Lecture 15:

PDE Boundary Value Problem: the Finite Difference Method

PDE Boundary Value Problem

- Need to solve a PDE in 2+ dimensions within a finite domain (volume) with given boundary conditions

- We will consider linear, elliptic PDEs

[e.g. $\nabla^2\Phi = 0;\quad \nabla^2\Phi = \rho(\vec{r})$] and 2 classes of approach:

  - The Finite Difference Method (FDM)
  - The Finite Element Method (FEM)

- Will concentrate on

  - discretisation
  - reduction to a "Numerical Analysis" problem
  - how to deal with boundary conditions

    (Dirichlet, Neumann, mixed)

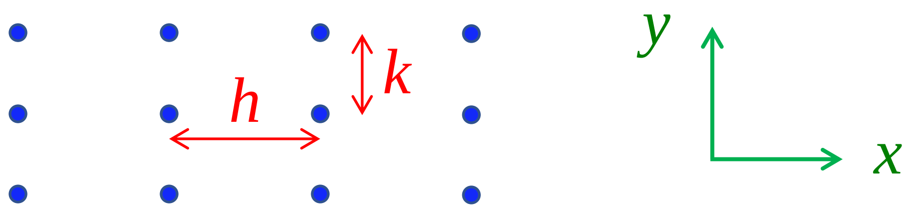
For convenience we will stick to 2D ($\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ etc)

The Finite Difference Method (FDM)

- **Basic idea:** Discretise using grids…

$$\text{e.g. } \nabla^2\Phi = \rho(\vec{r})$$

2d Cartesian: $$\frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2} = \rho(x, y)$$



$$x_{i+1} = x_i + h \qquad y_{j+1} = y_j + k$$

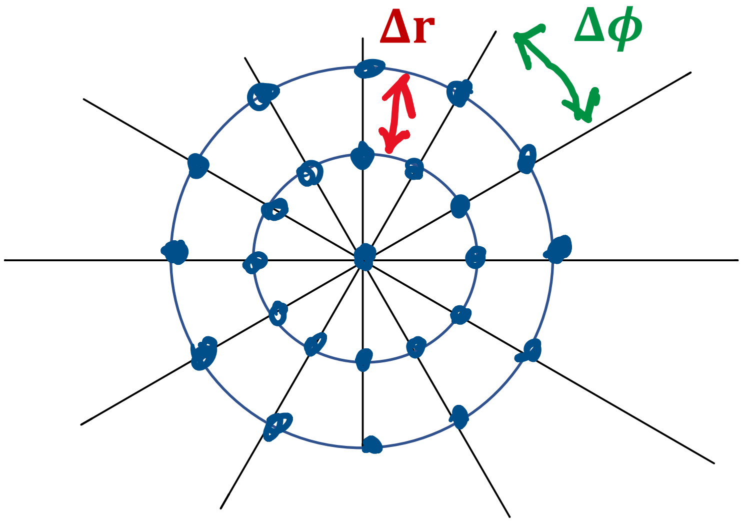
The Finite Difference Method (FDM)

- **Basic idea:** Discretise using grids…

$$\text{e.g. } \nabla^2\Phi = \rho(\vec{r})$$

Polar:

$$\frac{\partial^2\Phi}{\partial r^2} + \frac{1}{r}\frac{\partial\Phi}{\partial r} + \frac{1}{r^2}\frac{\partial^2\Phi}{\partial\phi^2} = \rho(r,\phi)$$



$$r_{i+1} = r_i + \Delta\mathrm{r} \qquad \phi_{j+1} = \phi_j + \Delta\phi$$

$$\nabla^2\Phi = \rho$$

Functions

$$\rho(x,y) \rightarrow \rho(x_i,y_j) = \rho_{i,j} \quad \text{or} \quad \rho_I$$

$$\Phi(x,y) \rightarrow \Phi(x_i,y_j) = \Phi_{i,j} \quad \text{or} \quad \Phi_I$$

I is the "super-index" running over all points

Note: Indices (i,j) are directly linked to (x,y) coordinates ->convenient for interpretation
"Super-index" I is convenient **for numerical implementation** (as we will see below)

Derivatives: apply CDA formulas for x- and y-derivatives, e.g.:

$$\frac{\partial^2\Phi}{\partial x^2}(x,y) \rightarrow \frac{1}{h^2}\left[\Phi_{i+1,j} + \Phi_{i-1,j} - 2\Phi_{i,j}\right]$$

$$\frac{\partial^2\Phi}{\partial y^2}(x,y) \rightarrow \frac{1}{k^2}\left[\Phi_{i,j+1} + \Phi_{i,j-1} - 2\Phi_{i,j}\right]$$

Note: for CDA a regular rectangular grid is required!

$$\nabla^2\Phi = \rho$$

- **Easier to combine derivatives** if step size in x and y is the same, $h = k$:

$$\frac{\partial^2\Phi}{\partial x^2}(x,y) \rightarrow \frac{1}{h^2}\left[\Phi_{i+1,j} + \Phi_{i-1,j} - 2\Phi_{i,j}\right]$$

$$\frac{\partial^2\Phi}{\partial y^2}(x,y) \rightarrow \frac{1}{h^2}\left[\Phi_{i,j+1} + \Phi_{i,j-1} - 2\Phi_{i,j}\right]$$

⬇

$$\nabla^2\Phi = \frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2} \rightarrow \frac{1}{h^2}\left[\Phi_{i+1,j} + \Phi_{i-1,j} + \Phi_{i,j+1} + \Phi_{i,j-1} - 4\Phi_{i,j}\right]$$

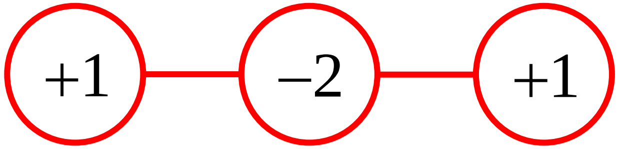
Note: sometimes, it is more convenient to use different step sizes in different directions

(e.g. heat distribution in a thin wire: transversal and longitudinal dimensions have different scales).

“Discretisation stencils” visualise these $O(h^2)$ difference approximations…

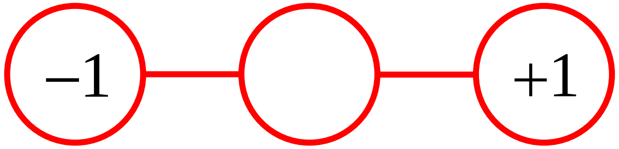
$$\frac{1}{h^2}\left[\Phi_{i+1,j}+\Phi_{i-1,j}-2\Phi_{i,j}\right]$$

$$\frac{\partial^2\Phi}{\partial x^2}$$

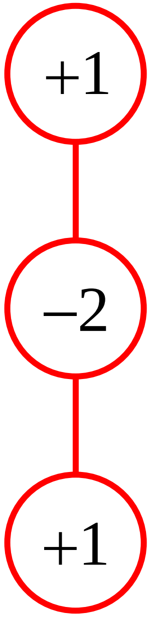


$$\frac{1}{2h}\left[\Phi_{i+1,j}-\Phi_{i-1,j}\right]$$

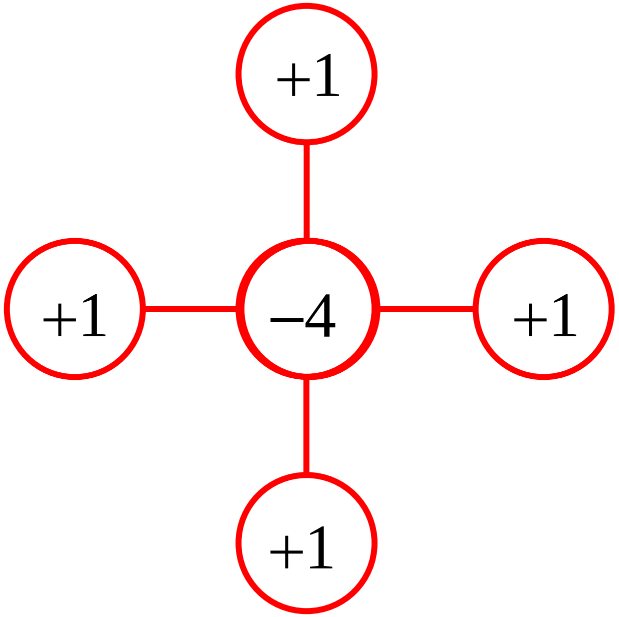
$$\frac{\partial\Phi}{\partial x}$$



$$\frac{\partial^2\Phi}{\partial y^2}$$



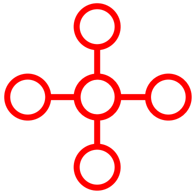
$$\nabla^2\Phi$$



So, on a simple square grid $\quad x_{i+1} = x_i + h \qquad y_{j+1} = y_j + h$

Poisson's equation $\nabla^2\Phi = \rho$ in 2-d is approximated at the general point (i, j) by

$$\Phi_{i+1,j} + \Phi_{i-1,j} + \Phi_{i,j+1} + \Phi_{i,j-1} - 4\Phi_{\mathrm{i,j}} = h^2\rho_{i,j} \qquad ①$$



Expect an equation like ① at each grid point.

If there are N points at which $\Phi_{i,j}$ is unknown, we get N equations and N unknowns.

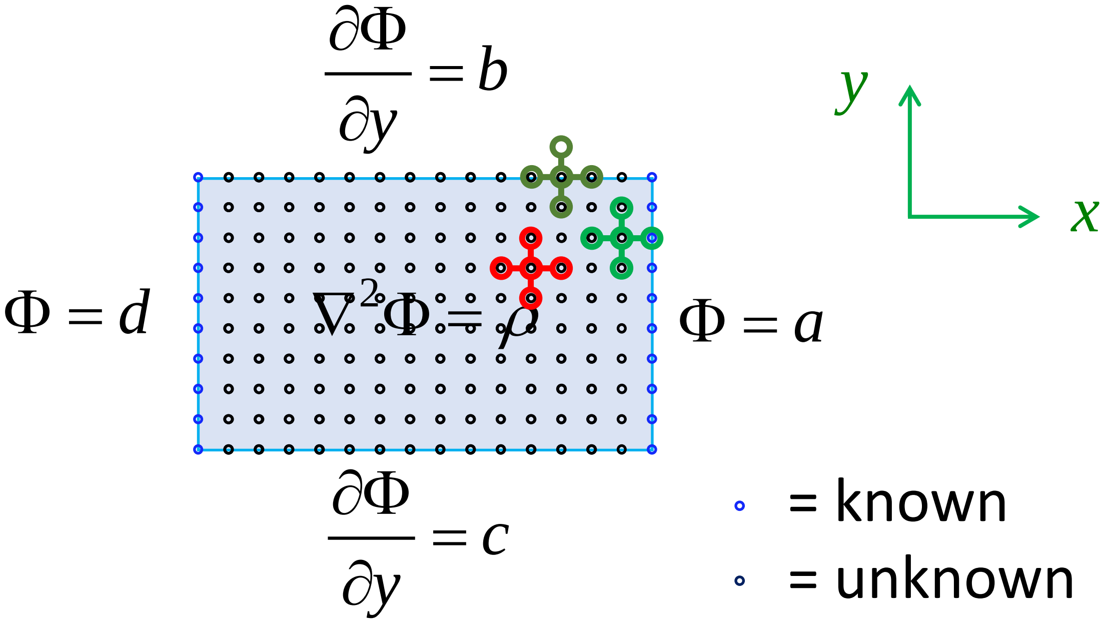
As PDE is linear, solve these as a matrix problem $\widehat{M}\vec{\phi} = \vec{S} \quad ②$

In super-index notation $\sum_J \widehat{M}_{I,J}\phi_J = S_I$, $\vec{S}$ contains values of ρ_I in the r.h.s. and known values of Φ from the boundary conditions

Note: double-indexing like $\phi_{i,j}$ is convenient for conceptualisation, but using super-index notation allows us to formulate the matrix problem in Eq. (2).

Implementation

Illustrate with simple example:



$$\frac{\partial\Phi}{\partial y}=b$$

$$\Phi=d \qquad \nabla^2\Phi=\rho \qquad \Phi=a$$

$$\frac{\partial\Phi}{\partial y}=c$$

∘ = known

∘ = unknown

$N=14\times10=140,$ so $\widehat{M}$ is 140×140

Superindices $I,J=1,\ldots,N$.

Consider equation ① at 3 representative points:

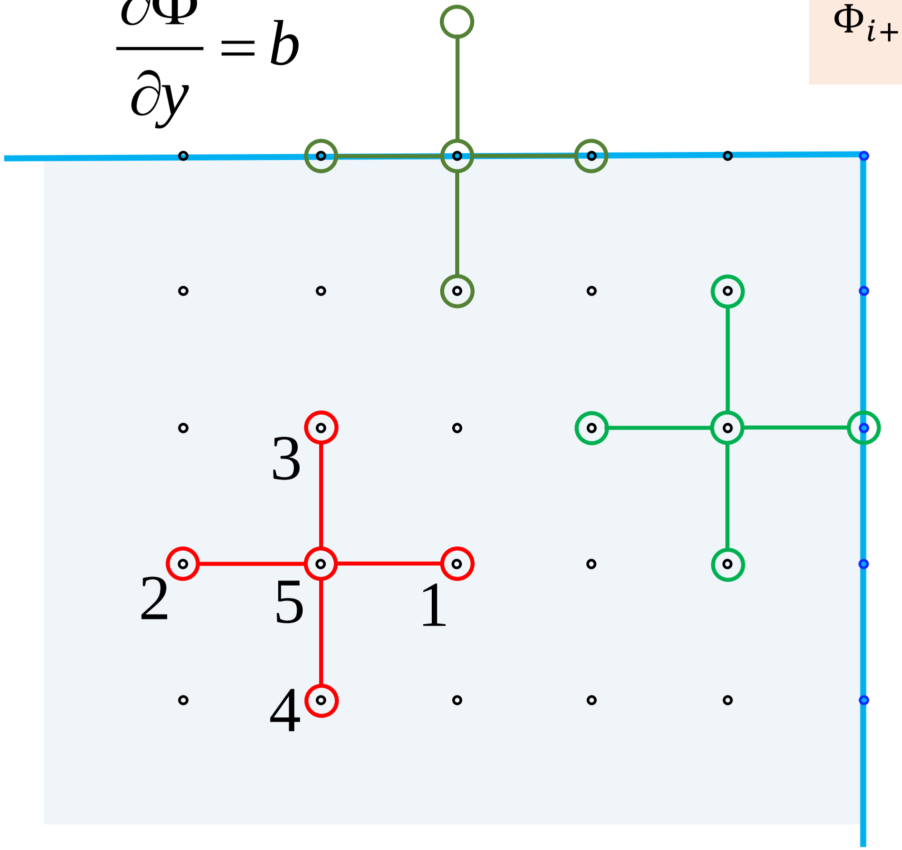
A) in the interior of the region

B) next to a Dirichlet boundary

C) on a Neumann boundary

$$\Phi_{i+1,j} + \Phi_{i-1,j} + \Phi_{i,j+1} + \Phi_{i,j-1} - 4\Phi_{\mathrm{i,j}} = h^2\rho_{i,j} \quad \text{①}$$

