# CPSC 303: ADJACENCY MATRICES, SPLINES, AND THE HEAT EQUATION

JOEL FRIEDMAN

## Contents

**Disclaimer:** The material may sketchy and/or contain errors, which I will elaborate upon and/or correct in class. For those not in CPSC 303: use this material at your own risk...

*Date*: Thursday 11$^{\text{th}}$ April, 2024, at 13:18(get rid of time in final version).
Research supported in part by an NSERC grant.

## 1. Review of Splines

In class we explained the sense in which splines are "localized:" namely, given a function $f\colon \mathbb{R} \to \mathbb{R}$ and real numbers $A = x_0 < x_1 < \cdots < x_n = B$, there is a unique function $v \in C^2[A, B]$ that minimizes the energy

$$\mathrm{Energy}_2(u) \overset{\mathrm{def}}{=} \int_A^B \bigl(u''(x)\bigr)^2\,dx$$

(hence we view $\mathrm{Energy}_2\colon C^2[A, B] \to \mathbb{R}$) subject to the conditions

$$v(x_0) = f(x_0), v(x_1) = f(x_1), \ldots, v(x_n) = f(x_n).$$

$v = v(x)$ is known as the "natural cubic spline" through $(x_i, f(x_i))$ for $i = 0, 1, \ldots n$. Moreover, for each $i$, for $x_i \le x \le x_{i+1}$ we have $v(x) = s_i(x)$, where $s_i$ is a cubic polynomial

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3,$$

we have that $a_i, b_i, d_i$ can be written as functions of

$$c_i,\ c_{i+1},\ h_i = x_{i+1} - x_i,\ f(x_i),\ f[x_i, x_{i+1}].$$

In this sense, the $a_i, b_i, d_i$ dependend only on "nearby" values of $\mathbf{c}$, i.e., $c_i, c_{i+1}$, and the values $x_i, x_{i+1}$ and the values of $f$ there. It turns out that setting $\mathbf{c} = (c_1, \ldots, c_{n-1})$, and setting $c_0 = c_n = 0$ (when needed), we have that $\mathbf{c}$ can be determined by the equations

(1)

$$\begin{bmatrix} 2 & \frac{h_1}{h_0+h_1} & & & \\ \frac{h_1}{h_1+h_2} & 2 & \frac{h_2}{h_1+h_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{h_{n-2}}{h_{n-2}+h_{n-1}} & 2 & \frac{h_{n-1}}{h_{n-2}+h_{n-1}} \\ & & & \frac{h_{n-1}}{h_{n-1}+h_n} & 2 \end{bmatrix} \mathbf{c} = 3\Phi, \quad \text{where} \quad \Phi = \begin{bmatrix} f[x_0, x_1, x_2] \\ f[x_1, x_2, x_3] \\ \vdots \\ f[x_{n-3}, x_{n-2}, x_{n-1}] \\ f[x_{n-2}, x_{n-1}, x_n] \end{bmatrix}$$

(see class notes or [A&G], top of page 343, Section 11.3, where the $i$-th equation/row is divided by $h_{i-1} + h_i$).

To understand (1) in a concrete example, if $h_i = x_{i+1} - x_i$ are all equal, so we may write $h = h_i$ for all $i$, (1) becomes

$$(4 + N_{\mathrm{rod},n})\mathbf{c} = 6\Phi,$$

where

$$N_{\mathrm{rod},n} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \end{bmatrix}.$$

It follows that

$$\mathbf{c} = (4I + N_{\mathrm{rod},n})^{-1} 6\Phi,$$

where $I$ is the identity matrix (of size $n \times n$); we easily see that $\|N_{\mathrm{rod},n}\|_\infty \le 2$, and hence we get a convergent series

$$(4I + N_{\mathrm{rod},n})^{-1} = (1/4)\Big(I - \big(N_{\mathrm{rod},n}/4\big) + \big(N_{\mathrm{rod},n}/4\big)^2 - \cdots\Big)$$

and therefore

$$\mathbf{c} = \Big(I - \big(N_{\mathrm{rod},n}/4\big) + \big(N_{\mathrm{rod},n}/4\big)^2 - \cdots\Big)(3/2)\Phi. \tag{2}$$

We will see below that for any $k \in \mathbb{N}$, $N^k_{\mathrm{rod},n}$ has all its entries "near the diagonal" in the sense that its $i, j$ entry is 0 if $|i-j| \ge k+1$. Hence we have

$$(4I + N_{\mathrm{rod},n})^{-1} = (1/4)\sum_{m=0}^{\infty} \big(-N_{\mathrm{rod},n}/4\big)^k$$

which for any $k \in \mathbb{N}$ can be written as

$$(1/4)\sum_{m=0}^{k} \big(-N_{\mathrm{rod},n}/4\big)^k + (1/4)\sum_{m=k+1}^{\infty} \big(-N_{\mathrm{rod},n}/4\big)^k, \tag{3}$$

where the first sum has its $(i,j)$-th entry 0 unless $|i-j| \le k$, and the second sum has $\infty$-norm (and hence each entry) bounded by

$$(1/4)\sum_{m=k+1}^{\infty} \big\|(-N_{\mathrm{rod},n}/4)^k\big\|_\infty \le (1/4)\sum_{m=k+1}^{\infty} \|N_{\mathrm{rod},n}/4\|^k_\infty = (1/4)\sum_{m=k+1}^{\infty} (1/2)^k = 1/2^{k+2}.$$

Hence (3) means that up difference of at most $1/2^{k+2}$ in each row, $(4I + N_{\mathrm{rod},n})^{-1}$ has all its nonzero entries within $k$ of the diagonal. Hence, for any $k \in \mathbb{N}$, (2) implies that each $c_i$ depends only on the $f[x_{j-1}, x_j, x_{j+1}]$ where $|i-j| \le k$ up to a difference of at most $(1/2^{k+2})\|\Phi\|_\infty$. In this sense each piece of the natural spline $s_i(x)$ depends only on the values of $f(x_j)$ for $j$ "near $i$."

### 1.1. Why the Name $N_{\mathrm{rod},n}$?

We use the notation $N_{\mathrm{rod},n}$ because this matrix features as the one-dimensional Laplacian of a one-dimensional metal rod. Roughly speaking, the reason is that we have second derivative approximation:

$$f''(x_0) = \frac{f(x_0+h) + f(x_0-h) - 2f(x_0)}{h^2} + O(h^2)$$

for a fixed, four times differentiable function $f\colon \mathbb{R} \to \mathbb{R}$, a fixed $x_0 \in \mathbb{R}$, and $h \to 0$ (see the middle of page 412, Subsection 14.1.4). Hence if $x_0, x_1, x_2, \ldots, x_n$ are evenly spaced reals with $h = x_{i+1} - x_i$, and $f$ is a function with $f(x_0) = f(x_n) = 0$ (this condition is called the *Dirichlet condition on* $f$), and for any $f\colon \mathbb{R} \to \mathbb{R}$ we set

$$f(\mathbf{x}) \overset{\mathrm{def}}{=} \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n-1}), \end{bmatrix}$$

then

$$f''(\mathbf{x}) \approx \frac{N_{\mathrm{rod},n} - 2I}{h^2} f(\mathbf{x}) + O(h^2).$$

## 2. Directed Graphs and Graphs, and their Adjacency Matrices

One easy way to understand the powers of $N_{\mathrm{rod},n}$ comes from the fact that this matrix is the *adjacency matrix* of a very simple graph, namely the graph $P_{n-1}$ often called the *path of length* $n-1$. Let us review the definitions.

