

RECAP



CLASSIFICATION OF DE'S BY CONDITIONS

To integrate a function we produce integration constants. This also occurs with differential equations. To determine these integration constants we can have two types of extra conditions:

- **Initial value problems.** Given the system of ODE's

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{g}(t, \mathbf{y}(t))$$

and the **initial conditions**

$$\mathbf{y}(t_0) = \mathbf{y}_0$$

one can solve the differential equation to obtain

$$\mathbf{y}(t) \text{ for } t > t_0$$

This is what we did so far in the second term.



- **Boundary value problems.** Instead of all the conditions at the same value t_0 , they are given at different values. For example

$$\frac{d^2y(t)}{dt^2} = g(t, y(t), y'(t))$$

We could have $y(t_0) = y_0$ and $y(t_0)' = y_0'$ making it an initial value problem. If instead $y(t_0) = y_0$ and $y(t_1) = y_1$ where $t_1 \neq t_0$ we have **boundary conditions** and hence a *boundary value problem*.

The latter implies we can't just integrate forward from $t = t_0$, since we don't know the value of the derivative at $t = t_0$.

Sometimes a differential equation has both initial and boundary conditions.



PDE

As mentioned before, a partial differential equation (PDE) is a differential equation involving *more* than 1 independent variable.



PDE

As mentioned before, a partial differential equation (PDE) is a differential equation involving *more* than 1 independent variable.

DIFFUSION EQUATION

An example of a PDE is the **diffusion equation**

$$\frac{\partial u(t, \mathbf{r})}{\partial t} = \alpha \nabla^2 u(t, \mathbf{r})$$

with α a diffusion constant. This is typically used in the study of heat diffusion, where u would be the temperature. The independent variables are time t and the spatial coordinate(s) \mathbf{r} , which in 3D is

$$\mathbf{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$



The diffusion equation is a combination of a initial and boundary value problem:

- the temperature of the whole body at the start time t_0
- typically (but not always) the temperature at the boundary of the body is specified

Then future times $t > t_0$ are acquired by (numerically) solving the PDE.



In the diffusion equation the nabla symbol ∇ stands for the gradient,

$$\nabla \doteq \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix}$$

and $\nabla^2 = \nabla \cdot \nabla$, i.e., the inner product of the gradient with itself, is known as the Laplacian

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

For only one spatial dimension we have $\mathbf{r} = x$ and the Laplacian simplifies to

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



INITIAL CONDITION

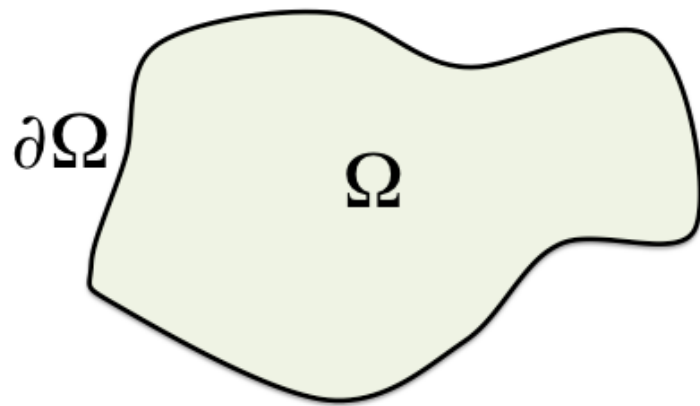
Initial condition:

$$u(t_0, \mathbf{r}) = f_1(\mathbf{r})$$

where $f_1(\mathbf{r})$ is a known function. For heat diffusion this is the initial temperature profile.



BOUNDARY CONDITION



Assume the domain of \mathbf{r} is Ω , then the boundary can be denoted as $\partial\Omega$.

DIRICHLET BOUNDARY CONDITION

When the unknown variable u is prescribed at the boundary, we have the boundary condition

$$u(t, \mathbf{r})|_{\mathbf{r} \in \partial\Omega} = f_2(\mathbf{r})$$

For heat diffusion, this would be the temperature at the boundary.

This is known as a **Dirichlet boundary condition**: the unknown variable u is prescribed at the boundary.

