LIÈGE université
Sciences Appliquées

Lecture 9 Approximations of Laplace's Equation, Finite Element Method

Mathématiques appliquées (MATH0504-1)

B. Dewals, C. Geuzaine

Learning objectives of this lecture

Apply the finite difference method to Laplace's equation

Understand why iterative linear solvers are useful in this context

Understand the basics of the finite element method

Outline

1. Approximations of Laplace's equation
2. Finite Element Method

LIÈGE université
Sciences Appliquées

1 –Approximations of Laplace’s equation

Finite differences for the Laplace equation

Let's consider Laplace's equation in two dimensions:

$$u_{xx} + u_{yy} = 0$$

Approximating both terms with centered differences leads to

$$\frac{u_{j+1,k} - 2u_{j,k} + u_{j-1,k}}{(\Delta x)^2} + \frac{u_{j,k+1} - 2u_{j,k} + u_{j,k-1}}{(\Delta y)^2} = 0$$

with $u_{j,k}$ an approximation of $u(j\Delta x, k\Delta y)$

Finite differences for the Laplace equation

Choosing $\Delta x = \Delta y$, we get

$$u_{j,k} = \tfrac{1}{4}(u_{j+1,k} + u_{j-1,k} + u_{j,k+1} + u_{j,k-1})$$

Thus $u_{j,k}$ is the *average* of the values at the four neighboring grid points.

The discrete scheme thus has the same *mean value property* as the Laplace equation!

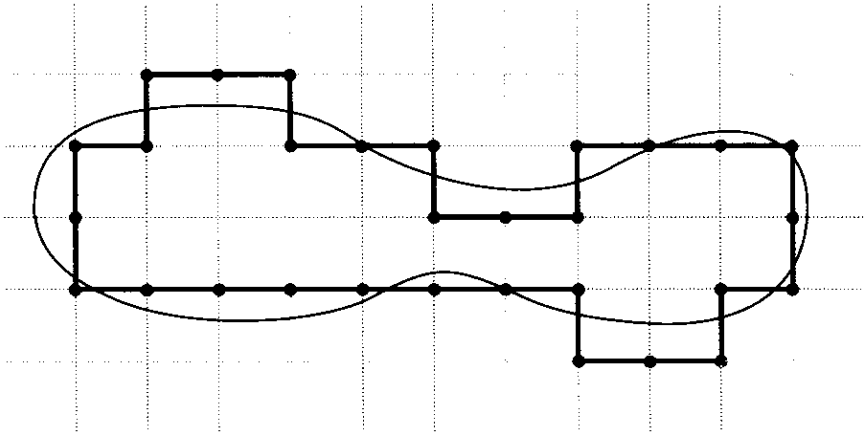
Finite differences for the Laplace equation

A solution $u_{j,k}$ cannot take its maximum or minimum value at an interior point, unless it is a constant: otherwise it couldn't be the average of its neighbors.

Thus the maximum and minimum values can be taken only at the boundary – we thus recover the *maximum principle*!

Finite differences for the Laplace equation

To solve the Dirichlet problem for $u_{xx} + u_{yy} = 0$ in D with given boundary values, we draw a grid covering D and approximate D by a union of squares:



Finite differences for the Laplace equation

In contrast to time-dependent problems, no marching method is readily available: if we have N interior grid points, we obtain a linear system of N linear equations in N unknowns.

The system matrix can become *very* large, but it is

- non singular
- very sparse (many entries are zero)

Finite differences for the Laplace equation

For example, multiplying

$$u_{j,k} = \frac{1}{4}(u_{j+1,k} + u_{j-1,k} + u_{j,k+1} + u_{j,k-1})$$

by 4 leads to the following matrix system for a grid with 4-by-4 interior points:

