9th SimLab Course on Parallel Numerical Simulation, 4.10 – 8.10.2010

Iterative Solvers for Linear Systems

Bernhard Gatzhammer

Chair of Scientific Computing in Computer Science
Technische Universität München
Germany

Outline

- Poisson-Equation: From the Model to an Equation System
- Direct Solvers vs. Iterative Solvers
- Jacobi and Gauss-Seidel Method
- Residual and Smoothness
- Multigrid Solver

Outline

- Poisson-Equation: From the Model to an Equation System
- Direct Solvers vs. Iterative Solvers
- Jacobi and Gauss-Seidel Method
- Residual and Smoothness
- Multigrid Solver

The Poisson Equation – An Elliptic PDE

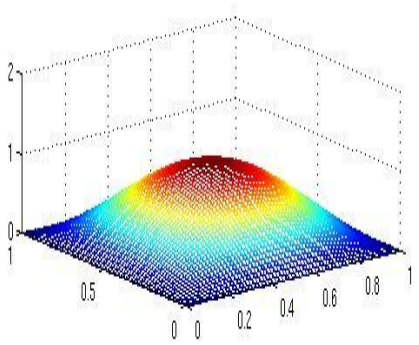
- General: $\in \mathbb{R}^d, \quad \delta \in \mathbb{R}^{d-1}, \quad u : \to \mathbb{R}$

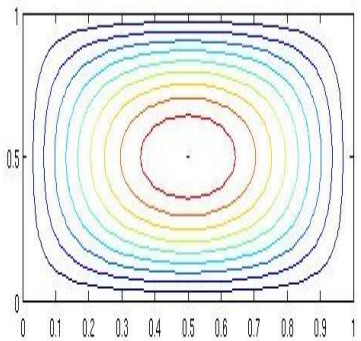
$$\Delta u = f \quad \text{in} \qquad (\Delta = \nabla^2 = \nabla \cdot \nabla)$$

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \ldots + \frac{\partial^2 u}{\partial x_d^2} = f \quad \text{in}$$

$$u = g \quad \text{on } \delta$$

- 1D: $\in \mathbb{R}, \quad \frac{\partial^2 u}{\partial x^2} = f \text{ in}$
- 2D: $\in \mathbb{R}^2, \quad \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = f \text{ in}$

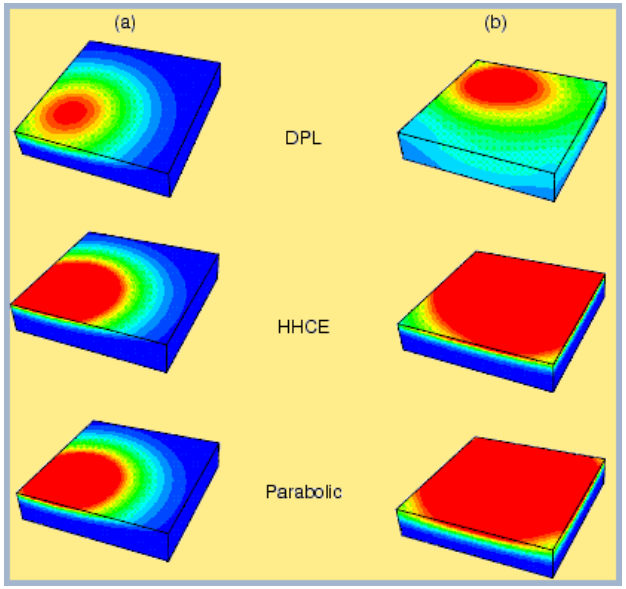




Why the Poisson Equation is of Importance

The Laplace operator Δ is part of many important PDEs:

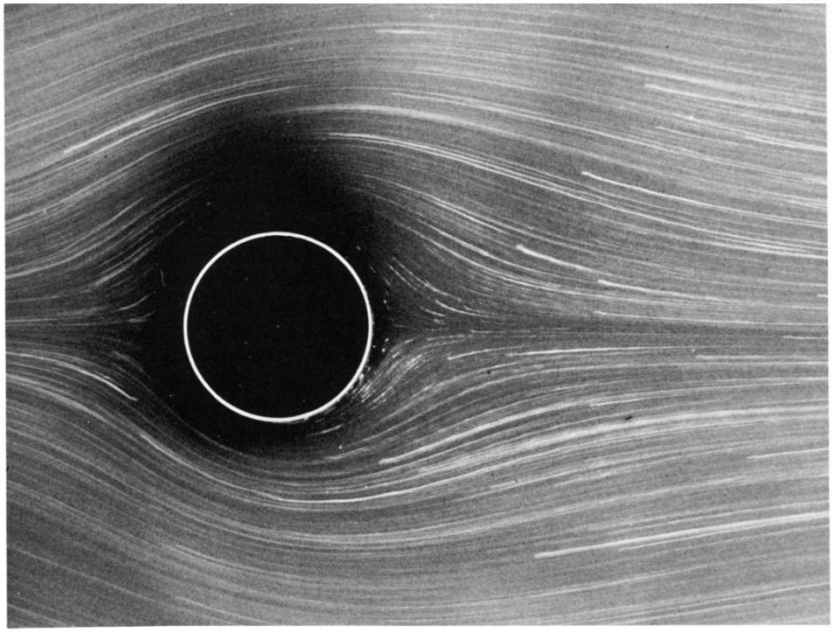
- Heat equation: diffusion of heat



Why the Poisson Equation is of Importance

The Laplace operator Δ is part of many important PDEs:

- Flow dynamics: friction between molecules



Why the Poisson Equation is of Importance

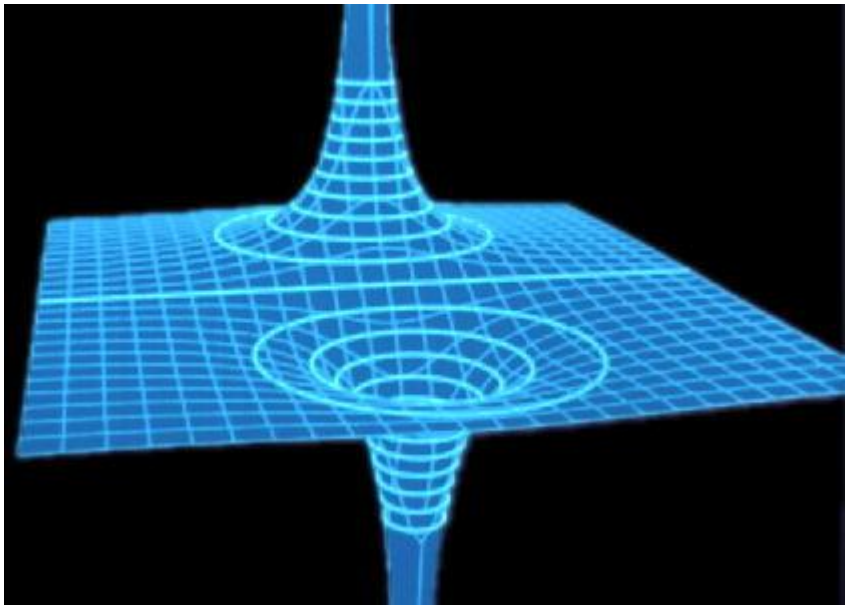
The Laplace operator Δ is part of many important PDEs:

- Structural mechanics: displacement of membrane structures

Why the Poisson Equation is of Importance

The Laplace operator Δ is part of many important PDEs:

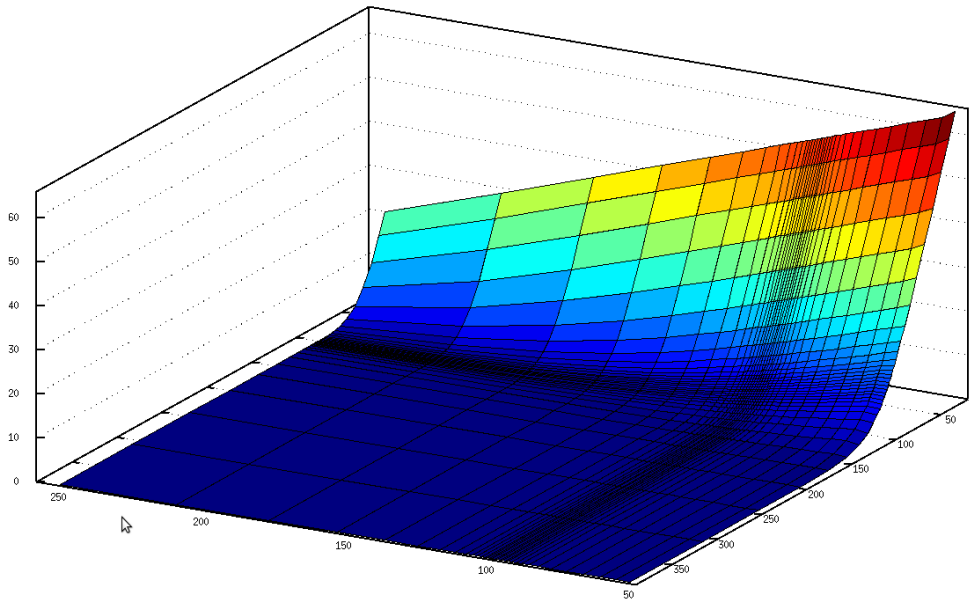
- Electrics: electric-, magnetic potentials



Why the Poisson Equation is of Importance

The Laplace operator Δ is part of many important PDEs:

- Financial mathematics: Diffusive process for option pricing



Why the Poisson Equation is of Importance

The Laplace operator Δ is part of many important PDEs:

- Heat equation: diffusion of heat
- Flow dynamics: friction between molecules
- Structural mechanics: displacement of membrane structures
- Electrics: electric-, magnetic potentials
- Financial mathematics: Diffusive process for option pricing
- ...

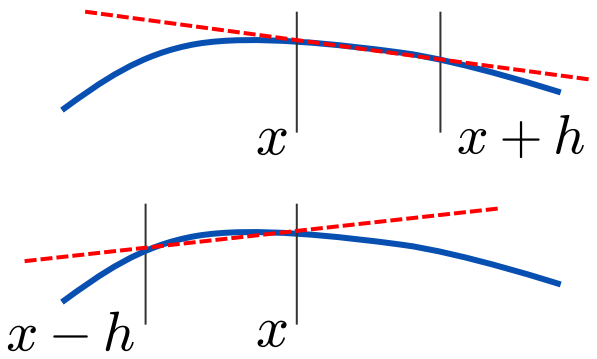
Discretization with Finite Differences I

Find solution to Poisson equation numerically on a computer:

- Computer has only a finite amount of memory/power → discretization
- Finite Differences as simple discretization
- Idea: Replace derivative by difference

$$\frac{u(x+h)-u(x)}{h} \xrightarrow{h\to 0} \frac{\partial u}{\partial x}$$

$$\frac{u(x)-u(x-h)}{h} \xrightarrow{h\to 0} \frac{\partial u}{\partial x}$$



Discretization with Finite Differences II

Use forward and backward differences for 2nd derivatives:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x}\frac{\partial u}{\partial x}$$

forward diff.:

$$\approx \frac{\partial}{\partial x}\frac{u(x+h)-u(x)}{h}$$

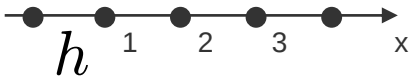
backward diff.:

$$\approx \frac{\frac{u(x+h)-u(x)}{h} - \frac{u(x+h-h)-u(x-h)}{h}}{h}$$

$$= \frac{u(x+h)-2u(x)+u(x-h)}{h^2}$$

Grid, Stencil, and Matrix 1D

Regular Cartesian grid:

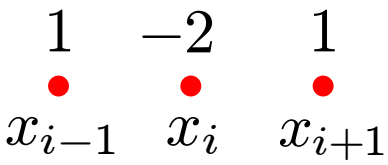


$$x_i = ih$$

Nodal equation:

$$u_{i-1} - 2u_i + u_{i+1} = h^2 f_i$$

Stencil:

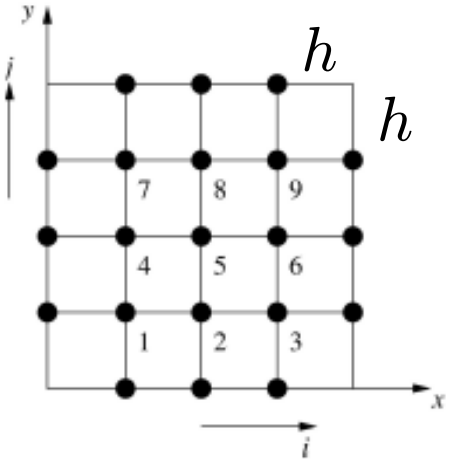


System matrix:

$$A = \frac{1}{h^2} \begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}$$

Grid, Stencil, and Matrix 2D

Regular Cartesian grid:



$$x_{i,j} = (ih, jh)$$

Nodal equation:

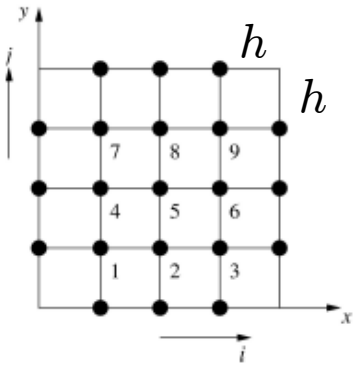
$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = h^2 f_{i,j}$$

Stencil:

$$\begin{array}{ccc} & 1 & \\ & x_{i,j+1} & \\ 1 & -4 & 1 \\ x_{i-1,j} & x_{i,j} & x_{i+1,j} \\ & 1 & \\ & x_{i,j-1} & \end{array}$$

Grid, Stencil, and Matrix 2D

System matrix:



$$A = \frac{1}{h^2} \begin{pmatrix} -4 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & -4 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & -4 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & -4 & 1 & \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 & -4 & 1 & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot & 1 & -4 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & -4 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & -4 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & -4 \end{pmatrix}$$

$$\begin{array}{ccc} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{array}$$

Discretized Poisson Equation

We obtain a system of equations to be solved:

$$Au = f, \quad A \in \mathbb{R}^{N \times N}, \quad u, f \in \mathbb{R}^N$$

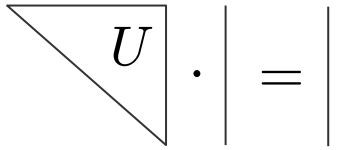
- N is the number of grid unknowns, i.e. equations
- f contains sink/source terms and boundary influences
- We want to obtain $u = A^{-1} f$

Outline

- Poisson-Equation: From the Model to an Equation System
- Direct Solvers vs. Iterative Solvers
- Jacobi and Gauss-Seidel Method
- Residual and Smoothness
- Multigrid Solver

Direct Solvers of Systems of Linear Equations

Popular direct solvers for $Au = f$:



- Gaussian elimination $T_N \cdot \ldots \cdot T_2 \cdot Au = T_N \cdot \ldots \cdot T_2 \cdot f, \quad Uu = f'$
- LU decomposition $A = LU, \quad LUu = f, \quad Uu = L^{-1} f$
- Cholesky factorization $A = U^T U$

→ compute $u = A^{-1} f$ directly up to numerical accuracy

→ computational complexity of GEM in 2D: $O(h^6)$, in 3D: $O(h^9)$

→ GEM optimized for band-structure: in 2D: $O(h^4)$, in 3D: $O(h^7)$

Runtime of GEM for Solving Poisson 2D

h	N	runtime (33 TFlop/s)
2^{-8}	65.536	0.5 sec
2^{-9}	262.144	8.3 sec
2^{-10}	1.048.576	2 min 23 sec
2^{-11}	4.194.304	35 min 32 sec
2^{-12}	16.777.216	9 h 28 min 38 sec
2^{-13}	67.108.864	6 d 7 h 28 min 10 sec

Runtime of GEM for Solving Poisson 3D

h	N	runtime (33 TFlop/s)
2^{-6}	262.144	8 min 53 sec
2^{-7}	2.097.152	18 h 57 min 26 sec
2^{-8}	16.777.216	100 d 26 h 10 min 53 sec
2^{-9}	134.217.728	35 a 156 d 57 h 53 min 04 sec

Iterative Solvers of Systems of Linear Equations

In contrast to direct solvers, iterative solvers

- Compute approximate solutions $u^k \approx A^{-1} f$
- Improve the approximation in an iterative process
- One iteration is cheap to compute, in 2D $O(h^2)$, in 3D $O(h^3)$
- Fixpoint iteration: $u^{k+1} = F(u^k) \quad \xrightarrow{k \to \infty} \quad u = F(u)$
- Types of methods:

  Splitting methods, Krylov-subspace methods, Multigrid methods

Outline

- Poisson-Equation: From the Model to an Equation System
- Direct Solvers vs. Iterative Solvers
- Jacobi and Gauss-Seidel Method
- Residual and Smoothness
- Multigrid Solver

Splitting Methods

Split up matrix A:

$$Au = f$$

$$(S + T)u = f$$

Update rule:

$$Su^{k+1} = f - Tu^k$$

$$u^{k+1} = S^{-1}(f - Tu^k)$$

$$u^{k+1} = F(u^k)$$

→ Matrix S has to be of simple form to to get u^{k+1}

The Jacobi Method

- Choose: $S = diag(A) := D_A$
- Splitting: $A = S + T = D_A + (A - D_A)$
- Update rule:

$$Au = f$$

$$(D_A + (A - D_A))u = f$$

$$D_A u^{k+1} = (f - (A - D_A)u^k)$$

$$u^{k+1} = D_A^{-1}(f - (A - D_A)u^k)$$

$$u_i^{k+1} = \frac{1}{a_{ii}}(f_i - \sum_{\substack{j=1 \\ j \neq i}}^{N} a_{ij} u_j^k)$$

Jacobi Method Example – 1D Heat Equation

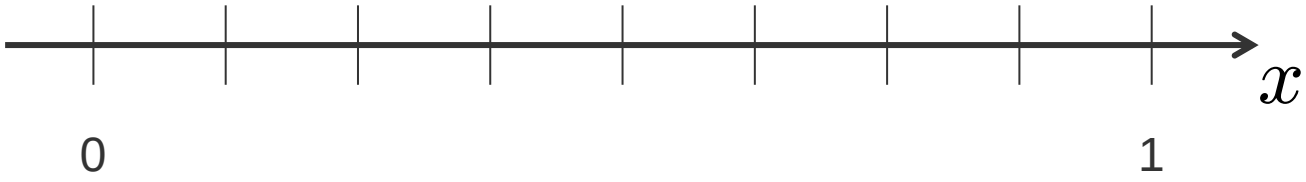
- Stationary heat equation with u as temperature: $\frac{\partial^2 u}{\partial x^2} = f \text{ in}$
- Boundaries have zero temperature, no sources: $f = 0$
- Solution is known: $u = 0$

$$u_i^{k+1} = \frac{1}{a_{ii}}(f_i - \sum_{\substack{j=1 \\ j \neq i}}^{N} a_{ij} u_j) \quad \Longrightarrow \quad \boxed{u_i^{k+1} = \frac{1}{2}(u_{i-1}^k + u_{i+1}^k)}$$

Jacobi Method Example – 1D Heat Equation

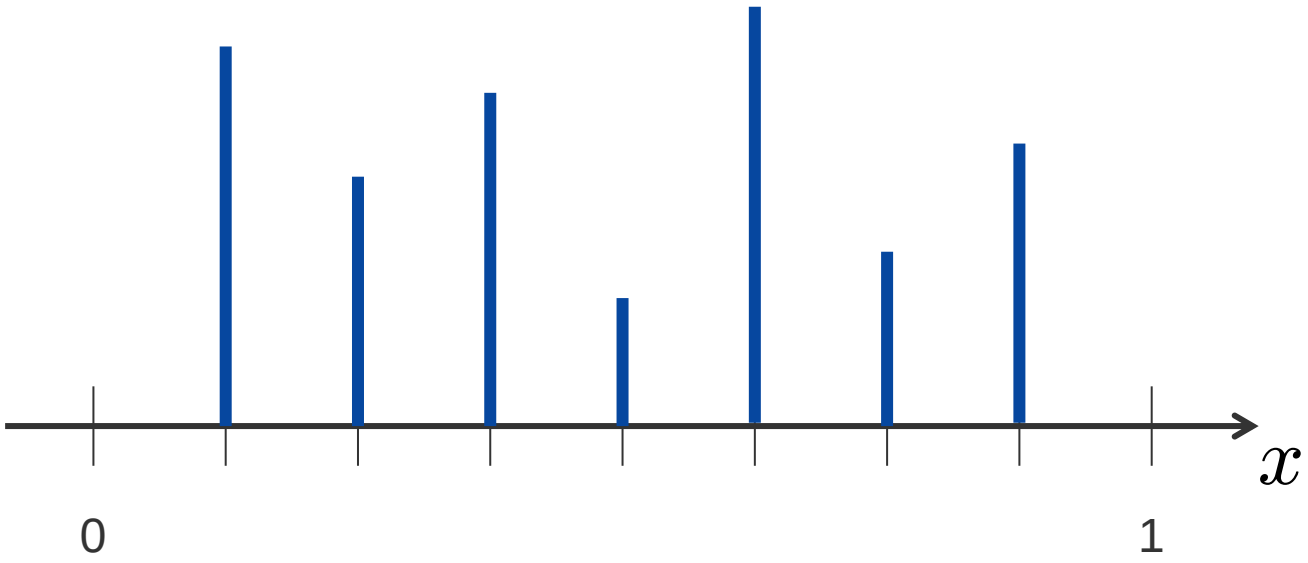
Unit domain: $:= [0,1], \quad h = \frac{1}{8}, \quad N = 7$

$$A = \begin{pmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{pmatrix}$$



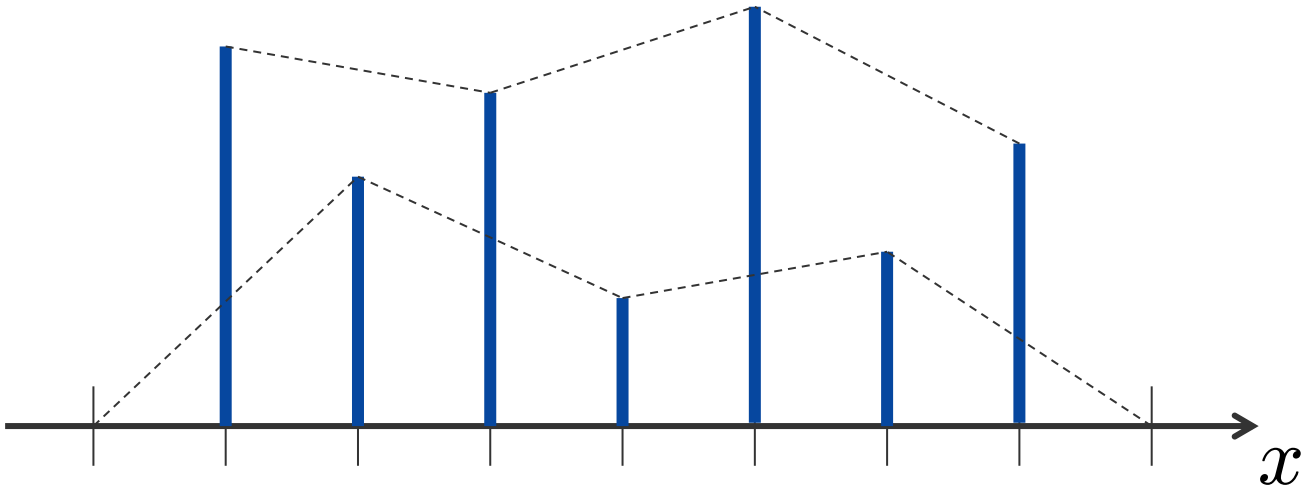
Jacobi Method Example – 1D Heat Equation