NEUMANN BOUNDARY CONDITION

Another possibility would be to prescribe the derivative of the unknown u :

$$\frac{\partial u(t, \mathbf{r})}{\partial \mathbf{n}} \Big|_{\mathbf{r} \in \partial\Omega} = f_3(\mathbf{r})$$

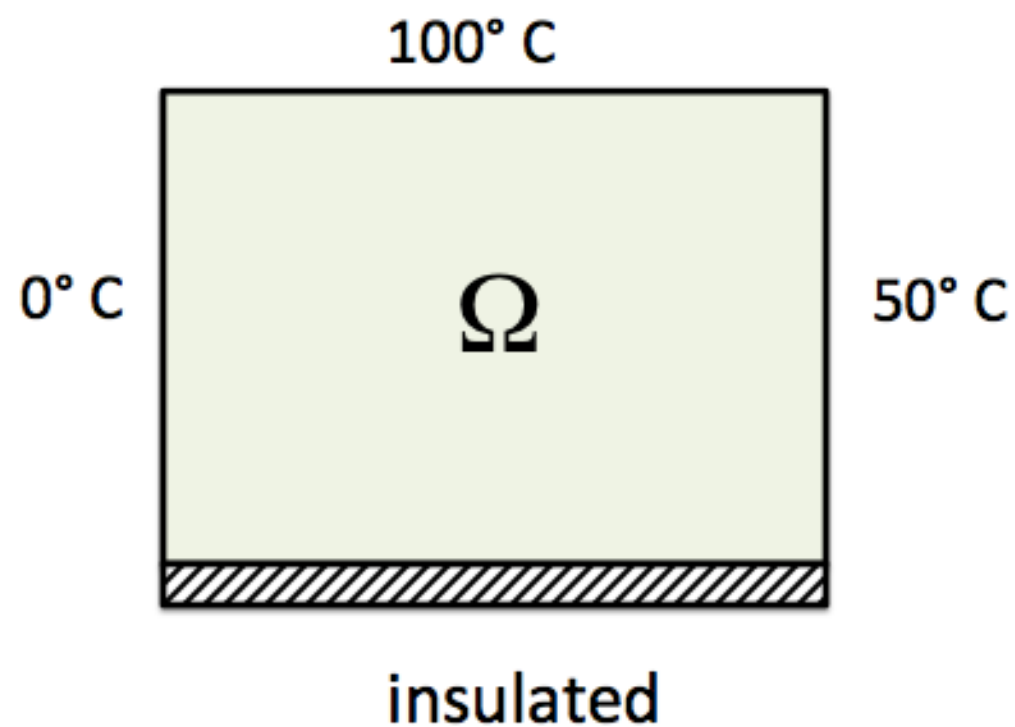
where \mathbf{n} is the normal to the surface $\partial\Omega$. For heat diffusion, this could correspond to an insulated body, in which case $f_3 = 0$.

This is known as a **Neumann boundary condition**: the *derivative* of the unknown variable is prescribed at the boundary.

