MA500-1: Numerical Differential Equations

Niall Madden

November 5, 2014

Contents

1 MA500-1 Introductory Lecture

This class began with a welcome greetings from various staff at the School of Mathematics, Statistics and Applied Mathematics, and a discussion of the Masters programme.

After the other staff had left, I gave a brief introduction to this module: Numerical Differential Equations.

2 Lecture 2: Approximation of derivatives

This class followed Chapter 1 of LeVeque pretty closely. We introduced the idea of approximating the derivative of a function by computing (finite) differences of values of the function at certain points. We derived the forward and backward difference schemes to estimate first derivatives. Then we went on to discuss the central difference scheme, and even a four-point scheme.

3 Lecture 3: Taylor's series

In this class we concentrated on how we could use Taylor's series to but construct finite difference formulae, and derive truncation error estimates.

4 Lecture 4: Introduction to finite difference methods

The use of finite difference formulae to estimate derivatives is not actually that interesting: for a given function there are far more reliable computer methods, such as so-called *automatic differentiation*. Our interest in approximating derivatives is actually based on the goal of approximating differential equations by difference equations, which can then be solved to give approximate solutions to DEs.

In this class, we followed the early sections of Chapter 2 of LeVeque. We began with the heat equation:

$$u_t(x,t) = \big(\kappa(x)u_x(x,t)\big)_x + \phi(x,t),$$

with suitable boundary and initial conditions. Much simplification followed until we reached the stead problem:

$$u''(x) = f(x) \quad \text{on } (0,1), \tag{4.1a}$$

with boundary conditions

$$u(0) = \alpha, \quad u(1) = \beta. \tag{4.1b}$$

The class culminated in proposing a simple finite difference scheme for this problem.

5 Lecture 5: Towards error analysis

We restated the finite difference method for solving (4.1), and wrote down the finite difference scheme:

$$\begin{gathered} U_0 = \alpha \\ \frac{1}{h^2}\big(U_{k-1} - 2U_k + U_{k+1}\big) = f(x_k) \quad \text{for } k = 1, \dots, N-1 \\ U_N + \beta. \end{gathered}$$

We then discussed the idea of *error*, and looked at a Matlab implementation. Using the Matlab code, we convinced ourselves that

(i) The error is (usually) proportional to h^2;

(ii) The error is proportional to $u^{(4)}$. In particular, since $u''(x) = f(x)$, this means that, if $f''(x) \equiv 0$, then the error is zero.

Remarkably, this investigation took up most of the class!

. . . . . .

6 Lecture 6: Error Analysis Part 1

The steps to doing an error analysis of our finite difference (FD) method are

(i) **Global error:** Define the *global error* (with respect to some norm);

(ii) **Local Truncation Error (LTE)**: estimate the *local truncation error*;

(iii) **Consistency:** relate the LTE to the *consistency* of the method.

(iv) **Stability:** prove the stability of the FD operator, with respect to some suitable norm.

(v) **Convergence**

We started with a discussion of "Big-$\mathcal{O}$" and "little-o" notation, based on A.2 of LeVeque.
We then covered items (i) and (ii) above, as explained in Sections 2.4 and 2.5 of LeVeque.
On Monday, we will digress for a discussion of vector and matrix norms.
On Tuesday we'll return to items (iii)-(v).

. . . . . .

7 Lecture 7: Norms

We are in processing of studying the error analysis on the finite difference method applied to problem (4.1). However, before we can complete this, we need to revise some important properties of *norms*.

First (today) we'll look at **vector** and **matrix** norms. Then we'll look at **grid** norms that were mentioned briefly during Lecture 6.

7.1 Three vector norms

When we want to consider the size of a real number, without regard to sign, we use the *absolute value*. Important properties of this function are:

(i) $|x| \geqslant 0$ for all x. (ii) $|x| = 0$ if and only if $x = 0$. (iii) $|\lambda x| = |\lambda||x|$. (iv) $|x + y| \leqslant |x| + |y|$.

This notion can be extended to vectors and matrices.

Definition 7.1. Let $\mathbb{R}^n$ be the set of all the vectors of length n of real numbers. The function $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ is called a **norm** on $\mathbb{R}^n$ if, for all $u, v \in \mathbb{R}^n$

1. $\|v\| \geqslant 0$,
2. $\|v\| = 0$ if and only if $v = 0$.
3. $\|\lambda v\| = |\lambda|\|v\|$ for any $\lambda \in \mathbb{R}$,
4. $\|u + v\| \leqslant \|u\| + \|v\|$ (triangle inequality).

