# Diffusion Processes

# Diffusion processes

Examples of diffusion processes

- Heat conduction
  - Heat moves from hot to cold places
- Diffusive (molecular) transport of a substance
  - Ink in water
  - Sugar/Cream in coffee
  - Perfume/Gas in air
- Thin-film fluid flow

## Diffusion processes

- Diffusion processes smoothes out differences
- A physical property (heat/concentration) moves from high concentration to low concentration
- Convection is another (and usually more efficient) way of smearing out a property, but is not treated here

# One dimension

- For simplicity, we will in the following focus on one dimensional examples
- This simplifies the complexity of the numerics and codes, but it would still be realistic in examples with
  - Long thin geometries
  - One dimensional variation only
  - Cylindrical or spherical symmetry
  - Mathematical splitting of dimension

$$u(x,y,z,t) = F(x,t) + G(y,z,t)$$

or

$$u(x,y,z,t) = F(x,t)G(y,z,t)$$

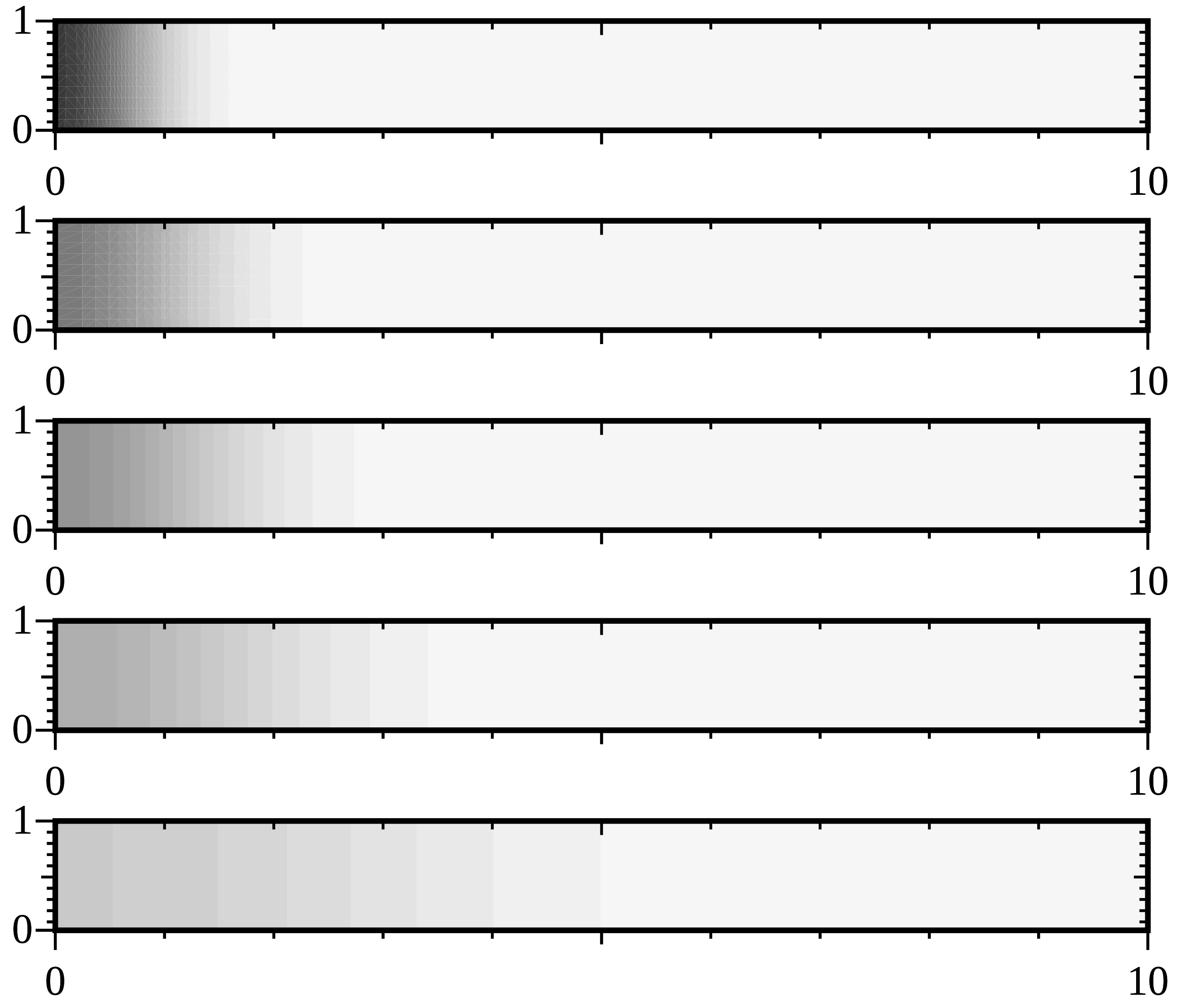

Figure 1: Diffusion of ink in a long and thin tube. The top figure shows the initial concentration (dark is ink, white is water). The three figures below show the concentration of ink at (scaled) times $t = 0.25$, $t = 0.5$, $t = 1$, and $t = 3$, respectively. The evolution is clearly one-dimensional.

