INTRODUCTION

Scientists and engineers try to understand physical phenomena by developing models and by obtaining solutions to those models. The relationship between the reality and the mathematical model is described by Validation. Often, the mathematical models are characterized by a system of coupled nonlinear partial differential equations. As it is impossible in most cases to obtain analytical forms of the solutions to those PDEs, the main tool for obtaining those solutions is the use of numerical methods. The relationship between the mathematical model and the approximate solution is described by Verification.

Validation: Validation is the process determining if the mathematical model describes sufficiently well the reality with respect to the decision that has to be made.

Verification: Verification is a process of determining if the numerical treatment and its implementation lead to the prediction with sufficient accuracy, i.e. the difference between the exact and computed quantity of interest is sufficiently small.

This process is done in several steps, described below.

Step one: the starting point is a system of PDEs. We will consider the following:

$$-\Delta u = f \tag{1}$$

$$\frac{\partial u}{\partial t} - \Delta u = f \tag{2}$$

Equation (1) is an elliptic equation, also called diffusion equation. The exact solution u only depends on the spatial variable $x \in \mathbb{R}^n$. Equation (2) is a parabolic equation, also refered to as the heat equation. The exact solution depends on both space x and time t.

Step two: The computational domain (denoted by Ω) is partitioned into a grid (or mesh) of cells (or elements) of a given size (usually denoted by h). In 1D, we will have a partition $x_0 < x_1 < \cdots < x_N$ with $x_i = ih$ for a uniform grid. In 2D, the cells can be triangles or quadrilaterals. An example of mesh is given in Fig. 1. In 3D, the elements can be prisms, tetrahedra, hexahedra... For time-dependent problem (such as the heat equation above), the time interval $[0, T]$ is also discretized, simply by $t^i = i\Delta t$, where Δt measures the fineness of the time partition. As h goes to zero, the number of cells increases and similarly as Δt goes to zero, the number of time steps increases. Overall, the size of the numerical problem to solve increases as both $h, \Delta t$ decrease.

Step three: Using a given numerical method, we obtain a linear system of the form $AU_{h,\Delta t} = b$ where the components of the vector $U_{h,\Delta t}$ approximate the exact solution u in some way. A good numerical method is such that

$$\lim_{\substack{h \to 0 \\ \Delta t \to 0}} \|U_{h,\Delta t} - u\| = 0$$

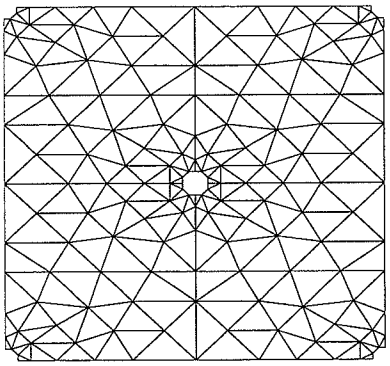

Figure 1: Example of mesh (grid) with triangular elements (cells).

in some norm. We say that the numerical error decreases and the numerical method converges.

Step four: Using numerical linear algebra, we solve the linear system $AU_{h,\Delta t} = b$ as efficiently as possible. For instance, one can use LU factorization, or conjugate gradient, or GMRES... In this class, we will simply use the Matlab command $U_{h,\Delta t} = A\ b$. Another possibility is to use Lapack, or routines from the package PetsC.

Step five: Finally, we postprocess the vector solution $U_{h,\Delta t}$ to obtain the quantity of interest. For instance, we want to plot the numerical solution, or we might want to compute its average over the whole domain.

During the semester we will see different numerical methods, focusing on finite difference and finite elements. We will compare those methods by pointing out their advantages/disadvantages. In particular, we will evaluate the cost versus accuracy property.

I. FINITE DIFFERENCE APPROXIMATIONS

An important tool is the Taylor series of a function u about a point x. Assume that $u \in C^{k+1}(a,b)$ and let $h > 0$.

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \cdots + \frac{h^k}{k!}u^{(k)}(x) + \frac{h^{k+1}}{(k+1)!}u^{(k+1)}(\xi)$$

for some $\xi \in (x, x+h)$.

Using the Taylor series, we can approximate derivatives by finite differences.

1. First-order derivatives

A one-sided approximation uses the function u evaluated at x and $x+h$:

$$D_+u(x) = \frac{u(x+h) - u(x)}{h}$$

If we use the function evaluated at x and $x-h$, we obtain another one-sided approximation:

$$D_-u(x) = \frac{u(x) - u(x-h)}{h}$$

We can evaluate the function at both sides of x, and we obtain the centered approximation:

$$D_0u(x) = \frac{u(x+h) - u(x-h)}{2h}$$

The quantities $D_+u(x), D_-u(x), D_0u(x)$ all approximate the derivative $u'(x)$.
How accurate are those approximations? We need to compute the errors $D_+u(x) - u'(x), D_-u(x) - u'(x)$ and $D_0u(x) - u'(x)$. This is done by using Taylor series.

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(\xi)$$

Equivalently, we have:

$$D_+u(x) - u'(x) = \frac{h}{2}u''(\xi) = \mathcal{O}(h)$$

The error goes to zero with h. We say that D_+ is a first-order finite difference approximation of $u'(x)$. Similarly, we can show that $D_-u(x) - u'(x) = \mathcal{O}(h)$. For $D_0u(x)$, we write two Taylor series:

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{3!}u'''(\xi_1)$$
$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{3!}u'''(\xi_2)$$

So, we have

$$D_0u(x) - u'(x) = \frac{h^2}{3!}(u'''(\xi_1) + u'''(\xi_2)) = \mathcal{O}(h^2)$$

The centered finite difference approximation is of second order, it is more accurate than the one-sided approximations.

2. Second-order and higher order derivatives

Consider the centered finite difference approximation:

$$D^2u(x) = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2}$$

One can show that

$$D^2u(x) - u''(x) = \frac{h^2}{12}u^{(4)}(x) + \mathcal{O}(h^4)$$

We also remark that:

$$D^2u = D_+D_-u$$

So to obtain a second order finite difference of $u''(x)$ we have applied twice a first order finite difference of $u'(x)$.
Can we always obtain a high order finite difference by applying several times a lower order finite difference? The answer is no, as shown by the following example.

Ex 1: let us compute D_+D^2u, which is a combination of one first order and one second order approximations. Let $f(x) = D^2u(x)$.

$$D_+f = \frac{1}{h}(f(x+h) - f(x))$$

$$= \frac{1}{h}\Big(\frac{1}{h^2}(u(x) - 2u(x+h) + u(x+2h)) - \frac{1}{h^2}(u(x-h) - 2u(x) + u(x+h))\Big)$$

$$= \frac{1}{h^3}(3u(x) - 3u(x+h) + u(x+2h) - u(x-h))$$

$$= u'''(x) + \frac{h}{2}u^{(4)}(x) + \mathcal{O}(h^2)$$

where the last equality is obtained by writing several Taylor series. This example yields a first order finite difference approximation of $u'''(x)$.

Ex 2: Let us compute $D_0D_+D_-u$.

$$D_0D_+D_-u(x) = \frac{1}{2h^3}\big(u(x+2h) - 2u(x+h) + 2u(x-h) - u(x-2h)\big) = u'''(x) + \frac{h^2}{4}u^{(4)}(x) + \mathcal{O}(h^4)$$

We have obtained a second order finite difference approximation of $u'''(x)$.

3. A general method: undetermined coefficients

Assume we want to approximate $u^{(k)}(x)$ by a finite difference involving function evaluations at n given points: $x_1, \ldots, x_n$.

$$u^{(k)}(x) \approx c_1 u(x_1) + c_2 u(x_2) + \cdots + c_n u(x_n)$$

We assume that $n \geq k+1$ and that

$$\max_{1 \leq i \leq n} |x - x_i| \leq Ch$$

for some positive constant C.