$$\left[\begin{array}{cccc|cccc|cccc|cccc}
4 & -1 & & & -1 & & & & & & & & & & & \\
-1 & 4 & -1 & & & -1 & & & & & & & & & & \\
 & -1 & 4 & -1 & & & -1 & & & & & & & & & \\
 & & -1 & 4 & & & & -1 & & & & & & & & \\
\hline
-1 & & & & 4 & -1 & & & -1 & & & & & & & \\
 & -1 & & & -1 & 4 & -1 & & & -1 & & & & & & \\
 & & -1 & & & -1 & 4 & -1 & & & -1 & & & & & \\
 & & & -1 & & & -1 & 4 & & & & -1 & & & & \\
\hline
 & & & & -1 & & & & 4 & -1 & & & -1 & & & \\
 & & & & & -1 & & & -1 & 4 & -1 & & & -1 & & \\
 & & & & & & -1 & & & -1 & 4 & -1 & & & -1 & \\
 & & & & & & & -1 & & & -1 & 4 & & & & -1 \\
\hline
 & & & & & & & & -1 & & & & 4 & -1 & & \\
 & & & & & & & & & -1 & & & -1 & 4 & -1 & \\
 & & & & & & & & & & -1 & & & -1 & 4 & -1 \\
 & & & & & & & & & & & -1 & & & -1 & 4
\end{array}\right]
\left[\begin{array}{c} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{4,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{4,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \\ u_{4,3} \\ u_{1,4} \\ u_{2,4} \\ u_{3,4} \\ u_{4,4} \end{array}\right]
=
\left[\begin{array}{c} b_{1,1} \\ b_{2,1} \\ b_{3,1} \\ b_{4,1} \\ b_{1,2} \\ b_{2,2} \\ b_{3,2} \\ b_{4,2} \\ b_{1,3} \\ b_{2,3} \\ b_{3,3} \\ b_{4,3} \\ b_{1,4} \\ b_{2,4} \\ b_{3,4} \\ b_{4,4} \end{array}\right]$$

Jacobi iteration

Instead of direct solution methods (LU factorization), various *iterative* methods can be designed.

Jacobi iteration: starting from an initial guess $u_{j,k}^{(1)}$, successively solve

$$u_{j,k}^{(n+1)} = \frac{1}{4}\left[u_{j+1,k}^{(n)} + u_{j-1,k}^{(n)} + u_{j,k+1}^{(n)} + u_{j,k-1}^{(n)}\right]$$

Convergence analysis and matrix form: cf. your numerical analysis class from last year.

Jacobi iteration

PS:

$$u_{j,k}^{(n+1)} = \frac{1}{4}\left[u_{j+1,k}^{(n)} + u_{j-1,k}^{(n)} + u_{j,k+1}^{(n)} + u_{j,k-1}^{(n)}\right]$$

is actually the same as as solving the 2D diffusion equation $v_t = v_{xx} + v_{yy}$ using centered differences for v_{xx} and v_{yy} and using the forward difference for v_t, with $\Delta x = \Delta y$ and $\Delta t = (\Delta x)^2/4$.

In effect, the Jacobi iteration is solving the Laplace problem by taking the limit of the discretized diffusion equation solution $\mathrm{v}(x, t)$ as $t \to \infty$

Gauss-Seidel method

If we compute $u_{j,k}^{(n+1)}$ one row at a time starting at the bottom row, and going left to right, by directly using any computed value, we get the Gauss-Seidel method:

$$u_{j,k}^{(n+1)} = \frac{1}{4}\left[u_{j+1,k}^{(n)} + u_{j-1,k}^{(\boldsymbol{n+1})} + u_{j,k+1}^{(n)} + u_{j,k-1}^{(\boldsymbol{n+1})}\right]$$

What's next?

These are all so-called *stationary* iterative methods, whose general matrix form for solving $Ax = b$ is

$$x_n = x_{n-1} + K^{-1}(b - Ax_{n-1})$$

If A is decomposed into its diagonal, lower triangular and upper triangular parts D, L and U, we have

- Jacobi: $K = D$
- Gauss-Seidel: $K = D + L$

What's next?

Next week we will introduce more general, non-stationary methods, whose general form is

$$x_n = x_{n-1} + \sum_{k<n} K^{-1}(b - Ax_k)\alpha_{k,n-1}$$

We will study in more detail the Congugate Gradient method, which is part of the family of so-called *Krylov subspace methods*.

LIÈGE université
Sciences Appliquées

2 – Finite Element Method

A simple 1D example

Consider the following (ordinary) differential equation

$$-\frac{d}{dx}\left(a(x)\frac{du}{dx}\right) = f(x)$$

where $a(x)$ and $f(x)$ are given functions.