$$\frac{\partial \Phi}{\partial y} = b$$



$$\Phi = a$$

$$\boxed{\hat{M}\vec{\phi} = \vec{S}}$$

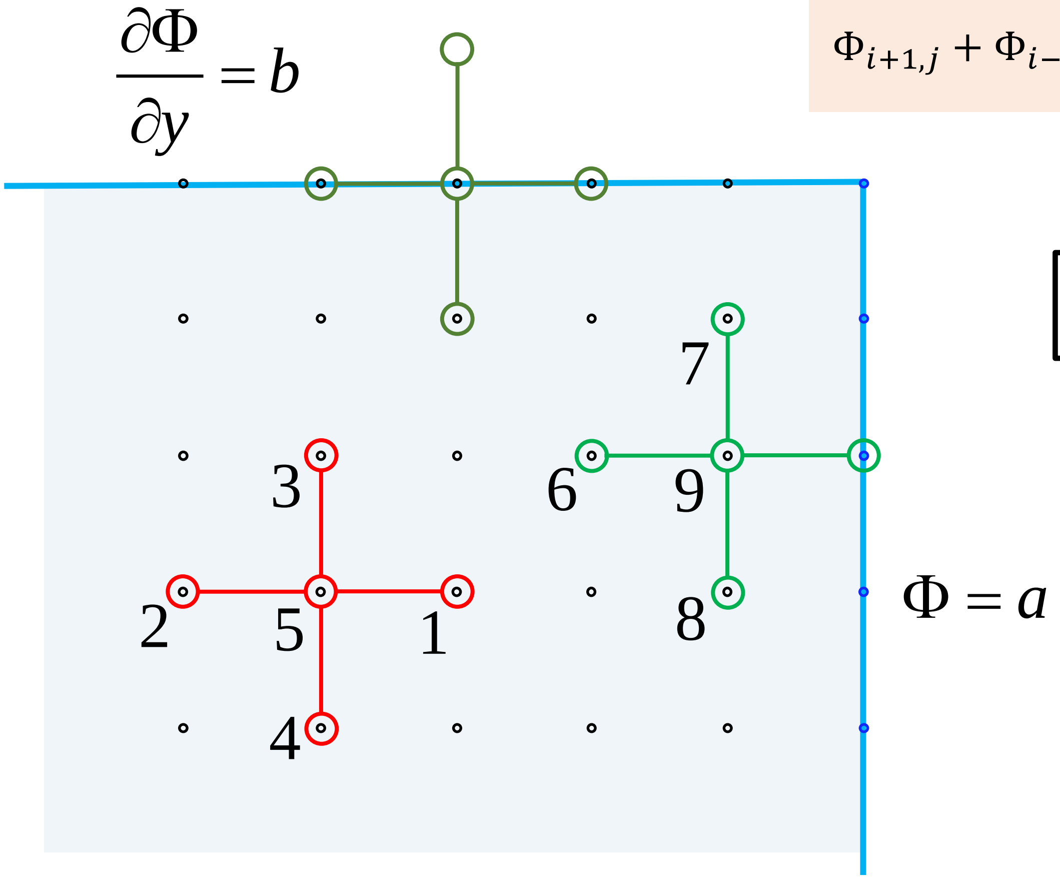
$$\vec{\phi} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \vdots \\ \vdots \\ \Phi_{140} \end{bmatrix}$$

A) $I = 5;\ \ J = 1,2,3,4,5.$ Eqn ① → $\Phi_1 + \Phi_2 + \Phi_3 + \Phi_4 - 4\Phi_5 = h^2\rho_5$

So Row 5 of $\hat{\mathrm{M}}$: $[1 \quad 1 \quad 1 \quad 1 \quad -4 \quad 0 \quad \cdots];\quad S_5 = h^2\rho_5.$

Note that $\hat{M}$ is *sparse*: each row/column has max 5 non-zero elements.

$$\Phi_{i+1,j} + \Phi_{i-1,j} + \Phi_{i,j+1} + \Phi_{i,j-1} - 4\Phi_{\mathrm{i,j}} = h^2\rho_{i,j} \quad ①$$



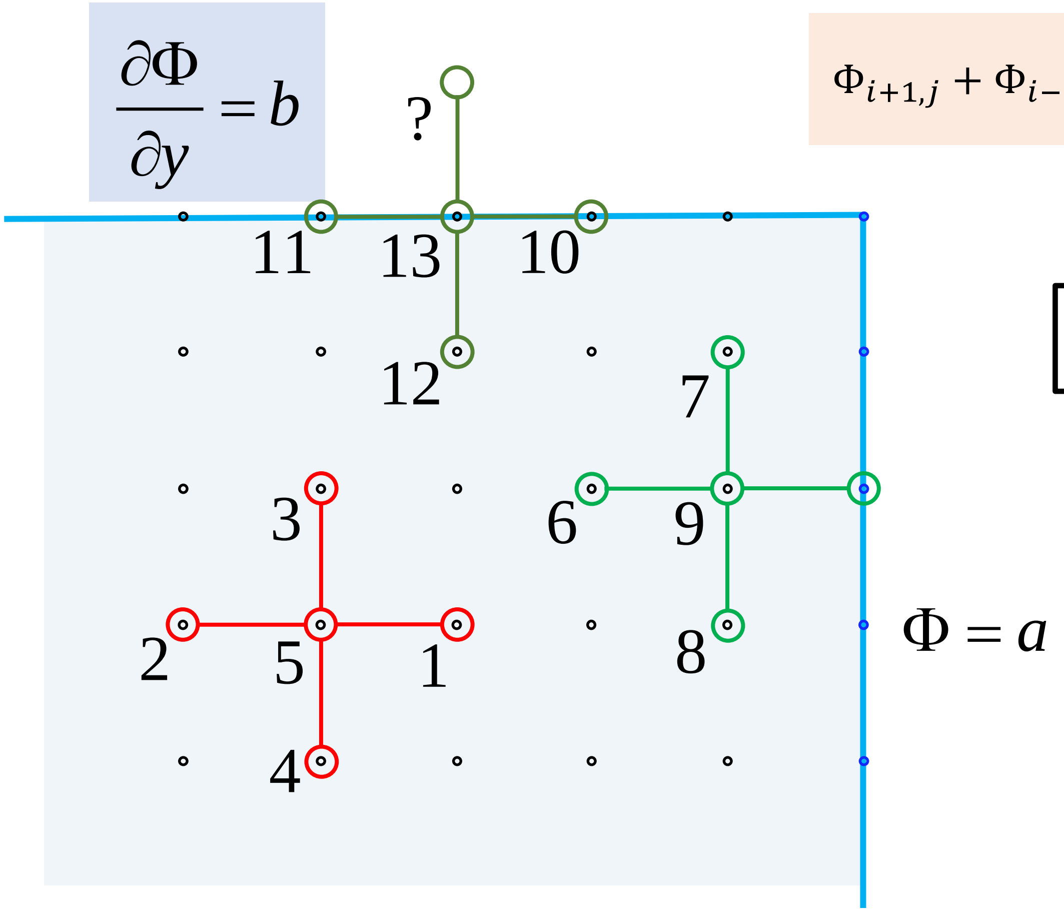
$$\boxed{\hat{M}\vec{\phi} = \vec{S}}$$

$$\vec{\phi} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \vdots \\ \vdots \\ \Phi_{140} \end{bmatrix}$$

B) $I = 9;\ J = 6,7,8,9.$ Eqn ① → $a + \Phi_6 + \Phi_7 + \Phi_8 - 4\Phi_9 = h^2\rho_9$

So Row 9 of $\hat{M}$: $[\cdots \quad \underset{\substack{\uparrow \\ \text{column 5}}}{0} \quad 1 \quad 1 \quad 1 \quad -4 \quad \cdots];\quad S_9 = h^2\rho_9 - a$

$$\Phi_{i+1,j} + \Phi_{i-1,j} + \Phi_{i,j+1} + \Phi_{i,j-1} - 4\Phi_{\mathrm{i,j}} = h^2\rho_{i,j}$$



$$\boxed{\hat{M}\vec{\phi} = \vec{S}}$$

$$\vec{\phi} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \vdots \\ \vdots \\ \Phi_{140} \end{bmatrix}$$

C) $I = 13;\ \ J = 10,11,12,13,?.$ ① → $\Phi_{10} + \Phi_{11} + \Phi_? + \Phi_{12} - 4\Phi_{13} = h^2\rho_{13}$

For site "?" apply BC: $\frac{\partial \Phi}{\partial y} \approx \frac{1}{2h}(\Phi_? - \Phi_{12}) = b, \quad \Rightarrow \Phi_? \approx 2hb + \Phi_{12}.$

So Row 13 of $\hat{M}$: $[\cdots \quad 0 \quad 1 \quad 1 \quad 2 \quad -4 \quad \cdots];\quad S_{13} = h^2\rho_{13} - 2hb.$

↑ column 9

So far, so good. But what if we have

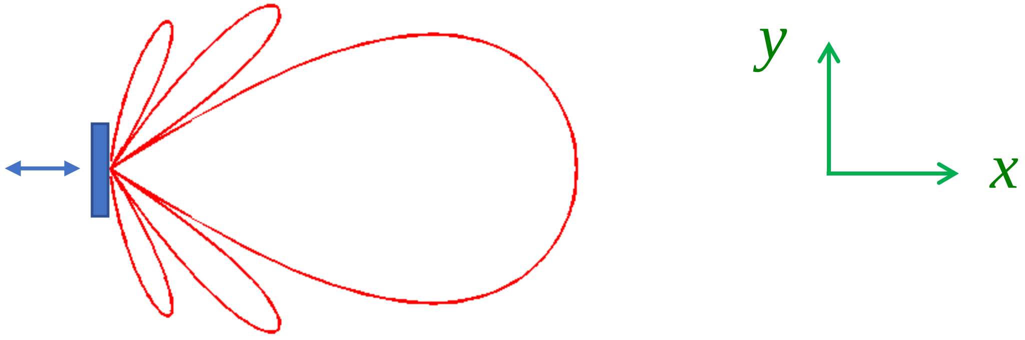
1. A small defect in otherwise a homogeneous medium

   Need more points to resolve the defect BUT if grid is regular need **more points *everywhere***. Will be computationally expensive!

2. Open regions – those with no obvious boundary?

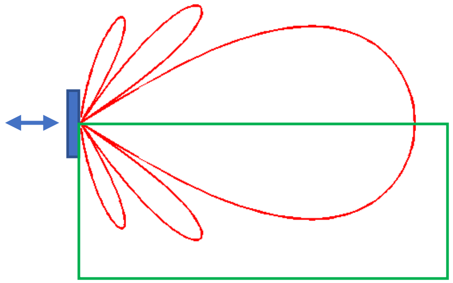
   Eg waves from a circular piston

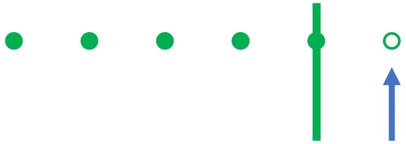
Expect



Radiation pattern is essentially 2D, and is, in principle, ∞ in extent.

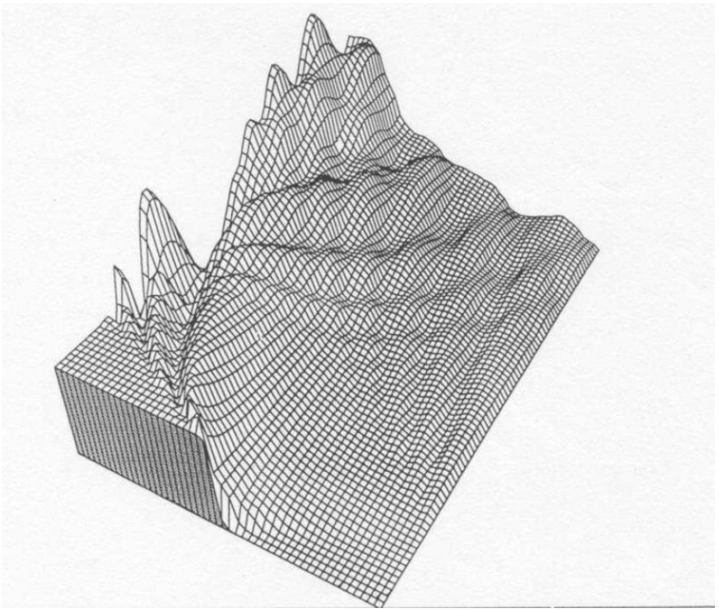
With a finite grid of points in x and y, what happens at (artificial) boundary?



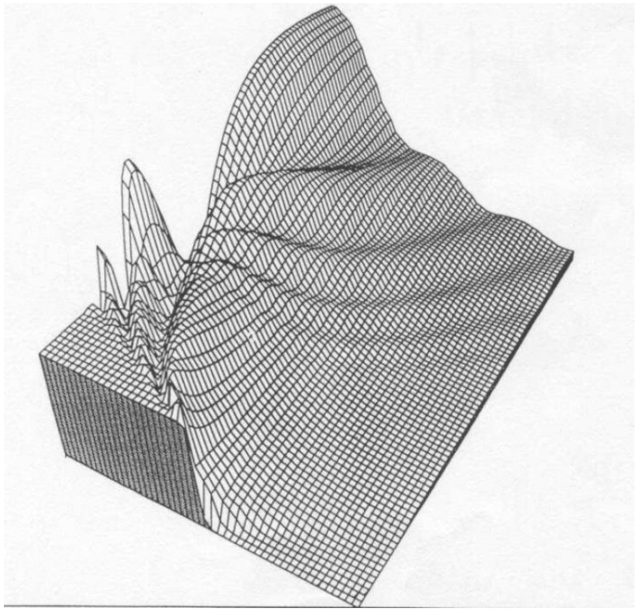


no grid point here: Numerically equivalent to imposing the Dirichlet condition $\Phi = 0$

Leads to unwanted & unphysical reflections:



Bad



Better – how?

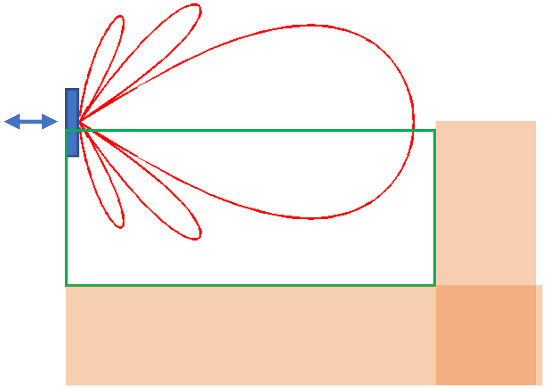
Solutions for open regions?

- Invent a pseudo-boundary condition to prevent reflections

If you just attempt to introduce an absorption along the boundary, it will give you a strong reflection anyway

Need to introduce artificial layers of an absorbing material, with a gradually increasing absorption away from the real boundaries

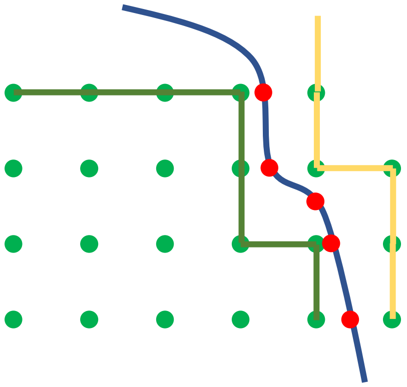
Keywords: *"perfectly matched layer", "impedance matched"*



It only works perfectly when you know exactly what kind of waves are "escaping" your window… In reality you can minimize reflection, but never fully suppress it.

- In practice, the only way to fully eliminate boundary reflections is…. to **make the window size larger**

Boundaries which don't follow the grid?



- more serious

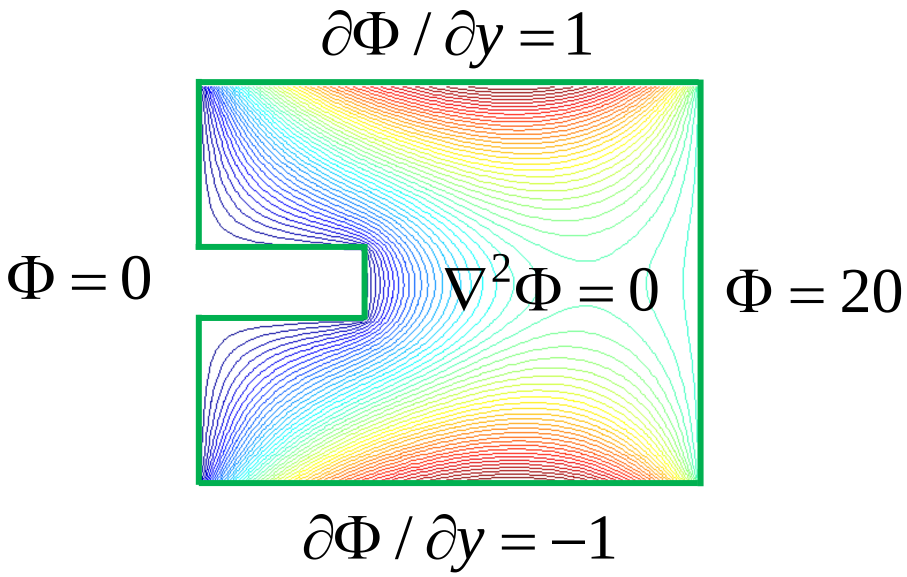
- might *interpolate*: use ● + ●
usually → $O(h)$ errors @ boundary

- might apply *"stair-step" approximation* –
average over 2 (false) boundaries .