Ex 3: Assume we want

$$u'(x) \approx c_1 u(x) + c_2 u(x-h) + c_3 u(x-2h)$$

Here we have 3 given points: $x_1 = x, x_2 = x - h, x_3 = x - 2h$. We would like to set up a linear system $\boldsymbol{Ac} = \boldsymbol{b}$ where the vector $\boldsymbol{c}$ has the c_i's for components. We write the Taylor series at each point.

$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''(x) + \ldots$$
$$u(x-2h) = u(x) - 2hu'(x) + 2h^2u''(x) - \frac{8h^3}{6}u'''(x) + \ldots$$

Multiply by the coefficients c_i's and add:

$$c_1 u(x) + c_2 u(x-h) + c_3 u(x-2h) = (c_1 + c_2 + c_3)u(x) - h(c_2 + 2c_3)u'(x)$$
$$+h^2(\frac{1}{2}c_2 + 2c_3)u''(x) - \frac{1}{6}(c_2 + 8c_3)h^3u'''(x) + \mathcal{O}(h^4)$$

We want to obtain the highest possible order approximation of $u'(x)$. Since we have 3 unknowns, we need 3 equations.

$$c_1 + c_2 + c_3 = 0$$
$$-h(c_2 + 2c_3) = 1$$
$$h^2(\frac{1}{2}c_2 + 2c_3) = 0$$

Solving for the coefficients yields

$$c_1 = \frac{3}{2h}, \quad c_2 = \frac{-2}{h}, \quad c_3 = \frac{1}{2h}$$

The remaining term is $-\frac{1}{6}(c_2 + 8c_3)h^3u'''(x) = \frac{1}{12}h^2u'''(x)$. So we have obtained:

$$u'(x) = \frac{3}{2h}u(x) - \frac{2}{h}u(x-h) + \frac{1}{2h}u(x-2h) + \mathcal{O}(h^2).$$

It is a second order finite difference approximation.

The method of undetermined coefficients is based on the Taylor series written at each point x_i.

$$u(x_i) = u(x) + (x_i - x)u'(x) + \cdots + \frac{1}{j!}(x_i - x)^j u^{(j)}(x) + \ldots$$

Multiply this equation by the coefficient c_i and sum over all i.

$$\sum_{i=1}^{n} c_i u(x_i) = (\sum_{i=1}^{n} c_i)u(x) + u'(x)(\sum_{i=1}^{n} c_i(x_i - x)) + \cdots + u^{(j)}(x)\frac{1}{j!}(\sum_{i=1}^{n} c_i(x_i - x)^j) + \ldots$$

We want this expression to be equal to $u^{(k)}(x) + \mathcal{O}(h^p)$ for p as high as possible. In fact, we expect the order of the method $p \geq n - k$. We set for $0 \leq j \leq n-1$ (to get a square system)

$$\frac{1}{j!}(\sum_{i=1}^{n} c_i(x_i - x)^j) = \begin{cases} 0 \text{ if } j \neq k \\ 1 \text{ if } j = k \end{cases}$$

Let $\xi_i = x_i - x$. The system of equations is equivalent to $\boldsymbol{Ac} = \boldsymbol{b}$ with

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & \cdot & \cdot & 1 \\ \xi_1 & \xi_2 & \cdot & \cdot & \xi_n \\ \xi_1^2 & \xi_2^2 & \cdot & \cdot & \xi_n^2 \\ \cdot & & & & \\ \cdot & & & & \\ \xi_1^{n-1} & \xi_2^{n-1} & \cdot & \cdot & \xi_n^{n-1} \end{pmatrix}$$

We obtain a VanderMonde matrix, which has a large condition number for n large.

The following Matlab code `fdcoeffV` returns a vector containing coefficients to approximate $u^{(k)}(xbar)$ using finite differences at $x(1), x(2), \ldots, x(n)$. Here n is simply the length of the vector x. The code is available on the class web site.

```
function c=fdcoeffV(k,xbar,x)

n=length(x);
A=ones(n,n);
xrow = (x(:)-xbar)';
for i=2:n
   A(i,:)=(xrow .^ (i-1)) ./ factorial(i-1);
end

b=zeros(n,1);
b(k+1) = 1;

c=A\b;
c=c';
```

If we run the code, we can obtain the finite difference approximations seen in this chapter. The following script

```
h=1;
xbar=h;
x=[0 h];
k=1;
fdcoeffV(k,xbar,x)
```

gives the result:

```
ans = -1    1
```

You should check that this is indeed a one-sided approximation of $u'(h)$ already seen. If one changes the value for h, then we see its influence in the coefficients. If we repeat the script above with h=0.1, we obtain

```
ans = -10    10
```

This corresponds to the FD approximation:

$$u'(h) \approx \frac{-u(0) + u(h)}{h}$$

Another example is:

```
h=1;
xbar=h;
x=[0 h 2*h];
k=1;
fdcoeffV(k,xbar,x)
```

This yields

```
ans = -0.5   0 0.5
```

This corresponds to the FD approximation:

$$u'(h) \approx \frac{-u(0) + u(2h)}{2h}$$

If we simply change the value xbar to xbar=0 and repeat, we obtain

```
ans=-1.5   2   -0.5
```

This corresponds to the FD approximation:

$$u'(0) \approx -\frac{3}{2h}u(0) + \frac{2}{h}u(h) - \frac{1}{2h}u(2h)$$

A more stable algorithm for finding the coefficients is given by the code fdcoeffF.m available on the class web site (it is based on an algorithm developed by Fornberg).

II. TWO-POINT BOUNDARY VALUE PROBLEM

1. Dirichlet problem

We want to solve the second order problem:

$$\begin{aligned} -u''(x) &= f(x) \quad 0 < x < 1 \\ u(0) &= \alpha \\ u(1) &= \beta \end{aligned}$$

The domain is the unit interval $(0, 1)$ that we first discretize by considering a partition: $0 = x_0 < x_1 < \cdots < x_N < x_{N+1} = 1$. See Fig. 2 for an example of uniform grid with 8 intervals. For simplicity, we assume the partition is uniform. Let

$$h = \frac{1}{N+1}, \quad x_j = jh$$

For each interior node, we have:

$$-u''(x_j) = f(x_j), \quad 1 \leq j \leq N$$

Then we replace the second derivative by a finite difference approximation:

$$u''(x_j) \approx \frac{u(x_j + h) - 2u(x_j) + u(x_j - h)}{h^2}$$

We next introduce the approximations U_j^h defined by:

$$\begin{aligned} \frac{-U_{j+1}^h + 2U_j^h - U_{j-1}^h}{h^2} &= f(x_j), \quad 1 \leq j \leq N \\ U_0^h &= \alpha \\ U_{N+1}^h &= \beta \end{aligned}$$

The finite difference solution is the vector $\boldsymbol{U}^h$ with components U_j^h. We may or may not include the components U_0^h and U_{N+1}^h. We claim that

$$U_j^h \approx u(x_j) \quad \forall j$$

Clearly for $j = 0$ and $j = N + 1$ we have exactly $U_j^h = u(x_j)$. We say the Dirichlet boundary conditions are enforced strongly (or exactly) in the finite difference method.

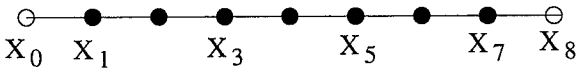

Figure 2: Example of uniform grid with 7 interior nodes. Here the grid size is $h = 1/8$.

