

Recap: Lecture 3

- Introduced the *general form of ODEs* suitable for numerical solution and the *Runge-Kutta (RK)* (Taylor-Series) methods for solving them.
 - Just need to define the *right-hand-side function* which defines the first derivatives of all variables
- Higher-order methods, e.g., fourth-order Runge-Kutta (RK4) outperform Euler's method ('first-order RK')
 - Clever way of incorporating 'half-steps' to reduce truncation error to $O(\tau^5)$
- Played around with nonlinear dynamics of a simple pendulum



Computational Physics: Lecture

4

- *Partial Differential Equations (PDEs)*
 - Multiple independent variables! (\mathbf{x}, t).
- *Diffusion equation*
 - Evolution in time and space of an initial temperature spike.
 - *Forward Time Centered Space (FTCS)* discretization.
 - Solution by marching forwards in time (by matrix power iteration).
- *Stability analysis*
 - Derive dependence of algorithm stability on the time step.
 - (can explain in terms of matrix spectral radius).
- Code: `diffusion_ftcs.m`



[FYI] Classification of linear second-order PDEs

$$a(x, t) \frac{\partial^2 u}{\partial x^2} - b(x, t) \frac{\partial^2 u}{\partial x \partial t} - c(x, t) \frac{\partial^2 u}{\partial t^2} + \dots = 0.$$

- Consider the quantity: $d = b^2 - 4ac$. The PDE is:
 - *hyperbolic* if $d > 0$ (e.g., wave equation),
 - *parabolic* if $d = 0$ (e.g., diffusion equation),
 - *elliptic* if $d < 0$ (e.g., Laplace's equation).

Have you ever played *Guess the PDE* before?!

1. ? $\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}$ ($\kappa > 0$).

- (heat) diffusion

2. ? $\frac{\partial^2 A}{\partial t^2} = c^2 \frac{\partial^2 A}{\partial x^2}.$

- wave equation

3. ? $\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0.$

- Laplace or Poisson equations
(e.g., in electrostatics,
 $\nabla^2 \phi = -\sigma/\epsilon_0$).

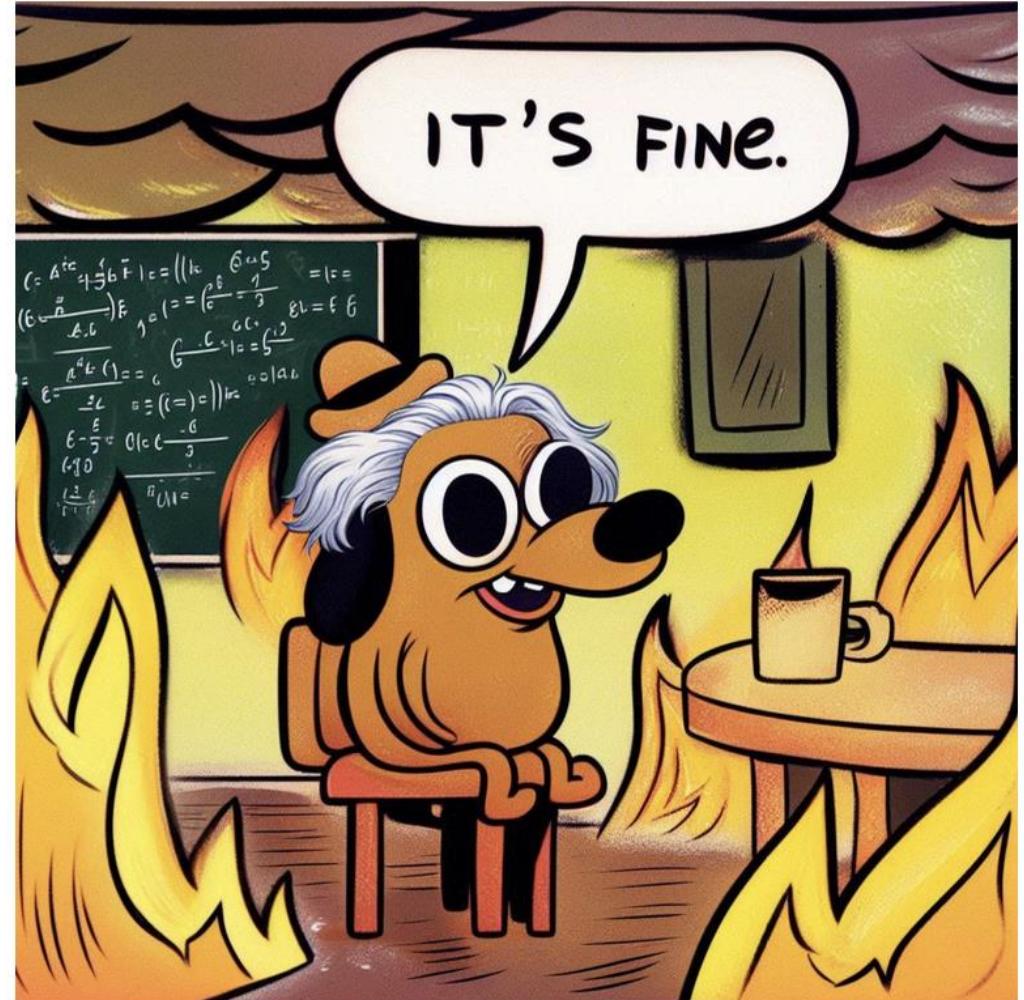


Partial Differential Equations (PDEs)

- PDEs involve more than one independent variable
 - e.g., position \mathbf{x} and time t .
- Spatial boundaries require setting of **boundary conditions**.
 - Common ones:
 - *Dirichlet*: dependent variable is given on the boundary.
 - *Neumann*: (normal) derivative of dependent variable is given on the boundary (in the direction perpendicular to the boundary).
 - *Periodic*: matching conditions at boundaries.

