TMA4212 Numerical Solution of Differential Equations by Difference Methods

Semester Project

Candidate Numbers
10019, 10048, 10064 and 10079

Department of Mathematical Sciences
Norwegian University of Science and Technology
Trondheim, Norway
April 30, 2021

Preface

This report is a result of the course *TMA4212 Numerical Solution of Differential Equations by Difference Methods* performed in the spring of 2021. The main learning outcome of this course is to gain experience with difference schemes for solving different types of partial differential equations. Moreover, a basic understanding of the finite element method is to be aquired.

An important skill in the course is to be able to choose suitable numerical solvers for elliptic, parabolic and hyperbolic partial differential equations. These solvers should be constructed, implemented and analyzed in order to verify that the solvers are implemented correctly, i.e. to verify that the constructed solvers can be used in practice. As a sidenote, equation solvers play an important role in effective implementations of these PDE-solvers. These skills are displayed in this report.

Finally, a written presentation of scientific problems and of results obtained in project work should be practiced in this course. This practice and gain of experience emerges from the work with this report.

Trondheim, April 30, 2021

Abstract

The main topic of this project report is examination of finite difference methods for solving different types of partial differential equations.

Equations that are studied are Poisson's equation in one dimension with different boundary conditions, the heat equation, inviscid Burgers' equation, the two-dimensional Laplace equation, the linearized Korteweg-deVries equation and the Sine-Gordon equation. Solvers of different orders are developed, implemented and analyzed for each of these equations.

Simulations with uniform refinement, as well as some adaptive refinement methods, show that the numerical solutions converge, in different orders, to the analytical solutions.

Contents

Chapter 1

Introduction

Part 1 of the project report consists of solving five different problems. The aim of this part is to dive deep into the material of the course and understand how the computations work through practical and theoretical experience. The problems consist of implementing finite difference methods for: (1) boundary value problems, (2) parabolic equations, (3) elliptic equations and (4) hyperbolic equations. The fifth and final problem consists of implementing the finite element method for a boundary value problem. Different boundary conditions are imposed in the different problems and the discretization for the numerical solutions vary from problem to problem. Convergence plots are constructed based on relative L_2- and ℓ_2-norms in several of the problems, in order to quantify the convergence order of the difference schemes to the analytical solution in each case.

Part 2 of the project report consists of an in-depth examination of the Sine-Gordon equation in the context of this course. The analytical solution of the equation is found and energy conservation is shown. Moreover, a semi-discretization method is constructed and used to solve the equation numerically. In order to implement this method, different Runge-Kutta and Runge-Kutta-Nyström schemes will be used to solve ODEs. Finally, a boundary value problem is solved numerically with some of the Runge-Kutta and Runge-Kutta-Nyström integrators. The results are reported and discussed.

The simulations are executed in Python, since it is a high-level language that is easy to use. Python has an abundance of helpful packages that make visualization easy, and simulations can be performed relatively efficiently. For code efficiency, sparse matrix algebra modules are used to solve linear systems.

List of Figures

Chapter 2

Mathematical Theory and Definitions

The mathematical notation and theory used throughout this report is largely gathered from Brynjulf Owren's note, which is specifically intended for this course [7]. The notation is defined, and some essential results are highlighted, in this section.

2.1 General Notation

Most of the problems are solved on either one x-axis or on both an x-axis and a t- or y-axis. When these axes are discretized into grids, the uniform step length in the x-direction will be denoted by h. Moreover, the associated number of nodes in the x-direction will usually be denoted by M. Likewise, the uniform step length in the t-direction or y-direction will be denoted by k and the associated number of nodes will usually be denoted by N.

2.2 Difference Schemes

Let $f(x)$ be a twice differentiable function. Define the following operators

$$\begin{aligned}
\Delta f(x) &= f(x+h) - f(x) && \text{(Forward Difference)},\\
\nabla f(x) &= f(x) - f(x-h) && \text{(Backward Difference)},\\
\delta f(x) &= f\left(x+\frac{h}{2}\right) - f\left(x-\frac{h}{2}\right) && \text{(Central Difference)},\\
\mu f(x) &= \frac{1}{2}\left(f\left(x+\frac{h}{2}\right) + f\left(x-\frac{h}{2}\right)\right) && \text{(Mean Value)}.
\end{aligned}$$

These operators can be used to approximate first order derivatives. Expanding the function $f(x)$ in a Taylor series we find

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \mathcal{O}(h^3) \Rightarrow \frac{1}{h}\Delta f(x) = f'(x) + \mathcal{O}(h).$$

A similar calculation can be done with the backward difference, which gives the same order of truncation error. A second order approximation for the first derivative can be

found by

$$f(x+h)-f(x-h)=\mathcal{O}(h^4)+f(x)+hf'(x)+\frac{h^2}{2}f''(x)+\frac{h^3}{6}f'''(x)$$
$$-\left(f(x)-hf'(x)+\frac{h^2}{2}f''(x)-\frac{h^3}{6}f'''(x)\right)=2hf'(x)+\mathcal{O}(h^3)$$
$$\Rightarrow \frac{1}{2h}(f(x+h)-f(x-h))=\frac{1}{h}\mu\delta f(x)=f'(x)+\mathcal{O}(h^2).$$

The following results are highlighted

$$f'(x)=\begin{cases}\frac{1}{h}\Delta f(x)+\mathcal{O}(h)\\ \frac{1}{h}\nabla f(x)+\mathcal{O}(h)\\ \frac{1}{h}\mu\delta f(x)+\mathcal{O}(h^2).\end{cases} \tag{2.1}$$

The same operators can be used to define difference schemes to approximate $f''(x)$. The squared operators are

$$\Delta^2 f(x)=f(x+2h)-2f(x+h)+f(x),$$
$$\nabla^2 f(x)=f(x)-2f(x-h)+f(x-2h),$$
$$\delta^2 f(x)=f(x+h)-2f(x)+f(x-h).$$

Using Taylor series for each term $f(x+2h)$ and $f(x+h)$ in the uppermost equation above, gives

$$\Delta^2 f(x)=\mathcal{O}(h^4)+f(x)+2hf'(x)+\frac{4h^2}{2}f''(x)+\frac{8h^3}{6}f'''(x)$$
$$-2\left(f(x)+hf'(x)+\frac{h^2}{2}f''(x)+\frac{h^3}{6}f'''(x)\right)+f(x)=h^2f''(x)+\mathcal{O}(h^3)$$
$$\Rightarrow \frac{1}{h^2}\Delta^2 f(x)=f''(x)+\mathcal{O}(h).$$

An analogous expansion can be done with the backward difference, which yields the same order of truncation error. The squared central difference operator leads to

$$\delta^2 f(x)=\mathcal{O}(h^5)+f(x)+hf'(x)+\frac{h^2}{2}f''(x)+\frac{h^3}{6}f^{(3)}(x)+\frac{h^4}{24}f^{(4)}(x)$$
$$-2f(x)+f(x)-hf'(x)+\frac{h^2}{2}f''(x)-\frac{h^3}{6}f^{(3)}(x)+\frac{h^4}{24}f^{(4)}$$
$$=h^2f''(x)+\mathcal{O}(h^4)\Rightarrow\frac{1}{h^2}\delta^2 f(x)=f''(x)+\mathcal{O}(h^2).$$

The results are highlighted here

$$f''(x)=\begin{cases}\frac{1}{h^2}\Delta^2 f(x)+\mathcal{O}(h)\\ \frac{1}{h^2}\nabla^2 f(x)+\mathcal{O}(h)\\ \frac{1}{h^2}\delta^2 f(x)+\mathcal{O}(h^2).\end{cases} \tag{2.2}$$

2.3 Error Measures

In the following, bold characters denote vectors. The discrete ℓ_2-norm and the continuous L_2-norm are defined as

$$\|\boldsymbol{V}\|_2 := \sqrt{\frac{1}{N}\sum_{i=1}^{N} V_i^2}, \tag{2.3}$$

$$\|v(x)\|_2 := \sqrt{\int_\Omega v^2(x)\mathrm{d}\Omega}, \tag{2.4}$$

respectively, where $\boldsymbol{V} \in \mathbb{R}^N$ and the function $v \in L_2(\Omega)$. Furthermore, the relative errors e_ℓ^r and $e_{L_2}^r$ are defined as

$$e_\ell^r := \frac{\|\boldsymbol{u} - \boldsymbol{U}\|_2}{\|\boldsymbol{u}\|_2}, \tag{2.5}$$

$$e_{L_2}^r := \frac{\|u(x) - U(x)\|_2}{\|u(x)\|_2}, \tag{2.6}$$

respectively, where $u(x)$ and $\boldsymbol{u}$ denote analytical solutions, while $U(x)$ and $\boldsymbol{U}$ denote numerical solutions.

Some implementation details in Python regarding the error measures are given. The continuous L_2-norm was implemented by interpolating between each grid point with cubic polynomials and using numerical quadrature to integrate the difference between the analytical and the interpolated numerical solution. More precisely, this was implemented in Python by means of *scipy.interpolate.interp1d*, which was used to construct cubic interpolation polynomials, and *scipy.integrate.quad*, which was used as the numerical quadrature.

2.4 Adaptive Mesh Refinement

In some problems, we will consider grid refinement-approaches where the sub-intervals of a grid are split, i.e. a grid point is added in the middle of the subinterval, only if they satisfy some requirement. These approaches are referred to as Adaptive Mesh Refinement (AMR). Two such approaches will be considered. Firstly, the *average*-error will be used, where at each refinement step the intervals satisfying the inequality

$$\|u - u_h\|_{L_2(I_i)} > \alpha N^{-1}\|u - u_h\|_{L_2(I)},$$

will be split. In the above equation, u denotes the exact solution, u_h denotes the numerical solution, I is the whole interval, I_i is the i'th subinterval, N is the total number of sub-intervals and $\alpha \in \mathbb{R}$ is a constant. Secondly, the *max*-error will be used, where at each refinement step the intervals satisfying the inequality

$$\|u - u_h\|_{L_2(I_i)} > \alpha \max_i \|u - u_h\|_{L_2(I_i)},$$

will be split.

Chapter 3

Part 1

3.1 Problem 1 - Poisson Equation in One Dimension

In this problem, a function $u(x)$, defined on the unit interval $[0, 1]$, is considered. The Poisson equation with a Neumann boundary condition

$$u_{xx} = f(x),\ u(0) = \alpha,\ u_x(1) = \sigma, \tag{3.1}$$

will be solved analytically. Moreover, it will be solved numerically on a grid of equidistant points

$$x_0 = 0,\ x_1 = \frac{1}{M+1},\ \ldots,\ x_M = \frac{M}{M+1},\ x_{M+1} = 1.$$

A node on the x-axis will be denoted by $x_m = mh$ where $0 \leq m \leq M+1$ and the step length is $h = \frac{1}{M+1}$.

a)

Let $f(x) := \cos(2\pi x) + x$ and $\alpha = \sigma = 0$. The analytical solution to (3.1), denoted by $u(x)$, is found by integrating twice, before applying the boundary conditions

$$\begin{aligned}
u_{xx} &= \cos(2\pi x) + x, \\
u_x(x) &= \frac{1}{2\pi}\sin(2\pi x) + \frac{1}{2}x^2 + C_1, \\
u_x(1) &= \frac{1}{2} + C_1 = 0 \implies C_1 = -\frac{1}{2}, \\
u(x) &= -\frac{1}{4\pi^2}\cos(2\pi x) + \frac{1}{6}x^3 + C_1 x + C_2, \\
u(0) &= -\frac{1}{4\pi^2} + C_2 = 0 \implies C_2 = \frac{1}{4\pi^2}, \\
\implies u(x) &= -\frac{1}{4\pi^2}\cos(2\pi x) + \frac{1}{6}x^3 - \frac{1}{2}x + \frac{1}{4\pi^2}.
\end{aligned}$$

In order to approximate the analytical solution numerically, the linear system $A_h\boldsymbol{U} = \boldsymbol{f}$, with

$$A_h = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ddots & 1 & -2 & 1 \\ 0 & \dots & \frac{h}{2} & -2h & \frac{3h}{2} \end{pmatrix}, \boldsymbol{U} = \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_M \\ U_{M+1} \end{pmatrix}, \boldsymbol{f} = \begin{pmatrix} f(x_1) - \alpha/h^2 \\ f(x_2) \\ \vdots \\ f(x_M) \\ \sigma \end{pmatrix},$$

is constructed. Note that the numerical solution in each grid point x_m is denoted by U_m. The central difference approximation

$$u_m'' = \frac{1}{h^2}\delta^2 u_m = \frac{1}{h^2}(u_{m-1} - 2u_m + u_{m+1}) + \mathcal{O}(h^2) = f_m \quad 1 \leq m \leq M, \tag{3.2}$$

where $u_m := u(x_m)$ and $f_m := f(x_m)$, is used for all internal points on the grid. The truncation error for the central difference approximation is justified in section 2.2. For $x = 1$, the approximation used is

$$\frac{\frac{1}{2}u(x-2h) - 2u(x-h) + \frac{3}{2}u(x)}{h} = u'(x) + \mathcal{O}(h^2). \tag{3.3}$$

It can be verified that this is a second order approximation of the first derivative by inserting the Taylor expansion up to third order for each term

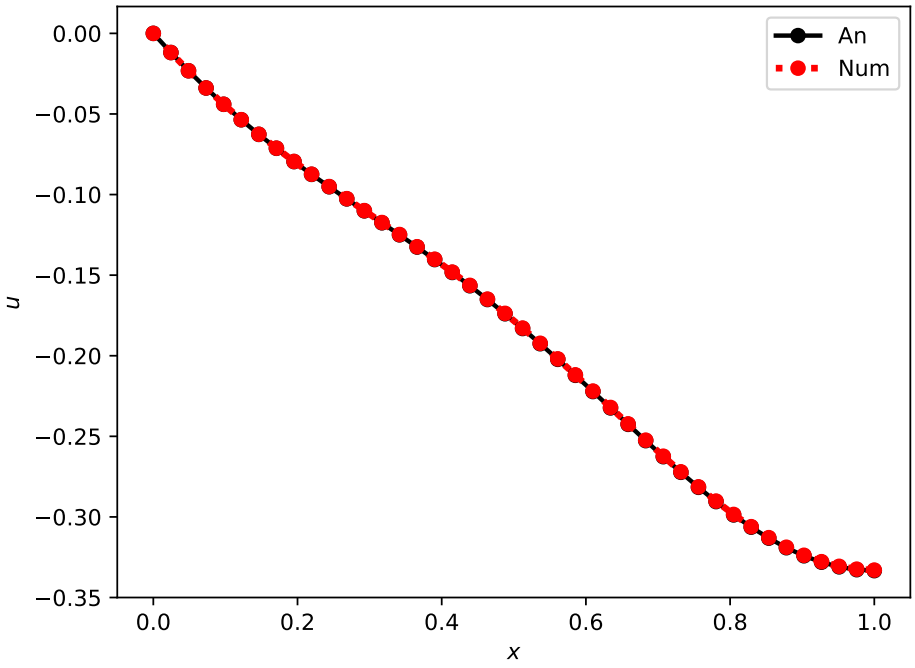
$$\begin{aligned} &\frac{1}{2h}\left\{u(x) - 2hu'(x) + 2h^2u''(x) - \frac{4h^3}{3}u^{(3)}(x)\right. \\ &\quad \left. -4\left(u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u^{(3)}(x)\right) + 3u(x) + \mathcal{O}(h^4)\right\} \\ &= \frac{1}{2h}\left\{2hu'(x) - \frac{2h^3}{3}u^{(3)}(x)\right\} = u'(x) + \mathcal{O}(h^2). \end{aligned}$$

Inserting the numerical solution U_{M+1} at $x_{M+1} = 1$ and neglecting the truncation error $\mathcal{O}(h^2)$ gives

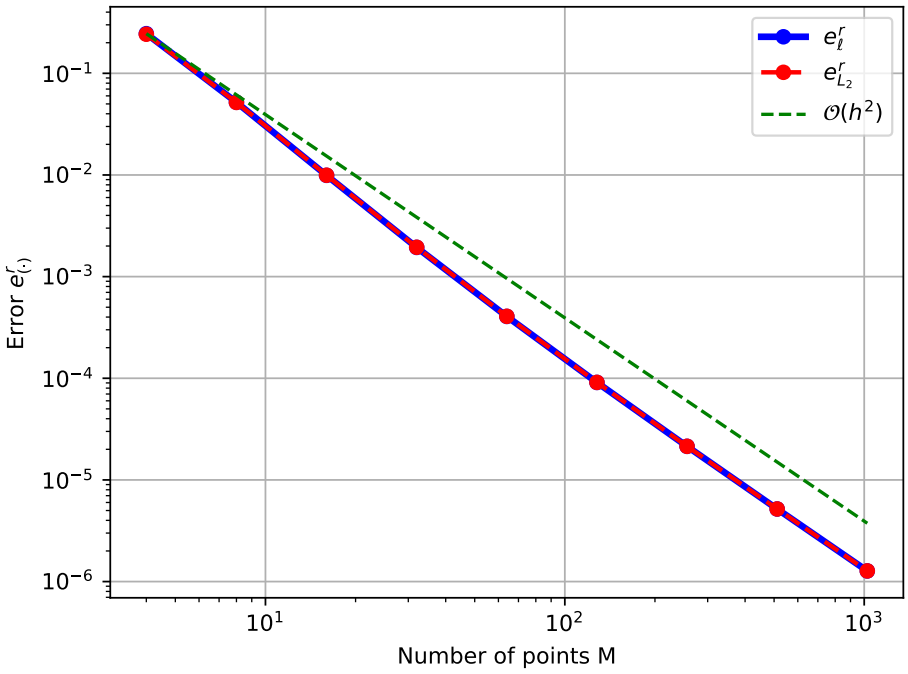
$$\frac{\frac{1}{2}U_{M-1} - 2U_M + \frac{3}{2}U_{M+1}}{h} = \sigma,$$

which coincides with the last row in the linear system above. This system is solved numerically via a linear equation solver in Python, which gives the approximate solution. Both the analytical and the numerical solutions are plotted in figure 3.1a.