Neither solution is particularly good…

Even worse, if $\frac{\partial \Phi}{\partial n}$ is specified (Neumann boundary), when the boundary is not || to the grid

A simple FDM example



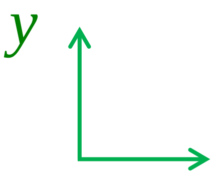
$\partial\Phi / \partial y = 1$

$\Phi = 0$

$\nabla^2\Phi = 0$

$\Phi = 20$

$\partial\Phi / \partial y = -1$

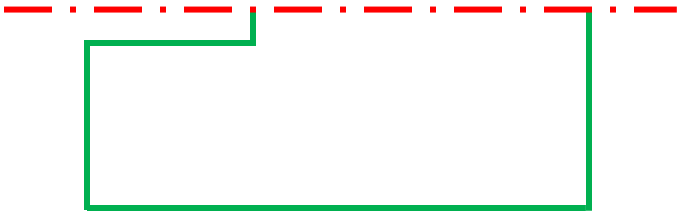


Easy geometry

Easy BCs

BUT $\widehat{M}\vec{\phi} = \vec{S}$
is $N \times N$ problem,
$\rightarrow T \sim N^3$

Use symmetry to reduce N & T:



mirror here (including BCs)

BC on “false boundary”?

Symmetry: Φ is the same just either side of mirror, so $\frac{\partial\Phi}{\partial y} = 0$

Gain? For same grid size, $N \rightarrow\sim N/2$, so $\boldsymbol{T \rightarrow\sim T/8}$.

The FDM for eigenvalues & eigenvectors

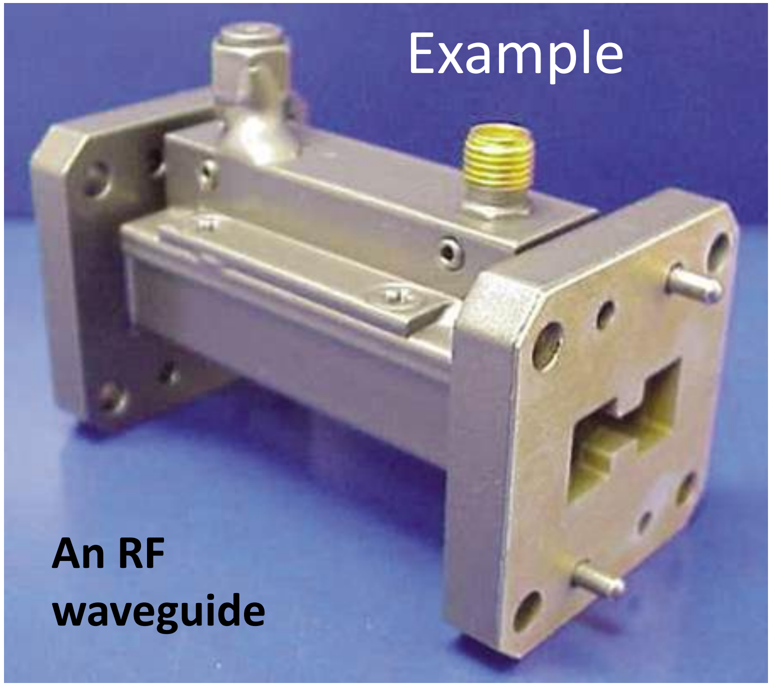
PDE eigenvalue problem: $\widehat{D}(\vec{r},\nabla,\nabla^2)\Phi(\vec{r}) = \lambda\Phi(\vec{r})$

(e.g. electron wave-functions in a crystal, RF and optical waveguides)

Discretise derivatives on a grid: $\widehat{M}\overrightarrow{\Phi} = \lambda\overrightarrow{\Phi}$

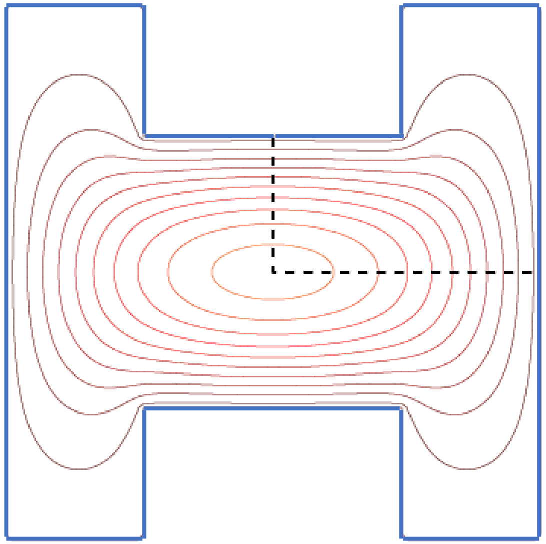
- obtain a matrix eigenvalue problem
- solve numerically for (approx) eigenvalues λ

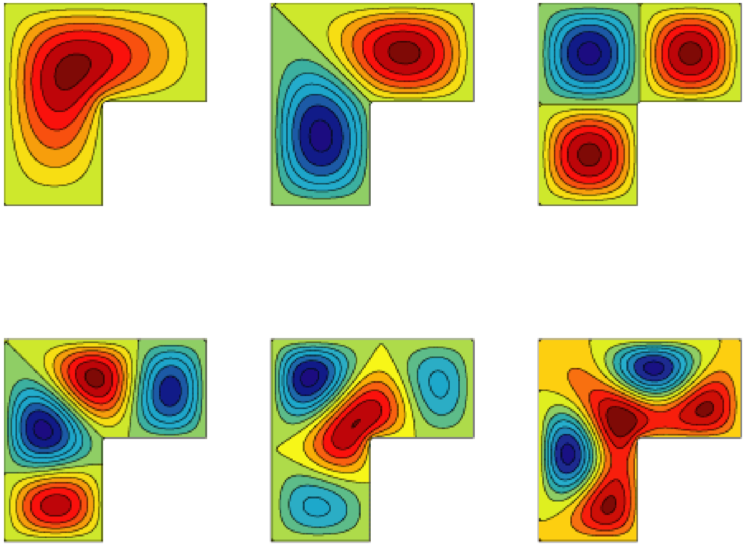
eigenvectors $\overrightarrow{\Phi}$

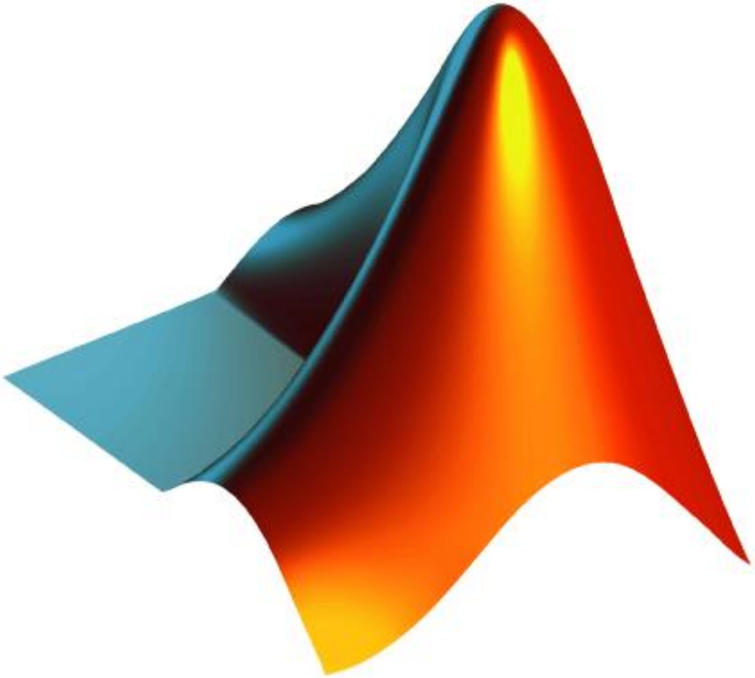


Example

An RF waveguide







Summary:

- The great convenience of the Finite Difference Method (FDM) is in the ease of its implementation

- But it requires a regular grid, and does not work very well with non-rectangular geometries

  - *A combination of **different scales***

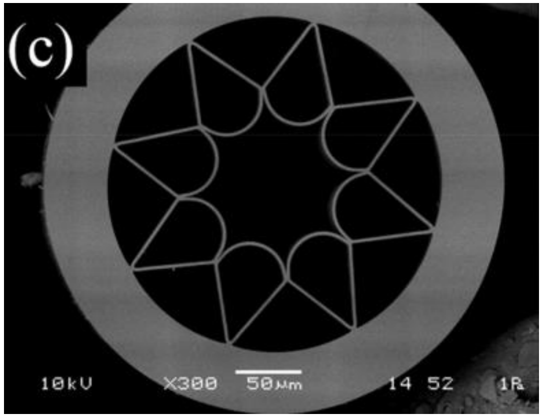
    (thin ~$0.01\mu m$ silica glass inclusions in a large area ~$100\mu m^2$ air void)

  - *A complex network of inner boundaries*

    (non-compatible with either rectangular or polar coordinate regular grids)

  - *Neumann BC at each inner boundary*

A "nightmare" for FDM…



F. Yu and J.C. Knight, *"Negative curvature Hollow-Core Optical Fiber,"* IEEE J. Sel. Top. Quant. El. 22, 4400610 (2016)

There are better methods to tackle such problems…

Lecture 16:

The Finite Element Method: part 1

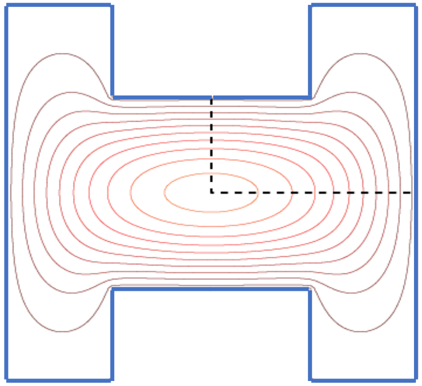
Difficulties with FDM

- The Finite Difference Method is easy to implement, but it has two weaknesses:

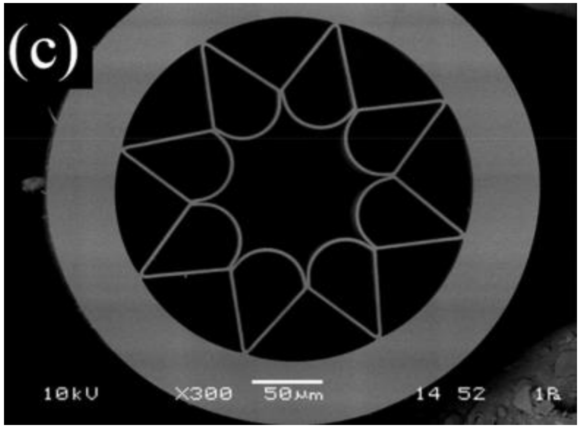
  - ***It requires a regular grid*** *=> hence computationally challenging for problems where a mixture of large-scale and small-scale features is present*

  - ***Difficult to implement boundary conditions*** *for boundaries which are not parallel to the grid*

A straightforward application of FDM



A challenge for FDM



The Finite Element Method (FEM)

- FEM takes a different approach to solve the problem, based on variational calculus:

  - *Instead of looking at the local (in time and space) balances between rates of changes/flux and values of functions…*

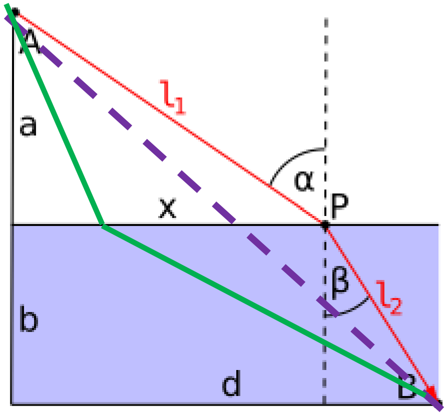
  the idea of variational calculus *is in finding a function which minimizes a certain "global" characteristic*

  (in a way this is similar to how the least squares interpolation method works: we are less concerned with point-by-point agreement, but more with the global error)

- But what is this "global" characteristic?

An example: Fermat's principle

Path taken by a ray between two given points is the path that **can be travelled in the least time**.



https://en.wikipedia.org/wiki/Fermat's_principle

And therefore:

1) We could solve Maxwell equations and find the full path (i.e. derive the Snells' law)

OR:

2) We could consider all possible paths which connect points A and B, and find which one requires the least time for the light to travel.

- Travel time is **a function of the path**, which, in turn, is a function itself... Such "functions of functions" are called **functionals.**

To find a minimum (extremum) of a functional, we use **Variational calculus**

Pierre de Fermat

1607-1665

https://en.wikipedia.org/wiki/Pierre_de_Fermat

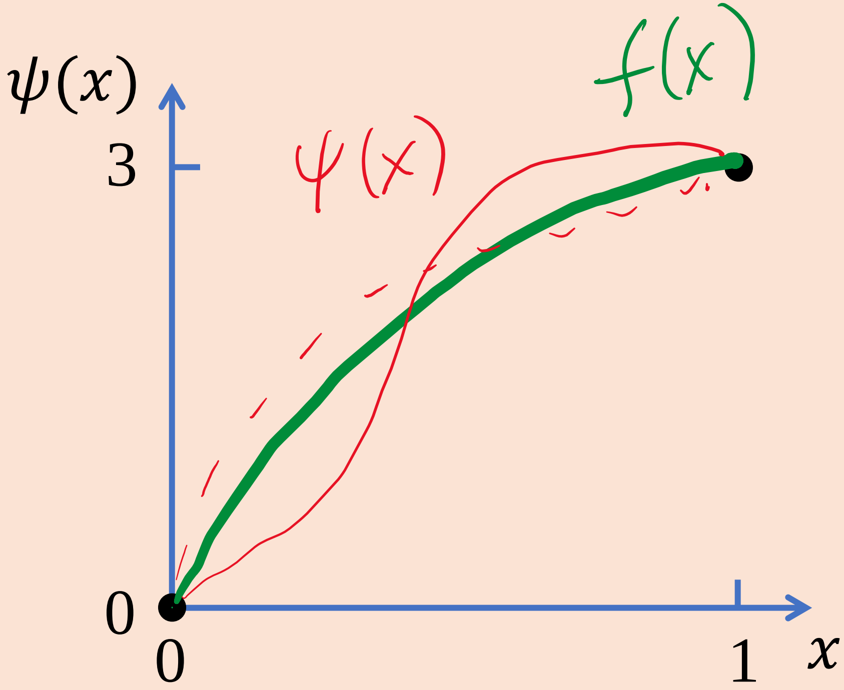
The Finite Element Method (FEM)

- In FEM the functional is always a definite integral, and we are always interested in its minimum.
- To understand the connection between finding the minimum of a functional and solving a differential equation, let us consider the following example:

Out of all possible trial functions $\psi(x)$ in the region $[0,1]$, find the function $f(x)$ which minimises the functional

$$U[\psi] = \int_0^1 \left[\left(\frac{d\psi}{dx} \right)^2 + \psi^2 \right] dx$$

given that $f(0) = 0$ and $f(1) = 3$.