Let's try to solve the boundary value problem that consists in solving the above equation on an interval $D = [0, L]$ with homogenous Dirichlet boundary conditions $u(0) = u(L) = 0$ on both extremities.

A simple 1D example

Recall the introduction to distributions from last week.

Introducing smooth test functions $v(x)$ vanishing on 0 and L, we reformulate the problem as finding $u(x)$ such that

$$-\int_0^L \frac{d}{dx}\left(a(x)\frac{du}{dx}\right)v(x)\,dx = \int_0^L f(x)v(x)\,dx$$

holds for all test functions $v(x)$. This is only valid in the classical sense if $a(x)$ is smooth enough.

What if $a(x)$ is only piece-wise continuous?

A simple 1D example

We can proceed exactly as last week when we considered differential equations in the sense of distributions!

Integrating by parts leads to

$$\int_0^L a(x)\frac{du}{dx}\frac{dv}{dx}\,dx - \left[a(x)\frac{du}{dx}v(x)\right]_0^L = \int_0^L f(x)v(x)\,dx$$

i.e.

$$\int_0^L a(x)\frac{du}{dx}\frac{dv}{dx}\,dx = \int_0^L f(x)v(x)\,dx$$

which is the weak form of the problem.

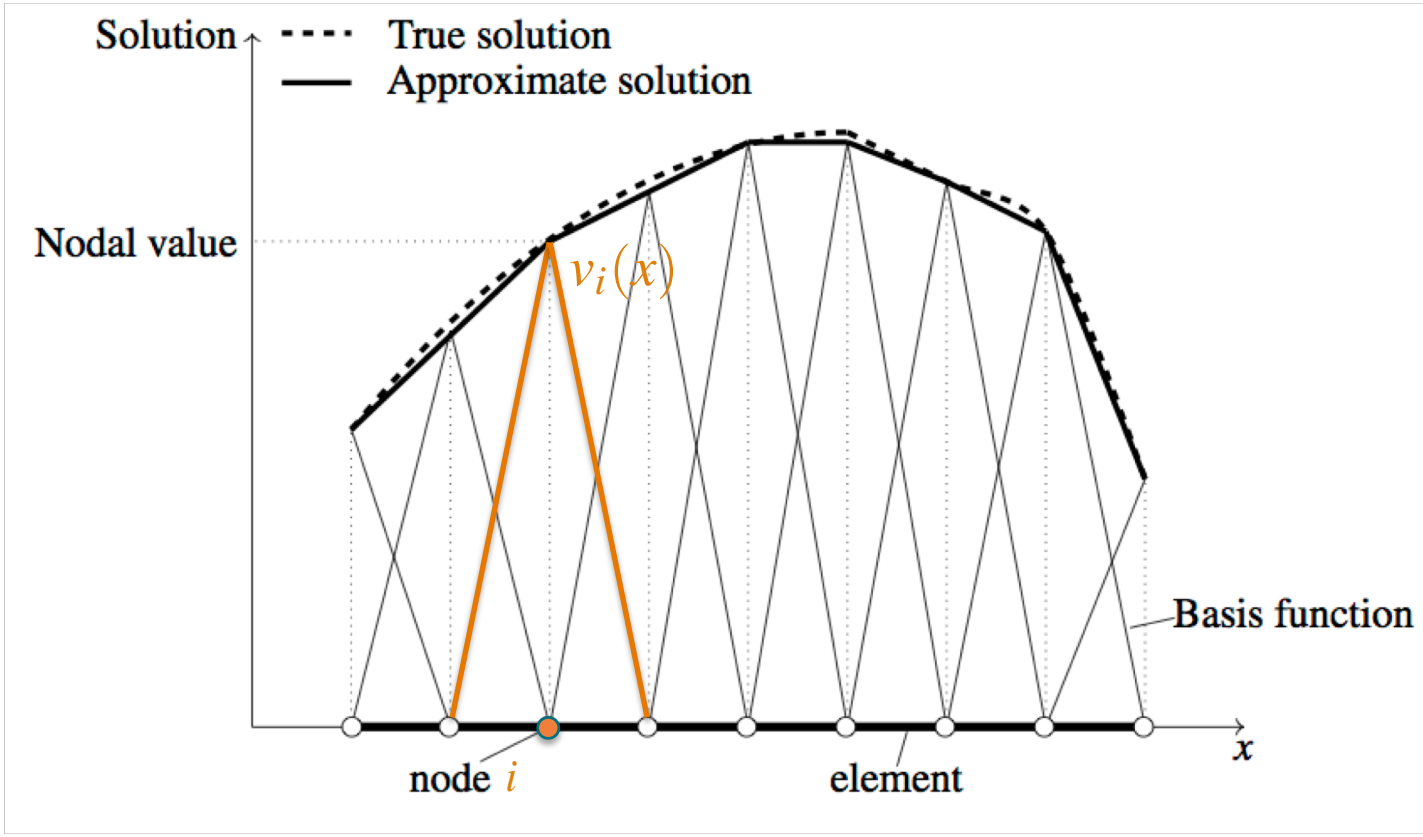
A simple 1D example

The finite element method then consists in

- Discretizing the domain D into several elements (here line segments)
- Approximating the solution $u(x)$ by a linear combination of basis functions (usually piecewise polynomials), with local support on elements
- Choosing a finite number of test functions (usually the same as the basis functions used to approximate $u(x)$)

A simple 1D example

For our problem, we can e.g. choose piecewise linear functions $v_i(x)$ and write: $u(x) \approx U_1 v_1(x) + \cdots + U_N v_N(x)$



A simple 1D example

Using the N basis functions also as test functions,

$$\int_0^L a(x)\frac{du}{dx}\frac{dv}{dx}\,dx = \int_0^L f(x)v(x)\,dx$$

becomes

$$\sum_{i=1}^{N} U_i \int_0^L a(x)\frac{dv_i}{dx}\frac{dv_j}{dx}\,dx = \int_0^L f(x)v_j(x)\,dx \quad (j = 1,\ldots,N)$$

This is a system of N linear equations in the N unknowns U_1, ..., U_N:

$$\sum_{i=1}^{N} m_{ij}U_i = f_j \qquad (j = 1,\ldots,N)$$

A simple 1D example

The entries of the system matrix and the right hand side are respectively

$$m_{ij} = \int_0^L a(x) \frac{dv_i}{dx} \frac{dv_j}{dx} dx \qquad f_j = \int_0^L f(x) v_j(x) dx$$

With piecewise linear basis and test functions, the system matrix is very sparse: $m_{ij} = 0$ if i and j are not neighboring nodes!

(The matrix is actually *the same* as for a centered finite difference scheme if $a(x)$ is a constant.)

Another 1D example

Consider solving the diffusion problem

$$u_t = \kappa u_{xx} + f(x,t); \qquad u = 0 \text{ at } x = 0, l; \quad u = \phi(x) \text{ at } t = 0$$

Partition the interval $[0, l]$ into l equal subintervals. We assign the piecewise linear basis function $\mathrm{v}_j(x)$ to each of the $N = l - 1$ interior vertices.

Multiply the diffusion equation by any function $v(x)$ that vanishes on the boundary and integrating by parts, we get:

$$\frac{d}{dt}\int_0^l uv\,dx = -\kappa\int_0^l \frac{\partial u}{\partial x}\frac{dv}{dx}\,dx + \int_0^l f(x,t)\,v(x)\,dx$$

Another 1D example

In order to use the finite-element method, we look for a solution of the form

$$u(x,t) = \sum_{i=1}^{N} U_i(t)\, v_i(x)$$

and we only require the weak form to hold for v = v_1, ..., v_N. Then

$$\sum_{i=1}^{N}\left(\int_0^l v_i v_j dx\right)\frac{dU_i}{dt} = -\kappa\sum_{i=1}^{N}\left(\int_0^l \frac{dv_i}{dx}\frac{dv_j}{dx}dx\right)U_i(t) + \int_0^l f(x,t)v_j(x)\,dx$$

This is a system of ODEs for U_1(t), ..., U_N(t).

Another 1D example

If we define the entries of vectors U and F as

$$u_j = U_j, \qquad f_j = \int_0^l f(x,t)v_j(x)dx$$

and entries of matrices K and M as

$$k_{ij} = \int_0^l v_i v_j \, dx, \qquad m_{ij} = \int_0^l \frac{dv_i}{dx}\frac{dv_j}{dx} \, dx$$

the system of N ODEs in N unknowns takes the simple vector form

$$K\frac{dU}{dt} = -\kappa MU(t) + F(t)$$

Another 1D example

M is called the stiffness matrix, K the mass matrix, and F the forcing vector.