In order to quantify the convergence of the numerical solution to the analytical solution, the relative errors e_ℓ^r and $e_{L_2}^r$, as defined in section 2.3, are computed and plotted. A "log-log" plot of the relative errors in terms of increasing M (decreasing $h \to 0$) are shown in figure 3.1b, i.e. log-scales are used on both axes. The relative errors in both norms look very similar, and they show a convergence of order 2. This is as expected.



(a) Analytical and numerical solution



(b) Relative error

Figure 3.1: Poisson equation with $f(x) := \cos(2\pi x) + x$, and boundary conditions $u(0) = 0$ and $u_x(1) = 0$. The analytical and numerical solution, solved and plotted on a grid with $M = 40$ points, is shown in (a). In (b), a "log-log" plot of relative errors when $h \to 0$ is shown. $e^r_{L_2}$ is plotted with a dotted red line and e^r_ℓ is shown in blue. A green line of order $\mathcal{O}(h^2)$ is added to confirm the convergence order of the method.

b)

A similar analysis is performed, but the boundary conditions are changed to two Dirichlet conditions

$$u(0) = 1,\, u(1) = 1.$$

In order to calculate the new analytical solution, a similar procedure as in part **a)** is followed. Thus, the analytical solution is given by

$$\begin{aligned}
u_{xx} &= \cos(2\pi x) + x, \\
u_x(x) &= \frac{1}{2\pi}\sin(2\pi x) + \frac{1}{2}x^2 + C_1, \\
u(x) &= -\frac{1}{4\pi^2}\cos(2\pi x) + \frac{1}{6}x^3 + C_1 x + C_2, \\
u(0) &= -\frac{1}{4\pi^2} + C_2 = 1 \implies C_2 = 1 + \frac{1}{4\pi^2}, \\
u(1) &= -\frac{1}{4\pi^2} + \frac{1}{6} + C_1 + 1 + \frac{1}{4\pi^2} = 1 \implies C_1 = -\frac{1}{6}, \\
\implies u(x) &= -\frac{1}{4\pi^2}\cos(2\pi x) + \frac{1}{6}x^3 - \frac{1}{6}x + 1 + \frac{1}{4\pi^2}.
\end{aligned}$$

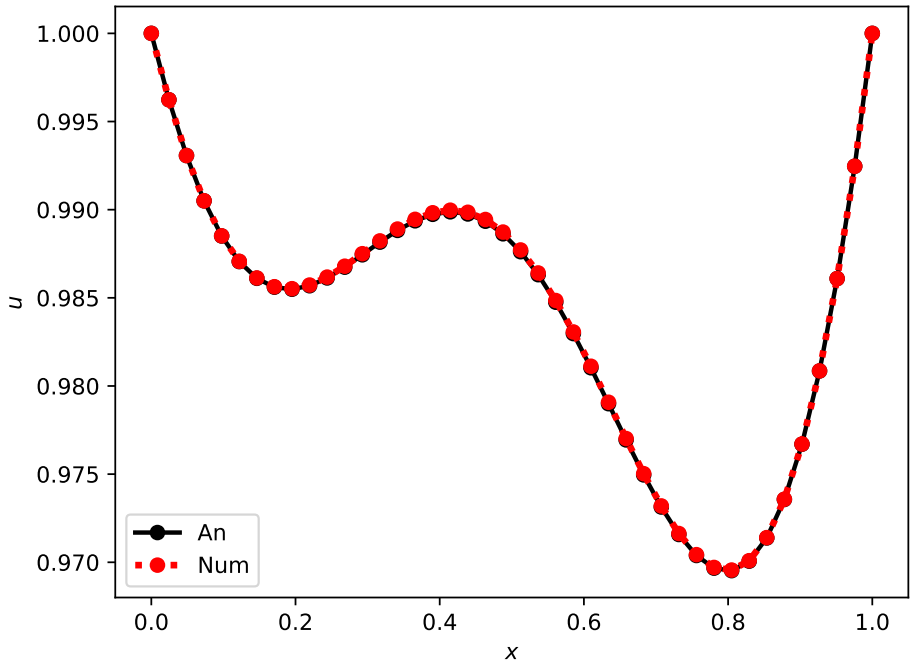
The linear system $A_h \boldsymbol{U} = \boldsymbol{f}$ is constructed, with

$$A_h = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ddots & 1 & -2 & 1 \\ 0 & \ddots & \ddots & 1 & -2 \end{pmatrix}, \boldsymbol{U} = \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_M \end{pmatrix}, \boldsymbol{f} = \begin{pmatrix} f(x_1) - \frac{1}{h^2} \\ f(x_2) \\ \vdots \\ f(x_M) - \frac{1}{h^2} \end{pmatrix}.$$

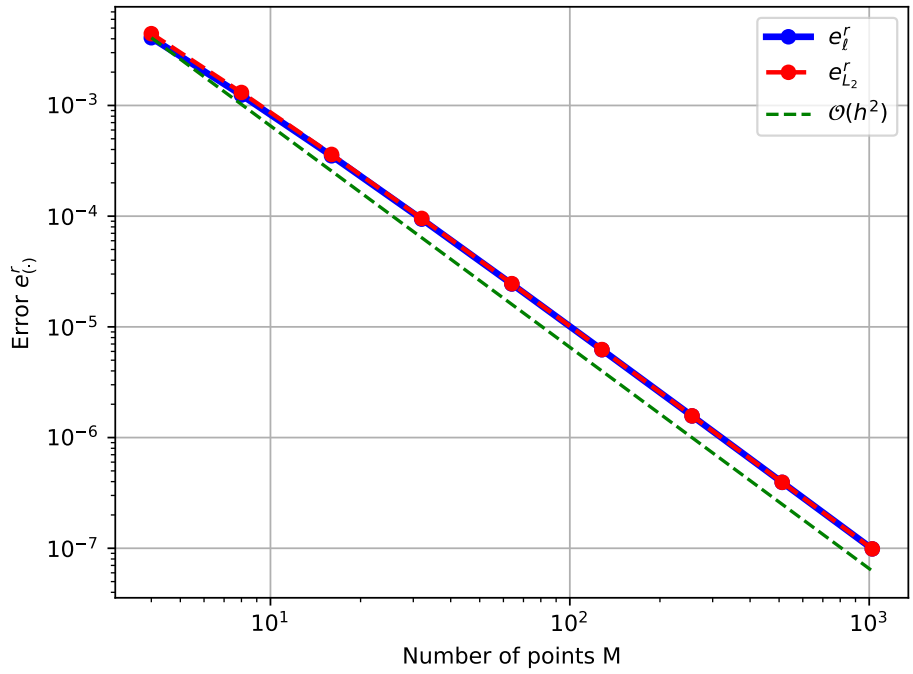
In this case, the central difference approximation (3.2) is used for all points on the grid, except for the end points, where the function values are know. The analytical solution, as well as the numerical solution, is plotted in figure 3.2a. Second order convergence is expected, because the central finite difference scheme has a local truncation error of order $\mathcal{O}(h^2)$. The "log-log" plot, shown in figure 3.2b, confirms that the convergence is of second order.

c)

The same problem is considered, but with two Neumann boundary conditions $u_x(0) = 0$, $u_x(1) = \frac{1}{2}$ instead. The issue with this specification is that the two Neumann conditions are the only conditions imposed, so the solution of the equation is ambiguous. The equation has either infinitely many solutions or zero solutions. This can be readily seen when solving the equation analytically, since there is one extra constant, C_2, that cannot be determined without imposing more conditions. A remedy could be to add a Dirichlet condition in at least one of the endpoints. In this way, one unambiguous solution can be determined down to the constant C_2. If this is done, the problem becomes very similar to the problem of task **a)**, and the numerical solution can be found in a similar manner.



(a) Analytical and numerical solution



(b) Relative error

Figure 3.2: Poisson equation with $f(x) := \cos(2\pi x) + x$, and boundary conditions $u(0) = 1$ and $u(1) = 1$. The analytical and numerical solution, solved and plotted on a grid with $M = 40$ points, is shown in (a). In (b), a "log-log" plot of relative errors when $h \to 0$ is shown. $e^r_{L_2}$ is plotted with a dotted red line and e^r_ℓ is shown in blue. A green line of order $\mathcal{O}(h^2)$ is added to confirm the convergence order of the method.

Another way to notice that the system is ambiguous is to consider a finite difference scheme with two fictitious nodes. A second order discretization of the Neumann boundary conditions using two fictitious external nodes x_{-1} and x_{M+2} is given by

$$\frac{U_1 - U_{-1}}{2h} = 0, \quad \frac{U_{M+2} - U_M}{2h} = \frac{1}{2}.$$

These approximate boundary conditions lead to the elimination of the fictitious nodes $U_{-1} = U_1$ and $U_{M+2} = h + U_M$. Combining this with the second order central difference approximation (3.2), where the numerical solutions U_m are inserted and the truncation error neglected, gives the equation for $m = 0$

$$\frac{U_1 - U_0}{h} = \frac{h}{2} f(x_0)$$

and the equation for $m = M + 1$

$$\frac{U_M - U_{M+1}}{h} = \frac{h}{2} f(x_{M+1}) - \frac{1}{2}$$

Hence, the linear system takes the form

$$A_h = \frac{1}{h^2} \begin{pmatrix} -h & h & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ddots & 1 & -2 & 1 \\ 0 & \dots & \dots & h & -h \end{pmatrix}, \boldsymbol{U} = \begin{pmatrix} U_0 \\ U_1 \\ \vdots \\ U_{M+1} \end{pmatrix}, \boldsymbol{f} = \begin{pmatrix} \frac{h}{2} f(x_0) \\ f(x_1) \\ \vdots \\ \frac{h}{2} f(x_{M+1}) - \frac{1}{2} \end{pmatrix}.$$

It is apparent that the matrix A_h is singular. Finally, the conclusion is that it cannot be solved.

d)

In this problem, the function $u(x) = \exp\left(-\frac{1}{\epsilon}(x - \frac{1}{2})^2\right)$ will be used as a manufactured solution for the boundary value problem

$$u_{xx} = f(x) \text{ in } \Omega = (0, 1), \tag{3.4}$$

with Dirichlet boundary conditions $u(0) = u(1) = \exp\left(-\frac{1}{4\epsilon}\right) = \beta$. The constant ϵ determines the "steepness" of the curve, where the value $\epsilon = 0.01$ is used in our implementation. The function $f(x)$ can be calculated analytically from the manufactured solution

$$\begin{aligned} f(x) &= \frac{\mathrm{d}^2}{\mathrm{d}x^2} \left(\exp\left(-\frac{1}{\epsilon} \left(x - \frac{1}{2} \right)^2 \right) \right) \\ &= \frac{\mathrm{d}}{\mathrm{d}x} \left(-\frac{2}{\epsilon} \left(x - \frac{1}{2} \right) \exp\left(-\frac{1}{\epsilon} \left(x - \frac{1}{2} \right)^2 \right) \right) \\ &= -\frac{2}{\epsilon} \exp\left(-\frac{1}{\epsilon} \left(x - \frac{1}{2} \right)^2 \right) + \frac{2}{\epsilon} \left(x - \frac{1}{2} \right) \frac{2}{\epsilon} \left(x - \frac{1}{2} \right) \exp\left(-\frac{1}{\epsilon} \left(x - \frac{1}{2} \right)^2 \right) \\ &= \frac{2}{\epsilon^2} \exp\left(-\frac{1}{\epsilon} \left(x - \frac{1}{2} \right)^2 \right) \left(2 \left(x - \frac{1}{2} \right)^2 - \epsilon \right). \end{aligned}$$

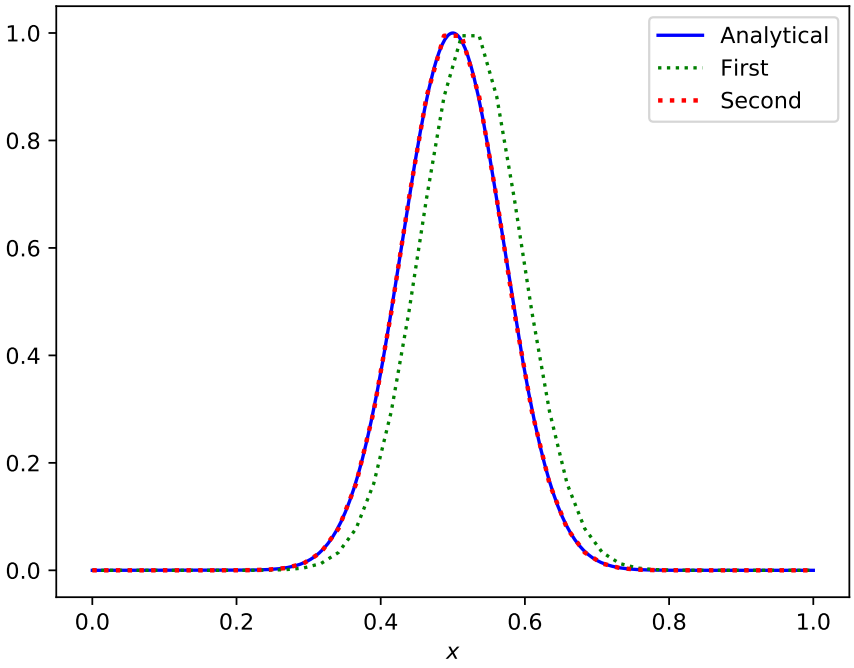


Figure 3.3: Poisson equation with manufactured solution $u(x) = \exp\left(-\frac{1}{\epsilon}(x-\frac{1}{2})^2\right)$, with Dirichlet boundary conditions $u(0) = u(1) = \exp\left(-\frac{1}{4\epsilon}\right)$. The manufactured solution, and numerical solutions solved with **UMR**, are plotted on a grid with $M = 40$ points. "First" refers to the solution using the first order method, while "Second" refers to the solution using the second order method.

How the numerical solutions converges to the manufactured solution is investigated for both second and first order methods using uniform mesh refinement (UMR) and adaptive mesh refinement (AMR).

