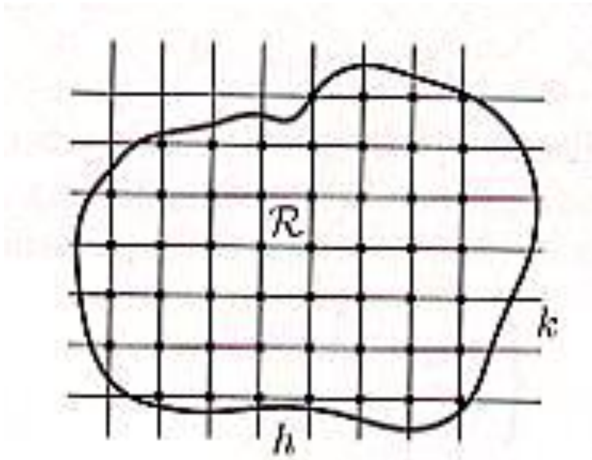
Numerical Methods for Partial Differential Equations (PDE) (2)

Finite Difference Methods

PDE - Finite Differences

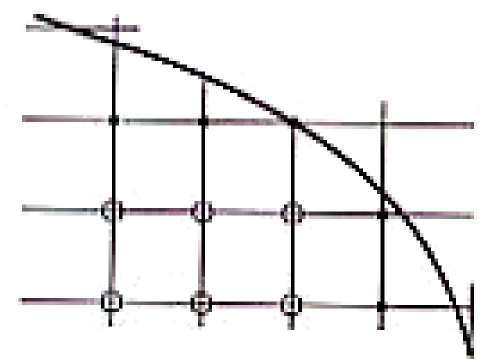
Basic Idea:

- Replace the region R with a (rectangular) grid (mesh) of points of R
- “collocate” the differential equation on the nodes of the mesh
- approximate (at nodes) the partial derivatives with numerical differentiation formulas (finite differences)



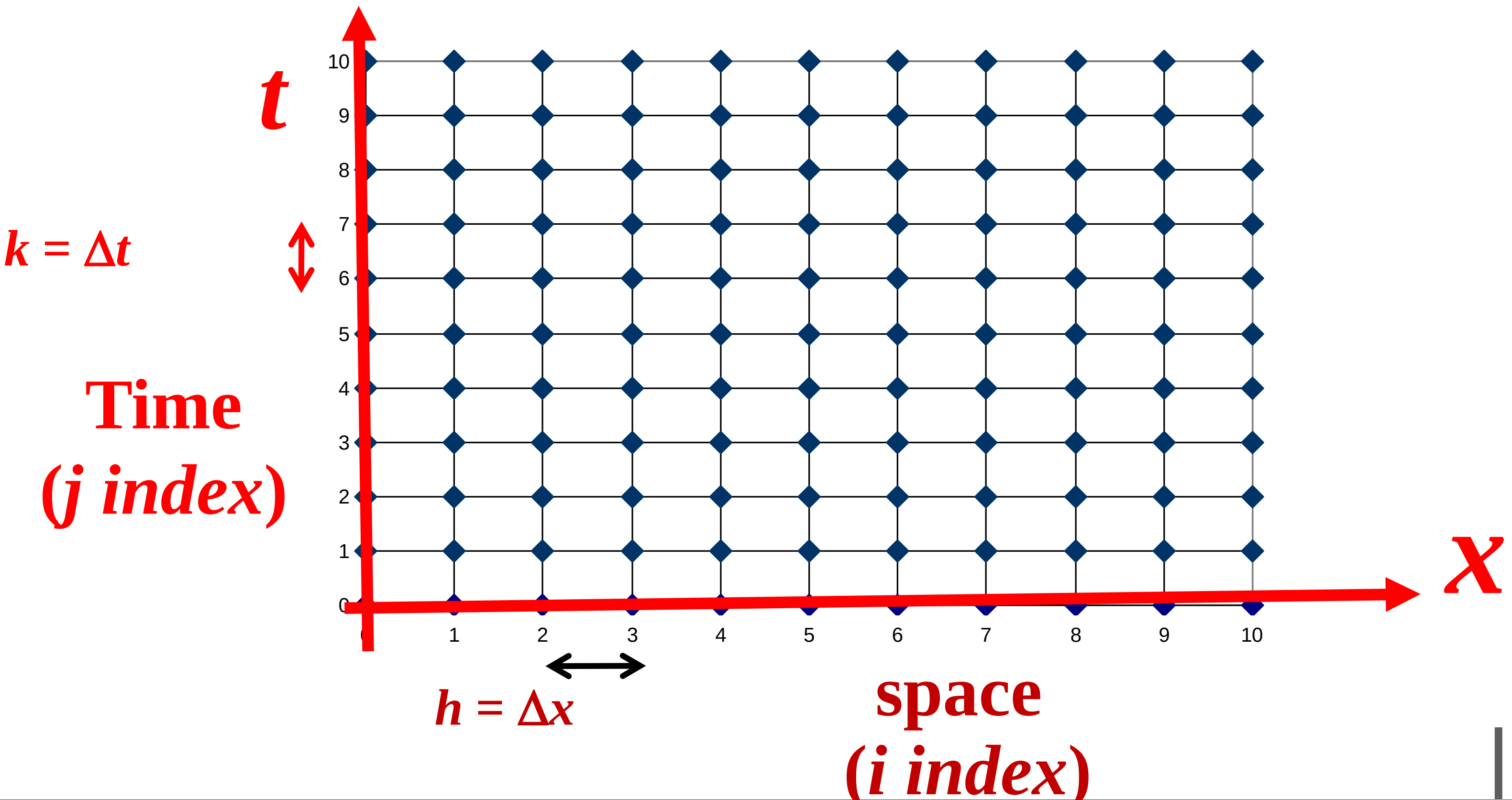
○ inner points

• boundary points



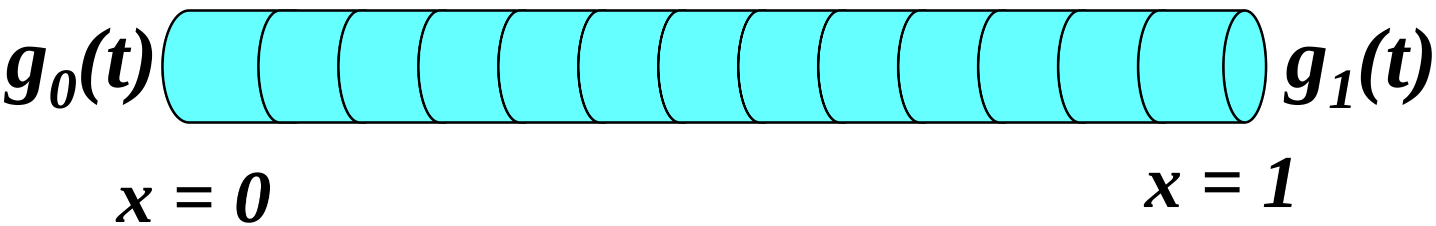
Domain Discretization

- Define the grid of points $(x_i, t_j) = (i\Delta x, j\Delta t)$,
- Discretize the continuous function u(x,t): $u_{i,j} = u(i\Delta x, j\Delta t)$.



1)Parabolic PDE 1D of order 2 heat or diffusion

Wire (of unit length) thermally isolated, the initial distribution of known temperature. The ends of the wire are maintained at known temperatures, in each instant t> 0. Compute the temperature distribution u(x,t) at subsequent instants to the initial time.



$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}; \qquad 0 < x < 1, \quad 0 \le t \le T$$

$$\mathrm{CI:} \quad u(x,0) = f(x), \qquad 0 < x < 1$$

$$\mathrm{CB:} \begin{cases} u(0,t) = g_0(t) \\ u(1,t) = g_1(t) \end{cases}, \qquad 0 \le t \le T$$

Dirichlet

Parabolic PDE : Heat or Diffusion

Well posed problem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}; \qquad 0 \le t \le T$$

Init.Cond. $\quad u(x,0) = f(x), \qquad -\infty < x < \infty$

Exact Solution

$$u(x,t) = \frac{1}{\sqrt{4\pi t}} \int_{-\infty}^{\infty} e^{-(x-s)^2/4t} f(s)\,ds$$

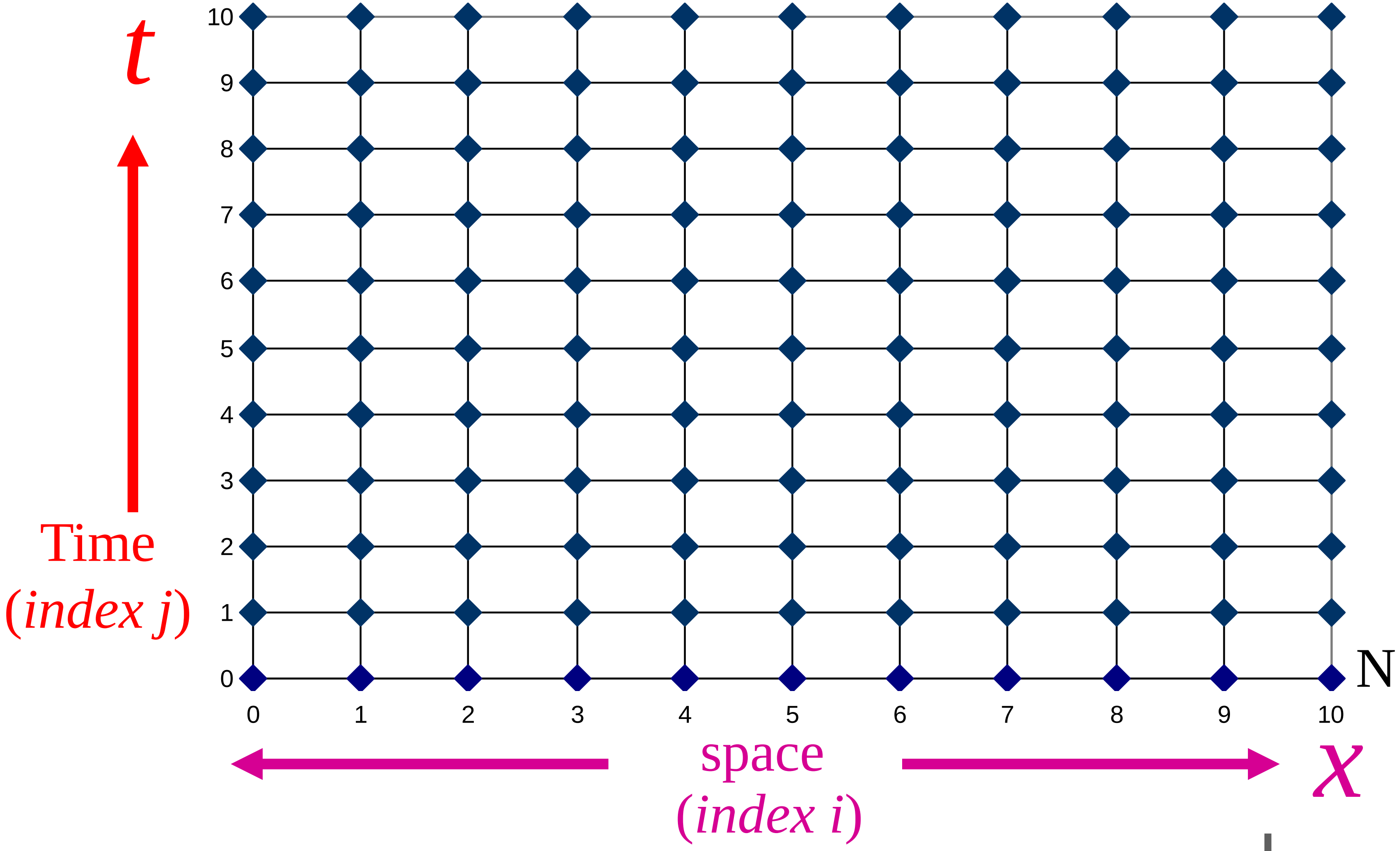
Smoothing effect: e.g., even if the initial datum *f(s)* is only bounded and locally piecewise continuous in the x direction, the solution *u(x,t)* is infinitely derivable for each t>0. Moreover $\lim_{\substack{x\to\xi \\ t\to 0}} u(x,t) = f(\xi)$

This behaviour is typical of parabolic PDEs.

