Machine Learning: Bra Tittel

Vetle Nevland, Vetle Vikenes & Sigurd Sørlie Rustad

FYS-STK4155 – Applied Data Analysis and Machine Learning
Autumn 2021
Department of Physics
University of Oslo

November 23, 2021

ABSTRACT: Coming soon!

Contents

1 Introduction

We will in no way answer all questions linked to the aforementioned methods. So that anyone can reproduce or continue our studies, we list all the code, results and instructions on running the code in our GitHub repository[1].

2 Theory

In the theory-section we aim to give a brief explanation of the main concepts and terminology used in this report. For a more in-depth explanation we recommend reading the appropriate sections in [3], which has been of great inspiration and help for us throughout the project.

The diffusion equation

The full diffusion equation reads

$$\frac{\partial u(\mathbf{r},t)}{\partial t} = \nabla \cdot \left[D(u,\mathbf{r}) \nabla u(\mathbf{r},t) \right], \tag{2.1}$$

where $\mathbf{r}$ is a positional vector and $D(u,r)$ the collective diffusion coefficient. If $D(u,\mathbf{r}) = 1$ the equation simplifies to a linear differential equation

$$\frac{\partial u}{\partial t} = \nabla^2 u(\mathbf{r},t), \tag{2.2}$$

or

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) u(x,y,z,t) = \frac{\partial u(x,y,z,t)}{\partial t} \tag{2.3}$$

in cartesian coordinates. In this report we are going to study a one dimensional rod of length $L = 1$. I.e. we need the one dimensional diffusion equation

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t}, \tag{2.4}$$

with boundary conditions

$$u(x,0) = \sin(\pi x) \quad 0 \leq x \leq L, \tag{2.5}$$

$$u(0,t) = 0 \quad t \geq 0 \text{ and} \tag{2.6}$$

$$u(L,t) = 0 \quad t \geq 0. \tag{2.7}$$

Analytical solution

An analytical solution of the 1D diffusion equation can be dervied using the method of separation of variables.

$$u(x,t) = X(x)T(t)$$

[1] https://github.com/sigurdru/FYS-STK4155/tree/main/project3

The solution is separated into a function X only depending on the independent variable x, and a function T only depending on the independent variable t. Equation 2.4 can then be rewritten as

$$\begin{aligned}
\frac{\partial^2 X(x)T(t)}{\partial x^2} &= \frac{\partial X(x)T(t)}{\partial t} \\
T(t)\frac{\partial^2 X(x)}{\partial x^2} &= X(x)\frac{\partial T(t)}{\partial t} \\
\frac{1}{X}\frac{\partial^2 X(x)}{\partial x^2} &= \frac{1}{T}\frac{\partial T(t)}{\partial t}
\end{aligned}$$

The core of the method now becomes clear. The independent variables are separated and put on either side of the equation. Because x and t are independent we may fix one of them, say x, while letting the other (t) vary. The left side of the equation is thus constant, and since we have equality the expression on the right side must equal the same constant, for arbitrary t. Therefore, we can set the left side and right side equal to a constant $-k^2$. The reason for defining a negative constant is to prevent a growing solution, which will be clear on the derivation.

$$\frac{1}{X}\frac{\partial^2 X(x)}{\partial x^2} = \frac{1}{T}\frac{\partial T(t)}{\partial t} = -k^2$$

This is solved for the functions X and T separately. ...

Explicit forward Euler

In this section we want to cover the explicit forward Euler. By explicit we mean that the value at the next grid point is determined entirely by known or previously calculated values.

The one-dimensional diffusion equation (2.4) reads

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t} \quad \text{or} \quad u_{xx} = u_t. \tag{2.8}$$

In this report we are going to study a one dimensional rod of length $L = 1$, with boundary conditions

$$u(x,0) = \sin(\pi x) \quad 0 \leq x \leq L, \tag{2.9}$$

$$u(0,t) = 0 \quad t \geq 0 \text{ and} \tag{2.10}$$

$$u(L,t) = 0 \quad t \geq 0. \tag{2.11}$$

To approximate the solution, we have to discretize the position and time coordinates. We can choose $\Delta x = L/N$ and Δt as small steps in x-direction and time, where N are the number of discretized points in x-direction. Then we can define the value domain of t and

x,

$$t_j = j\Delta t, \quad j \in \mathbb{N}_0 \quad \wedge \quad x_i = i\Delta x, \quad \{i \in \mathbb{N}_0 | i \leq N\}.$$

The algorithm for explicit forward Euler in one dimension (from [1] chapter 10.2.1) reads

$$u_{i,j+1} = \alpha u_{i-1,j} + (1-2\alpha)u_{i,j} + \alpha u_{i+1,j} \tag{2.12}$$

where

$$\alpha = \frac{\Delta t}{\Delta x^2}.$$

This has a local approximate error of $O(\Delta t)$ and $O(\Delta x^2)$.