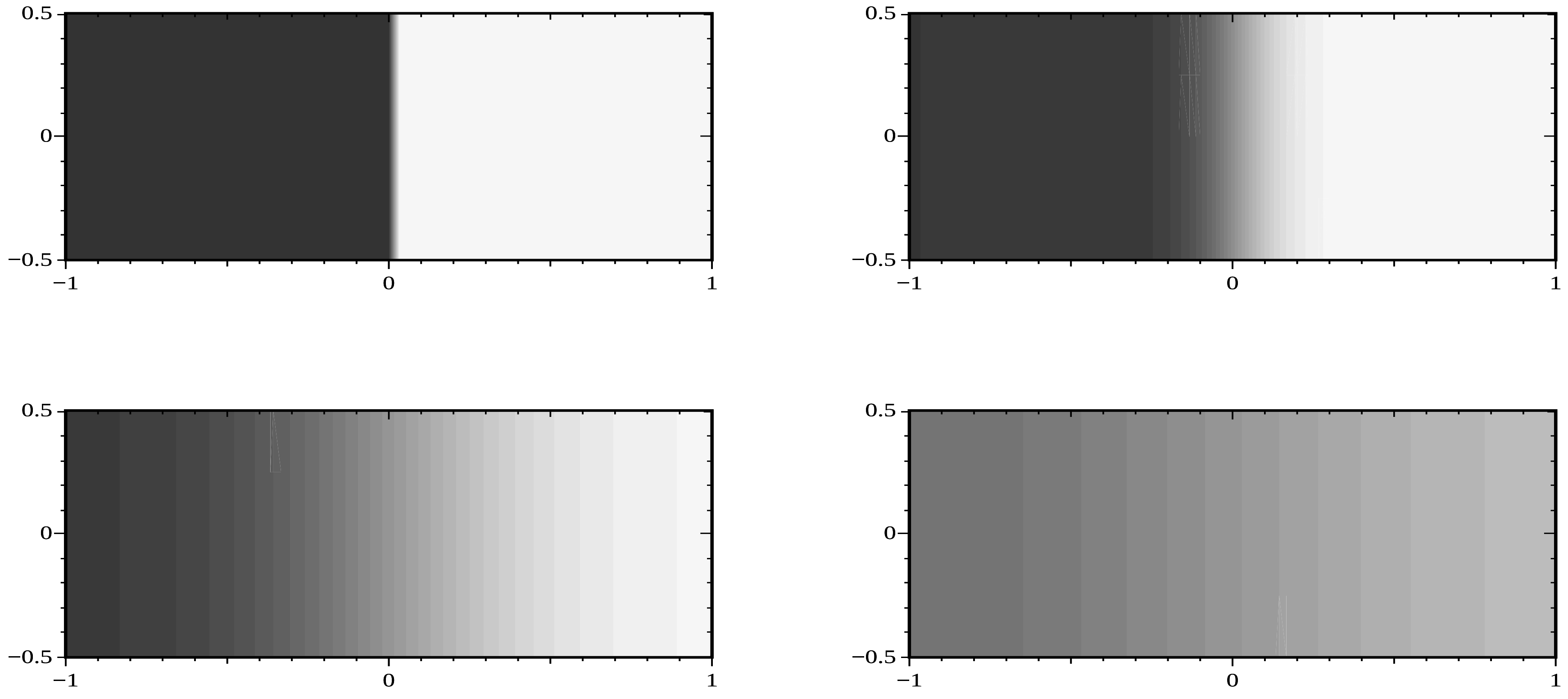

Figure 2: The evolution of the temperature in a medium composed of two pieces of metal, at different initial temperatures. In the gray scale plots, dark is hot and white is cool. The plots correspond to $t = 0$, $t = 0.01$, $t = 0.1$, and $t = 0.5$. All boundaries are insulated, and the temperature approaches a constant value, equal to the average $(T_1 + T_2)/2$ of the initial temperature values.

## The Basics of the Mathematical Model

The diffusion equation reads

$$\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2} + f(x,t), \quad x \in (a,b),\ t > 0 \tag{1}$$

- $k$ is a physical parameter
- Large $k$ implies that $u$ spreads quickly

## Initial and Boundary conditions

- Let $u$ be a solution of $(1)$, then for any constant $C$, $u+C$ will also be a solution $(1)$
- Thus, there are infinitely many solutions of $(1)$
- In order to make a problem with unique solution we need some initial and boundary conditions
- Initial conditions is that we now the solution initially $u(x,0)$ for $x \in [a,b]$
- Boundary conditions is that we have some information about the solution at the endpoints $u(a,t)$ and $u(b,t)$

# Diffusion equation

- In 3 dimensions the diffusion equation reads

$$\frac{\partial u}{\partial t} = k\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + f(x,y,z,t) \qquad (2)$$

- This equation is sometimes written on a more compact form

$$\frac{\partial u}{\partial t} = k\nabla^2 u + f, \qquad (3)$$

  where the operator $\nabla^2$ is defined by $\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$

- $\nabla^2$ is called the Laplace operator

# Derivation of Diffusion equations

- We shall derive the diffusion equation for diffusion of a substance
- Think of some ink placed in a long, thin tube filled with water
- We study the concentration $c(x,t)$, $x \in (a,b)$, $t > 0$
- The motion of the substance will be determined by two physical laws:
  - Conservation of mass
  - Fick's law relating the velocity of the substance (flux) to the concentration

# Mass conservation

Let $c(x,t)$ denote the concentration of the ink, $q(x,t)$ denotes the velocity of it and ρ denotes mass density of pure ink

- For a system without any source, the net inflow on the interval equals the increase in mass

$$\rho q(a)\Delta t - \rho q(b)\Delta t = \int_a^b \rho \Delta c \, dx \qquad (4)$$

- Introducing a source term $f$, the mass balance is

$$\rho q(a)\Delta t - \rho q(b)\Delta t + \int_a^b \rho f \Delta t \, dx = \int_a^b \rho \Delta c \, dx,$$

where $f > 0$ corresponds to mass injection and $f < 0$ means mass extraction

# Mass conservation

- For small values of $\Delta t$ we have ($\Delta c = c(x,t+\Delta t) - c(x,t)$)

$$\Delta c = \frac{\partial c}{\partial t}\Delta t \tag{5}$$

- To study the left hand side of (4), we note that integration by parts give

$$\int_a^b \rho \frac{\partial q}{\partial x} dx = -\int_a^b q \frac{\partial \rho}{\partial x} dx + \rho [q]_a^b$$

- We assume that the mass density is constant, i.e. $\frac{\partial \rho}{\partial x} = 0$, thus

$$\rho(q(b,t) - q(a,t)) = \int_a^b \rho \frac{\partial q}{\partial x} dx \tag{6}$$

# Mass conservation

- Collecting the integrals, we can write the mass conservation principle on the form

$$\int_a^b \rho \left[\frac{\partial c}{\partial t} + \frac{\partial q}{\partial x} - f\right] dx = 0$$

- Since this integral is zero for any interval $[a,b]$, one can argue that the integrand must be zero for all values of $x$ and $t$, thus

$$\frac{\partial c}{\partial t} + \frac{\partial q}{\partial x} = f, \tag{7}$$

which is referred to as the law of mass conservation on partial differential equation form

# Mass conservation

- Let $c(x,t)$ denote the concentration of the ink, let $q(x,t)$ denote the velocity of it (from left to right) and let $f(x,t)$ denote the mass injection of ink
- The law of Conservation of mass, in PDE form, reads

$$\frac{\partial c}{\partial t} + \frac{\partial q}{\partial x} = f \tag{8}$$

- This equation states that temporal change in concentration plus the spatial change in velocity equals the injection of ink
- This means that ink can neither appear nor disappear (mass conservation)

# Fick’s law

- Fick’s law reads

$$q = -k\frac{\partial c}{\partial x} \tag{9}$$

- This law states that the flow of ink is proportional to the spatial change in concentration
- The minus sign makes sure that the ink diffuses from regions with high concentration to regions with low concentration

## Diffusion of a substance

- By inserting Fick's law (9) in the mass conservation equation (8), we can eliminate $q$ and get a PDE with only one unknown function, $c$:

$$\frac{\partial c}{\partial t} = k\frac{\partial^2 c}{\partial x^2} + f(x,t) \tag{10}$$

# Initial conditions

In order to solve the diffusion equation we need some initial condition and boundary conditions.

- The initial condition gives the concentration in the tube at t=0

$$c(x,0) = I(x), \quad x \in (0,1) \tag{11}$$

- Physically this means that we need to know the concentration distribution in the tube at a moment to be able to predict the future distribution

## Boundary conditions

Some common boundary conditions are

- Prescribed concentrations, $S_0$ and $S_1$, at the endpoints

$$c(0,t) = S_0 \quad \text{and} \quad c(1,t) = S_1$$

- Impermeable endpoints, i.e. no out flow at the endpoints

$$q(0,t) = 0 \quad \text{and} \quad q(1,t) = 0$$

- By Fick's law we get

$$\frac{\partial c(0,t)}{\partial x} = 0 \quad \text{and} \quad \frac{\partial c(1,t)}{\partial x} = 0$$

# Boundary conditions

- Prescribed outflows $Q_0$ and $Q_1$ at the endpoints

$$-q(0,t) = Q_0 \quad \text{and} \quad q(1,t) = Q_1$$

  - Here the minus sign in the first expression, $-q(0,t) = Q_0$, comes since $Q_0$ measures the flow out of the tube, and that is the negative direction (from right to left)
  - By Fick's law we get

$$k\frac{\partial c(0,t)}{\partial x} = Q_0 \quad \text{and} \quad -k\frac{\partial c(1,t)}{\partial x} = Q_1$$

# Derivation of the heat equation

- We shall derive the diffusion equation for heat conduction
- We consider a rod of length 1 and study how the temperature distribution $T(x,t)$ develop in time, i.e. we study $T(x,t)$ for $x \in (0,1)$ and $t \geq 0$
- Our derivation of the heat equation is based on
  - The first law of Thermodynamics (conservation of energy)
  - A relation between inner energy and temperature
  - Fourier's law of heat conduction

# Derivation of the heat equation

Let $e(x,t)$ denote the internal energy per unit mass, let ρ be the mass density, and let $q(x,t)$ be the flow of heat (from left to right - defined per unit time).