Parabolic PDE : Heat or Diffusion

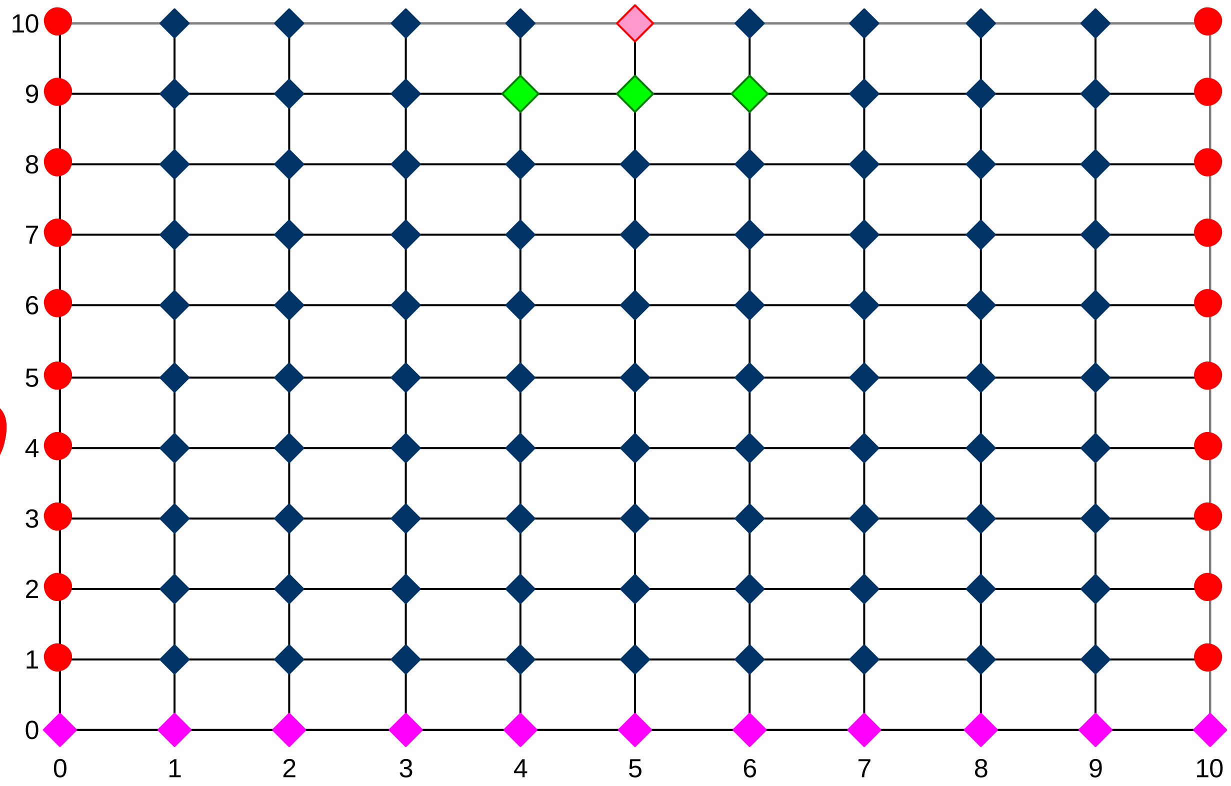
$$h = \Delta x$$

$$k = \Delta t$$



Boundary and initial conditions

$u(0, t) = g_0(t)$

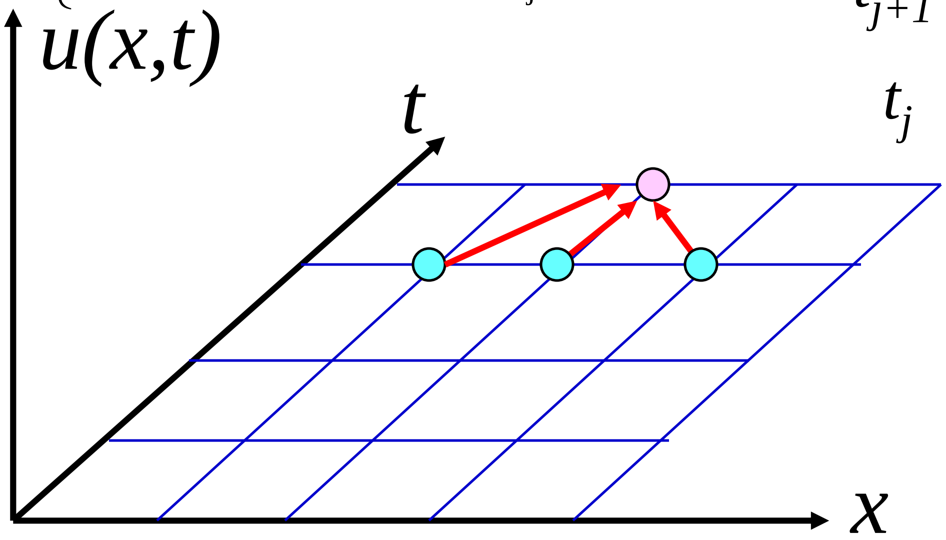


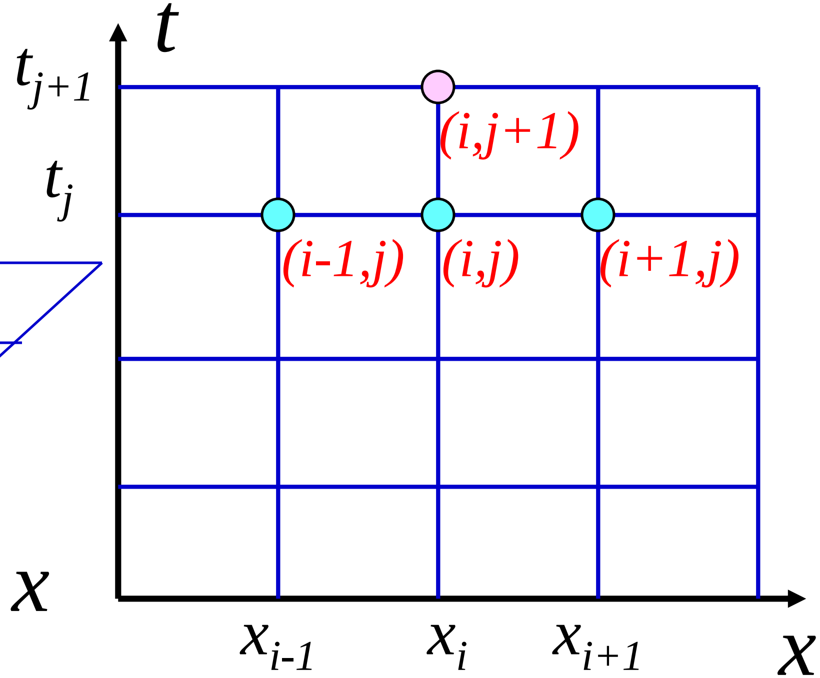
$u(1, t) = g_1(t)$

Initial conditions: $u(x,0) = f(x)$

Explicit method

Let $\begin{cases} h = \Delta x = 1/N, & x_i = ih \\ k = \Delta t = T/M, & t_j = jk \end{cases}$





Forward Difference

$$\frac{\partial u}{\partial t}\left(x_i, t_j\right) = \frac{1}{k}\left(u_i^{j+1} - u_i^j\right) \quad +O(k)$$

Centered Difference at time j

$$\frac{\partial^2 u}{\partial x^2}\left(x_i, t_j\right) = \frac{1}{h^2}\left(u_{i-1}^j - 2u_i^j + u_{i+1}^j\right) \quad +O(h^2)$$

Explicit method

$$u_t = u_{xx}$$

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

$$\begin{aligned} u_{i,j+1} &= u_{i,j} + \frac{k}{h^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) \\ &= ru_{i-1,j} + (1-2r)u_{i,j} + ru_{i+1,j} \end{aligned}$$

$$r = \frac{k}{h^2} = \frac{\Delta t}{\Delta x^2}$$

$$CI: u_{i,0} = u_0(x_i) \quad i = 0,1,..,n$$

$$CB: u_{0,t} = g_0(t), \quad u_{n,t} = g_1(t)$$

$$\begin{bmatrix} u_{1,j+1} \\ .. \\ .. \\ u_{N-1,j+1} \end{bmatrix} = \begin{bmatrix} 1-2r & r & & 0 \\ r & 1-2r & r & \\ & .. & .. & r \\ 0 & & r & 1-2r \end{bmatrix} \begin{bmatrix} u_{1,j} \\ .. \\ .. \\ u_{N-1,j} \end{bmatrix} + \begin{bmatrix} ru_{0,j} \\ 0 \\ 0 \\ ru_{N,j} \end{bmatrix}$$

$$U_{j+1} = AU_j + v \qquad A \in R^{(N-1)x(N-1)} \qquad v, U \in R^{(N-1)}$$

Conditioned Stability: $0 < r \leq 0.5$

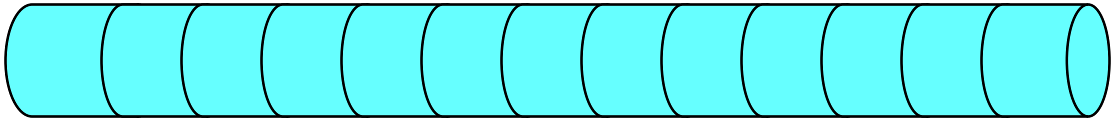
$$\begin{bmatrix} u_{1,j+1} \\ .. \\ \\ .. \\ u_{N-1,j+1} \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & & 0 \\ 0 & 1 & 0 & \\ & 0 & .. & 0 \\ 0 & & 0 & 1 \end{bmatrix} + r \begin{bmatrix} -2 & 1 & & 0 \\ 1 & -2 & 1 & \\ & .. & .. & 1 \\ 0 & & 1 & -2 \end{bmatrix} \right) \begin{bmatrix} u_{1,j} \\ .. \\ \\ .. \\ u_{N-1,j} \end{bmatrix} + \begin{bmatrix} ru_{0,j} \\ 0 \\ \\ 0 \\ ru_{N,j} \end{bmatrix}$$

$$U_{j+1} = (I + rL)U_j + v \qquad L \in R^{(N-1)x(N-1)} \quad v, U \in R^{(N-1)}$$

$\frac{1}{h^2} L$ second order derivative matrix, discretization of the Laplace operator

Heat equation and Neumann boundary cond.

No heat flow at boundaries $x = 0$ e $x = 1$



$$\frac{\partial u}{\partial t} = c\frac{\partial^2 u}{\partial x^2}; \qquad 0 < x < 1,\ 0 \leq t \leq T$$

$$u(x,0) = f(x), \qquad 0 < x < 1$$

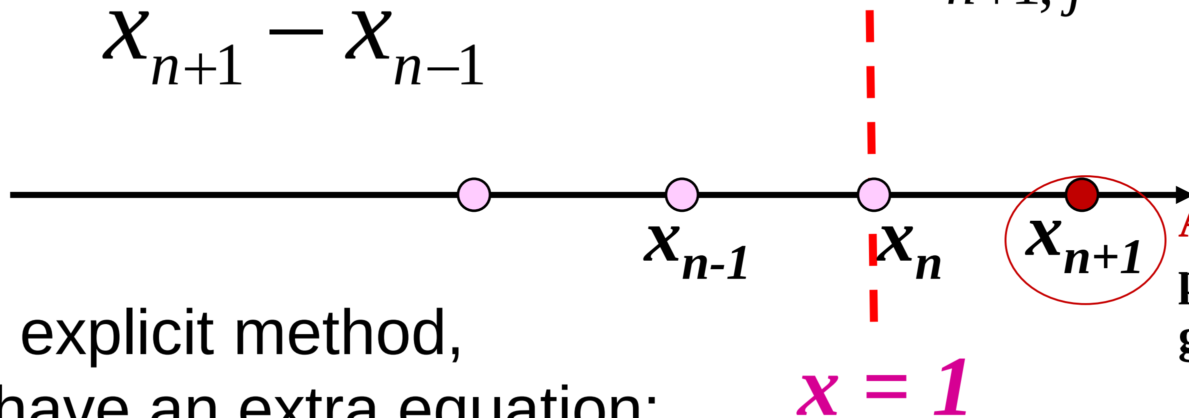
CB:
Neumann

$$\begin{cases} u_x(0,t) = 0 \\ u_x(1,t) = 0 \end{cases}, \qquad 0 \leq t \leq T$$