UMR will be used first, starting with the **first order method**. As seen from equation (2.2), applying the forward difference operator twice gives a first order approximation to the second derivative

$$\begin{aligned} u_{xx}(x_m) &= \frac{1}{h^2}\Delta^2 u(x_m) + \mathcal{O}(h) \\ \Rightarrow f(x_m) &= \frac{1}{h^2}(U_m - 2U_{m+1} + U_{m+2}),\ 0 \leq m \leq M-1. \end{aligned}$$

The implication follows from inserting the numerical approximation U_m at each grid point x_m and neglecting the truncation error $\mathcal{O}(h)$. Adding the boundary conditions $U_0 = U_{M+1} = \beta$ gives the linear system $A_h \boldsymbol{U} = \boldsymbol{f}$, with

$$A_h = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ddots & 1 & -2 & 1 \\ 0 & \dots & \dots & 1 & -2 \end{pmatrix}, \boldsymbol{U} = \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_M \end{pmatrix}, \boldsymbol{f} = \begin{pmatrix} f(x_0) - \beta/h^2 \\ f(x_1) \\ \vdots \\ f(x_{M-1}) - \beta/h^2 \end{pmatrix}.$$

The **second order method** is

$$A_h = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ddots & 1 & -2 & 1 \\ 0 & \dots & \dots & 1 & -2 \end{pmatrix}, \boldsymbol{U} = \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_M \end{pmatrix}, \boldsymbol{f} = \begin{pmatrix} f(x_1) - \beta/h^2 \\ f(x_2) \\ \vdots \\ f(x_M) - \beta/h^2 \end{pmatrix},$$

where a central difference scheme, as in equation (3.2), is used. The manufactured solution and numerical solutions when using both first and second order methods with UMR are shown in figure 3.3. A "log-log" plot of the relative errors when using both first and second order methods with UMR are shown in figure 3.4a. It is apparent that the convergence orders obtained in practice match the expected theoretical convergence orders for both schemes. These expectations are based on the $\mathcal{O}(h)$ truncation error of the first order method and the $\mathcal{O}(h^2)$ truncation error of the second order method. Figure 3.4b depicts how the distribution of the error in the numerical solution evolves throughout the uniform refinement.

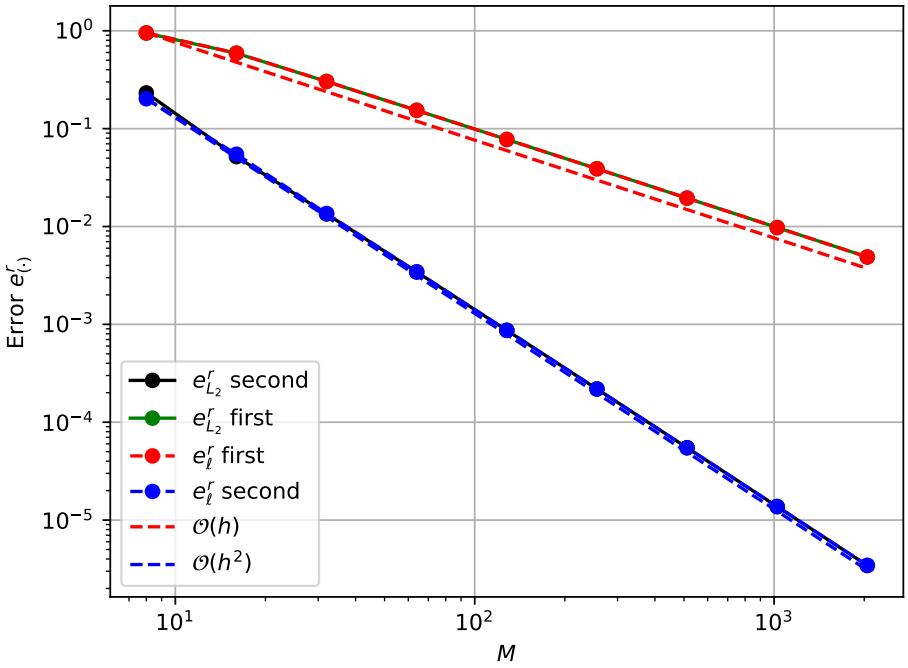
In order to solve the boundary value problem using AMR, first and second order methods where the coefficients are dependent on the local step size need to be constructed. Liu et al. (1995) present stencils for methods of both orders in the article *A high-resolution finite-difference scheme for nonuniform grids* [4]. These are shown below. The implementation of AMR uses the criteria for splitting given in section 2.4, with $\alpha = 1$ for the average-error, and $\alpha = 0.7$ for the max-error. There is one exception from what is given in section 2.4: when the average-error is calculated, the difference in L_2-norm is not calculated over the entire interval, I, but on every sub-interval, before finding the average. This means that $N^{-1}\sum_i \|u - u_h\|_{L_2(I_i)}$ is used as the average-error in this problem. If, on the contrary, $N^{-1}\|u - u_h\|_{L_2(I)}$ is used, our implementation essentially yields uniform refinement.

A **first order method** with AMR is given by the three point stencil [4]

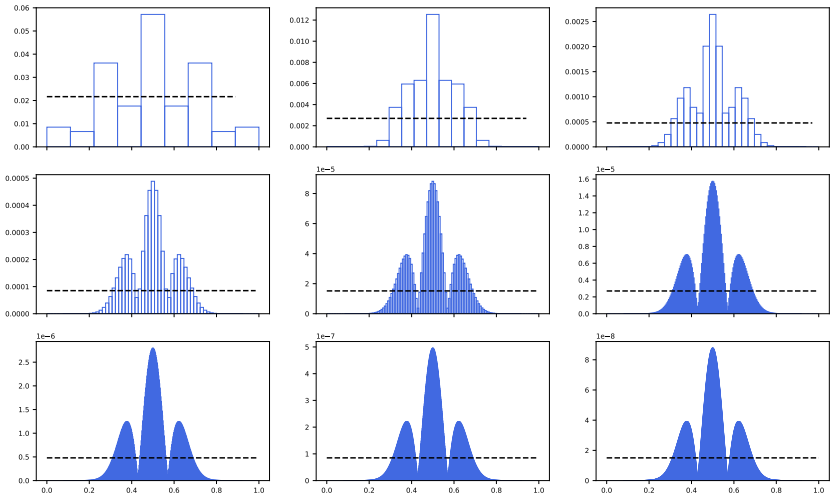
$$(U_{xx})_m = b_m U_{m-1} - (b_m + c_m)U_m + c_m U_{m+1},$$

where the coefficients are given as

$$b_m = \frac{2}{h_{m-1}(h_{m-1} + h_m)},$$
$$c_m = \frac{2}{h_m(h_{m-1} + h_m)},$$
$$h_m = x_{m+1} - x_m.$$



(a)



(b)

Figure 3.4: Poisson equation with manufactured solution $u(x) = \exp\left(-\frac{1}{\epsilon}(x - \frac{1}{2})^2\right)$, with Dirichlet boundary conditions $u(0) = u(1) = \exp\left(-\frac{1}{4\epsilon}\right)$. (a) A "log-log" plot of relative errors when $h \to 0$, when solving the problem using **UMR**, is shown. First and second order functions are added in order to compare to the convergence graphs of the numerical solutions. (b) A bar plot of the error associated with the second order solution on each sub-interval is shown. The dashed line indicates the average error.

With the Dirichlet boundary conditions, the system of equations becomes

$$A_h = \begin{pmatrix} -(b_1+c_1) & c_1 & 0 & \dots & 0 \\ b_2 & -(b_2+c_2) & c_2 & \ddots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ddots & b_{M-1} & -(b_{M-1}+c_{M-1}) & c_{M-1} \\ 0 & \dots & \dots & b_M & -(b_M+c_M) \end{pmatrix},$$

$$\boldsymbol{U} = \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_M \end{pmatrix}, \boldsymbol{f} = \begin{pmatrix} f(x_1) - b_1\beta \\ f(x_2) \\ \vdots \\ f(x_M) - c_M\beta \end{pmatrix}.$$

A **second order method** with AMR is given by the four point stencil [4]

$$(U_{xx})_m = a_m U_{m-2} + b_m U_{m-1} - (a_m + b_m + c_m)U_m + c_m U_{m+1} \tag{3.5}$$

where the coefficients are defined as

$$\begin{aligned} a_m &= \frac{2(d_{m+1} - d_{m-1})}{d_{m-2}(d_{m-2} + d_{m+1})(d_{m-2} - d_{m-1})} \\ b_m &= \frac{2(d_{m-2} - d_{m+1})}{d_{m-1}(d_{m-2} - d_{m-1})(d_{m-1} + d_{m+1})} \\ c_m &= \frac{2(d_{m-2} + d_{m-1})}{d_{m+1}(d_{m-1} + d_{m+1})(d_{m-2} + d_{m+1})} \\ d_{m+1} &= h_m, \;\; d_{m-1} = h_{m-1}, \;\; d_{m-2} = h_{m-2} + h_{m-1} \\ h_m &= x_{m+1} - x_m. \end{aligned}$$

By including the boundary conditions, the complete difference scheme becomes

$$A_h = \begin{pmatrix} -(a_1+b_1+c_1) & c_1 & 0 & \dots & \dots & 0 \\ b_2 & -(a_2+b_2+c_2) & c_2 & \ddots & \ddots & 0 \\ a_3 & b_3 & -(a_3+b_3+c_3) & c_3 & \ddots & \vdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & a_M & b_M & -(a_M+b_M+c_M) \end{pmatrix},$$

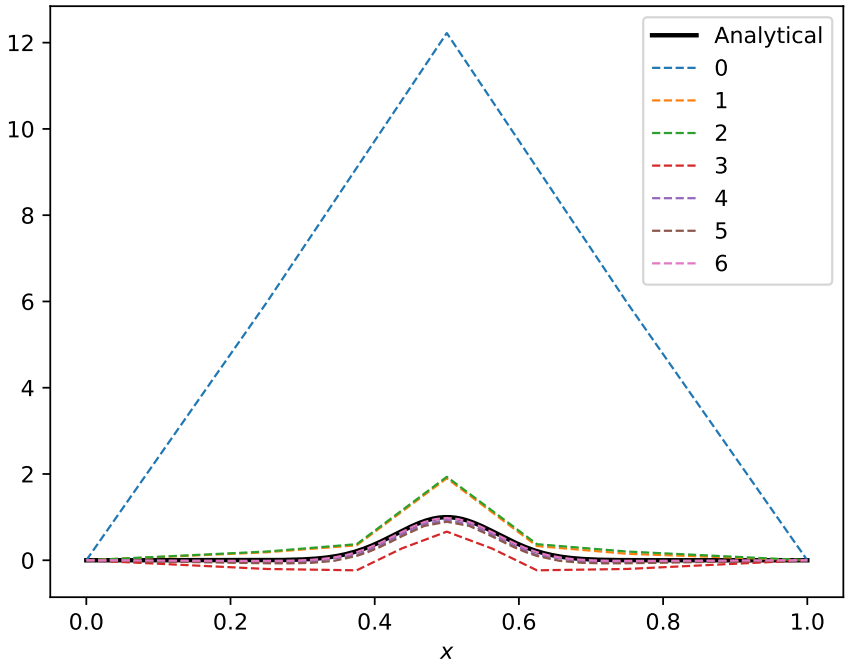
$$\boldsymbol{U} = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_M \end{pmatrix}, \quad \boldsymbol{f} = \begin{pmatrix} f(x_1) - b_1\beta \\ f(x_2) - a_2\beta \\ f(x_3) \\ \vdots \\ f(x_M) - c_M\beta \end{pmatrix}.$$

We remark that for $m = 1$ in equation (3.5), the fictitious node U_{-1} becomes an unknown in the linear system. To fix this, we immediately set $a_1 = 0$, by ensuring that $h_0 = h_1$ for the first two intervals. Hence, the fictitious node U_{-1} is eliminated from the equation and the first equation in the linear system yields $(U_{xx})_1 = \frac{1}{h_0^2}(U_0 - 2U_1 + U_2)$. This equation also has a convergence order of $\mathcal{O}(h^2)$, because of the central difference approximation, so that the order in this AMR method is conserved.

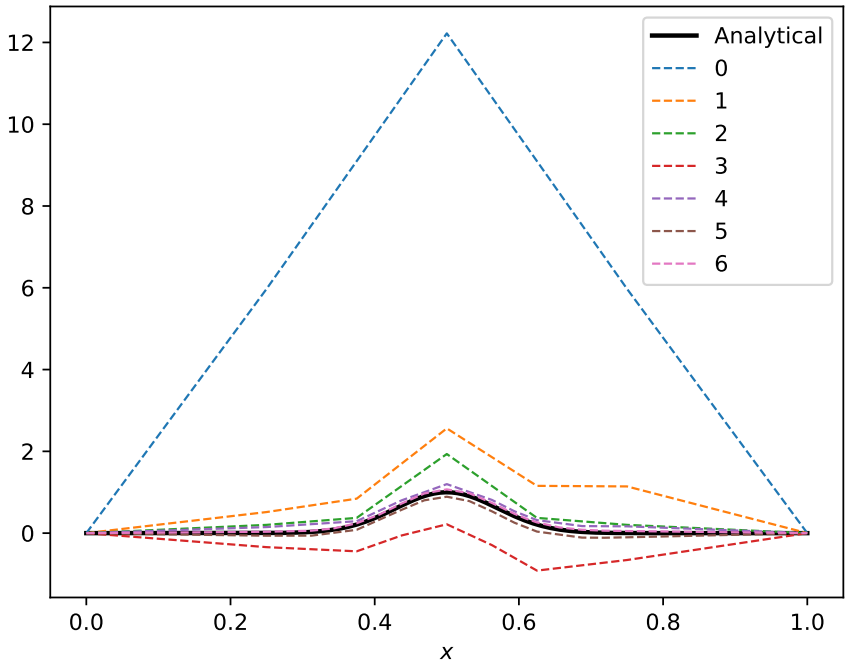
In figure 3.5, the manufactured solution is plotted together with the numerical solutions using max-error AMR, with both the first and second order methods. The starting discretization of the grid in the unit interval has $M = 3$. It can be observed that the first order method appears to refine the grid almost uniformly, while for the second order method one can clearly see that the grid is non-uniform. This could indicate that the splitting criterion used in AMR is not suited for the first order scheme.

Figure 3.6a shows a "log-log" plot of the discrete and continuous relative errors with average-error AMR. The second order scheme has an approximate convergence rate of $\mathcal{O}(h^2)$, which is expected. Unexpectedly, the first order scheme also yields second order convergence. An explanation for this is that the splitting criterion used in the adaptive refinement essentially results in uniform refinement for this method. Figure 3.6b depicts how the error distribution of the numerical solution evolves for each refinement step for the second order scheme.

Furthermore, figure 3.7a presents a "log-log" plot of the discrete and continuous relative errors with max-error AMR. As for the average-error AMR, it is observed that both schemes give second order convergence. Figure 3.7b depicts how the error distribution of the numerical solution evolves for each refinement step for the second order scheme.

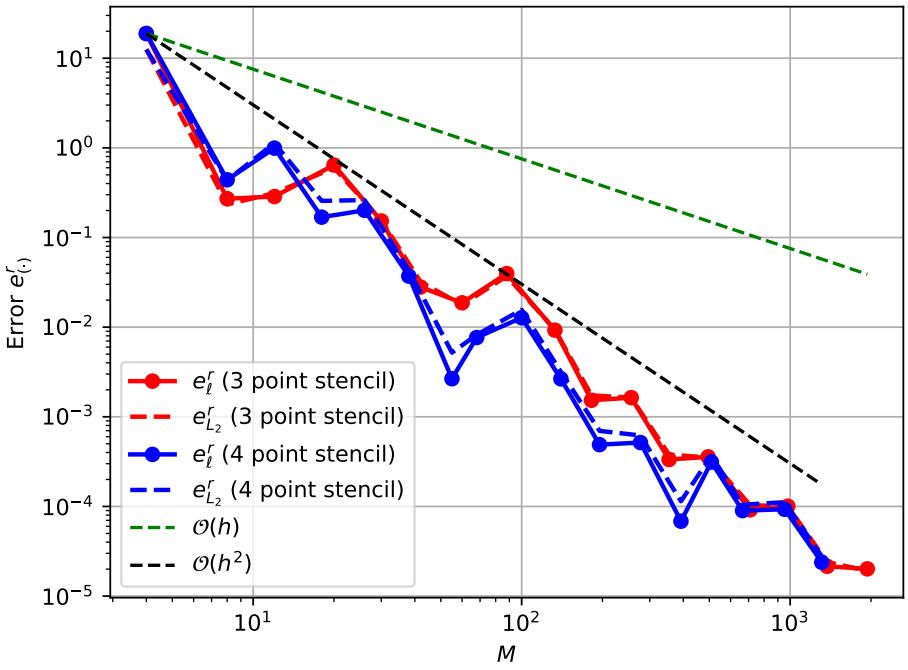


(a) **AMR first order**

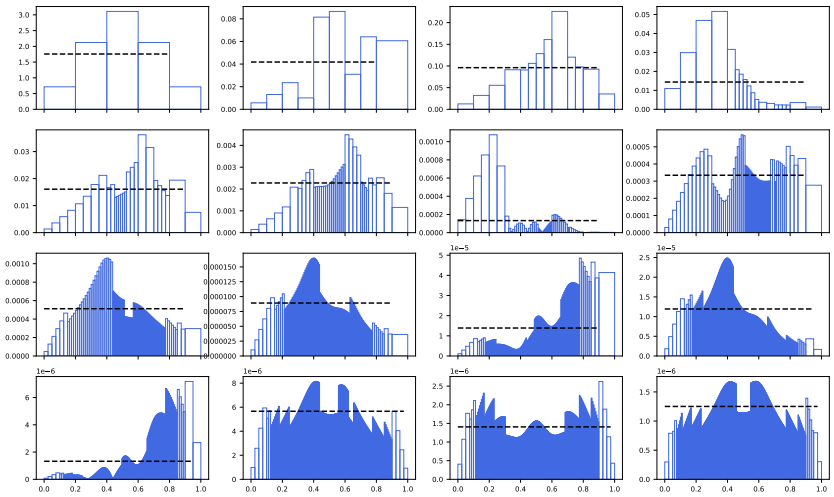


(b) **AMR second order**

Figure 3.5: Poisson equation with manufactured solution $u(x) = \exp\left(-\frac{1}{\epsilon}(x - \frac{1}{2})^2\right)$, with Dirichlet boundary conditions $u(0) = u(1) = \exp\left(-\frac{1}{4\epsilon}\right)$. The numerical solution is computed with max-error **AMR**, combined with a **first order method** (a) and **second order method** (b). The integers in the legend refer to the number of iterations in the refinement, i.e. 6 refers to a solution where the grid has been adaptively refined 6 times. The initial grid has $M = 3$ points.