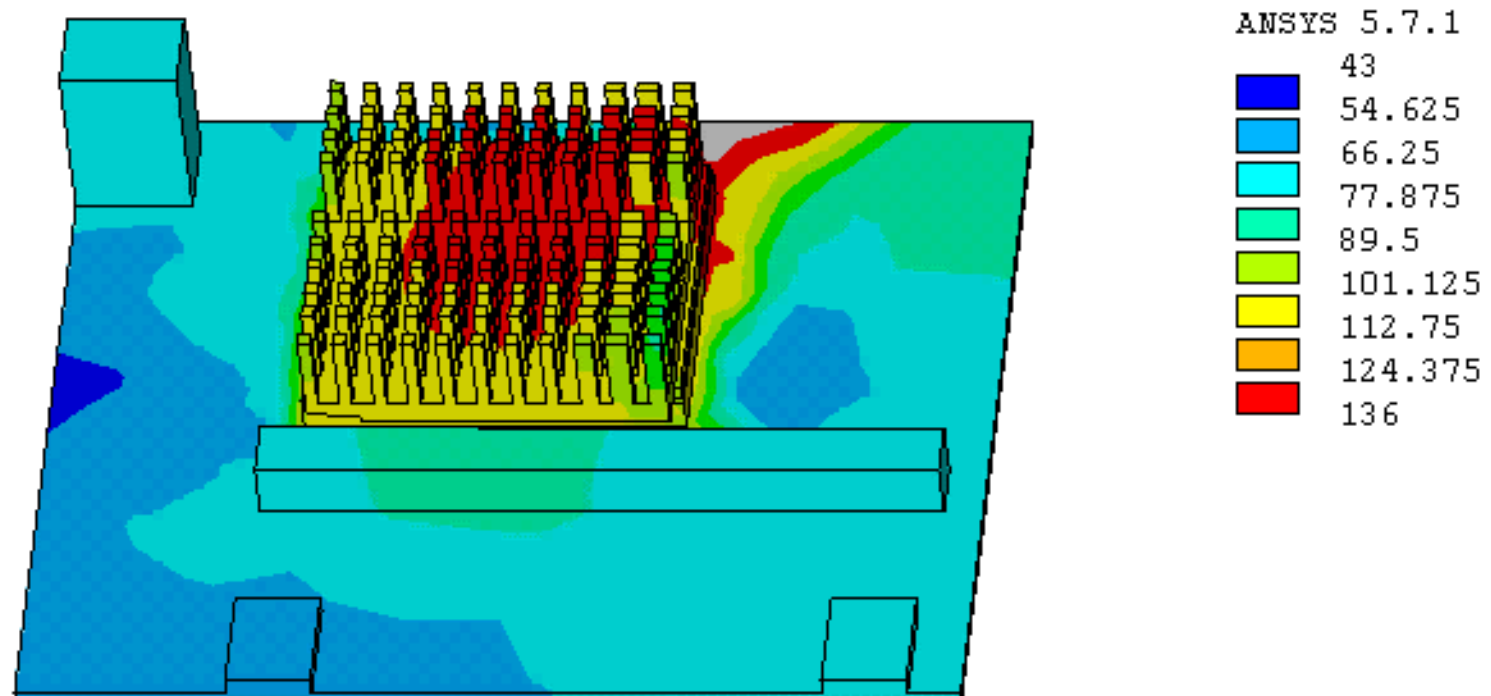


MIXED BOUNDARY CONDITIONS

Mixed boundary conditions could be that at some parts of the boundary u is prescribed, and other parts $\partial u / \partial \mathbf{n}$. For heat diffusion this could correspond to a body partly insulated ($\partial u / \partial \mathbf{n} = 0$) and partly with a prescribed temperature ($u = u_0$).



EXAMPLE



Temperature distribution of a circuit board with heat sink element

LAPLACE EQUATION

The **Laplace equation** is the steady state solution of the diffusion equation ($\nabla^2 u = \partial u / \partial t$):
 $\partial u / \partial t = 0$, and therefore

$$\nabla^2 u(\mathbf{r}) = 0$$



POISSON EQUATION

The **Poisson equation** is similar to the Laplace equation, but now the RHS is generally not zero:

$$\nabla^2 u(\mathbf{r}) = g(\mathbf{r})$$

Hence, if we have a solver for the Poisson equation, it is easy to have one for the Laplace equation, by setting $g = 0$ in the above equation.

For these equations we only have *boundary conditions*, no initial conditions.



1D POISSON EQUATION

In 1D the Poisson equation reduces to

$$\frac{d^2u}{dx^2} = g(x)$$



DOUBLE DERIVATIVE

By employing the finite difference method, we replace derivatives by finite differences. The higher order derivatives can be determined by employing finite differences for each derivative.

We start with the first derivative

$$\frac{du}{dx} \approx \frac{u(x + \Delta x) - u(x)}{\Delta x}$$

The second derivative then is

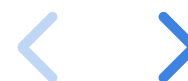
$$\frac{d}{dx} \frac{du}{dx} \approx \frac{d}{dx} \frac{u(x + \Delta x) - u(x)}{\Delta x}$$

The first term on the RHS can be replaced by using

$$\frac{du(x + \Delta x)}{dx} \approx \frac{u(x + 2\Delta x) - u(x + \Delta x)}{\Delta x}$$

We already had the finite difference for $du(x)/dx$, so

$$\begin{aligned} \frac{d}{dx} \frac{du}{dx} &\approx \frac{u(x + 2\Delta x) - u(x + \Delta x) - (u(x + \Delta x) - u(x))}{\Delta x^2} \\ &= \frac{u(x + 2\Delta x) - 2u(x + \Delta x) + u(x)}{\Delta x^2} \end{aligned}$$