- The first law of Thermodynamics on PDE form reads

$$\rho\frac{\partial e}{\partial t}+\frac{\partial q}{\partial x}=f, \tag{12}$$

  where $f$ denotes the energy production

- This equation states that the temporal change in energy times the mass density plus the energy flow in a point equals the production of energy in the same point (conservation of energy)

# Derivation of the heat equation

A relation between internal energy $e$ and temperature $T$ is given by

$$e = c_v T. \tag{13}$$

In practice this relation might be more complicated
Thus

- The inner energy is proportional to the temperature
- The proportionality constant, $c_v$, is heat capacity

## Derivation of the heat equation

- Fourier's law reads

$$q = -k\frac{\partial T}{\partial x} \tag{14}$$

- In words: the heat flow is proportional to the spatial change in temperature
- $k$ is called the conductivity
- The minus sign means that the heat flows from hot to cold regions

## The heat equation

- We will now allow the physical parameters $\rho$, $c_v$ and $k$ to vary in space, i.e.

$$\rho = \rho(x), \quad c_v = c_v(x) \quad \text{and} \quad k = k(x)$$

- Inserting (14) and (13) in (12) gives us the heat conduction equation

$$\rho(x)c_v(x)\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k(x)\frac{\partial T}{\partial x}\right) + f \qquad (15)$$

# Initial conditions

In order to solve the heat equation we need some initial- and boundary conditions.

- The initial condition gives the temperature distribution in the rod at t=0

$$T(x,0) = I(x), \quad x \in (0,1) \tag{16}$$

- Physically this means that we need to know the temperature in the rod at a moment to be able to predict the future temperature distribution

## Boundary conditions

There are three types of linear boundary conditions:

- Dirichlet conditions:
  - The temperatures at the endpoints of the rod, $T(0,t)$ and $T(1,t)$, are prescribed at all time
  - Physically, this corresponds to a situation where you have a heat source which keep the temperature at given values at the endpoints
- Neumann condition:
  - The heat flow at the endpoints, $k\frac{\partial T(0,t)}{\partial x}$ and $-k\frac{\partial T(1,t)}{\partial x}$, is prescribed at all time (The difference plus sign in front of $k\frac{\partial T(0,t)}{\partial x}$ comes from the fact that we consider inflow)
  - The case $\frac{\partial T(0,t)}{\partial x} = \frac{\partial T(1,t)}{\partial x} = 0$ corresponds to insulated endpoints

# Boundary conditions

- Robin conditions:
  - Most common example of a Robin condition is Newton's law of cooling

$$k\frac{\partial T(0,t)}{\partial x} = h_T(T(0,t) - T_s) \quad \text{and} \quad -k\frac{\partial T(1,t)}{\partial x} = h_T(T(1,t) - T_s)$$

  - This law states that the heat flow at the endpoint is proportional to the difference between the temperature in the rod, $T(0,t)$ and $T(1,t)$, and the temperature in the surroundings, $T_s$
  - The constant $h_T$ is called the heat transfer coefficient and has to be determined for a given experiment

# Scaling

Suppose we work with the following diffusion equation:

$$\rho c_v \frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2}, \quad x \in (a,b),\ t > 0, \tag{17}$$

$$u(a,t) = U_a, \quad t > 0, \tag{18}$$

$$u(b,t) = U_b, \quad t > 0, \tag{19}$$

$$u(x,0) = I(x), \quad x \in [a,b] \tag{20}$$

with

$$I(x) = \begin{cases} U_a, & a \le x < c, \\ U_b, & c \le x \le b \end{cases}$$

- It is clear that the solution $u(x,t)$ will depend on all the input parameters $\rho$, $c_v$, $k$, $U_a$, $U_b$, $a$ and $b$

$$u(x,t;\rho,c_v,k,U_a,U_b,a,b)$$

## Scaling

- If we want to test how the solution depend on the seven parameters, it might be a very time consuming job
- Testing 3 values for each parameter would require $3^7 = 2187$ experiments, or 5 values for each parameter would require $5^7 = 78125$ experiments
- If the problem is scaled, we shall see that it is sufficient to perform just a single experiment

## Scaling

- The purpose of scaling a variable $q$, is to introduce a new variable $\bar{q}$, such that $\bar{q}$ varies between zero and about one
- If $q_r$ is a characteristic reference value of $q$ and $q_c$ is a characteristic magnitude of $q - q_r$, a common scaling is

$$\bar{q} = \frac{q - q_r}{q_c}$$

## Scaling

- We shall now see how the general interval $(a,b)$ can be scaled to the standard unity interval $(0,1)$
- A scaled parameter for $x$ can be

$$\bar{x} = \frac{x-a}{b-a},$$

which fulfills $\bar{x} \in (0,1)$ while $x \in (a,b)$

## Scaling

- Further, a scaled parameter for time can be

$$\bar{t} = \frac{t}{t_c},$$

where $t_c$ is the time it takes to make significant changes in $u$

- A scaling of the initial condition might be

$$\bar{I} = \frac{I - U_a}{U_b - U_a}$$

- Finally, a scaling of $u$ can be

$$\bar{u} = \frac{u - U_a}{U_b - U_a}$$

# Scaling

- We can now replace the physical variables $x$, $t$, $u$, and $I$, with

$$\bar{x} = \frac{x-a}{b-a}, \quad \bar{t} = \frac{t}{t_c}, \quad \bar{I} = \frac{I-U_a}{U_b-U_a}, \quad \bar{u} = \frac{u-U_a}{U_b-U_a}$$

  which will be inserted to (17)–(20)

- Solving the above formulas for $x$, $t$, $u$, and $I$ gives

$$x = a+(b-a)\bar{x}, \quad t = t_c\bar{t}, \quad I = U_a+(U_b-U_a)\bar{I}, \quad u = U_a+(U_b-U_a)\bar{u}$$

- Note that

$$\frac{\partial u}{\partial t} = \frac{\partial \bar{t}}{\partial t}\frac{\partial}{\partial \bar{t}}(U_a+(U_b-U_a)\bar{u}) = \frac{1}{t_c}(U_b-U_a)\frac{\partial \bar{u}}{\partial \bar{t}}$$