We are to use the forward Euler method to discretize the diffusion equation to be solved numerically. The forward Euler is an explicit scheme, meaning that the derivative in time is approximated at the current time level. That is, the derivative is discretized only by known values. As a consequence, the discretized diffusion equation can be explicitly be solved for the next time step without the need of any matrix inversion to arrive at a coupled set of discretized equations. The drawback of explicit methods is that it is less stable than implicit schemes, which approximate the derivative at the next time step. If the derivative is calculated at the current time level, we miss any information about how the solution changes at the next time level. Hence, if the gradient of the next time level is significantly different than of the current time level, the numerical solution may deviate considerably from the true solution. If the deviation is large enough, further iterations could potentially cause instable solutions in the sense that it diverges.

3 Method

Unit testing

Before the numerical explicit scheme is applied to a particular problem, it is important to test that the discretized equations return expected results. Unit tests are constructed to test the implementation . This is done by manually calculating the solution of the first two time steps given the initial condition, and comparing the result with that obtained by the numerical scheme. Since the same recursive formula is used for all time steps, it is sufficient to test the two first time steps. To avoid too much computations we choose five equally sized intervals between $x = 0$ and $x = L = 1$, that is $\Delta x = 0.2$. The time step is set to $\Delta t = 0.01$, ensuring stability. For the first time step $j = 1$, that is $t = \Delta t$, we have the following two boundary values and four interior points

$$\begin{aligned}
u_0^1 &= u_5^1 = 0 \\
u_i^1 &= \frac{\Delta t}{\Delta x^2}u_{i+1}^0 + (1 - 2\frac{\Delta t}{\Delta x^2})u_i^0 + \frac{\Delta t}{\Delta x^2}u_{i-1}^0 \\
&= 0.05u_{i+1}^0 + 0.9u_i^0 + 0.05u_{i-1}^0
\end{aligned}$$

Using the initial condition $u_*^0 = \Big(\sin(0), \sin(0.2\pi), \sin(0.4\pi), \sin(0.6\pi), \sin(0.8\pi), \sin(\pi)\Big)$, we get

$$
\begin{aligned}
u_0^1 &= 0 \\
u_1^1 &= 0.531656755 \\
u_2^1 &= 0.8602387 \\
u_3^1 &= 0.8602387 \\
u_4^1 &= 0.531656755 \\
u_5^1 &= 0
\end{aligned}
$$

The forward euler scheme prints out the following values for the first time step

$$(0., 0.53165676, 0.8602387, 0.8602387, 0.53165676, 0.)$$

Hence, the result from forward euler is equal to the manual calulations up to machine precision.

Using the values from the first time step, we can calculate the solution at second time step. The recursive formula makes the manual calculations straightforward.

$$
\begin{aligned}
u_0^2 &= 0 \\
u_1^2 &= 0.480888053 \\
u_2^2 &= 0.778093215 \\
u_3^2 &= 0.778093215 \\
u_4^2 &= 0.480888053 \\
u_5^2 &= 0
\end{aligned}
$$

The forward euler scheme gives the following for the second time step

$$(0., 0.48088805, 0.77809321, 0.77809321, 0.48088805, 0.)$$

Once again, the two approaches provide the same result up to machine precision. Thus, the internal functionality of forward euler is validated. It remains to verify the accuracy of the scheme by comparing it with the analytical solution of the diffusion equation 2.4 with provided initial and boundary conditions.

Accuracy assessment

The unit test shows that the forward euler scheme works as expected, but to what accuracy can it reproduce the analytical solution? If To assess this we compare the numerical solution to the analytical solution by the total integrated error [2]

$$E = ||e_i^j||_{L_2}^2 = \sqrt{\sum_{j=1}^{N_t}\sum_{i=1}^{N_x}(e_i^j)^2\,\Delta x \Delta t},$$

where $e_i^j = u_i^j - u_e(x_i, t_j)$ and $u_e(x_i, t_j)$ is the analytical solution at step i in space and step j in time. If $E \to 0$ as $\Delta t \to 0$ the numerical scheme converges to the analytical solution.

4 Results

5 Discussion

6 Conclusion

References

[1] Morten Hjorth-Jensen. Computational physics, lecture notes fall 2015. *Department of Physics, University of Oslo*, August 2015. https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf.

[2] Svein Linge and Hans Petter Langtangen. *Diffusion Equations*, pages 207–322. Springer International Publishing, Cham, 2017.

[3] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 810:1–124, May 2019.