1D SCHEME

For the Poisson equation, $u(x)'' = g(x)$, the resulting double derivative is equal to $g(x)$:

$$\frac{u(x + 2\Delta x) - 2u(x + \Delta x) + u(x)}{\Delta x^2} = g(x) \quad (*)$$

Introduce now an index notation: $x_n = n\Delta x$, $u(x) = u(x_n) = u_n$. Then eq. (*) becomes

$$\frac{u_{n+2} - 2u_{n+1} + u_n}{\Delta x^2} = g_n \quad (**)$$

One can proof that instead of taking

$$\lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h} = \frac{df(x)}{dx}$$

one can equivalently take

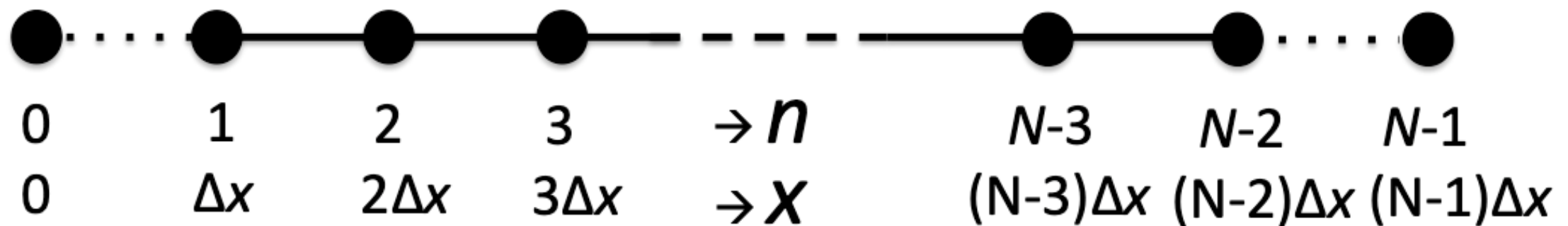
$$\lim_{h \rightarrow 0} \frac{f(x) - f(x - h)}{h} = \frac{df(x)}{dx}$$

Hence, an alternative discretisation approximation to the second derivative would be

$$\frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta x^2} = g_n$$

This is more favourable than (**), because it is *symmetric*, and we will use it from now on.





Take now as boundary conditions $u_0 = u_L$ and $u_{N-1} = u_R$ with u_L and u_R known constants.

These conditions give relations for u_0 and u_{N-1} . Relations for the internal points u_1, u_2, \dots, u_{N-2} , follow from using the finite difference scheme for the second order derivative in the Poisson equation.

Therefore, the equations for every point $u_n, n \in 0, 1, \dots, N - 1$ is given by

$$\begin{aligned}
 u_0 &= u_L \\
 u_2 - 2u_1 + u_0 &= g_1 \Delta x^2 \\
 u_3 - 2u_2 + u_1 &= g_2 \Delta x^2 \\
 &\vdots \\
 u_{n+1} - 2u_n + u_{n-1} &= g_n \Delta x^2 \\
 &\vdots \\
 u_{N-1} - 2u_{N-2} + u_{N-3} &= g_{N-2} \Delta x^2 \\
 u_{N-1} &= u_R
 \end{aligned}$$

The system of equations can be written as an equation involving the unknown vector \mathbf{u} and a matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} u_L \\ g_1 \Delta x^2 \\ g_2 \Delta x^2 \\ g_3 \Delta x^2 \\ \vdots \\ g_{N-2} \Delta x^2 \\ u_R \end{pmatrix}$$

Equivalently,

$$\begin{pmatrix} -2 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} -2u_L \\ g_1 \Delta x^2 \\ g_2 \Delta x^2 \\ g_3 \Delta x^2 \\ \vdots \\ g_{N-2} \Delta x^2 \\ -2u_R \end{pmatrix}$$

or

$$\mathbf{A}\mathbf{u} = \mathbf{g}$$

and can be solved using Gaussian elimination (or Matrix inversion). The Gaussian elimination has been carried out in the first term with Matt.



DIFFERENTIAL ANALYZER



In the first half of the 20th century, before the existence of computer chips, electro-mechanical analogue machines were used to solve differential equations.