The norms of a vector give us some information about the *size* of the vector. But there are different ways of measuring the size: you could take the absolute value of the largest entry, you cold look at the "distance" for the origin, etc... There are three important examples.

Definition 7.2. Let $\boldsymbol{v} \in \mathbb{R}^n$: $\boldsymbol{v} = (v_1, v_2, \ldots, v_{n-1}, v_n)^T$.

(i) The 1-*norm* (also known as the *Taxi cab* or *Manhattan* norm) is: $\|v\|_1 = \sum_{i=1}^{n} |v_i|$.

(ii) The 2-*norm* (a.k.a. the *Euclidean norm*) is: $\|v\|_2 = \left(\sum_{i=1}^{n} v_i^2 \right)^{1/2}$.

Note, if $\boldsymbol{v}$ is a vector in $\mathbb{R}^n$, then

$$\boldsymbol{v}^T\boldsymbol{v} = v_1^2 + v_2^2 + \cdots + v_n^2 = \|\boldsymbol{v}\|_2^2.$$

(iii) The ∞-*norm* (also known as the $\max$-*norm*) $\|v\|_\infty = \max_{i=1}^{n} |v_i|$.

Example 7.3. If $\boldsymbol{v} = (-2, 4, -4)^T$ then ...

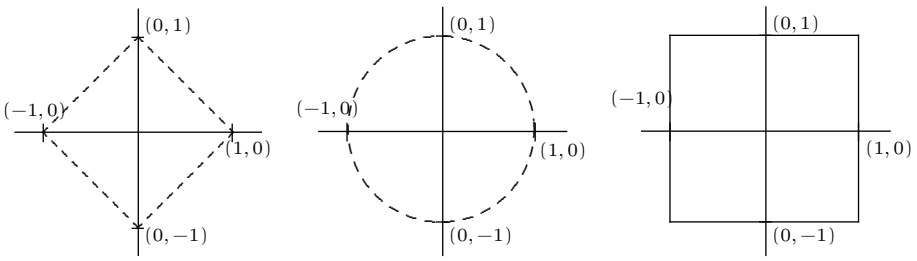

Fig. 7.1: The unit vectors in $\mathbb{R}^2$: $\|\boldsymbol{x}\|_1 = 1$, $\|\boldsymbol{x}\|_2 = 1$, $\|\boldsymbol{x}\|_\infty = 1$,

In Figure 7.1, the first diagram shows the unit ball in $\mathbb{R}^2$ given by the 1-norm: the vectors $\boldsymbol{x} = (x_1, x_2)$ in $\mathbb{R}^2$ are such that $\|\boldsymbol{x}\|_1 = |x_1| + |x_2| = 1$ are all found on the diamond (top left). In the second diagram, the vectors have $\sqrt{x_1^2 + x_2^2} = 1$ and so are arranged in a circle (top right). The bottom diagram gives the unit ball in $\|\cdot\|_\infty$, for which the largest component of each vector is 1.

It is easy to show that $\|\cdot\|_1$ and $\|\cdot\|_\infty$ are norms (see exercise). And it is not hard to show that $\|\cdot\|_2$ satisfies conditions (1), (2) and (3) of Definition 7.1. But it takes a little bit of effort to show that $\|\cdot\|_2$ satisfies the triangle inequality. First we need

Lemma 7.4 (Cauchy-Schwarz)**.**

$$|\sum_{i=1}^{n} u_i v_i| \leqslant \|\mathbf{u}\|_2 \|\boldsymbol{v}\|_2, \qquad \forall \mathbf{u}, \boldsymbol{v} \in \mathbb{R}^n.$$

The proof can be found in any text-book on analysis. Now can now apply it to show that

Lemma 7.5.

$$\|\mathbf{u} + \boldsymbol{v}\|_2 \leqslant \|\mathbf{u}\|_2 + \|\boldsymbol{v}\|_2.$$

It follows directly that

Corollary 7.6. $\|\cdot\|_2$ *is a norm.*

8 Lecture 8: Matrix Norms

8.1 Subordinate matrix norms

Definition 8.1. Given any norm $\|\cdot\|$ on $\mathbb{R}^n$, there is a *subordinate matrix norm* on $\mathbb{R}^{n\times n}$ defined by

$$\|A\| = \max_{\boldsymbol{v}\in\mathbb{R}^n_\star} \frac{\|A\boldsymbol{v}\|}{\|\boldsymbol{v}\|}, \tag{8.2}$$

where $A \in \mathbb{R}^{n\times n}$ and $\mathbb{R}^n_\star = \mathbb{R}^n/\{\mathbf{0}\}$.