Recall how do we find a local minimum of a function $g(x)$?

- Suppose x_0 is the minimum. Then, for any small η we should observe $g(x_0+\eta) > g(x_0)$, in other words:

$$\boxed{g(x_0+\eta) - g(x_0) > 0}$$

- for any small η !

- Expand in Taylor series:

$$g(x_0+\eta) = g(x_0) + g'(x_0)\cdot\eta + \frac{1}{2}g''(x_0)\cdot\eta^2 + \cdots$$

$$\Rightarrow \quad g(x_0+\eta) - g(x_0) = g'(x_0)\cdot\eta + \frac{1}{2}g''(x_0)\cdot\eta^2 + \cdots > 0$$

- Assume small η, can ignore all terms $O(\eta^2)$ and smaller:

$$g(x_0+\eta) - g(x_0) \approx g'(x_0)\cdot\eta \geq 0$$

- this must be true for any η !

- This can only be true if $\boxed{g'(x_0) = 0}$

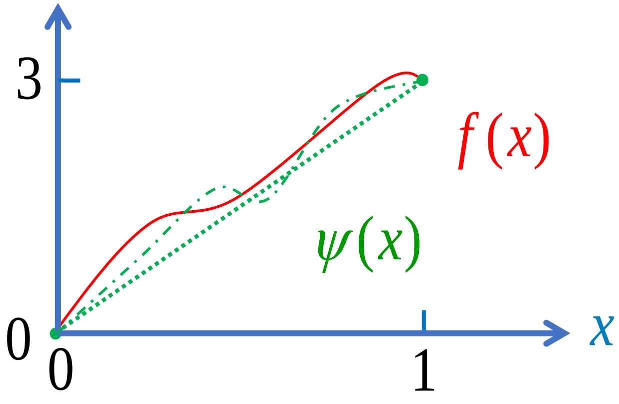
- **We will follow a similar logic for finding a minimum of a functional**

$$U(\psi) = \int_0^1 \left[\left(\frac{d\psi}{dx} \right)^2 + \psi^2 \right] dx$$

$f(0) = 0$; $f(1) = 3$

Let trial function $\psi(x) = f(x) + \eta(x)$

$\eta(x)$ is a “small” variation of $\psi(x)$ from $f(x)$.



Force all $\psi(x)$ to obey the (Dirichlet) BCs:

$$\eta(0) = \eta(1) = 0$$

Substitute ψ into $U(\psi)$…

$$U(f + \eta) = \int_0^1 \left[\left(\frac{df}{dx} \right)^2 + 2\frac{df}{dx}\frac{d\eta}{dx} + \left(\frac{d\eta}{dx} \right)^2 + f^2 + 2f\eta + \eta^2 \right] dx$$

Next recall that $U(f) = \int_0^1 \left[\left(\frac{df}{dx} \right)^2 + f^2 \right] dx$ is the *minimum U,*

so $\delta U = U(f + \eta) - U(f) = \int_0^1 \left[2\frac{df}{dx}\frac{d\eta}{dx} + 2f\eta + O(\eta^2) \right] dx$ is > 0.

$$\delta U = U(f+\eta) - U(f) = \int_0^1 \left[2\frac{df}{dx}\frac{d\eta}{dx} + 2f\eta + O(\eta^2) \right] dx > 0$$

compare with:

$$g(x_0+\eta) - g(x_0) = g'(x_0)\cdot\eta + \frac{1}{2}g''(x_0)\cdot\eta^2 + \cdots > 0$$

- Assume small $\eta(x)$, and hence require:

$$\boxed{\delta U \approx \int_0^1 \left[2\frac{df}{dx}\frac{d\eta}{dx} + 2f\eta \right] dx \geq 0}$$

- To proceed, we want to express this integral in the form η ·something

- For that, we are going to use:

  - in 1D case (like this example): integration by parts

  - 2D, 3D: divergence or Green's theorem

$$2\int_0^1 \left[\frac{df}{dx}\frac{d\eta}{dx} + f\,\eta \right] dx = 0$$

Here

$$\int_0^1 \frac{df}{dx}\frac{d\eta}{dx}\,dx = \underbrace{\left[\frac{df}{dx}\eta \right]_0^1}_{=\,0 \text{ as } \eta(0)\,=\,\eta(1)\,=\,0} - \int_0^1 \eta \frac{d^2 f}{dx^2}\,dx$$

Our minimising condition is $-2\int_0^1 \eta \left(\frac{d^2 f}{dx^2} - f \right) dx = 0.$

This can only be true for all variations $\eta(x)$ if $\boxed{\frac{d^2 f}{dx^2} - f = 0}$

"Find $f(x)$ that minimises $U[\psi]$ given $f(0) = 0$, $f(1) = 3$."

$$U[\psi] = \int_0^1 \left[\left(\frac{d\psi}{dx} \right)^2 + \psi^2 \right] dx$$

is equivalent to

"Find $f(x)$ satisfying $\frac{d^2 f}{dx^2} - f = 0$ given $f(0) = 0$, $f(1) = 3$."

Reduction to a numerical problem

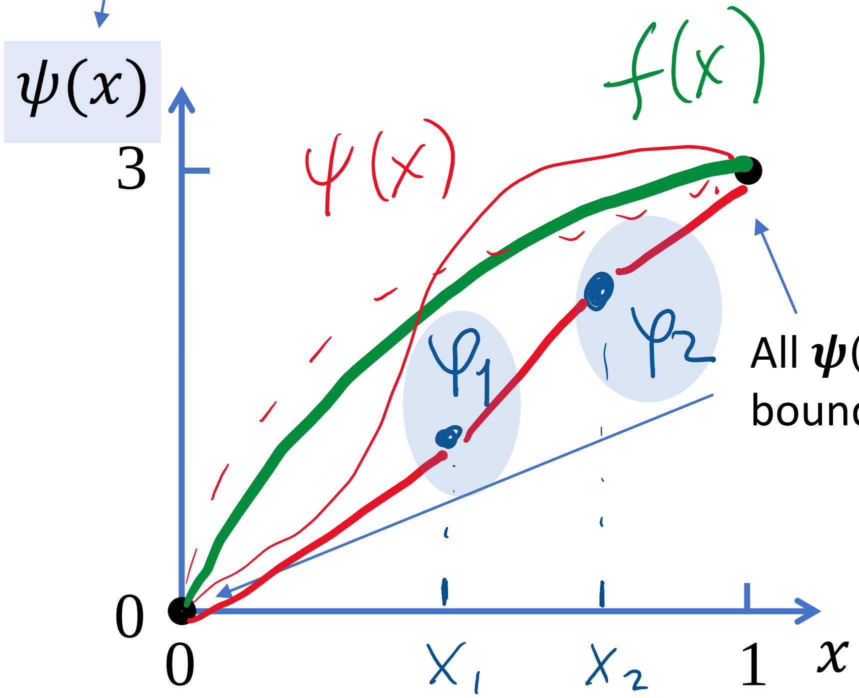
$$U[\psi] = \int_0^1 \left[\left(\frac{d\psi}{dx} \right)^2 + \psi^2 \right] dx$$

“Find $f(x)$ that minimises $U[\psi]$ given $f(0) = 0, f(1) = 3.$”

We are going to try different $\boldsymbol{\psi(x)}$, which **are approximations** to the **true minimizing function f(x)**

We will constrain our search by looking at piecewise-linear functions with nodes at some fixed points $x = x_1, x_2, \ldots$

Variation of $\psi(x)$ thus **reduces to variation of the discrete set** of its values at the nodal points $\psi(x_1) = \phi_1, \psi(x_2) = \phi_2, \ldots$



All $\boldsymbol{\psi(x)}$ should satisfy the same boundary conditions as $f(x)$

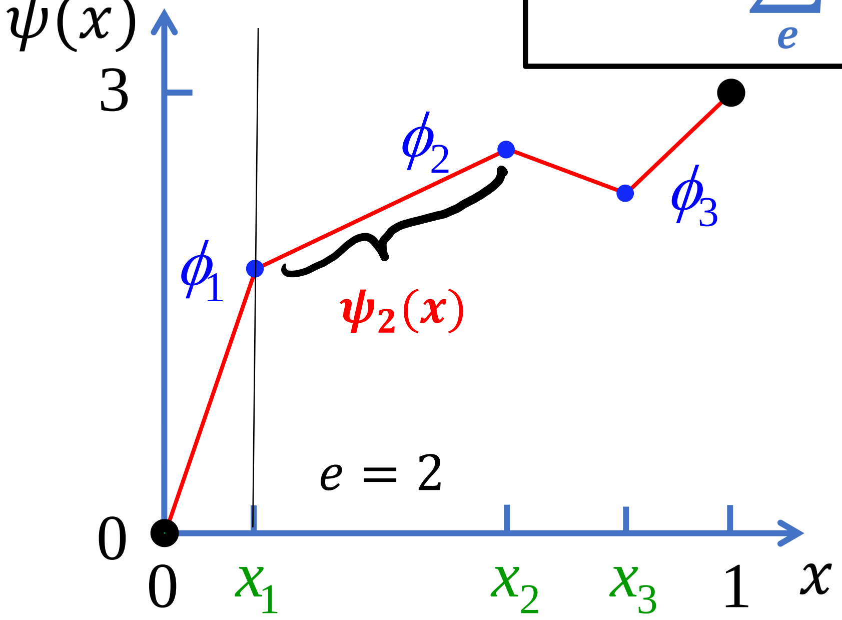
Reduction to a numerical problem

$$U[\psi] = \int_0^1 \left[\left(\frac{d\psi}{dx}\right)^2 + \psi^2\right] dx$$

"Find $f(x)$ that minimises $U[\psi]$ given $f(0) = 0, f(1) = 3$."

- Choose position of N *nodes* $x_1, x_2 \ldots$ no need to space equally
- Regions between nodes are *elements* $e = 1,2,3 \ldots$
- Value of trial function at the nodes is $\phi(x_1), \phi(x_2) \ldots = \phi_1, \phi_2 \ldots$
- Interpolate between nodes → elemental trial functions $\psi_e(x)$
- Overall trial function: $\boxed{\psi(x) = \sum_e \psi_e(x)}$

0 outside element e



Vary ψ by varying the ϕ_i

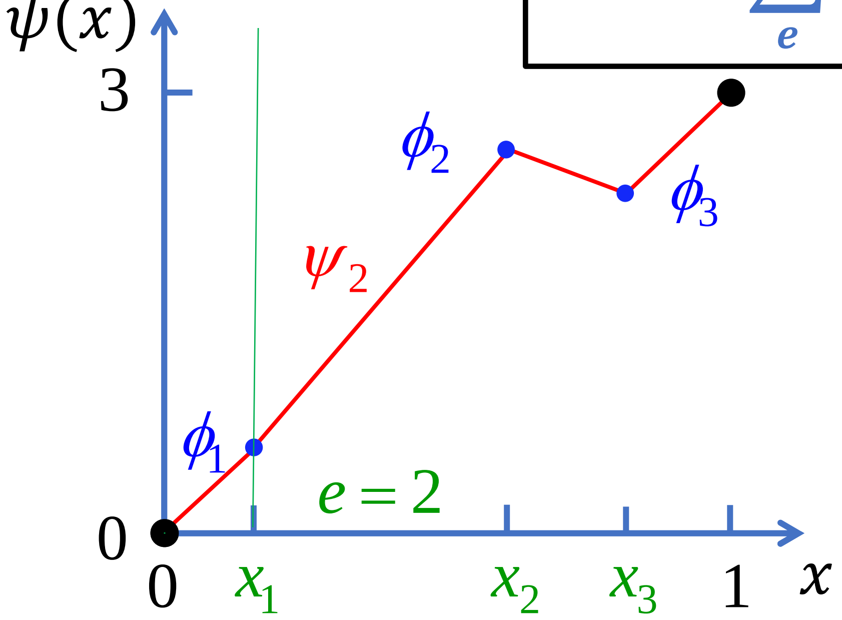
Reduction to a numerical problem

$$U[\psi] = \int_0^1 \left[\left(\frac{d\psi}{dx} \right)^2 + \psi^2 \right] dx$$

"Find $f(x)$ that minimises $U[\psi]$ given $f(0) = 0, f(1) = 3$."

- Choose position of N *nodes* $x_1, x_2 \ldots$ no need to space equally
- Regions between nodes are *elements* $e = 1,2,3 \ldots$
- Value of trial function at the nodes is $\phi(x_1), \phi(x_2) \ldots = \phi_1, \phi_2 \ldots$
- Interpolate between nodes → elemental trial functions $\psi_e(x)$
- Overall trial function: $\boxed{\psi(x) = \sum_e \psi_e(x)}$

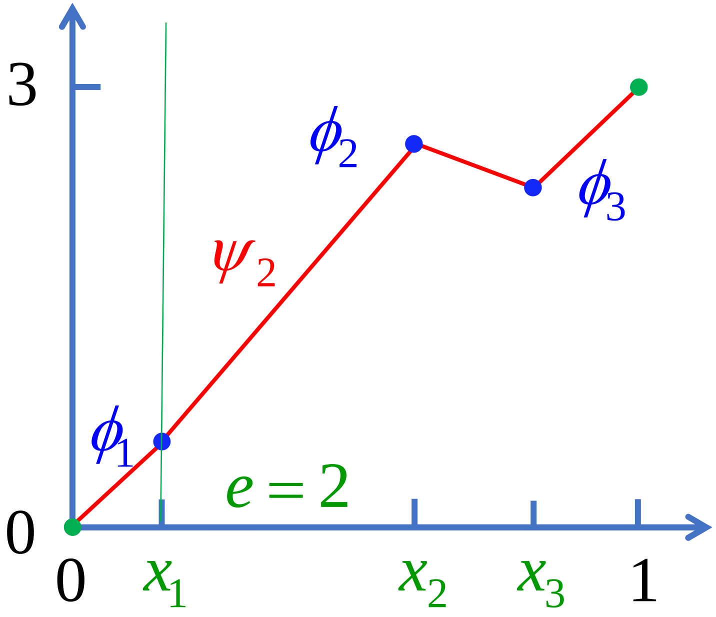
0 outside element e



Vary ψ by varying the ϕ_i

Note:

- $\psi(x)$ is continuous $\quad \psi(x) = \sum_e \psi_e(x)$
- $\psi(x)$ is defined at all x
- Discretisation error arises as not all $\psi(x)$ can be represented



- Overall functional $U[\psi]$ is split into corresponding segments:

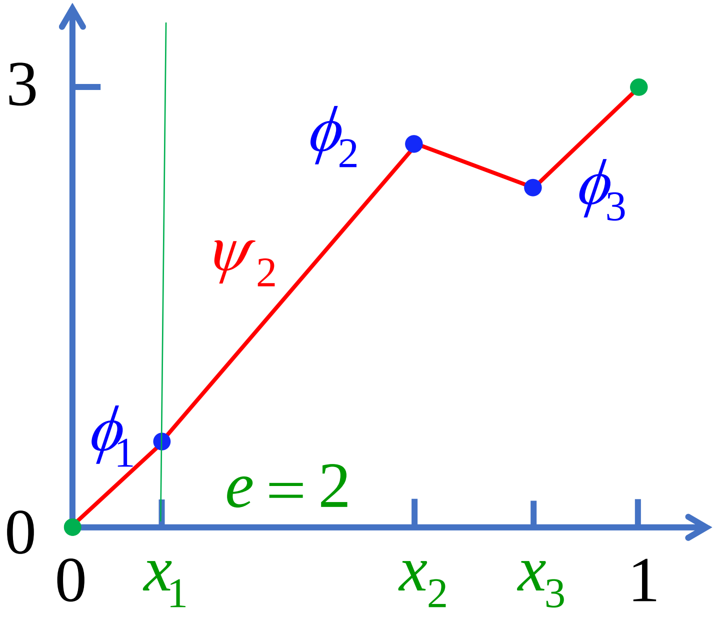
$$U[\psi] = \int_0^1 \left[\left(\frac{d\psi}{dx} \right)^2 + \psi^2 \right] dx \ = \int_0^{x_1} \left[\left(\frac{d\psi_1}{dx} \right)^2 + \psi_1^2 \right] dx + \int_{x_1}^{x_2} \left[\left(\frac{d\psi_2}{dx} \right)^2 + \psi_2^2 \right] dx$$

$$+ \int_{x_2}^{x_3} \left[\left(\frac{d\psi_3}{dx} \right)^2 + \psi_3^2 \right] dx + \int_{x_3}^{x_4} \left[\left(\frac{d\psi_4}{dx} \right)^2 + \psi_4^2 \right] dx$$

⇒ $$\boxed{U(\psi) = \sum_{e=1}^{4} U(\psi_e)}$$

Note:

- $\psi(x)$ is continuous $\qquad \boldsymbol{\psi(x)} = \sum_{e} \boldsymbol{\psi_e(x)}$
- $\psi(x)$ is defined at all x
- Discretisation error arises as not all $\psi(x)$ can be represented
- Overall $U(\psi) = \sum_{e=1}^{4} U(\psi_e)$



Our aim now is to express U only in terms of node values ϕ_i .