Assume that the vector $\boldsymbol{U}^h$ has components $(U_1^h, U_2^h, \ldots, U_N^h)$. We obtain a linear system

$$\boldsymbol{A}^h \boldsymbol{U}^h = \boldsymbol{b}^h \tag{3}$$

where

$$\boldsymbol{A}^h = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & & & & 0 \\ -1 & 2 & -1 & 0 & & & 0 \\ 0 & -1 & 2 & -1 & 0 & & 0 \\ & & & \cdot & \cdot & \cdot & \\ & & & 0 & -1 & 2 & -1 \\ & & & & 0 & -1 & 2 \end{pmatrix} \quad \boldsymbol{b}^h = \begin{pmatrix} f(x_1) + \frac{\alpha}{h^2} \\ f(x_2) \\ f(x_3) \\ \cdot \\ f(x_{N-1}) \\ f(x_N) + \frac{\beta}{h^2} \end{pmatrix}$$

The matrix $\boldsymbol{A}^h$ is symmetric, tridiagonal. It is very easy to solve for the vector of unknowns $\boldsymbol{U}^h$.

Question: How accurate is the finite difference solution? We define the error $\boldsymbol{e}^h$, that is the vector with components:

$$e_j^h = u(x_j) - U_j^h, \quad 1 \leq j \leq N$$

We wish that

$$\lim_{h \to 0} \|\boldsymbol{e}^h\| = \lim_{N \to \infty} \|\boldsymbol{e}^h\| = 0$$

for a given norm of interest (for instance $\|\cdot\|_\infty$ or $\|\cdot\|_2$).

Def: *Local truncation error* at each interior node:

$$\tau_j^h = \frac{-u(x_{j+1}) + 2u(x_j) - u(x_{j-1})}{h^2} - f(x_j), \quad 1 \leq j \leq N$$

The local truncation error measures the error made by replacing the numerical solution by the exact solution in the finite difference method. The local truncation error only depends on the exact solution. We already know by Taylor series:

$$\tau_j^h = -u''(x_j) - \frac{h^2}{12} u^{(4)}(\xi_j) - f(x_j) = -\frac{h^2}{12} u^{(4)}(\xi_j)$$

since u is the exact solution. Therefore if $\boldsymbol{\tau}^h$ is the vector with components τ_j^h, we have:

$$\|\boldsymbol{\tau}^h\|_\infty \leq \frac{h^2}{12} \|u^{(4)}\|_\infty$$

This implies that

$$\lim_{h \to 0} \|\boldsymbol{\tau}^h\|_\infty = 0$$

We say that the finite difference method is *consistent*. To show consistency of any finite difference method, it suffices to show that $\|\boldsymbol{\tau}^h\|_\infty = \mathcal{O}(h^q)$ for some $q > 0$.

We now want to relate the local truncation error with the error $\boldsymbol{e}^h$. For this we introduce a vector $\boldsymbol{W}$ with components $u(x_j)$, where u is the exact solution. From the definition of the local truncation error, one obtain:

$$\boldsymbol{\tau}^h = \boldsymbol{A}^h \boldsymbol{W} - \boldsymbol{b}^h \tag{4}$$

Substracting (4) from (3) yields:

$$\boldsymbol{A}^h \boldsymbol{e}^h = \boldsymbol{\tau}^h$$

or equivalently if the matrix $\boldsymbol{A}^h$ is invertible:

$$\boldsymbol{e}^h = (\boldsymbol{A}^h)^{-1} \boldsymbol{\tau}^h$$

This implies:

$$\|\boldsymbol{e}^h\| \leq \|(\boldsymbol{A}^h)^{-1}\| \|\boldsymbol{\tau}^h\| \tag{5}$$

Def: Let $\boldsymbol{A}^h \boldsymbol{U}^h = \boldsymbol{b}^h$ be the linear system obtained by applying a finite difference method to a boundary value problem. The method is *stable* if $\boldsymbol{A}^h$ is invertible for all h sufficiently small and if there is a constant C independent of h such that

$$\|(\boldsymbol{A}^h)^{-1}\| \leq C$$

If the method is stable, (5) implies

$$\|\boldsymbol{e}^h\| \leq C \|\boldsymbol{\tau}^h\|$$

If in addition, the method is consistent, we have $\lim_{h \to 0} \|\boldsymbol{e}^h\| = 0$, which means that the method is convergent.

Therefore we have the fundamental result:

CONSISTENCY AND STABILITY IMPLIES CONVERGENCE.

This result is part of the Lax equivalence theorem: *For a consistent finite difference method for a well-posed linear initial value problem, the method is convergent if and only if it is stable.* In other words, if we have consistency, then convergence is equivalent to stability.

Clearly if the method is stable and $\|\boldsymbol{\tau}^h\|_\infty = \mathcal{O}(h^q)$ for some $q > 0$, then the method is convergent of order q.

Remark: proving consistency is easy and requires the use of Taylor series. Proving stability of the method can be challenging. We will give an outline of the proof for stability in $\|\cdot\|_2$.

We recall that for any matrix A that is symmetric, $\|A\|_2 = \rho(A)$, where $\rho(A)$ is the spectral radius. Namely $\rho(A)$ is the maximum $|\lambda|$ over all eigenvalues λ of A. Let $\sigma(A)$ denote the set of all eigenvalues of A. Since the matrix $(\boldsymbol{A}^h)^{-1}$ is symmetric, we have:

$$\|(\boldsymbol{A}^h)^{-1}\|_2 = \rho((\boldsymbol{A}^h)^{-1}) = \frac{1}{\min_{\lambda \in \sigma(\boldsymbol{A}^h)} |\lambda|}$$

Therefore we need to compute the eigenvalues of $\boldsymbol{A}^h$ and check that its minimum in absolute value is bounded away from zero by a constant independent of h.

Claim: the eigenvalues of $\boldsymbol{A}^h$ are

$$\lambda_k = \frac{2}{h^2}(1 - \cos(k\pi h)), \quad k = 1, \ldots, N$$

On Fig. 3, the plot of the function $1 - cos(x)$ is given for $0 \leq x \leq \pi$. From this graph,

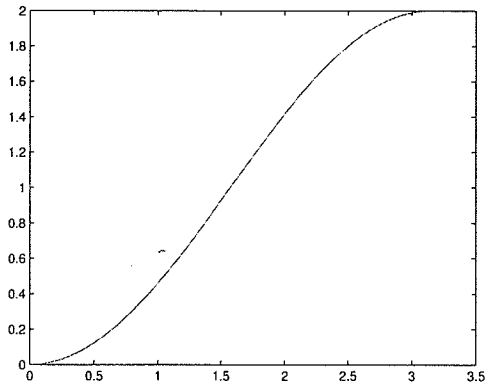

Figure 3: Plot of $1 - \cos(x)$

we see that the smallest $|\lambda_k|$ is obtained for $k = 1$.

$$|\lambda_1| = \frac{2}{h^2}(1 - \cos(\pi h))$$

As h tends to zero, the quantity πh also goes to zero. A Taylor expansion of $\cos(x)$ around 0 gives

$$\cos(x) = 1 - \frac{1}{2}x^2 + \frac{1}{4!}x^4 + \ldots$$

Thus we have

$$|\lambda_1| \approx \frac{2}{h^2}(\frac{(\pi h)^2}{2} + \mathcal{O}(h^4)) \approx \pi^2 + \mathcal{O}(h^2)$$

This implies that for h small enough

$$\|(\boldsymbol{A}^h)^{-1}\| \leq \frac{1}{\pi^2}$$

We conclude that the finite difference (3) is stable, and convergent of second order.

Remark: the convergence rate (or order) of the method can be verified numerically by computing the ratios

$$\frac{\log(\frac{\|e^h\|}{\|e^{h/2}\|})}{\log(2)}$$

on successively refined grids (i.e. grids obtained by dividing h by 2 successively).