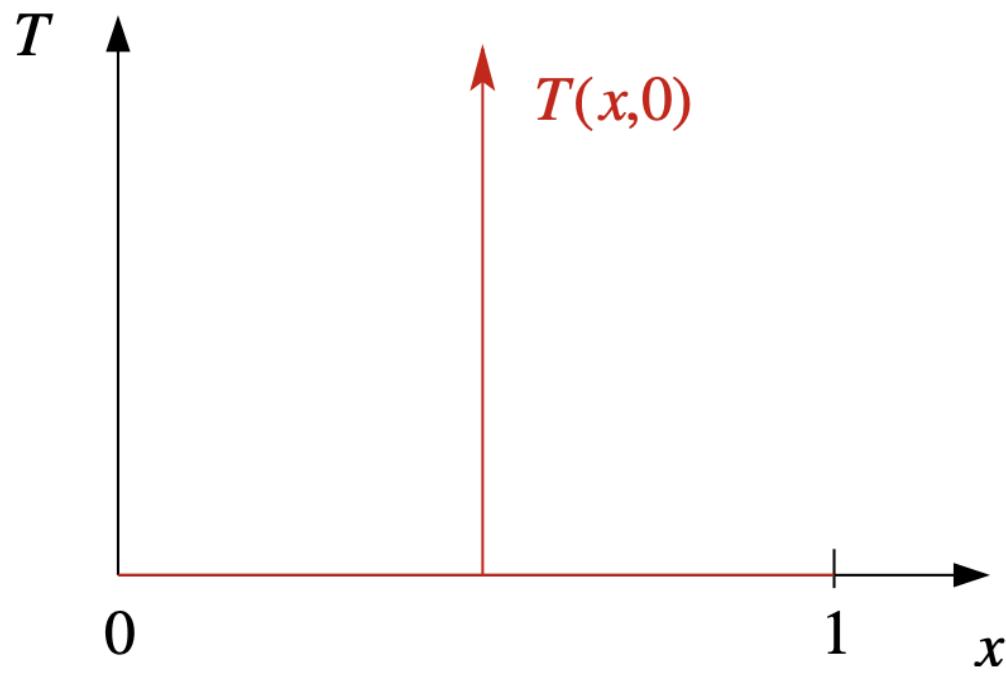
This week's problem: Spread of a heat spike

- Newton's equation for heat conduction
in 1D: $\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2}$,
 - Temperature, $T(x, t)$
 - Thermal conductivity/diffusion coefficient, $\kappa > 0$ (constant).
- Our goal (IVP): Determine $T(x, t)$ for $t > 0$ on $0 < x < 1$.
 - We'll work on the unit interval (by non-dimensionalizing x with a scale factor L_s).

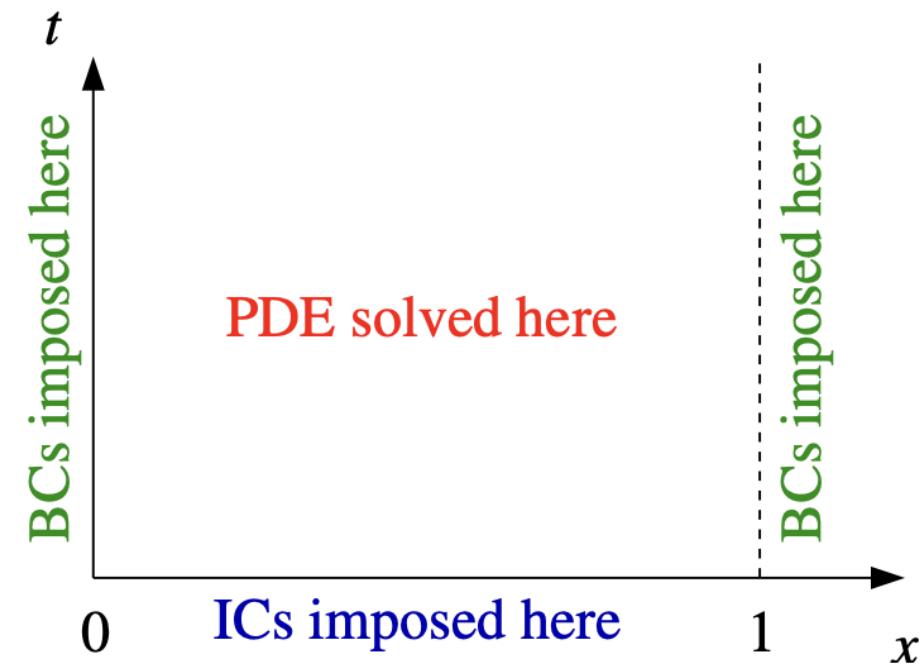


Heat spike: initial and boundary conditions

- *Initial condition:* $T(x, 0) = \delta(x - \frac{1}{2})$.
 - A spike of heat at $x = \frac{1}{2}$.
 - $\delta(u)$: the *Dirac delta function*. $\delta(u) = 0$ if $u \neq 0$ and $\int_{-\infty}^{\infty} \delta(u) du = 1$.



- *Dirichlet boundary conditions:*
 - $T(0, t) = T(1, t) = 0$.
- We are solving in space and time (the $x - t$ plane): note axes of plot!