## Scaling

- A similar development for the $\partial^2 u/\partial x^2$ expression, gives

$$\rho c_v \frac{U_b - U_a}{t_c}\frac{\partial \bar{u}}{\partial \bar{t}} = k\frac{U_b - U_a}{(b-a)^2}\frac{\partial^2 \bar{u}}{\partial \bar{x}^2}, \quad \bar{x} \in (0,1), \bar{t} > 0, \tag{21}$$

$$\bar{u}(0,\bar{t}) = 0, \quad \bar{t} > 0, \tag{22}$$

$$\bar{u}(1,\bar{t}) = 1, \quad \bar{t} > 0, \tag{23}$$

$$\bar{u}(\bar{x},0) = \begin{cases} 0, & 0 \leq x \leq \bar{c}, \\ 1, & \bar{c} < x \leq 1 \end{cases} \tag{24}$$

## Scaling

- Note that the PDE (21) can be written

$$\frac{\partial \bar{u}}{\partial \bar{t}} = \alpha \frac{\partial^2 \bar{u}}{\partial \bar{x}^2} \tag{25}$$

- Here $\alpha$ is a dimensionless number,

$$\alpha = \frac{k t_c}{\rho c_v (b-a)^2}$$

- Choosing $t_c = \frac{1}{k}\rho c_v (b-a)^2$ (corresponding to $\alpha = 1$) gives

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

## Scaling

- We can now summarize the result of the scaled diffusion problem:

$$\frac{\partial \bar{u}}{\partial \bar{t}} = \frac{\partial^2 \bar{u}}{\partial \bar{x}^2}, \quad \bar{x} \in (0,1),\ \bar{t} > 0, \tag{26}$$

$$\bar{u}(0,\bar{t}) = 0, \quad \bar{t} > 0, \tag{27}$$

$$\bar{u}(1,\bar{t}) = 1, \quad \bar{t} > 0, \tag{28}$$

$$\bar{u}(\bar{x},0) = \begin{cases} 0, & 0 \le \bar{x} \le \bar{c}, \\ 1, & \bar{c} < \bar{x} \le 1 \end{cases} \tag{29}$$

- After solving this PDE, the real temperatures can be found by

$$u(x,t) = U_a + (U_b - U_a)\bar{u}\left(\frac{x-a}{b-a}, \frac{tk}{\rho c_v (b-a)^2}\right) \tag{30}$$

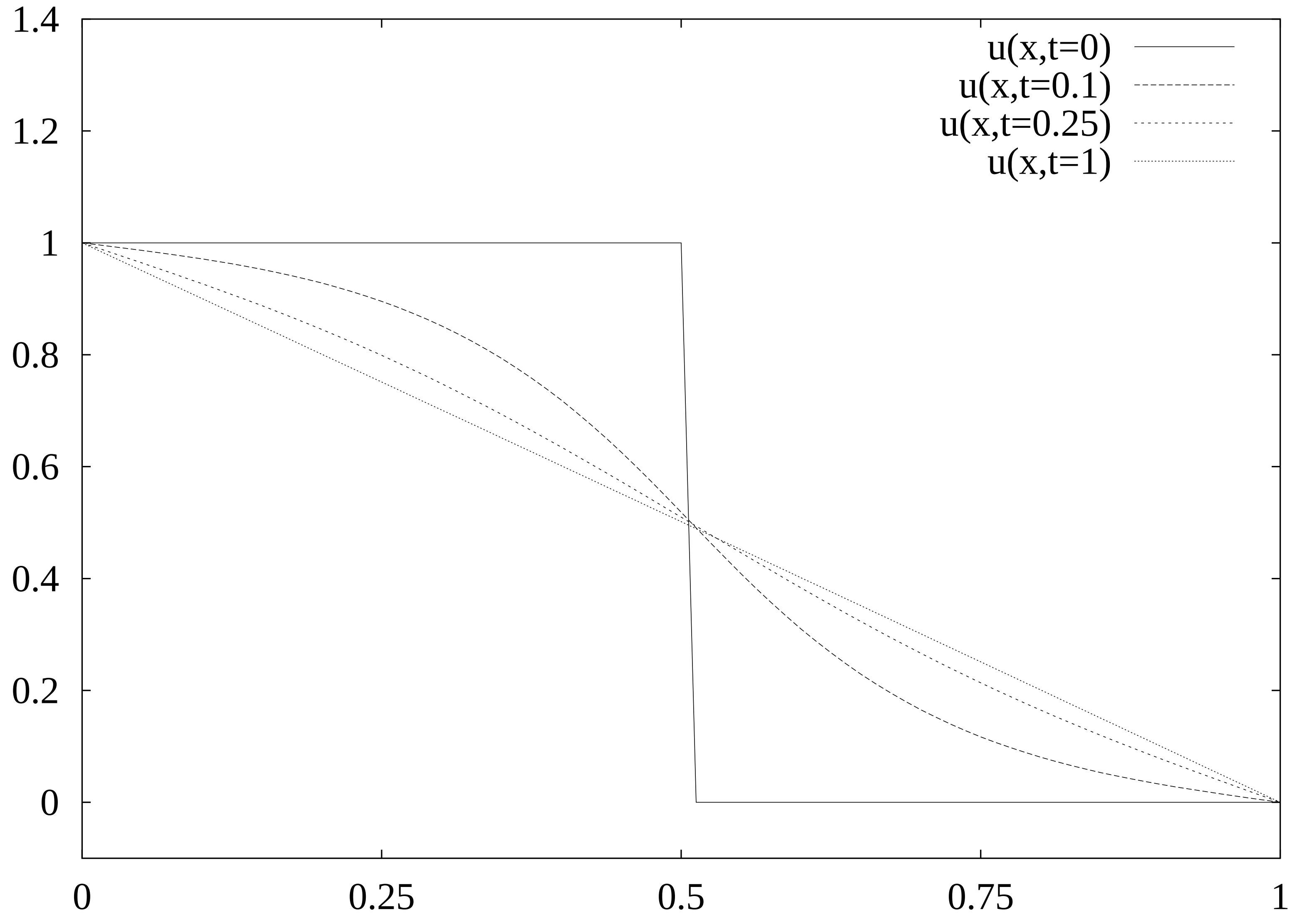

Figure 3: Solution of (26)–(29).

# Numerical methods

First we consider a version of the heat equation where any varying parameters are scaled away:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x,t), \quad x \in (0,1), \; t > 0. \tag{31}$$

- The solution of this equation is a continuous function of time and space
- We approximate the solution at a finite number of space points and at a finite number of time levels
- This approximation is referred to as discretization
- There are several ways of discretizing (31) - in the following we will consider a technique which is called the finite difference method

# Numerical methods

Applying the finite difference method to the problem $(31)$ implies

1. constructing a *grid*, with a finite number of points in $(x,t)$ space, see Figure 4
2. requiring the PDE $(31)$ to be satisfied at each point in the grid
3. replacing derivatives by finite difference approximations
4. calculating $u$ at the grid points only