Here is an example of errors and rates we obtain when we apply the method to the exact solution $u(x) = e^x$.

h	Error	Rates
2.0000e-01	6.948e-04	--
1.0000e-01	1.753e-04	1.98
5.0000e-02	4.41e-05	1.99
2.5000e-02	1.1e-5	1.99

Question: for which exact solutions is the error equal to zero?

Remark: One can also show that the finite difference method is stable with respect to $\|\cdot\|_\infty$.

2. Neumann problem

We want to solve the second order problem:

$$\begin{aligned} -u''(x) &= f(x) \quad 0 < x < 1 \\ u'(0) &= \sigma \\ u(1) &= \beta \end{aligned}$$

The boundary condition $u'(0) = \sigma$ is called a Neumann boundary condition. As in Section 1, we start by defining a grid $x_j = jh$ for some $h > 0$ (say $h = 1/(N+1)$). There are several ways to modify the finite difference method defined above to handle the Neumann boundary condition.

2.1. Using a first order one-sided finite difference approximation

We replace $u'(0) = \sigma$ by:

$$\frac{U_1^h - U_0^h}{h} = \sigma$$

We consider the augmented vector of unknowns $\tilde{\boldsymbol{U}}^h = (U_0^h, U_1^h, \ldots, U_N^h, U_{N+1}^h)$. If we use the same second order centered finite difference approximation of $u''(x_j)$ as in Section 1, we obtain a new linear system:

$$\tilde{\boldsymbol{A}}^h \tilde{\boldsymbol{U}}^h = \tilde{\boldsymbol{b}}^h$$

with

$$\tilde{\boldsymbol{A}}^h = \frac{1}{h^2}\begin{pmatrix} -h & h & 0 & & & & 0 \\ -1 & 2 & -1 & 0 & & & 0 \\ 0 & -1 & 2 & -1 & 0 & & 0 \\ & & & \cdot & \cdot & \cdot & \\ & & & 0 & -1 & 2 & -1 \\ & & & & 0 & 0 & h^2 \end{pmatrix} \quad \tilde{\boldsymbol{b}}^h = \begin{pmatrix} \sigma \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \cdot \\ f(x_{N-1}) \\ f(x_N) \\ \beta \end{pmatrix}$$

The matrix $\tilde{\boldsymbol{A}}^h$ is tridiagonal of size $(N+2) \times (N+2)$. The resulting accuracy of this method is first order only. This is expected as we have used a first order approximation of the Neumann boundary condition.

2.2. Using a second order centered finite difference approximation

We replace $u'(0) = \sigma$ by:

$$\frac{U_1^h - U_{-1}^h}{2h} = \sigma$$

where U_{-1}^h is an intermediate value, that would approximate $u(-h)$. We need another equation to eliminate this additional unknown. We write the finite difference approximation of $-u''(x_0) = f(x_0)$:

$$-\frac{U_1^h - 2U_0^h + U_{-1}^h}{h^2} = f(x_0)$$

Combining the two equations above yields:

$$\frac{-U_0^h + U_1^h}{h} = \sigma - \frac{h}{2} f(x_0) \tag{6}$$

Another way to obtain (6) is to use a Taylor series at x_0:

$$u(x_1) = u(x_0) + hu'(x_0) + \frac{h^2}{2} u''(x_0) + \mathcal{O}(h^3)$$

If we assume that $u''(x_0) = -f(x_0)$, this yields

$$U_1^h = U_0^h + h\sigma - \frac{h^2}{2} f(x_0)$$

The resulting matrix $\tilde{\boldsymbol{A}}^h$ is the same as in Section 2.1, but the right-hand side changes:

$$\tilde{\boldsymbol{b}}^h = \begin{pmatrix} \sigma - \frac{h}{2} f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ . \\ f(x_{N-1}) \\ f(x_N) \\ \beta \end{pmatrix}$$

This simple change is enough to obtain a second order method.

2.3. Using a second order one-sided finite difference approximation

This technique is preferred over the previous one as it is the easiest to generalize for higher order finite difference methods. One replace $u'(0) = \sigma$ by

$$\frac{-\frac{3}{2}U_0^h + 2U_1^h - \frac{1}{2}U_2^h}{h} = \sigma$$

The linear system becomes:

$$\tilde{\boldsymbol{A}}^h = \frac{1}{h^2} \begin{pmatrix} -\frac{3}{2}h & 2h & -\frac{1}{2}h & & & & 0 \\ -1 & 2 & -1 & 0 & & & 0 \\ 0 & -1 & 2 & -1 & 0 & & 0 \\ & & & . & . & . & \\ & & & 0 & -1 & 2 & -1 \\ & & & & 0 & 0 & h^2 \end{pmatrix} \quad \tilde{\boldsymbol{b}}^h = \begin{pmatrix} \sigma \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ . \\ f(x_{N-1}) \\ f(x_N) \\ \beta \end{pmatrix}$$

The accuracy of this method is $\mathcal{O}(h^2)$.

2.4. Pure Neumann problem

Assume we want to solve

$$\begin{aligned} -u''(x) &= f(x) \quad 0 < x < 1 \\ u'(0) &= \sigma_0 \\ u'(1) &= \sigma_1 \end{aligned}$$

We first integrate the PDE:

$$-u'(1) + u'(0) = \int_0^1 u''(x)dx = \int_0^1 f(x)dx$$

or equivalently

$$-\sigma_1 + \sigma_0 = \int_0^1 f(x)dx$$

Thus the pure Neumann problem requires the data to satisfy a *compatibility* condition. Assuming this is satisfied, it is easy to see that if u is an exact solution, then $u + c$ is another exact solution, for any constant c. The pure Neumann problem does not have a unique solution, we say that it is not well-posed. This is also true for the numerical solution. The matrix obtained in Section 2.1 (or 2.2) is in this case:

$$\tilde{\boldsymbol{A}}^h = \frac{1}{h^2}\begin{pmatrix} -h & h & 0 & & & & 0 \\ -1 & 2 & -1 & 0 & & & 0 \\ 0 & -1 & - & -1 & 0 & & 0 \\ & & & \cdot & \cdot & \cdot & \\ & & & 0 & -1 & 2 & -1 \\ & & & & 0 & -h & h \end{pmatrix}$$

Clearly if $\boldsymbol{v}$ is the vector with components all equal to 1, we see that $\tilde{\boldsymbol{A}}^h\boldsymbol{v} = 0$. Thus the matrix $\tilde{\boldsymbol{A}}^h$ is not invertible.

3. General boundary value problem

We consider a general elliptic problem with function coefficients $k(x)$ and $d(x)$. with Dirichlet boundary conditions:

$$\begin{gathered} -(k(x)u'(x))' + d(x)u(x) = f(x) \quad a < x < b \\ u(a) = \alpha \\ u(b) = \beta \end{gathered}$$

A first approach would be to differentiate the term $(ku')'$ as

$$(ku')' = k'u' + ku''$$

and use finite difference approximations of u' and u''. However the physical phenomena described by this elliptic problem may not allow for u' to be continuous. Rather the flux ku' is continuous. The physics tell us that we should consider the product ku' as a whole. We now describe the derivation of a FD method that is prefered.