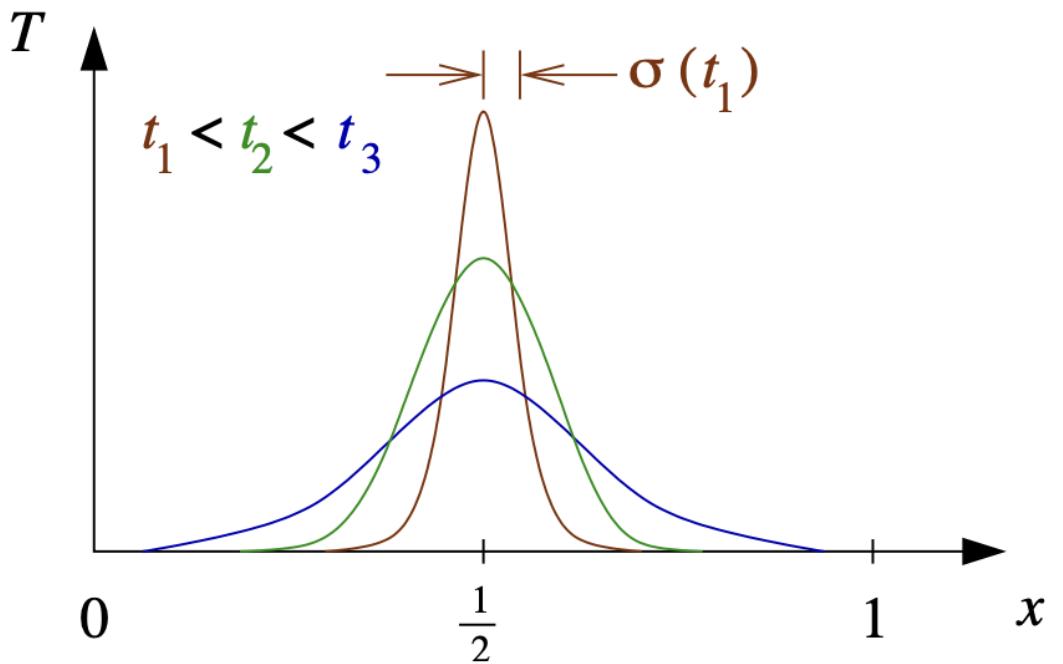


Approximate analytic solution for the heat spike IVP

For small t :

$$T(x, t) = (4\pi\kappa t)^{-1/2} \exp\left[-\frac{1}{4\kappa t}(x - \frac{1}{2})^2\right].$$

- A *Gaussian* with 'width' $\sigma(t) = \sqrt{2\kappa t}$.
- It is exact in an *unbounded* region.
- A good approximation if $\sigma(t) \lesssim \frac{1}{2}$, i.e., for $t \lesssim (8\kappa)^{-1}$.
- (BCs, $T(0, t) = T(1, t) = 0$, are not met exactly. See Garcia textbook for a more complicated solution which enforces the BCs exactly.)

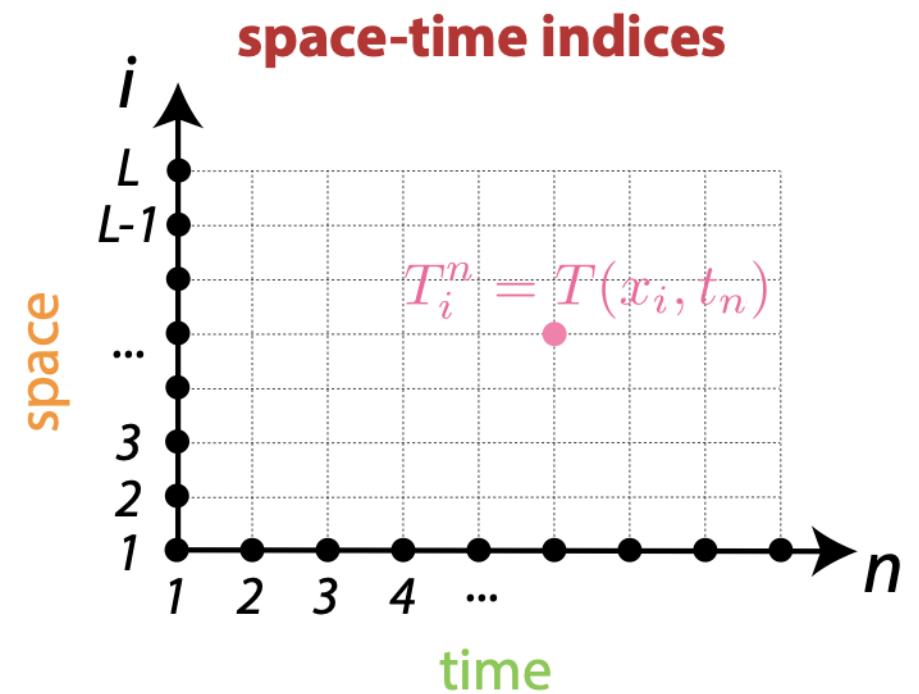
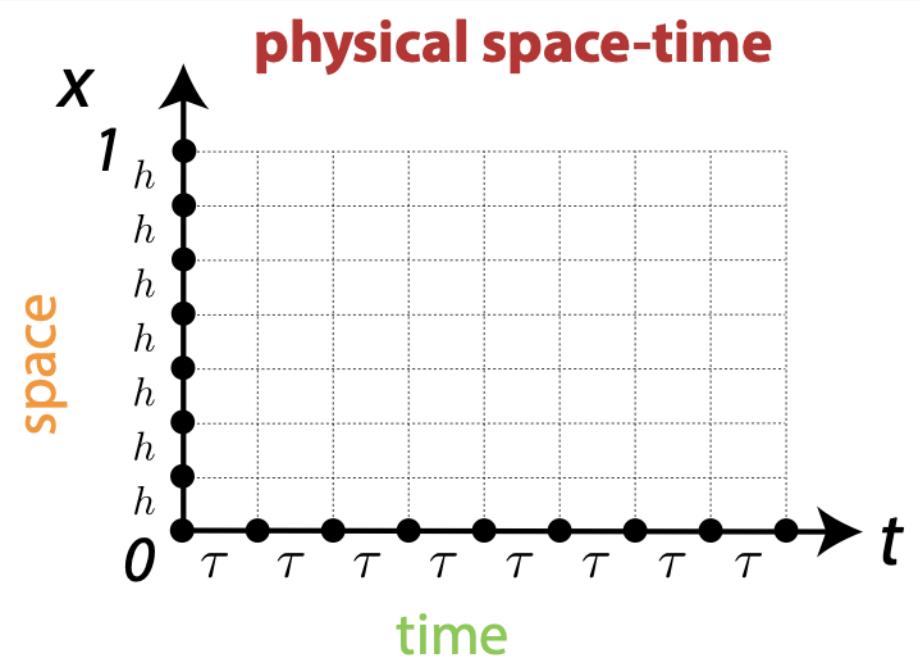