By a *simple directed graph* we mean a pair $G = (V, E)$, where $V$ is a finite set — called the *vertex set* — and $E \subset V \times V$ — called the *(directed) edge set*, i.e., $E$ consist of ordered pairs of element of $V$.[1] For such a graph, we define the *adjacency matrix of* $G$, denoted $A_G$, to be the square matrix indexed on the set $V$, whose entries are

$$(A_G)_{v,v'} = \begin{cases} 1 & \text{if } (v,v') \in E, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

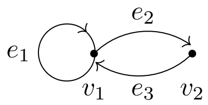


Figure 1. The Fibonacci Graph

**Example 2.1.** In class we likely discussed the Fibonacci graph, $G_{\mathrm{Fib}} = (V, E)$, where

$$V = \{v_1, v_2\}, \quad E = \{e_1 = (v_1, v_1), e_2 = (v_1, v_2), e_3 = (v_2, v_1)\};$$

see Figure 1. Once we order the vertices of $G$ as $V = \{v_1, \ldots, v_n\}$, we can view $A_G$ as an $n \times n$ matrix. The matrix $A_{G_{\mathrm{Fib}}}$ is therefore a $2 \times 2$ matrix. In class we explain that for any $k \in \mathbb{N}$ the entries of $A_G^k$ are given by $(A_G^k)_{v,v'}$ is the number of *walks of length* $k$ *from* $v$ *to* $v'$ *of length* $k$, i.e., the number of sequences

$$(v = u_0, u_1, \ldots, u_{k+1} = v')$$

such that $(u_i, u_{i+1}) \in E$ for all $i$. For example, the Fibonacci graph has 5 walks of length 2:

$$(v_1, v_1, v_1),\ (v_1, v_1, v_2),\ (v_1, v_2, v_1),\ (v_2, v_1, v_1),\ (v_2, v_1, v_2),$$

and we easily check

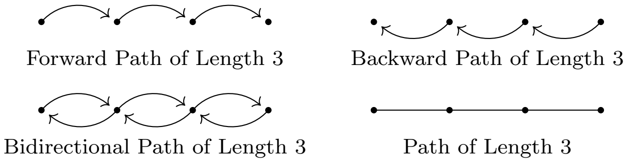
$$A_{G_{\mathrm{Fib}}} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},\ A_{G_{\mathrm{Fib}}}^2 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix};$$

so that fact that $A_{G_{\mathrm{Fib}}}^2$ has top left entry 2 is a reflection of the fact that there are two walks from $v_1$ to $v_1$ of length 2. As an aside, we mention that by induction we can show that

$$A_{G_{\mathrm{Fib}}}^k = \begin{bmatrix} F_{k+1} & F_k \\ F_k & F_{k-1} \end{bmatrix},$$

where $F_k$ denote the $k$-th Fibonacci number.

[1] A *directed graph* allows one to have "multiple edges," meaning possible multiple edges associated to the same tuple $(v, v')$; hence one usually defines a directed graph to be a tuple $G = (V, E, t, h)$, where $V, E$ are sets — the *vertex set* and *edge set* — and $t, h$ are maps $E \to V$ — the *tails map* and *heads map*. Much of mathematics requires us to work with directed graphs, but when a directed graph is simple, one can merely regard $E$ as a subset of $V \times V$. This will suffice for our needs.



Forward Path of Length 3

Backward Path of Length 3

Bidirectional Path of Length 3

Path of Length 3

Figure 2. Path Digraphs and the Path Graph of Length 3

**Example 2.2.** Let $n \in \mathbb{N}$. By the *forward directed path of length* $n-1$ we mean the directed graph with vertex set $\{1, 2, \ldots, n\}$, and edge set $\{(1,2),(2,3),\ldots(n-1,n)\}$; the *backward directed path of length* $n-1$ we mean the directed graph with the same vertex set, but "opposite" edge set $\{(2,1),(3,2),\ldots,(n,n-1)\}$. By the *bidirectional path of length* $n-1$ we mean the directed graph with the same vertex set, but edge set that is the union of the forward and backward directed path of length $n-1$. See Figure 2.

A *simple graph*[2] is a directed graph $G = (V, E)$ such that for all $v, v' \in V$, $(v, v') \in E$ implies both (1) $(v', v) \in E$, and (2) $v \neq v'$. We typically depict a graph by drawing a single line segment (or curve) bewteen any pair of vertices $v, v'$ such that $(v, v') \in E$, rather than draw both an arrow from $v$ to $v'$ and another from $v'$ to $v$. We also typically denote the edges by unordered pairs, so the unordered pair $\{v, v'\}$ refers to the two directed edges $(v, v')$ and $(v', v)$.

**Example 2.3.** The bidirectional path of length 3 is a graph, and depicted in Figure 2, where each pair of arrows is replaced by a single line segment. More generally, the path of length $n-1$ is the graph $P_{n-1} = (V, E)$ where $V = \{1, 2, \ldots, n\}$, and $E = \{\{1,2\},\{2,3\},\ldots\{n-1,n\}\}$.

**Example 2.4.** Let $P_{n-1}$ be the path of length $n-1$, and $A_{P_{n-1}}$ its adjacency matrix. We easily check that $N_{\mathrm{rod},n} = A_{P_{n-1}}$. If $3 \leq i \leq n-2$, then there are four walks of length two from $i$: namely one to $i-2$ (namely $(i, i-1, i-2)$), one to $i+2$ (namely $(i, i+1, i+2)$), and two from $i$ to itself (namely $(i, i-1, i)$ and $(i, i+1, i)$). We similarly determine the number of walks of length two from the vertices 1, 2, $n-1$, and $n$. This gives a simple formula for $A^2_{P_{n-1}}$, and therefore for $N^2_{\mathrm{rod},n}$. One can similary determine $A^k_{P_{n-1}}$ for any fixed $k$ (although the first $k$ and last $k$ rows are a bit trickier to determine); hence this gives a concrete way to understand powers of $N_{\mathrm{rod},n}$.

## 3. Some Matrices of Interest

There is another way to understand powers of $N_{\mathrm{rod},n}$.

### 3.1. **Another Interpretation of $N_{\mathrm{rod},n}$.**

First, for any $n \in \mathbb{N}$, note that

$$N_{\mathrm{rod},n} = S_{n,1} + S_{n,-1},$$

[2] Similarly to simple directed graphs, in many mathematical settings simple graphs are inadequate for discussions; one typically wants to allow "multiple edges," and "self-loops," and at times the "self-loops" fall into two different types: "whole-loops," a pair of distinct directed self-loops, and a "half-loop," a single self-loop paired with itself.

where

$$S_{n,1}=\begin{bmatrix}0&1&0&0&\cdots&0&0&0&0\\0&0&1&0&\cdots&0&0&0&0\\0&0&0&1&\cdots&0&0&0&0\\0&0&0&0&\cdots&0&0&0&0\\\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots&\vdots\\0&0&0&0&\cdots&0&1&0&0\\0&0&0&0&\cdots&0&0&1&0\\0&0&0&0&\cdots&0&0&0&1\\0&0&0&0&\cdots&0&0&0&0\end{bmatrix},\quad S_{n,-1}=\begin{bmatrix}0&0&0&0&\cdots&0&0&0&0\\1&0&0&0&\cdots&0&0&0&0\\0&1&0&0&\cdots&0&0&0&0\\0&0&1&0&\cdots&0&0&0&0\\\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots&\vdots\\0&0&0&0&\cdots&0&0&0&0\\0&0&0&0&\cdots&1&0&0&0\\0&0&0&0&\cdots&0&1&0&0\\0&0&0&0&\cdots&0&0&1&0\end{bmatrix};$$

hence $S_{n,1}$ is the nonzero part of $N_{\mathrm{rod},n}$ that lies above the diagonoal, and $S_{n,-1}$ the part below. However, $S_{n,1}$ has a simple interpretation as "shifting up by one," in the sense that for any $\mathbf{x}=(x_1,x_2,\ldots,x_n)\in\mathbb{R}$ (which, as always, we think of as a column vector),

$$S_{n,1}\,\mathbf{x}=\begin{bmatrix}0&1&0&0&\cdots&0&0&0&0\\0&0&1&0&\cdots&0&0&0&0\\0&0&0&1&\cdots&0&0&0&0\\0&0&0&0&\cdots&0&0&0&0\\\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots&\vdots\\0&0&0&0&\cdots&0&1&0&0\\0&0&0&0&\cdots&0&0&1&0\\0&0&0&0&\cdots&0&0&0&1\\0&0&0&0&\cdots&0&0&0&0\end{bmatrix},\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\\vdots\\x_{n-3}\\x_{n-2}\\x_{n-1}\\x_n\end{bmatrix}=\begin{bmatrix}x_2\\x_3\\x_4\\x_5\\\vdots\\x_{n-2}\\x_{n-1}\\x_n\\0\end{bmatrix}$$

