Chapter 7 BOUNDARY VALUE PROBLEMS

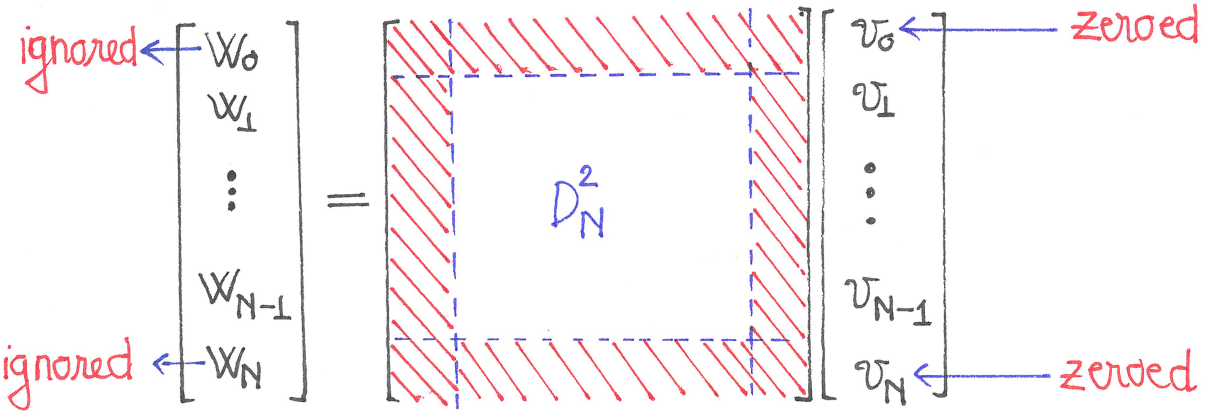
* Boundary value problem 1.

$$\begin{cases} u_{xxx} = e^{4x} & -1<x<1 \\ u(\pm 1) = 0 \end{cases}$$

This is a 1d-Poisson equation, with solution $u(x)=\frac{1}{16}(e^{4x} - x\sinh 4 - \cosh 4)$

(i) To solve the problem numerically, we can compute the second derivative via D_N^2, the square of D_N. D_N^2 can be computed either by squaring D_N, which costs $O(N^3)$ floating point operations, or by explicit formulas or recurrences, which costs $O(N^2)$ flops.

(ii) Imposition of the boundary condition: We take the interior Chebyshev points $x_1, \dots, x_{N-1}$ as our computational grid, with $v=(v_1, \dots v_{N-1})^T$ as the corresponding vector of unknowns.



To solve our 1D Poisson problem by a Chebyshev spectral method, we can make use of the $(N-1)\times(N-1)$ matrix $\underline{\tilde{D}}_N^2$ obtained by stripping $\underline{D}_N^2$ of the first & last rows and columns.

$$\tilde{D}_N^2 = \tilde{D}_N^2(1:N-1,\ 1:N-1) \qquad \text{MATLAB Notation}$$

$$\updownarrow \text{ Matlab Code}$$

$$D2 = D2(2:N,\ 2:N) = D2(2:\text{end}-1,\ 2:\text{end}-1)$$

$\Longrightarrow$ Numerical solution: Solving a linear system $\underline{\tilde{D}}_N^2\,\underline{v} = \underline{f}$

✳ BVP 1: $\begin{cases} u_{xx} = e^{4x} & -1<x<1 \\ u(\pm 1) = 0 \end{cases}$

Spectral differentiation is then carried out like this.

- Let $p(x)$ be the unique polynomial of degree $\leq N$ with $p(\pm 1)=0$ and $p(x_j) = v_j \quad 1 \leq j \leq N-1$.
- Set $w_j = p''(x_j) \quad 1 \leq j \leq N-1$.

D_N^2 is an $(N+1)\times(N+1)$ matrix that maps a vector $(v_0, \ldots, v_N)^T$ to a vector $(w_0, \ldots, w_N)^T$. The procedure just described amounts to the decision:

- Fix v_0 and v_N at zero.
- Ignore w_0 and w_N.

Since the Dirichlet condition is imposed at the whole boundary, the so-call reaction as the derivative u_x at $x=\pm 1$ will happen to be computable in the post-processing – thus derivative $u_x(\pm 1)$ is ignored.

Although the algorithm calculates the vector $(v_1, \ldots, v_N)^T$ of approximations to u at the grid points, as always with spectral methods, we really have more information about the numerical solution than just point values. Implicitly we are dealing with a polynomial interpolant $p(x)$.