- Random initial value $u^0 = rand(N)$



Jacobi Method Example – 1D Heat Equation

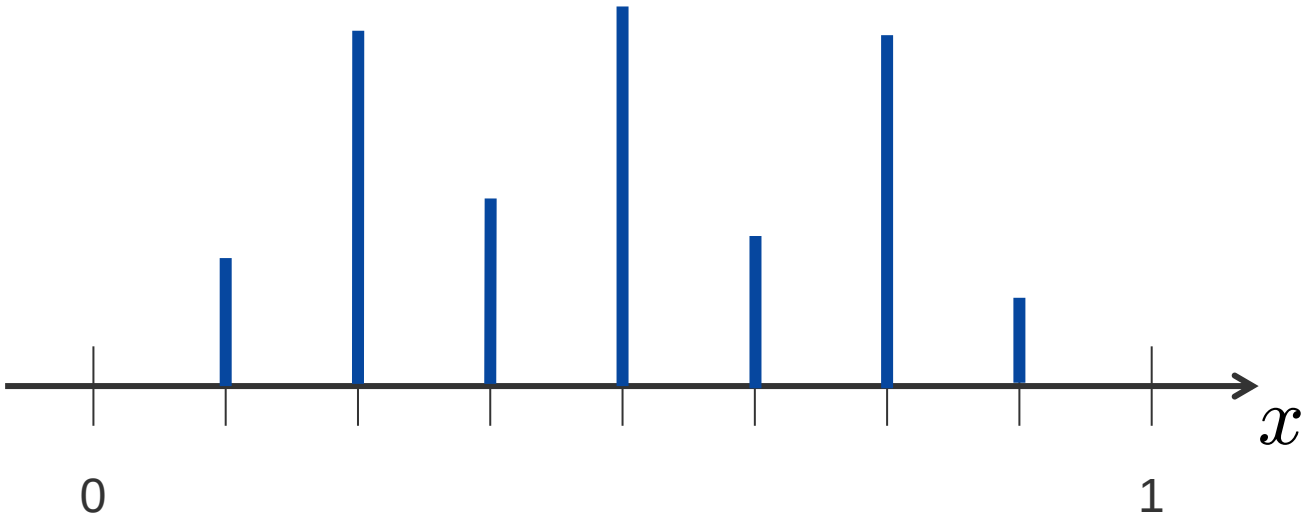
$$k = 0$$



$$u_i^{k+1} = \frac{1}{2}(u_{i-1}^k + u_{i+1}^k)$$

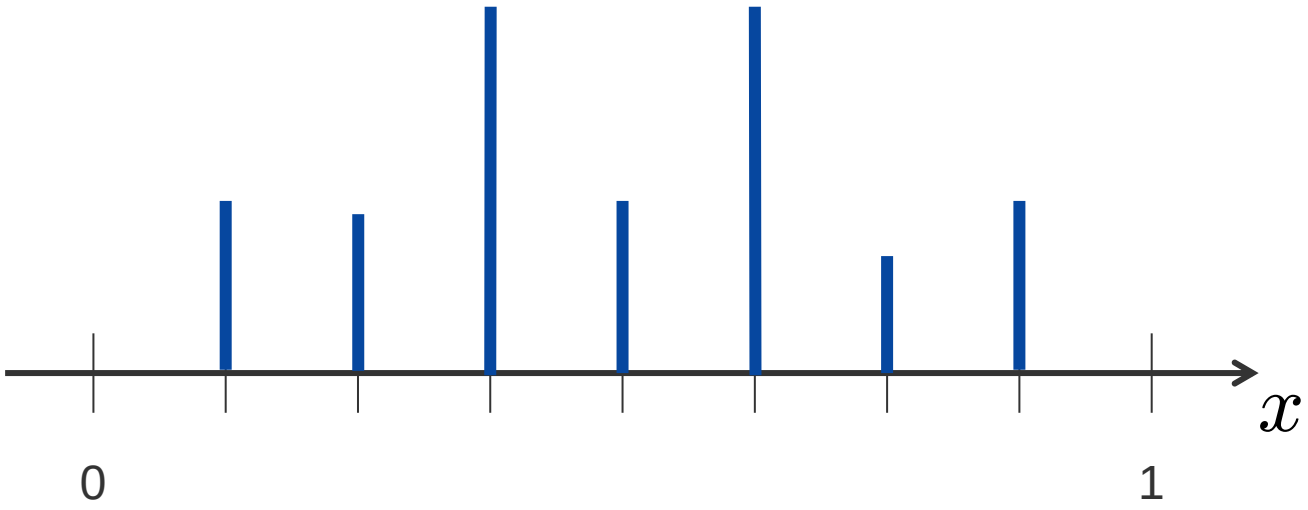
Jacobi Method Example – 1D Heat Equation

$$k = 1$$



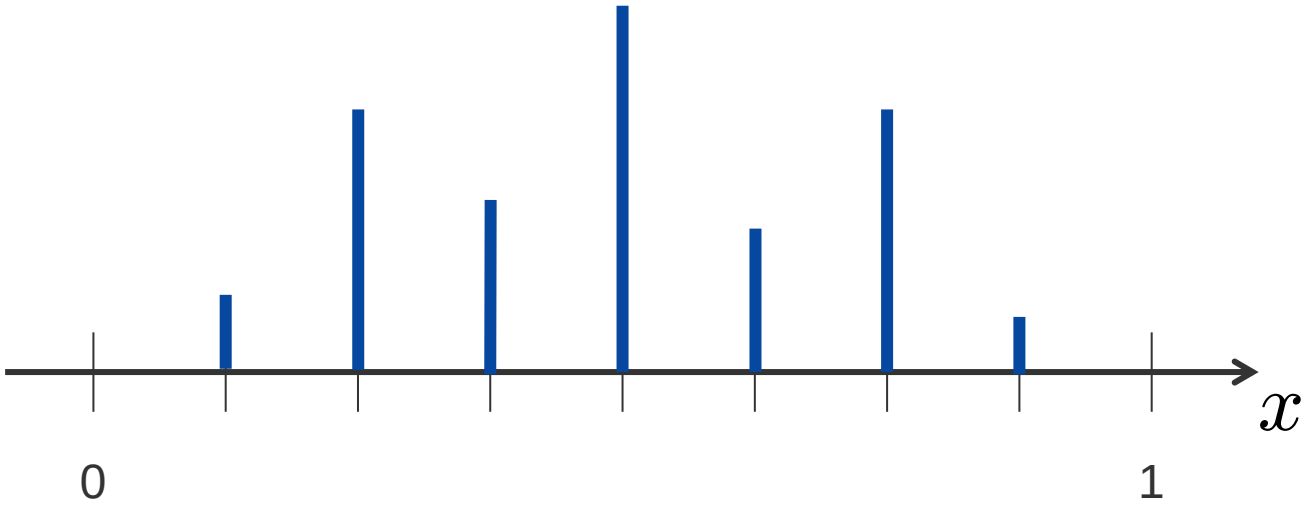
Jacobi Method Example – 1D Heat Equation

$$k = 2$$



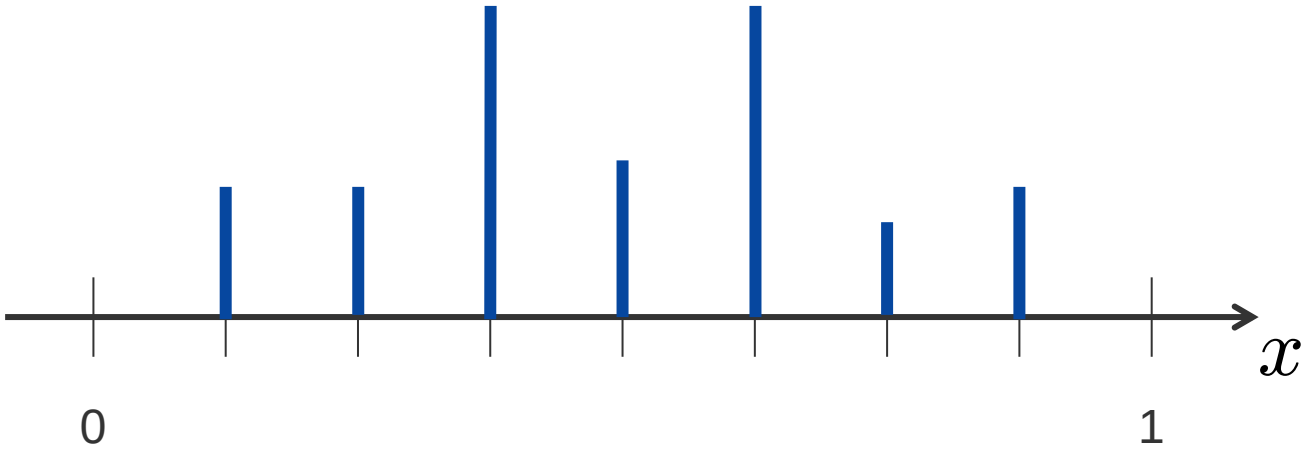
Jacobi Method Example – 1D Heat Equation

$$k = 3$$



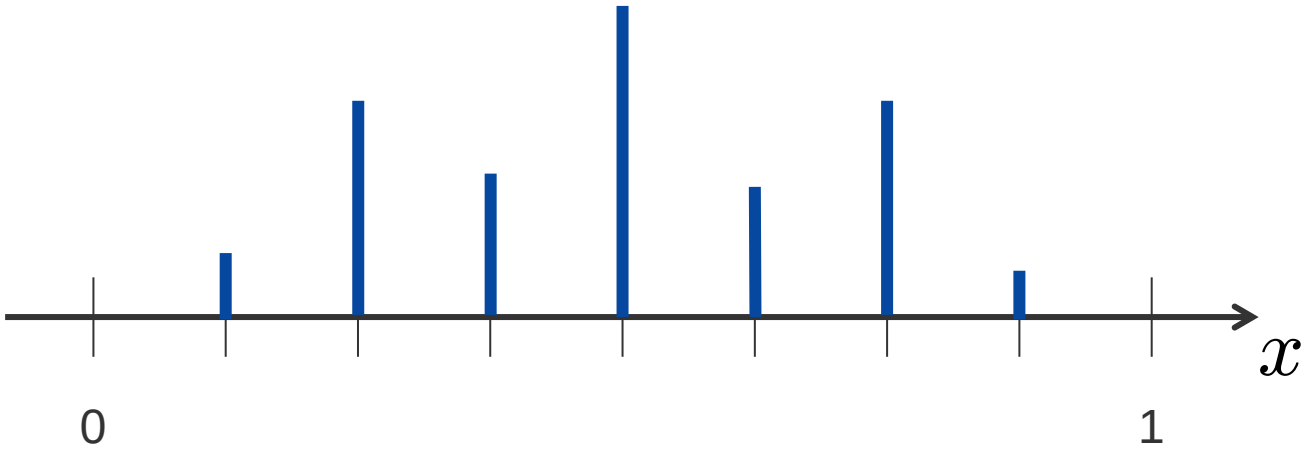
Jacobi Method Example – 1D Heat Equation

$$k = 4$$



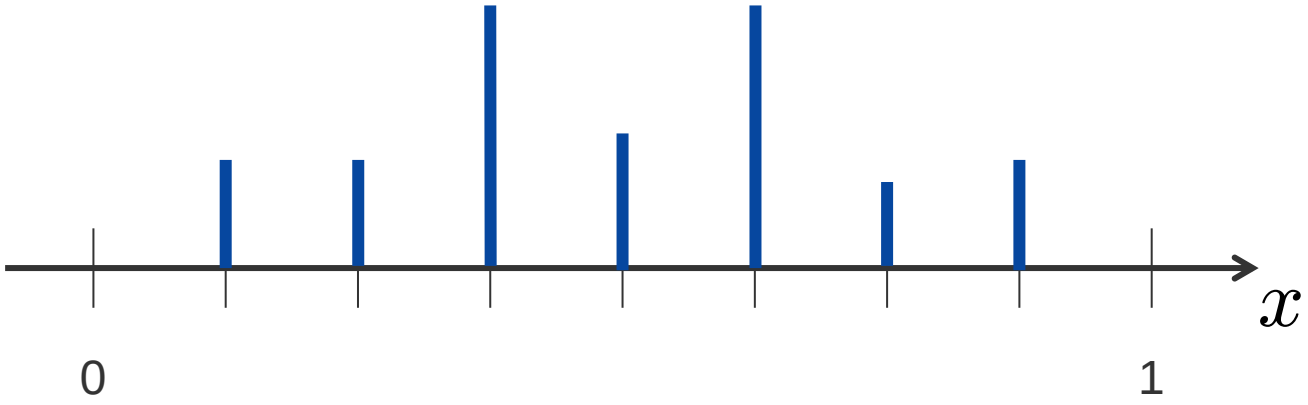
Jacobi Method Example – 1D Heat Equation