EXPLICIT NUMERICAL SCHEME FOR THE 1D DIFFUSION EQUATION

Let us now return to the diffusion equation:

$$\frac{\partial u(t, \mathbf{r})}{\partial t} = \alpha \nabla^2 u(t, \mathbf{r})$$

For simplicity take $\alpha = 1$.

Then in one spatial dimension this diffusion equation reduces to

$$\frac{\partial u(t, x)}{\partial t} = \frac{\partial^2 u(t, x)}{\partial x^2}$$

To solve this equation we also need boundary ($u(t, x_L) = u_L$ and $u(t, x_R) = u_R$, where u_L and u_R are known constants) and initial ($u(t_0, x) = f_0(x)$, where $f_0(x)$ is a known function) conditions.



GRID

The equation will be solved on a grid. We solve the function $u(t, x)$ at the N_x spatial grid points

$$x_n = x_L + n \cdot \Delta x$$

with $\Delta x = (x_R - x_L)/(N_x - 1)$; and N_t temporal grid points:

$$t_i = t_0 + i \cdot \Delta t$$

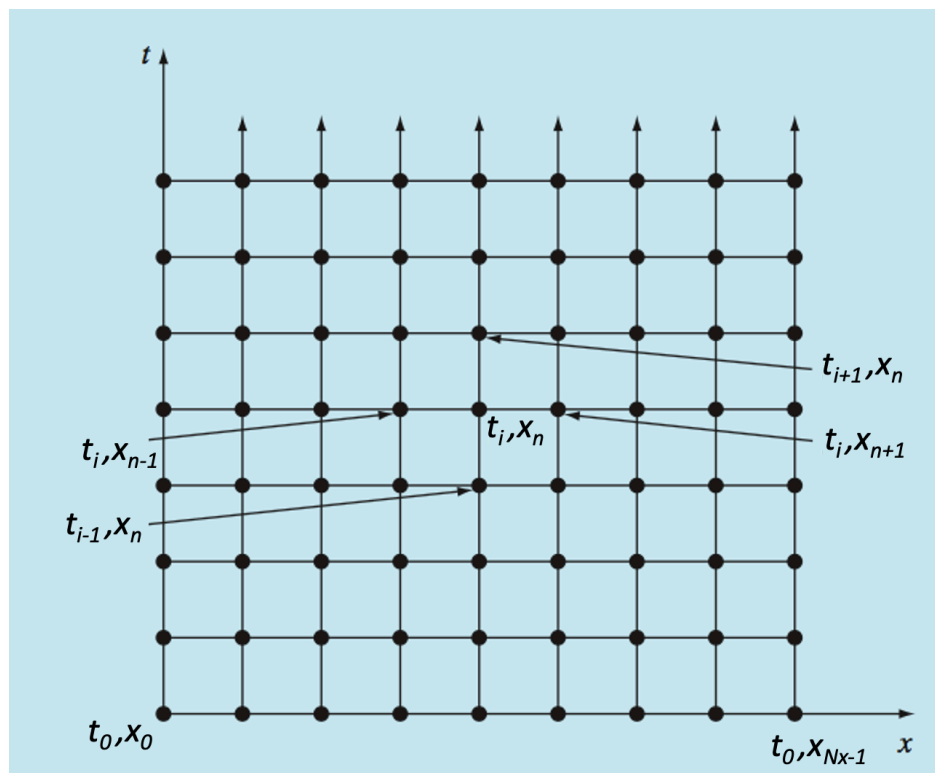
with $\Delta t = (t_{\max} - t_0)/(N_t - 1)$ and $N_t - 1$ integration steps.

Set the origin so that $x_L = 0$ and $t_0 = 0$ and introduce the index notation $u_{i,n}$

$$u(t, x) = u(t_i, x_n) = u(i\Delta t, n\Delta x) = u_{i,n}$$



The points with indices are given by



The associated values for t and x are then obtained by multiplying the indices by Δt and Δx , respectively.