Hence the way that $S_{n,1}$ *operates* on a vector $\mathbf{x}$ is to move all its components up by one, and introduce a zero in the bottom component. Similarly we have

$$S_{n,-1}\,\mathbf{x}=\begin{bmatrix}0\\x_1\\x_2\\\vdots\\x_{n-3}\\x_{n-2}\\x_{n-1},\end{bmatrix}$$

so $S_{n,-1}\mathbf{x}$ *operates* by shifting the components of $\mathbf{x}$ down by one and introduces a 0 on the top.

**Remark 3.1.** Note that $S_{n,1}$ is the adjacency matrix of the forward path of length $n-1$ (see Figure 2 and Example 2.4). Similarly for $S_{n,-1}$ and the backward path of length $n-1$. This gives another way to understand $S_{n,\pm1}$ and their sum, $N_{\mathrm{rod},n}$. The only problem is that in graph theory and Markov chain theory (and symbolic dynamics, etc.), matrices typically act on row vectors (with the matrix to the right of the vector), so things look backward when we act on column vectors (with the matrix to the left of the vector), which is common elsewhere in linear algebra. So in graph theory and Markov chain theory one notes that

$$\begin{bmatrix}x_1&x_2&\cdots&x_n\end{bmatrix}S_{n,1}=\begin{bmatrix}0&x_1&\cdots&x_{n-1}\end{bmatrix},$$

which makes $S_{n,1}$ the "shift to the right by 1" when acting on row vectors, which more closely resembles is the same way that $S_{n,-1}$ acts on column vectors.

### 3.2. **Ring Matrices and Cyclic Shift Operators.**

There is a simple variant of $N_{\mathrm{rod},n}$ that is much easier to understand, namely:

$$N_{\mathrm{ring},n} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \end{bmatrix},$$

which is just the matrix $N_{\mathrm{rod},n}$ with a 1 added to the top right and to the bottom left corners. This matrix has each column sum and each row sum equal to 2; it also has a *cyclic symmetry* that makes it a *Toeplitz matrix* (see the Wikipedia page on *Toeplitz Matrix*); we may return to Toeplitz matrices later.

Working with ring matrices is much simpler, because they can be described as a sum of cyclic shift operators: indeed, for any $n \in \mathbb{N}$, note that

$$N_{\mathrm{ring},n} = C_{n,1} + C_{n,-1}, \tag{4}$$

where

$$C_{n,1} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix}, \quad C_{n,-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Similarly to the previous subsection, we have

$$C_{n,1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{n-1} \\ x_n \\ x_1, \end{bmatrix} \tag{5}$$

and hence $C_{n,1}$ has the effect of "cyclically rotating the components of $\mathbf{x}$ up by one," taking the $x_1$ to be its bottom component, instead of the 0 that $S_{n,1}$ introduces.

Similarly we have

$$C_{n,-1}\begin{bmatrix} x_1\\ x_2\\ x_3\\ \vdots\\ x_{n-2}\\ x_{n-1}\\ x_n \end{bmatrix} = \begin{bmatrix} x_n\\ x_1\\ x_2\\ \vdots\\ x_{n-3}\\ x_{n-2}\\ x_{n-1} \end{bmatrix} \tag{6}$$

3.3. **Ring Matrices and the $C_{n,\pm 1}$ are Easier to Work With than Rod Matrices and the $S_{n,\pm 1}$.** For many computations, it is easier to work with the $C_{n,\pm 1}$ than the $S_{n,\pm 1}$, and to see how powers and polynomials of

$$N_{\mathrm{ring},n} = C_{n,1} + C_{n,-1},$$

behave as opposed to

$$N_{\mathrm{rod},n} = S_{n,1} + S_{n,-1}.$$

For example, to interpret $N^2_{\mathrm{ring},n}$, we have

$$N^2_{\mathrm{ring},n} = \bigl(C_{n,1} + C_{n,-1}\bigr)^2$$

which equals

$$\bigl(C_{n,1} + C_{n,-1}\bigr)\bigl(C_{n,1} + C_{n,-1}\bigr) \tag{7}$$

To simiplify such an expression we note that

$$C^2_{n,1}\begin{bmatrix} x_1\\ x_2\\ x_3\\ \vdots\\ x_{n-2}\\ x_{n-1}\\ x_n \end{bmatrix} = \begin{bmatrix} x_3\\ x_4\\ x_5\\ \vdots\\ x_n\\ x_1\\ x_2, \end{bmatrix}$$

which just cyclically shifts the components of $\mathbf{x}$ by 2. More generally, if for any $k \in \mathbb{N}$, if we set

$$C_{n,k} = C^k_{n,1},$$

then $C_{n,k}$ is the operator that cyclically rotates the components of a vector up by $k$; similarly for $C_{n,-k} = C^k_{n,-1}$. We similarly see that $C_{n,-1}C_{n,1}\mathbf{x}$ is just $\mathbf{x}$, and hence

$$C_{n,-1}C_{n,1} = C_{n,1}C_{n,-1} = I_n$$

the identity matrix. Hence all the $C_{n,\pm k}$ are invertible, and they all commute; setting $C_{n,0} = I_n$ (which makes sense, in that shifting by 0 does nothing to a vector), we conclude that

$$\bigl(C_{n,1} + C_{n,-1}\bigr)^2 = C_{n,2} + 2I_n + C_{n,-2}.$$

Furthermore, this can be seen as a manifestation of the identity

$$\bigl(x + x^{-1}\bigr)^2 = x^2 + 2 + x^{-2}.$$

Similarly, for any $n, k \in \mathbb{N}$, the value of

$$\bigl(C_{n,1} + C_{n,-1}\bigr)^k$$

can inferred from that of

$$(\bigl(x+x^{-1}\bigr)^k,$$

which by the binomial theorem equals

$$x^{-k}\bigl(1+x^2\bigr)^k = x^{-k}\sum_{m=0}^{k}\binom{k}{m}x^{2m} = \sum_{m=0}^{k}\binom{k}{m}x^{2m-k}.$$

This calculation proves the following proposition.

**Proposition 3.2.** *For any $n,k\in\mathbb{N}$,*

$$N_{\mathrm{ring},n}^k = \sum_{m=0}^{k}\binom{k}{m}C_{n,2m-k}. \tag{8}$$

*In other words, $(i,j)$-th entry of $N_{\mathrm{ring},n}^k$ is $\binom{k}{m}$ if $i-j+k$ is even and $2m=i-j+k$, and otherwise $0$ (hence if $m\le -1$ or $m\ge k+1$ this entry is $0$).*

Now using adjacency matrices we easily get the following partial description of powers of $N_{\mathrm{rod},n}$.