$$k = 5$$



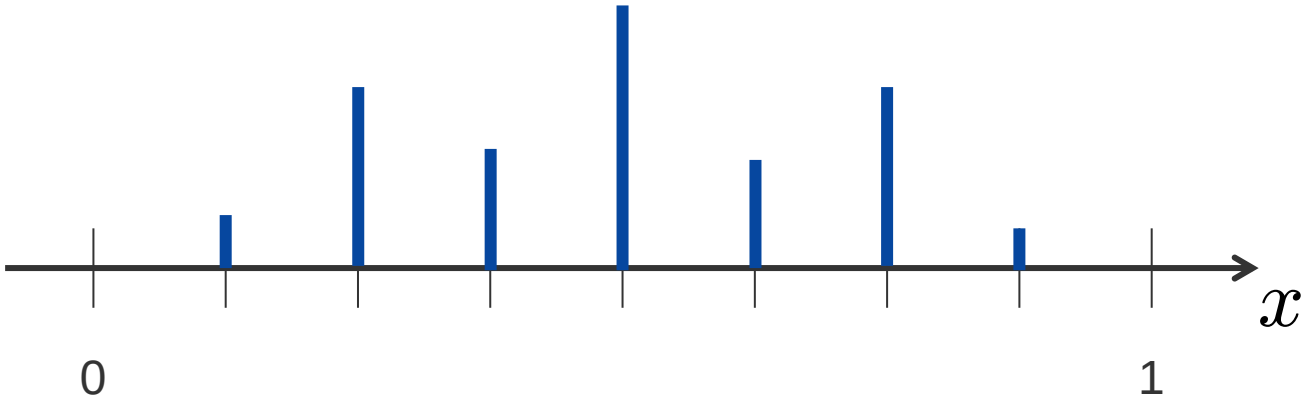
Jacobi Method Example – 1D Heat Equation

$$k = 6$$



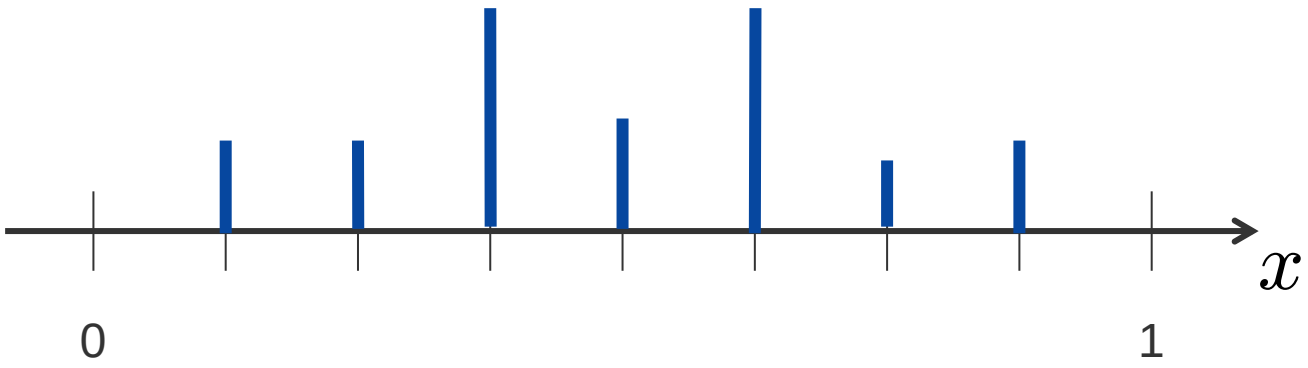
Jacobi Method Example – 1D Heat Equation

$$k = 7$$



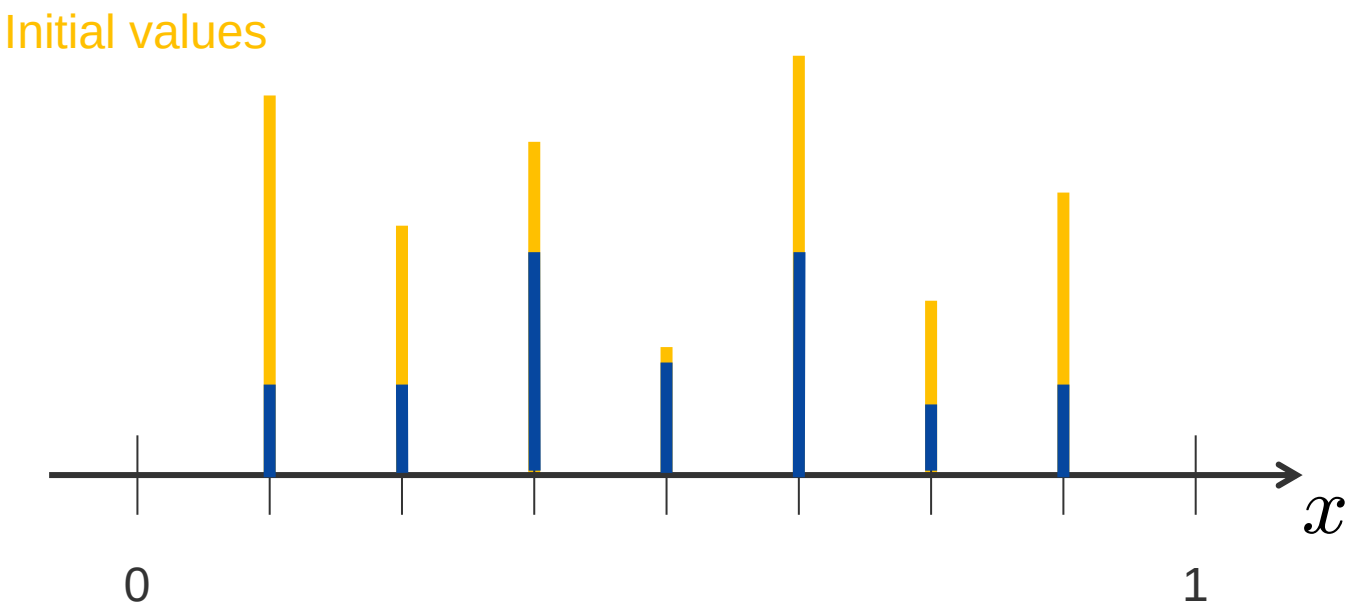
Jacobi Method Example – 1D Heat Equation

$$k = 8$$



Jacobi Method Example – 1D Heat Equation

$$k = 8$$



The Gauss-Seidel Method

- Choose: $S = L(A) := L_A$
- Splitting: $A = S + T = L_A + (A - L_A)$
- Update rule:

$$Au = f$$

$$(L_A + (A - L_A))u = f$$

$$L_A u^{k+1} = (f - (A - L_A)u^k)$$

$$u^{k+1} = L_A^{-1}(f - (A - L_A)u^k)$$

$$u_i^{k+1} = \frac{1}{a_{ii}}(f_i - \sum_{j=1}^{i-1} a_{ij}u_j^{k+1} - \sum_{j=i+1}^{N} a_{ij}u_j^k)$$

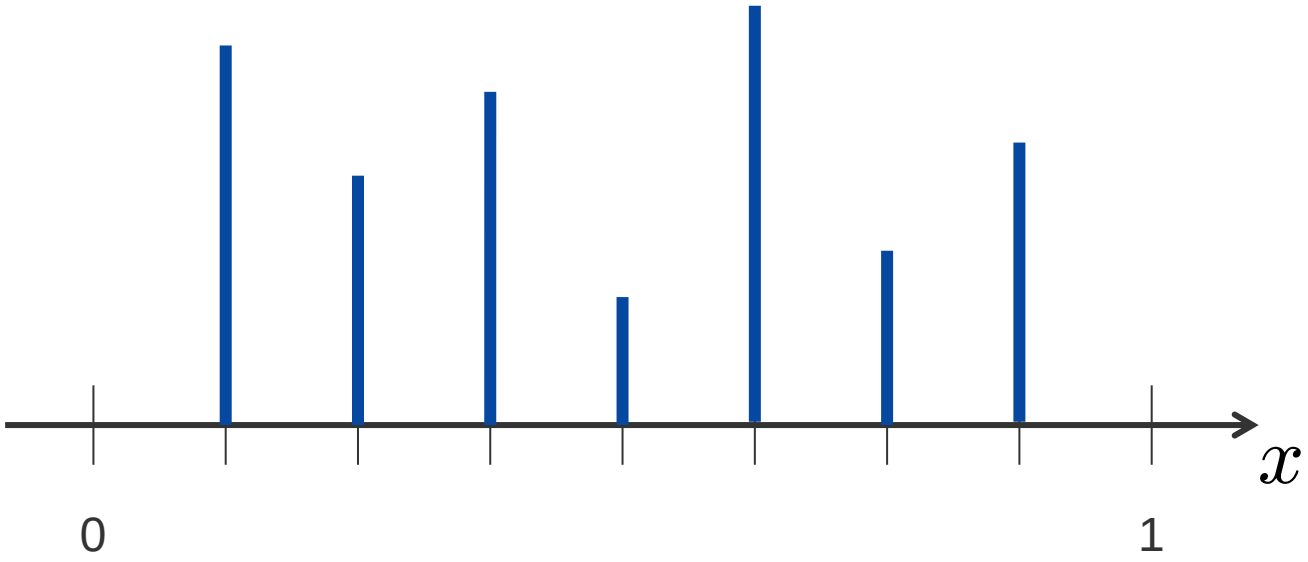
Gauss-Seidel Method Example – 1D Heat Equation

- Stationary heat equation with u as temperature: $\frac{\partial^2 u}{\partial x^2} = f \text{ in}$
- Boundaries have zero temperature, no sources: $f = 0$
- Solution is known: $u = 0$

$$u_i^{k+1} = \frac{1}{a_{ii}}(f_i - \sum_{j=1}^{i-1} a_{ij} u_j^{k+1} - \sum_{j=i+1}^{N} a_{ij} u_j^k) \quad \Longrightarrow \quad \boxed{u_i^{k+1} = \frac{1}{2}(u_{i-1}^{k+1} + u_{i+1}^k)}$$

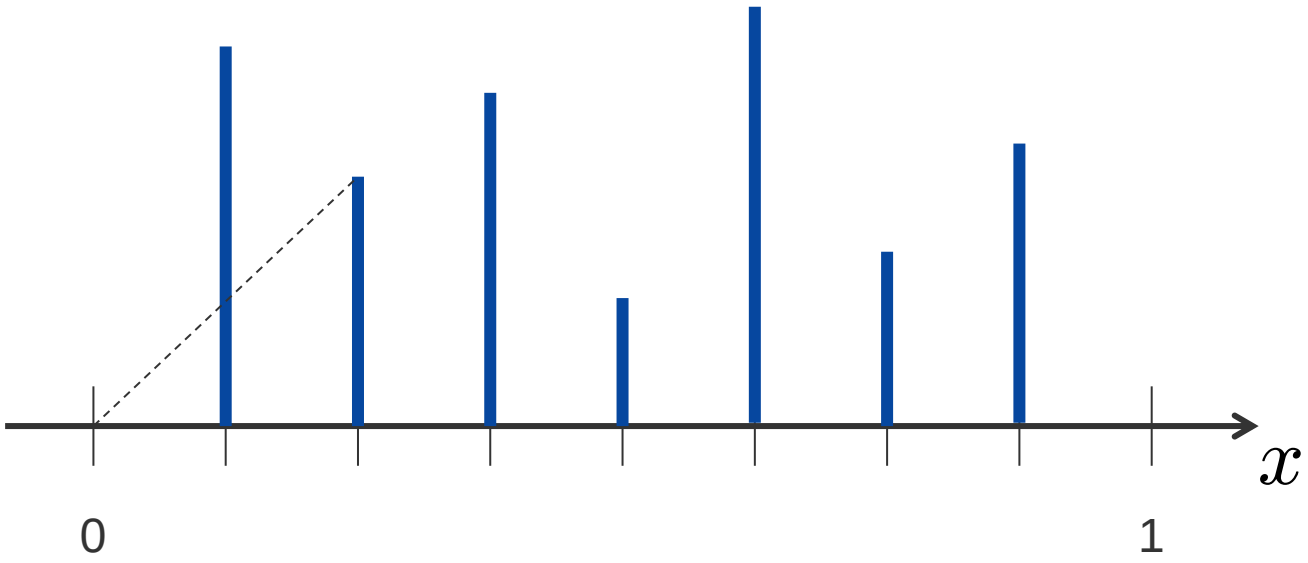
Gauss-Seidel Method Example – 1D Heat Equation