Let $x_0 < x_1 < \cdots < x_{N+1}$ be a uniform grid of the interval (a, b) of size h. Denote by $x_{i+1/2}$ the midpoint of the interval (x_i, x_{i+1}). A centered finite difference approximation of the flux function $g = ku'$ at the node $x_{i+1/2}$ is:

$$g(x_{i+1/2}) = k(x_{i+1/2})u'(x_{i+1/2}) \approx k(x_{i+1/2})\frac{u(x_{i+1}) - u(x_i)}{h} \tag{7}$$

A centered finite difference approximation of $g'(x_i)$ is:

$$g'(x_i) \approx \frac{g(x_{i+1/2}) - u(x_{i-1/2})}{h}$$

We substitute (7) in the equation above for the points $x_{i+1/2}$ and $x_{i-1/2}$:

$$g'(x_i) \approx \frac{1}{h^2}\big(k(x_{i+1/2})(u(x_{i+1}) - u(x_i)) - k(x_{i-1/2})(u(x_i) - u(x_{i-1}))\big)$$

Thus the FD method becomes: for all interior nodes $1 \leq i \leq N$:

$$-\frac{1}{h^2}\big(k(x_{i+1/2})(U_{i+1} - U_i) - k(x_{i-1/2})(U_i) - U_{i-1})\big) + d(x_i)U_i = f(x_i)$$

This yields the system

$$(\boldsymbol{B} + \boldsymbol{M})\boldsymbol{U} = \boldsymbol{b}$$

where $\boldsymbol{M}$ is a diagonal matrix with entries $d(x_i)$ and $\boldsymbol{B}$ is a tridiagonal matrix with entries:

$$B_{ii} = \frac{k(x_{i-1/2}) + k(x_{i+1/2})}{h^2}, \quad B_{i,i+1} = -\frac{k(x_{i+1/2})}{h^2}, \quad B_{i,i-1} = -\frac{k(x_{i-1/2})}{h^2}$$

The matrix $\boldsymbol{A} = \boldsymbol{B} + \boldsymbol{M}$ is symmetric. This method is convergent with second order.

III. ELLIPTIC PROBLEMS IN 2D

1. Definition

Consider the problem on the unit square $\Omega = (0,1)^2$ with boundary denoted by $\partial\Omega$.

$$-\Delta u = f, \quad \text{in} \quad (0,1)^2 = \Omega$$
$$u = g, \quad \text{on} \quad \partial\Omega$$

We recall that for a function $u = u(x,y)$:

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

The finite difference grid is a tensor-product grid with size $h_x > 0$ in the $x-$direction and $h_y > 0$ in the $y-$direction.

$$x_i = ih_x, \quad y_j = jh_y$$

We replace $\Delta u(x_i, y_j)$ by a five-point stencil that is second order accurate

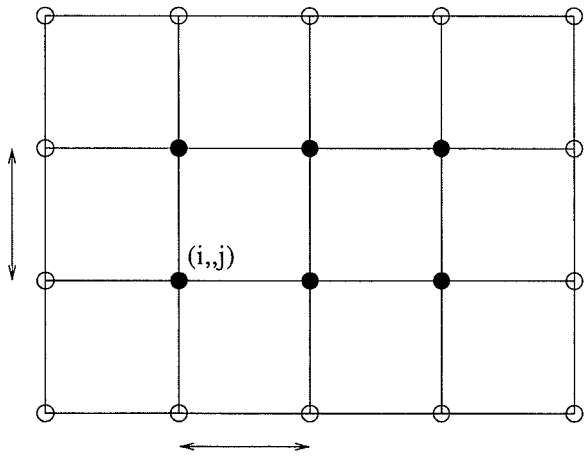

Figure 4: Two-dimensional grid.

$$\frac{-1}{h_x^2}\big(u(x_{i-1},y_j) - 2u(x_i,y_j) + u(x_{i+1},y_j)\big) + \frac{-1}{h_y^2}\big(u(x_i,y_{j-1}) - 2u(x_i,y_j) + u(x_i,y_{j+1})\big)$$

Denote by U_{ij} the finite difference solution, that approximates $u(x_i,y_j)$. Then the FD scheme is:

$$\frac{-1}{h_x^2}\big(U_{i-1,j} - 2U_{ij} + U_{i+1,j}\big) + \frac{-1}{h_y^2}\big(U_{i,j-1} - 2U_{ij} + U_{i,j+1}\big) = f(x_i,y_j), \quad \forall i,j$$

For simplicity, assume that $h_x = h_y = h = \frac{1}{N+1}$ for some integer $N \geq 1$. The scheme becomes:

$$\frac{1}{h^2}\big(- U_{i-1,j} - U_{i+1,j} + 4U_{ij} - U_{i,j-1} - U_{i,j+1}\big) = f(x_i, y_j), \quad \forall 1 \leq i,j \leq N$$

The boundary conditions become:

$$U_{0,j} = g(0, y_j), \quad U_{N+1,j} = g(1, y_j), \quad U_{i,0} = g(x_i, 0), \quad U_{i,N+1} = g(x_i, 1)$$

The vector of unknowns contains the FD solution at the interior nodes only, since the values on the boundary of the domain are known. We obtain a matrix of size $N^2 \times N^2$. The matrix is sparse: at most five entries per row are nonzero. The structure of the matrix strongly depends on the ordering of the unknowns.

We will consider two types of ordering: a natural rowwise ordering and a checkerboard ordering.

1.1. Natural rowwise ordering

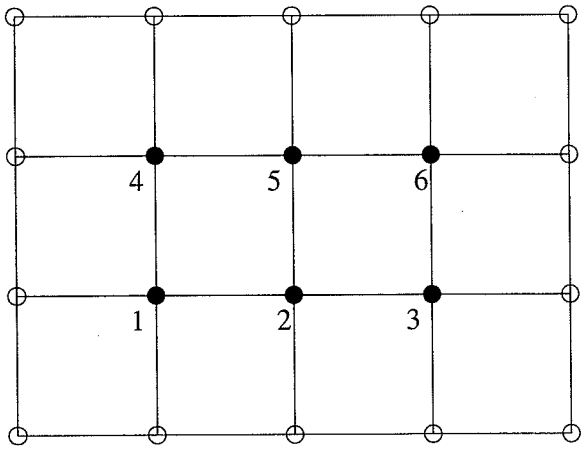

Figure 5: Rowwise ordering.

We start with the bottom row and number the unknowns from left to right, then move to the second row, and repeat. This gives the vector

$$\boldsymbol{U}^h = (U_{11}, U_{21}, \ldots, U_{N1}, U_{12}, U_{22}, \ldots, U_{N2}, \ldots, U_{1N}, U_{2N}, \ldots, U_{NN})$$

The resulting matrix is a block tridiagonal matrix:

$$\boldsymbol{A}^h = \frac{1}{h^2} \begin{pmatrix} \boldsymbol{T} & -\boldsymbol{I} & 0 & & & & 0 \\ -\boldsymbol{I} & \boldsymbol{T} & -\boldsymbol{I} & 0 & & & 0 \\ 0 & -\boldsymbol{I} & \boldsymbol{T} & -\boldsymbol{I} & 0 & & 0 \\ & & & \cdot & \cdot & \cdot & \\ & & & 0 & -\boldsymbol{I} & \boldsymbol{T} & -\boldsymbol{I} \\ & & & & 0 & -\boldsymbol{I} & \boldsymbol{T} \end{pmatrix}$$

where each block $\boldsymbol{T}$ or $\boldsymbol{I}$ is a matrix of size $N \times N$. The matrix $\boldsymbol{I}$ is simply the identity matrix, and the matrix $\boldsymbol{T}$ is a tridiagonal matrix:

$$\boldsymbol{T} = \begin{pmatrix} 4 & -1 & 0 & & & & 0 \\ -1 & 4 & -1 & 0 & & & 0 \\ 0 & -1 & 4 & -1 & 0 & & 0 \\ & & & \cdot & \cdot & \cdot & \\ & & & 0 & -1 & 4 & -1 \\ & & & & 0 & -1 & 4 \end{pmatrix}$$

The right-hand side vector $\boldsymbol{b}$ is the sum of two vectors: $\boldsymbol{b} = \boldsymbol{b}_1 + \boldsymbol{b}_2$, where

$$\boldsymbol{b}_1 = (f(x_1, y_1), f(x_2, y_1), \ldots, f(x_N, y_1), f(x_1, y_2), f(x_2, y_2), \ldots, f(x_N, y_2), \\ \ldots, f(x_1, y_N), f(x_2, y_N), \ldots, f(x_N, y_N))$$

The vector $\boldsymbol{b}_2$ has mostly zero entries, the only non-zero entries correspond to the first N entries, the last N entries, and another $2(N-2)$ nonzero entries.