Homogeous Neumann Boundary Conditions: isolated bounds

No heat flow at boundaries *x = 1* $u_x(1,t)=0$

$$u_x(1,t_j) = \frac{u_{n+1,j} - u_{n-1,j}}{x_{n+1} - x_{n-1}} = 0 \Rightarrow u_{n+1,j} = u_{n-1,j}$$



Add a ghost point to the grid

Applying the explicit method,
if i = n, we have an extra equation:

$$u_{n,j+1} = ru_{n-1,j} + (1-2r)u_{n,j} + ru_{n+1,j}$$
$$= 2ru_{n-1,j} + (1-2r)u_{n,j}$$

$$r = c\frac{k}{h^2} = c\frac{\Delta t}{\Delta x^2}$$

$$CI : u_{i,0} = u_0(x_i) \quad i = 0,1,..,n$$

$$CB : u_x(0,t) = u_x(1,t) = 0$$

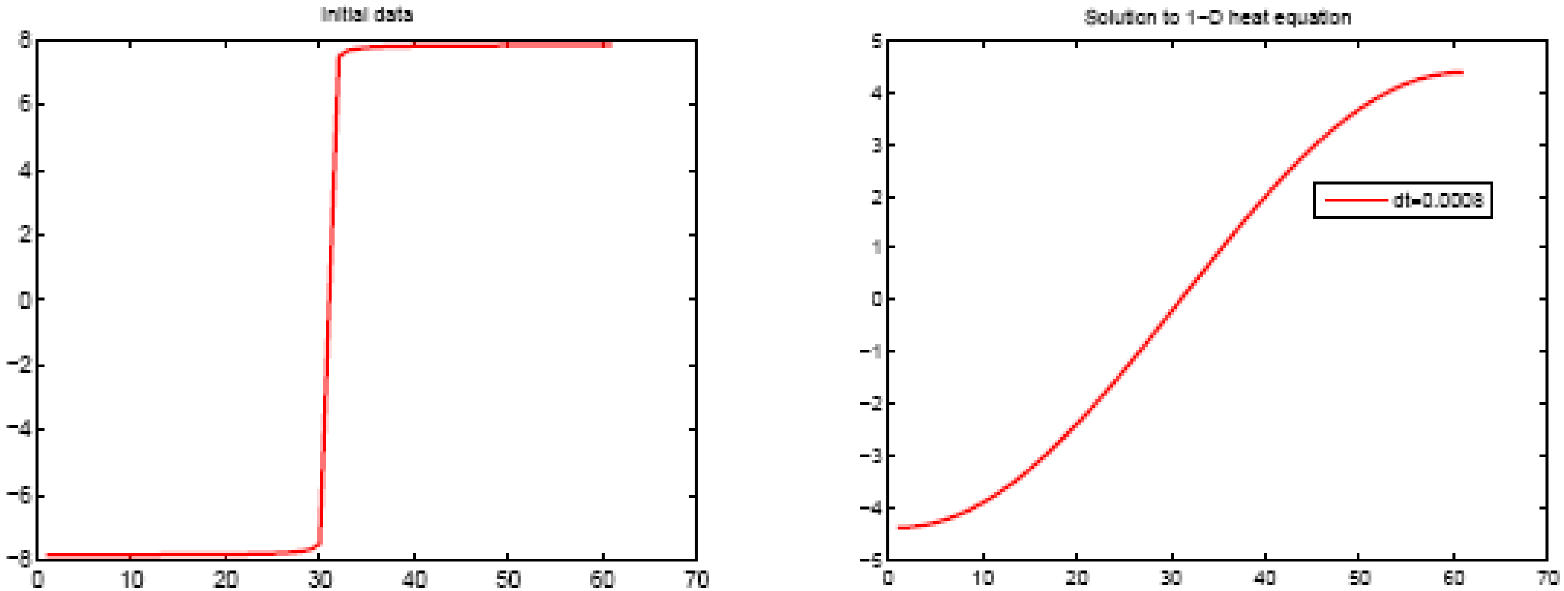
$$\begin{bmatrix} u_{0,j+1} \\ u_{1,j+1} \\ .. \\ u_{i,j+1} \\ .. \\ u_{N-1,j+1} \\ u_{N,j+1} \end{bmatrix} = \begin{bmatrix} 1-2r & 2r & 0 & & \\ r & 1-2r & r & & \\ & .. & .. & r \\ & & 2r & 1-2r \end{bmatrix} \begin{bmatrix} u_{0,j} \\ u_{1,j} \\ .. \\ u_{i,j} \\ .. \\ u_{N-1,j} \\ u_{N,j} \end{bmatrix}$$

$$U_{j+1} = AU_j \quad A \in R^{(N+1)x(N+1)}, U \in R^{N+1}$$

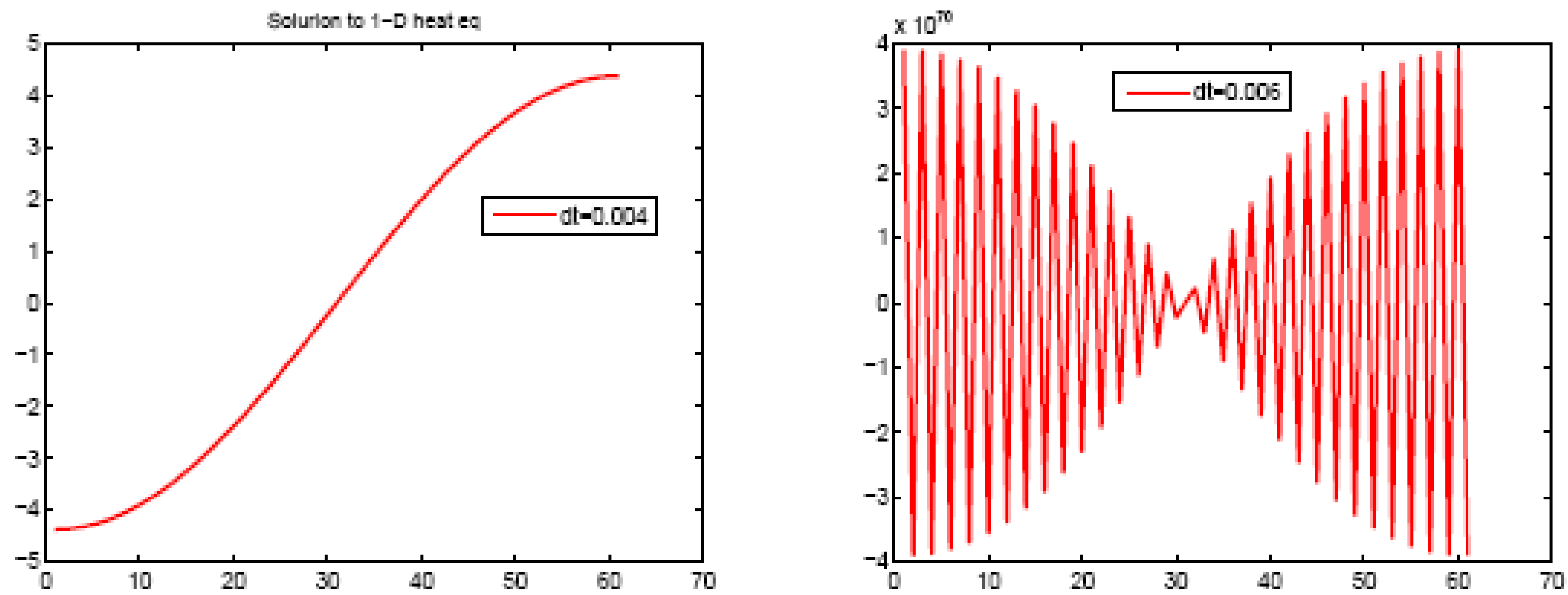
Truncation Error $O(k + h^2)$

Conditioned Stability: $0 < r \leq 0.5$

Stability: example heat eq.



u_0 initial



Local Truncation Error and order of accuracy

- **Consistency**: For a given numerical scheme, the local truncation error is the error that is generated by demanding that the exact solution verifies the numerical scheme itself.

  If the truncation error $\tau(k,h) = \max_{i,j} \left| \tau_i^j \right| \quad k = \Delta t$

  tends to zero as h and k tend to zero, independently, then the numerical scheme will be defined **consistent.**

- A numerical scheme is said to be order p accurate in space and order q accurate in time (for suitable integers p and q), if for a sufficiently smooth solution of the exact problem, we have

$$\tau(k,h) = O(k^p + h^q)$$

Stability and convergence

- A method is said to be **convergent** *(in max-norm) if*

$$\lim_{k,h\to 0}\left(\max\left|y(x_i,t^j)-u_{i,j}\right|\right)=0$$

Consistency + stability $\Rightarrow$ convergence

- **Stability**

  Perturbations on the initial data (initial conditions) are not amplified by the numerical procedure in the solution. Numerical scheme in general is

(*)

$$U_{j+1}=A(k)U_j+b_j(k)$$

$$A(k)\in R^{NxN} \quad \text{space grid } h=\frac{1}{N+1} \quad b(k)\in R^N$$

Convergence

Def. A linear method of the form (*) is Lax-Richtmyer stable if, for each timeT, there is a constant $C_T>0$ such that

$$\exists C_T > 0 \quad s.t.$$

$$\left\| A(k)^j \right\| \leq C_T$$

$$\forall k > 0, \textit{ and integers } j \textit{ for which } kj \leq T$$

Lax Equivalence Theorem

A consistent linear method of the form (*) is convergent if and only if it is Lax-Richtmyer stable.

Stability of the Explicit Method

Exact solution at time step j: U_j

U_{j+1} is given by multiplying the tridiagonal matrix A for the solution at the previous time step j-th:

$$U_{j+1} = AU_j$$

$$U_{j+1} = A^2 U_{j-1} = \dots = A^{j+1} U_0$$

Let us introduce some perturbations E_0 on the data U_0

$$\bar{U}_0 = U_0 + E$$

Computed Solution after j+1 steps

$$\bar{U}_{j+1} = A^{j+1} \bar{U}_0 = A^{j+1} (U_0 + E) = A^{j+1} U_0 + A^{j+1} E$$

The effect of the error E after j+1 steps is **$A^{j+1}E$**

Stability of the Explicit Method

$$\left\| A^{j+1} E \right\| \leq \left| \lambda \right|^{j+1} \left\| E \right\|$$

$$\left\| A \right\|_2 = \rho(A) = \max_{1 \leq p \leq N+1} \left| \lambda_p \right|$$

where λ is the dominant eigenvalue of A

- The method is strongly stable when $\left\| \boldsymbol{A} \right\|_2 \leq 1$
- Moreover, if the eigenvalues of A are all distinct, the method is stable if and only if

$$\boldsymbol{\rho(A) \leq 1}$$

That is $\left| \lambda \right| \leq 1$

Stability of the Explicit Method

The matrix A can be decomposed in the form **A = I+rL** where

$$L = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & .. & ... & .. & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}$$

The eigenvalues of L are $\lambda_p = 2(\cos(p\pi h) - 1) \quad p = 1, .., N-1, h = 1/N$

$$\lambda_{max} = -4$$

Then, the eigenvalues of A satisfy : $|1 + r\lambda| \leq 1$

The method is then stable when
that is

$$-2 \leq -4\frac{k}{h^2} \leq 0$$

$$\frac{k}{h^2} \leq \frac{1}{2} \rightarrow k(=\Delta t) \leq \frac{h^2}{2}$$

$r \leq 0.5$

Example: Explicit Method

$$u_{i,j+1} = ru_{i-1,j} + (1-2r)u_{i,j} + ru_{i+1,j}$$

- Stable

$$r = 0.01 \Rightarrow u_{i,j+1} = 0.01\,u_{i-1,j} + 0.98\,u_{i,j} + 0.01\,u_{i+1,j}$$

$$r = 0.1 \Rightarrow u_{i,j+1} = 0.1\,u_{i-1,j} + 0.8\,u_{i,j} + 0.1\,u_{i+1,j}$$