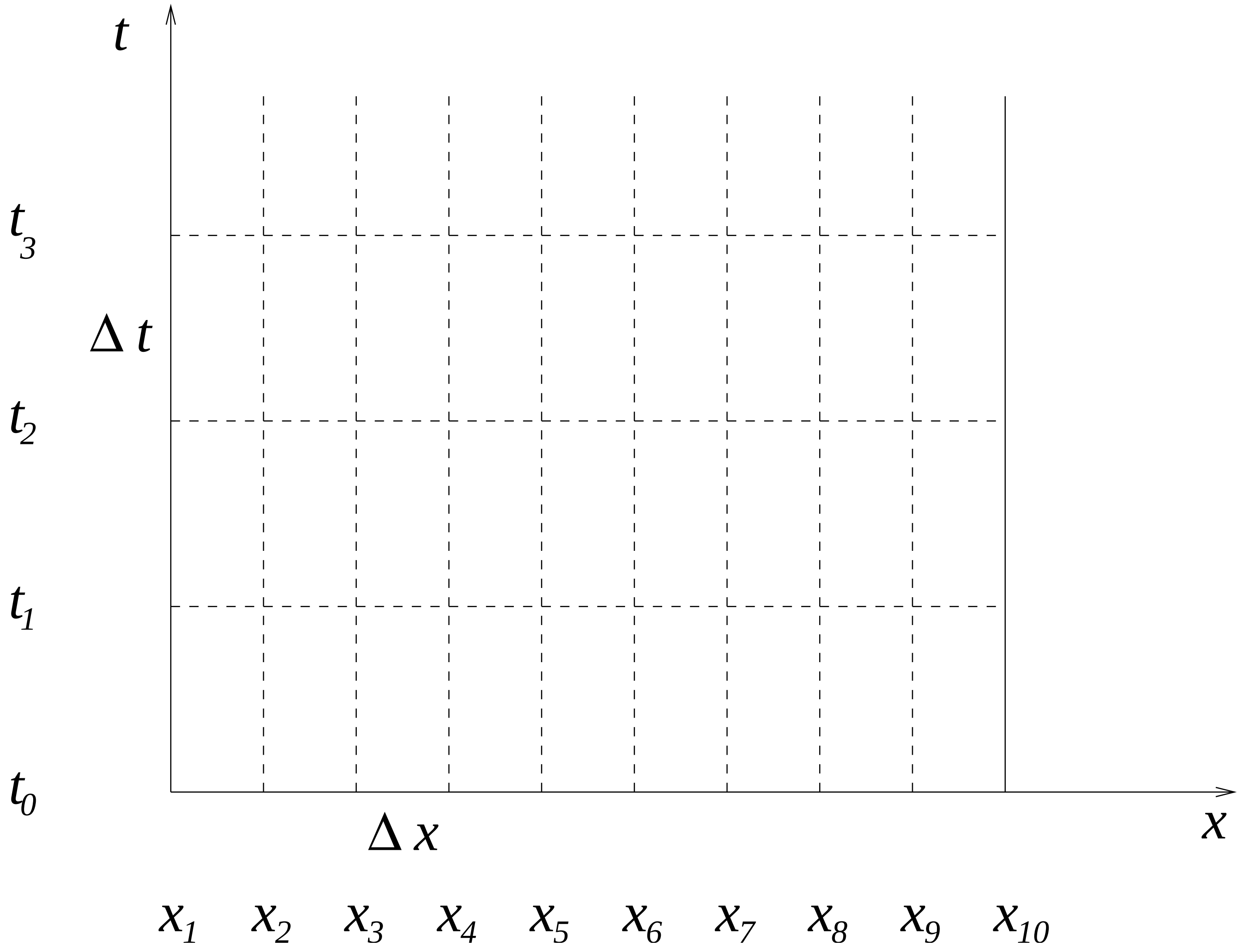
$t$
$t_3$
$\Delta t$
$t_2$
$t_1$
$t_0$
$\Delta x$
$x$
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$ $x_8$ $x_9$ $x_{10}$

Figure 4: Computational grid in the $x,t$-plane. The grid points are located at the points of intersection of the dashed lines.

# Discrete functions on a grid

- Chose a spatial discretization size $\Delta x$ and a temporal discretization size $\Delta t$
- Functions are only defined in the grid points

$$(x_i, t_\ell),$$

  for $i = 1, \ldots, n$ and $\ell = 0, \ldots, m$ where
    - $n$ is the number of approximation points in space $(\Delta x = \frac{1}{n-1})$
    - $m+1$ is the number of time levels
- The value of an arbitrary function $Q(x,t)$ at a grid point $(x_i, t_\ell)$ is denoted

$$Q_i^\ell = Q(x_i, t_\ell), \quad i = 1, \ldots, n, \; \ell = 0, \ldots, m$$

# Discrete functions on a grid

- The purpose of a finite difference method is to compute the values $u_i^\ell$ for $i = 1, \ldots, n$ and $\ell = 0, \ldots, m$
- We can now write the PDE (31) as

$$\frac{\partial}{\partial t} u(x_i, t_\ell) = \frac{\partial^2}{\partial x^2} u(x_i, t_\ell) + f(x_i, t_\ell), \qquad (32)$$
$$i = 1, \ldots, n, \; \ell = 1, \ldots, m$$

# Finite difference approximation

Now we approximate the terms in (32) that contains derivatives. The approximation is done as follows

- The right hand side is approximated

$$\frac{\partial}{\partial t}u(x_i,t_\ell) \approx \frac{u_i^{\ell+1}-u_i^\ell}{\Delta t} \tag{33}$$

- The first term on left hand side is approximated

$$\frac{\partial^2}{\partial x^2}u(x_i,t_\ell) \approx \frac{u_{i-1}^\ell - 2u_i^\ell + u_{i+1}^\ell}{\Delta x^2} \tag{34}$$

- The first approximation (33) can be motivated directly from the definition of derivatives, since $\Delta t$ is small, and it is called a finite difference approximation

## Finite difference approximation

The motivation for $(34)$ is done in two steps and the finite difference approximation is based on centered difference approximations.

- We first approximate the“outer” derivative at $x = x_i$ (and $t = t_\ell$), using a fictitious point $x_{i+\frac{1}{2}} = x_i + \frac{1}{2}\Delta x$ to the right and a fictitious point $x_{i-\frac{1}{2}} = x_i - \frac{1}{2}\Delta x$ to the left

$$\frac{\partial}{\partial x}\left[\left(\frac{\partial u}{\partial x}\right)\right]_i^\ell \approx \frac{1}{\Delta x}\left[\left[\frac{\partial u}{\partial x}\right]_{i+\frac{1}{2}}^\ell - \left[\frac{\partial u}{\partial x}\right]_{i-\frac{1}{2}}^\ell\right]$$

## Finite difference approximation

- The first-order derivative at $x_{i+\frac{1}{2}}$ can be approximated by a centered difference using the point $x_{i+1}$ to the right and the point $x_i$ to the left:

$$\left[\frac{\partial u}{\partial x}\right]^{\ell}_{i+\frac{1}{2}} \approx \frac{u^{\ell}_{i+1} - u^{\ell}_{i}}{\Delta x}$$

- Similarly, the first-order derivative at $x_{i-\frac{1}{2}}$ can be approximated by a centered difference using the point $x_i$ to the right and the point $x_{i-1}$ to the left

$$\left[\frac{\partial u}{\partial x}\right]^{\ell}_{i-\frac{1}{2}} \approx \frac{u^{\ell}_{i} - u^{\ell}_{i-1}}{\Delta x}$$

- Combining these finite differences gives (34)

# The Finite Difference Scheme

- Inserting the difference approximations (33) and (34) in (32) results in the following finite difference scheme

$$\frac{u_i^{\ell+1} - u_i^{\ell}}{\Delta t} = \frac{u_{i-1}^{\ell} - 2u_i^{\ell} + u_{i+1}^{\ell}}{\Delta x^2} + f_i^{\ell} \qquad (35)$$