**Corollary 3.3.** *For any $n,k,m\in\mathbb{N}$ with $0\le m\le k$, the $(i,i+k-2m)$-th entry of $N_{\mathrm{rod},n}^k$ is at most $\binom{k}{m}$, and equality holds provided $k+1\le i\le n-k$; and all other entries of $N_{\mathrm{rod},n}^k$ are $0$.*

The proof is to consider the *cycle of length $n$*, defined as the graph $C_n=(V,E)$ with

$$V=\{1,\dots,n\},\ E=\bigl\{\{1,2\},\dots,\{n-1,n\},\{n,1\}\bigr\},$$

which is just the path of length $n-1$ with one extra edge $\{n,1\}$. We easily see that $N_{\mathrm{ring},n}=A_{C_n}$, the adjacency matrix of $C_n$. We have the $i$-th rows of $A_{C_n}^k$ and $A_{P_{n-1}}^k$ are the same when all walks of length $k$ from $i$ in $C_n$ do not traverse the edge $\{n,1\}$.

3.4. **Commutators.** If $A,B$ are $n\times n$ matrices, the *commutator of $A$ and $B$* refers to the matrix

$$[A,B]=AB-BA$$

(hence $[B,A]=-[A,B]$). We have $[A,B]=0$ iff $AB=BA$ iff multiplication by $A$ and $B$ "commutes." It is useful to note that $[A,B]$ is "bilinear in $A$ and in $B$," in the sense that

$$[A_1+A_2,B_1+B_2]=[A_1,B_1]+[A_1,B_2]+[A_2,B_1]+[A_2,B_2].$$

Hence, with $C_{n,k}$ as above, $[C_{n,1},C_{n,-1}]=0$; this commutation shows that

$$N_{\mathrm{ring},n}^k=\sum_{m=0}^{k}\binom{k}{m}C_{n,-1}^{m}C_{n,1}^{k-m},$$

which leads to (8). Note that $C_{n,1}=S_{n,1}+L_n$ where $L_n$ is the matrix with a single nonzero entry in its lower left entry (equal to 1), and similarly $S_{n,-1}=C_{n,-1}+R_n$ where $R_n=(L_n)^{\mathrm{T}}$ is defined similarly. However, one cannot write $N_{\mathrm{rod},n}^k$ as

$$\sum_{m=0}^{k}\binom{k}{m}S_{n,-1}^{m}S_{n,1}^{k-m},$$

since the commutator

$$[S_{n,1}, S_{n,-1}] = [C_{n,1} - L_n, C_{n,-1} - R_n]$$

is not 0; in fact the above commutator equals

$$[C_{n,1}, C_{n,-1}] - [C_{n,1}, R_n] - [L_n, C_{n,-1}] + [L_n, R_n],$$

and we easily see that $[C_{n,1}, R_n] = [L_n, C_{n,-1}]$, and hence the above expression equals $[L_n, R_n]$ (which has a 1 in the top right entry, a $-1$ in the bottom left). Hence $S_{n,1}$ and $S_{n-1}$ "almost commute," but our formula for $N^k_{\mathrm{rod},n}$ has corrections in the top $k$ and bottom $k$ rows (or columns).

**Example 3.4.** The "Heisenberg uncertainty principle" is often stated as arising from the fact

$$\left[\frac{d}{dx}, x\right] = 1,$$

in the sense that for any differentiable $f\colon \mathbb{R} \to \mathbb{R}$ we have

$$\left[\frac{d}{dx}, x\right] f = \frac{d}{dx}(xf) - x\frac{d}{dx}f = f + x\frac{df}{dx} - x\frac{df}{dx} = f.$$

To write this for polynomials of degree at most 4, we consider any such polynomial as a polynomial of degree at most 5, and note that in usual monomial basis, multiplication by $x$ takes $x^4$ to $x^5$, $x^3$ to $x^4$, etc., and hence is represented by $S_{6,1}$ with respect to the monomial basis $x^5, x^4, \ldots, x, 1$. Moreover differentiation of polynomials of degree at most 5 takes $x^5$ to $5x^4$, $x^4$ to $4x^3$, etc., and hence is represented by

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

We easily verify that $[D, S_{6,1}]$ is the identity matrix except for having its top left entry equal to 0.

## 4. The Heat Equation

One of the most fundamental partial differential equations is the heat equation.

### 4.1. Understanding the One-Dimensional Heat Equation.

We imagine a function $u = u(x,t)$ of two real variables, where $x \in \mathbb{R}$ represents a space variable, $t \in \mathbb{R}$ represents time. Because $x \in \mathbb{R}$ and not $\mathbb{R}^3$, if you are thinking in terms of three dimensions, you can imagine an "infinitely thin" rod or wire (which is insulated, so that it doesn't lose any heat in the middle). For simplicity, we assume:

(1) the rod's endpoints at $x = 0$ and $x = 1$ are held at a constant temperature 1; hence $u(0,t) = u(1,t) = 0$ for all $t > 0$;

(2) the rod's initial temperature profile at $t = 0$, i.e., $f(x) = u(x,0)$, is a known function $f\colon (0,1) \to \mathbb{R}$; and

(3) the rod is composed of a single material of thermal conductivity $c > 0$, and hence the classical heat equation is

$$\text{for all } 0 < x < 1 \text{ and } t > 0, \quad \frac{\partial}{\partial t}u(x,t) = c\frac{\partial^2}{\partial x^2}u(x,t) \tag{9}$$

(however, the above equation may not necessarily make much intuitive sense as written).

In (9), the $\partial$ symbol means that we take "partial derivatives," e.g., $(\partial/\partial t)u(x,t)$ mean that we hold $x$ fixed and differentiate with respect to $t$, i.e.,

$$\frac{\partial}{\partial t}u(x,t) = \lim_{H\to 0}\frac{u(x,t+H)-u(x,t)}{H}.$$

We often use $u_t$ to denote $\frac{\partial}{\partial t}u(x,t)$ and similarly for $u_x$ and $u_{xx}$ (the second partial derivative in $x$), so that we write (9) as

$$\text{for all } 0 < x < 1 \text{ and } t > 0, \quad u_t(x,t) = cu_{xx}(x,t). \tag{10}$$

**Example 4.1.** The function $u(x,t) = \sin(\pi x)e^{-tc\pi^2}$ solves the heat equation $u_t = cu_{xx}$ and satisfies the boundary conditions $u(0,t) = u(1,t) = 0$. In this example, as $t \to \infty$, the function $u(x,t)$ decays exponentially in $t$. It turns out that exponential decay always holds, although this is not obvious from just looking at $u_t = cu_{xx}$.

To us, (10) is much easier to understand by applying discrete approximations: for small $h, H > 0$, Taylor's theorem implies that for any $0 < x < 1$ and $t > 0$ and small $h, H > 0$,

$$\frac{u(x,t+H)-u(x,t)}{H} = u_t(x,t) + O(H)$$

and

$$\frac{u(x+h,t)+u(x-h,t)-2u(x,t)}{h^2} = u_{xx}(x,t) + O(h^4),$$

and hence $u_t = cu_{xx}$ approximately (to within the $O(H), O(h^2)$ terms):

$$\frac{u(x,t+H)-u(x,t)}{k} \approx u_t(x,t) = cu_{xx}(x,t) \approx (c)\frac{u(x+h,t)+u(x-h,t)-2u(x,t)}{h^2} \tag{11}$$

and hence

$$u(x,t+H) \approx u(x,t) + \frac{cH}{h^2}\Big(u(x+h,t)+u(x-h,t)-2u(x,t)\Big)$$

and so

$$u(x,t+H) \approx u(x,t) + 2\rho\left(\frac{u(x+h,t)+u(x-h,t)}{2} - u(x,t)\right), \text{ where } \rho = \frac{cH}{h^2} \tag{12}$$

This equation should make sense: if at a fixed time, $t$, the average temperature of your neighbours is higher than yours, then your temperature at time $t + H$ should be slightly higher than your temperature at time $t$, and similarly if the average temperature is the same or lower.