1.2. Checkerboard ordering

This type of ordering is also called black-red ordering. We assign each interior node a color (red or black) so that colors are alternated. For each red node, all neighbors (in the five-point stencil) are black nodes. Vice-versa for each black node, all neighbors are red ones. The unknowns are ordered so that all unknowns corresponding to nodes of the same color are grouped together. The resulting matrix takes the form

$$\boldsymbol{A}^h = \begin{pmatrix} \boldsymbol{D} & \boldsymbol{H} \\ \boldsymbol{H}^T & \boldsymbol{D} \end{pmatrix}$$

The matrix $\boldsymbol{D} = (4/h^2)\boldsymbol{I}$ is of size $(N^2/2) \times (N^2/2)$. The matrix $\boldsymbol{H}$ has at most 4 nonzero entries per row.

2. Examples of stencils

The local truncation error for the five-point stencil given above is:

$$\tau_{ij} = \frac{1}{h^2}\Big(- u(x_{i-1}, y_j) + 4u(x_i, y_j) - u(x_{i+1}, y_j) - u(x_i, y_{j-1}) - u(x_i, y_{j+1})\Big) - f(x_i, y_j)$$

Using a Taylor expansion in both x and y directions, we obtain

$$\tau_{ij} = -\frac{h^2}{12}\left(\frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4}\right) + \mathcal{O}(h^4)$$

Another example of five-point stencil is:

$$\frac{-1}{h^2}\left(\frac{1}{2}U_{i-1,j-1} + \frac{1}{2}U_{i+1,j+1} - 2U_{ij} + \frac{1}{2}U_{i+1,j-1} + \frac{1}{2}U_{i-1,j+1}\right)$$

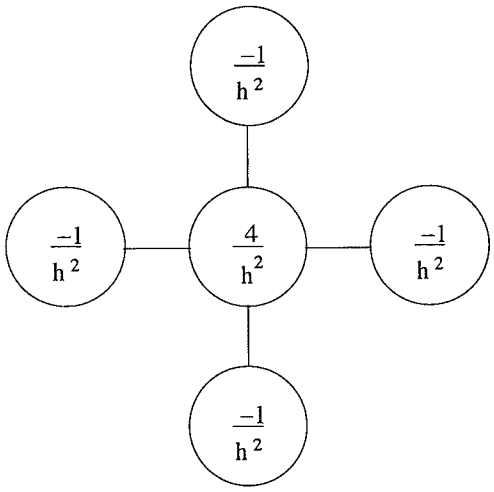

Figure 6: Standard five-point stencil.

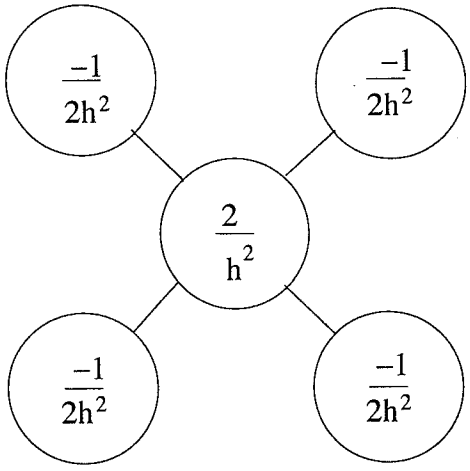

Figure 7: Another five-point stencil.

which has a local truncation error $\mathcal{O}(h^2)$.

The following is a nine-point stencil

$$\frac{-1}{3h^2}(\frac{1}{2}U_{i-1,j+1}+2U_{i,j+1}+\frac{1}{2}U_{i+1,j+1}+2U_{i-1,j}-10U_{ij}+2U_{i+1,j}+\frac{1}{2}U_{i-1,j-1}+2U_{i,j-1}+\frac{1}{2}U_{i+1,j-1})$$

which has a local truncation error of the form

$$\tau = -\frac{h^2}{12}(\frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4}) + \mathcal{O}(h^4)$$

3. Analysis of five-point stencil

A first approach is to follow the argument given in 1D. Let $\boldsymbol{U}^h$ be the vector of unknowns and let $\boldsymbol{W}$ be the vector with components $u(x_i, y_j)$ in the same order than the

U_{ij}'s. We have:

$$\boldsymbol{A}^h(\boldsymbol{W} - \boldsymbol{U}^h) = \boldsymbol{\tau}$$

where $\boldsymbol{\tau}$ is the vector with components τ_{ij} in the same order than the U_{ij}'s. So the error vector $\boldsymbol{e}^h$ is obtained by solving:

$$\boldsymbol{A}^h\boldsymbol{e}^h = \boldsymbol{\tau}$$

From this we obtain, if $\boldsymbol{A}^h$ is invertible:

$$\|\boldsymbol{e}^h\| \leq \|(\boldsymbol{A}^h)^{-1}\|\|\boldsymbol{\tau}\|$$

If we have stability (i.e. $\|(\boldsymbol{A}^h)^{-1}\|$ bounded by a constant independent of h), then the FD method converges with the same order as the local truncation error. To prove stability, we need to look at the specific norm, and this can be tedious.

A more powerful proof of convergence of the FD method is now presented. The proof uses ideas of maximum principle for elliptic PDEs. We now recall some maximum principles results from PDE theory.

Theorem: Define the differential operator $Lu = -\Delta u$ for $u \in \mathcal{C}^2(\Omega) \cup \mathcal{C}(\bar{\Omega})$. The set $\bar{\Omega}$ is the closure of Ω: $\bar{\Omega} = \Omega \cup \partial\Omega$.

- If $Lu \leq 0$ in Ω, then
$$\max_{(x,y)\in\bar{\Omega}} u(x,y) = \max_{(x,y)\in\partial\Omega} u(x,y)$$
- If $Lu \geq 0$ in Ω, then
$$\min_{(x,y)\in\bar{\Omega}} u(x,y) = \min_{(x,y)\in\partial\Omega} u(x,y)$$

Let $\mathcal{S}_h$ denote the set of grid nodes: it contains both interior nodes and nodes on the boundary. We consider the five-point stencil and we denote by N, S, W, E the neighbors of any given point P. Define the function B that acts on the nodes by:

$$\forall P \in \mathcal{S}_h, \quad B(P,Q) = \begin{cases} \frac{4}{h^2} & \text{if } Q = P \\ \frac{-1}{h^2} & \text{if } Q \in \{N,S,W,E\} \\ 0 & \text{otherwise} \end{cases}$$

The finite difference method can be written as:

$$\sum_{Q\in\mathcal{S}_h} B(P,Q)\hat{U}(Q) = f(P), \quad \forall P \in \mathcal{S}_h \setminus \partial\Omega$$

where $\hat{U}$ is a discrete function defined only at the nodes in $\mathcal{S}_h$. The boundary condition becomes:

$$\hat{U}(P) = g(P), \quad \forall P \in \mathcal{S}_h \cap \partial\Omega$$

Let us define the operator L_h that acts on the set of such discrete functions:

$$L_h\hat{V}(P) = \sum_{Q \in \mathcal{S}_h} B(P,Q)\hat{V}(Q)$$

If P has for coordinates (x_i, y_j), we can define the local truncation error and the error as:

$$\tau(P) = \tau_{ij}, \quad e(P) = u(P) - \hat{U}(P) = u(x_i, y_j) - U_{ij}$$

Theorem Assume that Ω is contained in the strip $\{(x,y) : \quad a < x < b\}$ for some $a, b \in \mathbb{R}$. Then, for any point $P \in \mathcal{S}_h$ we have

$$|e(P)| \leq \frac{1}{2}\max(a^2, b^2) \max_{Q \in \mathcal{S}_h \setminus \partial\Omega} |\tau(Q)|$$

Proof Define the function Φ for all nodes

$$\Phi(P) = \frac{1}{2}(\max(a^2, b^2) - x_P^2) \max_{Q \in \mathcal{S}_h \setminus \partial\Omega} |\tau(Q)|$$

where x_P is the $x-$ coordinate of P. Because of the assumption on Ω, we see that:

$$\Phi(P) \geq 0, \quad \forall P \in \mathcal{S}_h$$

We now state two claims that the proof of the theorem uses.