- Random initial value $u^0 = rand(N)$



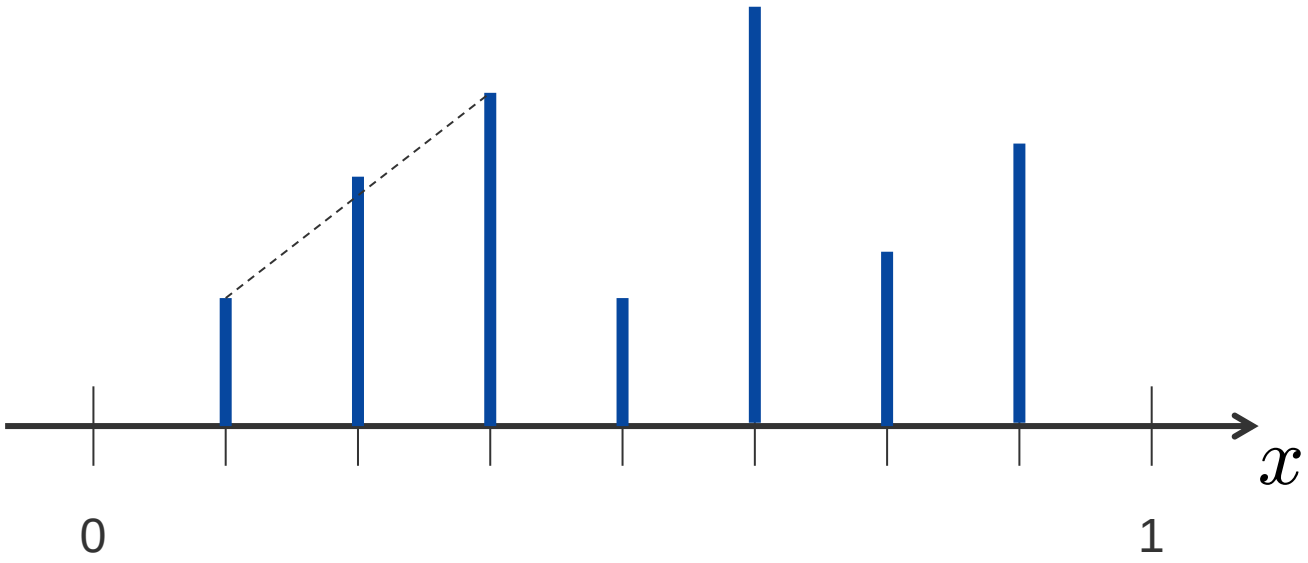
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 0$$



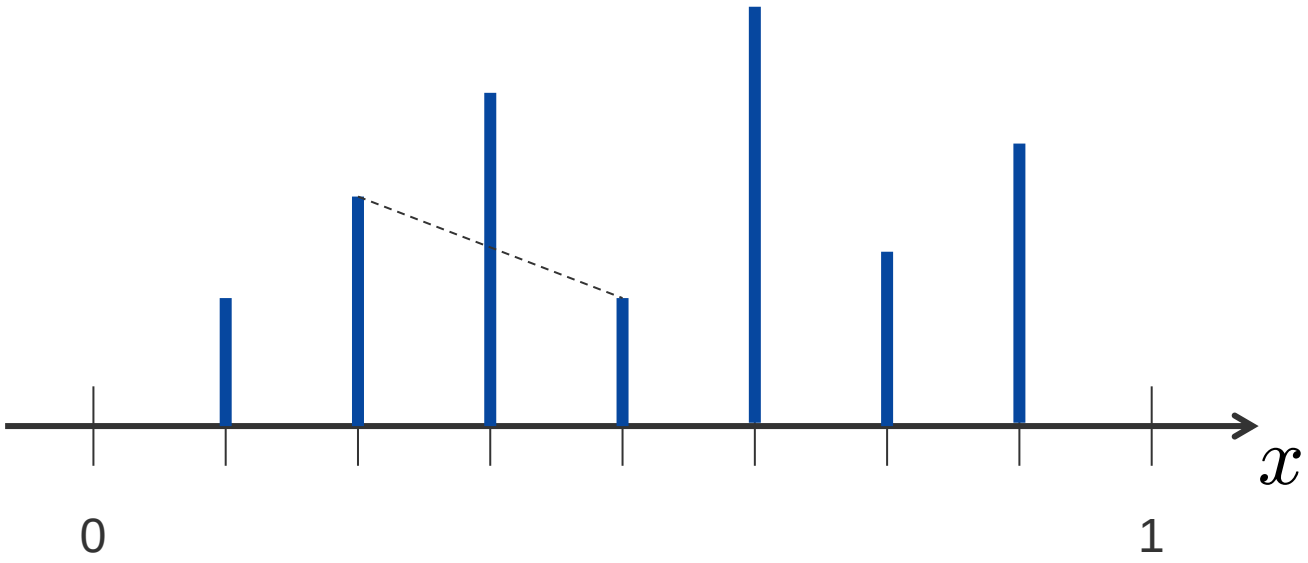
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 0$$



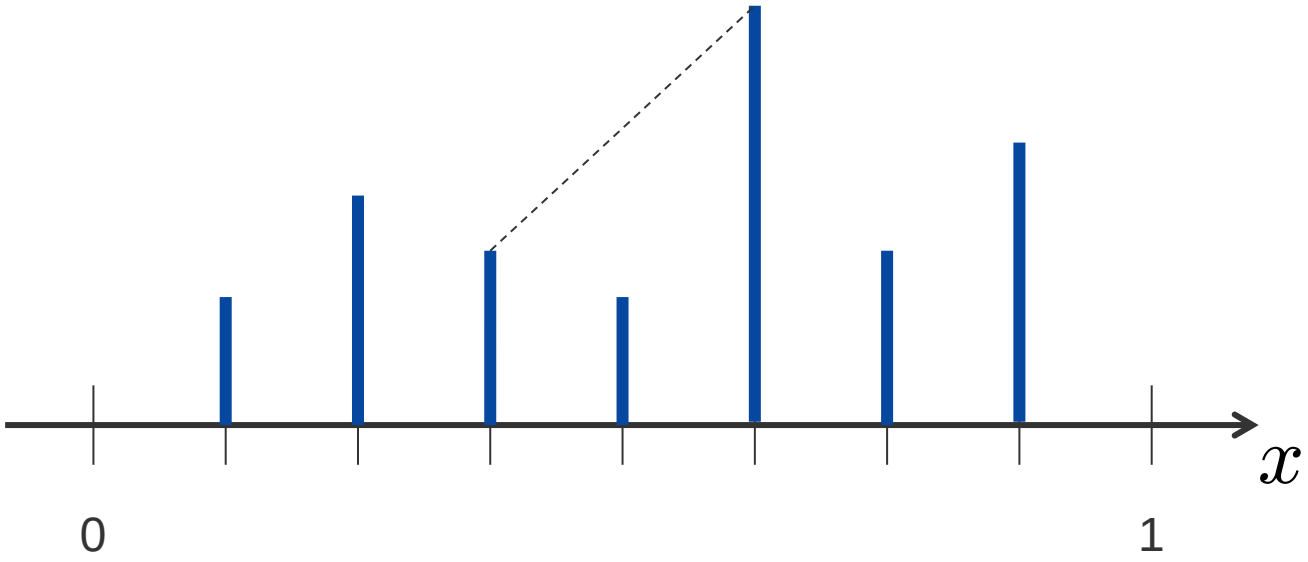
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 0$$



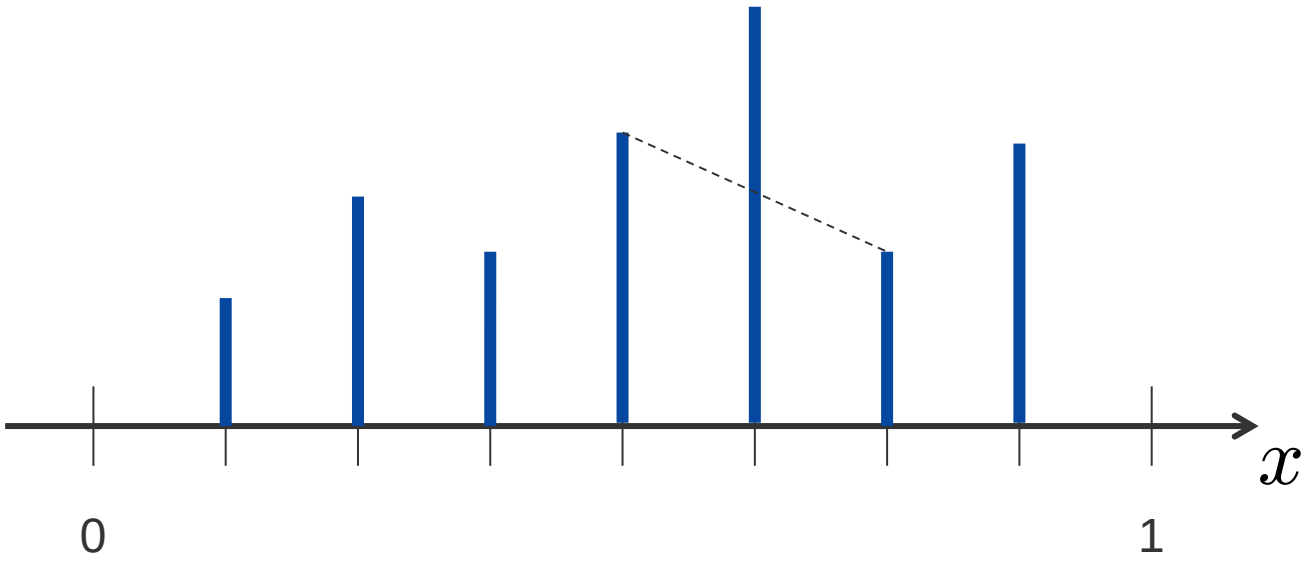
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 0$$



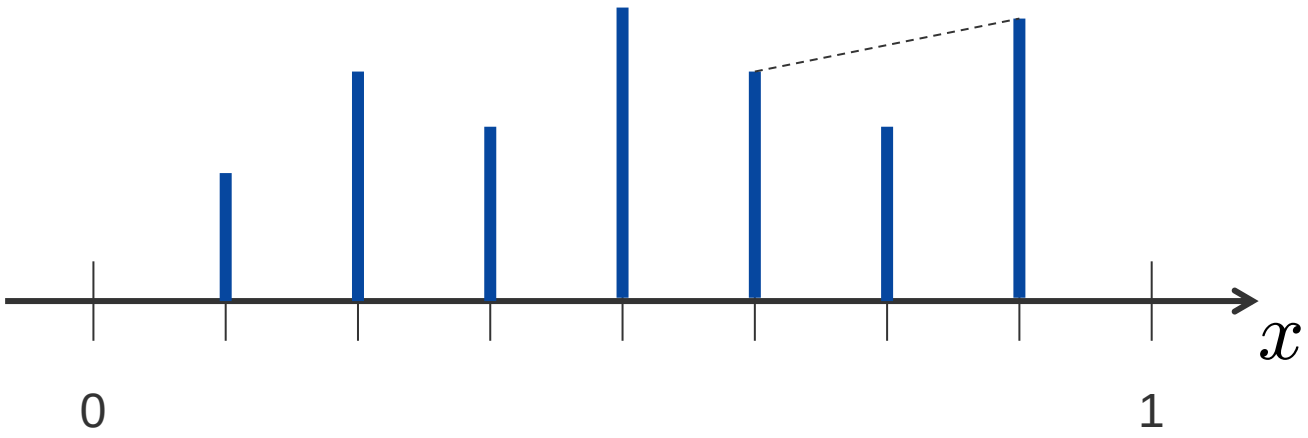
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 0$$



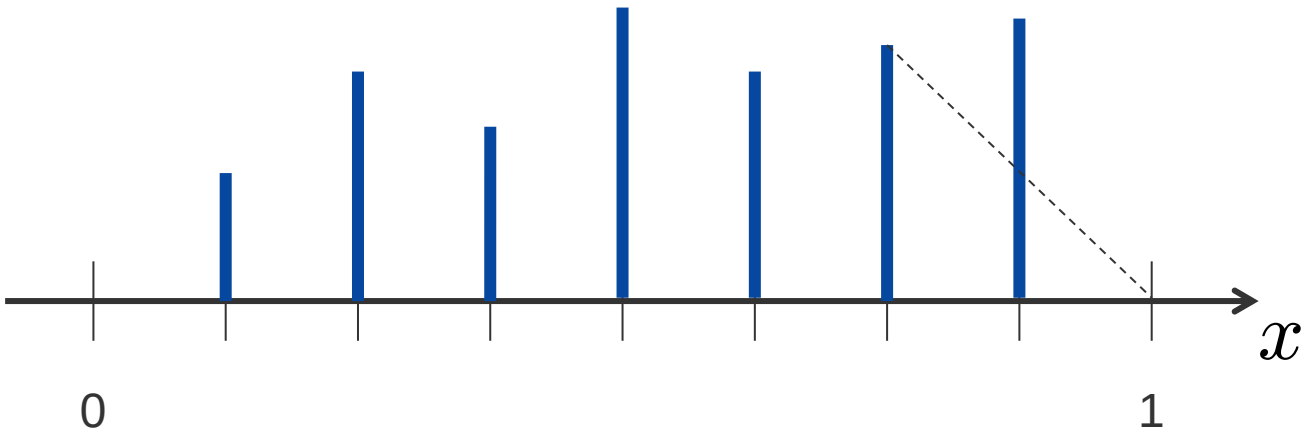
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 0$$