Linear interpolation: $\psi_2(x) = \phi_1 + \dfrac{\phi_2 - \phi_1}{x_2 - x_1}(x - x_1) = mx + c$

where $c = \phi_1 - (\phi_2 - \phi_1)lx_1$

$m = l(\phi_2 - \phi_1); \quad c = \phi_1 - lx_1; \quad l = (x_2 - x_1)^{-1}.$

$$\psi_2(x) = mx + c$$

Consider e.g. $U[\psi_2] = \int_{x_1}^{x_2} \left[\left(\frac{d\psi_2}{dx} \right)^2 + \psi_2^2 \right] dx$

We have $\left(\psi_2'\right)^2 = m^2 = l^2(\phi_2^2 - 2\phi_1\phi_2 + \phi_1^2)$ and $\psi_2^2 = m^2x^2 + 2mcx + c^2$.

Now we can evaluate all definite integrals. Eg $\int_{x_1}^{x_2} x^2\,dx = \frac{1}{3}(x_2^3 - x_1^3)$.

Once that's done, $U(\psi_2)$ can be written as a sum of terms that are *quadratic*, *linear* and *constant* in the node values:

$$U(\psi_2) = \tfrac{1}{2}(k_{11}\phi_1\phi_1 + k_{12}\phi_1\phi_2 + k_{22}\phi_2\phi_2) - r_1\phi_1 - r_2\phi_2 + g_2$$

Do the same for all elements. Some ϕ_i contribute to >1 element. Overall we get

$$U(\psi) = \sum_e U(\psi_e) = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} K_{ij}\phi_i\phi_j - \sum_{i=1}^{3} R_i\phi_i + \sum_{e=1}^{4} g_e$$

Minimise *U*

$$U(\psi) = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} K_{ij}\phi_i\phi_j - \sum_{i=1}^{3} R_i\phi_i + \sum_{e=1}^{4} g_e$$

To do this, need $\frac{\partial U}{\partial \phi_i} \quad \forall i$

Compare $u(x) = \frac{1}{2}ax^2 - bx + c.$

at minimum $\frac{du}{dx} = ax - b = 0$. Same idea here…

$$\frac{\partial U}{\partial \phi_i} = \sum_{j=1}^{3} K_{ij}\phi_j - R_i = 0$$

This is a standard matrix problem $\boxed{\hat{K}\vec{\phi} = \vec{R}}$

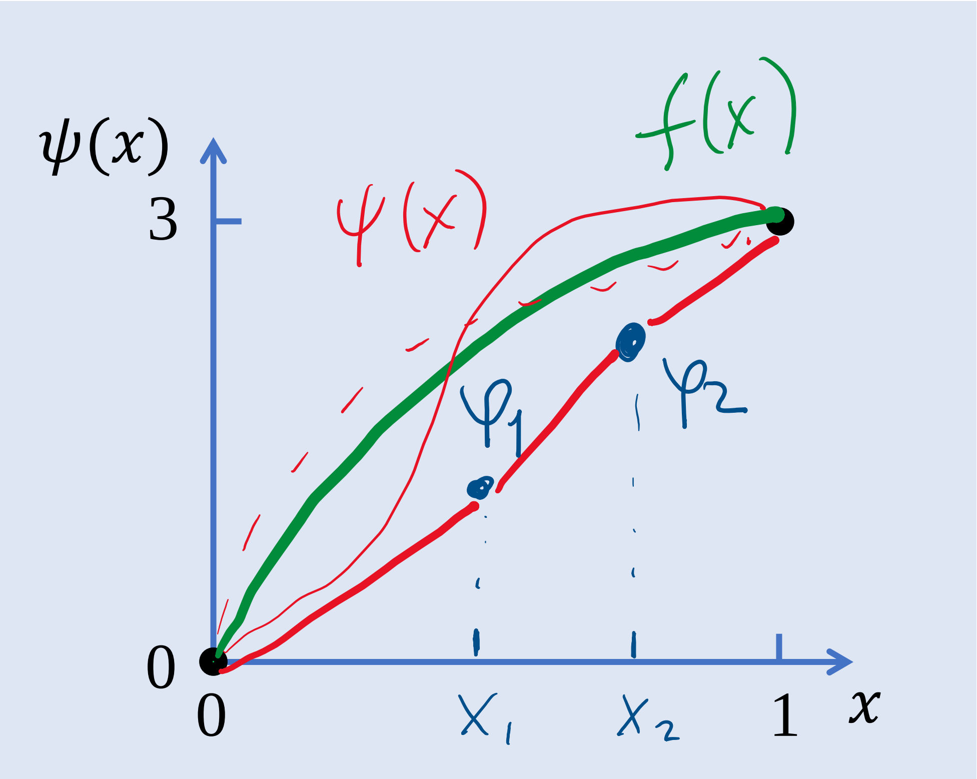
Use linear algebra algorithms to obtain approximate *(given segmentation, choice of interpolants etc)* variational solution

$$\phi_i \approx f(x_i)$$

Let us summarize the entire process *(in case you are already lost)*

- We wanted to solve: $\frac{d^2 f}{dx^2} - f = 0$ **with BC** $f(0) = 0, f(1) = 3$
- Stated (derived) it is equivalent to minimization of $U[\psi] = \int_0^1 \left[\left(\frac{d\psi}{dx}\right)^2 + \psi^2\right] dx$

with the constrain $\psi(0) = 0, \psi(1) = 3$



- Introduced piecewise-linear functions $\psi_e(x)$ with nodes at some set of points x_j
- Expressed $U[\psi]$ through values $\boldsymbol{\phi_i = \psi(x_i) \approx f(x_i)}$ at the nodes – appears to be a quadratic form

$$U(\psi) = \frac{1}{2}\sum_i \sum_j K_{ij}\phi_i\phi_j - \sum_i R_i\phi_i + \sum_i g_i$$

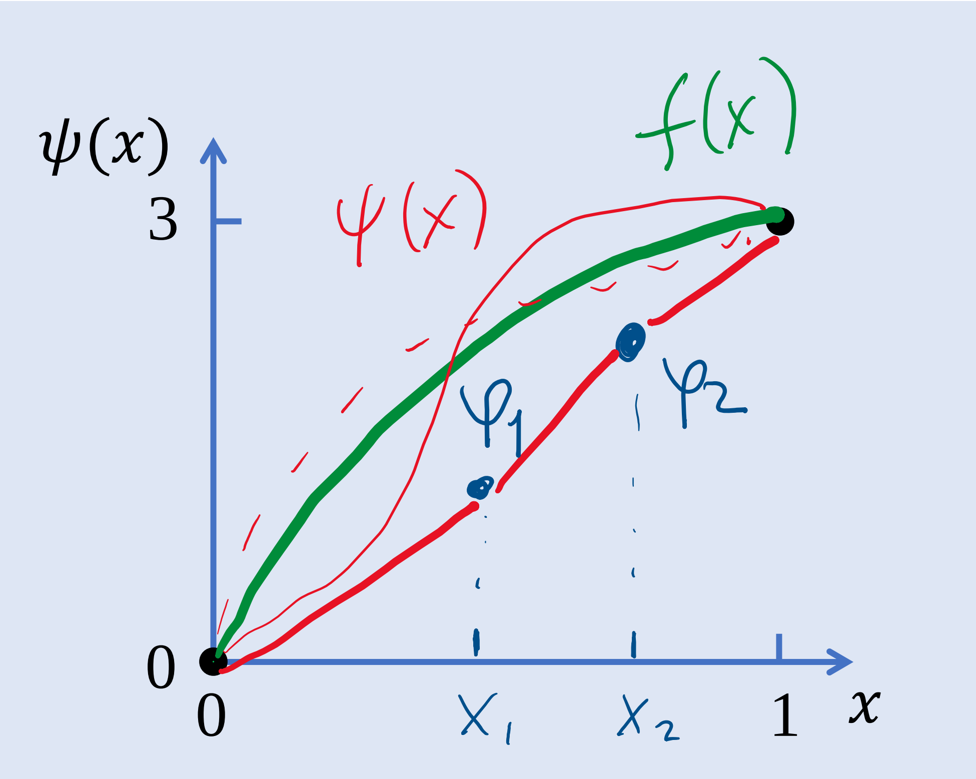
***Note:** think why quadratic?*

- Minimize the quadratic form => linear set of equations for $\boldsymbol{\phi_i}$

What happened to the boundary conditions?

- We wanted to solve: $\frac{d^2 f}{dx^2} - f = 0$ **with BC** $f(0) = 0, f(1) = 3$

- BCs resulted in the constrain for the minimization procedure: $\psi(0) = 0, \psi(1) = 3$



- When linear segments $\psi_e(x)$ are introduced, **the boundary values are included** in the expressions for **the first and the last segments**, respectively

- These values eventually propagated into the coefficients of the resulting quadratic form

$$U(\psi) = \frac{1}{2}\sum_i \sum_j K_{ij}\phi_i\phi_j - \sum_i R_i\phi_i + \sum_i g_i$$

This is similarly to the FDM method, where BCs are "absorbed" in the r.h.s. vector of the resulting matrix equation.

What if we have other boundary conditions?

- BCs are important at this step of the minimization of $U[\psi]$ *(slide 8)*

$$\int_0^1 \frac{df}{dx}\frac{d\eta}{dx}dx = \underbrace{\left[\frac{df}{dx}\eta\right]_0^1} - \int_0^1 \eta \frac{d^2 f}{dx^2}dx$$

This term will be zero if: (a) $\eta = 0$ (i.e. Dirichlet BCs)

or (b) $\frac{df}{dx} = 0$ (i.e. Neumann BCs)

- **For all such BCs the resulting functional $U[\psi]$ will be the same**

- If we have a different Neumann BC: $\frac{df}{dx} = c$, the functional needs to be changed, but this is still doable

Summary:

- A few things we have learnt today about the Finite Element Method (FEM)

  - *It is based on the idea of minimization of a certain global property (functional) – **a conceptually different approach** to solving PDEs.*

  - ***It requires some effort to setup!***

  - *The main advantage of FEM over FDM is: **FEM does not require a fixed grid***

  - *Compatible with different boundary conditions*

- You would not normally use it for simple 1D problems, but it is the 2+ dimensional problems where FEM becomes very useful – there's more to come on next lecture…

Lecture 17:

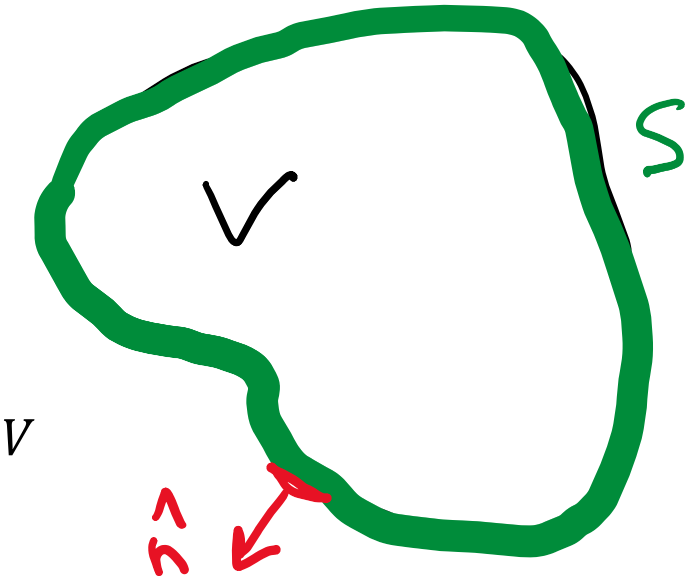
The Finite Element Method: part 2

Variational method in 2+ dimensions

- Consider e.g. the Poisson equation in 3D:

  (electric potential due to charge density
  ρ distribution in a volume)

$$\nabla^2\Phi = -\frac{\rho(\vec{r})}{\epsilon_0}$$



- Need to solve it within a certain finite volume V

- Boundary conditions: set Φ or $\frac{\partial\Phi}{\partial n}$ along the **surface enclosing the volume**

- Could be $\Phi_S = f(\vec{r})$ *(e.g. fixed distribution of charges along the surface)*
(Dirichlet BCs)

- Could be $\frac{\partial\Phi}{\partial n} = \hat{n}\cdot\nabla\Phi = 0$ *(no charge flux through the surface)*
(Neumann BCs)

- Could be a combination of both: Dirichlet BCs in some surface regions, and Neumann elsewhere

Functional minimisation to solve a PDE

$$\nabla^2 \Phi = -\frac{\rho}{\varepsilon_0}$$

We shall demonstrate, that the functional we need to minimise is

$$U(\psi) = \int_V \left[\frac{\varepsilon_0}{2}(\nabla\psi)^2 - \rho\psi\right] dV \quad (1)$$

We define: $\Phi(\vec{r}) =$ function that minimises (1),

$\psi(\vec{r}) = \Phi(\vec{r}) + \eta(\vec{r}) =$ trial function

$\delta U(\eta) = U(\psi) - U(\Phi) =$ *increase* in U due to variation η

Substitute $\psi = \Phi + \eta$ into (1):

$$U(\psi) = \int_V \left[\frac{\varepsilon_0}{2}((\nabla\Phi)^2 + 2\nabla\Phi\cdot\nabla\eta + (\nabla\eta)^2) - \rho(\Phi+\eta)\right] dV,$$

So to 1st order in η :

$$\boxed{\delta U(\eta) = \int_V [\varepsilon_0 \nabla\Phi\cdot\nabla\eta - \rho\eta]\, dV = 0}$$

$$\delta U(\eta) = \int_V [\varepsilon_0 \nabla\Phi \cdot \nabla\eta - \rho\eta]\, dV = 0$$

- We want to re-write this result as an integral of η ·something
- Manipulate first term in the integral using vector identity:

$$\boldsymbol{\nabla} \cdot (\boldsymbol{\phi}\vec{\boldsymbol{A}}) = \boldsymbol{\phi}(\boldsymbol{\nabla} \cdot \vec{\boldsymbol{A}}) + \vec{\boldsymbol{A}} \cdot \boldsymbol{\nabla}\boldsymbol{\phi}$$

use $\vec{A} = \nabla\,\Phi$ & $\phi = \eta$: $\quad \nabla\,\Phi \cdot \nabla\eta = \nabla \cdot (\eta\nabla\Phi) - \eta \cdot \nabla^2\Phi$

$$\Rightarrow \quad \delta U(\eta) = \varepsilon_0 \int_V \nabla \cdot (\eta\nabla\Phi) dV - \int_V \eta\,(\varepsilon_0\nabla^2\Phi + \rho) dV$$

- Use divergence theorem to manipulate the first integral:

$$\int_V \nabla \cdot (\eta\nabla\Phi) dV = \oint_S \eta(\underbrace{\hat{n} \cdot \nabla\Phi}) dS = \oint_S \eta \cdot \frac{\partial\Phi}{\partial n}\, dS$$

Integral over the surface enclosing the volume

Normal derivative

$$\delta U(\eta) = \varepsilon_0 \oint_S \eta \cdot \frac{\partial \Phi}{\partial n}\, dS - \int_V \eta\, (\varepsilon_0 \nabla^2 \Phi + \rho) dV = 0$$

- The first integral is zero if:

$\boldsymbol{\eta = 0}$ at the surface (Dirichlet boundary conditions)

or $\boldsymbol{\frac{\partial \Phi}{\partial n} = 0}$ at the surface (Neumann boundary conditions)

- In our problem we have either of the two conditions everywhere on the surface.