✳ Boundary value problem 2 Nonlinear differential equation

$$u_{xxx} = e^u \qquad -1 < x < 1, \quad u(\pm 1) = 0.$$

As we know, we normally solve the nonlinear problem iteratively by Newton method or fixed point method. By using the spectral differentiation, we may rewrite the differential equation in the discrete form formally as.

$$\underline{\tilde{D}}_N^2 \underline{u} = e^{\underline{u}} = \exp(\underline{u}).$$

⟹ Solution Method: It is obvious that the fixed point method is the first candidate

$\Longrightarrow$ Solution Method: Fixed point scheme

$$\underline{\widetilde{D}}_N^2 \underline{v}_{new} = \exp(\underline{v}_{old})$$

where $\exp(\underline{v})$ is the column vector defined componentwise by $[\exp(v)]_j = \exp(v_j)$. The initial guess, such as the vectors of zeros, must be supplied to the fixed point solver.

$*$ Boundary value problem 3: Eigenvalue boundary value problem

$$u_{xx} = \lambda u \qquad -1 < x < 1. \qquad u(\pm 1) = 0.$$

Again, we need to discretize this equation taking into account the boundary conditions. Note that the eigenvalue problem in the current form of differential equation depends mainly on the boundary condition. The appropriate discretized form is: $\underline{\widetilde{D}}_N^2 \underline{u} = \lambda \underline{u} \Rightarrow (\underline{\widetilde{D}}_N^2 - \lambda I)\underline{u} = \underline{0}$

According to the discretized equation:

$$\widetilde{D}_N^2 \underline{u} = \lambda \underline{u} \iff (\widetilde{D}_N^2 - \lambda \underline{I})\underline{u} = 0$$

Thus the eigenvalues of this boundary value problem can be approximately evaluated as those of the matrix $\widetilde{D}_N^2$. This computation can be easily carried out by using the built-in function in MATLAB. The theoretical eigenvalues of this BVP are

$$\lambda = -\frac{\pi^2 n^2}{4} \quad n = 1, 2, \ldots$$

with corresponding eigenfunctions $\sin\left(n\pi \frac{x+1}{2}\right)$.

Exercise Prove that the eigenvalues & the corresponding eigenfunctions of the BVP: $u_{xx} = \lambda u \quad -1 < x < 1, \quad u(1) = u(-1) = 0$. are

$$\lambda_n = -\frac{\pi^2 n^2}{4}, \qquad u_n = \sin\left(n\pi \frac{x+1}{2}\right), \quad n = 1, 2, \ldots,$$

respectively.

✳ Boundary value problem 4: 2D Poisson equation with homogeneous Dirichlet boundary condition

$$u_{xx} + u_{yy} = 10 \sin[8x(y-1)], \quad -1 < x, y < 1, \quad u = 0 \text{ on the boundary}$$

For such a problem we naturally set up a grid based on Chebyshev points independently in each direction, called a tensor product grid. The easiest way to solve a problem on a tensor product spectral grid is to use tensor product in linear algebra, also known as Kronecker products. The Kronecker product of two matrices A and B is denoted by $A \otimes B$ and is computed in MATLAB by the command kron(A, B).

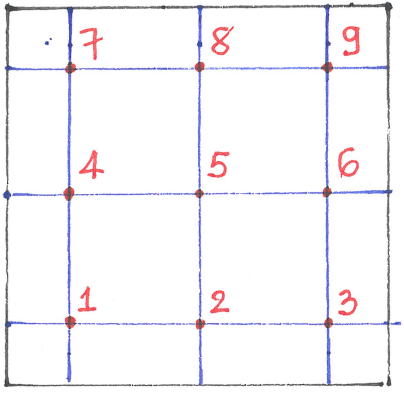
If A and B are of dimensions $p \times q$ and $r \times s$, respectively, then $A \otimes B$ is the matrix of dimension $pr \times qs$. with $p \times q$ block form, where the (i,j) block is $a_{ij} B$

Example: Kronecker product

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \otimes \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \left[\begin{array}{cc:cc:cc} a & b & 2a & 2b & 3a & 3b \\ c & d & 2c & 2d & 3c & 3d \\ \hdashline 4a & 4b & 5a & 5b & 6a & 6b \\ 4c & 4d & 5c & 5d & 6c & 6d \end{array}\right]$$

Illustration of how Kronecker products can be used for spectral methods

Consider the case $N=4$, and suppose we number the internal nodes in the lexicographic ordering



Task: We wish to approximate the Laplacian by differentiating spectrally in the x and y directions independently.

The 3×3 differentiation matrix with $N = 4$ in 1D:

```
D = cheb(4);   D2 = D^2;   D2 = D2(2:4, 2:4)
```

I denotes the 3×3 identity.

$$\tilde{D}_4^2 = \begin{bmatrix} -14 & 6 & -2 \\ 4 & -6 & 4 \\ -2 & 6 & -14 \end{bmatrix}$$

⟹ Second derivative with respect to x

```
kron(I, D2)
```

⟹ Second derivative with respect to y

```
kron(D2, I)
```

⟹ Discrete Laplacian

$$L_N = I \otimes \tilde{D}_N^2 + \tilde{D}_N^2 \otimes I$$

Explanation comes next: how kron(I, D_2) corresponds to D_{xx}.