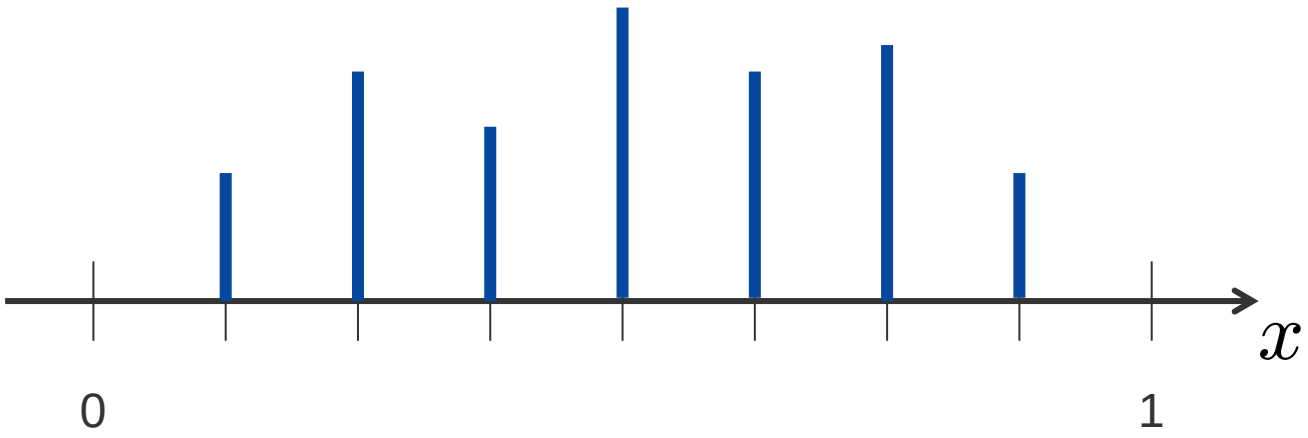
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 0$$



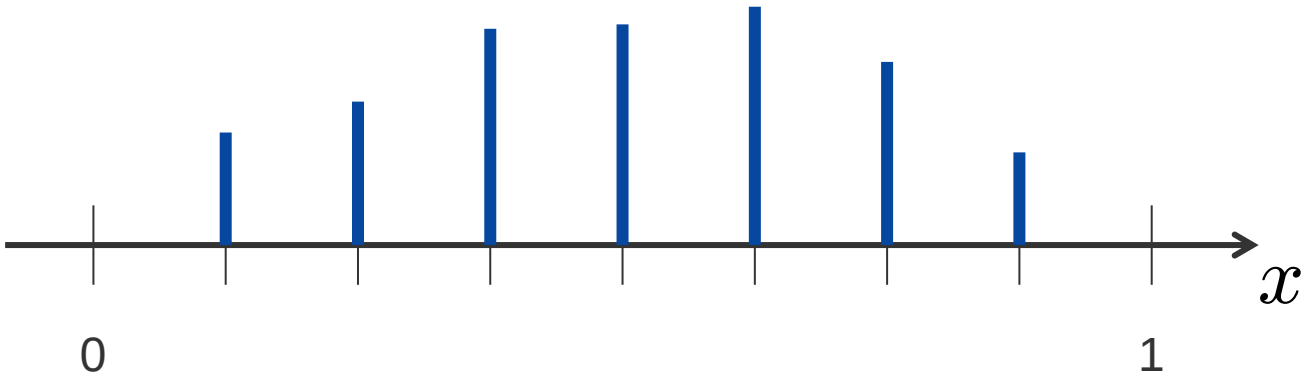
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 1$$



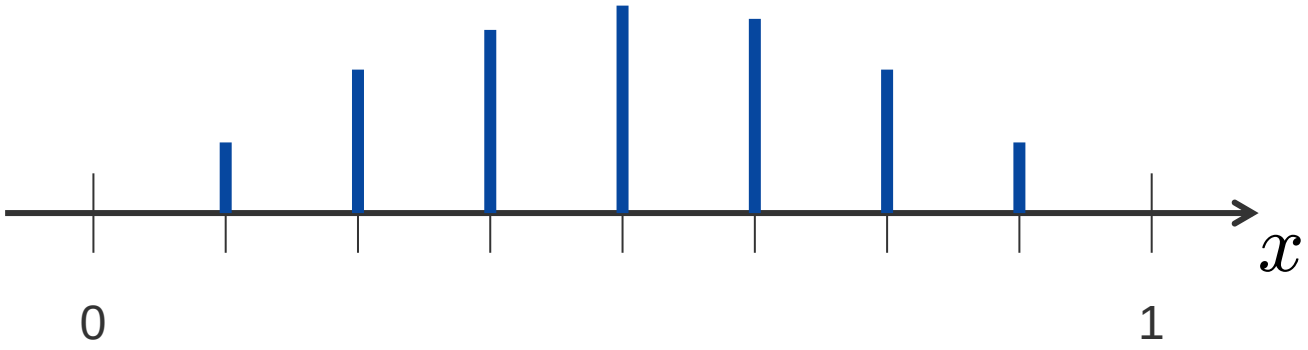
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 2$$



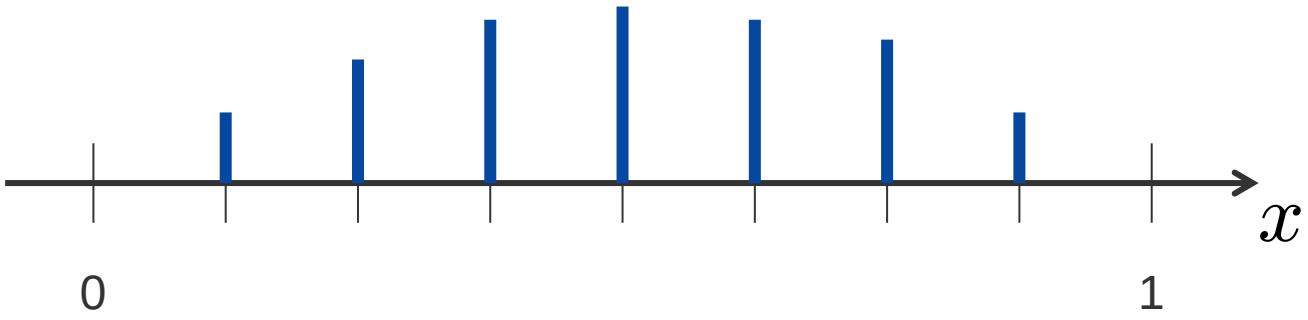
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 3$$



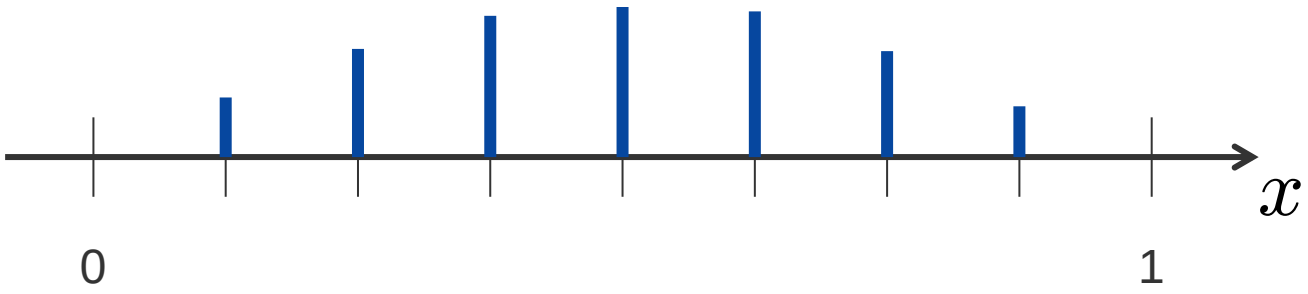
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 4$$



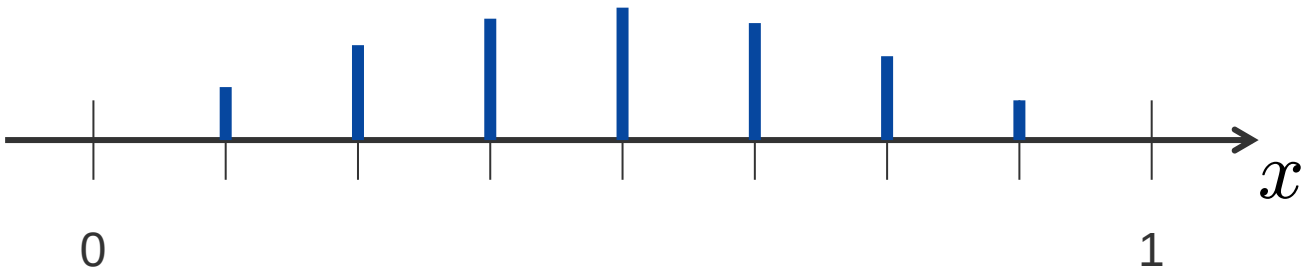
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 5$$



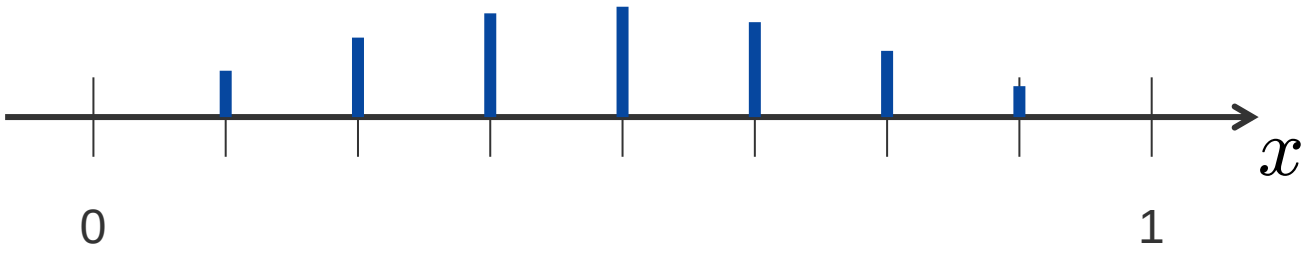
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 6$$



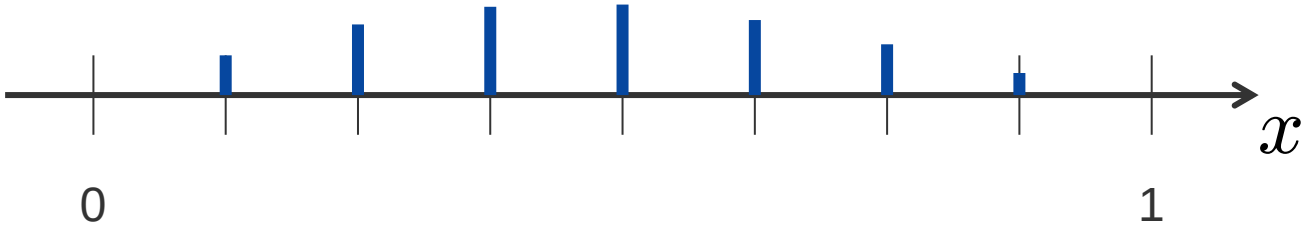
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 7$$