⇒ $$\delta U(\eta) = \int_V \eta\, (\varepsilon_0 \nabla^2 \Phi + \rho) dV = 0$$

- But this is only **true for all variations** η if $\boxed{\nabla^2 \Phi = -\rho/\epsilon_0}$ (1)

- Hence solution of Eq. (1) with either Dirichlet or Neumann BCs **is equivalent to minimization of**

$$U(\psi) = \int_V \left[\frac{\varepsilon_0}{2} (\nabla \psi)^2 - \rho \psi\right] dV \quad (2)$$

Some important observations so far:

- For a given PDE, need to find the appropriate functional $U(\psi)$.

  - Calculus of Variations (and some practice) give wide range

  - There are few "key" equations in physics (see Lecture 1).
  No need to re-invent the wheel!

- The derived functional is compatible with the two most common types of boundary conditions:

  - Dirichlet condition $\Phi_S = f$ (is implemented numerically in a similar way to FDM: i.e. you make sure the values of the function are fixed at the respective boundaries)

  *- Neumann condition $\frac{\partial \Phi}{\partial n} = 0$ (**is implemented automatically!** i.e. you do not need to do anything at the stage of numerical implementation)*

- Other boundary conditions (e.g. $\frac{\partial \Phi}{\partial n} = g$) can also be implemented, will need adjustments to the functional U

Numerical implementation of FEM: a 2D example

The idea is similar to 1D case discussed previously, i.e.:

- Need to minimize a functional $U[\psi]$, e.g. for the Poisson equation

$$U(\psi) = \int \left[\frac{\varepsilon_0}{2}(\nabla\psi)^2 - \rho\psi\right] dS$$

Note: *in 2D the functional is a surface integral, and the boundary is the curve enclosing the domain*

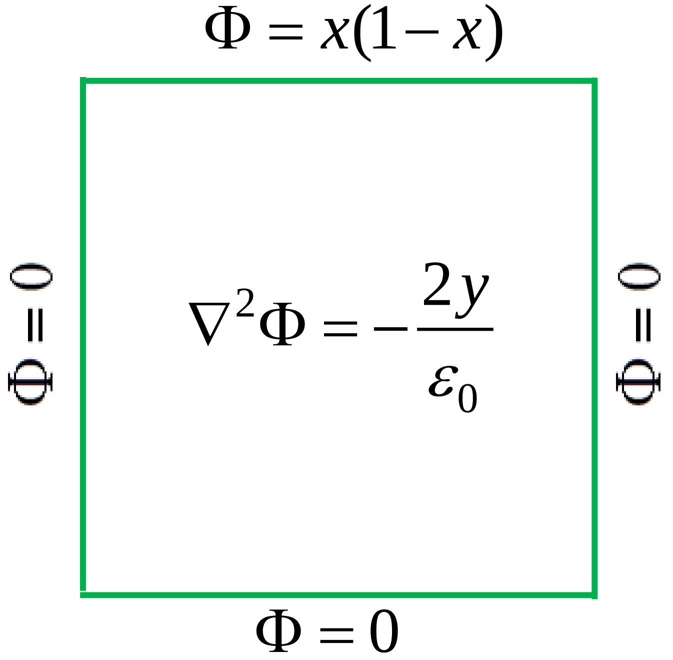
- Sub-divide the domain/volume of computation into elements. **We have full freedom to choose the shape and size** of each individual element *=> use this to make the resulting mesh denser in regions where you expect stronger field variation, and coarser otherwise. This will save you a lot of computational effort (especially in 2D and 3D)*

- Use interpolation (e.g. linear) to express ψ *[and ultimately the functional $U(\psi)$]* via the values of the trial function at the nodes of the mesh $\psi(\vec{r}_i) = \phi_i$

- Minimization of $U(\psi)$: derive the resulting condition for ϕ_i and solve it

Note: *for any linear PDE the resulting functional will be quadratic in ψ. Hence it will be come a quadratic form when expressed via ϕ_i. Hence the final condition is a linear problem $\widehat{M}\vec{\boldsymbol{\phi}} = \vec{\boldsymbol{S}}$*

PDE example (2d unit square; Dirichlet problem)

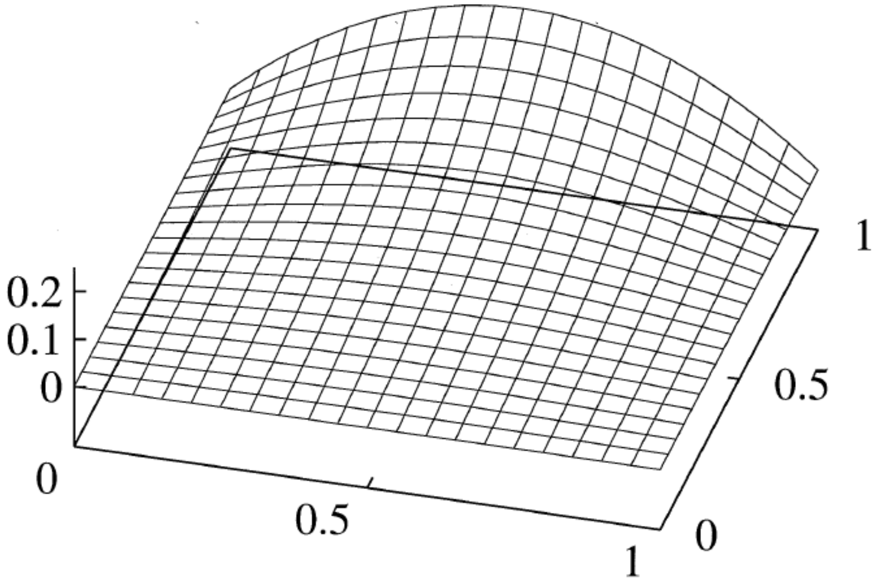


$$\Phi = x(1-x)$$

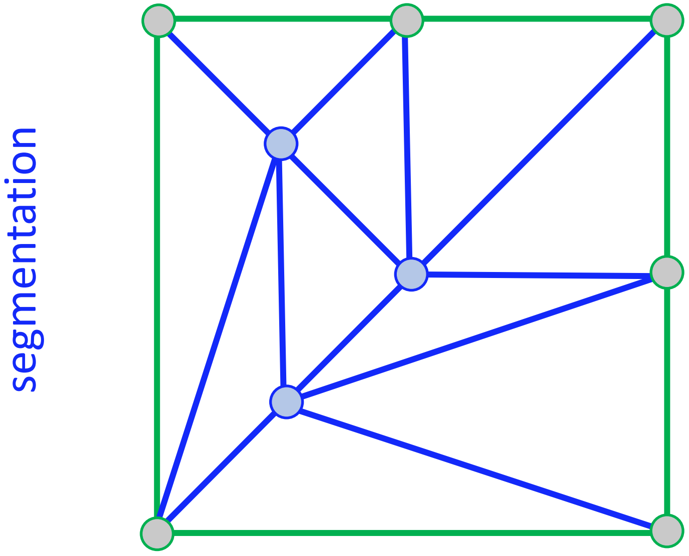
$$\nabla^2\Phi = -\frac{2y}{\varepsilon_0}$$

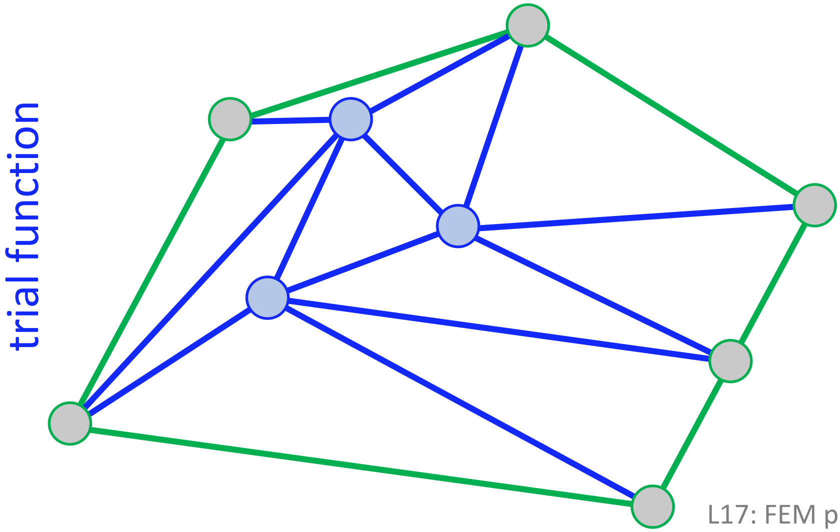
$$\Phi = 0$$

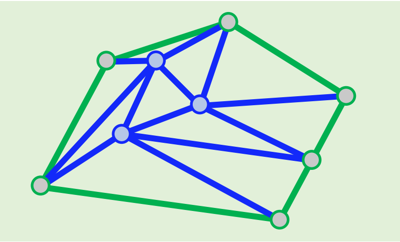
Exact sol^{n}



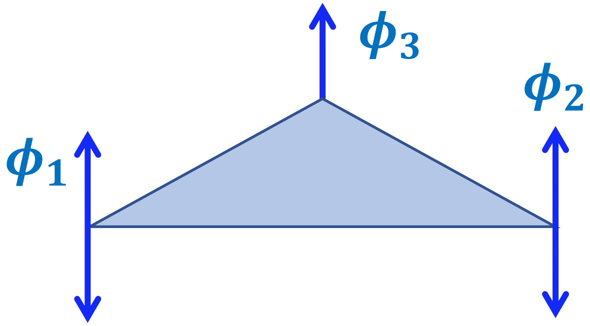
Use simplest scheme: triangular elements, linear interpolants







- Elements are flat triangular planes, nodes (x_1, y_1), (x_2, y_2), (x_3, y_3) at corners



- Variation of $\psi(x, y)$ by adjusting the values ϕ_i

***Note:** each node is usually shared between several adjacent elements*

Again $\psi(x,y) = \sum_{(e)} \psi^{(e)}(x,y)$ and $U(\psi) = \sum_{(e)} U(\psi^{(e)})$

In 2d, the interpolating function describing a triangle with nodes (x_1, y_1, ϕ_1), (x_2, y_2, ϕ_2), and (x_3, y_3, ϕ_3) is:

$$\boxed{\psi^{(e)}(x,y) = \sum_{i=1}^{3} N_i^{(e)}(x,y)\phi_i^{(e)}}$$

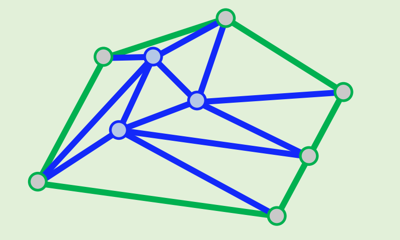
where

$$N_1^{(e)} = \frac{1}{2A^{(e)}}[(x_2y_3 - x_3y_2) + (y_2 - y_3)x - (x_2 - x_3)y]$$

$$N_2^{(e)} = \frac{1}{2A^{(e)}}[(x_3y_1 - x_1y_3) + (y_3 - y_1)x - (x_3 - x_1)y]$$

$$N_3^{(e)} = \frac{1}{2A^{(e)}}[(x_1y_2 - x_2y_1) + (y_1 - y_2)x - (x_1 - x_2)y]$$

A^e - area of the triangle



For this problem

$$U(\psi^{(e)}) = \int_{(e)} dxdy \left[\frac{1}{2}\varepsilon_0 \left(\frac{\partial\psi^{(e)}}{\partial x}\right)^2 + \frac{1}{2}\varepsilon_0 \left(\frac{\partial\psi^{(e)}}{\partial y}\right)^2 - 2y\psi^{(e)}\right]$$

Interpolants are linear in the ϕ_i, and terms in U are similar to 1d example. Once the x- and y-dependence has been integrated out

$$U(\psi) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} K_{ij}\phi_i\phi_j - \sum_{i=1}^{N} R_i\phi_i.$$

Again differentiate to find minimising trial function

$$\frac{\partial U}{\partial \phi_i} = \sum_{j=1}^{N} K_{ij}\phi_j - R_i = 0,$$

Or $\hat{K}\vec{\phi} = \vec{R}$ - a standard matrix problem.

FE programs are not as easily written as FD

Need sensible segmentation, integrals over irregular elements etc

They tend to come in packages, in which underlying variational scheme is usually obscured. Documentation deals with "matrix elements" $(K_{ij}; R_i)$

Why bother?

(1) Segmentation **easily adapted to geometry** – pack elements where most needed – no need for regular grids.

(2) Neumann conditions are "natural".

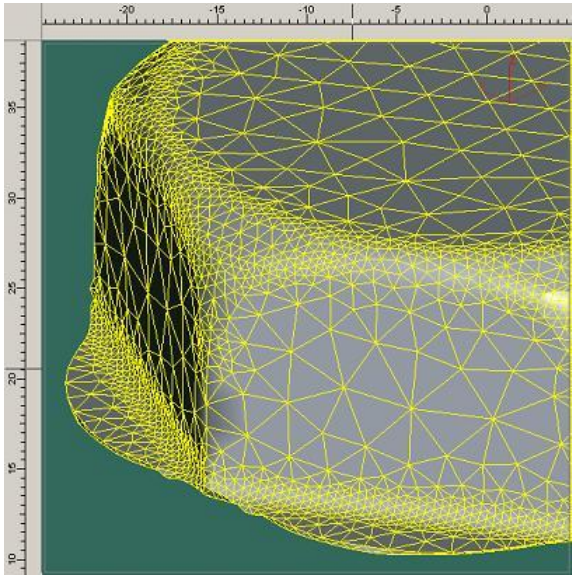
More advanced schemes use non-linear interpolants, curved elements etc.