You might wonder why we define a matrix norm like this. The reason is that we like to think of A as an *operator* on $\mathbb{R}^n$: if $\boldsymbol{v} \in \mathbb{R}^n$ then $A\boldsymbol{v} \in \mathbb{R}^n$. So rather than the norm giving us information about the "size" of the entries of a matrix, it tells us how much the matrix can change the size of a vector.

The formula for a subordinate matrix norm in Definition 8.1, is sensible, but not much use to if we actually want to compute, say, $\|A\|_1$, $\|A\|_\infty$ or $\|A\|_2$. For a given A, we'd have to calculate $\|A\boldsymbol{v}\|/\|\boldsymbol{v}\|$ for *all* $\boldsymbol{v}$. And there is rather a lot of them. Fortunately, there are some easier ways of computing the more important norms. We'll see that

- The ∞-norm of a matrix is just largest absolute-value row sum.
- The 1-norm of a matrix is just largest absolute-value column sum.
- The 2-norm of the matrix A is the square root of the largest eigenvalue of A^TA.

8.1.1 The max-norm and 1-norm on $\mathbb{R}^{n\times n}$

Theorem 8.2. *For any $A \in \mathbb{R}^{n\times n}$ the subordinate matrix norm associated with $\|\cdot\|_\infty$ on $\mathbb{R}^n$ can be computed by*

$$\|A\|_\infty = \max_{i=1,\dots n} \sum_{j=1}^{n} |a_{ij}|.$$

A similar result holds for the 1-norm, the proof of which is left as an exercise:

Theorem 8.3.

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{i,j}|. \tag{8.3}$$

Computing the 2-norm of a matrix is a little harder that computing the 1- or ∞-norms. However, later we'll need estimates not just for $\|A\|$, but also $\|A^{-1}\|$. And, unlike the 1- and ∞-norms, we can estimate $\|A^{-1}\|_2$ without explicitly forming A^{-1}.

8.2 Eigenvalues

We begin by recalling some important facts about eigenvalues and eigenvectors.

Definition 8.4. Let $A \in \mathbb{R}^{n\times n}$. We call $\lambda \in \mathbb{C}$ an *eigenvalue* of A if there is a non-zero vector $\boldsymbol{x} \in \mathbb{C}^n$ such that

$$A\boldsymbol{x} = \lambda\boldsymbol{x}.$$

We call any such $\boldsymbol{x}$ an *eigenvector associated with* A.

Some properties of eigenvalues:

(i) If A is a real symmetric matrix (i.e., $A = A^T$), its eigenvalues and eigenvectors are all real-valued.

(ii) If λ is an eigenvalue of A, the $1/\lambda$ is an eigenvalue of A^{-1}.

(iii) If $\boldsymbol{x}$ is an eigenvector associated with the eigenvalue λ then so too is $\eta\boldsymbol{x}$ for any non-zero scalar η.

(iv) An eigenvector may be *normalised* as
$\|\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{x} = 1$.

(v) There are n eigenvectors $\lambda_1, \lambda_n, \ldots, \lambda_n$ associated with the real symmetric matrix A. Let $\mathbf{x}^{(1)}$, $\boldsymbol{x}^{(2)}$, ..., $\boldsymbol{x}^{(n)}$ be the associated normalised eigenvectors. Then the eigenvectors are linearly independent and so form a basis for $\mathbb{R}^n$. That is, any vector $v \in \mathbb{R}^n$ can be written as a linear combination:

$$\boldsymbol{v} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}^{(i)}.$$

(vi) Furthermore, these eigenvectors are *orthogonal* and *orthonormal*:

$$\left(x^{(i)}\right)^T x^{(j)} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

8.3 Singular values

The *singular values* of a matrix A are the square roots of the eigenvalues of A^TA. They play a very important role in matrix analysis and in areas of applied linear algebra, such as image and text processing. Our interest here is in their relationship to $\|A\|_2$.