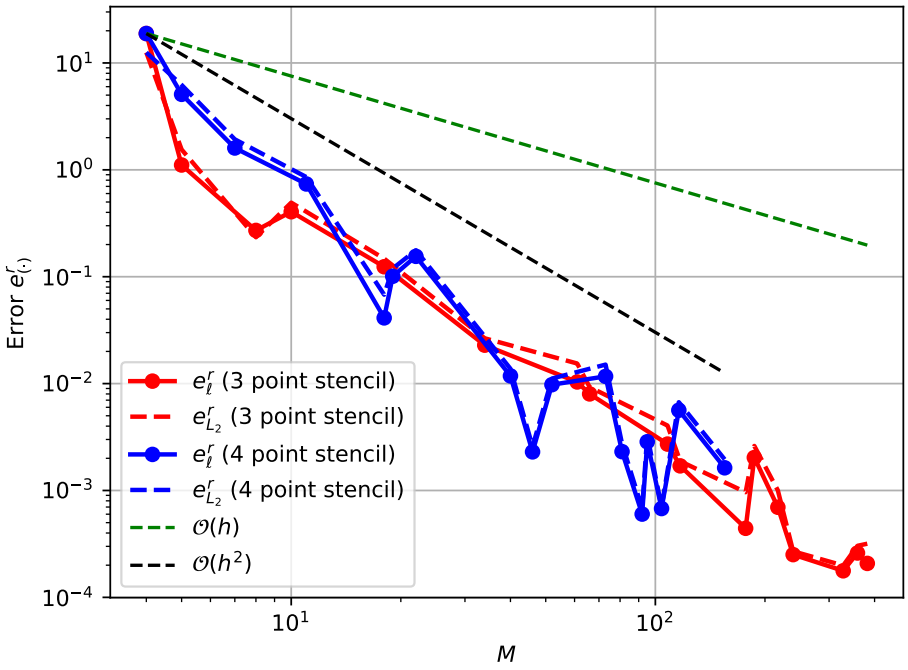


(a)

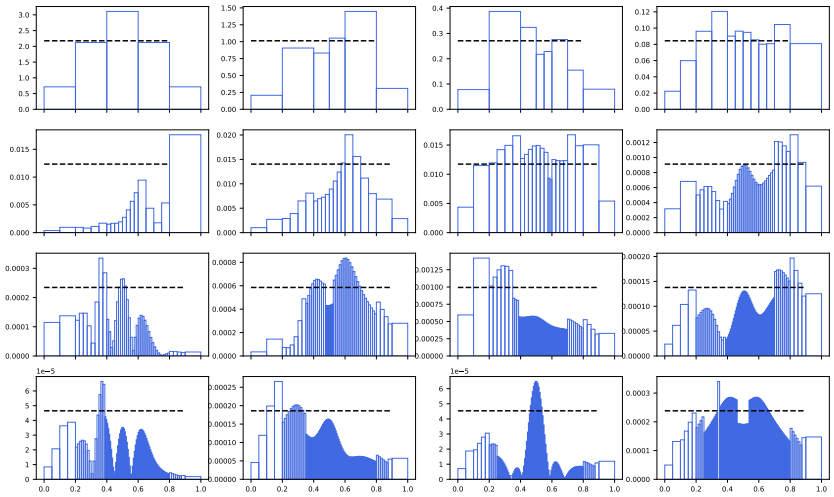


(b)

Figure 3.6: Poisson equation with manufactured solution $u(x) = \exp\left(-\frac{1}{\epsilon}(x-\frac{1}{2})^2\right)$, with Dirichlet boundary conditions $u(0) = u(1) = \exp\left(-\frac{1}{4\epsilon}\right)$. (a) "log-log" plot of relative errors when $h \to 0$, when solving the problem using average-error **AMR**. First and second order functions of h are added in order to compare the convergence of the numerical solution to the theoretical results. (b) Bar plot of the error associated with each sub-interval for the second order scheme. The dashed line indicates the tolerance; sub-intervals with error larger than the tolerance are refined.



(a)



(b)

Figure 3.7: Poisson equation with manufactured solution $u(x) = \exp\left(-\frac{1}{\epsilon}(x-\frac{1}{2})^2\right)$, with Dirichlet boundary conditions $u(0) = u(1) = \exp\left(-\frac{1}{4\epsilon}\right)$. (a) "log-log" plot of relative errors when $h \to 0$, when solving the problem using max-error **AMR**. First and second order functions of h are added in order to compare the convergence of the numerical solution to the theoretical results. (b) Bar plot of the error associated with each sub-interval for the second order scheme. The dashed line indicates the tolerance; sub-intervals with error larger than the tolerance are refined.

3.2 Problem 2 - Heat and Inviscid Burgers' Equations

a)

The heat equation

$$u_t = u_{xx},\ u_x(0,t) = u_x(1,t) = 0,\ u(x,0) = 2\pi x - \sin(2\pi x), \tag{3.6}$$

on $x \in [0,1], t > 0$, with Neumann boundary conditions, will be studied in this problem. An equidistant grid of points in the x-direction

$$x_0 = 0,\ x_1 = \frac{1}{M+1},\ \ldots,\ x_M = \frac{M}{M+1},\ x_{M+1} = 1,$$

will be utilized when computing the numerical solution. Semi-discretization will be used to solve the problem numerically; the PDE is discretized in the x-direction, before the resulting ODEs are solved with first and second order methods in time. Let $v_m(t) \approx u(x_m, t)$ for $0 \leq m \leq M+1$, i.e. the numerical solution along the line (x_m, t) is denoted by $v_m(t)$. For ease of notation, the t-dependency may not be explicitly included in the following, i.e. let $v_m := v_m(t)$. Furthermore, let

$$\dot{v}_m = \frac{1}{h^2}\delta^2 v_m = \frac{1}{h^2}(v_{m-1} - 2v_m + v_{m+1}) \quad 1 \leq m \leq M, \tag{3.7}$$

where $\dot{v}_m = \frac{\mathrm{d}v_m(t)}{\mathrm{d}t}$ and $h = \frac{1}{M+1}$. The boundary conditions will be discretized with both first and second order methods.

First Order Discretization of Boundary Conditions

First, the left and right end points are discretized with forward and backward differences, respectively. Using the above notation, let

$$\begin{aligned} \dot{v}_m &= \frac{1}{h^2}\delta^2 v_m, \quad 1 \leq m \leq M, \\ \dot{v}_0 &= \frac{1}{h^2}\Delta^2 v_0, \\ \dot{v}_{M+1} &= \frac{1}{h^2}\nabla^2 v_{M+1}. \end{aligned} \tag{3.8}$$

The boundary conditions are also discretized with forward and backward differences, which yields the following first order approximations in h

$$\begin{aligned} -\frac{v_1 - v_0}{h} &= 0, \\ \frac{v_{M+1} - v_M}{h} &= 0. \end{aligned} \tag{3.9}$$

Combining (3.8) and (3.9) gives

$$\begin{aligned} \dot{v}_0 &= \frac{1}{h^2}(v_2 - v_0), \\ \dot{v}_{M+1} &= \frac{1}{h^2}(v_{M-1} - v_{M+1}). \end{aligned}$$

Finally, a linear system of ordinary differential equations can be assembled as

$$\dot{\boldsymbol{v}} = \frac{1}{h^2} Q \boldsymbol{v},$$

where

$$Q = \begin{pmatrix} -1 & 0 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & 1 & 0 & -1 \end{pmatrix} \in \mathbb{R}^{(M+2)\times(M+2)}, \; \dot{\boldsymbol{v}} = \begin{pmatrix} \dot{v}_0 \\ \dot{v}_1 \\ \vdots \\ \dot{v}_{M+1} \end{pmatrix} \in \mathbb{R}^{M+2}.$$

Second Order Discretization of Boundary Conditions

The same notation as above is adopted in this section. In order to use central differences on the boundary conditions, the fictitious nodes $x_{-1} = -h$ and $x_{M+2} = 1 + h$ are introduced. The discretization of the boundary conditions becomes

$$-\frac{v_1 - v_{-1}}{2h} = \frac{v_{M+2} - v_M}{2h} = 0. \tag{3.10}$$

Also, let

$$\dot{v}_m = \frac{1}{h^2}(v_{m-1} - 2v_m + v_{m+1}) \quad 0 \leq m \leq M + 1. \tag{3.11}$$

Combining (3.10) with (3.11) gives $\dot{v}_0 = \frac{2}{h^2}(v_1 - v_0)$ and $\dot{v}_{M+1} = \frac{2}{h^2}(v_M - v_{M+1})$. Thus, the following system of equations

$$\dot{\boldsymbol{v}} = \frac{1}{h^2} Q \boldsymbol{v},$$

where

$$Q = \begin{pmatrix} -2 & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 \end{pmatrix},$$

is constructed.

Solution of the System of ODEs

After a discretization of the boundary conditions is chosen and the system of ODEs

$$\dot{\boldsymbol{v}} = \frac{1}{h^2} Q \boldsymbol{v},$$

is assembled, a procedure for calculating the evolution of $\boldsymbol{v}$ in time needs to be chosen. This can be done with the trapezoidal rule, which amounts to the Crank-Nicolson method (hereafter denoted by CN). The Backward Euler method (hereafter denoted by BE) can also be used. Letting $\boldsymbol{V}^0 = (u(x_0, 0), \ldots, u(x_{M+1}, 0))^T$, CN may be written as

$$\boldsymbol{V}^{n+1} = \boldsymbol{V}^n + \frac{k}{2}\left(\frac{1}{h^2}Q\boldsymbol{V}^n + \frac{1}{h^2}Q\boldsymbol{V}^{n+1}\right),$$

where $k = \frac{T}{N}$ is the step length in time and n denotes the current iteration. The system of equations is solved iteratively from $n = 0$ to $n = N - 1$ until the solution at $t_N = T$ is found. Equivalently, the method can be written as

$$(I - \frac{k}{h^2}Q)\boldsymbol{V}^{n+1} = (I + \frac{k}{h^2}Q)\boldsymbol{V}^n \quad \text{(Crank-Nicolson)}. \tag{3.12}$$

Moreover, BE can be written as

$$(I - \frac{k}{h^2}Q)\boldsymbol{V}^{n+1} = \boldsymbol{V}^n \quad \text{(Backward Euler)}. \tag{3.13}$$

The local truncation error of CN and BE with second order discretization of boundary conditions is

$$\begin{aligned} \tau_{CN} &= \mathcal{O}(k^2 + h^2), \\ \tau_{BE} &= \mathcal{O}(k + h^2), \end{aligned} \tag{3.14}$$

respectively. These truncation errors can be shown using Taylor expansions around (x_m, t_n). Inserting the exact solutions $\boldsymbol{U}^n = (u(x_0, t_n), \ldots, u(x_{M+1}, t_n))^T$ into BE gives the truncation error

$$k\tau_{BE} = (I - \frac{k}{h^2}Q)\boldsymbol{U}^{n+1} - \boldsymbol{U}^n.$$

This system can be written row-wise as

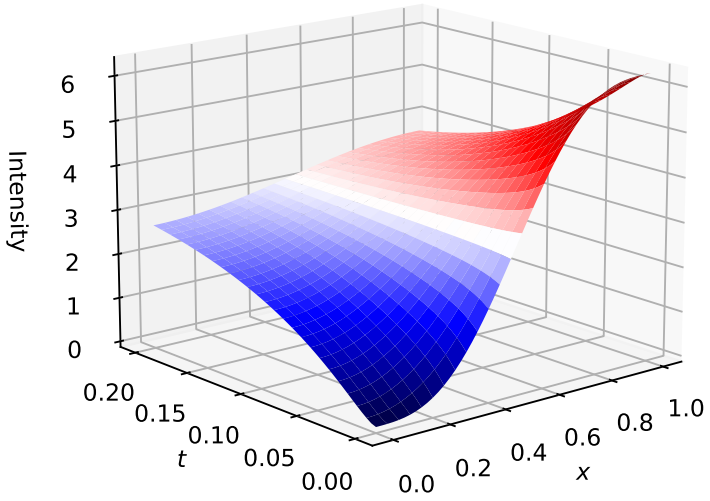
$$\begin{aligned} k\tau_{BE} &= (1 - \frac{k}{h^2}\delta_x^2)u_m^{n+1} - u_m^n, \quad 0 \le m \le M+1 \\ &= \left(1 - k(\partial_x^2 + \mathcal{O}(h^2))(1 + k\partial_t + \frac{1}{2}k^2\partial_t^2 + \mathcal{O}(k^3))\right)u_m^n - u_m^n \\ &= \left(k\partial_t + \frac{1}{2}k^2\partial_t^2 - k(\partial_x^2 + \mathcal{O}(h^2)) - k^2(\partial_x^2 + \mathcal{O}(h^2))\partial_t + \mathcal{O}(k^3)\right)u_m^n \\ &= \left(-\frac{1}{2}k^2\partial_t^2 + \mathcal{O}(kh^2) + \mathcal{O}(k^2h^2) + \mathcal{O}(k^3)\right)u_m^n \\ &= \mathcal{O}(k^2 + kh^2), \end{aligned}$$

which shows that $\tau_{BE} = \mathcal{O}(k + h^2)$. Proving the order of the local truncation error of CN is analogous, and is therefore left out [7].

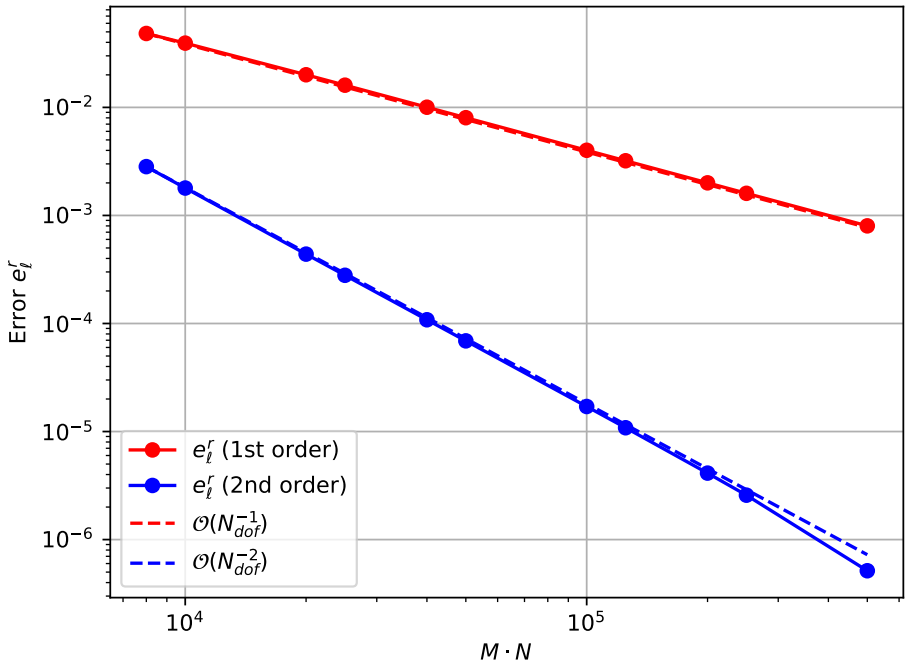
The numerical solution using CN (with second order discretization of boundary conditions) is plotted in figure 3.8a.