RELATION NEXT TIME STEP

As with the 1D Poisson equation we can discretize the RHS of $\frac{\partial u(t,x)}{\partial t} = \frac{\partial^2 u(t,x)}{\partial x^2}$:

$$\frac{\partial^2 u(t, x)}{\partial x^2} \approx \frac{u_{n+1}(t) - 2u_n(t) + u_{n-1}(t)}{\Delta x^2}$$

and hence we have

$$\frac{du_n(t)}{dt} \approx \frac{u_{n+1}(t) - 2u_n(t) + u_{n-1}(t)}{\Delta x^2}$$

Notice that this is a *system of coupled first order ordinary differential equations* (one coupled ODE for each value of n , except for the boundary values), the same as what we saw before!



We can therefore use the same technique for solving this PDE.

For example, the time derivative can be determined using the explicit Euler method:

$$u_n(t + \Delta t) \approx u_n(t) + \frac{\Delta t}{\Delta x^2} (u_{n+1}(t) - 2u_n(t) + u_{n-1}(t))$$

Rewrite in time index i for t :

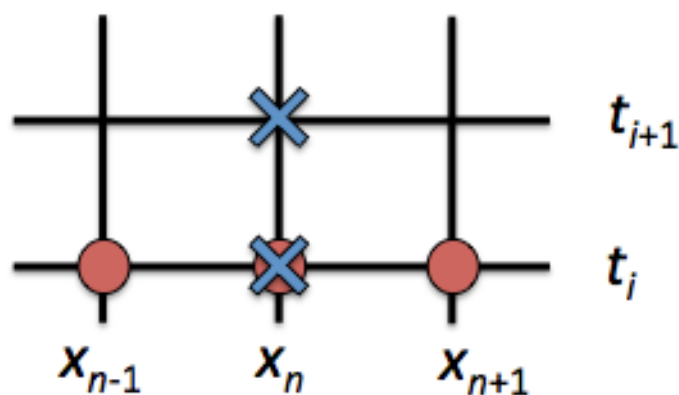
$$u_{i+1,n} \approx u_{i,n} + \frac{\Delta t}{\Delta x^2} (u_{i,n+1} - 2u_{i,n} + u_{i,n-1})$$



Notice we can now move everything depending on $i + 1$ (at the *next* point of time) to the LHS, and everything on i (at the *current* point of time) to the RHS:

$$\begin{aligned} u_{i+1,n} &\approx u_{i,n} + \frac{\Delta t}{\Delta x^2} (u_{i,n+1} - 2u_{i,n} + u_{i,n-1}) \\ &= \left(1 - 2\frac{\Delta t}{\Delta x^2}\right) u_{i,n} + \frac{\Delta t}{\Delta x^2} (u_{i,n+1} + u_{i,n-1}) \end{aligned}$$

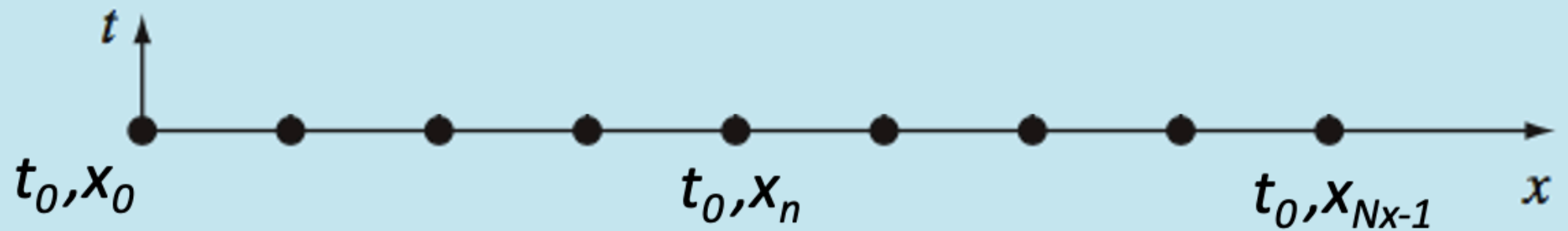
This is an explicit scheme and has the following *stencil*