Lemma 8.5. *For any matrix A, the eigenvalues of A^TA are real and non-negative.*

9 Lecture 9: More matrix norms

9.1 Singular values, again

At the end of Lecture 7 we proved that if we take any real, square matrix, A, and set $B = A^TA$, then the eigenvalues of B are non-negative real numbers. A consequence of this is that the square root of the eigenvalues of A^TA are also non-negative and real. Furthermore, part of the above proof involved showing that, if $\left(A^TA\right)\boldsymbol{x} = \lambda x$, then

$$\sqrt{\lambda} = \frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}.$$

This at the very least tells us that

$$\|A\|_2 := \max_{\boldsymbol{x} \in \mathbb{R}^n_\star} \frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \geqslant \max_{i=1,\ldots,n} \sqrt{\lambda_i}.$$

With a bit more work, we can show that if $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$ are the the eigenvalues of $B = A^TA$, then

$$\|A\|_2 = \sqrt{\lambda_n}.$$

Theorem 9.1. *Let $A \in \mathbb{R}^{n\times n}$. Let $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$, be the eigenvalues of $B = A^TA$. Then*

$$\|A\|_2 = \max_{i=1,\ldots,n} \sqrt{\lambda_i} = \sqrt{\lambda_n},$$

9.2 Consistency of matrix norms

It should be clear from (8.2) that, if $\|\cdot\|$ is a subordinate matrix norm then, for *any* $\mathbf{u} \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$,

$$\|A\mathbf{u}\| \leqslant \|A\|\|\mathbf{u}\|.$$

It is an important result: we'll need it tomorrow. There is an analogous statement for the product of two matrices:

Definition 9.2. A matrix norm $\|\cdot\|$ is *consistent* if

$$\|AB\| \leqslant \|A\|\|B\|, \qquad \text{for all } A, B \in \mathbb{R}^{n\times n}.$$

Theorem 9.3. *Any subordinate matrix norm is consistent.*

10 Class 10: Finite differences in Matlab (1)

(a) Download the Matlab programmes `>> Test_FD.m` and `>> Solve_FD.m` , from the course website, that implement the finite method for solving

$$u''(x) = f(x) \text{ on } (a,b); \quad u(a) = \alpha, u(b) = \beta.$$

Use it to verify that the method is indeed second order.

(b) The `>> for` -loop used in `>> Solve_FD.m` is very slow. Use the `>> sparse` function to do this more efficiently.

(c) Modify the program so that it can solve

$$u''(x) - u(x) = f(x) \text{ on } (a,b); \quad u(a) = \alpha, u(b) = \beta.$$

The method is

$$\begin{gathered} U_0 = \alpha \\ \frac{1}{h^2}(U_{k-1} - 2U_k + U_{k+1}) - u(x_k) = f(x_k) \quad \text{for } k = 1, \ldots, N-1 \\ U_N + \beta. \end{gathered}$$

Is this still second-order accurate?

(d) Next modify the program so that it can solve

$$u''(x) - u'(x) = f(x) \text{ on } (a,b); \quad u(a) = \alpha, u(b) = \beta.$$

For this we need to approximate $u'(x)$. Revise the methods we derived in Lectures 2 and 3, including central differences (D_0), backward differences (D_-), and forward differences (D_+). Which is most accurate?

(e) Finally, modify fy the program so that it can solve

$$\varepsilon u''(x) - u'(x) = f(x) \text{ on } (a,b); \quad u(a) = \alpha, u(b) = \beta.$$

where $\varepsilon = 10^{-2}$. How which method is most accurate?

11 Class 11: Consistence + Stability $\implies$ Convergence

This class followed Sections 2.6, 2.7, 2.8 and 2.9 of LeVeque pretty closely.

12 Class 12: Convergence in the 2-norm

Let the vector U be the finite difference solution, and $\hat{U}$ be the vector obtained by evaluating the true solution at mesh points. Set $E = U - \hat{U}$. We proved that there is a constant which is independent of N (and pretty close to 1) such that

$$\|E\|_2 \leqslant \frac{1}{c\pi^2}\|\tau\|_2.$$

It follows that $\|E\|_2$ is $\mathcal{O}(h^2)$.

