(Thermal) Mechanics:
\[
\rho C_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + q'' + \left\{ \begin{array}{l} \text{BC} \\ \text{IC} \end{array} \right. \\
\]

- For \( k = \text{constant} \) (big assumption),
\[
\frac{\partial T}{\partial t} = \frac{k}{\rho C_p} \frac{\partial^2 T}{\partial x^2} + \frac{q''}{\rho C_p} + \left\{ \begin{array}{l} \text{BC} \\ \text{IC} \end{array} \right. \\
\]
\[
= \alpha \frac{\partial^2 T}{\partial x^2} + \tilde{q}'' \\
\]

- Consider \( q'' = 0 \), let \( u := T, u(0) = u(1) = 0 \) (or, more generally, \( u(L) = 0 \)).
Separation of Variables:

- Neglect I.C. for now.
- Just suppose
  \[ u(x, t) = c(t) s(x). \]
- Consider \( s(x) = \sin(k \pi x) \).
- Satisfies BCs: \( u(0) = u(1) = 0 \).
- Insert into PDE:
  \[
  \frac{\partial u}{\partial t} = \frac{dc}{dt} s(x) = \alpha c(t) s''(x) = \alpha \frac{\partial^2 u}{\partial x^2} \\
  = -\alpha k^2 \pi^2 c(t) s(x)
  \]
• Time coefficient satisfies:
\[
\frac{dc}{dt} = -\alpha k^2 \pi^2 c(t)
\]
\[\Rightarrow c(t) = c^0 e^{-k^2 \pi^2 \alpha t}.\]

• Denote this solution as
\[
\hat{u}_k(x, t) = c_k(t) s_k(x) = c_k(t) \sin(k\pi x)
\]
\[= c^0_k e^{-\alpha k^2 \pi^2 t} \sin(k\pi x).\]

• **Main idea:** Choose \(c^0_k\)’s to satisfy initial condition.

• **Aside:**
  - **Q:** What are the units of \(\alpha\)?
  - **Q:** What are the units of \(k\)?
  - **Q:** What happens to the time-scale if domain is \(x \in [0, \frac{1}{2}]\)?
Behavior of Solutions for Various Wavenumber, $k$.

Figure 1: Solution behavior for first three eigenmodes, $c_k(t) s_k(x)$.

Figure 2: Coefficient amplitude for first three solutions, $c_k(t) s_k(x)$: (left) linear scale; (right) semi-log. Straight-line behavior in semi-log plot indicates exponential decay and reveals that by time $t = 20$ the 3rd mode has decayed significantly below even the 2nd mode. This observation is not evident in the linear-scale.
Constructing Full Solution:

- Choose $c_k^0 := c_k(0)$ values, $k = 1, \ldots,$ such that the initial condition is satisfied:

$$u(x, t = 0) = u^0(x).$$

- Problem is linear. Therefore, use superposition:

$$u(x, t) = \sum_{k=1}^{\infty} c_k(t) \sin(k\pi x).$$

- Satisfies $u_t = -\alpha u_{xx}$ and $u(0, t) = u(1, t) = 0$.

- To find $c_k^0$, use IC:

$$u^0(x) = \sum_{k=1}^{\infty} c_k^0 \cdot \sin(k\pi x).$$
• Sine functions are orthogonal on $[0,1]$:

$$(s_k, s_j) := \int_0^1 \sin(k\pi x) \sin(j\pi x) \, dx = 0, \quad k \neq j,$$

$$= \frac{1}{2}, \quad k = j.$$

• This is easily shown by using integration-by-parts, twice.

$$(s_k, s_j) = \frac{k^2}{j^2} (s_k, s_j),$$

which implies $(s_k, s_j) = 0$ for $i \neq j$. 
• Now consider

\[ \int_0^1 \sin(k\pi x) u^0(x) \, dx = \int_0^1 s_k(x) \left[ \sum_{j=1}^{\infty} c_j^0 s_j(x) \right] \, dx \]

\[ = \sum_{j=1}^{\infty} c_j^0 \cdot \int_0^1 s_k(x) s_j(x) \, dx \]

\[ = \sum_{j=1}^{\infty} c_j^0 \cdot (s_k, s_j) \]

\[ = \frac{1}{2} c_k^0. \]

• Therefore,

\[ c_k^0 = 2 \int_0^1 \sin(k\pi x) u^0(x) \, dx \]

\[ = 2(s_k, u^0) \]

• The process of finding the \( c_k^0 \)'s is called harmonic analysis.

• It yields the amplitude of the frequency content associated with wavenumber \( k \).
Finding the coefficients:

- It’s annoying to have to integrate $\sin(k\pi x)u^0(x)$ for arbitrary $u^0$.