- We solve (35) with respect to $u_i^{\ell+1}$, yielding a simple formula for the solution at the new time level

$$u_i^{\ell+1} = u_i^{\ell} + \frac{\Delta t}{\Delta x^2}\left(u_{i-1}^{\ell} - 2u_i^{\ell} + u_{i+1}^{\ell}\right) + \Delta t\, f_i^{\ell} \qquad (36)$$

- This is referred to as a numerical scheme for the diffusion equation

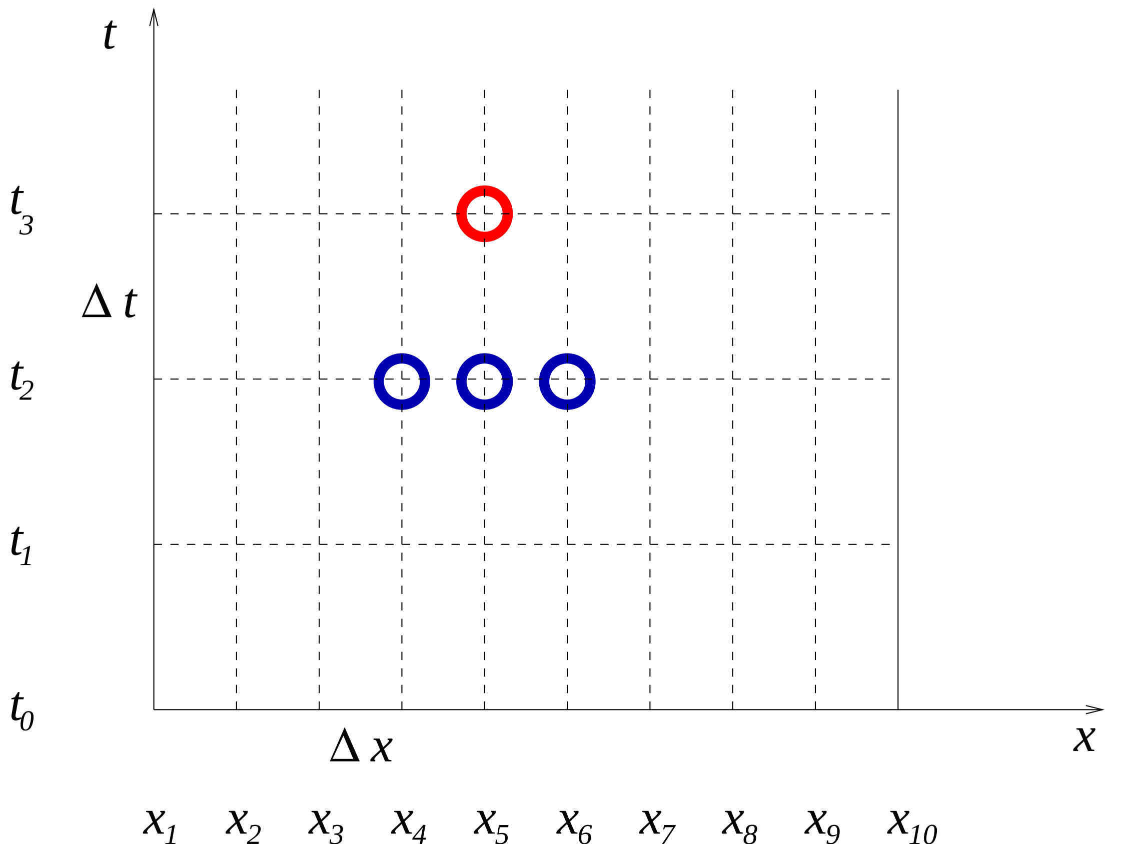
t
$t_3$
$\Delta t$
$t_2$
$t_1$
$t_0$
$\Delta x$
x
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$ $x_8$ $x_9$ $x_{10}$

Figure 5: Illustration of the updating formula (36); $u_5^3$ is computed from $u_4^2$, $u_5^2$, and $u_6^2$.

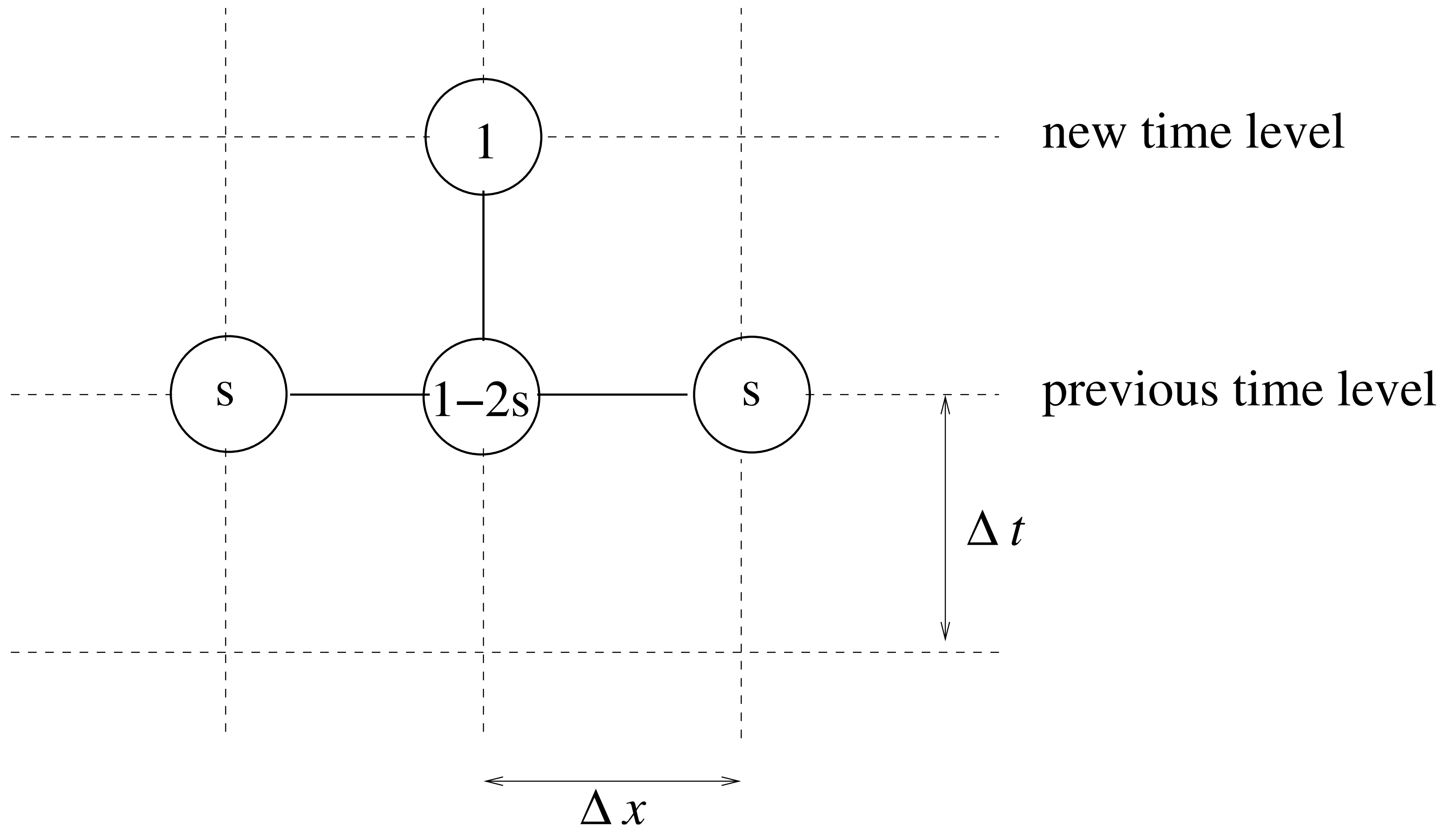

Figure 6: Illustration of the computational molecule corresponding to the finite difference scheme (36). The weight $s$ is $\Delta t/\Delta x^2$.