Discretization

- Remember how we discretized time into discrete *time steps*...?
 - We'll need to do the same now for space: *spatial steps*.
- *Each spatial point* (along the domain of x) evolves forward through time (and for this PDE should simulate 'spreading' of T)...
- We know about ways of numerically approximating the time derivative, $\partial/\partial t$ to take *time steps* τ ...
 - What about evaluating the $\partial^2/\partial x^2$?

Discretization

- Discretize in *position*: $x_i = (i - 1)h$ for *spatial step* $h = (L - 1)^{-1}$ for a spatial index: $i = 1, 2, \dots, L$.
- Discretize in *time*: $t_n = (n - 1)\tau$ for *time step* τ for a time index: $n = 1, 2, \dots$
- *Notation*: $T_i^n = T(x_i, t_n)$: T at spatial index i and time index n .
 - [Don't be confused: n is not an exponent!]



★ Forward Time Centered Space (FTCS) Scheme

- To solve the discretized diffusion PDE, $\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2}$:

- **1)** *Forward difference* approximation to *time* derivative (time step τ):

- $$\frac{\partial T}{\partial t} \Big|_{(x_i, t_n)} = \frac{T(x_i, t_n + \tau) - T(x_i, t_n)}{\tau} + O(\tau),$$
$$\approx \frac{1}{\tau} (T_i^{n+1} - T_i^n).$$

- **2)** *Centered difference* approximation to *spatial* derivative (spatial step h):

- $$\frac{\partial^2 T}{\partial x^2} \Big|_{(x_i, t_n)} = \frac{T(x_i + h, t_n) - 2T(x_i, t_n) + T(x_i - h, t_n)}{h^2} + O(h^2),$$
$$\approx \frac{1}{h^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n).$$

- Together we get a *discrete* version of the PDE as an update equation 😊
$$T_i^{n+1} = T_i^n + f (T_{i-1}^n - 2T_i^n + T_{i+1}^n)$$

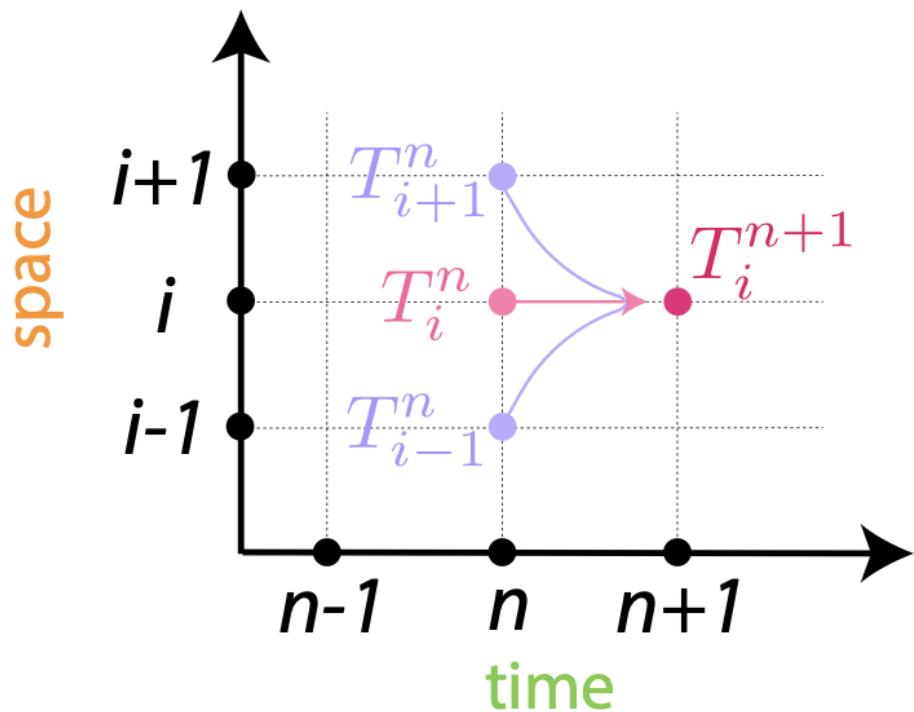
- where our constants are wrapped up in $f = \frac{\kappa \tau}{h^2}$.