One can easily show that in this 1D problem with unit length elements, we get

$$M = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ & \cdots & & & \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}, \quad K = \begin{pmatrix} \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \cdots & 0 \\ & \cdots & & & \\ 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} \end{pmatrix}$$

Another 1D example

The system of ODEs can be solved e.g. with the backward Euler method, using the initial condition

$$U_i(0) = \Phi_i \equiv \int_0^l \phi(x) v_i(x)\, dx$$

leading to

$$K\left[\frac{U^{(p+1)} - U^{(p)}}{\Delta t}\right] = -\kappa M U^{(p+1)} + F(t_{p+1})$$

i.e.

$$[K + \kappa \Delta t M]\, U^{(p+1)} = K U^{(p)} + \Delta t\, F(t_{p+1})$$

which is solved recursively for $U^{(1)}, U^{(2)}, \ldots$

Generalization to higher dimensions

The generalization is natural: higher dimensional domains are subdivided into simple polygons (in 2D) or polyhedral (in 3D)

In addition to the rigorous handling of discontinuities, the method then also allows to handle curved or irregularly shaped domains – which are not easily handled with finite differences.

Generalization to higher dimensions

Let's consider the Dirichlet problem for Poisson's equation in the plane

$$-\Delta u = f \quad \text{in } D, \quad u = 0 \quad \text{on bdy } D$$

We proceed as in 1D: restate the problem as finding u such that

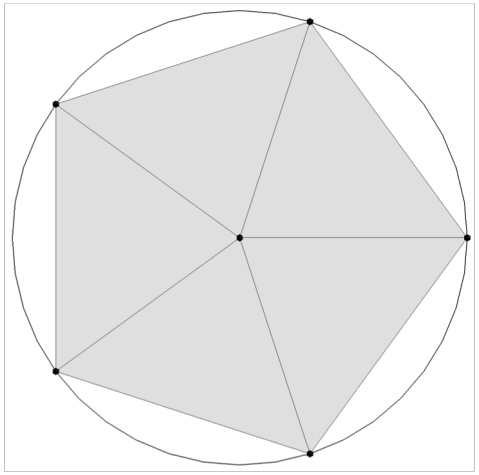
$$-\iint_D \Delta u\, v\, dxdy = \iint_D f\, v\, dxdy$$

holds for any test function v (vanishing on bdy D). Recalling that $\Delta := \nabla \cdot \nabla$ and integrating by parts, we get:

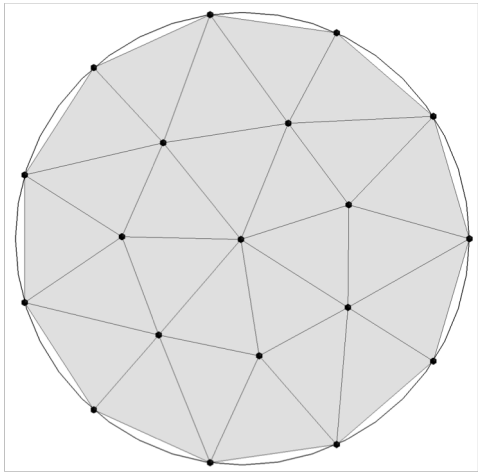
$$\iint_D \nabla u \cdot \nabla v\, dxdy - \int_{\text{bnd } D} (\mathbf{n} \cdot \nabla u) v\, dl = \iint_D f\, v\, dxdy$$

Generalization to higher dimensions

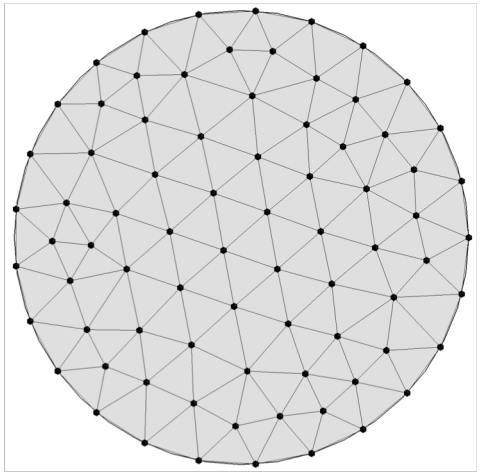
We then triangulate the domain with N interior nodes