# Incorporating Boundary Conditions

- (36) can not be used for computing new values at the boundary $u_1^{\ell+1}$ and $u_n^{\ell+1}$, because (36) for $i = 1$ and $i = n$ involves values $u_{-1}^{\ell}$ and $u_{n+1}^{\ell}$ *outside* the grid.
- Therefore we need to use the boundary conditions to update on the boundary $u_1^{\ell+1}$ and $u_n^{\ell+1}$

# Dirichlet Boundary Condition

- Suppose we have the following Dirichlet boundary conditions

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

  where $g_0(t)$ and $g_1(t)$ are prescribed functions
- The new values on the boundary can then be updated by

$$u_1^{\ell+1} = g_0(t_{\ell+1}), \quad u_n^{\ell+1} = g_1(t_{\ell+1})$$

- The numerical scheme $(36)$ update all inner points

**Algorithm 1.** Diffusion equation with Dirichlet boundary conditions.
Set initial conditions:

$$u_i^0 = I(x_i), \quad \text{for } i = 1, \ldots, n$$

for $\ell = 0, 1, \ldots, m$:

- Update all inner points:

$$u_i^{\ell+1} = u_i^\ell + \frac{\Delta t}{\Delta x^2}\left(u_{i-1}^\ell - 2u_i^\ell + u_{i+1}^\ell\right) + \Delta t\, f_i^\ell$$

$$\text{for } i = 2, \ldots, n-1$$

- Insert boundary conditions:

$$u_1^{\ell+1} = g_0(t_{\ell+1}), \quad u_n^{\ell+1} = g_1(t_{\ell+1})$$

# Neumann Boundary Conditions

Assume that we have Neumann conditions on the problem

$$\frac{\partial}{\partial x}u(0,t) = h_0 \quad \text{and} \quad \frac{\partial}{\partial x}u(1,t) = h_1$$

Implementing the first condition, $\frac{\partial}{\partial x}u(0,t) = h_0$, can be done as follows

- We introducing a fictisous value $u_0^\ell$
- The property $\frac{\partial}{\partial x}u(0,t)$ can then be approximated with a centered difference

$$\frac{u_2^\ell - u_0^\ell}{2\Delta x} = h_0$$

## Neumann Boundary Conditions

- The discrete version of the boundary condition then reads

$$\frac{u_2^\ell - u_0^\ell}{2\Delta x} = h_0 \tag{37}$$

or

$$u_0^\ell = u_2^\ell - 2h_0\Delta x$$

- Setting $i = 1$ in (36), gives

$$\begin{aligned} u_1^{\ell+1} &= u_1^\ell + \frac{\Delta t}{\Delta x^2}\left(u_0^\ell - 2u_1^\ell + u_2^\ell\right) + f_1^\ell \\ &= u_1^\ell + \frac{\Delta t}{\Delta x^2}\left(u_2^\ell - 2h_0\Delta x - 2u_1^\ell + u_2^\ell\right) + f_1^\ell \end{aligned}$$

# Neumann Boundary Conditions

- We now have a formula for updating the boundary point

$$u_1^{\ell+1} = u_1^{\ell} + 2\frac{\Delta t}{\Delta x^2}\left(u_2^{\ell} - u_1^{\ell} - h_0\Delta x\right) + f_1^{\ell}$$

- For the condition $\frac{\partial}{\partial x}u(1,t) = h_1$, we can define a fictitious point $u_{n+1}^{\ell}$
- Similar to above we can use a centered difference approximation of the condition, use $(36)$ with $i = n$ and get

$$u_n^{\ell+1} = u_n^{\ell} + 2\frac{\Delta t}{\Delta x^2}\left(u_{n-1}^{\ell} - u_n^{\ell} + h_1\Delta x\right) + f_n^{\ell} \qquad (38)$$

**Algorithm 2.** Diffusion equation with Neumann boundary conditions.
Set initial conditions:

$$u_i^0 = I(x_i), \quad \text{for } i = 1, \dots, n$$

for $\ell = 0, 1, \dots, m$:

- Update all inner points:

$$u_i^{\ell+1} = u_i^\ell + \frac{\Delta t}{\Delta x^2}\left(u_{i-1}^\ell - 2u_i^\ell + u_{i+1}^\ell\right) + \Delta t\, f_i^\ell$$

$$\text{for } i = 2, \dots, n-1$$

- Insert boundary conditions:

$$u_1^{\ell+1} = u_1^\ell + 2\frac{\Delta t}{\Delta x^2}\left(u_2^\ell - u_1^\ell - h_0 \Delta x\right) + f_1^\ell$$

$$u_n^{\ell+1} = u_n^\ell + 2\frac{\Delta t}{\Delta x^2}\left(u_{n-1}^\ell - u_n^\ell + h_1 \Delta x\right) + f_n^\ell$$

## Implementation

We study how Algorithm 1 can be implemented in Python

- Arrays in Python has zero as the first index
- We rewrite Algorithm 1 so that the index $i$ goes from $0$ to $n-1$
- That is, we change $i$ with $i-1$

# Implementation

- In Algorithm 1, we see that we need to store $n$ numbers for $m+1$ time levels, i.e. $n(m+1)$ numbers in a two-dimensional array
- But, when computing the solution at one time level, we only need to have stored the solution at the previous time level - older levels are not used
- So, if we do not need to store all time levels, we can reduce the storage requirements to $2n$ in two one-dimensional arrays
- Introducing $u_i$ for $u_i^{\ell+1}$ and $u_i^-$ for $u_i^{\ell}$, we arrive at the mathematical pseudo code presented as Algorithm 3

**Algorithm 3.** Pseudo code for diffusion equation with general Dirichlet conditions.

```
Set initial conditions:

        $u_i^- = I(x_i)$,    for $i = 0,\ldots,n-1$

for $\ell = 0,1,\ldots,m$:

    • Set $h = \frac{\Delta t}{\Delta x^2}$ and $t = \ell \Delta t$
    • Update all inner points:
          $u_i = u_i^- + h\left(u^- - 2u_i^- + u_{i+1}^-\right) + \Delta t\, f(x_i,t)$
                                          for $i = 1,\ldots,n-2$

    • Insert boundary conditions:
          $u_0 = g_0(t),\;\; u_{n-1} = g_1(t)$
    • Update data structures for next step:
          $u_i^- = u_i,\; i = 0,\ldots,n-1$
```

```
def diffeq(I, f, g0, g1, dx, dt, m, action=None):
    n = int(1/dx + 1)    h = dt/(dx*dx)  # help variable in the scheme
    x = arrayrange(0, 1+dx/2, dx, Float)  # grid points in x dir
    user_data = []  # return values from action function
    # set initial condition:
    um = I(x)
    u = zeros(n, Float)  # solution array

    for l in range(m+1):  # l=0,...,m
        t = l*dt
        # update all inner points:
        for i in range(1,n-1,1):  # i=1,...,n-2
            u[i] = um[i] + h*(um[i-1] - 2*um[i] + um[i+1]) +  dt*f(x[i],
        # insert boundary conditions:
        u[0] = g0(t);  u[n-1] = g1(t)

        # update data structures for next step:
        for i in range(len(u)): um[i] = u[i]
        if action is not None:
            r = action(u, x, t)  # some user-defined action
            if r is not None:
                user_data.append(r)  # r can be arbitrary data...
    return user_data
```