Convergence Plots

For this problem, the analytical solution is not available in a closed form. Thus, in order to make convergence plots, a *reference solution*, $\boldsymbol{u}_{M^*}(t)$, must be constructed. It is called a reference solution, since it is a numerical solution with a large number of points $M = M^*$ (small $h = h^*$) in the x-direction. Hence, it should be more precise than



(a) Numerical Solution



(b) Relative error with h-refinement, 1st and 2nd order discretization of BC's

Figure 3.8: Heat equation with two Neumann boundary conditions and initial condition $u(x,0) = 2\pi x - \sin(2\pi x)$ on $x \in [0,1]$ and $t \in [0,0.2]$. The numerical solution, calculated using CN with $M = N = 50$ and a second order discretization of the boundary conditions, is plotted in (a). The relative error with h-refinement for first and second order discretizations of the boundary conditions, is plotted in (b). CN was used also here with $N = 1000$. The x-axis shows the number of degrees of freedom in the linear system.

numerical solutions computed on grids with lower resolutions, which is why it is used as a replacement for the analytical solution when making convergence plots. Throughout this problem has the reference solution been calculated using $M^* = 1000$ with second order discretization of the boundary conditions, in combination with CN.

First, how the order of the boundary condition discretization impacts the convergence order with h-refinement, is investigated. CN is used, in order to minimize the error in time. The relative error e_ℓ^r, as defined in equation (2.5), is calculated when increasing the resolution in the x-direction. Remember that the analytical solution is replaced by the reference solution in the relative error. The result is depicted in figure 3.8b. It is apparent that a first order discretization of the boundary conditions results in first order convergence, even though the finite difference scheme used on the remaining grid points is of second order. Hence, use of the first order discretization of the boundary conditions will be avoided in the following.

Next, h-refinement is considered with both BE and CN. The relative errors are computed and plotted in figure 3.9a. It is apparent that CN yields second order convergence, while the relative error for BE stagnates at a certain level. This is because BE is less accurate in time; the truncation error for BE is of order one, while for CN it is of order two, as seen in equation (3.14). In other words, the error in time is dominating the error in BE, which yields no decrease in the relative error when further increasing the spatial resolution. The point of stagnation of the decrease in the relative error for BE could be postponed to a larger number of degrees of freedom and a smaller error by, e.g., increasing the number of time steps N.

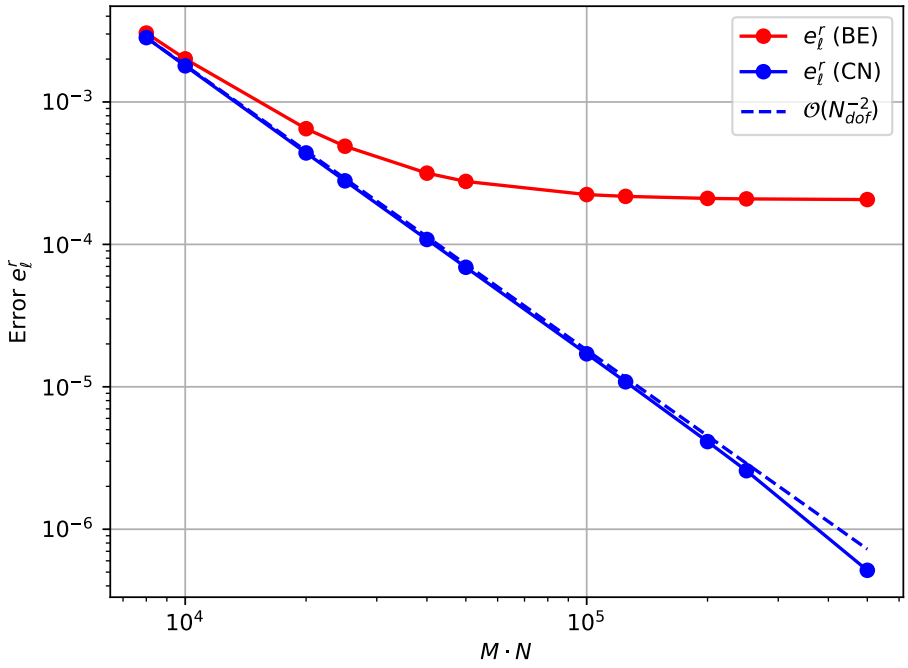
Furthermore, k-refinement, i.e. refinement along the t-axis, is considered. The relative errors are computed and plotted in figure 3.9b. Note that CN yields second order convergence, while BE yields first order convergence. This is as expected from equation (3.14), because, in theory, M is set to a large number ($M = 1000$ in the figure), such that the error in x can be neglected, i.e. $\mathcal{O}(k^2 + h^2) = \mathcal{O}(k^2)$ and $\mathcal{O}(k + h^2) = \mathcal{O}(k)$ for CN and BE, respectively.

Finally, $h \propto k$-refinement is considered, This means that the step lengths are proportional, while they are increased at the same rate. The relative errors are calculated and the resulting convergence plot is depicted in figure 3.10. Note that CN yields convergence of first order, while BE yields convergence of order $\frac{1}{2}$. This is in accordance with the expected theoretical convergence orders, which are

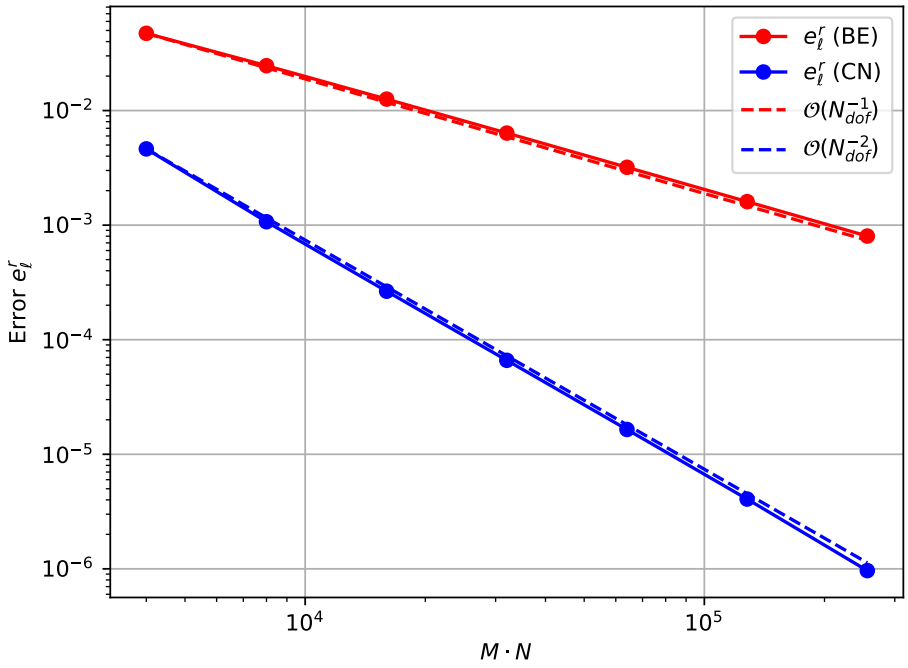
$$\tau_{CN} = \mathcal{O}(k^2 + h^2) \overset{h \propto k}{=} \mathcal{O}(kh) = \mathcal{O}\left(\frac{1}{MN}\right) = \mathcal{O}(N_{\text{dof}}^{-1}),$$

and

$$\tau_{BE} = \mathcal{O}(k + h^2) \overset{h \propto k}{=} \mathcal{O}(h) = \mathcal{O}\left(\frac{1}{M}\right) = \mathcal{O}(N_{\text{dof}}^{-1/2}).$$



(a) h-refinement



(b) k-refinement

Figure 3.9: Heat equation with two Neumann boundary conditions and initial condition $u(x,0) = 2\pi x - \sin(2\pi x)$ on $x \in [0,1]$ and $t \in [0, 0.2]$. The relative error obtained with h-refinement, with $N = 1000$, is plotted in (a). Similarly, the relative error obtained with k-refinement, with $M = 1000$, is plotted in (b). A second order discretization of the boundary conditions is used when calculating the numerical solution with both BE and CN. The x-axis shows the number of degrees of freedom in the linear system.

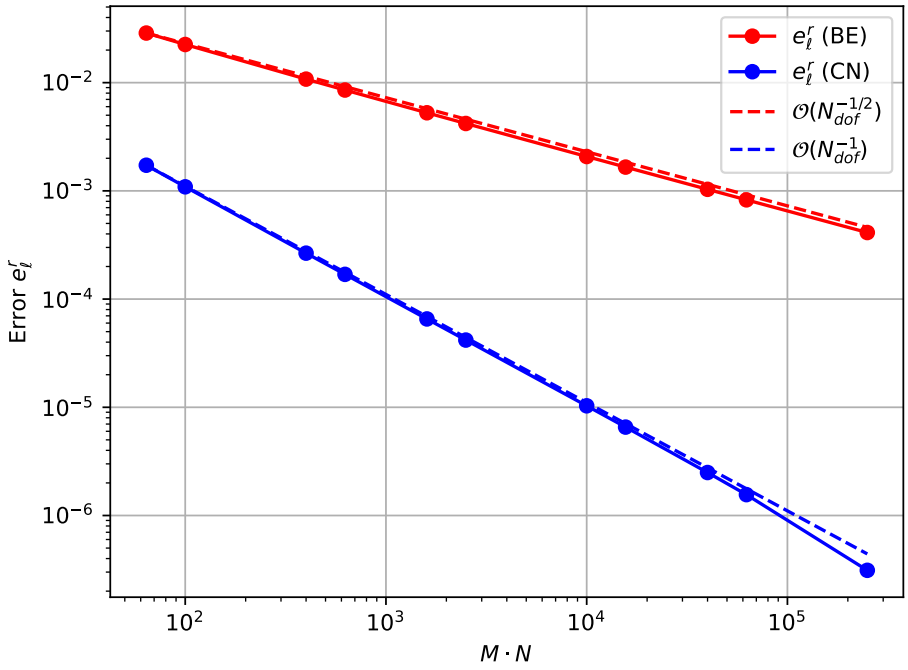


Figure 3.10: Heat equation with two Neumann boundary conditions and initial condition $u(x,0) = 2\pi x - \sin(2\pi x)$ on $x \in [0,1]$ and $t \in [0, 0.2]$. The relative error obtained with $h \propto k$-refinement, is plotted. A second order discretization of the boundary conditions is used when calculating the numerical solution with both BE and CN. The x-axis shows the number of degrees of freedom in the linear system.

b)

In this subsection, the heat equation

$$\begin{aligned} &u_t = u_{xx} \text{ in } \Omega(x,t) : x \in [0,1], t \in [0,T], \\ &u(0,t) = u(1,t) = 0, \end{aligned} \tag{3.15}$$

will be solved. $T = 0.2$ is used in the following. By Fourier analysis, a general solution is on the form [3]

$$u(x,t) = \sum_{n=1}^{\infty} D_n e^{-\pi^2 n^2 t} \sin(n\pi x).$$

Given an initial value $u(x,0) = f(x)$, the coefficients D_n can be determined using the sine orthogonality

$$D_n = 2 \int_0^1 f(x) \sin(n\pi x) \mathrm{d}x.$$

Now, for simplicity, the initial value $f(x) = 3\sin(2\pi x)$ is chosen, which yields a simple manufactured solution on the form

$$u(x,t) = 3e^{-4\pi^2 t} \sin(2\pi x), \tag{3.16}$$

since $D_n = 0$, $\forall n \in \{1\} \cup [3, \infty)$, and $D_2 = 3$. The numerical and manufactured solution is plotted in figure 3.11.

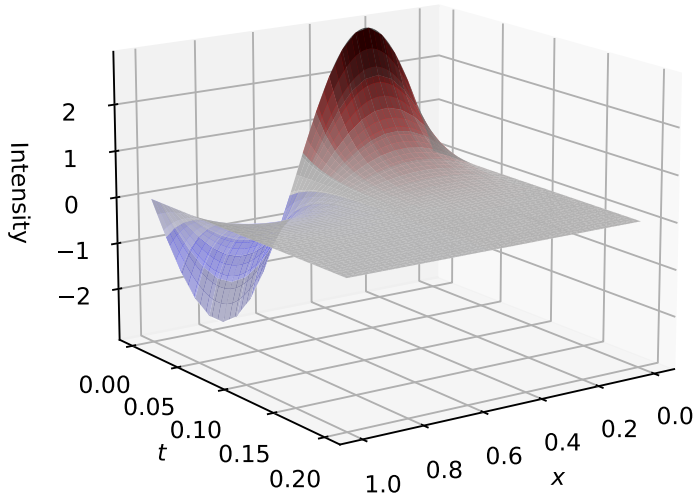


Figure 3.11: Heat equation on $x \in [0,1]$ and $t \in [0,0.2]$ with homogeneous Dirichlet boundary conditions and initial condition $f(x) = 3\sin(2\pi x)$. The numerical solution is calculated with CN with $M = N = 20$ and plotted with a *seismic* color map. The manufactured solution $u(x,t) = 3e^{-4\pi^2 t}\sin(2\pi x)$ is plotted in grey.

Convergence Plots

In the implementation of the numerical methods, both BE and CN are used to yield a method of second order in space, and first and second order in time, respectively (see equation (3.14)). This means that the schemes (3.12) and (3.13), with the matrix Q on the form

$$Q = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix},$$

are employed.

Four types of refinement, namely h-, k-, $(h = ck)$- and $(r = \frac{k}{h^2})$-refinement, will be considered. Convergence rates are found by calculating the relative errors in both the ℓ_2 and L_2 norm, as defined in section 2.3. The errors are calculated with respect to the analytical solution (3.16) at $T = 0.2$ every time the grid is refined. For each type of refinement are the convergence rates calculated with both BE and CN.

First, h-refinement is performed. The convergence plot is depicted in figure 3.12a. As observed in problem **a)**, the error reaches a minimum when BE is used, due to the error in time. Nevertheless, the convergence is of order 2 if N is increased, which is expected

from equation (3.14), because, in theory, N is set to a large number (only $N = 1000$ in the figure) such that the error in t can be neglected, i.e. $\mathcal{O}(k^2 + h^2) = \mathcal{O}(h^2) = \mathcal{O}(N_{\text{dof}}^{-2})$ and $\mathcal{O}(k + h^2) = \mathcal{O}(h^2) = \mathcal{O}(N_{\text{dof}}^{-2})$ for CN and BE, respectively.

Next, k-refinement is performed. The convergence plot is depicted in figure 3.12b. The convergence plot is in compliance with the truncation orders mentioned in equation (3.14). The theory predicts these convergence rates because, when M is set to a large number ($M = 1000$ in the figure), the error in x can be neglected, i.e. $\mathcal{O}(k^2 + h^2) = \mathcal{O}(k^2) = \mathcal{O}(N_{\text{dof}}^{-2})$ and $\mathcal{O}(k + h^2) = \mathcal{O}(k) = \mathcal{O}(N_{\text{dof}}^{-1})$ for CN and BE, respectively.

Next, mesh refinement where $h = ck$, where $c := 1$, is performed. From equation (3.14), it is expected that CN should yield a convergence of order $\mathcal{O}(k^2+h^2) \overset{k\propto h}{=} \mathcal{O}(kh) = \mathcal{O}(N_{\text{dof}}^{-1})$. On the other hand, BE is expected to yield convergence of order $\mathcal{O}(k + h^2) \overset{k\propto h}{=} \mathcal{O}(N_{\text{dof}}^{-\frac{1}{2}} + N_{\text{dof}}^{-1}) = \mathcal{O}(N_{\text{dof}}^{-\frac{1}{2}})$. The relative errors for both BE and CN are displayed in the convergence plot in figure 3.13a, which shows that the expectations are met.