4.2. **Numerical Solution of the Heat Equation.** To solve the heat equation $u_t = cu_{xx}$ subject to the "boundary conditions," above, namely $u(0,t) = u(1,t) = 0$ and $u(x,0) = f(x)$ given, we subdivide the interval $[0,1]$ into equally spaced points $0 = x_0, x_1, \ldots, x_m = 1$, so that $h = 1/m = x_{i+1} - x_i$ is independent of $i$, and we consider the equally spaced times $0 = t_0 < t_1 < t_2 < \ldots$, so that $H = t_{j+1} - t_j$ is independent of $j$. Then for all $0 \le i \le m$ we set $U(i,j) = U(h,H;i,j)$ as an approximation to $u(ih, jH)$; hence we set $U(i,0) = u(i/m,0) = f(i/m)$; for $j = 0,1,2,\ldots$ (12) suggests the approximation:

$$\begin{bmatrix} U(1,j+1) \\ U(2,j+1) \\ \vdots \\ U(m-1,j+1) \end{bmatrix} = \Bigl(I(1-2\rho) + \rho N_{\mathrm{rod},n}\Bigr) \begin{bmatrix} U(1,j) \\ U(2,j) \\ \vdots \\ U(m-1,j), \end{bmatrix}$$

or, in shorthand,

$$\mathbf{U}(\,\cdot\,,j+1) = \Bigl(I(1-2\rho) + \rho N_{\mathrm{rod},m-1}\Bigr)\mathbf{U}(\,\cdot\,,j), \tag{13}$$

where

$$\mathbf{U}(\,\cdot\,,j) = \begin{bmatrix} U(1,j) \\ U(2,j) \\ \vdots \\ U(m-1,j). \end{bmatrix}$$

**Remark 4.2.** Since $N_{\mathrm{rod},m-1} = A_{P_{m-2}}$, the adjacency matrix of the path of length $m-2$, one can understand the above equation purely graph theoretically. In fact, the above motivates the usual definition of the so-called "Laplacian" of a graph.

4.3. **Some Exact Solutions.** One way to test the above numerical approximation is against exact solutions.

**Example 4.3.** Let $s \in \mathbb{N}$, and $u(x,t) = sin(s\pi x)e^{-cs^2\pi^2 t}$. Then we easily check that $u(0,t) = u(1,t) = 0$ for all $t$, $u(x,0) = \sin(s\pi x)$, and $u_t = cu_{xx}$ for all $x,t \in \mathbb{R}$.

**Example 4.4.** *Fourier analysis* lets us write any function $f\colon [0,1] \to \mathbb{R}$ (under mild conditions, such as $f \in C^0[0,1]$, i.e., $f$ is continuous on $[0,1]$)[3] as an infinite sum

$$f(x) = \sum_{s=1}^{\infty} a_s \sin(s\pi x)$$

(where $\sum_s a_s^2 = (1/2)\int_0^1 f^2(x)\,dx < \infty$)[4]. Using the previous examples, we solve $u_t = cu_{xx}$ subject to $u(0,t) = u(1,t) = 0$ and $u(x,0) = f(x)$ as

$$u(x,t) = \sum_{s=1}^{\infty} a_s \sin(s\pi x)e^{-cs^2\pi^2 t}. \tag{14}$$

Notice that even if the $a_s$ do not decay very fast, it is easy to see that for each $t > 0$, $u(x,t)$ has quickly decaying Fourier coefficients: indeed, we have that $\sum_s a_s^2$

[3] More generally it suffices to have $f$ measurable and $\int_0^1 f^2(x)\,dx < \infty$.

[4] In particular, it turns out that $a_s = 2\int_0^1 f(x)\sin(s\pi x)\,dx$. However, things can get a bit subtle... For example, if $f(x) = 1$ for all $0 \le x \le 1$, then $a_s = 2/(s\pi)$ for $s$ odd, and $a_s = 0$ for $s$ even, and it may not be clear in what sense the infinite sum $\sum_s a_s \sin(s\pi x)$ converges...

is finite, and hence $|a_s| \le C$ for all $s$, for some constant, $C$. Since for any $t > 0$

$$u(x,t) = \sum_{s=1}^{\infty} A_s \sin(s\pi x), \quad \text{where} \quad A_s = a_s e^{-cs^2\pi^2 t},$$

the Fourier coefficients of $u(x,t)$, with $t$ fixed, satisfy

$$|A_s| \le Ce^{-C'(t)s^2},$$

where $C'(t) = c\pi^2 t$. Hence the $|A_s|$ decay exponentially in $s^2$.

4.4. **The Weak Maximum Principle.** The weak maximum principle gives some insight into solutions of the heat equation, that should be intuitive but that is not apparent from its "exact solution" (14).

First, fix some real $c, T > 0$, and say that a continous function $u = u(x,t)$ defined for $0 \le x \le 1$ and $0 \le t \le T$ satisfies the heat equation $u_t(x,t) = cu_{xx}(x,t)$ for all $0 < x < 1$ and $0 < t \le T$.[5] We define the *time $T$ boundary* of this heat equation to be

$$B_T = \Bigl([0,1] \times \{0\}\Bigr) \cup \Bigl(\{0,1\} \times [0,T]\Bigr) \tag{15}$$
$$= \Bigl\{(x,t) \;\Bigm|\; 0 \le x \le 1 \text{ and } t = 0, \text{ or } x = 0,1 \text{ and } 0 \le t \le T\Bigr\}$$

This is because the temperature $u(x,T)$, for any $0 < x < 1$, should depend only on its boundary conditions, meaning its initial temperature, $u(x,0)$, and the temperature on its endpoints $x = 0,1$ for times between $0$ and $T$. (Notice that $(0,1) \times T$ is not considered part of the "boundary," at least for the heat equation, even though these points are part of the boundary of the rectangle $[0,1] \times [0,T]$.)

**Proposition 4.5** (The Weak Maximum Principle)**.** *Say that for some reals $T, c > 0$, there is a continuous function $u\colon [0,1] \times [0,T] \to \mathbb{R}$ that satisfies $u_t = cu_{xx}$ for all $0 < x < 1$ and $0 < t \le T$. let $B_T$ be as in (15). Let*

$$M = \max_{(x,t)\in B_T} u(x,0).$$

*Then for all $(x,t) \in [0,1] \times [0,T]$ we have $u(x,t) \le M$.*

It should make intuitive sense that if the initial temperature and endpoint temperatures of a rod (up to any time $T$) are all at most $M$, then the temperature in the interior of the rod (up to time $T$) should be at most $M$. Since $-u$ satisfies the same heat equation, the above proposition implies that if

$$M' = \min_{(x,t)\in B} u(x,0),$$

then similarly $u(x,t) \ge M'$ throughout the rectangle $[0,1] \times [0,T]$.