Lemma 1

$$\forall P \in \mathcal{S}_h \setminus \partial\Omega, \quad L_h\Phi(P) \geq \max_{Q \in \mathcal{S}_h \setminus \partial\Omega} |\tau(Q)|$$

Lemma 2 Let $\hat{V}$ be a discrete function defined on the nodes such that

$$\forall P \in \mathcal{S}_h \setminus \partial\Omega, \quad L_h(\hat{V})(P) \leq 0$$
$$\forall P \in \mathcal{S}_h \cap \partial\Omega, \quad \hat{V}(P) \leq 0$$

Then we have

$$\max_{P \in \mathcal{S}_h} \hat{V}(P) \leq 0$$

We will prove these claims at the end. The error function satisfies:

$$L_h e(P) = \sum_Q B(P,Q)e(Q) = \sum_Q B(P,Q)u(Q) - f(P) = \tau(P)$$

So

$$L_h e(P) = \tau(P) \leq \max_{Q \in \mathcal{S}_h \setminus \partial\Omega} |\tau(Q)| \leq L_h\Phi(P)$$

by the first lemma. Therefore, the function $\hat{W} = e - \Phi$ is a discrete function that satisfies:

$$\forall P \in \mathcal{S}_h \setminus \partial\Omega, \quad L_h(\hat{W})(P) \leq 0$$

Furthermore on the boundary we have

$$e(P) = 0 \leq \Phi(P)$$

or equivalently

$$\hat{W}(P) \leq 0$$

So from Lemma 2, we conclude that

$$\max_{P \in \mathcal{S}_h} \hat{W}(P) \leq 0$$

which implies that $e(P) \leq \Phi(P)$ for all P. Similarly we apply the same argument above to the function $-e$:

$$L_h(-e)(P) = -L_h e(P) = -\tau(P) \leq \max_{Q \in \mathcal{S}_h \setminus \partial\Omega} |\tau(Q)| \leq L_h \Phi(P)$$

Then apply Lemma 2 to the function $\hat{V} = -e - \Phi$:

$$-e(P) \leq \Phi(P) \quad \forall P$$

Since Φ is non-negative, then this means that

$$|e(P)| \leq \Phi(P) \quad \forall P$$

which implies the theorem.

Now it remains to prove the two lemmas. For Lemma 1, we want to show that for all interior node P

$$\sum_{Q \in \mathcal{S}_h} B(P,Q)\Phi(Q) \geq \max_{Q \in \mathcal{S}_h \setminus \partial\Omega} |\tau(Q)|$$

which (by plugging in the definition of Φ is equivalent to

$$\sum_{Q \in \mathcal{S}_h} B(P,Q)(\max(a^2, b^2) - x_Q^2) \geq 2$$

which is again equivalent to

$$B(P,P)(\max(a^2, b^2) - x_P^2) + \sum_{Q \in \mathcal{S}_h, Q \neq P} B(P,Q)(\max(a^2, b^2) - x_Q^2) \geq 2$$

Since the five-point stencil satisfies the property:

$$B(P,P) = -\sum_{Q \in \mathcal{S}_h, Q \neq P} B(P,Q)$$

we can rewrite the inequality as:

$$\sum_{Q\in\mathcal{S}_h, Q\neq P} B(P,Q)(x_P^2 - x_Q^2) \geq 2$$

The left-hand side of the inequality reduces to:

$$B(P,W)(x_P^2 - x_W^2) + B(P,E)(x_P^2 - x_E^2) = \frac{1}{h^2}(x_P^2 - x_W^2 + x_P^2 - x_E^2) = 2$$

So in fact, we have equality. This proves the result.

Now for Lemma 2, we denote by R the node for which

$$\hat{V}(R) = \max_{P\in\mathcal{S}_h} \hat{V}(P)$$

If R belongs to the boundary, we are done. Let us assume that R is an interior node, and let $P_1, P_2, \ldots, P_n$ be a finite number of points that connect R to a point P_n on the boundary. We now show that $\hat{V}(R) = \hat{V}(P_1) = \hat{V}(P_2) = \cdots = \hat{V}(P_n)$ by looking at successive pairs. By contradiction, assume that $\hat{V}(P_1) < \hat{V}(R)$. Since $L_h\hat{V}(R) \leq 0$ we have:

$$B(R,R)\hat{V}(R) + \sum_{Q\neq R} B(P,Q)\hat{V}(Q) \leq 0$$

Using again the property

$$0 \leq B(P,P) = -\sum_{Q\in\mathcal{S}_h, Q\neq P} B(P,Q)$$

we have

$$\hat{V}(R) \leq -\sum_{Q\neq R} \frac{B(R,Q)}{B(R,R)}\hat{V}(Q) < -\sum_{Q\neq R} \frac{B(R,Q)}{B(R,R)}\hat{V}(R)$$

since $\hat{V}(P_1) < \hat{V}(R)$ and $\hat{V}(R)$ is the maximum value. But the ratio $-\sum_{Q\neq R}\frac{B(R,Q)}{B(R,R)}$ is equal to 1, so we have

$$\hat{V}(R) < \hat{V}(R)$$

which is impossible. We consider next the following pair and show $\hat{V}(P_1) = \hat{V}(P_2)$.

4. Some generalizations

4.1. Non-uniform grids

One can generalize the finite difference method to non-uniform grids. Fig. 8 defines the lengths of the grid space in each direction around the grid node (i,j). In that case, we have

$$\Delta u(x_i,y_j) \approx \frac{2}{h_W h_E(h_W+h_E)}(h_W u(x_{i+1},y_j) + h_E u(x_{i-1},y_j))$$
$$+\frac{2}{h_S h_N(h_S+h_N)}(h_S u(x_i,y_{j+1}) + h_N u(x_i,y_{j-1})) - 2(\frac{1}{h_W h_E} + \frac{1}{h_S h_N})u(x_i,y_j)$$

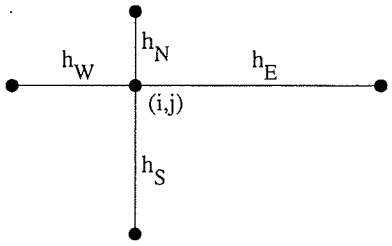

Figure 8: Finite difference approximations on non-uniform grids.