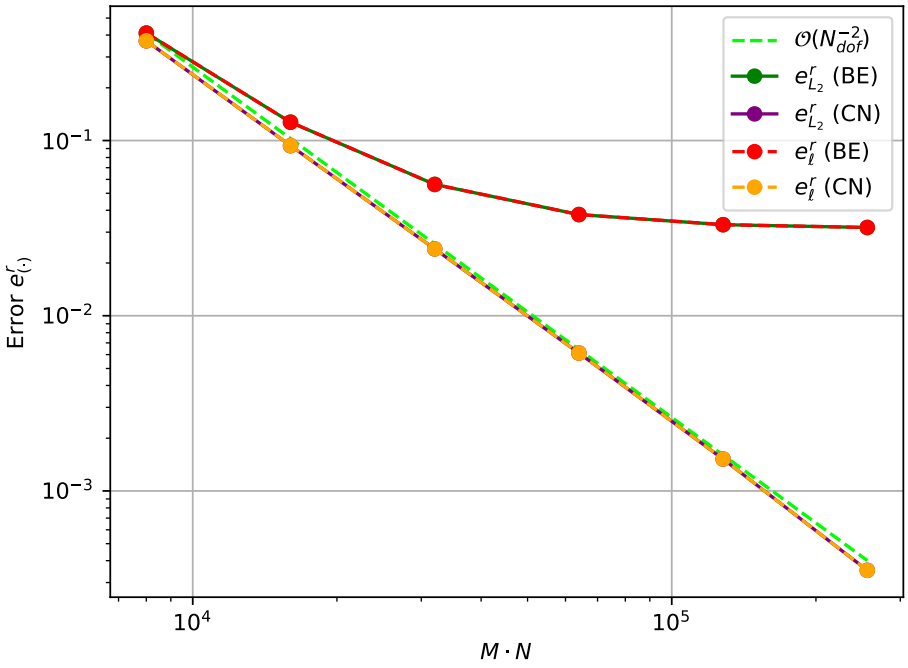
Finally, mesh refinement where $r = \frac{k}{h^2}$ is held constant, is performed. Notice that this implies that $k \propto h^2$ and hence $hk \propto h^3$. Thus, from equation (3.14), CN is expected to yield convergence of order

$$\begin{aligned}\mathcal{O}(k^2 + h^2) &\overset{k^2\propto h^4}{=} \mathcal{O}(h^4 + h^2) \\ &= \mathcal{O}(h^2) \overset{hk\propto h^3}{=} \mathcal{O}((hk)^{\frac{2}{3}}) = \mathcal{O}(N_{\text{dof}}^{-\frac{2}{3}}).\end{aligned}$$

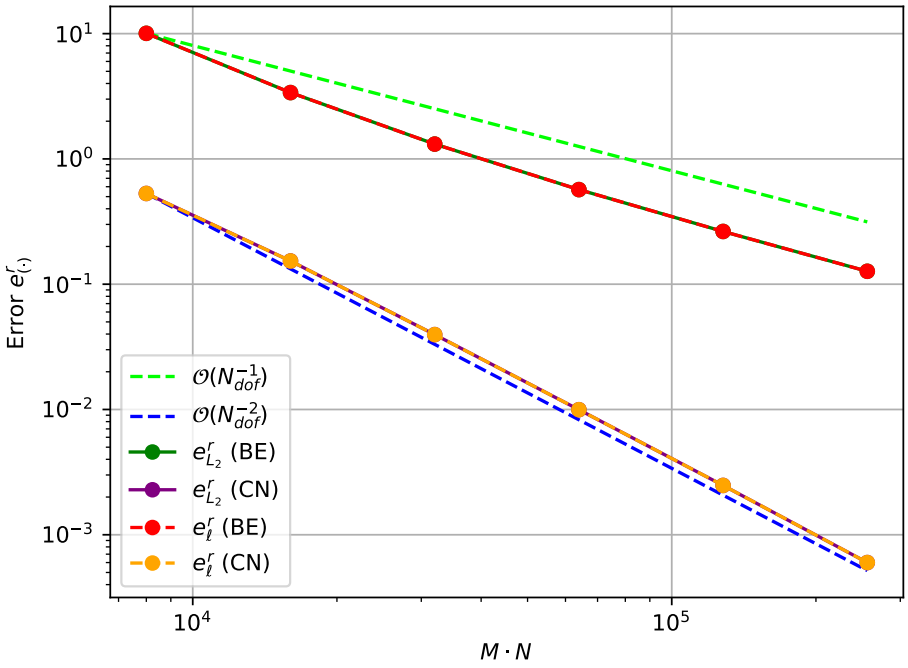
BE is expected to yield the same order of convergence

$$\mathcal{O}(k + h^2) = \mathcal{O}(h^2) = \mathcal{O}((hk)^{\frac{2}{3}}) = \mathcal{O}(N_{\text{dof}}^{-\frac{2}{3}}).$$

The relative errors for both BE and CN are displayed in the convergence plot in figure 3.13b, which shows that the expectations are partly met.

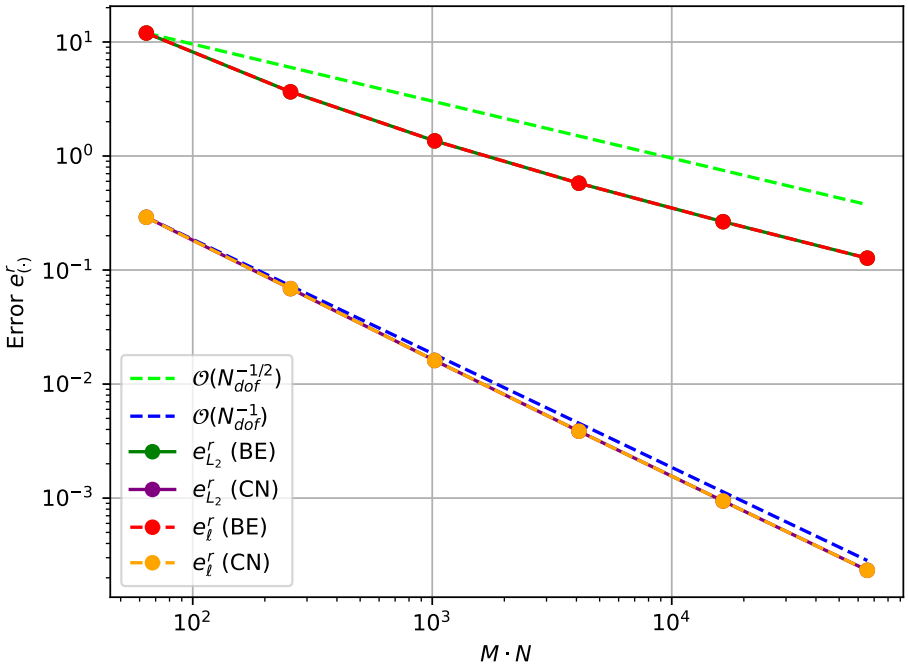


(a) h-refinement

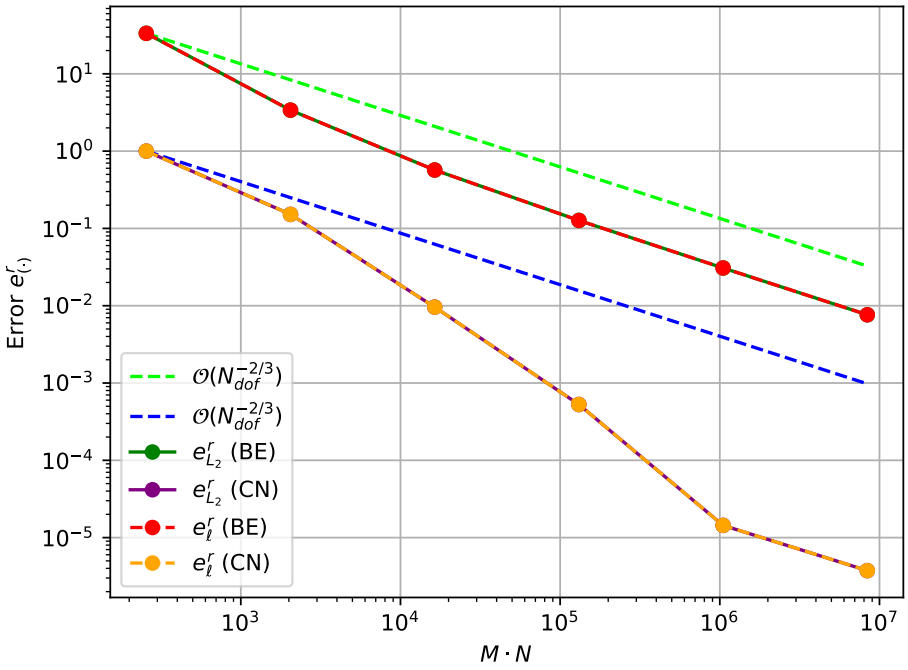


(b) k-refinement

Figure 3.12: Heat equation with homogeneous Dirichlet boundary conditions, with a manufactured solution $u(x,t) = 3e^{-4\pi^2 t}\sin(2\pi x)$, on $x \in [0,1]$ and $t \in [0, 0.2]$. The relative errors e^r_ℓ and $e^r_{L_2}$, obtained with h-refinement, are plotted in (a) with $N = 1000$. The relative errors, obtained with k-refinement, are plotted in (b) with $M = 1000$. Both BE and CN have been used as integrators. The x-axis shows $M \cdot N$, i.e. the number of degrees of freedom of the linear system.



(a) $(h = k)$-refinement



(b) Mesh refinement with $r = k/h^2$ constant

Figure 3.13: Heat equation with homogeneous Dirichlet boundary conditions, with a manufactured solution $u(x,t) = 3e^{-4\pi^2 t}\sin(2\pi x)$, on $x \in [0,1]$ and $t \in [0, 0.2]$. The relative errors e^r_ℓ and $e^r_{L_2}$, obtained with $(h = k)$-refinement, are plotted in (a). The relative errors, obtained with $r = k/h^2$-refinement, are plotted in (b). Both BE and CN have been used as integrators. The x-axis shows $M \cdot N$, i.e. the number of degrees of freedom of the linear system.

c)

The inviscid Burgers' equation

$$u_t = -uu_x,\ u(0,t) = u(1,t) = 0,\ u(x,0) = \exp\left(-400(x-1/2)^2\right), \tag{3.17}$$

will be considered on $x \in [0,1]$, $t > 0$. This is a special case of the parabolic Burgers' equation

$$u_t = \epsilon u_{xx} - uu_x,\ \epsilon \to 0.$$

The x-axis is discretized such that

$$x_0 = 0,\ x_1 = \frac{1}{M+1},\ \ldots,\ x_M = \frac{M}{M+1},\ x_{M+1} = 1.$$

Let $v_m(t) \approx u(x_m,t)$ for $0 \leq m \leq M+1$ be the numerical solution along the line (x_m,t). Using a central difference approximation for the first spatial derivative, which, from section 2.2, is known to be a second order approximation, leads to

$$\dot{v}_m = -v_m \frac{1}{2h}(v_{m+1} - v_{m-1}), \quad 1 \leq m \leq M, \tag{3.18}$$

The differential equations can be written in vector format with $\boldsymbol{v} = [v_1, v_2, \ldots, v_M]^T$. Combining the boundary conditions $v_0 = v_{M+1} = 0$ with (3.18) gives

$$\dot{\boldsymbol{v}} =: \boldsymbol{F}(\boldsymbol{v}) = \frac{1}{2h}\begin{pmatrix} -v_1 v_2 \\ -v_2(v_3 - v_1) \\ \vdots \\ -v_{M-1}(v_M - v_{M-2}) \\ v_M v_{M-1} \end{pmatrix} \tag{3.19}$$

The system (3.19) is solved in time with the ODE-solving method RK4. A detailed description of the RK4 method can be found in chapter 4. The solution shows a breaking time in the interval $t^* \in [0.057, 0.060]$, which agrees with the theoretical value of $t^* = -1/\min\{u'(x,0)\} \approx 0.0583$ found in [9]. Figure 3.14 shows a visualization of the breaking point of the numerical solution.

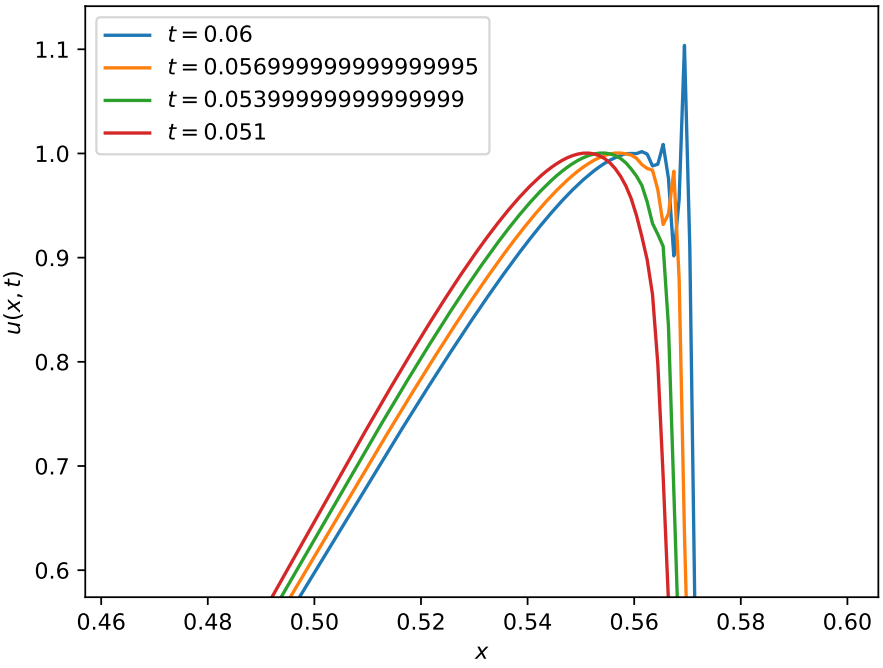


Figure 3.14: Inviscid Burger's equation on $x \in [0, 1]$, $t > 0$, with homogeneous Dirichlet boundary conditions and one initial condition. The numerical solution is plotted along the x-axis for different values of t. The plot illustrates the "breaking" behavior of the solution.

3.3 Problem 3 - Laplace's Equation in Two Dimensions

In this problem, the 2D Laplace equation on the unit square $(x, y) \in [0, 1]^2 =: \Omega$, of the form

$$\Delta u = u_{xx} + u_{yy} = 0,\ u(x, y) = g(x, y) \text{ where } (x, y) \in \partial\Omega, \tag{3.20}$$

is considered. The boundary condition $g(x, y)$ is given by

$$\begin{aligned} g(0, y) &= 0,\ \ 0 \leq y \leq 1, \\ g(x, 0) &= 0,\ \ 0 \leq x \leq 1, \\ g(1, y) &= 0,\ \ 0 \leq y \leq 1, \\ g(x, 1) &= \sin(2\pi x),\ \ 0 \leq x \leq 1. \end{aligned}$$

a)

The analytical solution of equation (3.20) is derived by using separation of variables [7]. Assume solutions on the form $u(x, y) = X(x)Y(y)$. Inserted into the PDE (3.20), this assumption gives

$$X''(x)Y(y) + X(x)Y''(y) = 0.$$

This expression may be rearranged to

$$\frac{X''(x)}{X(x)} = -\frac{Y''(y)}{Y(y)} = \lambda,$$

since the variables x and y are separated. This means that both sides of the PDE must be a constant, denoted by λ. Hence, the PDE has been transformed into two separate ODEs

$$X''(x) - \lambda X(x) = 0, \tag{3.21}$$
$$Y''(y) + \lambda Y(y) = 0. \tag{3.22}$$

The boundary conditions now become

$$g(0, y) = X(0)Y(y) = 0,\ \ 0 \leq y \leq 1, \tag{3.23}$$
$$g(1, y) = X(1)Y(y) = 0,\ \ 0 \leq y \leq 1, \tag{3.24}$$
$$g(x, 0) = X(x)Y(0) = 0,\ \ 0 \leq x \leq 1, \tag{3.25}$$
$$g(x, 1) = X(x)Y(1) = \sin(2\pi x),\ \ 0 \leq x \leq 1. \tag{3.26}$$

Equation (3.21) is examined first. The boundary conditions (3.23) and (3.24) give $X(0) = 0$ and $X(1) = 0$, assuming that $Y(y) \neq 0$, because that would only lead to $u(x, y) = X(x)Y(y) = 0$, which is of no interest. When solving the equation (3.21), there are three different cases of the constant λ to be considered. Firstly, if $\lambda = 0$, the solution becomes a linear function $X(x) = A_1 x + A_2$. The boundary conditions then give $A_2 = A_1 = 0$, which is uninteresting. Secondly, for positive λ, a general solution is $X(x) = A_1 e^{\sqrt{\lambda}x} + A_2 e^{-\sqrt{\lambda}x}$. Using the boundary conditions, this gives the linear system

$$\begin{aligned} X(0) &= A_1 + A_2 = 0, \\ X(1) &= A_1 e^{\sqrt{\lambda}} + A_2 e^{-\sqrt{\lambda}} = 0, \end{aligned}$$

which still gives $A_1 = A_2 = 0$. Finally, if λ is negative, say $\lambda = -k^2$, the general solution is $X(x) = A_1 \cos kx + A_2 \sin kx$. Inserting the boundary conditions, the constants become $A_1 = 0$ and

$$X(1) = A_2 \sin k = 0 \overset{A_2 \neq 0}{\Longrightarrow} \sin k = 0 \implies k = n\pi \text{ for } n = \ldots, -1, 0, 1, \ldots.$$

Setting $A_2 = 1$ gives infinitely many solutions $X_n(x) := X(x) = \sin(n\pi x)$, which satisfy $X''(x) - \lambda X(x) = 0$ when $\lambda := -k^2 = -n^2\pi^2$.