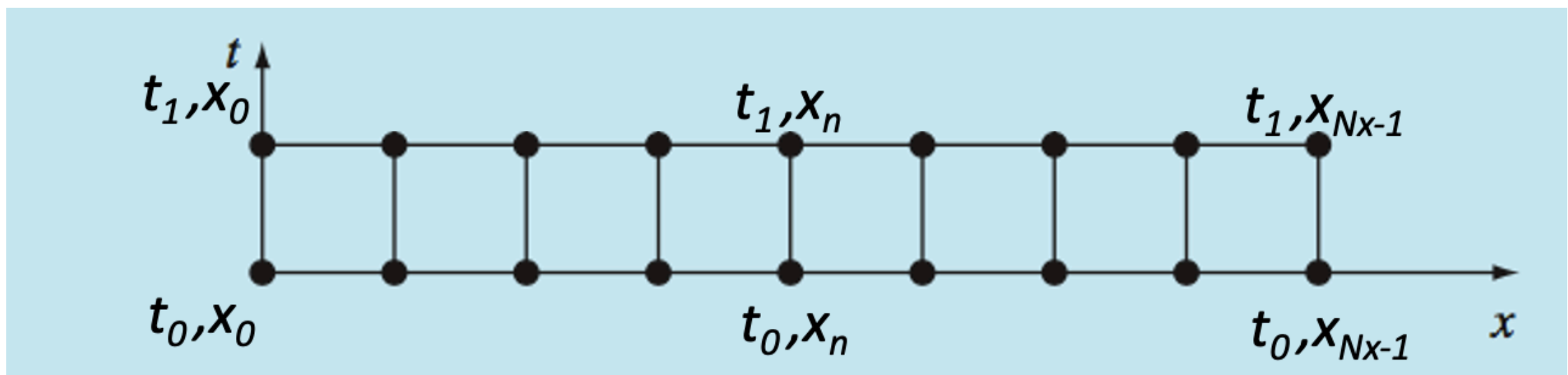
In the image *crosses* indicate grid points involved in time differences and *circles* grid points involved in space differences. Notice that for determining $u_{i+1,n}$ at the next time point t_{i+1} , we need information about the current time point t_i at three spatial points x_n , x_{n-1} and x_{n+1} .

ALGORITHM

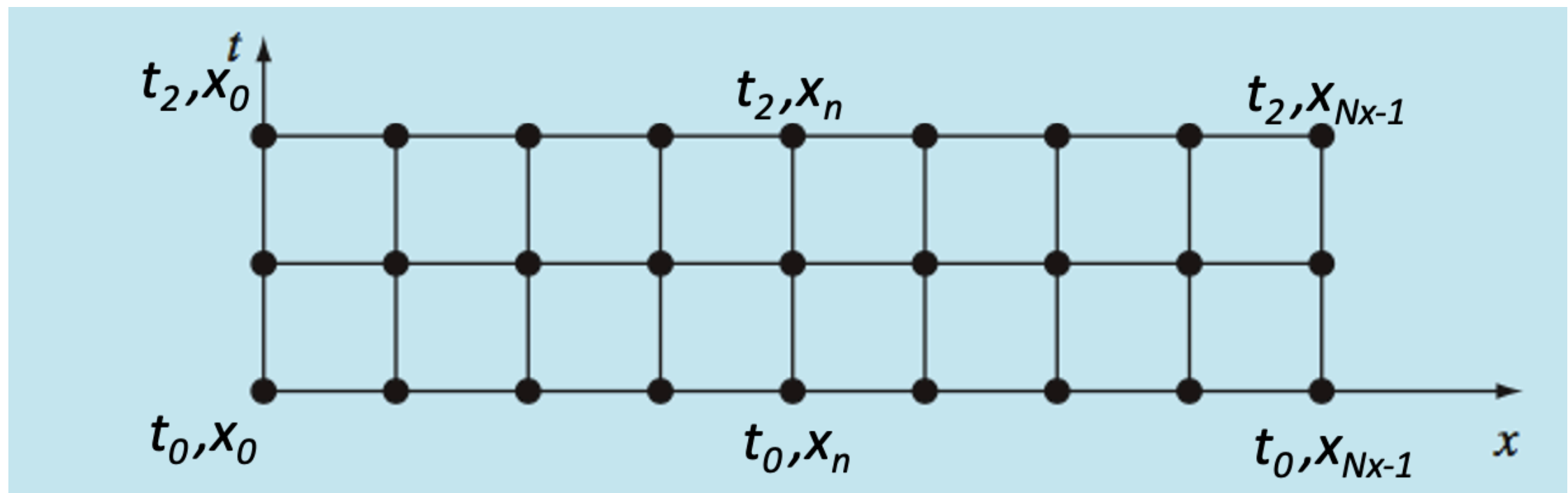
We start with the boundary and initial conditions at $t = t_0 = 0$ that will specify $u(t = 0, x)$.