The FTCS Scheme, $T_i^{n+1} = T_i^n + f(T_{i-1}^n - 2T_i^n + T_{i+1}^n)$

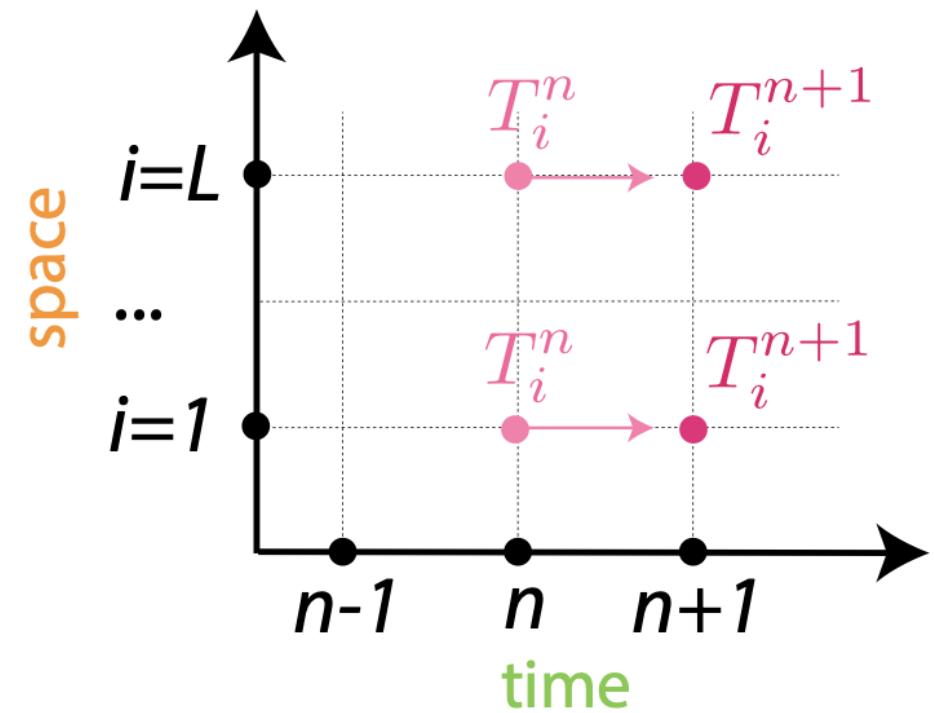
- A rule for *marching forward in time* .
- [called an *explicit* method: RHS depends on n but not $n + 1$].
- ? What range of i can this be used for?
 -  $i = 2, 3, \dots, L - 1$
- ? What do we do for the $i = 1$ and $i = L$ positions?
 -  Dirichlet BCs (fixed value at boundary) set by the initial conditions:
 - $T_1^{n+1} = T_1^n$ and $T_L^{n+1} = T_L^n$, for all n .

- (Like giving our two neighbors a piece of cake each at each time step 🍰)

$$T_i^{n+1} = T_i^n + f(T_{i-1}^n - 2T_i^n + T_{i+1}^n)$$



$$\begin{aligned} T_1^{n+1} &= T_1^n \\ T_L^{n+1} &= T_L^n \end{aligned}$$



FTCS updates, $T_i^{n+1} = T_i^n + f(T_{i-1}^n - 2T_i^n + T_{i+1}^n)$

Writing every update out:

$$i = 1 : \quad T_1^{n+1} = T_1^n \quad (\text{No change})$$

$$i = 2 : \quad T_2^{n+1} = T_2^n + f(T_1^n - 2T_2^n + T_3^n)$$

$$\vdots \qquad \vdots \qquad .$$

$$i = L - 1 : \quad T_{L-1}^{n+1} = T_{L-1}^n + f(T_{L-2}^n - 2T_{L-1}^n + T_L^n)$$

$$i = L : \quad T_L^{n+1} = T_L^n \quad (\text{No change})$$

- 🌈 We can write this in **matrix form**: $\mathbf{T}^{n+1} = \mathbf{A}\mathbf{T}^n$ (walking forward in time 🚶)
- ? What is the update matrix \mathbf{A} for the FTCS scheme??

Matrix form of FTCS, $T_i^{n+1} = T_i^n + f(T_{i-1}^n - 2T_i^n + T_{i+1}^n)$

- We can write $\mathbf{T}^{n+1} = (\mathbf{I} + \mathbf{D})\mathbf{T}^n = \mathbf{A}\mathbf{T}^n$, with: $\mathbf{T}^n = \begin{pmatrix} T_1^n \\ T_2^n \\ \vdots \\ T_L^n \end{pmatrix}$ and $\mathbf{D} = f \begin{pmatrix} 0 & 0 & 0 & 0 & \dots \\ 1 & -2 & 1 & 0 & \\ 0 & 1 & -2 & 1 & \\ 0 & 0 & 1 & -2 & \\ \vdots & & & & \\ 0 & 0 & 0 & 0 & \dots \end{pmatrix}$.
- Nice thing about matrix form: we can update n steps by *power iteration* 💪 with \mathbf{A} : $\mathbf{T}^n = \mathbf{A}^{n-1}\mathbf{T}^1$,
 - where \mathbf{T}^1 is the initial heat spike 🔥.
- ? Which rows of \mathbf{D} enforce the Dirichlet BCs?
 - 👀 The top and bottom rows.
- ? Do these BCs enforce:
 - a) $T_1^{n+1} = T_1^n$ and $T_L^{n+1} = T_L^n$? or
 - b) $T_1^{n+1} = 0$ and $T_L^{n+1} = 0$?