*Proof.* If not, then $u(x,t) > M$ for some $(x,t) \in (0,1) \times (0,T]$. It follows that for $\epsilon > 0$ sufficiently small,

$$v(x,t) = u(x,t) + \epsilon\bigl(x^2 - t\bigr)$$

has a value that is larger than its maximum over $B_T$; pick such an $\epsilon > 0$. It follows that the maximum of $v$ on $[0,1] \times [0,T]$ is attained at a non-boundary point, $(x_0, t_0)$, and hence $0 < x_0 < 1$ and $0 < t_0 \le T$. Since $v(x_0,t) \le v(x_0,t_0)$

[5] To make sense of $u_t = cu_{xx}$ holding at $0 < x < 1$ and $t = T$, we define $u_t(x,T)$ to be the left partial derivative in $t$ of $u$, i.e., based on the value of $u(x,t)$ with $t \le T$.

for $t < t_0$, we have $v_t(x_0, t_0) \geq 0$. Since $v(x, t_0)$ has a local maximum at $x = x_0$, we have $v_{xx}(x_0, t_0) \leq 0$ (as well as $v_x(x_0, t_0) = 0$, which we don't need). Hence at $(x_0, t_0)$ we have

$$v_t - cv_{xx} \geq 0. \tag{16}$$

But we easily see that

$$v_t = u_t - \epsilon, \quad v_{xx} = u_{xx} + 2\epsilon,$$

and so

$$v_t - cv_{xx} = u_t - cu_{xx} - \epsilon - 2c\epsilon = -(1 + 2c)\epsilon < 0,$$

which contradicts (16). □

We can get far more insight into the heat equation with some extra work. In particular we can prove the *strong maximum principle*, which states that in the above proposition, if $u(x, t) = M$ for any $0 < x < 1$ and any $0 < t \leq T$, then $u(x, t) = M$ throughout $[0, 1] \times [0, T]$.

4.5. **Stability and Instability of Numerical Schemes.** When you solve the heat equation, presumably you want to take $H, h \to 0$ and apply (13) or (12) with $\rho = cH/h^2$ to approximate the solution. You might guess that as $h, H \to 0$, it is not a good idea to choose $\rho = cH/h^2$ in (13) or (12) to be large. Namely, since

$$\|I(1 - 2\rho) + \rho N_{\mathrm{rod},m-1}\|_\infty = |1 - 2\rho| + 2\rho,$$

if $\rho > 1/2$, then this matrix has norm $> 1$, and you might expect bad things to happen as $t \to \infty$ in the numerical approximation, i.e., as you take successively higher powers of $I(1 - 2\rho) + \rho N_{\mathrm{rod},m-1}$ applied to the vector $U(\,\cdot\,, 0)$ representing the time $t = 0$ values of the temperature.

**Example 4.6.** For $\rho = 1$, we have $|1 - 2\rho| + 2\rho = 3$. Hence the $k$-th power of

$$I(1 - 2\rho) + \rho N_{\mathrm{rod},m-1} \tag{17}$$

should amplify relative errors by at worst $3^k$. Since double precision has relative error at worst (roughly) $2^{-53}$, then when $3^k 2^{-53}$ is small, then we shouldn't expect relative errors to be too bad. So consider the equation $u_t = u_{xx}$ for $0 < x < 1$, $t > 0$, subject to boundary conditions

$$u(0, t) = u(1, t) = 0, \quad u(x, 0) = sin(\pi x).$$

Consider $h = 0.1 = 1/10$, so $H = \rho h^2 = 1/100$. Since $3^k = 2^{53}$ for $k = 53 \log(2)/\log(3) = 33.4392...$, we might expect some trouble with double precision at around 33 or 34 iterations of (17), which corresponds when $t$ is roughly $33H = 0.33$ or $34H = 0.34$. Indeed, this seems to be (roughly) the case; see Figure 3.

We have included the MATLAB code we used to generate the plots of Figure 3 in the file `sine_intially.m` in Appendix A.

For a fuller discussion of the above, see *Lectures on Advanced Numerical Analysis*, by Fritz John.

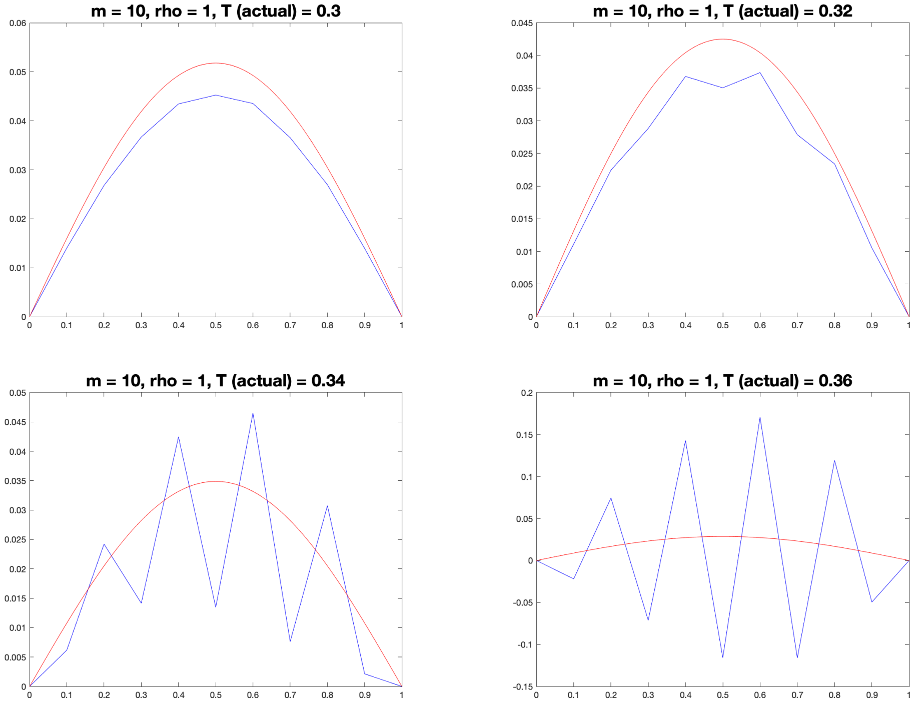


FIGURE 3. The Approximation to $u_t = u_{xx}$ with $m = 10$, $\rho = 1$

4.6. **A Higher Order Heat Equation Approximation.** It is not hard to guess which value of $\rho$ will give the best results, when taking $h, H \to 0$ in the above numerical schemes: by Taylor's theorem we have

$$u(x+h,t) - 2u(x,t) + u(x-h,t) = h^2 u_{xx}(x,t) + (h^4/12)u_{xxxx}(x,t) + O(h^6)$$

assuming that $u$ is sufficiently differentiable, and

$$u(x,t+H) - u(x,t) = Hu_t(x,t) + (H^2/2)u_{tt}(x,t) + O(H^3).$$

Moreover the equation $u_t = cu_{xx}$, for $u$ sufficiently differentiable, implies

$$u_{tt} = c\big(u_{xx}\big)_t = c\big(u_t\big)_{xx} = c\big(cu_{xx}\big)_{xx} = c^2 u_{xxxx}.$$

It follows that

$$\frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2} = u_{xx}(x,t) + (h^2/12)u_{xxxx}(x,t) + O(h^4),$$

and

$$\frac{u(x,t+H) - u(x,t)}{H} = u_t(x,t) + (H/2)u_{tt}(x,t) + O(H^2) = cu_x x(x,t) + (H/2)c^2 u_{xxxx}(x,t) + O(H^2).$$

Hence

$$c\frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2} - \frac{u(x,t+H) - u(x,t)}{H} = c(h^2/12 - c(H/2))u_{xxxx}(x,t) + O(h^4 + H^2).$$

Hence the approximation (11), which sets the left-hand-side of the above to being $\approx 0$, becomes a higher order scheme when $h^2/12 = cH/2$, i.e., when $\rho = cH/h^2 = 1/6$.

4.7. **Gaussians and the Heat Equation.** We mention, in passing, the relation between Gaussian distributions and the heat equation.