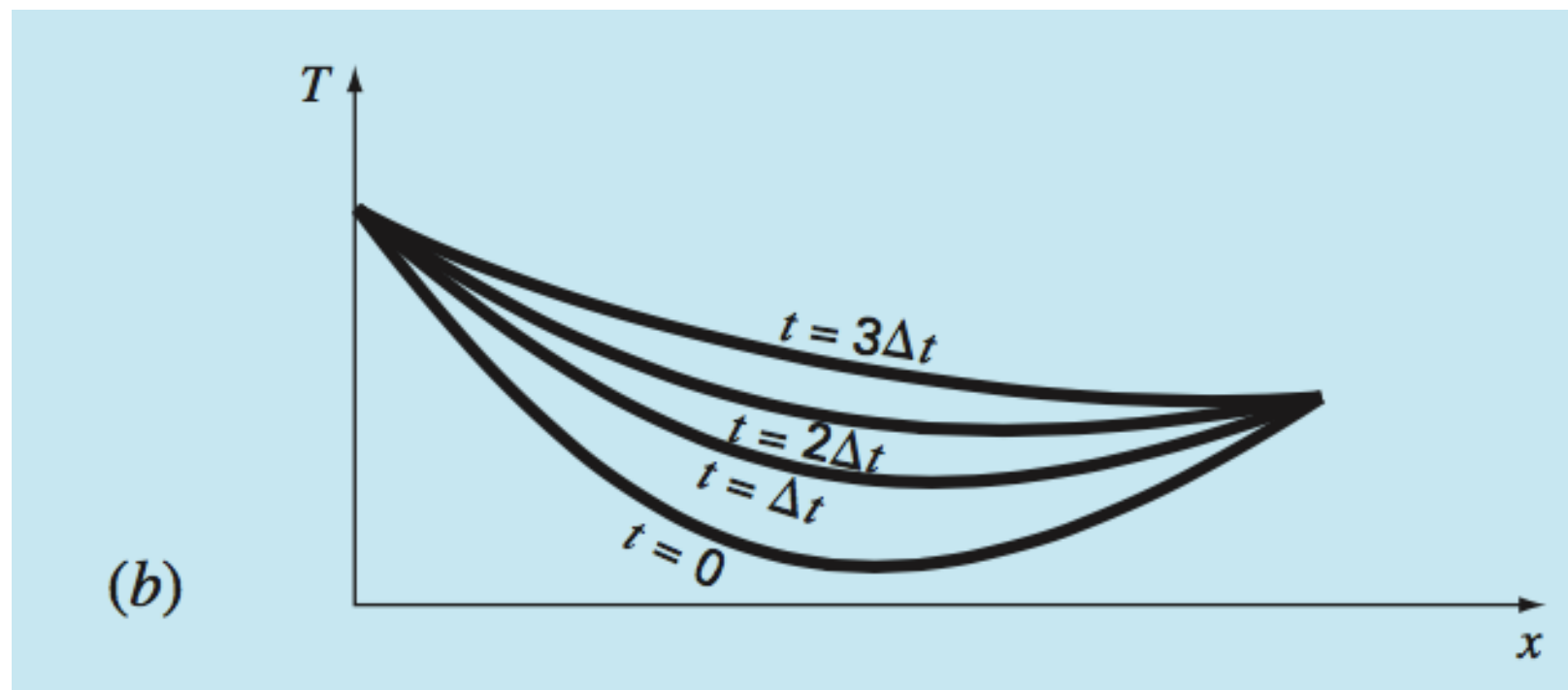
Once we used the initial condition to set the values of $u(0, x)$ at these grid points, we can determine then at the next time step, $t = \Delta t$, so $u(\Delta t, x)$.



We repeat the calculation for the next time step



The solution then looks like



TURNING INTO CODE

See hint (Python file) on Blackboard.



EXERCISES SESSION 6:

1. What is the difference between a Dirichlet boundary conditions and a Neumann boundary condition?
2. The remaining exercises will be marked (but count 0% for the final mark). It will help to get feedback on your code. See Assessment section on Blackboard.