Next, the solution of equation (3.22) is calculated. Inserting $\lambda := -k^2 = -n^2\pi^2$ gives $Y''(y) - n^2\pi^2 Y(y) = 0$. A general solution of this equation is $Y(y) = B_1 e^{n\pi y} + B_2 e^{-n\pi y}$. Thus, the solution in n, determined up to the constants B_1 and B_2, is

$$u_n(x, y) = (B_1 e^{n\pi y} + B_2 e^{-n\pi y}) \sin(n\pi x).$$

Because the equation (3.20) is linear and homogeneous, the superposition principle holds, which means that any linear combination of solutions to the PDE is also a solution. Hence, the infinite sum

$$u(x, y) = \sum_{n=1}^{\infty} u_n(x, y) = \sum_{n=1}^{\infty} (B_1 e^{n\pi y} + B_2 e^{-n\pi y}) \sin(n\pi x) \tag{3.27}$$

is a solution of (3.20). Note that the expression sums over $n \in [1, \infty)$ because $\sin(-n\pi x) = -\sin(n\pi x)$, which means that this sum incorporates the solutions for $n \in (\infty, 1)$ by means of a change in the coefficients B_1 and B_2.

Two remaining boundary conditions still need to be satisfied. The boundary condition (3.25) gives

$$0 = u(x, 0) = \sum_{n=1}^{\infty} (B_1 + B_2) \sin(n\pi x),$$

which, by Fourier analysis, means that $B_1 + B_2 = \int_0^1 0 \cdot \sin(n\pi x) \mathrm{d}x = 0$, which trivially gives $B_1 = -B_2$. Hence, this boundary condition is satisfied when the sum (3.27) takes the form

$$u(x, y) = \sum_{n=1}^{\infty} B_1 (e^{n\pi y} - e^{-n\pi y}) \sin(n\pi x) = \sum_{n=1}^{\infty} B_n^* \sinh(n\pi y) \sin(n\pi x),$$

where $B_n^* = 2B_1$. The last boundary condition (3.26) gives

$$u(x, 1) = \sum_{n=1}^{\infty} B_n^* \sinh(n\pi) \sin(n\pi x) = \sin(2\pi x),$$

which, by Fourier analysis, means that $B_n^* \sinh(n\pi) = 2 \int_0^1 \sin(2\pi x) \sin(n\pi x) \mathrm{d}x \implies B_n^* = \frac{2}{\sinh(n\pi)} \int_0^1 \sin(2\pi x) \sin(n\pi x) \mathrm{d}x$. Hence, the analytical solution is given by

$$\begin{aligned} u(x, y) &= \sum_{n=1}^{\infty} B_n^* \sinh(n\pi y) \sin(n\pi x), \\ B_n^* &= \frac{2}{\sinh(n\pi)} \int_0^1 \sin(2\pi x) \sin(n\pi x) \mathrm{d}x. \end{aligned} \tag{3.28}$$

Next, the coefficients B_n^* are calculated. From the expression of B_n^* in (3.28) it is apparent that

$$\begin{aligned} B_2^* &= \frac{2}{\sinh(2\pi)} \int_0^1 \sin(2\pi x) \sin(2\pi x) \mathrm{d}x \\ &= \frac{2}{\sinh(2\pi)} \cdot \frac{1}{2} = \frac{1}{\sinh(2\pi)}, \end{aligned}$$

is the only non-zero coefficient, since $\sin(mx)$ and $\sin(nx)$ are orthogonal functions when $m \neq n$. Hence, the final solution is

$$u(x, y) = \frac{1}{\sinh(2\pi)} \sinh(2\pi y) \sin(2\pi x). \tag{3.29}$$

b)

In order to solve equation (3.20) numerically, the five point formula

$$\delta_x^2 U_p + \delta_y^2 U_p = 0, \tag{3.30}$$

where p is the point in which the solution is to be found, is implemented on a grid. The x-axis is discretized as

$$x_0 = 0,\ x_1 = \frac{1}{M_x + 1},\ \ldots,\ x_{M_x} = \frac{M_x}{M_x + 1},\ x_{M_x+1} = 1,$$

and the y-axis is similarly discretized as

$$y_0 = 0,\ y_1 = \frac{1}{M_y + 1},\ \ldots,\ y_{M_y} = \frac{M_y}{M_y + 1},\ x_{M_y+1} = 1.$$

Constant step sizes $h = 1/(M_x + 1)$ and $k = 1/(M_y + 1)$ are used, where M_x and M_y are the number of internal nodes in the discretized grid in the x- and y-direction, respectively. In the general case, the five point formula yields

$$\frac{1}{h^2}(U_W - 2U_p + U_E) + \frac{1}{k^2}(U_S - 2U_p + U_N) = 0,$$

where the cardinal directions north (N), south (S), east (E) and west (W) are shown in figure 3.15 for a regular grid, i.e. $k = h$. When the grid is non-regular, the directions are still the same, but the lengths of the lines shown in figure 3.15 will change.

By traversing the x-axis from left to right, while traversing the y-axis from bottom to top, the five point stencil gives rise to a linear system $A\boldsymbol{U} = \boldsymbol{F}$. To illustrate, setting $M_x = M_y = 3$ gives 9 unknowns, and gives rise to the system

$$A = \begin{pmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{pmatrix}, \boldsymbol{U} = \begin{pmatrix} U_1^1 \\ U_1^2 \\ U_1^3 \\ U_2^1 \\ U_2^2 \\ U_2^3 \\ U_3^1 \\ U_3^2 \\ U_3^3 \end{pmatrix}, \boldsymbol{F} = \begin{pmatrix} 0 \\ 0 \\ -\sin(2\pi\frac{1}{4}) \\ 0 \\ 0 \\ -\sin(2\pi\frac{1}{2}) \\ 0 \\ 0 \\ -\sin(2\pi\frac{3}{4}) \end{pmatrix},$$

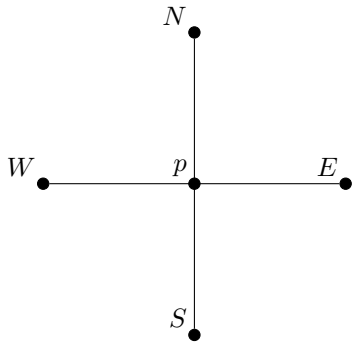


Figure 3.15: Five point stencil for the 2D Laplace equation, on a regular grid.

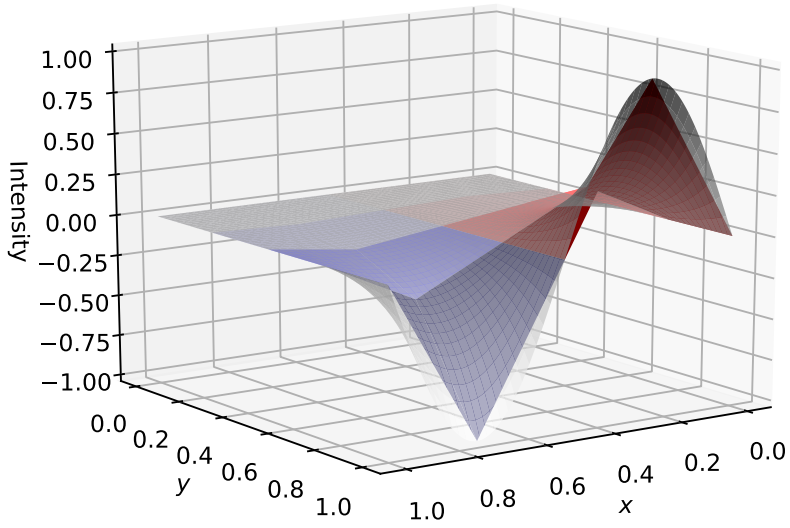
where the notation $U^*_{(\cdot)}$ encodes the solution in the $((\cdot), *)$-coordinate of the regular grid. Note that all the unknowns are internal points, since the boundary conditions are given. The solution to this system is shown in figure 3.16a. Since the solution is very "non-smooth", i.e. a bad approximation of the analytical solution, more internal points need to be added to the grid.

The linear system obviously changes when M_x and M_y change, but the manner in which the system is solved is similar. An example of a simple system where the grid is non-uniform is given next. Setting $M_y = 2$ and $M_x = 3$, gives $k = \frac{1}{3}$ and $h = \frac{1}{4}$. Then, the linear system is given by

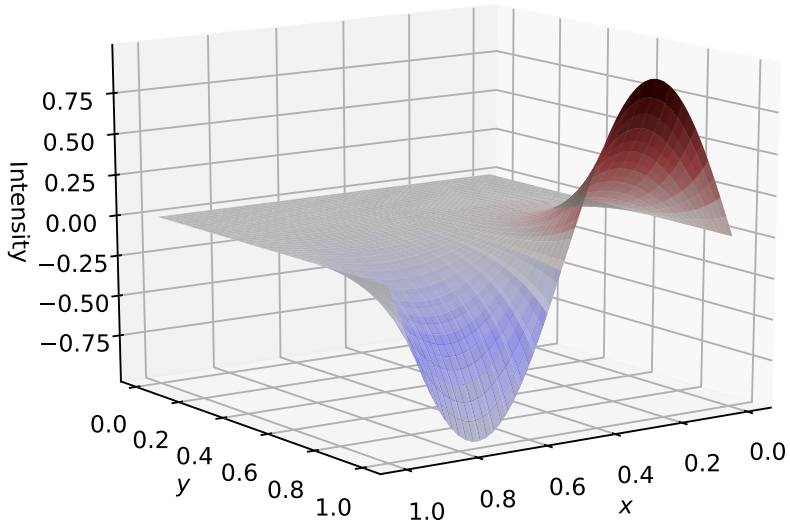
$$A = \begin{pmatrix} -2(\frac{1}{h^2}+\frac{1}{k^2}) & \frac{1}{k^2} & \frac{1}{h^2} & 0 & 0 & 0 \\ \frac{1}{k^2} & -2(\frac{1}{h^2}+\frac{1}{k^2}) & 0 & \frac{1}{h^2} & 0 & 0 \\ \frac{1}{h^2} & 0 & -2(\frac{1}{h^2}+\frac{1}{k^2}) & \frac{1}{k^2} & \frac{1}{h^2} & 0 \\ 0 & \frac{1}{h^2} & \frac{1}{k^2} & -2(\frac{1}{h^2}+\frac{1}{k^2}) & 0 & \frac{1}{h^2} \\ 0 & 0 & \frac{1}{h^2} & 0 & -2(\frac{1}{h^2}+\frac{1}{k^2}) & \frac{1}{k^2} \\ 0 & 0 & 0 & \frac{1}{h^2} & \frac{1}{k^2} & -2(\frac{1}{h^2}+\frac{1}{k^2}) \end{pmatrix},$$

$$\boldsymbol{U} = \begin{pmatrix} U_1^1 \\ U_1^2 \\ U_2^1 \\ U_2^2 \\ U_3^1 \\ U_3^2 \end{pmatrix}, \boldsymbol{F} = \begin{pmatrix} 0 \\ -\frac{1}{k^2}\sin(2\pi \cdot h) \\ 0 \\ -\frac{1}{k^2}\sin(2\pi \cdot 2h) \\ 0 \\ -\frac{1}{k^2}\sin(2\pi \cdot 3h) \end{pmatrix},$$

The solution of this system is not displayed, since it is very non-smooth as well. Instead, as a visual confirmation of the similarities between the numerical and the analytical solution, figure 3.16b shows the solution with $M_x = M_y = 50$, i.e. 2500 internal grid points. The solution looks relatively smooth in this case.



(a) $M_x = M_y = 3$



(b) $M_x = M_y = 50$

Figure 3.16: Two-Dimensional Laplace Equation, considered on the unit square. The analytical solution is shown in grey. The numerical solution is displayed in a *seismic* color map. In (a), the numerical solution is calculated with $M_x = M_y = 3$, i.e. with 9 unknowns on a regular grid. In (b), the numerical solution is calculated with $M_x = M_y = 50$, i.e. with 2500 unknowns on a regular grid.

In the next sections, the convergence of the numerical solution to the analytical solution will be quantified. Since the five point formula (3.30) uses a second order central difference approximation, as defined in section 2.2, in both directions x and y, the truncation error is of order $\mathcal{O}(h^2 + k^2)$. The relative error e_ℓ^r for three different types of grids is displayed in "log-log"-convergence plots. The types of grids are

- h-refinement: M_x increasing and M_y constant
- k-refinement: M_y increasing and M_x constant
- $(h = k)$-refinement: Both M_x and M_y increasing at the same rate

Convergence with h-refinement

Convergence plots for increasing M_x, with arbitrary constant values of M_y, are shown in figure 3.17. In all the cases presented, the convergence order is approximately $\mathcal{O}(N_{\text{dof}}^{-2})$ up to a certain point, where the convergence order "flattens out". This matches the expected convergence rate, which is

$$\mathcal{O}(h^2 + k^2) = \mathcal{O}(h^2) \overset{h \propto N_{\text{dof}}^{-1}}{=} \mathcal{O}(N_{\text{dof}}^{-2}),$$

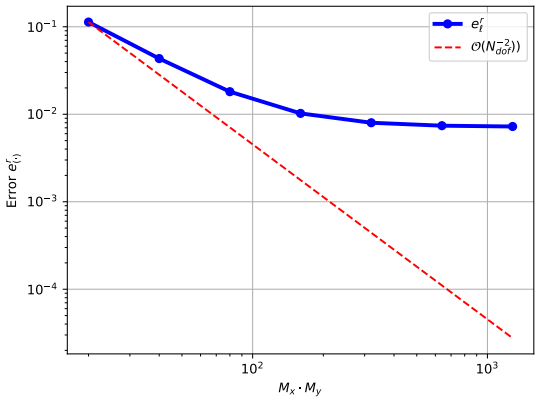
where $N_{\text{dof}} = M_x \cdot M_y$. This is expected when h-refinement is being performed, because h dominates the error when k is assumed to be sufficiently small. This is observed in practice when M_y is increased. Note that the point where the convergence rate becomes asymptotic is shifted towards smaller errors and larger degrees of freedom when the constant M_y is increased.

Convergence with k-refinement

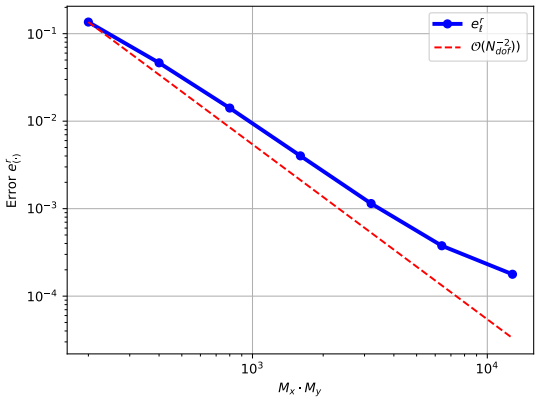
Convergence plots for increasing M_y, with arbitrary constant values of M_x, are shown in figure 3.18. The expected convergence order, in this case, is

$$\mathcal{O}(h^2 + k^2) = \mathcal{O}(k^2) \overset{k \propto N_{\text{dof}}^{-1}}{=} \mathcal{O}(N_{\text{dof}}^{-2}).$$

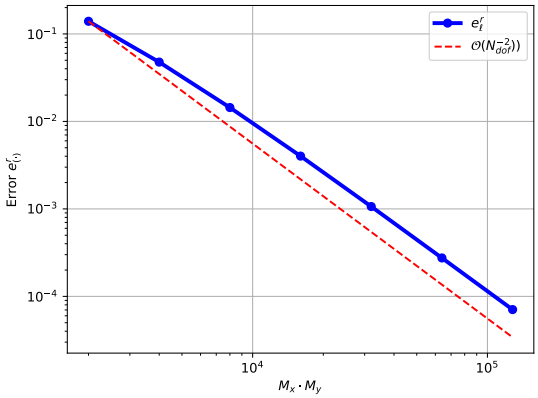
The reason behind this expectation is similar as for when h-refinement is performed. In theory, h is assumed to be a sufficiently small number, such that the error stemming from k will be asymptotically larger. In case (a) and (b), the expected convergence order of $\mathcal{O}(N_{\text{dof}}^{-2})$ is not reached. The reason is that the constant M_x is not large enough, which is assumed when neglecting the h^2-term when calculating the expected convergence order. Therefore, when the constant is increased, the expected convergence order is met, as observed in figures (c) and (d). Note that the point where the convergence rate becomes asymptotic is shifted towards smaller errors and larger degrees of freedom when the constant M_x is increased. Moreover, comparing to the plots in figure 3.17, it can be inferred that the refinement in the x-direction is crucial for reaching the expected convergence order. In other words, x requires a finer grid compared to y.



