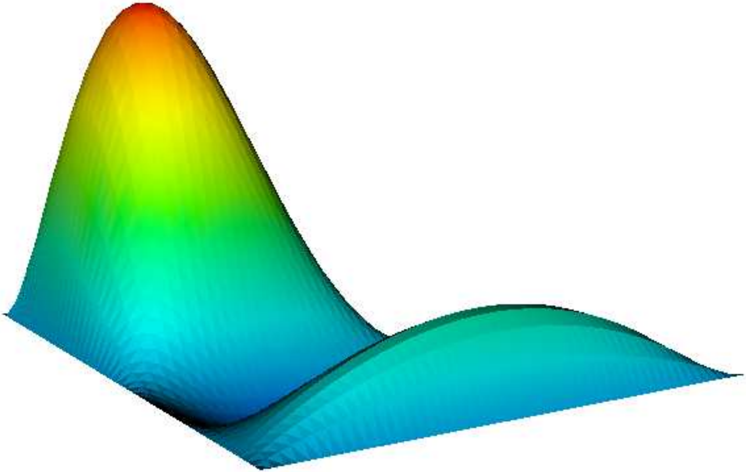
from a given discrete variational problem: Find $U \in V_h$ such that

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

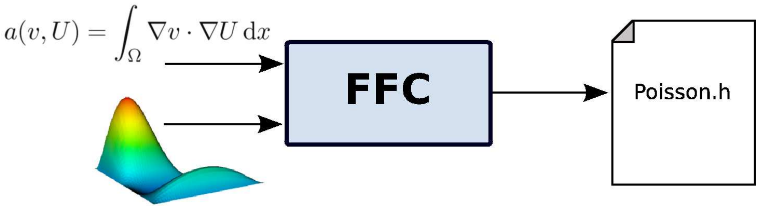
Key steps:

- Automating the tabulation of basis functions
- Automating the computation of the element tensor
- Automating the assembly of the discrete system

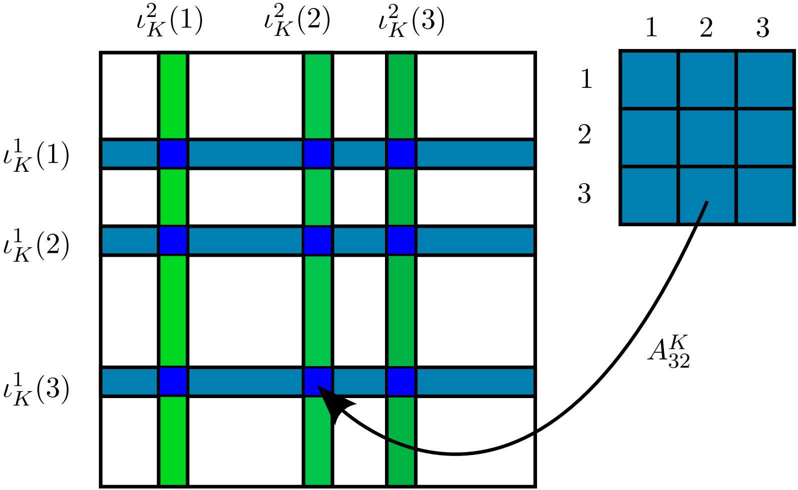
Automating the tabulation of basis functions



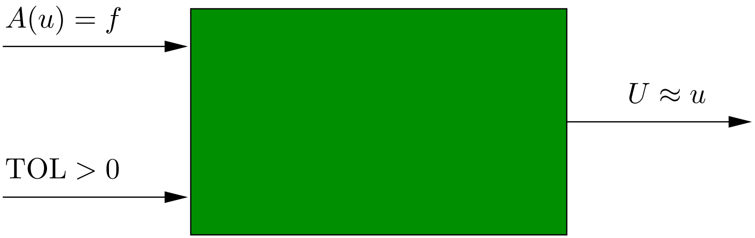
Automating the computation of the element tensor



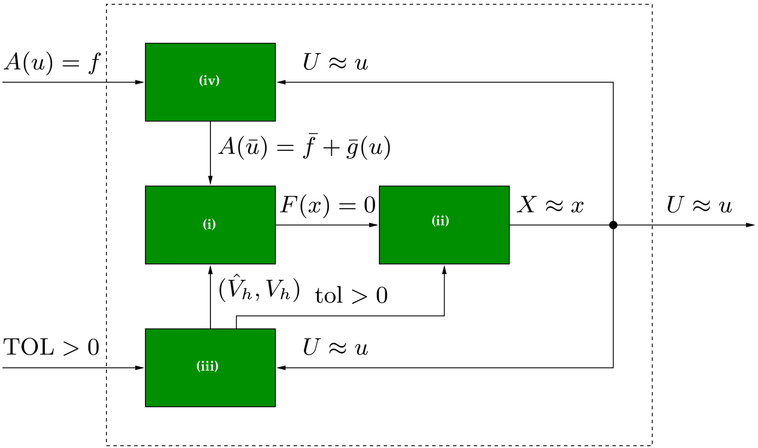
Automating the assembly of the discrete system



The Automation of CMM



Key steps


$A(u) = f$
(iv)
$U \approx u$
$A(\bar{u}) = \bar{f} + \bar{g}(u)$
(i)
$F(x) = 0$
(ii)
$X \approx x$
$U \approx u$
$(\hat{V}_h, V_h)$
tol > 0
TOL > 0
(iii)
$U \approx u$

Sunday March 5

- 16.40–17.25
  Automating the Finite Element Method
- 17.30–18.15
  Hoffman: Navier–Stokes

Monday March 6

- 09.00–10.30
  Hoffman: Navier–Stokes
- 15.45–16.30
  Survey of Current Finite Element Software
- 17.00–18.30
  The Finite Element Method

Tuesday March 7

- 09.00–10.30
  Automating Basis Functions and Assembly
- 15.45–16.30
  Hoffman: Navier–Stokes
- 17.00–18.30
  Hoffman: Navier–Stokes

Wednesday March 8

- 09.00–10.30
  Hoffman: Navier–Stokes
- 15.45–16.30
  Automating and Optimizing the
  Computation of the Element Tensor
- 17.00–18.30
  Automating and Optimizing the
  Computation of the Element Tensor (cont'd)

Thursday March 9

- 09.00–10.30
  FEniCS and the Automation of CMM
- 15.45–16.30
  Hoffman: Navier–Stokes
- 17.00–18.30
  Hoffman: Navier–Stokes

Friday March 10

- 09.00–10.30
  Hoffman: Navier–Stokes
- 11.00–12.30
  FEniCS Demo Session (laptops ready!)

Homework ;-)

Download and install:

- FIAT version 0.2.3
- FFC version 0.3.0
- DOLFIN version 0.6.0

http://www.fenics.org/