Representing the initial spike

- How can we represent the initial spike $T(x, 0) = \delta(x - \frac{1}{2})$ in our discretization?
- It should satisfy the definition of the delta function:
 - $T(x, 0) = 0$ for $x \neq \frac{1}{2}$ and $\int_0^1 T(x, 0) dx = 1$.
 - This does it the trick :
 - $T_i^1 = \begin{cases} h^{-1} & \text{if } i = \frac{L}{2} \\ 0 & \text{otherwise} \end{cases}$
 - Can verify: $T(x, 0) = 0$ for $x \neq \frac{1}{2}$ and $\int_0^1 T(x, 0) dx \approx h \sum_{i=1}^L T_i^0 = 1$.

Accuracy and the choice of time step, τ for

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2}.$$

- Assuming a spatial step h is chosen, how do you choose a suitable time step τ ?
- Let's be crude 😳:
 - The time, t_h , for a change, ΔT , over one spatial step, h (the resolution of our spatial grid), can be estimated as:
 - $\frac{\Delta T}{t_h} \sim \frac{\kappa \Delta T}{h^2}$, implies that $t_h \sim \frac{h^2}{\kappa}$.
 - So if we want to resolve the dynamics on a spatial scale h , we need a time step $\tau < t_h \equiv \frac{h^2}{\kappa}$.

- Also: the FTCS scheme involves a *weighted average* of neighboring values (👀):
 - $T_i^{n+1} = f [T_{i-1}^n + (f^{-1} - 2) T_i^n + T_{i+1}^n].$
 - For $\tau = \frac{1}{2}t_h = \frac{h^2}{2\kappa} \Rightarrow f = \frac{1}{2}$, the scheme reduces to a simple spatial average of neighboring values (this is diffusion):
 - $T_i^{n+1} = \frac{1}{2} (T_{i-1}^n + T_{i+1}^n).$
 - For $\tau > \frac{1}{2}t_h (\Rightarrow f > \frac{1}{2})$ a *negative* $\times T_i^n$ in the average? (weird much?)
 - But the diffusion equation needs to smooth out (average)... so this suggests $\tau < \frac{1}{2}t_h = \frac{h^2}{2\kappa}$ is needed ★.
 - So maybe terrible things will happen if we set $\tau > \frac{1}{2}t_h \dots?$ 💣

Solving the heat spike IVP 🔥: diffusion_ftcs.ipynb

- Construct update matrix, A , using built-in functions `eye` and `diag`:
- 🌈 Time evolution, $T^n \rightarrow T^{n+1}$, achieved by *power iteration* with A : `temp = A*temp;`
- Numerical and analytic profiles plotted at each time step.
- All $T(x, t)$ values displayed as a surface at the end.

```
# Construct the matrix D associated with the second spatial
# derivative and the boundary conditions
D = -2 * np.eye(L)
D = D + np.diag(np.ones(L-1), 1) + np.diag(np.ones(L-1), -1)
D = kappa * tau * D / h**2

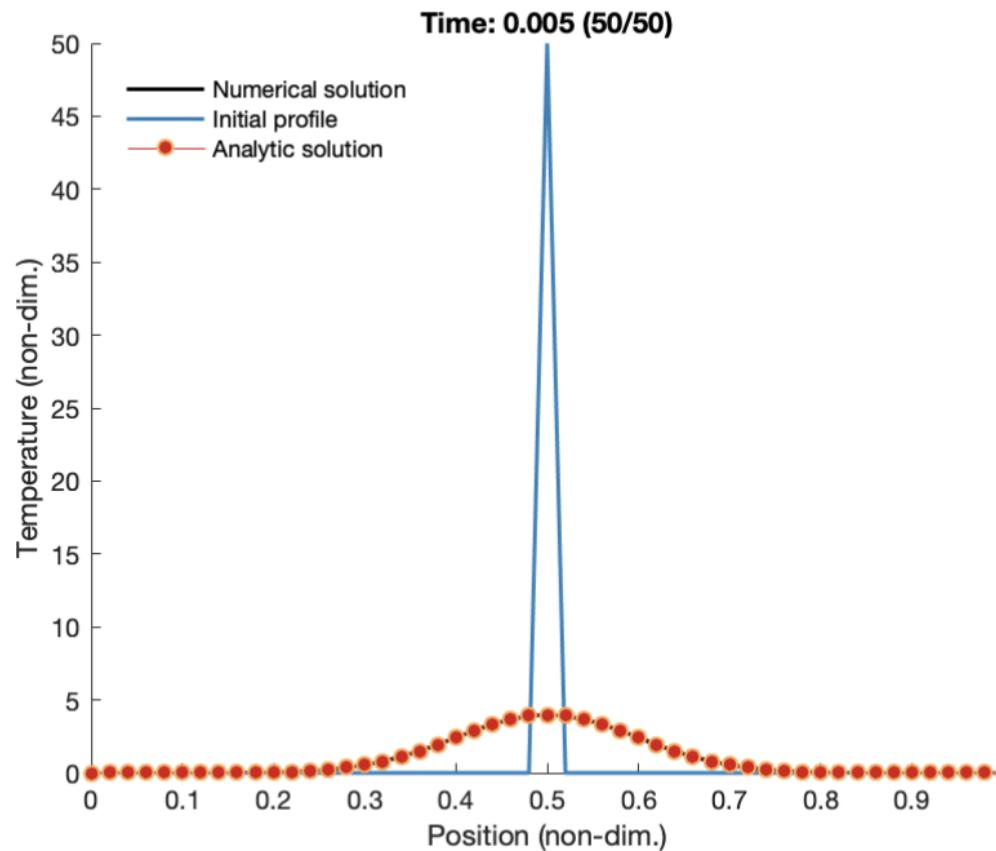
# Impose the Dirichlet boundary conditions
D[0, :] = np.zeros(L)
D[L-1, :] = np.zeros(L)

# Construct the update matrix
A = np.eye(L) + D
```