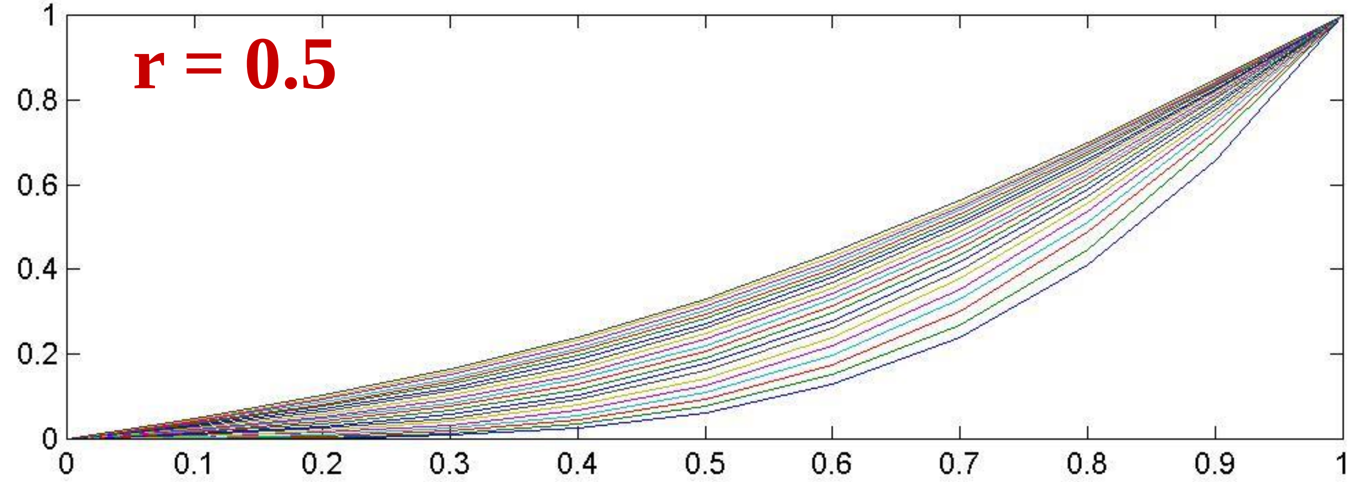
$$r = 0.4 \Rightarrow u_{i,j+1} = 0.4\,u_{i-1,j} + 0.2\,u_{i,j} + 0.4\,u_{i+1,j}$$

$$r = 0.5 \Rightarrow u_{i,j+1} = 0.5\,u_{i-1,j} + 0.5\,u_{i+1,j}$$

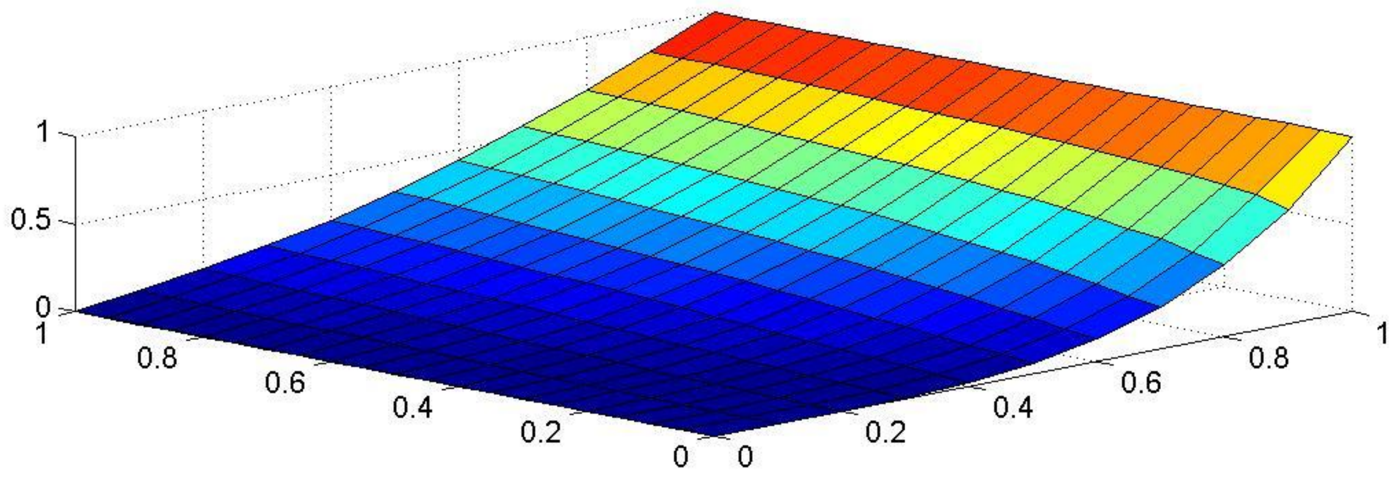
- Unstable (negative coefficients)

$$\begin{cases} r = 1 \Rightarrow u_{i,j+1} = u_{i-1,j} - u_{i,j} + u_{i+1,j} \\ r = 10 \Rightarrow u_{i,j+1} = 10\,u_{i-1,j} - 19\,u_{i,j} + 10\,u_{i+1,j} \\ r = 100 \Rightarrow u_{i,j+1} = 100u_{i-1,j} - 199\,u_{i,j} + 100u_{i+1,j} \end{cases}$$

Heat equation: explicit method



Accurato dell’ordine di $O(h^2)$, poichè richiesto $k=O(h^2)$ per stabilità



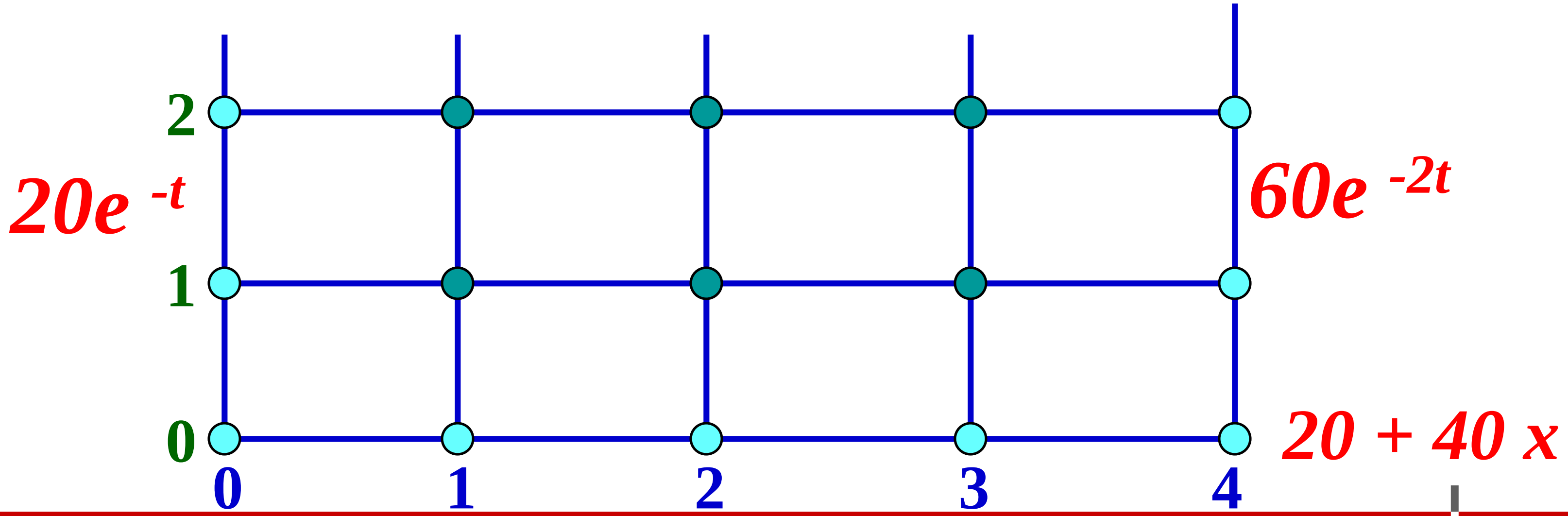
Example: explicit method, stable solution

- Heat Equation (Parabolic PDE)

$$u_t = c u_{xx}\,;\ 0 \le x \le 1$$

$$\begin{cases} u(x,0) = 20 + 40x \\ u(0,t) = 20e^{-t},\ u(1,t) = 60e^{-2t} \end{cases}$$

c = 0.5, h = 0.25, k = 0.05



Example: explicit method, stable solution

- **Compute r**

$$r = \frac{ck}{h^2} = \frac{(0.5)(0.05)}{(0.25)^2} = 0.4$$

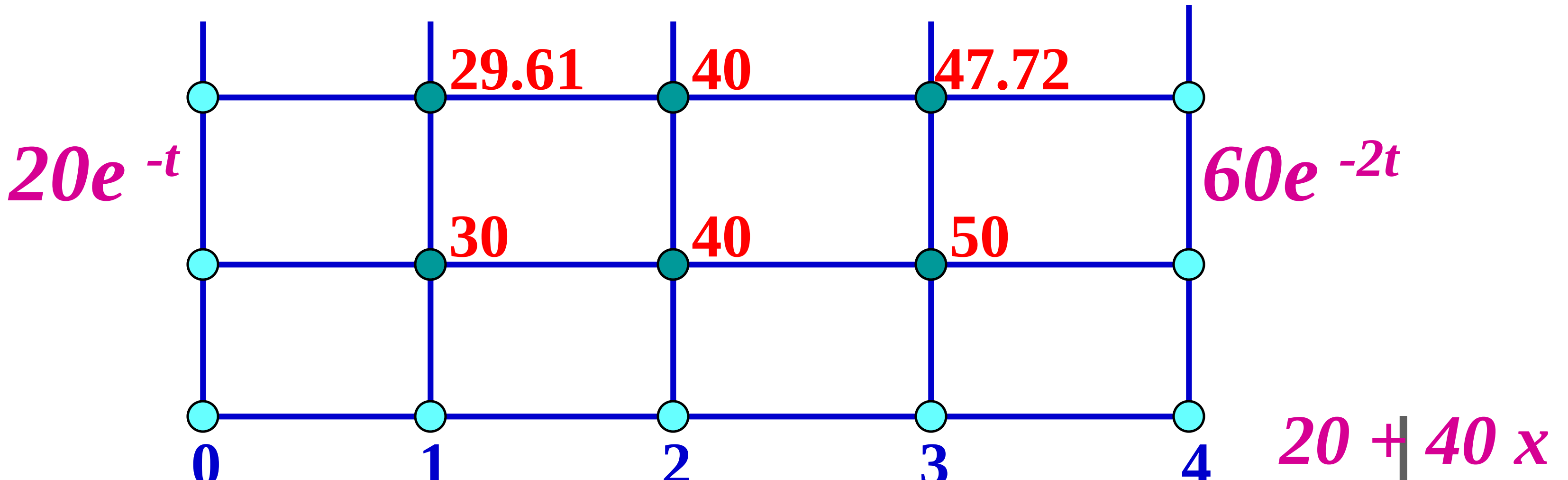
$$u_{i,j+1} = ru_{i-1,j} + (1-2r)u_{i,j} + ru_{i+1,j}$$
$$= 0.4u_{i-1,j} + 0.2u_{i,j} + 0.4u_{i+1,j}$$