## Comments

- The functions $f$, $g_0$, and $g_1$ are given as function arguments for convenience
- We need to specify each array element in the solution `u` to be a floating-point number, otherwise the array would consist of integers. The values of `u` are of no importance before the time loop.
- The `action` parameter may be used to invoke a function for computing the error in the solution, if the exact solution of the problem is known, or we may use it to visualize the graph of $u(x,t)$. The `action` function can return any type of data, and if the data differ from `None`, the data are stored in an array `user_data` and returned to the user.

## Verifications

- A well known solution to the diffusion equation is

$$u(x,t) = e^{-\pi^2 t}\sin \pi x, \tag{39}$$

  which is the solution when $f = 0$ and $I(x) = \sin \pi x$ and the Dirichlet boundary conditions are $g_0(t) = 0$ and $g_1(t) = 0$
- We shall see how this exact solution can be used to test the code
- In Python the initial and boundary conditions can specified by

```
def IC_1(x):   return sin(pi*x)
def g0_1(t):   return 0.0
def g1_1(t):   return 0.0
```

# Verifications

- We can now construct a function `compare_1` as `action` parameter, where we compute and return the error:

```
def error_1(u, x, t):
    e = u - exactsol_1(x, t)
    e_norm = sqrt(innerproduct(e,e)/len(e))
    return e_norm

def exactsol_1(x, t): return exp(-pi*pi*t)*sin(pi*x)
```

- The `e_norm` variable computes an approximation to the a scalar error measure

$$E = \sqrt{\int_0^1 (\hat{u} - u)^2 dx},$$

where $\hat{u}$ denotes the numerical solution and $u$ is the exact solution

# Verifications

- We actually computes a Riemann approximation of this integral since

$$E^2 = \int_0^1 (\hat{u} - u)^2 dx \approx \sum_{i=0}^{n-1} e_i^2 \Delta x = \frac{1}{n-1} \sum_{i=0}^{n-1} e_i^2,$$

  where

$$e_i = u_i^\ell - \exp\left(-\pi^2 \ell \Delta t\right) \sin(\pi i \Delta x)$$

  (the code divide by $n$ instead of $n-1$, for convenience)
- The final call to `diffeq` reads

```
e = diffeq(IC_1, f0, g0_1, g1_1, dx, dt, m, action=error_1)
print "error at last time level:", e[-1]
```

# Verifications

- Theoretically, it is known that

$$E = C_1 \Delta x^2 + C_2 \Delta t$$

- Choosing $\Delta t = D\Delta x^2$ for a positive constant $D$, we get

$$E = C_3 \Delta x^2, \quad C_3 = C_1 + C_2 D$$

- Hence, $E/\Delta x^2$ should be constant
- A few lines of Python code conduct the test

```
dx = 0.2
for counter in range(4):  # try 4 refinements of dx
    dx = dx/2.0; dt = dx*dx/2.0; m = int(0.5/dt)
    e = diffeq(IC_1, f0, g0_1, g1_1, dx, dt, m, action=error_1)
    print "dx=%12g error=%12g  ratio=%g" % (dx, e[-1], e[-1]/(dx*dx)
```

## Verifications

- The output becomes

```
dx=           0.1 error= 0.000633159  ratio=0.0633159
dx=          0.05 error=  0.00016196  ratio=0.0647839
dx=         0.025 error= 4.09772e-05  ratio=0.0655636
dx=        0.0125 error= 1.03071e-05  ratio=0.0659656
```

- This confirms that $E \sim \Delta x^2$

# Variable Coefficients

- The heat conduction equation (15) allows for variable coefficients
- We shall now see how we can discretize a diffusion equation with variable coefficients

$$\rho(x_i)c_v(x_i)\frac{\partial}{\partial t}u(x_t,t_\ell) = \left[\frac{\partial}{\partial x}\left(k(x)\frac{\partial u}{\partial x}\right)\right]_{x=x_i,t=t_\ell} + f(x_i,t_\ell)$$

- The left hand side can be discretized similar to above, and we abbreviate $\rho(x_i)c_v(x_i)$ with $\gamma_i$

# Variable Coefficients

- For the first term of the right hand side we approximate it similar to above - in two steps and based on centered differences
- We first approximate the outer derivative at $x = x_i$ (and $t = t_\ell$), using a fictitious point $x_{i+\frac{1}{2}} = x_i + \frac{1}{2}\Delta x$ to the right and a fictitious point $x_{i-\frac{1}{2}} = x_i - \frac{1}{2}\Delta x$ to the left

$$\frac{\partial}{\partial x} k(x) \left[ \left( \frac{\partial u}{\partial x} \right) \right]_i^\ell \approx \frac{1}{\Delta x} \left[ k_{i+\frac{1}{2}} \left[ \frac{\partial u}{\partial x} \right]_{i+\frac{1}{2}}^\ell - k_{i-\frac{1}{2}} \left[ \frac{\partial u}{\partial x} \right]_{i-\frac{1}{2}}^\ell \right],$$

where $k_{i-\frac{1}{2}} = k(x_{i-\frac{1}{2}})$ and $k_{i+\frac{1}{2}} = k(x_{i+\frac{1}{2}})$

# Variable Coefficients

- Further we approximate

$$k_{i+\frac{1}{2}}\left[\frac{\partial u}{\partial x}\right]^{\ell}_{i+\frac{1}{2}} \approx k_{i+\frac{1}{2}}\frac{u_{i+1}-u_i}{\Delta x}$$

  and

$$k_{i-\frac{1}{2}}\left[\frac{\partial u}{\partial x}\right]^{\ell}_{i-\frac{1}{2}} \approx k_{i-\frac{1}{2}}\frac{u_i-u_{i-1}}{\Delta x}$$

# Variable Coefficients

- Inserting these approximations in the heat conduction equation with variable coefficients gives

$$\gamma_i \frac{u_i^{\ell+1} - u_i}{\Delta t} = \frac{1}{\Delta x}\left(k_{i+\frac{1}{2}}\frac{u_{i+1} - u_i}{\Delta x} - k_{i-\frac{1}{2}}\frac{u_i - u_{i-1}}{\Delta x}\right) + f_i^\ell$$

- Solving for $u_i^{\ell+1}$ gives us

$$u_i^{\ell+1} = u_i^\ell + \frac{1}{\gamma_i}\frac{\Delta t}{\Delta x}\left(k_{i+\frac{1}{2}}\frac{u_{i+1}^\ell - u_i^\ell}{\Delta x} - k_{i-\frac{1}{2}}\frac{u_i^\ell - u_{i-1}^\ell}{\Delta x}\right) + \frac{\Delta t}{\gamma_i} f_i^\ell \qquad (40)$$

- Inserting the boundary conditions is similar to above