A key part of this was noting that our matrix A has eigenvalues

$$\lambda_p = \frac{2}{h^2}\big(\cos(p\pi h) - 1\big),$$

with corresponding eigenvectors whose entries are

$$(u_p)_j = \sin(p\pi j h).$$

For more details, see Section 2.10 of LeVeque.

13 Class 13: Convergence in the $\max$-norm

First I gave a brief (and probably confusing) description of Green's functions for differential and difference operators. I then left it as an **exercise** for you to read Section 2.11 of the text.

The alternative to using Green's functions techniques is to employ *maximum principles*, which usually require no more than elementary calculus.

Recalling that, as usual, we are trying to solve the differential equation

$$u''(x) = f(x) \text{ on } (a, b); \quad u(a) = \alpha, u(b) = \beta.$$

To allow for some generalisation later, we define the differential operator: $Lu := u''$. So now the problem is,

$$Lu(x) = f(x) \text{ on } (a, b); \quad u(a) = \alpha, u(b) = \beta.$$

When then proved our first result:

Lemma 13.1 (Continuous Maximum Principle)**.** *If* $Lu \geqslant 0$*, for all* $x \in [a, b]$*, and* $u(a) = \alpha$*,* $u(b) = \beta$*, then* $u \leqslant \min\{\alpha, \beta\}$.

14 Class 14: Maximum principles for differential equations

(After a little party), we reviewed (13.1), and its implications for the special case where $\alpha = \beta = 0$.

We then talked about norms of functions, and particularly the $\sup$-norm.

We also showed that, if $f > 0$, then

$$\frac{1}{2}\|f\|_\infty(x - a)(x - b) \leqslant u(x) \leqslant 0.$$

A direct corollary of this is that

$$\|u\|_\infty \leqslant \frac{1}{8}\|f\|_\infty.$$

15 Class 15: Maximum principles for difference equations

If the finite difference method is written as $AU = F$, we showed that it is possible to establish many results that are analogous to those for differential equations.

This culminated in showing that, if $AE = -\tau$, then there is a constant C, independent of N, such that $\|E\|_\infty \leqslant C\|\tau\|_\infty$. This implies convergence of the finite difference method.

We noted that, although this approach does not explicitly bound $\|A^{-1}\|_\infty$, it does give bounds for solutions to $AU = F$ in terms of $\|f\|_\infty$, which is equivalent.

If one did want an explicit bound for $\|A^{-1}\|$, derived by computing this directly, one can do this by showing that, if $Ad = e_k$, where e_k is the k^{th} column of the identity matrix, I, then

$$e_i = h \begin{cases} (x_j - 1)(x_i) & i \leqslant j \\ (x_i - 1)(x_j) & i \geqslant j. \end{cases}$$

This detail was taken directly from Section 2.11 of LeVeque.

This shows that the entries of A^{-1} are bounded by h, and so that the maximum row sum of A^{-1} is bounded by a constant.

16 Class 16: Neumann Boundary Conditions (1)

This closely followed LeVeque Sections 2.12 and 2.13 where we derived schemes for approximating problems of the form

$$u''(x) = f(x) \quad \text{on } (0,1), \tag{16.4a}$$

subject to the boundary conditions

$$u'(0) = \sigma, \qquad u(1) = \beta. \tag{16.4b}$$

We considered the physical meaning of such equations, and three different ways of deriving a FDM that implemented the Nuemann condition at the left-hand boundary.

17 Class 17: Neumann Boundary Conditions (2)

Following on from the previous class, we considered uniqueness of solutions to Bps. We showed that, if the differential operator satisfies a maximum principle, then it can have at most one solution.

In contrast, we saw that if the boundary conditions in (16.4) were changed to:

$$u'(0) = \sigma_0, \qquad u'(1) = \sigma_1,$$

then the problem may have no solution, or an infinite number of solutions, but never a single unique solution.

18 Class 18: General BVPs

In today's class, we considered how the methods we have studied so far could be extended to solve the problem

$$u''(x) + p(x)u'(x) + q(x)u(x) = f(x) \quad \text{on } (0,1),$$

with suitable boundary conditions.

However, most of the focus was on the case when $p \equiv 0$ and q is strictly negative.