- First Step: t = 0.05

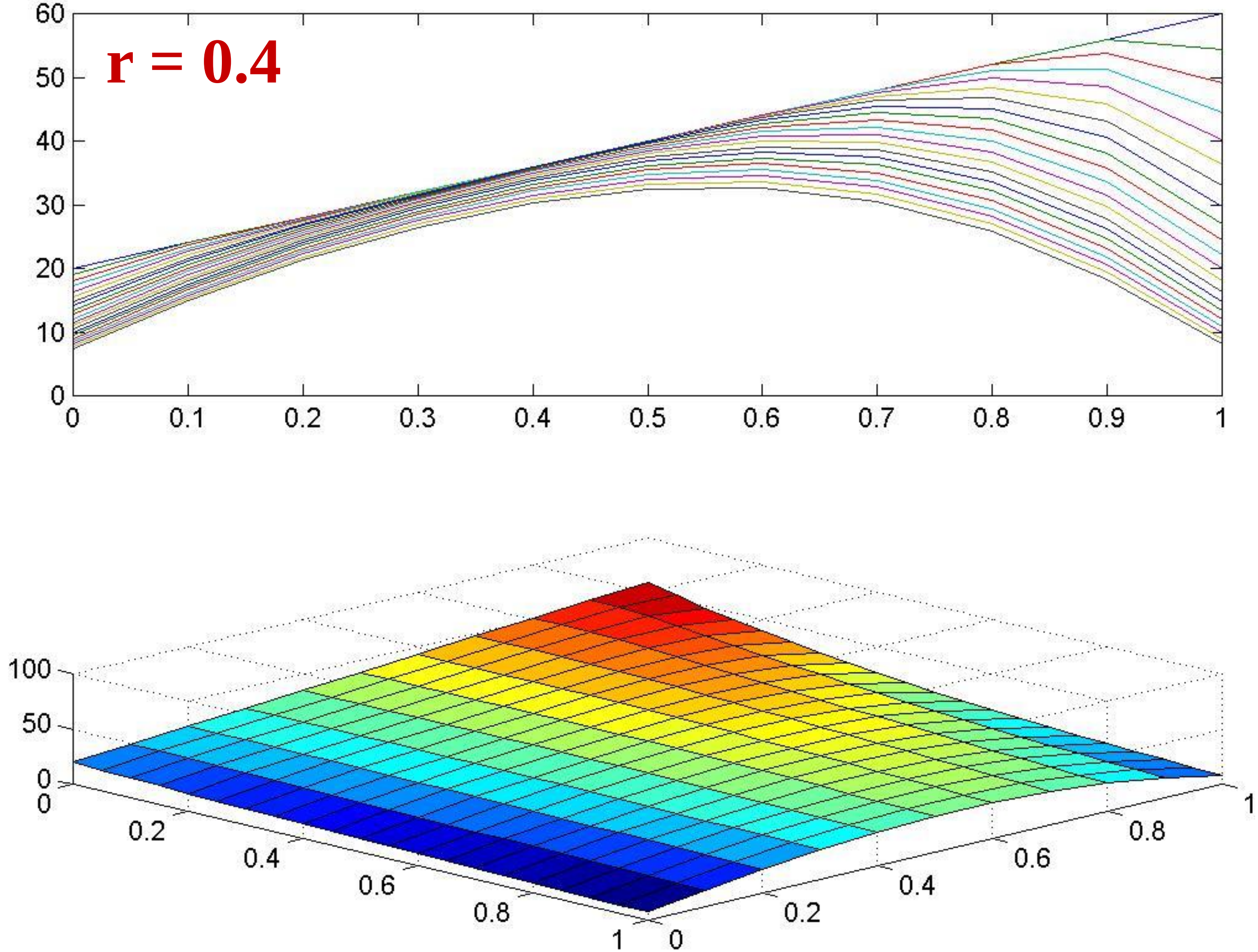
$$\begin{cases} u_{0,1} = 20e^{-0.05} = 19.02458849 \\ u_{1,1} = 0.4u_{0,0} + 0.2u_{1,0} + 0.4u_{2,0} = 0.4(20) + 0.2(30) + 0.4(40) = 30 \\ u_{2,1} = 0.4u_{1,0} + 0.2u_{2,0} + 0.4u_{3,0} = 0.4(30) + 0.2(40) + 0.4(50) = 40 \\ u_{3,1} = 0.4u_{2,0} + 0.2u_{3,0} + 0.4u_{4,0} = 0.4(40) + 0.2(50) + 0.4(60) = 50 \\ u_{4,1} = 60e^{-0.10} = 54.29024508 \end{cases}$$

- Second step t = 0.10

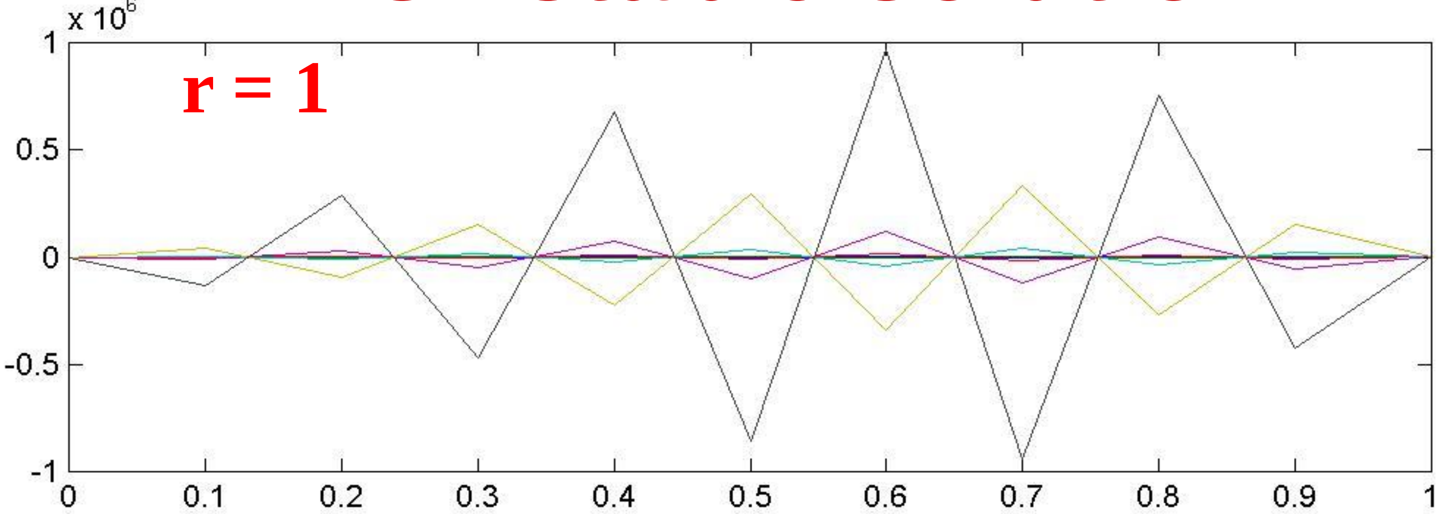
$$
\begin{cases}
u_{0,2} = 20e^{-0.10} = 18.09674836 \\
u_{1,2} = 0.4u_{0,1} + 0.2u_{1,1} + 0.4u_{2,1} \\
\qquad = 0.4(19.02458849) + 0.2(30) + 0.4(40) = 29.6098354 \\
u_{2,2} = 0.4u_{1,1} + 0.2u_{2,1} + 0.4u_{3,1} = 0.4(30) + 0.2(40) + 0.4(50) = 40 \\
u_{3,2} = 0.4u_{2,1} + 0.2u_{3,1} + 0.4u_{4,1} \\
\qquad = 0.4(40) + 0.2(50) + 0.4(54.2924508) = 47.71609803 \\
u_{4,2} = 60e^{-0.20} = 49.12384518
\end{cases}
$$



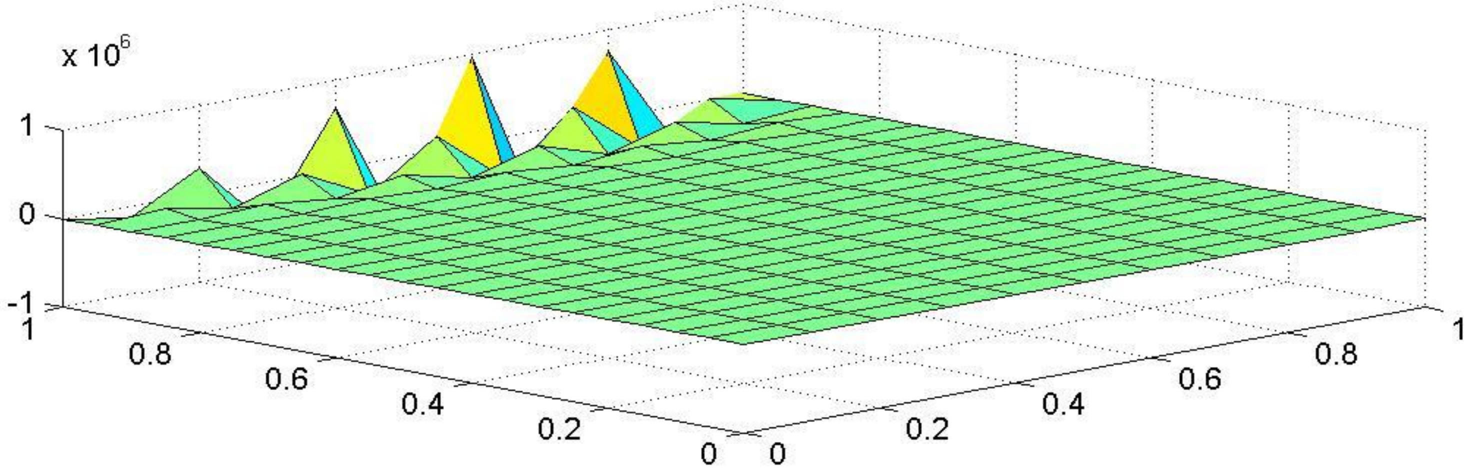
Heat Equation:
BC that depend on time



Example: explicit method, UNstable solution



Unstable !!



Stability Conditions

Conditioned Stability for the explicit method:

$$r\left(= \frac{ck}{h^2}\right) \leq \frac{1}{2} \qquad \text{that is} \qquad \Delta t \leq \frac{1}{2}\frac{\Delta x^2}{c}$$

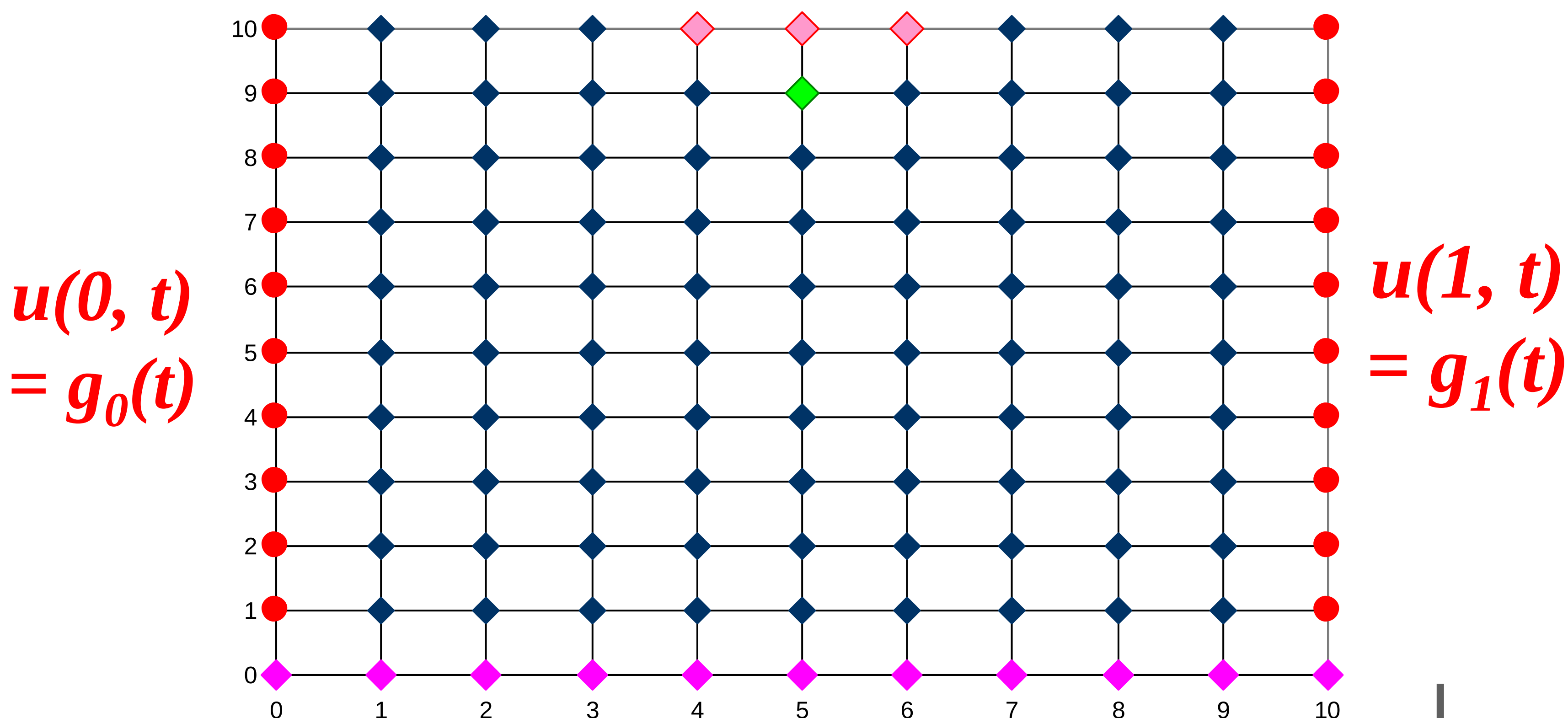
$\Rightarrow$ an implicit method to avoid the instability