- Therefore, use quadrature. (yay!)

- We’ll use trapezoidal rule (an excellent choice for periodic functions, but these functions are not generally periodic....)

- Define,

$$
(f, g) := \int_0^1 f(x) g(x) \, dx, \quad \text{(continuous inner product)},
$$

$$
(f, g)_n := h \sum_{j=0}^{n} f(x_j) g(x_j) w_j, \quad \text{(discrete inner product)},
$$

with quadrature weights $w_0 = w_n = \frac{1}{2}$, $w_j = 1$, $j = 1, \ldots, n - 1$.

- Orthogonality of basis functions, $s_k(x)$:

$$
(s_k, s_j) = \frac{1}{2} \delta_{kl}
$$

$$
(s_k, s_j)_n = \frac{1}{2} \delta_{kl}, \quad 0 < k, l < n.
$$

- Note that, because $s_k(0) = s_k(1) = 0$, we have

$$
(s_k, f)_n = h \sum_{j=1}^{n-1} s_k(x_j) f(x_j)
$$

$$
= \frac{1}{n} \sum_{j=1}^{n-1} s_k(x_j) f(x_j).
$$
Fourier (sine) Synthesis:

- Consider a truncated approximation (dropping superscript “0”):

\[
u_n(x) := \sum_{k=1}^{n-1} s_k(x) c_k \approx u(x)
\]

\[
u_n := \sum_{k=1}^{n-1} s_k(x_j) c_k, \ x_j = j \cdot h, \ h = \frac{1}{n},
\]

\[:= S\mathbf{c}.
\]

- Think of the columns of \(S\) as the basis functions and \(S\mathbf{c}\) as a linear combination of these basis functions.

- The matrix \(S\) is defined as

\[
S_{jk} := s_k(x_j) = \sin(k\pi x_j) = \sin(\pi kj/n) = S_{kj}.
\]

- Note that

\[
(S^T S)_{kj} = \sum_{l=1}^{n-1} \sin(k\pi l/n) \sin(j\pi l/n)
\]

\[= n \cdot h \cdot \sum_{l=1}^{n-1} \sin(k\pi l/n) \sin(j\pi l/n)
\]

\[= n \cdot (s_k, s_j) = \frac{n}{2} \delta_{ij}.
\]

Thus, \(S^T S = \frac{n}{2} I\).
• Use quadrature to find the basis coefficients, $c = [c_1, c_2, \ldots, c_{n-1}]^T$,

$$c_k = 2(s_k, u)_n,$$

$$= 2h \sum_{j=1}^{n-1} s_k(x_j) u_j,$$

$$c = 2h S^T u_n.$$

• Combining with the definition of $u_n$, we have

$$c = 2h S^T S c = c. \text{ (no surprise)}$$

• We call $c = 2h S^T u_n$ the discrete sine transform (DST),
and $u_n = S c$ synthesis.

• Generally, finding $c$ is hard—except here, because $S^T = S$!

• We have

$$c = 2h S^T u_n = 2h S u_n,$$

which is just as easy as synthesis.
• Note that once we have \( \zeta \), we can construct \( u(x) \) on a fine mesh, just as we did in interpolation.

• Define \( u_m = \{u(\tilde{x}_i)\} \), \( i = 0, \ldots, m \), \( \tilde{x}_i = i/m \), and

\[
\begin{align*}
  u_m &= S_m \zeta, \\
  (S_m)_{ik} &= \sin(k\pi \tilde{x}_i).
\end{align*}
\]

• \( S_m \) is a rectangular \((m + 1) \times (n - 1)\) matrix with \( m > n \).
Example:

- Suppose your boss asks you to do a sine-reconstruction of a square wave.
- Your answer:

```matlab
n=4;
x=1:(n-1); x=x'/n; n1=n-1; h=1./n;
S=zeros(n1,n1);
for k=1:n1; S(:,k)=sin(k*pi*x); end;
u0=0*x + 1;
c=(2*h)*S'*u0;
u=S*c;
plot(x,u,'ro','linewidth',2); axis([0 1 0 1.5]); axis square
```

Can you make it continuous?
• Boss is unimpressed.
• Wants a continuous line for unimaginative management.

```matlab
n=4;
x=1:(n-1); x=x'/n; n1=n-1; h=1./n;
S=zeros(n1,n1);
for k=1:n1; S(:,k)=sin(k*pi*x); end;
u0=0*x + 1;
c=(2*h)*S’*u0;
u=S*c;
plot(x,u,’ro-’,’linewidth’,2); axis([0 1 0 1.5]); axis square
title(’Fourier Reconstruction of Square Wave’)
```

• matlab demo: bq2.m