N=1



N=8



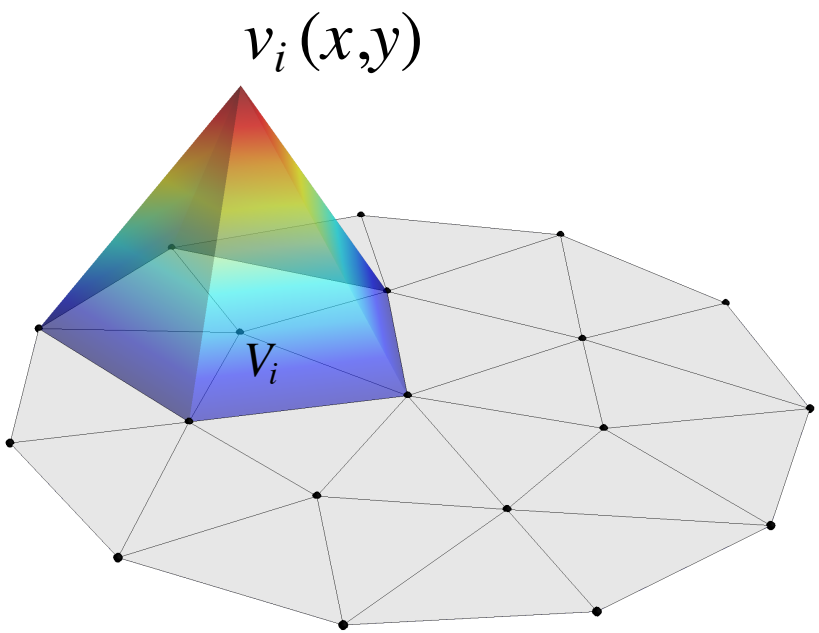
N=53

and approximate

$$u(x,y) \approx u_N(x,y) = U_1 v_1(x,y) + \cdots + U_N v_N(x,y)$$

Generalization to higher dimensions

The simplest basis functions are again piecewise linear: a basis function $v_i(x,y)$ is equal to 1 at "its" node i and to equal 0 at all the other nodes



Generalization to higher dimensions

Using the N basis functions also as test functions,

$$\iint_D \nabla u \cdot \nabla v \, dx dy = \iint_D f \, v \, dx dy$$

Becomes

$$\sum_{i=1}^{N} U_i \left(\iint_D \nabla v_i \cdot \nabla v_j \, dx \, dy \right) = \iint_D f \, v_j \, dx \, dy$$

This is a system of N linear equations ($j = 1, \ldots, N$) in the N unknowns $U_1, \ldots, U_N$.

Generalization to higher dimensions

Denoting

$$m_{ij} = \iint_D \nabla v_i \cdot \nabla v_j \, dx \, dy \quad \text{and} \quad f_j = \iint_D f v_j \, dx \, dy$$

the system again takes the form

$$\sum_{i=1}^{N} m_{ij} U_i = f_j \qquad (j = 1, \ldots, N)$$

Also notice that, at a node $V_k = (x_k, y_k)$,

$$u_N(x_k, y_k) = U_1 v_1(x_k, y_k) + \cdots + U_N v_N(x_k, y_k) = U_k$$

since $v_i(x_k, y_k)$ equals 0 for $i \neq k$ and equals 1 for $i = k$.

Thus *the coefficients are precisely the values of the approximate solution at the vertices.*

Other generalizations

- High-order basis functions
- Different basis and test functions (“Petrov-Galerkin” methods)
- Non-Lagrange basis functions (associated e.g. with edges of the mesh)
- Curved meshes
- Adaptive mesh refinement
- …

Take-home messages

- Approximations of Laplace's equation using finite differences and finite elements lead to sparse systems of linear equations
- Finite element methods provide a general mathematical framework to handle more complicated cases: discontinuities, curved or complex geometries, local refinements, …

Next week

- We will start Part 2 of the course on “Complements of linear algebra” by examining the Conjugate Gradient (CG) method
- CG is a powerful method to solve linear systems iteratively; it is in particular well-suited for the very large sparse matrices originating from the approximation of PDEs