kron(D_2, I) corresponds to D_{yy}

by a short/closer look at the result matrices

$$I \otimes \widetilde{D}_N^2 = \left[\begin{array}{ccc|ccc|ccc} -14 & 6 & -2 & & & & & & \\ 4 & -6 & 4 & & & & & & \\ -2 & 6 & -14 & & & & & & \\ \hline & & & -14 & 6 & -2 & & & \\ & & & 4 & -6 & 4 & & & \\ & & & -2 & 6 & -14 & & & \\ \hline & & & & & & -14 & 6 & -2 \\ & & & & & & 4 & -6 & 4 \\ & & & & & & -2 & 6 & -14 \end{array}\right] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \end{bmatrix}$$

v_1 v_2 v_3 — v_4 v_5 v_6 — v_7 v_8 v_9

We see that (v_1, v_2, v_3), (v_4, v_5, v_6), (v_7, v_8, v_9) stay in the same row with the same y-coordinate within each group. That is

$v_1 = u(x_3, y_3)$ $\quad v_2 = u(x_2, y_3)$ $\quad v_3 = u(x_1, y_3)$ ⟶ fix y-coord

$v_4 = u(x_3, y_2)$ $\quad v_5 = u(x_2, y_2)$ $\quad v_6 = u(x_1, y_2)$ & so on.

$$\widetilde{D}_N^2 \otimes I = \left[\begin{array}{ccc|ccc|ccc|c}
-14 & & & 6 & & & -2 & & & v_1 \\
 & -14 & & & 6 & & & -2 & & v_2 \\
 & & -14 & & & 6 & & & -2 & v_3 \\
\hline
4 & & & -6 & & & 4 & & & v_4 \\
 & 4 & & & -6 & & & 4 & & v_5 \\
 & & 4 & & & -6 & & & 4 & v_6 \\
\hline
-2 & & & 6 & & & -14 & & & v_7 \\
 & -2 & & & 6 & & & -14 & & v_8 \\
 & & -2 & & & 6 & & & -14 & v_9
\end{array}\right]$$

Similarly, we see that the node values $\{v_1, v_4, v_7\}$, $\{v_2, v_5, v_8\}$, $\{v_3, v_6, v_9\}$ stay in the same x-coordinate within each group. That is

$v_1 = u(x_3, y_3)$ $\quad v_4 = u(x_3, y_2)$ $\quad v_7 = u(x_3, y_1)$ → fix x-coord

$v_2 = u(x_2, y_3)$ $\quad v_5 = u(x_2, y_2)$ $\quad v_6 = u(x_2, y_1)$ → fix x-coord

$v_3 = u(x_1, y_3)$ $\quad v_6 = u(x_1, y_2)$ $\quad v_9 = u(x_1, y_1)$.

$v_1 = u(x_3, y_3)$ $\quad v_2 = u(x_2, y_3)$ $\quad v_3 = u(x_1, y_3) \longrightarrow$ fix y-coord

$v_4 = u(x_3, y_2)$ $\quad v_5 = u(x_2, y_2)$ $\quad v_6 = u(x_1, y_2) \longrightarrow$ fix y-coord

$v_7 = u(x_3, y_1)$ $\quad v_8 = u(x_2, y_2)$ $\quad v_9 = u(x_1, y_2) \longrightarrow$ fix y-coord

$\downarrow$ fix x-coord $\quad \downarrow$ fix x-coord $\quad \downarrow$ fix x-coord.

Laplacican operator in the discrete form by means of spectral differentiation matrix.

$$\underline{L}_N = \underbrace{\underline{I} \otimes \underline{\tilde{D}}_N^2}_{\text{Derivative } D_{xx}} + \underbrace{\underline{\tilde{D}}_N^2 \otimes \underline{I}}_{\text{Derivative } D_{yy}}$$

✳ Boundary value problem 5: Helmholtz equation

Helmholtz equation is a variation of the Poisson equation

$$u_{xx} + u_{yy} + k^2 u = f(x,y) \qquad -1<x,y<1$$

$u=0$ on the boundary

k is a real parameter

- This equation arises in the analysis of wave propagation governed by the equation

$$-U_{tt} + U_{xx} + U_{yy} = e^{ikt} f(x,y) \qquad -1<x,y<1$$

$U=0$ on the boundary

after separation of variables to get $U(x,y,t) = e^{ikt} u(x,y)$.

- In our numerical example we use

$k=9$ and $f(x,y) = \exp\{-10[(y-1)^2 + (x-\frac{1}{2})^2]\}$.

Exercise Solve the boundary value problem numerically

$$u_{xx} + 4u_x + e^x u = \sin(8x) \qquad -1 < x, y < 1$$

$$u(\pm 1) = 0.$$

To ten digits of accuracy, what is $u(0)$?

Exercise Devise an alternative to Program 14 which solves the nonlinear problem

$$u_{xx} = \exp(u) \qquad -1 < x < 1, \quad u(\pm 1) = 0.$$

based on Newton iteration rather than fixed-point iteration, and make it work. Do you observe quadratic convergence?