One message about solving for the heat equation is that for numerically stable solutions, we need to take $H = \rho h^2$, so the time step, $H$, is of order $h^2$, which is much smaller than $h$ as $h \to 0$. There is a reason for this: if we look at

$$N^k_{\mathrm{ring},n} = \sum_{m=0}^{k} \binom{k}{m} C^m_{n,-1} C^{k-m}_{n,1},$$

we see that although this matrix is nonzero at the $(i,j)$-th entries with $|i-j| \ge k+1$, this matrix is concentrated in a much smaller range, namely for $|i-j|$ of size roughly $O(\sqrt{k})$ rather than $k$. One can make this precise using the "central limit theorem," which implies that the numbers $\binom{k}{m}/2^k$ (whose sum equals 1, since we divide by $k$) look like a Guassian distribution where, in rough terms, $m$ is within order $\sqrt{k}$ of $k/2$.

Another way to see Gaussian's arise from the heat equation is that if one considers $u_t = cu_{xx}$ where $x \in \mathbb{R}$, one solution to this equation is

$$u(x,t) = \frac{1}{\sqrt{4\pi ct}} e^{-x^2/(4ct)}.$$

This solution tends to the "Dirac delta function" as $t \to 0$, and it follows that if $u(x,0) = f(x)$, where $f(x)$ satisfies some appropriate boundedness conditions (for $|x|$ large), then there is a solution

$$u(x,t) = \int_{y=-\infty}^{y=\infty} f(y) \frac{1}{\sqrt{4\pi ct}} e^{-(x-y)^2/(4ct)}\, dy, \tag{18}$$

and that solution is the unique solution that satisfies appropriate boundedness conditions.

Yet another connection to Gaussians is when we solve the above heat equation, either for $x \in (0,1)$ or all $x \in \mathbb{R}$, by the use of a *stochastic process*, specifically a *Brownian motion*. Roughly speaking, (18) can be viewed as saying that to compute $u(x,T)$ we start a Brownian motion (depending on $c$) from $(x,t)$, running it "backwards in time," yielding a function $B_\omega \colon [0,T] \to \mathbb{R}$; with $B_\omega(0) = x$; for the appropriate Brownian motion (depending on $c$), we have

$$u(x,T) = \mathbb{E}_\omega f(B_\omega(T)) = \mathbb{E}_\omega u(B_\omega(T), 0).$$

Similarly, if we solve the heat equation over $0 < x < 1$, then get a similar formula, where we stop the Brownian motion $B_\omega(t)$ at the first time (if it exists) $t = t_0$ such that $B_\omega(t_0) = 0, 1$, and for such $\omega$ we substitute $u(B_\omega(t_0), T - t_0)$ (the temperature at an endpoint) for $f(B_\omega(T)) = u(B_\omega(T), 0)$. For this stopping time $t_0 = t_0(\omega)$ (where $t_0(\omega) = T$ if $0 < B_\omega(t) < 1$ for all $t \le T$), we have the more general formula

$$u(x,T) = \mathbb{E}_\omega u\Bigl(B_\omega\bigl(t_0(\omega)\bigr), T - t_0(\omega)\Bigr).$$

4.8. **Rods of Varying Thermal Conductivity.** If a rod is made of materials that vary in $x$, then the classical heat equation is

$$u_t = (cu_x)_x = c'(x)u_x + c(x)u_{xx}$$

where the thermal conductivity, $c = c(x)$, is now a function of $x$. One can design an approximation scheme in this case; one might use a centered scheme for $u_x$, i.e., $u_x(x,t) \approx (u(x+h,t) - u(x-h,t))/2h$, since this has error $O(h^2)$ and the three-point approximation for $u_{xx}$ already uses these values of $u$.

Another case—which is easy to program—is where $c(x)$ is piecewise constant, and the discontinuities of $c(x)$ occur at a few grid points, say that $x_i$ for a few values of $1 \le i \le m-1$. When $c(x)$ is discontinuous at $x = x_i$, one imposes that $c(x)u_x(x,t)$ is continuous across $x = x_i$, in other words this value on the left, namely $c(x-)u_x(x-,t)$, must equal this value on the right, namely $c(x+)u_x(x+,t)$. In other words, if $c(x)$ for $x < x_i$ and near $x_i$ is $c_1$, and is $c_2$ for $x$ near $x_i$ with $x > x_i$, then we insist that

$$c_1 u_x(x_i-,t) = c_2 u_x(x_i+,t).$$

Hence we could numerically we should impose

$$c_1\bigl(u(x_i,t) - u(x_i-h,t)\bigr) = c_2\bigl(u(x_i+h,t) - u(x_i,t)\bigr),$$

and so impose

$$u(x_i,t) = \frac{c_1 u(x_i-h,t) + c_2 u(x_i+h,t)}{c_1+c_2};$$

if $x_{i\pm 1} = x_i \pm h$, then we impose

$$u(x_i,t) = \frac{c_1 u(x_{i-1},t) + c_2 u(x_{i+1},t)}{c_1+c_2}. \tag{19}$$

Hence to numerically update $u(\,\cdot\,,t+H)$ from $u(\,\cdot\,,t)$, assuming that $c(x)$ is piecewise constant, between grid points, and that these grid points aren't consective, we can use the usual method away from the discontinuities in $c(x)$, i.e.,

$$u(x_i,t+H) = u(x_i,t)(1-2\rho_i) + \rho_i\bigl(u(x_{i+1},t) + u(x_{i-1},t)\bigr) \tag{20}$$

whereever $c(x_{i-1}) = c(x_i) = c(x_{i+1})$, with $\rho_i = c(x_i)H/h^2$, and afterwards, when $c(x)$ is discontinuous at $x = x_i$, we set

$$u(x_i,t+H) = \frac{c_1 u(x_{i-1},t+H) + c_2 u(x_{i+1},t+H)}{c_1+c_2}, \quad c_1 = c(x_{i-1}),\ c_2 = c(x_{i+1}) \tag{21}$$

(since $u(x_{i\pm 1},t+H)$ will be been determined earlier, given that $c(x)$ is continuous at $x_{i\pm 1}$, i.e., we don't have two discontinuities at consecutive grid points).

[Exercise: Design a numerical experiment that tests this scheme numerically versus one where $c(x)$ is a smooth approximation of a piecewise-constant function.]

## Exercises (Preliminary Draft)

(1) Consider the numerical approximation to the heat equation

$$u_t = u_{xx}, \quad 0 < x < 1,\ t > 0$$

subject to

$$u(0,t) = u(1,t) = 0, \quad u(x,0) = \sin(\pi x).$$

[Hence the exact solution is $u(x,t) = \sin(\pi x)e^{-\pi^2 t}$.] In Example 4.6, it was shown that with $m = 10$ and $\rho = 1$ (hence $h = 1/m$, $H = \rho h^2 = \rho m^2$), using (13) to approximate $u(x,T)$ for all $x \in [0,1]$ becomes problematic around $T = 0.34$. (For these exercises you can use the software in Appendix A if you like.)

(a) Let $\rho = 1$ and $m = 20$. Mimick Example 4.6 in this case: plot some value of $T$ for which the computation of $u(x,T)$ becomes problematic due to errors in double precision. If the number of iterations is $k$, how does $(|1-2\rho| + 2\rho)^k$ compare with $2^{53}$?

(b) Same problem as (a), for $\rho = 2/3$ and $m = 10$.

(c) Let $m = 10$, $\rho = 1/3, 1/4, 1/6, 1/8, 1/10$, and $T = 1$. How does the exact solution of $u(x,T) = u(x,1)$ compare with the numerical solution? For which values of $\rho$ is the numerical solution smaller, and for which is it bigger?

