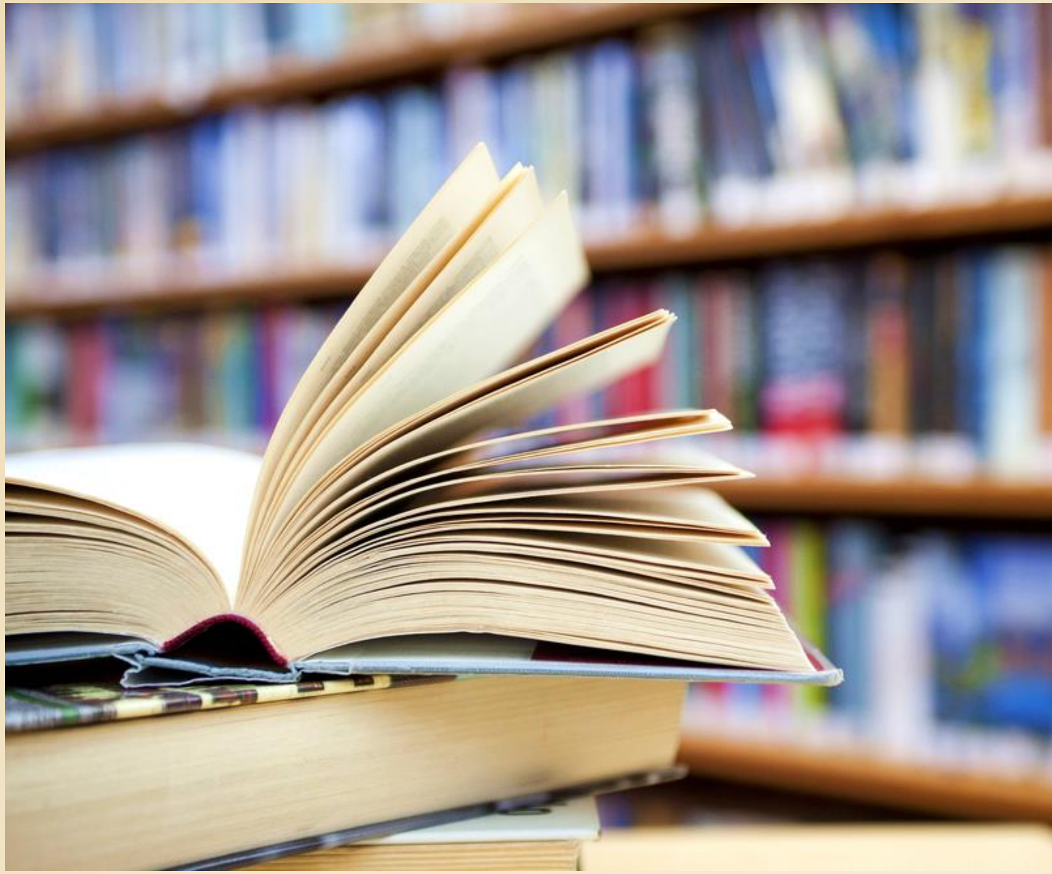
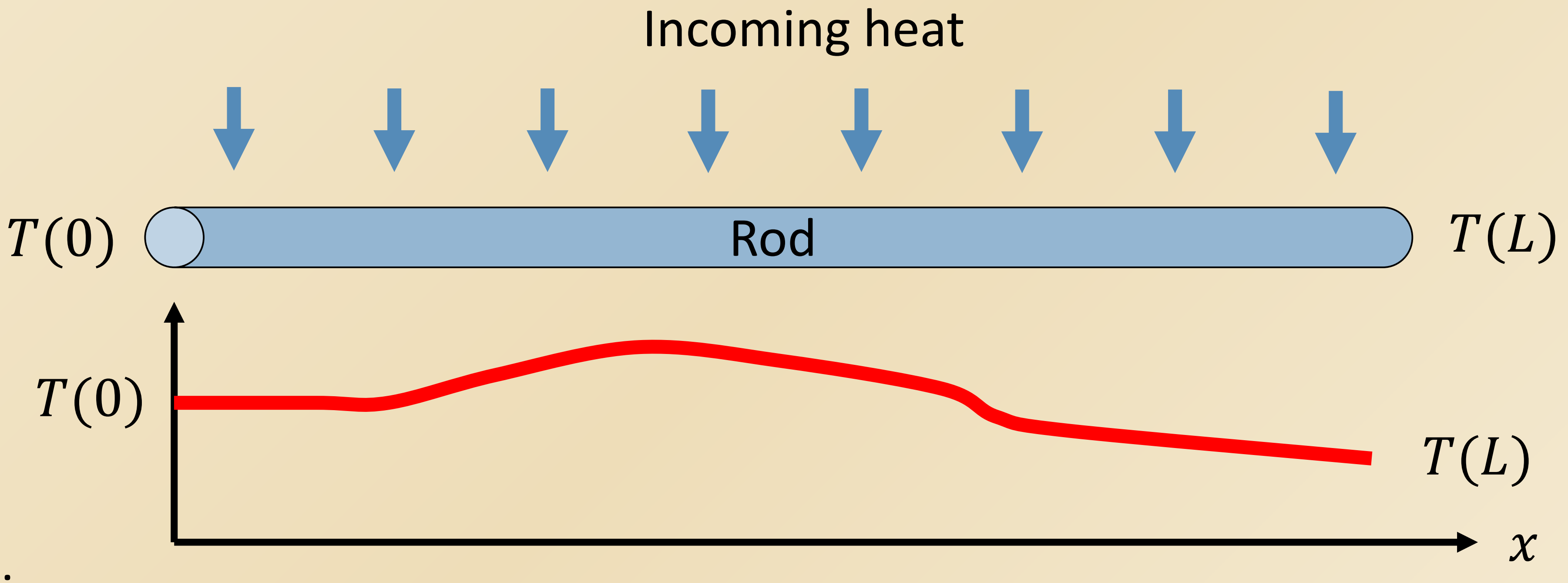
1D POISSON EQUATION WITH THE FINITE DIFFERENCE METHOD –