Summary:

- The Finite Element Method (FEM) requires some effort to implement. But the benefits are:

  - *Advanced options for mesh: no specific requirements/constrains as for the size or shape of individual elements*

  - *Much easier handling of Neumann boundary conditions*



- Many physical problems are eventually reduced to a few well-known equations: for many applications you can use available FEM solvers *(but most of them are not free!)*

Lecture 18:

Initial Value Problem parabolic PDEs

finite difference methods

General problem: a PDE with a combination of time and space derivatives, e.g.

$$i\hbar \frac{\partial}{\partial t}\Psi(\vec{r},t) = \left(-\frac{\hbar^2}{2m}\nabla^2 + V(\vec{r})\right)\Psi(\vec{\mathrm{r}},\mathrm{t})$$ - Schrödinger equation (parabolic)

$$\frac{\partial^2}{\partial t^2}u(x,t) = c^2\frac{\partial^2}{\partial x^2}u(x,t) + f(x,t)$$ - wave equation (hyperbolic)

- Auxiliary conditions: combination of **IVP** (time) and **BVP** (space):

e.g. $u(x,t=0) = U_0(x),$ $\quad u(0,t) = 0,$

$\frac{\partial}{\partial t}u(x,t=0) = V_0(x),$ $\quad u(L,t) = 0.$

- Time and space are treated differently: such systems are often referred to as (1+N)-dimensional problems *(e.g. (1+3)D Schrödinger equation)*

- **We will focus on (1+1)D linear parabolic problems**

e.g. $$i\frac{\partial}{\partial t}\mathrm{u}(x,t) = \left(-\frac{\partial^2}{\partial x^2} + V(x)\right)\mathrm{u}(x,\mathrm{t})$$

Semi-Analytical method

$$i\frac{\partial}{\partial t}\mathrm{u}(x,t)=\left(-\frac{\partial^2}{\partial x^2}+V(x)\right)\mathrm{u}(x,\mathrm{t})$$

- Eigenstates form a basis set for expansion (similar to Fourier transforms, but more generic):

$$\mathrm{u}(x,t)=\mathrm{e}_\omega(x)\exp(-\mathrm{i}\omega t)$$

- Stationary analysis

$$\omega\cdot\mathrm{e}_\omega(x)=\left(-\frac{\partial^2}{\partial x^2}+V(x)\right)\mathrm{e}_\omega(x)$$

- Solve as an eigenvalue problem *(for a given potential, solve e.g. numerically by discretising x – see Lecture 13)*

$$u(x,t=0)=U_0(x)=\int C_\omega e_\omega(x)d\omega$$

- Expand the initial solution into the basis set

$$C_\omega=\int u(x)e_\omega^*(x)d\omega$$

(in numerics, all integrals will be replaced by the sums over N normalised eigen-states)

- Hence at any time t the solution is given by:

$$\boxed{u(x,t)=\int C_\omega e_\omega(x)e^{-i\omega t}d\omega}$$

Semi-Analytical method

$$i\,\frac{\partial}{\partial t}\mathrm{u}(x,t)=\left(-\frac{\partial^2}{\partial x^2}+V(x)\right)\mathrm{u}(x,\mathrm{t})$$

- Numerical **errors only due to discretisation of x** (i.e. using finite basis set), but there is no discretisation of time

- Works only for **LINEAR** equations

- The method is **computationally expensive**:

$$\omega\cdot\mathrm{e}_\omega(x)=\left(-\frac{\partial^2}{\partial x^2}+V(x)\right)\mathrm{e}_\omega(x)$$

- Full eigenvalue problem (i.e. need to find ALL eigenvectors and eigenvalues). Computational time scales as $O(N^3)$

$$C_\omega=\int u(x)e_\omega^*(x)d\omega$$

- Compute all expansion coefficients: computational time ~$O(N^2)$

$$u(x,t)=\int C_\omega e_\omega(x)e^{-i\omega t}d\omega$$

- Resulting solution as a sum over all eigenstates: computational time ~$O(N^2)$

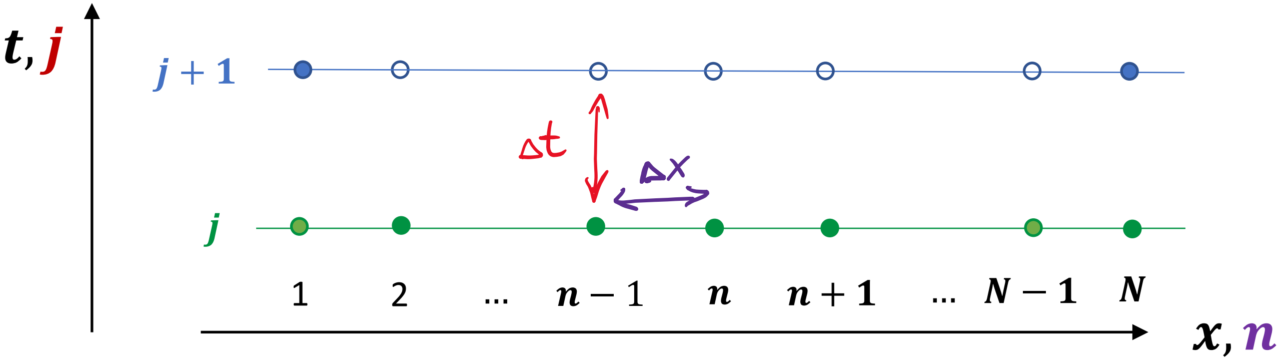
- **An elegant but slow method**

Finite-Difference methods: general idea

- Discretize time and space coordinates:

$x \rightarrow x_n = n \cdot \mathbf{\Delta x}$ - For **all spatial coordinates** we normally use **regular grid** *(such that CDA for all spatial derivatives can be applied)*

$t_{j+1} = t_j + \mathbf{\Delta t}$ - For **time** we can use schemes with **adaptive step.** *<u>Unless the regular grid is required</u> by a specific method (e.g. Leapfrog), we shall assume that <u>Δt can vary with each iteration</u>*



$\boldsymbol{t}, \boldsymbol{j}$
$j+1$
Δt
Δx
j
1
2
...
$n-1$
n
$n+1$
...
$N-1$
N
$\boldsymbol{x}, \boldsymbol{n}$

○ need to obtain

● fixed by BVP

● (for $j = 1$): fixed by IVP

(for $j > 1$): obtained at the previous iteration

Forward-Time Centred-Space method (FTCS)

$$i\,\frac{\partial}{\partial t}\mathrm{u}(x,t)=\left(-\frac{\partial^2}{\partial x^2}+V(x)\right)\mathrm{u}(x,\mathrm{t})$$

- Discretise time and space. Use FDA for time derivative, CDA for space:

$\boldsymbol{t=j\Delta t}\;, j=0,1,2,\ldots$

$\boldsymbol{x=\mathrm{n}\Delta x}\;, n=0,1,2,\ldots$

(e.g. solving on $t\in[0,T]$, $x\in[0,L]$ intervals)

$$\mathrm{u}(\boldsymbol{x_n},\boldsymbol{t_j})=u_{\boldsymbol{n}}^{(\boldsymbol{j})}$$

time index

space index

⇒ $$i\,\frac{1}{\Delta t}\left[\mathrm{u}_n^{(j+1)}-\mathrm{u}_n^{(j)}\right]=\frac{1}{(\Delta x)^2}\left(2u_n^{(j)}-u_{n+1}^{(j)}-u_{n-1}^{(j)}\right)+V_nu_n^{(j)}$$

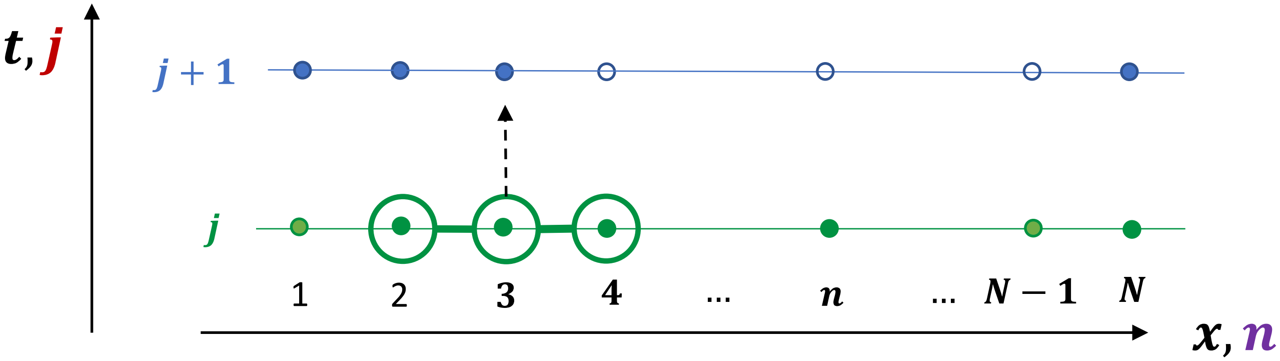
- Let $\boldsymbol{a=\frac{\Delta t}{(\Delta x)^2}}$, $b_n=i\Delta tV_n$

$$\boxed{\mathrm{u}_n^{(j+1)}=(1-i2a)u_n^{(j)}+ia\left(u_{n+1}^{(j)}+u_{n-1}^{(j)}\right)-b_nu_n^{(j)}}$$

Forward-Time Centred-Space method (FTCS)

$$i\,\frac{\partial}{\partial t}\mathrm{u}(x,t) = \left(-\frac{\partial^2}{\partial x^2} + V(x)\right)\mathrm{u}(x,\mathrm{t})$$

$$\boxed{\mathrm{u}_n^{(j+1)} = (1 - i2a)u_n^{(j)} + ia\left(u_{n+1}^{(j)} + u_{n-1}^{(j)}\right) - b_n u_n^{(j)}}$$



- Requires $O(N)$ calculations per iteration.

=> Computational time ~$O(N) \times number\_of\_iterations$

Forward-Time Centred-Space method (FTCS)

$$i\frac{\partial}{\partial t}\mathrm{u}(x,t) = \left(-\frac{\partial^2}{\partial x^2} + V(x)\right)\mathrm{u}(x,\mathrm{t})$$

$$\boxed{\mathrm{u}_n^{(j+1)} = (1-i2a)u_n^{(j)} + ia\left(u_{n+1}^{(j)} + u_{n-1}^{(j)}\right) - b_n u_n^{(j)}}$$

- **Stability?**

- Consider the case without a potential (i.e. $b_n = 0$), split the solution into exact and error: $u_n^{(j)} = \boldsymbol{V}_{\boldsymbol{n}}^{(j)} + \epsilon_n^{(j)}$. Substitute in the above equation and obtain for the error:

$$\epsilon_n^{(j+1)} = (1-i2a)\epsilon_n^{(j)} + ia\left(\epsilon_{n+1}^{(j)} + \epsilon_{n-1}^{(j)}\right)$$

- Use Discrete Fourier Transform for space coordinate:

$$\epsilon_n^{(j)} = \sum_{k=0}^{N-1} E_k^{(j)} \exp(iq_k n) \qquad q_k = \left(\frac{2\pi}{N}\right)k, k = 0,1, \dots N-1$$

Forward-Time Centred-Space method (FTCS)

$$\epsilon_n^{(j+1)} = (1-i2a)\epsilon_n^{(j)} + ia\left(\epsilon_{n+1}^{(j)} + \epsilon_{n-1}^{(j)}\right)$$

$$\epsilon_n^{(j)} = \sum_{k=0}^{N-1} E_k^{(j)} \exp(iq_k n) \qquad q_k = \left(\frac{2\pi}{N}\right)k, k = 0,1,\ldots N-1$$

- **Note:** $\epsilon_{n\pm1}^{(j)} = \sum_{k=0}^{N-1} E_k^{(j)} \exp\big(iq_k(n\pm1)\big) = \exp\big(iq_k(\pm1)\big)\cdot\epsilon_n^{(j)}$

- Substitute: $$E_k^{(j+1)} = (1-i2a)E_k^{(j)} + ia\left(e^{iq_k} + e^{-iq_k}\right)E_k^{(j)}$$ $$= [1 - i2a + i2a\cos(q_k)]E_k^{(j)} = \left[1 - i4a\sin^2\left(\frac{q_k}{2}\right)\right]E_k^{(j)}$$

$$\boxed{E_k^{(j+1)} = \left[1 - i4a\sin^2\left(\frac{q_k}{2}\right)\right]E_k^{(j)}}$$

- Stability: $$g_k = \left|1 - i4a\sin\left(\frac{q_k}{2}\right)\right| = \sqrt{1 + 16a^2\sin^4\left(\frac{q_k}{2}\right)} \geq \mathbf{1}$$

This scheme is unstable for any value of the discretisation parameter

$$\boxed{a = \frac{\Delta t}{(\Delta x)^2}}$$

Forward-Time Centred-Space method (FTCS)

$$i\,\frac{\partial}{\partial t}\mathrm{u}(x,t) = \left(-\frac{\partial^2}{\partial x^2} + V(x)\right)\mathrm{u}(x,\mathrm{t})$$

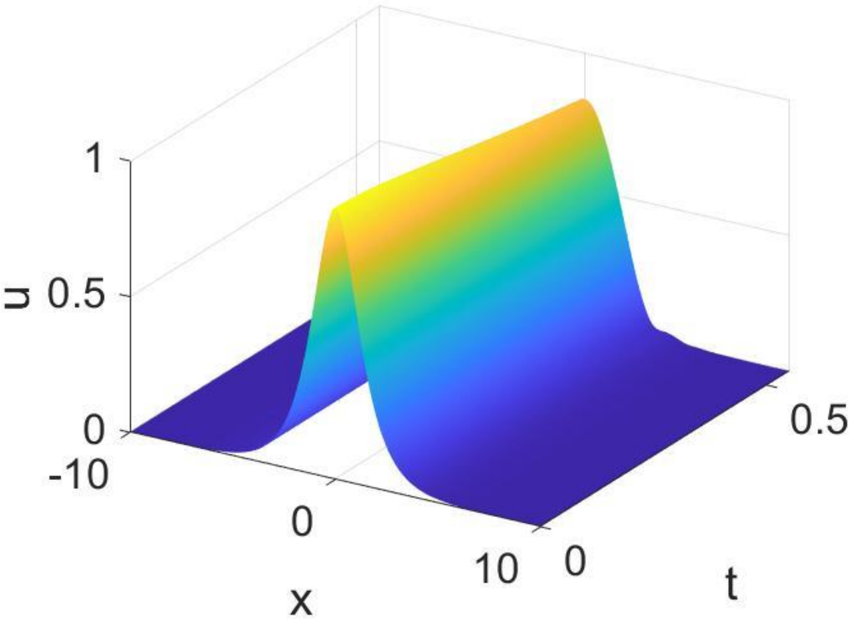
- Instability evolution:

Select $dx = 0.1$, $dt = 0.01$ (a =0.1)

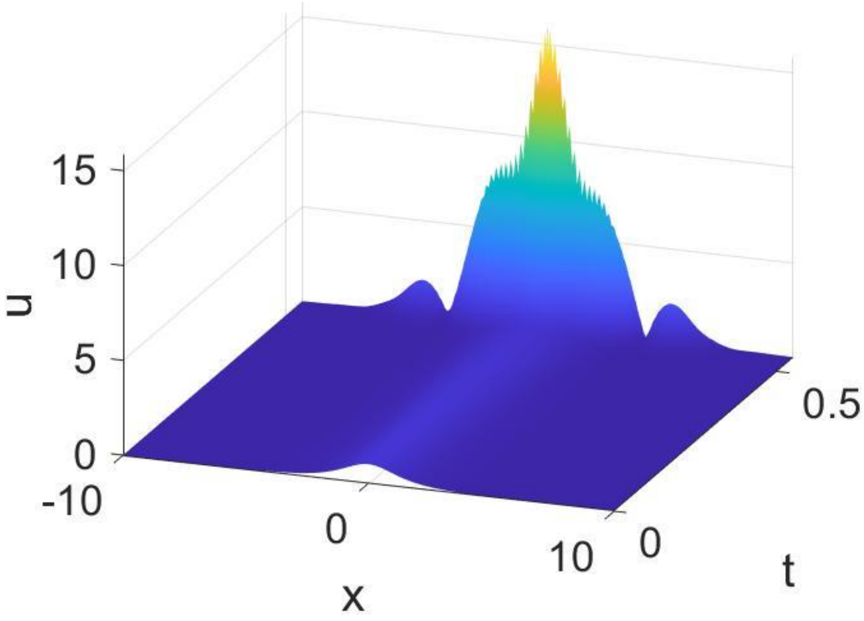
$V(x) = 0$

Initial condition: $u(x, t = 0) = \mathrm{sech}(x)$

Exact solution



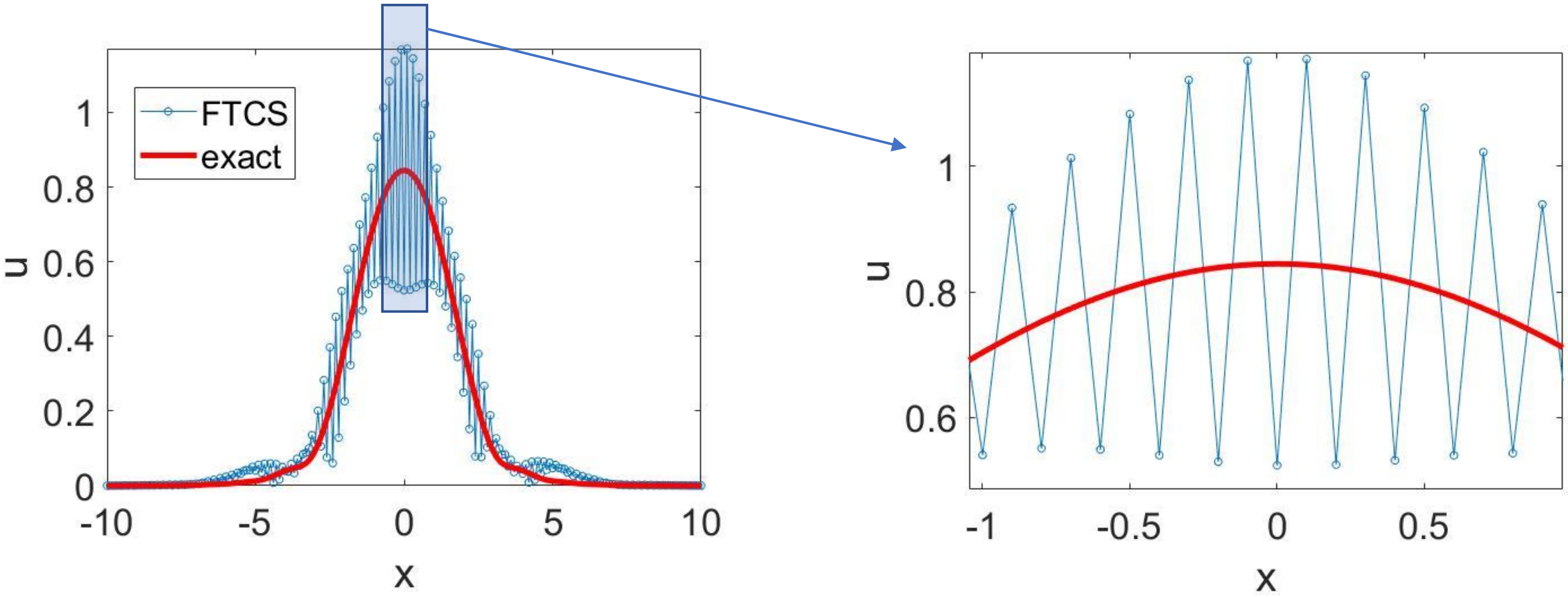
FTCS solution



Forward-Time Centred-Space method (FTCS)