(2) Consider

$$u_t = u_{xx}, \quad 0 < x < 1,\ t > 0$$

subject to

$$u(0,t) = u(1,t) = 0, \quad u(x,0) = \sin(\pi x).$$

[Hence the exact solution is $u(x,t) = \sin(\pi x)e^{-\pi^2 t}$.] Notice that standard trigonometric identities show that

$$\sin(\pi x + \pi h) + \sin(\pi x - \pi h) = 2\cos(\pi h)\sin(\pi x).$$

(a) Show that in exact computation, if $U(i,0) = \sin(i\pi/m)$, then for any $\rho > 0$, in (13) we have

$$\mathbf{U}(\,\cdot\,,j) = \Bigl(1 + 2\rho\bigl(\cos(\pi h) - 1\bigr)\Bigr)^j \mathbf{U}(\,\cdot\,,0)$$

(b) Consider the approximation one gets for $u(x,1)$ with $0 \le x \le 1$ with the above scheme. One therefore takes $H = \rho h^2$, and, assuming that $H = 1/M$ for some integer, $M$, we get the approximation to $u(x,1)$ for $x = 1/m, 2/m, \ldots, (m-1)/m$ to be

$$\mathbf{U}(\,\cdot\,,1/H) = \Bigl(1 + 2\rho\bigl(\cos(\pi h) - 1\bigr)\Bigr)^{1/(\rho h^2)} \mathbf{U}(\,\cdot\,,0).$$

Show that for $\rho$ fixed, as $h \to 0$,

$$\Bigl(1 + 2\rho\bigl(\cos(\pi h) - 1\bigr)\Bigr)^{1/(\rho h^2)} = e^{-\pi^2} + O(h^2).$$

(c) Find the function of $\rho$, $g(\rho)$, such that

$$\Bigl(1 + 2\rho\bigl(\cos(\pi h) - 1\bigr)\Bigr)^{1/(\rho h^2)} = e^{-\pi^2 + g(\rho)h^2 + O(h^4)}.$$

For what values of $\rho$ is $g(\rho)$ positive? Negative? Zero? (This may explain your answer to Exercise 1(d).)

(d) Show does the $\rho$ where $g(\rho) = 0$ compare with the $\rho$ found in Subsection 4.6 that gives a higher order method?

(3) Consider two different rods of length 1, each made of 50% chewing gum and 50% metal,[6] where

(a) the first rod has chewing gum on the outside, and metal on the inside; model this by setting
$$c_1(x) = \begin{cases} 2 & \text{if } 1/4 \le x \le 3/4, \text{ and} \\ 1 & \text{otherwise (i.e., } 0 \le x \le 1/4 \text{ or } 3/4 \le x \le 1); \end{cases}$$
and

(b) the second rod has metal on the outside, and chewing gum on the inside; model this by setting
$$c_2(x) = \begin{cases} 1 & \text{if } 1/4 \le x \le 3/4, \text{ and} \\ 2 & \text{otherwise (i.e., } 0 \le x \le 1/4 \text{ or } 3/4 \le x \le 1). \end{cases}$$

Plot the temperature profile of the two rods against each other, i.e., of $u(x,T)$ for various values of $T$, taking $h, H \to 0$ in such a way that $\rho_{\max} = 2h/H^2$ is less than $1/2$. Is one rod always warmer than the other in the middle, i.e., at $x = 1/2$ ?

(4) Exercise 4 will appear here.

(5) Exercise 5 will appear here.

## Appendix A. Some MATLAB Code

This section contains some MATLAB code to solve the heat equation $u_t = u_{xx}$ on $0 < x < 1$ and $t > 0$, where $u(x,0) = \sin(\pi x)$ and $u(0,t) = u(1,t) = 0$.

```
% April 3, 2024: Experiment on initial condition sin(pi x)
% Joel Friedman, CPSC 303, UBC

% the following function numerically approximates a solution to:
%   u_t = u_xx   for 0 < x < 1, and t>0
% subject to:
%   u(0,t)=u(1,t)=0, and u(x,0) = sin( pi x )
%
% It takes three inputs: m (an integer), and rho,T (two positive reals)
%
% It uses the standard way to solve the heat equation (see the course handout),
% using equally spaced grid points 0 = x_0, x_1, ..., x_m = 1,
%   (so h = x_{i+1}-x_i = 1/m for all i, and x_i = i/m)
% and equally spaced time grid points 0 = t_0,t_1,..., where
%   H = t_{i+1}-t_i satisfies  rho = H/h^2, so H = rho m^2
%
% Hence U(i,j) approximates u( (i-1) h , (j-1) H ), is given by
%   U(i,j+1) = (1-2 rho) U(i,j) + rho ( U(i+1,j) + U(i-1,j) )
%
```

[6] This experiment was a high school science project of mine supervised by Mr. Robert Bruce Horton, Evanston Township High School, with encouragement from my dad. The results of this experiment can be proven rigorously by rescaling the $x$ variable, so that each heat profile satisfies $u_t = u_{xx}$ with two conditions at the material interfaces; one compares each to $u_t = u_{xx}$ (without conditions), and scaling back. For $c_1 = 1$ and $c_2 = 2$, there is therefore no scaling at $x$ with $c(x) = 1$, and one scales $x$ by $\sqrt{2}$ where $c(x) = 2$; numerically one can observe this, noting that the ratio of the two profiles near $x = 0$ for small time is roughly $\sqrt{2}$.

```
% or, equivalently, we iterate on:
%
%   U_new(i) = (1-2 rho) U_curr(i) + rho ( U_curr(i+1) + U_curr(i-1) )
%     for i=2,...,m-1
%   U_new(1) = U_new(m+1) = 0
%
% and then set U_curr to U_new
%
% this function does two things: (1) it plots u(.,T) as a broken line, based
% on the values u(x_0,T), u(x_1,T), u(x_2,T), ... , u(x_m,T) in blue, with
% the exact solution in red, and (2) it returns the vector of u values above.

% Note that if T is not an integer multiple of H, it reports these values for
% T replaced by  H floor(T/H), i.e., the largest multiple of H less than T

function U_curr = first_sine(m,rho,T) % run heat eq u_t = u_xx, on [0,1]
                                 % with sin(pi x) as initial cond

% Hence the time grid points are 0,H,2H,3H,... where H = rho / m^2
%
% In case T/H = T m^2 / rho is not an integer, we give

time_iters  = floor(T * m^2 / rho)   % the number of iterations
T_actual = time_iters * rho / m^2    % the actual time we stop


% Here are the vectors for u(x_0,iH), ... , u(x_m,iH)

U_curr = zeros(1,m+1);
x = zeros(1,m+1);

for i=1:m+1
  U_curr(i) = sin( (i-1) * pi / m);
  x(i) = (i-1)/m;
end

% These vectors will be used to plot the exact solution

xfine = zeros(1,1001);
U_actual = zeros(1,1001);
for i=1:1001
  U_actual(i) = sin( (i-1) * pi / 1000); % this is the initial condition
  xfine(i) = (i-1)/1000;
end

U_actual = U_actual * exp(-T_actual * pi^2);  % The exact u(x_i,T_actual)

% This sets u(1)=u(m+1)=0, which should be done regardless of the initial
%   condition u

U_curr(1) = 0; U_curr(m+1)=0;

% This is the vector used to compute the new values

U_new = zeros(1,m+1);

for j= 1 : time_iters
  U_new(1)=0; U_new(m+1)=0;
  for i=1:m-1
    U_new(i+1) = (1 - 2 * rho) * U_curr(i+1) + rho * U_curr(i) + rho * U_curr(i+2);
  end
  U_curr = U_new;
end

hold off;
plot(x,U_curr,’Color’,’blue’);
hold on;
plot(xfine,U_actual,’Color’,’red’);
title([ ’m = ’ num2str(m), ’, rho = ’ num2str(rho), ’, T (actual) = ’ num2str(T_actual) ] , ’FontSize’, 20);
```

Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, CANADA.

*E-mail address*: jf@cs.ubc.ca

*URL*: http://www.cs.ubc.ca/~jf