Implicit Method

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}; \qquad 0 < x < 1, \quad 0 \le t \le T$$

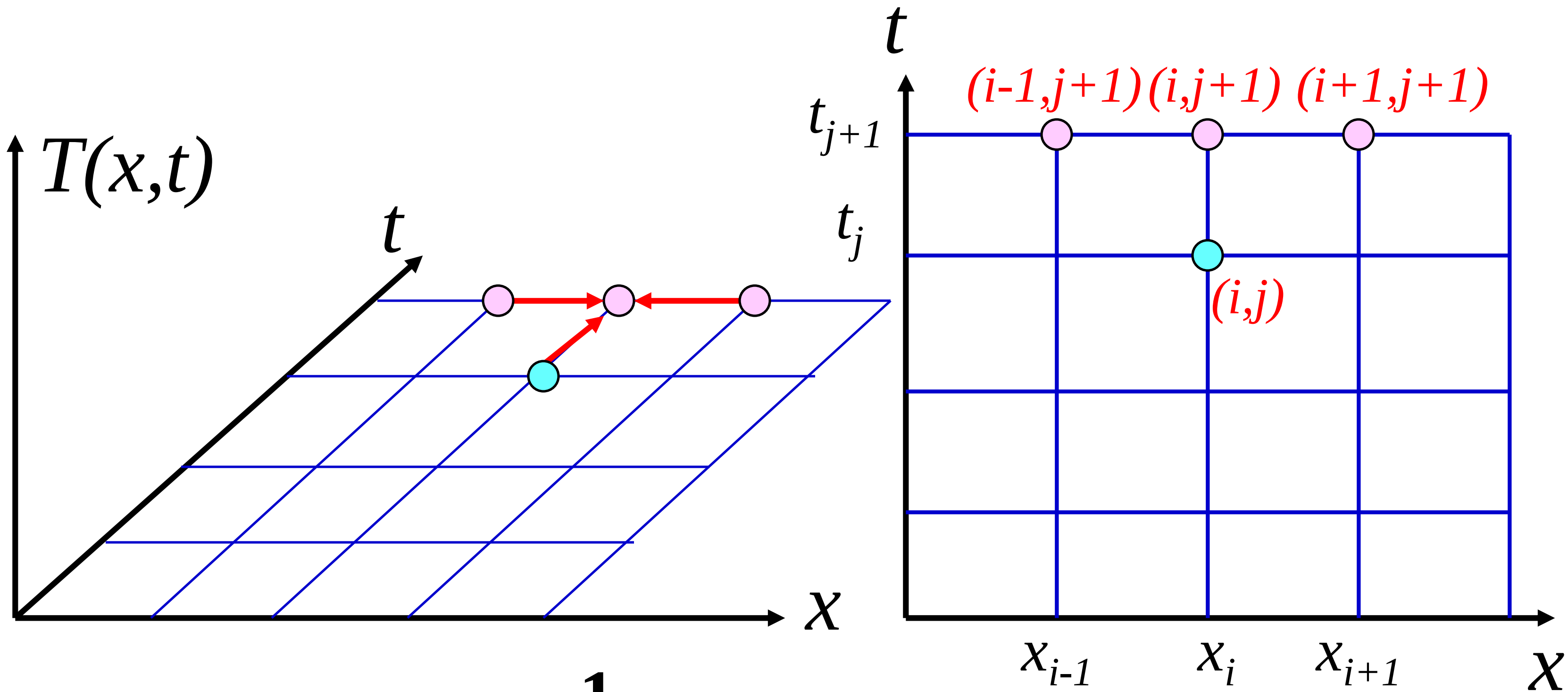
$$\text{CI}: \quad u(x,0) = f(x), \qquad 0 < x < 1$$

$$\text{CB}: \quad \begin{cases} u(0,t) = g_0(t) \\ u(1,t) = g_1(t) \end{cases}, \qquad 0 \le t \le T$$



Initial Conditions : $u(x,0) = f(x)$

Implicit Method



Forward difference

$$u_t = \frac{1}{k}(u_i^{j+1} - u_i^j) \quad +O(k)$$

Centered difference at time j+1

$$u_{xx} = \frac{1}{h^2}(u_{i-1}^{j+1} - 2u_i^{j+1} + u_{i+1}^{j+1}) \quad +O(h^2)$$

Implicit Method

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

$$u_i^{j} = -r u_{i-1}^{j+1} + (1+2r)u_i^{j+1} - r u_{i+1}^{j+1}$$

Linear system to be solved at each time step t

Tridiagonal coefficient matrix (Thomas alg.)

$$\begin{bmatrix} 1+2r & -r & & & \\ -r & 1+2r & -r & & \\ & -r & 1+2r & -r & \\ & & \ddots & \ddots & -r \\ & & & -r & 1+2r \end{bmatrix} \begin{Bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ \vdots \\ u_{N-1,j+1} \end{Bmatrix} = \begin{Bmatrix} u_{1,j} + r u_{0,j+1} \\ u_{2,j} \\ u_{3,j} \\ \\ u_{N-1,j} + r u_{N,j+1} \end{Bmatrix}$$

$$AU_{j+1} = U_j$$

Unconditionally stable

Stability of the Implicit Method

$$(I - r\,L)U_{j+1} = U_j$$

$$U_{j+1} = (I - r\,L)^{-1}U_j$$

$$U_{j+1} = A^{-1}U_j$$

$$U_{j+1} = A^{-2}U_{j-1} = \ldots = A^{-(j+1)}U_0$$

Let us introduce some perturbations E_0 on the data U_0

$$\bar{U}_0 = U_0 + E$$

$$\bar{U}_{j+1} = A^{-(j+1)}\bar{U}_0 = A^{-(j+1)}(U_0 + E) = A^{-(j+1)}U_0 + A^{-(j+1)}E$$

The effect of the error E after j+1 steps is

$$\left\| E^{(j+1)} \right\| \leq \left\| A^{-(j+1)} \right\|_2 \left\| E \right\|$$

Stability of the Implicit Method

$$\left\| E^{(j+1)} \right\| \le \left\| A^{-(j+1)} \right\|_2 \left\| E \right\|$$

$$\left\| A^{-1} \right\|_2 = \rho(\mathrm{A}^{-1}) = \max_{1 \le p \le N} \left| \lambda_p^{-1} \right| = \left(\min_{1 \le p \le N} \left| \lambda_p \right| \right)^{-1}$$

eigenvalues of L: $\lambda_p = 2(\cos(p\pi h) - 1) \quad p = 1,.., N-1, h = 1/N \Rightarrow$

smallest $\lambda_1 = 2(\cos(\pi h) - 1) = 2(-\frac{1}{2}\pi^2 h^2 + \frac{1}{24}\pi^4 h^4 + O(h^6)) \quad \approx -\pi^2 h^2 + O(h^4)$

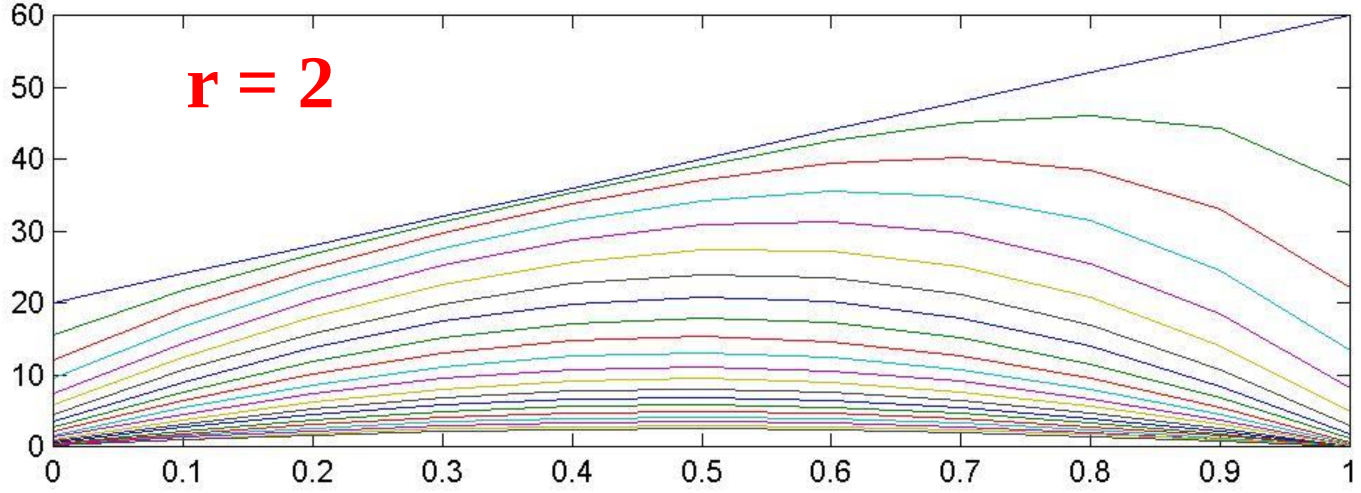
since $A = I - rL$ and $r = \frac{ck}{h^2}$

the method is unconditionally stable

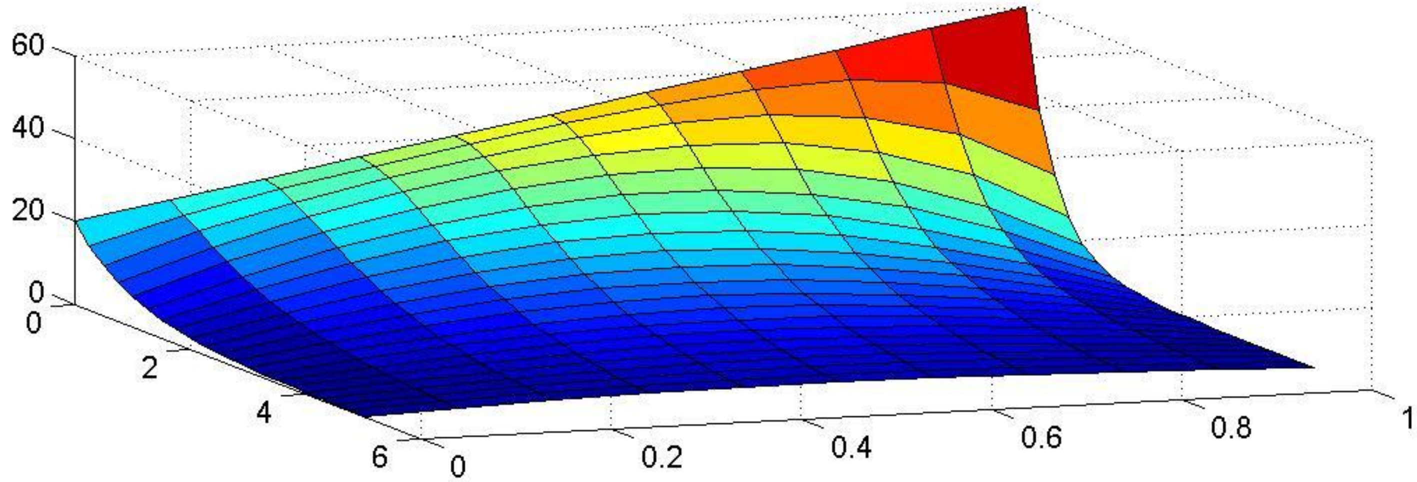
$$\left\| A^{-(j+1)} \right\|_2 \approx \frac{1}{\left| 1 + ck\pi^2 \right|} \le 1$$