DISCRETIZATION AND MATLAB VERSION



1D Heat Equation



Heat Equation

$$\frac{d}{dx}\left(k(x)\frac{dT}{dx}(x)\right) = -h_s(x) \quad x \in (0, L)$$

$k(x)$ thermal conductivity

$h_s(x)$ heat generation rate

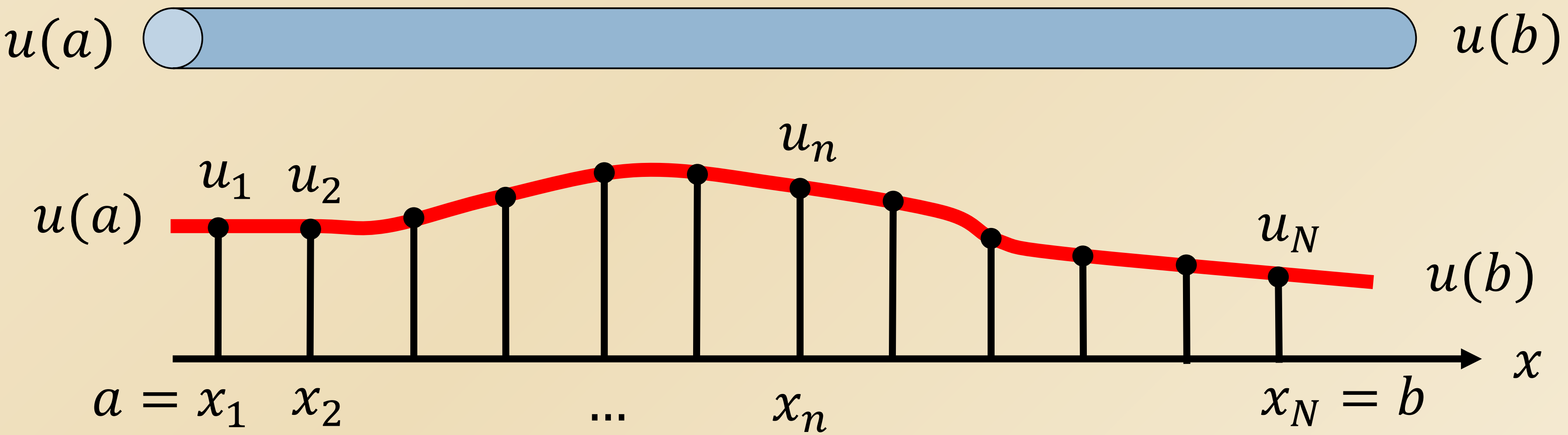
Poisson Equation
(Normalized Heat Equation)