4.2. Non-unity coefficients

The PDE is:

$$-\nabla \cdot K\nabla u = f$$

where K depends on the spatial location. As in 1D, we define fluxes

$$g_1 = K\frac{\partial u}{\partial x}, \quad g_2 = K\frac{\partial u}{\partial y}$$

The PDE becomes:

$$-\frac{\partial g_1}{\partial x} - \frac{\partial g_2}{\partial y} = f$$

Intermediate grid points are defined: they are located in the midpoints on the grid segments.

$$g_1(x_i, y_j) \approx \frac{g_1(x_{i+1/2}, y_j) - g_1(x_{i-1/2}, y_j)}{h}$$

and

$$\frac{\partial g_1}{\partial x}(x_{i+1/2}, y_j) \approx K(x_{i+1/2}, y_j)\frac{u(x_{i+1}, y_j) - u(x_i, y_j)}{h}$$

The same applies for the other terms.

IV. FINITE ELEMENT METHOD FOR ONE-DIMENSIONAL BOUNDARY VALUE PROBLEMS

1. Definition

We consider the two-point boundary value problem:

$$-\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) + c(x)u(x) = f(x), \quad 0 < x < 1$$
$$u(0) = u(1) = 0$$

where k and c are known functions satisfying

$$0 < k_0 \leq k(x) < k_1, \quad 0 \leq c(x)$$

Examples of such PDEs arise from many applications. For instance, in solid mechanics, the solution u is the displacement of an elastic bar fixed at both ends and subject to a tangential load f. In fluid mechanics, u is the fluid pressure.

The first step (step 0) is to discretize the domain. We consider the partition

$$0 = x_0 < x_1 < \cdots < x_N < x_{N+1} = 1$$

and we define

$$h_i = x_{i+1} - x_i, \quad h = \max_{0 \leq i \leq N} h_i$$

Let V_h be the space of continuous piecewise polynomials of degree p defined on this partition. In other words, if $v \in V_h$, then $v \in \mathcal{C}(0,1)$ and $v|_{(x_i,x_{i+1})}$ is a polynomial of degree p. To obtain a finite element approximation of u, we apply the following steps.

Step 1: multiply PDE by $v \in V_h$ and integrate over domain

$$\int_0^1 \Big(-\frac{d}{dx}\left(k(x)\frac{du}{dx}(x)\right) v(x) + c(x)u(x)v(x)\Big) = \int_0^1 f(x)v(x)$$

Step 2: integrate by parts the first term

$$\int_0^1 k(x)\frac{du}{dx}(x)\frac{dv}{dx}(x) - k(1)\frac{du}{dx}(1)v(1) + k(0)\frac{du}{dx}(0)v(0) + \int_0^1 c(x)u(x)v(x) = \int_0^1 f(x)v(x)$$

Step 3: Assume that functions in V_h also satisfy the Dirichlet boundary conditions: $v(0) = v(1) = 0$. The boundary terms drop.

$$\int_0^1 k(x)\frac{du}{dx}(x)\frac{dv}{dx}(x) + \int_0^1 c(x)u(x)v(x) = \int_0^1 f(x)v(x)$$

or equivalently

$$\int_0^1 (ku'v' + cuv) = \int_0^1 fv$$

Let $a(\cdot,\cdot)$ be a bilinear form and $\ell(\cdot)$ a linear form defined by

$$a(u,v) = \int_0^1 (ku'v' + cuv), \quad \ell(v) = \int_0^1 fv$$

The finite element space is

$$V_h = \{v \in \mathcal{C}(0,1) : v|_{(x_i,x_{i+1})} \in \mathbb{P}_p, \quad v(0) = v(1) = 0\}$$

and the finite element method is defined as: find $u_h \in V_h$ such that

$$\forall v \in V_h, \quad a(u_h, v) = \ell(v)$$

This type of problem is called a *variational formulation*; the function v is called a *test function* and the function u_h a *trial function.*

By varying the polynomial degree, we obtain several finite element solutions of different orders.

2 Finite element solutions of first order

2.1 Basis functions

Let $p = 1$ (piecewise linears). V_h is a finite dimensional space of dimension equal to N. We define standard basis functions for V_h.

$$1 \le i \le N, \quad \Phi_i(x) = \begin{cases} 0, & 0 \le x \le x_{i-1} \\ \frac{x - x_{i-1}}{h_{i-1}} & x_{i-1} \le x \le x_i \\ \frac{x_{i+1} - x}{h_i} & x_i \le x \le x_{i+1} \\ 0 & x_{i+1} \le x \le 1 \end{cases}$$

The function Φ_i is called a *hat* function (or chapeau), One can show that $\Phi_1, \Phi_2, \ldots, \Phi_N$ form a basis for V_h. We also note that

$$\Phi_i(x_j) = \delta_{ij}$$

Because of this property we say the Φ_i's form a *nodal* basis.

2.2 Linear system

We expand the solution u_h in the nodal basis:

$$u_h = \sum_{j=1}^{N} \alpha_j \Phi_j$$

with $\alpha_j \in \mathbb{R}$. The coefficients α_j's are the unknowns. They satisfy:

$$\forall v \in V_h, \quad \sum_{j=1}^{N} \alpha_j a(\Phi_j, v) = \ell(v)$$

which is equivalent to

$$1 \leq i \leq N, \quad \sum_{j=1}^{N} \alpha_j a(\Phi_j, \Phi_i) = \ell(\Phi_i)$$

Define the matrix $\boldsymbol{A} = (A_{ij})_{ij}$ with $A_{ij} = a(\Phi_j, \Phi_i)$ and the vectors $\boldsymbol{\alpha} = (\alpha_j)_j$, $\boldsymbol{b} = (b_j)_j$ with $b_j = \ell(\Phi_j)$. The matrix $\boldsymbol{A}$ is sometimes called the *stiffness* matrix and the right-hand side vector is called *the load* vector. The linear system is:

$$\boldsymbol{A\alpha} = \boldsymbol{b}$$

Let us now write the entries of the matrix in the case where $k = 1, c = 0$. Because the nodal basis functions have compact support, each row of $\boldsymbol{A}$ has at most three nonzero entries:

$$A_{ij} = 0, \quad \text{if } j \notin \{i-1, i, i+1\}$$

After some computations, we have

$$A_{ii} = \frac{1}{h_{i-1}} + \frac{1}{h_i}, \quad A_{i,i+1} = -\frac{1}{h_i}, \quad A_{i,i-1} = -\frac{1}{h_{i-1}}$$

If in addition, the mesh is uniform and $h = h_i$ for all i, we can write simply:

$$\boldsymbol{A} = \frac{1}{h} \begin{pmatrix} 2 & -1 & 0 & & & & 0 \\ -1 & 2 & -1 & 0 & & & 0 \\ 0 & -1 & 2 & -1 & 0 & & 0 \\ & & & \cdot & \cdot & \cdot & \\ & & & 0 & -1 & 2 & -1 \\ & & & & 0 & -1 & 2 \end{pmatrix}$$

To compute the entries of $\boldsymbol{b}$, we need to use a quadrature rule since we may not be able to compute the integral for any f. We choose to use the trapezoidal rule on each interval (x_i, x_{i+1}). This rule is exact for polynomials of degree one.

$$\begin{aligned} b_i = \int_0^1 f\Phi_i &= \int_{x_{i-1}}^{x_i} f\Phi_i + \int_{x_i}^{x_{i+1}} f\Phi_i \\ \approx \frac{h_{i-1}}{2}(f(x_i)\Phi_i(x_i) + f(x_{i-1})\Phi_i(x_{i-1})) &+ \frac{h_i}{2}(f(x_i)\Phi_i(x_i) + f(x_{i+1})\Phi_i(x_{i+1})) \\ &= \frac{1}{2} f(x_i)(h_{i-1} + h_i) \end{aligned}$$

3 Finite element solutions of second order

Let $p = 2$ (piecewise quadratics). The space V_h is a finite dimensional space of dimension equal to $2N + 1$. The usual basis functions for V_h are of two types. First we define