Note that ||.||<1 is more restrictive than the Lax-Richtmyer constant C_T

Implicit Method



Unconditionally stable first order accuracy in time



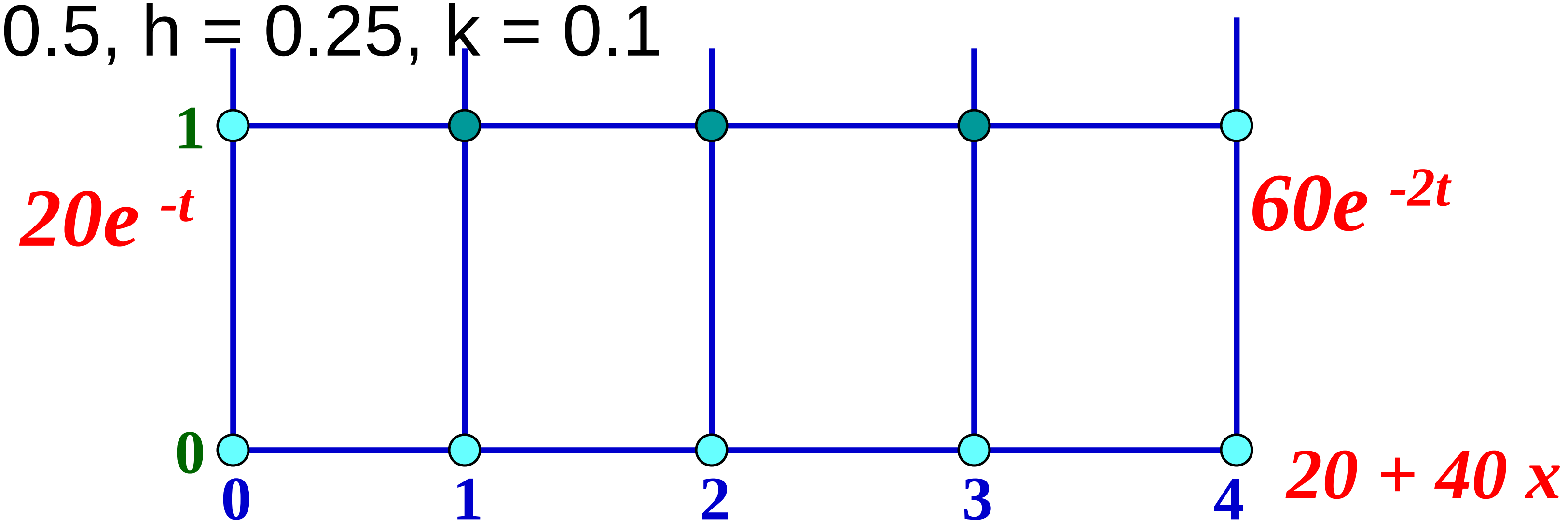
Example: Implicit Method, stable solution

- Heat Equation (Parabolic PDE)

$$u_t = cu_{xx};\ 0 \le x \le 1$$

$$\begin{cases} u(x,0) = 20 + 40x \\ u(0,t) = 20e^{-t},\ u(1,t) = 60e^{-2t} \end{cases}$$

c = 0.5, h = 0.25, k = 0.1



Example: Implicit Method, stable solution

- **Implicit Euler Method**

$$r = \frac{ck}{h^2} = \frac{(0.5)(0.10)}{(0.25)^2} = 0.8$$

$$(-r)u_{i-1,j+1} + (1+2r)u_{i,j+1} + (-r)u_{i+1,j+1} = u_{i,j}$$

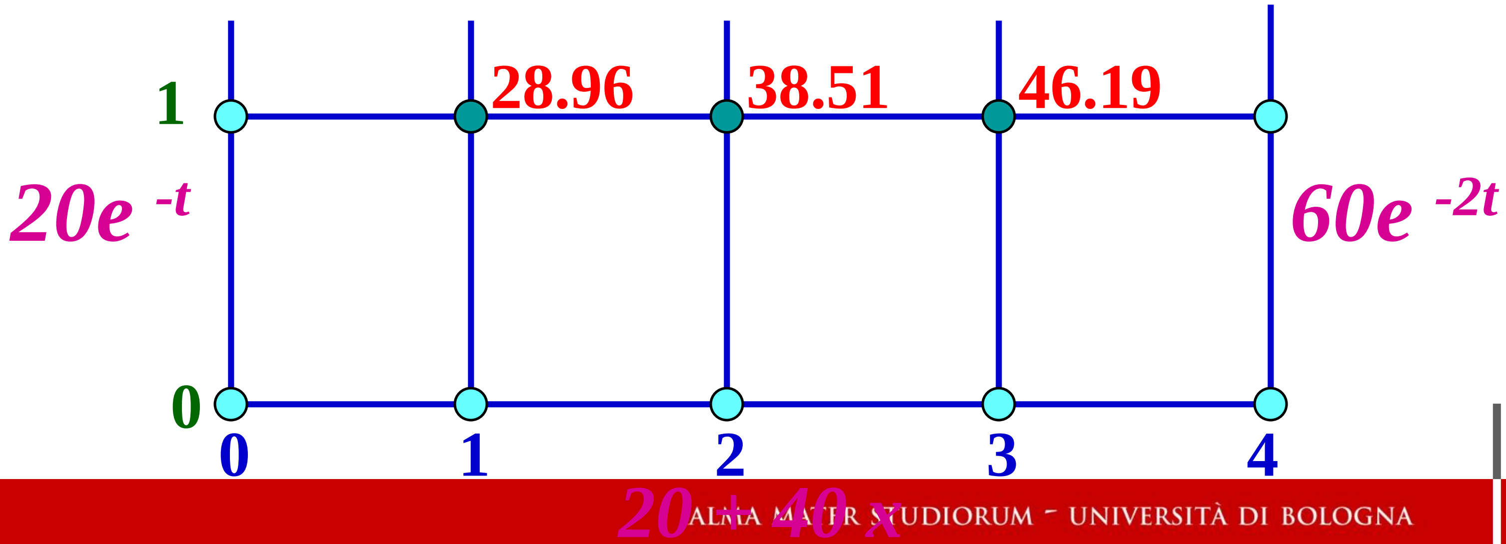
$$(-0.8)u_{i-1,j+1} + (2.6)u_{i,j+1} + (-0.8)u_{i+1,j+1} = u_{i,j}$$

$$\begin{bmatrix} 1+2r & -r & 0 \\ -r & 1+2r & -r \\ 0 & -r & 1+2r \end{bmatrix} \begin{Bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{Bmatrix} = \begin{Bmatrix} u_{1,0} + ru_{0,1} \\ u_{2,0} \\ u_{3,0} + ru_{4,1} \end{Bmatrix}$$

Solve the tridiagonal system

$$\begin{bmatrix} 2.6 & -0.8 & 0 \\ -0.8 & 2.6 & -0.8 \\ 0 & -0.8 & 2.6 \end{bmatrix} \begin{Bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{Bmatrix} = \begin{Bmatrix} 30 + 0.8(20e^{-0.1}) \\ 40 \\ 50 + 0.8(60e^{-0.2}) \end{Bmatrix}$$

$$\Rightarrow \begin{Bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{Bmatrix} = \begin{Bmatrix} 28.95515793 \\ 38.50751457 \\ 46.19426454 \end{Bmatrix}$$



Crank-Nicolson method (implicit)

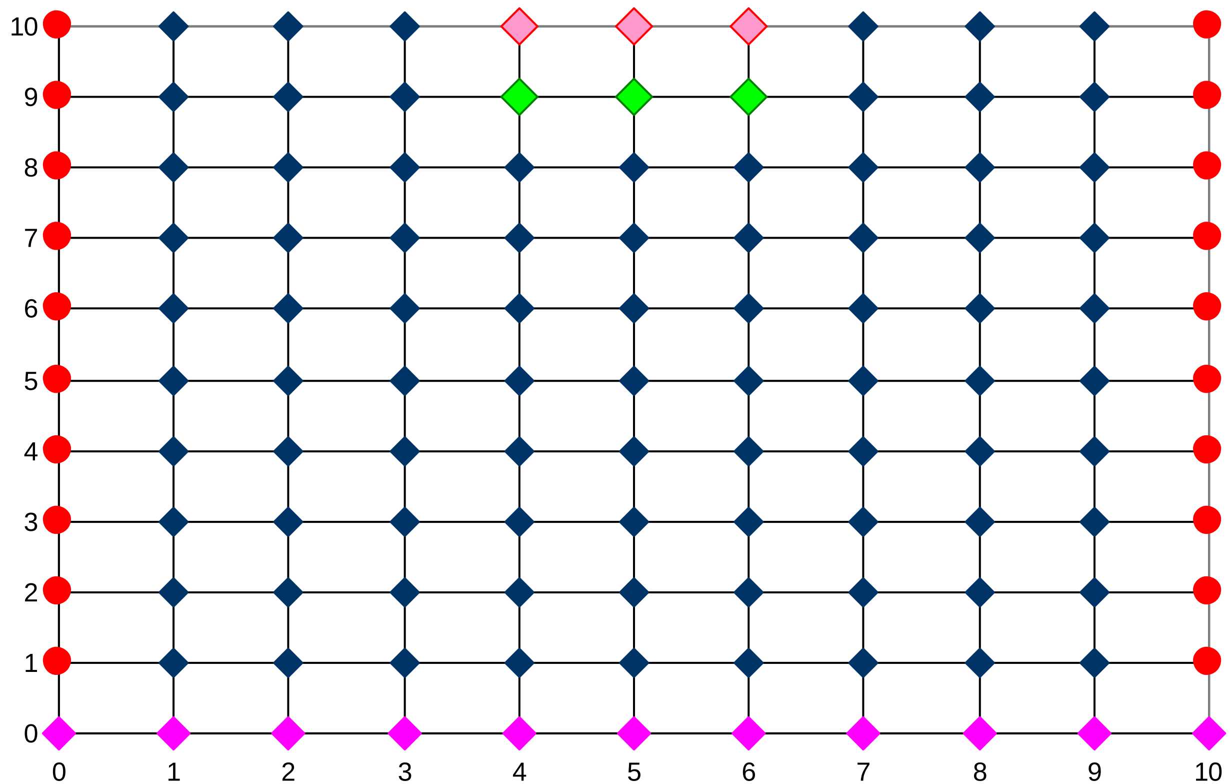
Implicit method: 1° order in time

Crank-Nicolson : 2° order in time

John Crank
1916-2006

Phyllis
Nicolson
1917-1968

u(0, t) = $g_0(t)$



u(1, t) = $g_1(t)$

Initial Conditions: *u(x,0) = f(x)*

Crank-Nicolson method (implicit)

- Crank-Nicolson method for heat equation

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2}; \qquad 0 < \mathrm{x} < 1, \quad 0 \le \mathrm{t} \le \mathrm{T}$$

$$u(x,0) = f(x), \qquad -\infty < \mathrm{x} < \infty$$

- We integrate with respect to time and we use the method of trapezoids to solve the integral in the second member

$$u(x,t+\Delta t) - u(x,t) = a\int_t^{t+\Delta t} u_{xx}(x,s)ds$$

$$u(x,t+\Delta t) - u(x,t) = a\frac{\Delta t}{2}(u_{xx}(x,t) + u_{xx}(x,t+\Delta t))$$

$$-\frac{1}{12}\Delta t^3 u_{xxtt}(x,\theta_t)$$

Crank-Nicolson method (implicit)

We replace the derivatives with finite differences (results in an average of the central differences at two successive time steps j and j +1)

$$\frac{1}{k}(u_{i,j+1} - u_{i,j}) = \frac{a}{2h^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j})$$
$$+ \frac{a}{2h^2}(u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1}) + O(h^2) + O(k^2)$$

Crank-Nicolson method (implicit)

$$\frac{1}{k}(u_{i,j+1}-u_{i,j}) = \frac{a}{2h^2}(u_{i-1,j}-2u_{i,j}+u_{i+1,j})$$

$$+\frac{a}{2h^2}(u_{i-1,j+1}-2u_{i,j+1}+u_{i+1,j+1})$$

We get the following implicit method $r=\frac{ak}{2h^2}$

$$-ru_{i-1,j+1}+(1+2r)u_{i,j+1}-ru_{i+1,j+1}=ru_{i-1,j}+(1-2r)u_{i,j}+ru_{i+1,j}$$