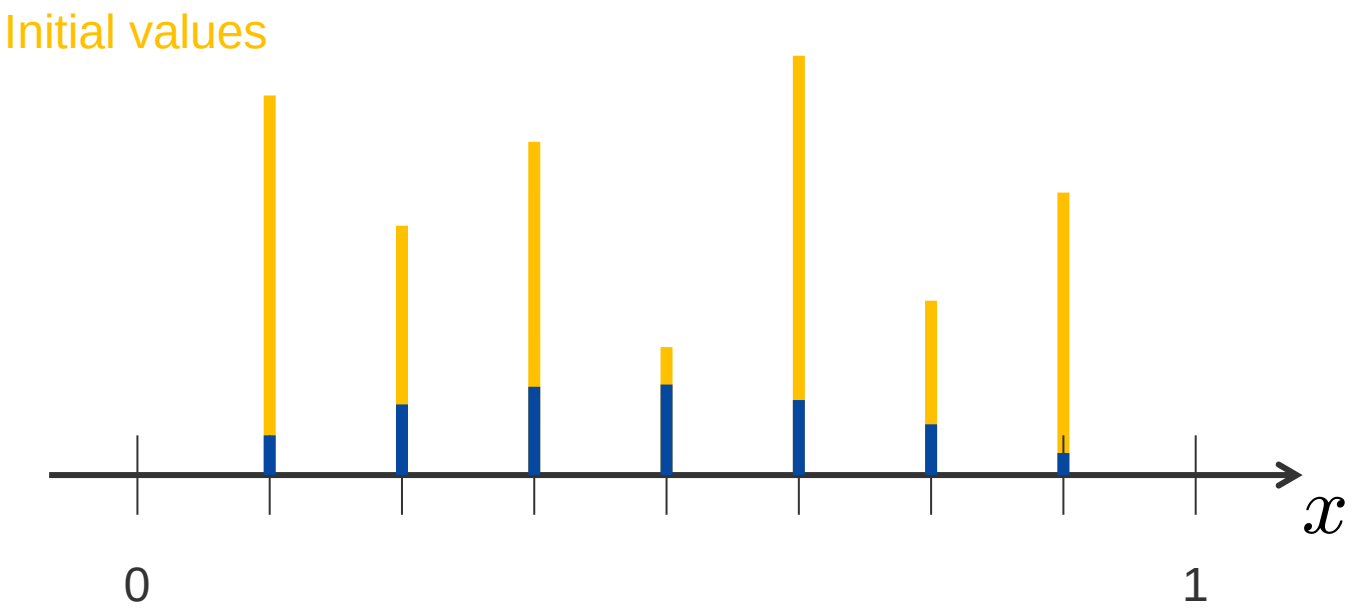
Gauss-Seidel Method Example – 1D Heat Equation

$$k = 8$$



Gauss-Seidel Method Example – 1D Heat Equation

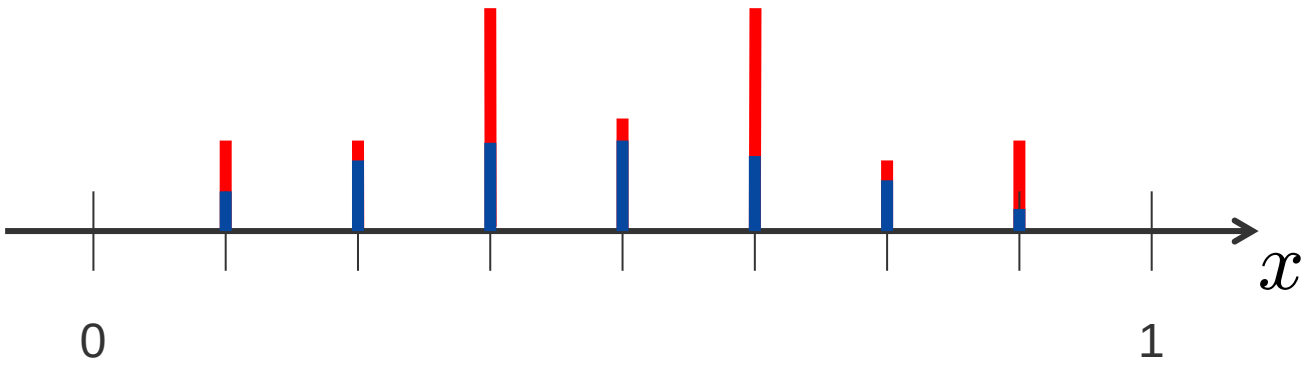
$$k = 8$$



Gauss-Seidel Method Example – 1D Heat Equation

$$k = 8$$

Jacobi after 8 iterations



Outline

- Poisson-Equation: From the Model to an Equation System
- Direct Solvers vs. Iterative Solvers
- Jacobi and Gauss-Seidel Method
- Residual and Smoothness
- Multigrid Solver

Residual

- Residual:

$$Au = f$$

$$Au^k \neq f$$

$$f - Au^k = res^k$$

- Splitting:

$$Au = (S + T)u = f$$

$$u^{k+1} = S^{-1}(f - Tu^k)$$

$$u^{k+1} = S^{-1}(f - (A - S)u^k)$$

$$u^{k+1} = S^{-1}(f - Au^k) + u^k$$

$$u^{k+1} = u^k + S^{-1}res^k$$

Residual for 1D Heat Equation Example

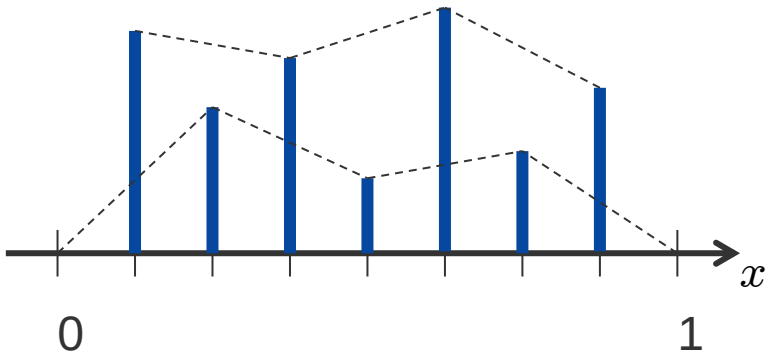
- Local influences form residual:

$$u_{i-1} - 2u_i + u_{i+1} = 0$$

$$u_{i-1}^k - 2u_i^k + u_{i+1}^k = res_i^k$$

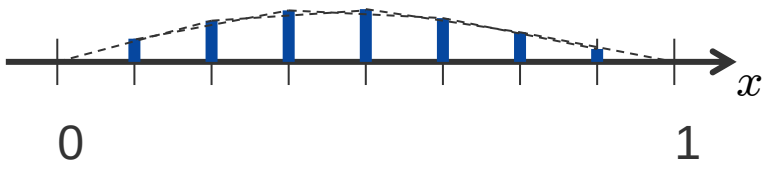
- GS $k = 0$: high residual

  → oscillatory error/fct values



- GS $k = 8$: low residual

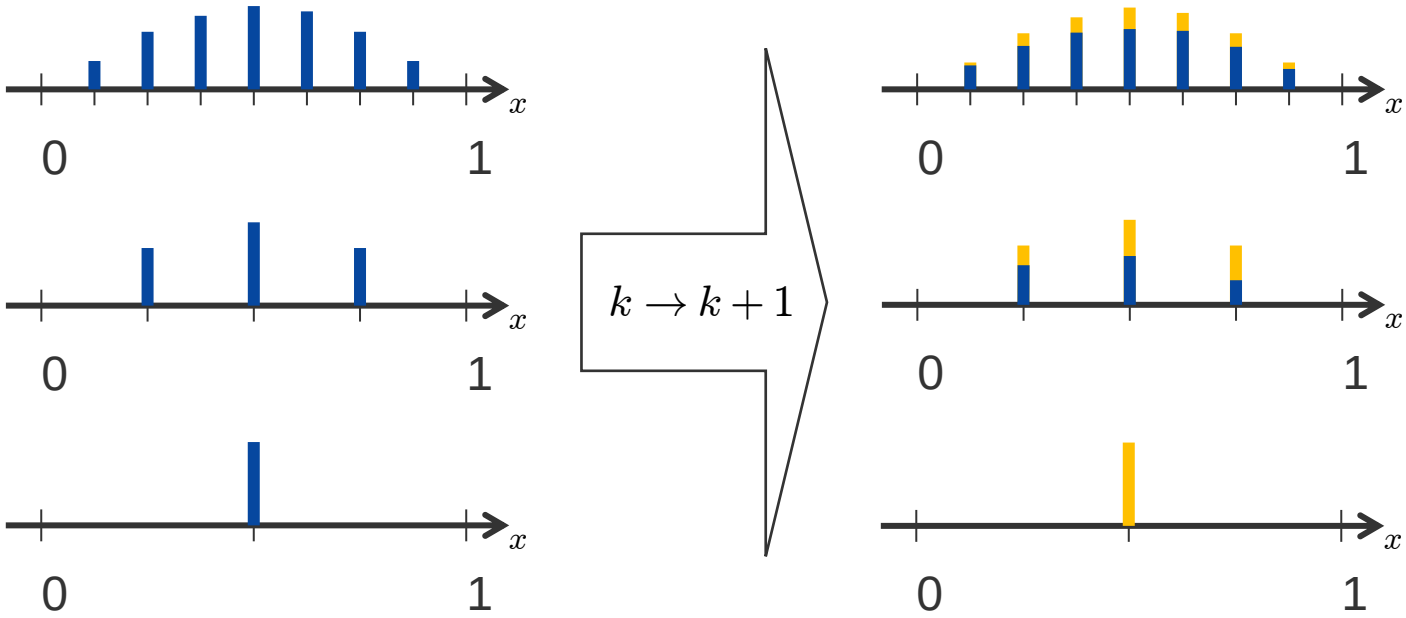
  → smooth error/fct values



Smoothness Observation

Smoothnes depends on grid resolution h :

- $h = \frac{1}{8}$:
- $h = \frac{1}{4}$:
- $h = \frac{1}{2}$:



Outline

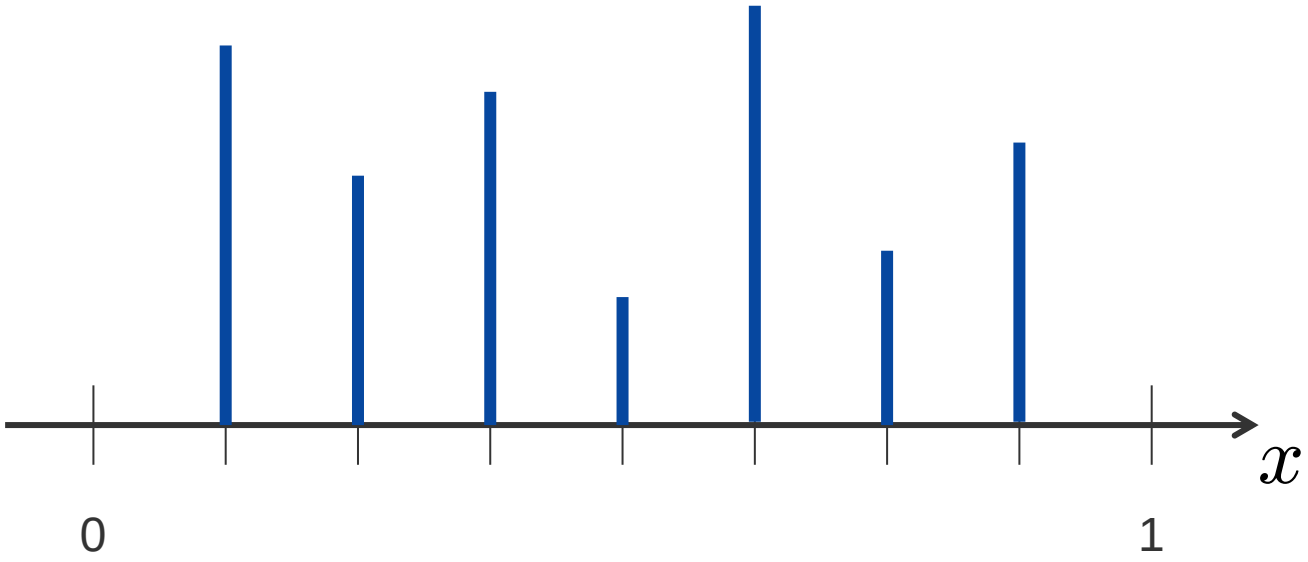
- Poisson-Equation: From the Model to an Equation System
- Direct Solvers vs. Iterative Solvers
- Jacobi and Gauss-Seidel Method
- Residual and Smoothness
- Multigrid Solver

Geometric Multigrid Idea

- Only high error frequencies are removed effectively
- Frequencies depend on grid resolution
- Idea:
  - Use several grid levels
  - Use splitting method to smooth error on each grid level