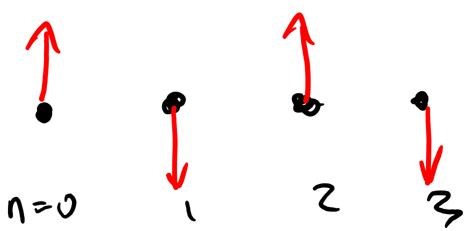
$$i\frac{\partial}{\partial t}\mathrm{u}(x,t)=\left(-\frac{\partial^2}{\partial x^2}+V(x)\right)\mathrm{u}(x,\mathrm{t})$$

- A snapshot at t=0.5



- Compare with stability analysis: $g_k=\sqrt{1+16a^2\sin^2\left(\frac{q_k}{2}\right)}$

Largest growth for $q_k=\pi$, i.e. $\epsilon_n\sim\exp(i\pi n)$



Stability of FTCS methods

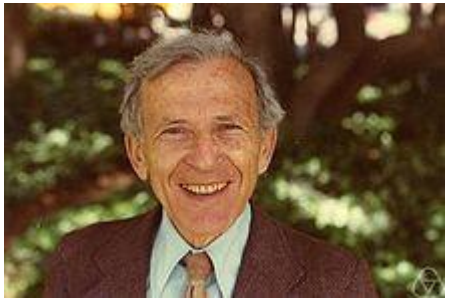
- More generally could replace simple Euler time iterations with a higher-order Runge-Kutta method
- Stability of any explicit method will generally impose a condition on the discretisation parameter:

$$\boxed{a = \frac{\Delta t}{(\Delta x)^2} \leq C}$$

This is known as Courant–Friedrichs–Lewy condition

Richard Courant
1888-1972
https://en.wikipedia.org/wiki/Richard_Courant

Kurt Otto Friedrichs
1901-1982
https://en.wikipedia.org/wiki/Kurt_Otto_Friedrichs



Hans Lewy
1904-1988
https://en.wikipedia.org/wiki/Hans_Lewy

Stability of FTCS methods

- More generally could replace simple Euler time iterations with a higher-order Runge-Kutta method

- Stability of any explicit method will generally impose a condition on the discretisation parameter:

$$\boxed{a = \frac{\Delta t}{(\Delta x)^2} \leq C}$$

This is known as Courant–Friedrichs–Lewy condition

- C is a specific constant, depends on a particular time-propagation explicit method, and a particular PDE. This is known as Courant number

- **Note:** the above condition locks together spatial (Δx) and temporal (Δt) resolutions. if you need to improve spatial resolution e.g. $\mathbf{\Delta x \rightarrow \Delta x/2}$, you need to decrease the time step accordingly: $\mathbf{\Delta t \rightarrow \Delta t/4}$

Crank-Nicolson Scheme

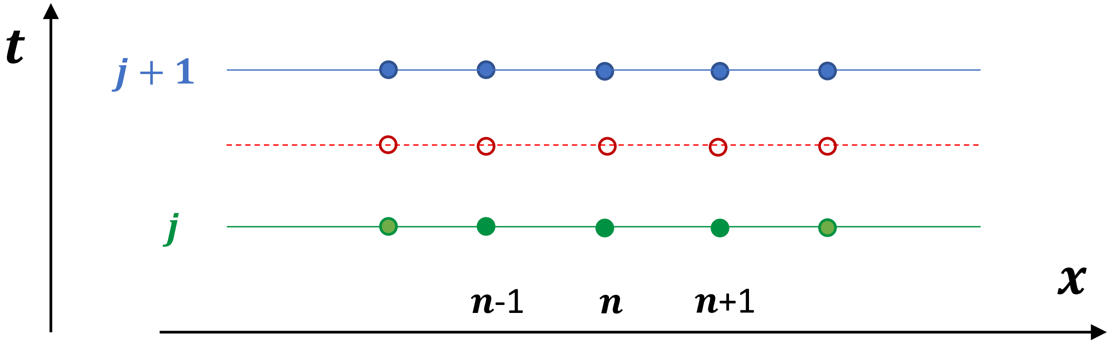
$$i\,\frac{\partial}{\partial t}\mathrm{u}(x,t) = \left(-\frac{\partial^2}{\partial x^2} + V(x)\right)\mathrm{u}(x,\mathrm{t})$$

- The FDA formula for the time derivative at $t = t_j$

$$i\,\frac{1}{\Delta t}\left[\mathrm{u}_n^{(j+1)} - \mathrm{u}_n^{(j)}\right] = \frac{1}{(\Delta x)^2}\left(2u_n^{(j)} - u_{n+1}^{(j)} - u_{n-1}^{(j)}\right) + V_n u_n^{(j)}$$

could be considered as a CDA formula for $\partial u/\partial t$ at **(non-existing in the grid)** $t = t_{j+1/2}$

- But then we would need to evaluate the right-hand side at this non-existing time layer $j + 1/2$



- Can replace it with the average of layers j and $j + 1$, e.g.:

$$V_n u_n^{(j)} \rightarrow V_n u_n^{(j+1/2)} \approx \frac{1}{2}\mathrm{V_n}\left(\mathrm{u_n^{(j+1)}} + u_n^{(j)}\right)$$

John Crank

1916-2006

https://en.wikipedia.org/wiki/John_Crank

Phyllis Nicolson

1917-1968

https://en.wikipedia.org/wiki/Phyllis_Nicolson

Crank-Nicolson Scheme

$$i\left[\mathrm{u}_n^{(j+1)} - \mathrm{u}_n^{(j)}\right] = a\left(2u_n^{(j)} - u_{n+1}^{(j)} - u_{n-1}^{(j)}\right) + V_n u_n^{(j)} \qquad \boldsymbol{a = \frac{\Delta t}{(\Delta x)^2}}$$

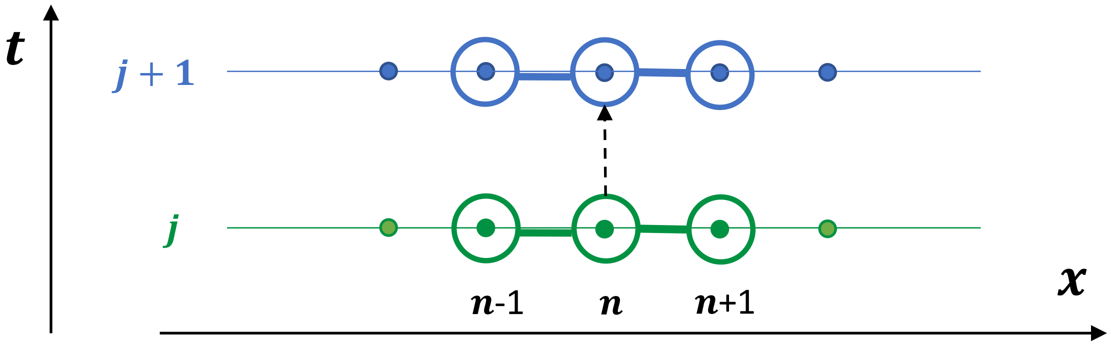
$$\Downarrow$$

$$i\left[\mathrm{u}_n^{(j+1)} - \mathrm{u}_n^{(j)}\right] = \frac{1}{2}\left[a\left(2u_n^{(j)} - u_{n+1}^{(j)} - u_{n-1}^{(j)}\right) + V_n u_n^{(j)}\right] + \frac{1}{2}\left[a\left(2u_n^{(j+1)} - u_{n+1}^{(j+1)} - u_{n-1}^{(j+1)}\right) + V_n u_n^{(j+1)}\right]$$

- Note: we have (still unknown) values of $u_n^{(j+1)}$ in the right-hand side => This is an implicit scheme

- Can re-write as:

$$(i - a - V_n)u_n^{(j+1)} + \frac{a}{2}\left(u_{n+1}^{(j+1)} + u_{n-1}^{(j+1)}\right) = (i + a + V_n)u_n^{(j)} - \frac{a}{2}\left(u_{n+1}^{(j)} + u_{n-1}^{(j)}\right)$$



Crank-Nicolson Scheme

$$(i\ - a - V_n)u_n^{(j+1)} + \frac{a}{2}\left(u_{n+1}^{(j+1)} + u_{n-1}^{(j+1)}\right) = (i + a + V_n)u_n^{(j)} - \frac{a}{2}\left(u_{n+1}^{(j)} + u_{n-1}^{(j)}\right)$$

- In the vector format: $\hat{A}\vec{u}^{(j+1)} = \hat{B}\,\vec{u}^{(j)}$

$$\boxed{\vec{u}^{(j+1)} = \hat{A}^{-1}\hat{B}\,\vec{u}^{(j)}}$$

Note: $\hat{A}$ is a tri-diagonal matrix. It does not require much effort to invert it.

- Stability?

- Repeating similar procedure as for FTCS, we can obtain for the error:

$$\boxed{E_k^{(j+1)} = \left[\frac{1 - i2a\sin^2\left(\frac{q_k}{2}\right)}{1 + i2a\sin^2\left(\frac{q_k}{2}\right)}\right] E_k^{(j)}}$$

$$|g_k| = \left|1 - i2a\sin\left(\frac{q_k}{2}\right)\right| / \left|1 + i2a\sin\left(\frac{q_k}{2}\right)\right| = 1$$ ⇒ **Marginally stable**

Crank-Nicolson Scheme

$$i\,\frac{\partial}{\partial t}\mathrm{u}(x,t) = \left(-\frac{\partial^2}{\partial x^2} + V(x)\right)\mathrm{u}(x,\mathrm{t})$$

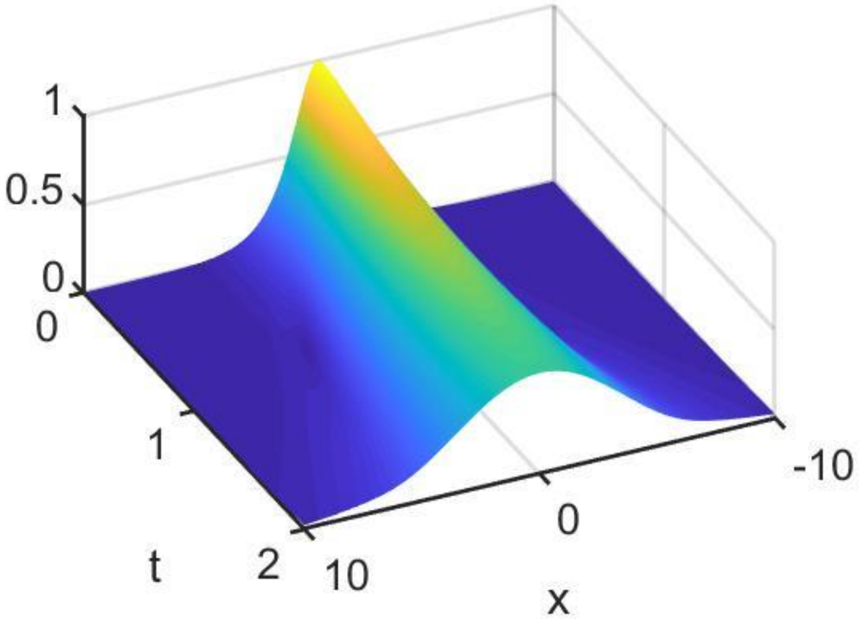
- Stable evolution:

Select $dx = 0.1, dt = 0.02$ (a =2)

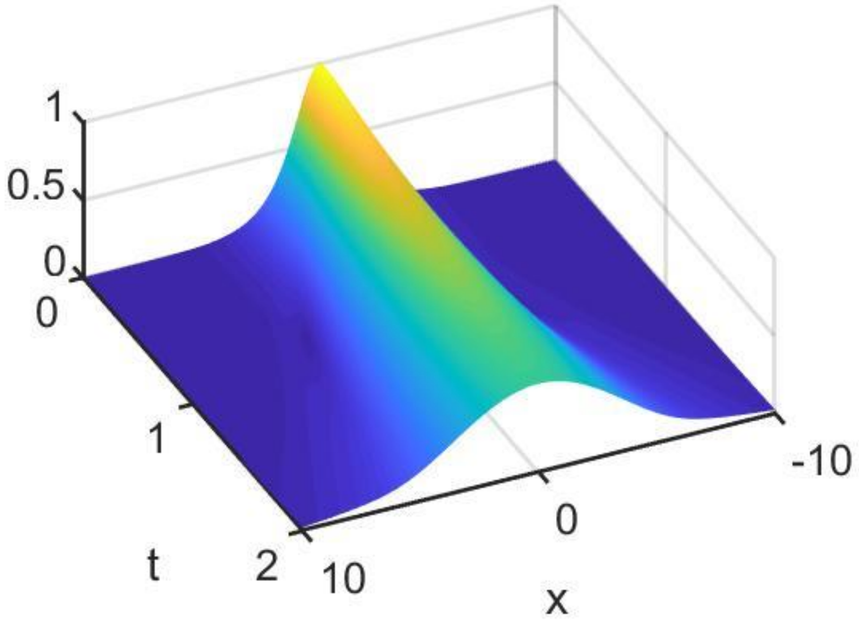
Initial condition: $u(x, t = 0) = \mathrm{sech}(x)$

$V(x) = 0$

Exact solution



CN solution



Summary:

- Finite-difference schemes offer an easy option for numerical implementation of IVP PDE problems.
- Stability requirements will often lock spatial and temporal resolutions (Courant–Friedrichs–Lewy condition):

$$\boldsymbol{a = \frac{\Delta t}{(\Delta x)^2} \leq C}$$ **for parabolic PDEs**

$$\boldsymbol{a = \frac{\Delta t}{\Delta \mathrm{x}} \leq C}$$ **for hyperbolic PDEs**

- Implicit schemes (such as Crank-Nicolson) often do not require much of an additional effort, but they are generally more stable.
- However, implicit schemes are not easy to implement for nonlinear PDEs.