$$\frac{d^2u}{dx^2}(x) = -f(x) \qquad x \in (a, b)$$

Finite Difference Discretization of the 1D Poisson Equation

$$\frac{d^2u}{dx^2}(x) = -f(x) \qquad x \in (a, b)$$

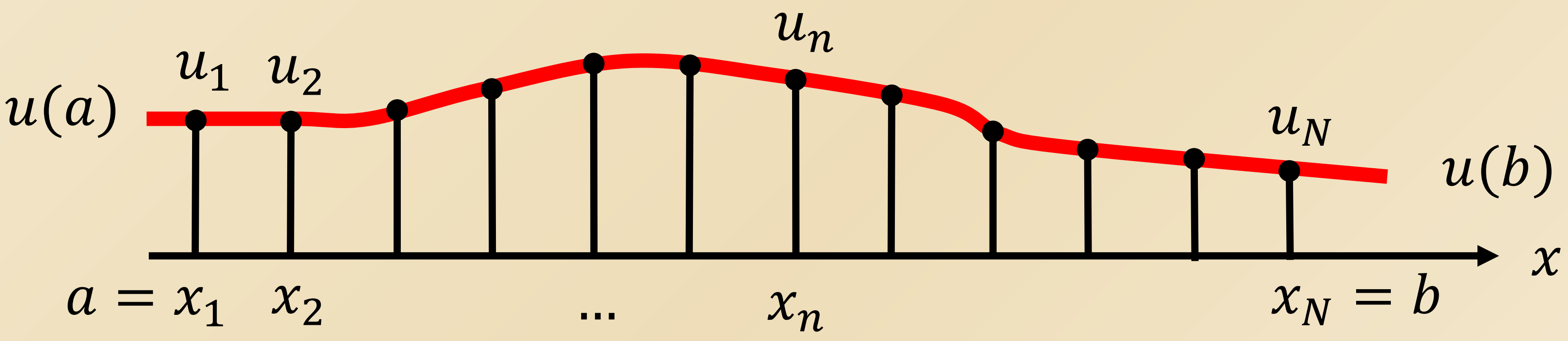
$u(x)$ – unknown of the problem
$f(x)$ – known function



We do not seek for the continuous behavior of $u(x)$, but for the samples

$$u_1, \ldots, u_N$$

Finite Difference Discretization of the 1D Poisson Equation



On assuming a uniform sampling step and equal to Δx

$$-f_n = -f(x_n) = \frac{d^2u}{dx^2}(x_n) \approx \frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta x^2} \qquad n = 2, \dots, N-1$$

Finite Difference Discretization of the 1D Poisson Equation

$$-f(x_n) = \frac{d^2u}{dx^2}(x_n) \approx \frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta x^2} \qquad n = 2, \dots, N-1$$

$n = 1 \qquad u(x_1) = u_1 = u_a$

$n = 2 \qquad \dfrac{-u_3 + 2u_2 - u_1}{\Delta x^2} = f_2 \quad \Rightarrow \quad \dfrac{-u_3 + 2u_2}{\Delta x^2} = f_2 + \dfrac{u_a}{\Delta x^2}$

$n = 3 \qquad \dfrac{-u_4 + 2u_3 - u_2}{\Delta x^2} = f_3$

$\dots$

$n = N-1 \qquad \dfrac{-u_N + 2u_{N-1} - u_{N-2}}{\Delta x^2} = f_{N-1} \quad \Rightarrow \quad \dfrac{-u_{N-2} + 2u_{N-1}}{\Delta x^2} = f_{N-1} + \dfrac{u_b}{\Delta x^2}$

$n = N \qquad u(x_N) = u_N = u_b$

Finite Difference Discretization of the 1D Poisson Equation

Solve a tridiagonal linear system

$$\begin{bmatrix} 2 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 & 0 & 0 \\ \dots & & & & & & & \\ 0 & 0 & 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ \dots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} f_2 + \dfrac{u_a}{\Delta x^2} \\ f_3 \\ f_4 \\ \dots \\ f_{N-2} \\ f_{N-1} + \dfrac{u_n}{\Delta x^2} \end{bmatrix}$$

1D Poisson Equation Solved with Finite Differences in Matlab – 1/2

```
% --- Resolution domain
a = -1;                          % --- Left border of the resolution domain
b = 1;                           % --- Right border of the resolution domain

% --- Dirichlet boundary conditions
Ta = 0.1;                        % --- Boundary conditions at the left border of the resolution domain
Tb = 0.1;                        % --- Boundary conditions at the right border of the resolution domain

Nintervals  = 100;               % --- Number of discretization intervals
Ninner      = Nintervals - 1;    % --- Number of inner mesh nodes
Deltax      = (b - a) / Nintervals;  % --- Discretization step
x           = a : Deltax : b;    % --- Node coordinates

% --- Right-hand side function
alpha       = -log(Ta) / b^2;
f           = @(x)(2 * alpha * exp(-alpha * x.^2) .* (1 - 2 * alpha * x.^2));

% --- Exact solution
Tref        = @(x)(exp(-alpha * x.^2));
```

1D Poisson Equation Solved with Finite Differences in Matlab – 2/2

```
% --- Linear system matrix
d          = ones(Ninner, 1);
A          = spdiags([-d , 2 * d , -d], -1 : 1, Ninner, Ninner) / (Deltax^2);

% --- Right-hand side vector
brhs       = feval(f, x(2 : (end - 1)));
brhs(1)    = brhs(1)   + Ta / (Deltax^2);
brhs(end)  = brhs(end) + Tb / (Deltax^2);

% --- Solve the linear system
ysol       = A \ brhs.';

% --- Analytical solution
yref       = feval(Tref, x(2 : (end - 1)));

% --- Percentage rms error
error      = 100 * sqrt(sum(abs(ysol.' - yref).^2) / sum(abs(yref).^2));
fprintf('Percentage rms error %f\n', error);
```