19 Class 19: More on general BVPs

Today we focused on solving

$$u''(x) + p(x)u'(x) = f(x) \quad \text{on } (0,1),$$

where $p \neq 0$. Usually, we discretize the second-order term as

$$u''(x_i) = \frac{1}{h}\big(u(x_{i-1}) - 2u(x_i) + u(x_{i+1})\big) + \mathcal{O}(h^2).$$

That is, we used a two-sided approximation for which the truncation error is $\mathcal{O}(h^2)$. Therefore, it seems reasonable to also approximate $u'(x)$ with two-sided approximation for which the truncation error is $\mathcal{O}(h^2)$:

$$u'(x_i) = \frac{1}{2h}\big(-u(x_{i-1}) + u(x_{i+1})\big) + \mathcal{O}(h^2).$$

However, we discovered that the resulting finite difference method satisfies a discrete maximum principle only if N is large enough, relative to $\|q\|_\infty$.

An alternative approach is, if $p < 0$, to approximate the first-order term using the backward difference scheme $D_-(u)(x_i)$:

$$u'(x_i) = \frac{1}{h}\big(-u(x_{i-1}) + u(x_{i+1})\big) + \mathcal{O}(h).$$

If $p > 0$, we can use the forward difference scheme, $d_+(u)(x_i)$. Either way, the rate of convergence of the method drops to $\mathcal{O}(h)$, but the method satisfies a maximum principle for all N, so is stable.

20 Class 20: Elliptic BVPs in two dimensions

Today we moved on to two-dimensional problems, and considered how we might solve the very general problem

$$a_1\frac{\partial^2}{\partial x^2}u + a_2\frac{\partial^2}{\partial x \partial y}u + a_3\frac{\partial^2}{\partial y^2}u + a_4\frac{\partial}{\partial x}u + a_5\frac{\partial}{\partial x}u + u_6 u = f$$

on some domain $\Omega \in \mathbb{R}^2$. We are interested in the specific case of *elliptic problems*, where $a_2^2 < 4a_1a_3$. However, for the most part we'll restrict our interest to the case $a_1 = a_3 = 1$, and $a_2 = 0$. Moreover, to keep everything simple, we'll set $a_4 = a_5 = a_6 = 0$. This leaves us with the classic *Poisson problem*:

$$u_{xx}(u, y) + u_{yy}(x, y) = f(x, y).$$

We went through some notation on the Laplacian (and gradient and divergence operators), before beginning think about how to apply a finite difference method.

21 Class 21: A finite difference method for the Poisson problem

We saw how to extend our finite difference method for solving

$$u''(x) = f(x) \qquad \text{on } (0,1), \quad u(0) = u(1) = 0,$$

to solving

$$u_{xx}(x, y) + u_{yy}(x, y) = f(x, y) \qquad \text{on } \Omega = (0,1)^2, \quad u(\partial\Omega) = 0. \tag{21.5}$$

We focused on solving the problem on the smallest nontrivial grid, taking $N = 4$ intervals in each coordinate direction.

22 Class 22: Ordering unknowns

After spending some time talk about solutions to problem sheets, we studied how to the Poisson problem with a 5-point FEM on an a uniform $N \times N$ grid. I'm not sure how we it happened, but we manged to spend almost the entire class just talking about lexicographic ordering, and how to represent the FEM as a matrix-vector equation.

23 Class 23: Expressing the discrete Laplacian using Kronecker products

(There was no class on the 27th because it was a public holiday).

When solving (21.5) with a finite difference method on an $N \times N$ grid leads to the linear system

$$AU = F,$$

where, for $N = 4$ (for example)

$$A = \frac{1}{h^2}\begin{pmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{pmatrix}$$

If we let T_3 be the matrix

$$T_3 = \begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}$$

then A has the block form

$$\frac{1}{h^2}\begin{pmatrix} T_3 - 2I & I & 0 \\ I & T_3 - 2I & I \\ 0 & I & T_3 - 2I \end{pmatrix}$$

This can be expressed even more succinctly using *Kronecker products*.

Definition 23.1. If $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, then the *Kronecker product* of A and B is the $A \otimes B \in \mathbb{R}^{(mp)\times(nq)}$ given by

$$\begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & & & \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{pmatrix}$$

Then, letting $T_{N,N} = h^2 A$, we have

$$T_{3,3} = I \otimes T_3 + T_3 \otimes I.$$

However, this extends to any N:

$$T_{N,N} = I \otimes T_N + T_N \otimes I.$$