Linear System of N-1 linear eqs with tridiagonal coefficient matrix

$$\mathbf{AU_{j+1}=BU_j}$$

$$IC: \quad u_{i,0} = u_0(x_i) \quad i = 0,..,N$$

$$BC: \quad u(0,t) = g_0(t); \quad u(1,t) = g_1(t)$$

$$\begin{bmatrix} 1+2r & -r & 0 & \\ -r & 1+2r & -r & \\ & .. & .. & -r \\ & & -r & 1+2r \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ .. \\ u_{i,j+1} \\ .. \\ u_{N-2,j+1} \\ u_{N-1,j+1} \end{bmatrix} = \begin{bmatrix} 1-2r & r & 0 & \\ r & 1-2r & r & \\ & .. & .. & r \\ & & r & 1-2r \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ .. \\ u_{i,j} \\ .. \\ u_{N-2,j} \\ u_{N-1,j} \end{bmatrix} + \begin{bmatrix} +r(g_{0,j} + g_{0,j+1}) \\ 0 \\ .. \\ \\ .. \\ 0 \\ +r(g_{1,j} + g_{1,j+1}) \end{bmatrix}$$

$$AU_{j+1} = BU_j + v \quad A, B \in R^{(N-1)x(N-1)}, U \in R^{N-1}$$

Truncation Error $O(k^2 + h^2)$

$$U_{j+1} = A^{-1}BU_j$$

- **Stability**

$$\left\| A^{-1}B \right\| \le C_T \quad \rightarrow \rho(A^{-1}B) \le 1 \quad \textit{eigenvalues of } A^{-1}B:$$

$$A^{-1}B = (I - \frac{r}{2}L)^{-1}(I + \frac{r}{2}L) \qquad r = \frac{k}{h^2}$$

$$\frac{1 + \frac{k}{h^2}(\cos(p\pi h) - 1)}{1 - \frac{k}{h^2}(\cos(p\pi h) - 1)} \le 1 \quad p = 1,.., N-1$$

Unconditionally stable

$$O(k^2 + h^2)$$

- **Consistency**

2nd order accurate in h (space) and k (time)

Θ-Method

$$\frac{1}{k}(u_{i,j+1} - u_{i,j}) = \frac{c\theta}{h^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{c(1-\theta)}{h^2}(u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1})$$

Weighted Average of partial differential equations between time steps j and j+1.

$$\begin{cases} \theta = 0: & \text{implicit method} \\ \theta = 1: & \text{explicit method} \\ \theta = 1/2: & \text{Crank-Nicolson} \end{cases}$$

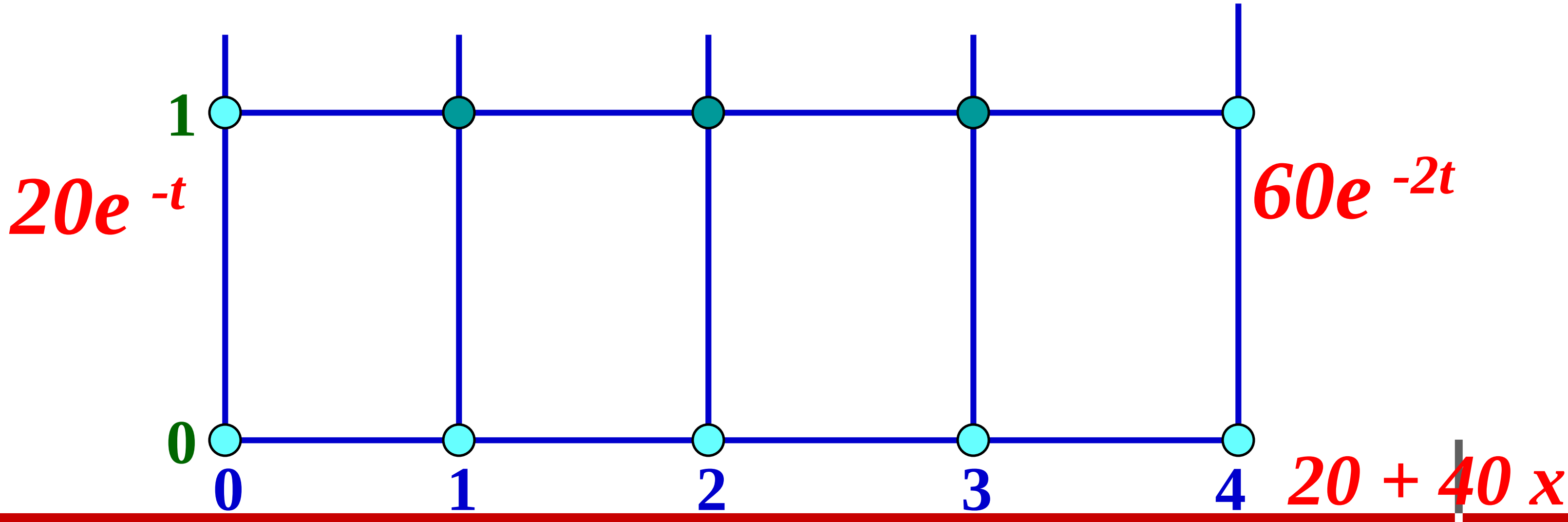
Example: CN method, stable solution

- Heat Equation (Parabolic PDE)

$$u_t = c u_{xx}\,;\ 0 \le x \le 1$$

$$\begin{cases} u(x,0) = 20 + 40x \\ u(0,t) = 20e^{-t},\ u(1,t) = 60e^{-2t} \end{cases}$$

c = 0.5, h = 0.25, k = 0.1



Example: CN method, stable solution

- **Crank-Nicolson**

$$r = \frac{ck}{h^2} = \frac{(0.5)(0.10)}{(0.25)^2} = 0.8$$

$$-\frac{r}{2}u_{i-1,j+1} + (1+r)u_{i,j+1} - \frac{r}{2}u_{i+1,j+1} = \frac{r}{2}u_{i-1,j} + (1-r)u_{i,j} + \frac{r}{2}u_{i+1,j}$$

$$-0.4u_{i-1,j+1} + 1.8u_{i,j+1} - 0.4u_{i+1,j+1} = 0.4u_{i-1,j} + 0.2u_{i,j} + 0.4u_{i+1,j}$$

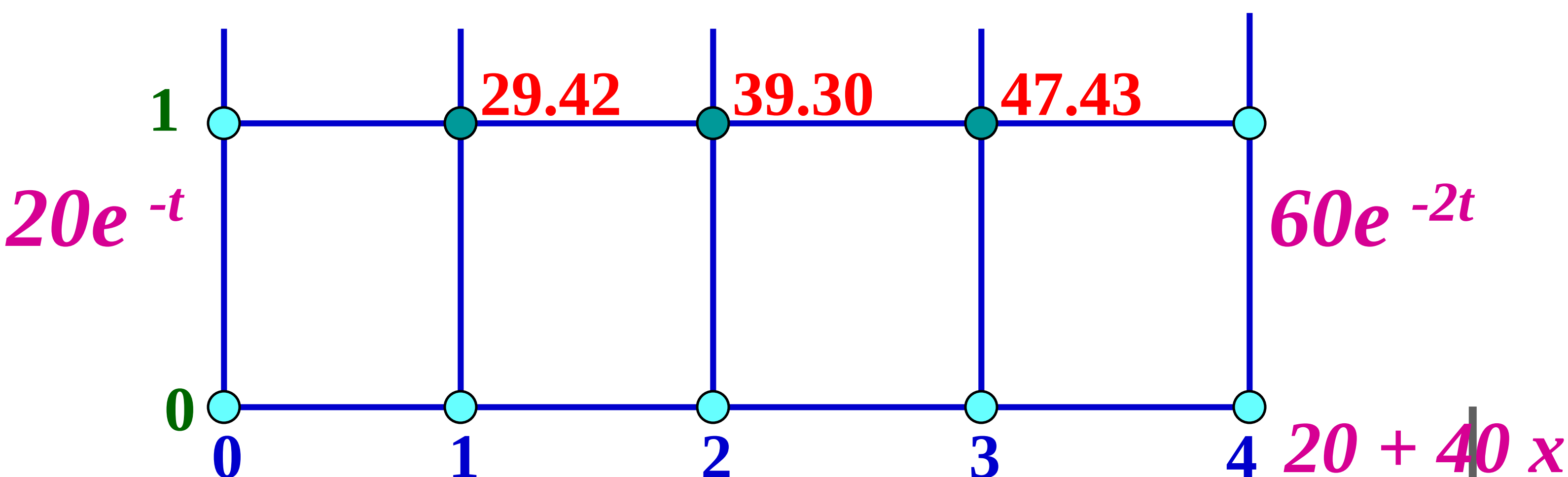
- **Tridiagonal matrix (r = 0.8)**

$$\begin{bmatrix} 1+r & -\frac{r}{2} & 0 \\ -\frac{r}{2} & 1+r & -\frac{r}{2} \\ 0 & -\frac{r}{2} & 1+r \end{bmatrix} \begin{Bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{Bmatrix} = \begin{Bmatrix} \frac{r}{2}u_{0,0} + (1-r)u_{1,0} + \frac{r}{2}u_{2,0} + \frac{r}{2}u_{0,1} \\ \frac{r}{2}u_{1,0} + (1-r)u_{2,0} + \frac{r}{2}u_{3,0} \\ \frac{r}{2}u_{2,0} + (1-r)u_{3,0} + \frac{r}{2}u_{4,0} + \frac{r}{2}u_{4,1} \end{Bmatrix}$$

Example: CN method, stable solution

$$\begin{bmatrix} 1.8 & -0.4 & 0 \\ -0.4 & 1.8 & -0.4 \\ 0 & -0.4 & 1.8 \end{bmatrix} \begin{Bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{Bmatrix} = \begin{Bmatrix} 0.4(20)+0.2(30)+0.4(40)+0.4(20e^{-0.1}) \\ 0.4(30)+0.2(40)+0.4(50) \\ 0.4(40)+0.2(50)+0.4(60)+0.4(60e^{-0.2}) \end{Bmatrix}$$

$$= \begin{Bmatrix} 37.23869934 \\ 40 \\ 69.64953807 \end{Bmatrix} \qquad \Rightarrow \begin{Bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{Bmatrix} = \begin{Bmatrix} 29.42144598 \\ 39.29975855 \\ 47.42746748 \end{Bmatrix}$$



Non-constant Diffusion Coefficient

$$u_t = c u_{xx}$$

c diffusion coefficient c>0, in general, c(x,t)>0 depends on x and t

Explicit

$$u_{i,j+1} = u_{i,j} + \frac{c_{i,j} k}{h^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j})$$

$$k \le \frac{c_{i,j}}{h^2}, \qquad k \le \frac{\bar{c}_j}{h^2} \qquad \bar{c} = \max_i c_{i,j}$$

Implicit

$$u_{i,j+1} = u_{i,j} + \frac{c_{i,j+1} k}{h^2}(u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1})$$

Θ-Method

$$c_{i,j+1/2} = (c_{i,j} + c_{i,j+1}) / 2$$

or

$$\frac{1}{k}(u_{i,j+1} - u_{i,j}) = \frac{c_{i,j}\theta}{h^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{c_{i,j+1}(1-\theta)}{h^2}(u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1})$$

Method of lines (MOL)

-We first discretize the PDE in space alone, which gives a large system of ODEs with each component of the system corrisponding to the solution at some grid point, as a function of time. (**semi-discretization**)

-Then we solve the system using one of the methods for ODEs that we have previously studied:

Explicit Euler, Implicit Euler, ϴ-method,..).

- Finally we fix the boundary conditions.

$$u_t - a u_{xx} = 0 \qquad + IC \ \ +BC \ (\ i.e. periodic)$$

$$Let \quad U(t) = [u_0(t), u_1(t), \dots, u_n(t)]$$

$$\Rightarrow \quad U'(t) = \frac{a}{h^2} LU(t)$$

System of ODEs

Method of lines (MOL)

$$\begin{bmatrix} u_0'(t) \\ u_1'(t) \\ .. \\ u_i'(t) \\ .. \\ u_{N-1}'(t) \\ u_N'(t) \end{bmatrix} = \frac{a}{h^2} \begin{bmatrix} -2 & 1 & 0 & \\ 1 & -2 & 1 & \\ & .. & .. & 1 \\ & & 1 & -2 \end{bmatrix} \begin{bmatrix} u_0(t) \\ u_1(t) \\ .. \\ u_i(t) \\ .. \\ u_{N-1}(t) \\ u_N(t) \end{bmatrix}$$

$$U'(t) = \frac{a}{h^2} LU(t) \quad L \in R^{(N+1)x(N+1)}, U \in R^{N+1}$$

L discretization of the Laplace operator

Method of lines (MOL): apply discretization in time

Explicit Euler Method $U^{n+1} = (I + r\,L)U^n \quad r = \frac{ak}{h^2}$

Implicit Euler Method $(I - r\,L)U^{n+1} = U^n$

Large Linear System of equations

Local Truncation Error $O(k + h^2)$

Crank-Nicolson Method

$$(I - \frac{r}{2}L)U^{n+1} = (I + \frac{r}{2}L)U^n$$

Local Truncation Error $O(k^2 + h^2)$

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Serena Morigi

Dipartimento di Matematica

serena.morigi@unibo.it

http://www.dm.unibo.it/~morigi