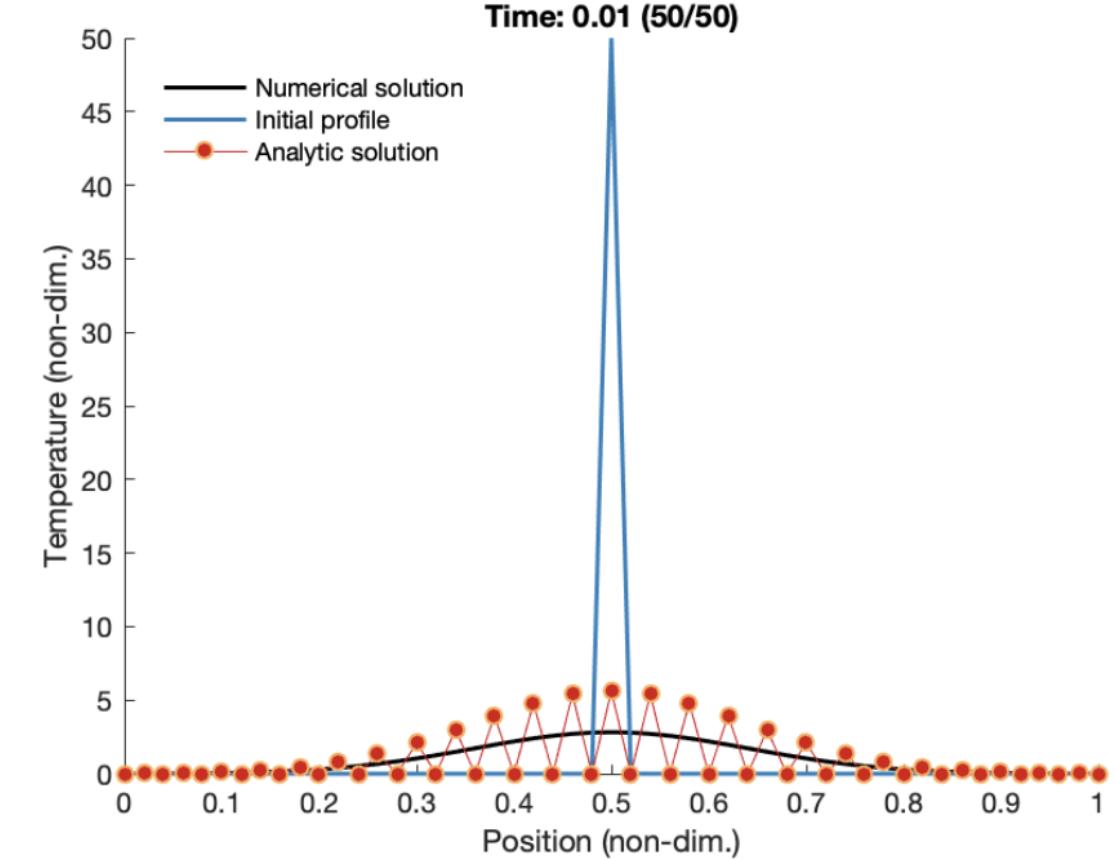
Numerical solution of the heat spike IVP 🔥 with diffusion_ftcs.m

- Default values: $\kappa = 1, h = 0.02, \tau = 10^{-4} = \frac{1}{4}t_h$
- 💻 *Let's play:*
 - $\tau = \frac{1}{4}t_h$ (default).
 - $\tau = \frac{1}{2}t_h$.
 - $\tau > \frac{1}{2}t_h$ 😱.
- 🕒 Time-step dependence:
 - $\tau = \frac{1}{4}t_h$ (default): an *accurate* solution 😊.
 - $\tau = \frac{1}{2}t_h$: *spatial oscillation* 🙏.
 - $\tau > \frac{1}{2}t_h$: *diverges* ($|T|$ grows with each time step). This is *numerical instability* 😬.

Accurate solution for default parameters, $h = 0.02$, $\tau = 10^{-4} = \frac{1}{4}t_h$.



Oscillatory solution for $h = 0.02$ and $\tau = 2 \times 10^{-4} = \frac{1}{2}t_h$.



Understanding stability of a numerical solution.