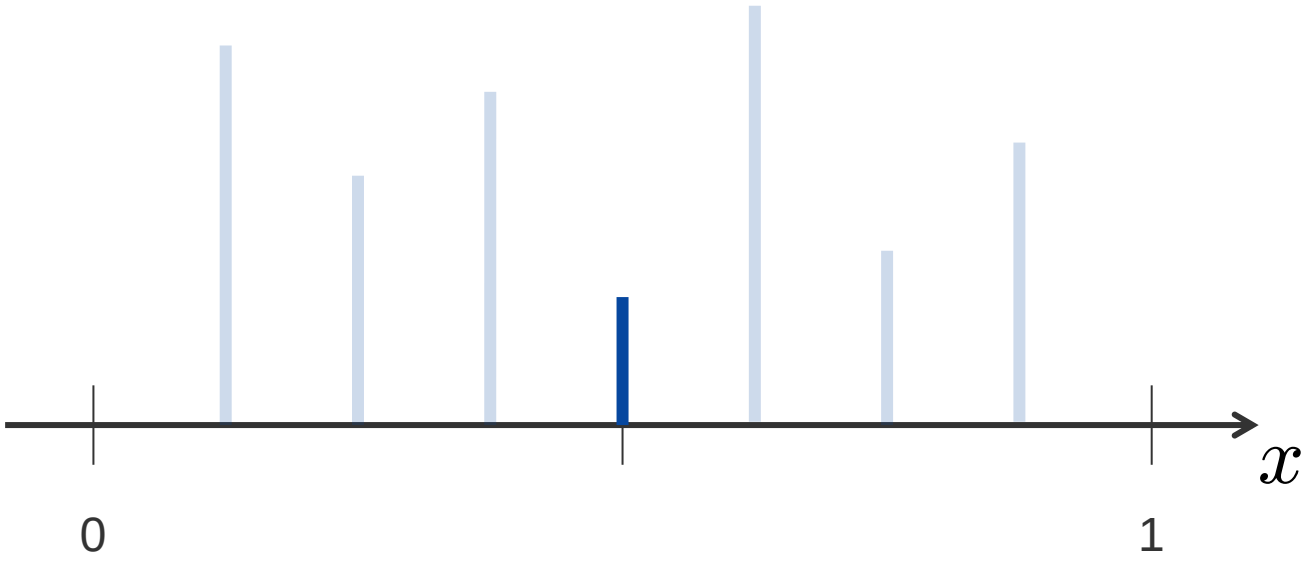
Multigrid Example – 1D Heat Equation

- Random initial value $u^0 = rand(N)$



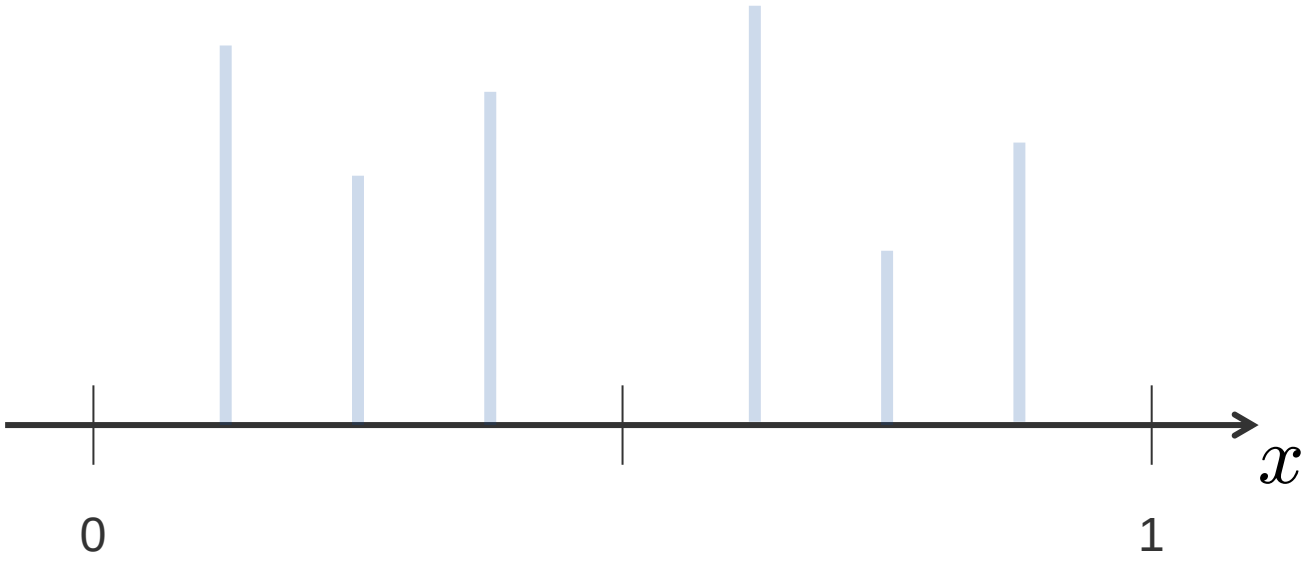
Multigrid Example – 1D Heat Equation

$$k = 0,\ N = 1$$



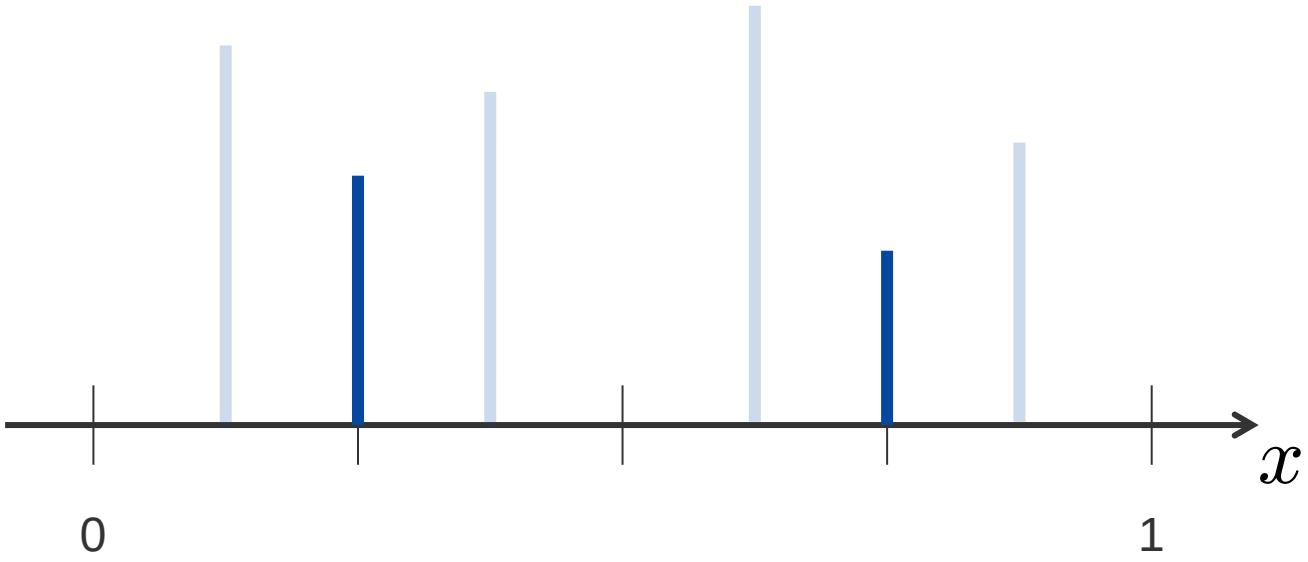
Multigrid Example – 1D Heat Equation

$$k = 0,\ N = 1$$



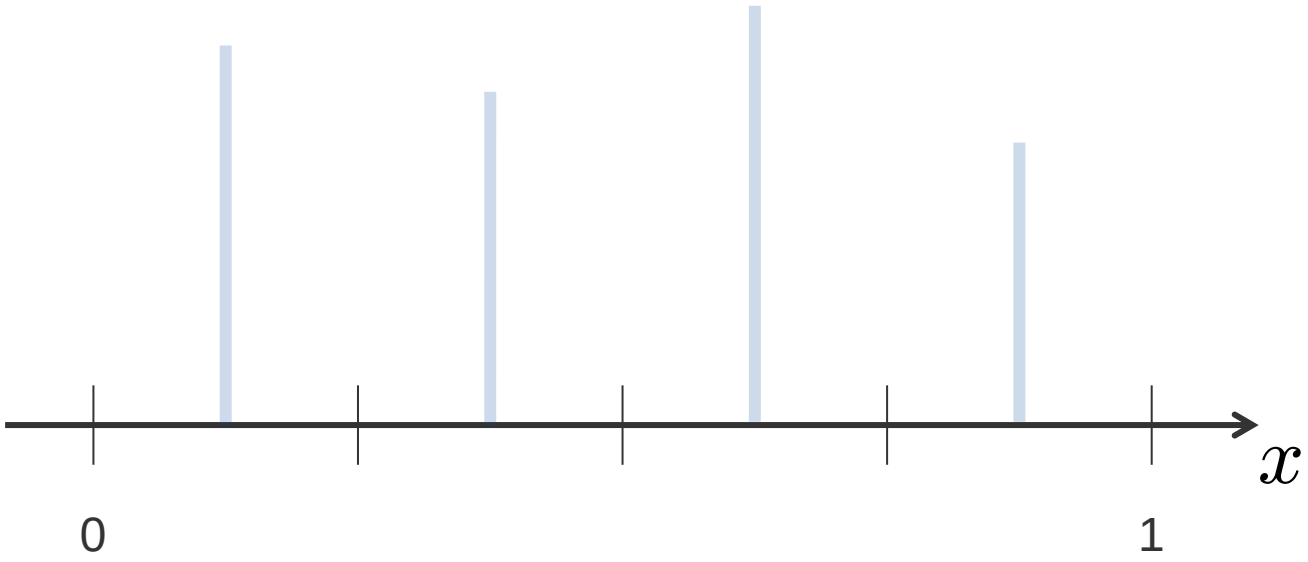
Multigrid Example – 1D Heat Equation

$$k = 0,\ N = 3$$



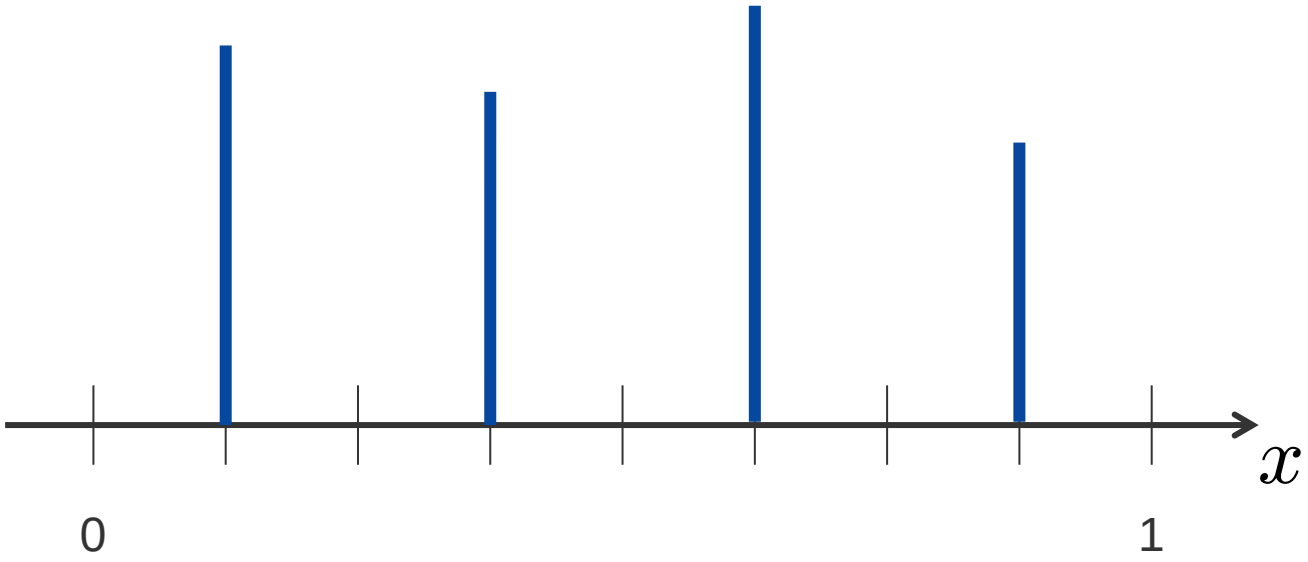
Multigrid Example – 1D Heat Equation

$$k = 0,\ N = 3$$



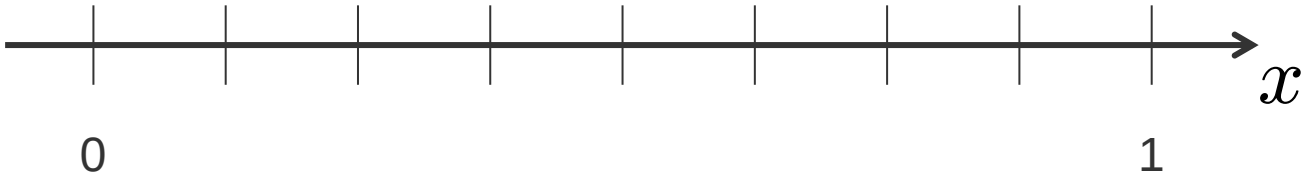
Multigrid Example – 1D Heat Equation

$$k = 0,\ N = 7$$



Multigrid Example – 1D Heat Equation

$$k = 1$$



Thank you for your attention!