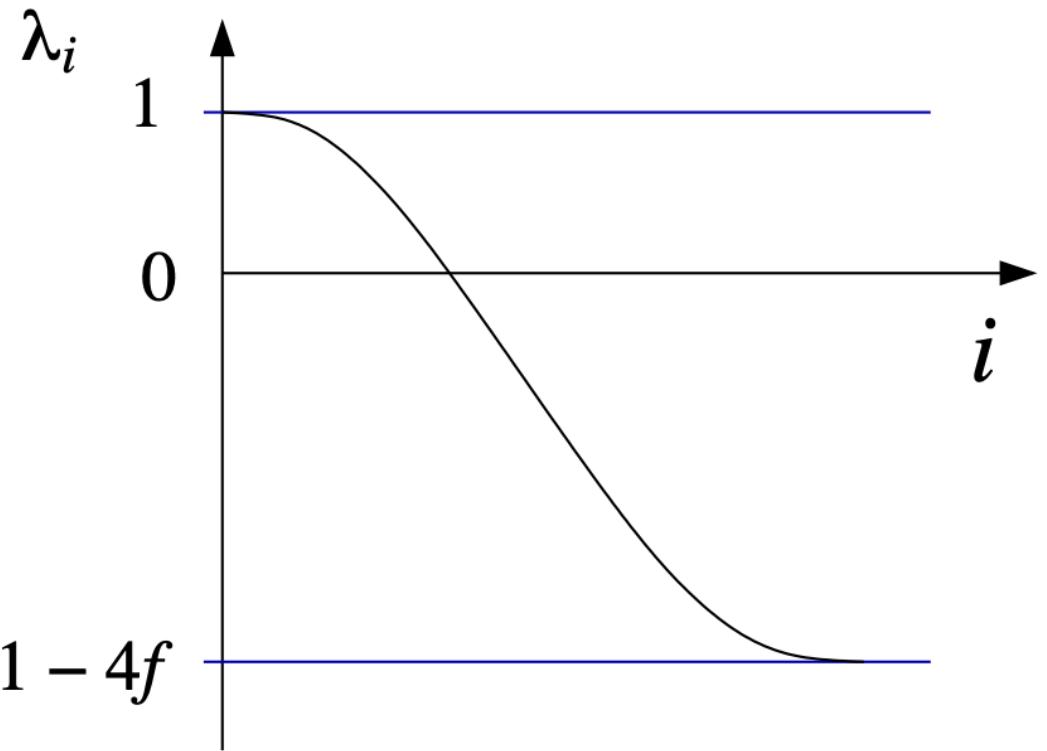
- Today: a *matrix approach*
 - ? What do you think we might need to do?
 - (a better approach next week 😊).
- *Result:*
 - The method is *stable* if $\frac{\kappa\tau}{h^2} \leq \frac{1}{2}$, i.e., $\tau \leq \frac{1}{2}t_h$.
 - The method is *marginally stable* if $\tau = \frac{1}{2}t_h$.

A stability condition from matrix eigenvalues

- Stability is determined by the *eigenvalues*, λ_i , of \mathbf{A} .
- The *spectral radius* of \mathbf{A} is $\rho(\mathbf{A}) \equiv \max(|\lambda_i|) = |\lambda_{\text{dom}}|$
 - λ_{dom} is the *dominant eigenvalue*.
- Time evolution is achieved by power iteration: $\mathbf{T}^m = \mathbf{A}^{m-1}\mathbf{T}^1$, which *projects out the dominant eigenvector* \mathbf{v}_{dom} of \mathbf{A}
 - (the eigenvector with λ_{dom} , the eigenvalue which is largest in magnitude; see Appendix):
 - $\mathbf{T}^m \approx \text{const} \times (\lambda_{\text{dom}})^{m-1} \mathbf{v}_{\text{dom}}$ for $m \gg 1$.
- If $\rho(\mathbf{A}) > 1$ then as $m \rightarrow \infty$, $(\lambda_{\text{dom}})^{m-1} \rightarrow \pm\infty$ and so $\mathbf{T}^m \rightarrow \pm\infty$.
- ★★ So a necessary condition for stability is $\rho(\mathbf{A}) \leq 1$.

FCTS Stability

- The eigenvalues of \mathbf{A} are:
 - $\lambda_1 = \lambda_L = 1$.
 - $\lambda_i = 1 - 4f \sin^2 \left[\frac{\pi(i-1)}{2(L-1)} \right]$, for $i = 2, \dots, L-1$. (with $f = \kappa\tau/h^2$).
- So a necessary condition for stability, $|\lambda_i| \leq 1$, is: $1 - 4f \geq -1$ and thus $f \leq \frac{1}{2}$.
- $\frac{2\kappa\tau}{h^2} \leq 1$ implies that $\boxed{\tau \leq \frac{1}{2}t_h}$.



Appendix: Power Iteration Projects out the Dominant Eigenvector

$$\mathbf{A}^n \mathbf{T}^1 \rightarrow \text{const} \times (\lambda_{\text{dom}})^n \mathbf{v}_{\text{dom}} \quad \text{as } n \rightarrow \infty$$

- \mathbf{A} has eigenvectors, \mathbf{v}_i , with eigenvalues, λ_i .
 - So $\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$ for $i = 1, 2, \dots, L$.
 - And $|\lambda_{\text{dom}}| = \max_i |\lambda_i| = |\lambda_j|$ for some j and $\mathbf{v}_{\text{dom}} = \mathbf{v}_j$.
- The eigenvectors of \mathbf{A} form a *complete set* (i.e., an arbitrary vector in the space may be expressed as a linear sum of the eigenvectors) so we can write $\mathbf{T}^1 = \sum_i C_i \mathbf{v}_i$, where C_i are constants.

Appendix: Power Iteration Projects out the Dominant Eigenvector

- Multiplying $n \gg 1$ times with \mathbf{A} :
 - $\mathbf{A}^n \mathbf{T}^1 = \sum_i C_i \mathbf{A}^n \mathbf{v}_i = \sum_i C_i (\lambda_i)^n \mathbf{v}_i \approx C_j (\lambda_{\text{dom}})^n \mathbf{v}_{\text{dom}}$.
- The sum is dominated by the largest term.
- This argument holds except when $\mathbf{T}^1 \cdot \mathbf{v}_{\text{dom}} = 0$.

In Da Lab

- Running some *fully sick simos*, as usual 😊
- Understanding the stability of the FTCS scheme for 1D diffusion of a heat spike.
- Fun with eigenvalues.
- Running the scheme for different initial temperature profiles.