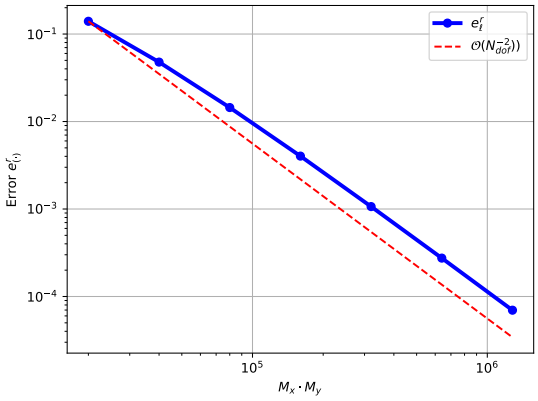
(a) $M_y = 10$ and M_x increasing



(b) $M_y = 100$ and M_x increasing



(c) $M_y = 10^3$ and M_x increasing



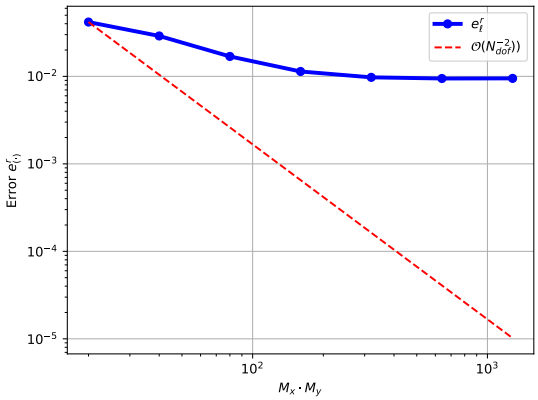
(d) $M_y = 10^4$ and M_x increasing

Figure 3.17: Two-Dimensional Laplace Equation, considered on the unit square. The relative error e_ℓ^r is shown for some arbitrary constant values of M_y and increasing M_x (h-refinement). The x-axis shows $M_x \cdot M_y$, i.e. the number of degrees of freedom of the linear system.

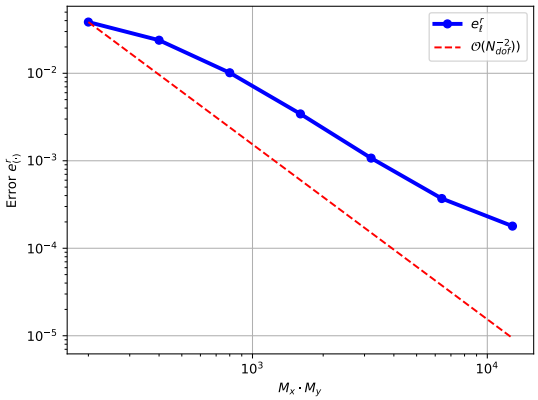
Convergence plot with $(h = k)$-refinement

A convergence plot where both M_x and M_y are increasing simultaneously, i.e. $M_x = M_y$ at all times, is shown in figure 3.19. As seen from the plot, the convergence order is approximately $\mathcal{O}(N_{\text{dof}}^{-1})$. This is in accordance with the expected order of convergence, which is

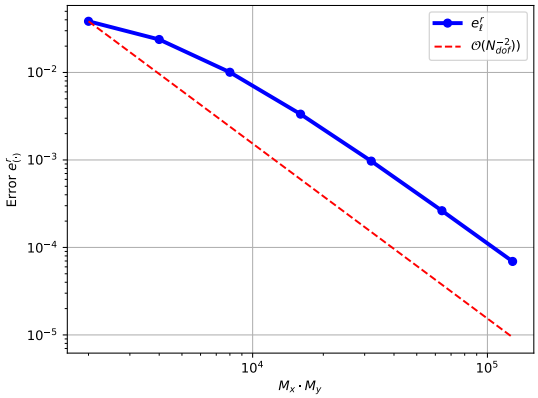
$$\mathcal{O}(h^2 + k^2) \overset{h=k}{=} \mathcal{O}(hk) = \mathcal{O}(N_{\text{dof}}^{-1}).$$



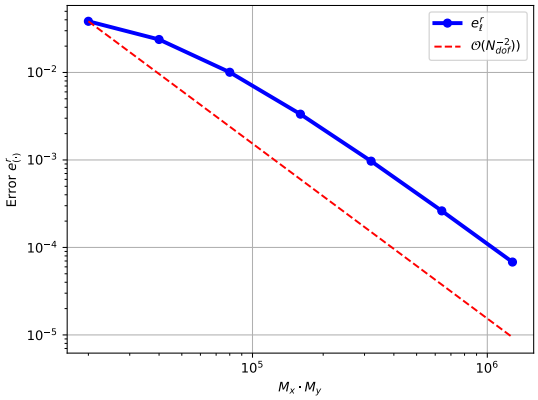
(a) $M_x = 10$ and M_y increasing



(b) $M_x = 100$ and M_y increasing



(c) $M_x = 10^3$ and M_y increasing



(d) $M_x = 10^4$ and M_y increasing

Figure 3.18: Two-Dimensional Laplace Equation, considered on the unit square. The relative error e_ℓ^r is shown for some arbitrary constant values of M_x and increasing M_y (k-refinement). The x-axis shows $M_x \cdot M_y$, i.e. the number of degrees of freedom of the linear system.

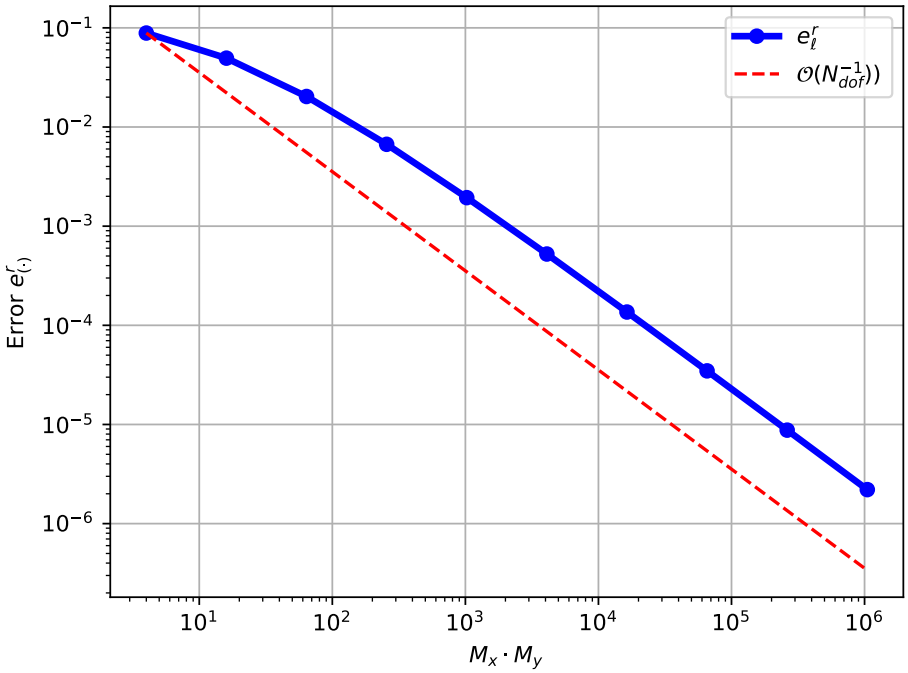


Figure 3.19: Two-Dimensional Laplace Equation, considered on the unit square. Plot of the relative error e_ℓ^r where both M_x and M_y are increasing monotonically ($(h = k)$-refinement). The x-axis shows $M_x \cdot M_y$, i.e. the number of degrees of freedom of the linear system.

3.4 Problem 4 - Linearized Korteweg-deVries Equation

In this problem, the following linearized Korteweg-deVries (KdV) equation is considered on the interval $x \in [-1, 1]$, $t > 0$,

$$u_t + (1 + \pi^2)u_x + u_{xxx} = 0,\ u(x, 0) = \sin(\pi x). \tag{3.31}$$

A periodic boundary condition with period 2, i.e. $u(x+2, t) = u(x, t)$, is assumed. Hence, the function may be studied only on the interval $[-1, 1]$. The grid points

$$x_0 = -1,\ x_1 = -1 + \frac{2}{M},\ \ldots,\ x_M = 1,$$

are considered.

a)

In order to solve equation (3.31) numerically, it has to be discretized in time and space. Let $u_m^n := u(x_m, t_n)$ where $x_m = -1 + mh$, $0 \leq m \leq M$ and $h = \frac{2}{M}$. The central difference approximation for the first derivative, as defined in equation (2.1), is used to approximate the first spatial derivative. Thus, the approximation of $(u')_m := u_x(x_m)$ at time step $t = t_n$ is

$$\begin{aligned}(u')_m &= \frac{1}{h}\mu\delta_x u_m + \mathcal{O}(h^2) \\ &= \frac{1}{2h}\left(u_{m+1} - u_{m-1}\right) + \mathcal{O}(h^2).\end{aligned} \tag{3.32}$$

Similarly, a discrete approximation for the third derivative $(u''')_m := u_{xxx}(x_m)$ at $t = t_n$ is

$$\begin{aligned}(u''')_m &= \frac{1}{h^3}(\mu\delta_x)^3 u_m + \mathcal{O}(h^2) = \frac{1}{2h^3}(\mu\delta_x)^2\left(u_{m+1} - u_{m-1}\right) + \mathcal{O}(h^2) \\ &= \frac{1}{4h^3}\mu\delta_x\left(u_{m+2} - 2u_m + u_{m-2}\right) + \mathcal{O}(h^2) \\ &= \frac{1}{8h^3}\left(u_{m+3} - 3u_{m+1} + 3u_{m-1} - u_{m-3}\right) + \mathcal{O}(h^2).\end{aligned} \tag{3.33}$$

Both these discretizations have a local truncation error of $\mathcal{O}(h^2)$. For the discretization in the temporal direction, both Euler's method and the Crank-Nicolson (trapezoidal) method are used.

The temporal discretization using Euler's method can be derived as follows. The step size in the temporal direction is denoted by k. An expansion of the exact solution u_m^{n+1} for constant $x = x_m$ around $t = t_n$ yields

$$\begin{aligned}u_m^{n+1} &= u_m^n + k\partial_t u_m^n + \frac{1}{2}k^2\partial_t^2 u_m^n + \ldots \\ &= u_m^n - k((1 + \pi^2)(u')_m^n + (u''')_m^n) + \mathcal{O}(k^2),\end{aligned}$$

where equation (3.31) has been used in the second equality. Also, a superscript n is added to the derivative approximations, to signal at which point in time the functions are evaluated. Now, inserting the two spacial discretizations (3.32) and (3.33) gives

$$\begin{aligned} u_m^{n+1} &= u_m^n - k((1+\pi^2)(u')_m^n + (u''')_m^n) + \mathcal{O}(k^2) \\ &= u_m^n - k\left(\frac{1+\pi^2}{2h}(u_{m+1}^n - u_{m-1}^n) + \frac{1}{8h^3}(u_{m+3}^n - 3u_{m+1}^n + 3u_{m-1}^n - u_{m-3}^n)\right) \\ &\quad + \mathcal{O}(k^2 + kh^2) \\ &= u_m^n - \frac{k}{2h}\left(\frac{1}{4h^2}u_{m+3}^n + \left[1+\pi^2 - \frac{3}{4h^2}\right]u_{m+1}^n - \left[1+\pi^2 - \frac{3}{4h^2}\right]u_{m-1}^n - \frac{1}{4h^2}u_{m-3}^n\right) \\ &\quad + \mathcal{O}(k^2 + kh^2), \end{aligned} \tag{3.34}$$

which has a truncation error, τ_m^n, of order $\mathcal{O}(k + h^2)$, since $k\tau_m^n = \mathcal{O}(k^2 + kh^2)$. The notation U_m^n is adopted to specify the approximate solution of u_m^n, i.e. in the point (x_m, t_n). Inserting this approximate solution into equation (3.34) and neglecting the truncation error yields the difference scheme

$$U_m^{n+1} = U_m^n + k\left(-aU_{m+3}^n - bU_{m+1}^n + bU_{m-1}^n + aU_{m-3}^n\right), \tag{3.35}$$

where the coefficients a and b are defined as

$$a = \frac{1}{8h^3}, \quad b = \frac{1+\pi^2}{2h} - \frac{3}{8h^3}. \tag{3.36}$$

Crank-Nicolson is based on the trapezoidal rule, which yields a slightly different derivation of the temporal discretization when using this method. The fundamental theorem of calculus, together with the trapezoidal quadrature, imply

$$\begin{aligned} u(x_m, t_{n+1}) - u(x_m, t_n) &= \int_{t_n}^{t_{n+1}} u_t(x_m, t)\mathrm{d}t \\ &= \frac{t_{n+1} - t_n}{2}\left(u_t(x_m, t_{n+1}) + u_t(x_m, t_n)\right)) + \mathcal{O}((t_{n+1} - t_n)^3) \\ &= \frac{1}{2}k\left(u_t(x_m, t_{n+1}) + u_t(x_m, t_n)\right)) + \mathcal{O}(k^3). \end{aligned}$$

This expression, together with the (KdV)-equation (3.31), leads to

$$\begin{aligned} u_m^{n+1} &= u_m^n + \frac{1}{2}k(\partial_t u_m^n + \partial_t u_m^{n+1}) + \mathcal{O}(k^3) \\ &= u_m^n - \frac{1}{2}k((1+\pi^2)(u')_m^n + (u''')_m^n + (1+\pi^2)(u')_m^{n+1} + (u''')_m^{n+1}) + \mathcal{O}(k^3) \\ &= u_m^n - \frac{1}{2}k((1+\pi^2)((u')_m^n + (u')_m^{n+1}) + (u''')_m^n + (u''')_m^{n+1}) + \mathcal{O}(k^3), \end{aligned}$$

where the $\mathcal{O}(k^3)$-error stems from the trapezoidal quadrature. Inserting the two spacial discretizations (3.32) and (3.33) yields

$$
\begin{aligned}
u_m^{n+1} &= u_m^n - \frac{1}{2}k((1+\pi^2)((u')_m^n + (u')_m^{n+1}) + (u''')_m^n + (u''')_m^{n+1}) + \mathcal{O}(k^3) \\
&= u_m^n - \frac{1}{2}k\left(\frac{1+\pi^2}{2h}(u_{m+1}^n - u_{m-1}^n + u_{m+1}^{n+1} - u_{m-1}^{n+1})\right. \\
&\left.+\frac{1}{8h^3}(u_{m+3}^n - 3u_{m+1}^n + 3u_{m-1}^n - u_{m-3}^n + u_{m+3}^{n+1} - 3u_{m+1}^{n+1} + 3u_{m-1}^{n+1} - u_{m-3}^{n+1})\right) + \mathcal{O}(k^3 + kh^2) \\
&= u_m^n + \frac{k}{2}\left(\frac{1}{8h^3}(-u_{m+3}^{n+1} + u_{m-3}^{n+1} - u_{m+3}^n + u_{m-3}^n)\right. \\
&\left.+\left(\frac{1+\pi^2}{2h} - \frac{3}{8h^3}\right)(-u_{m+1}^{n+1} + u_{m-1}^{n+1} - u_{m+1}^n + u_{m-1}^n)\right) + \mathcal{O}(k^3 + kh^2).
\end{aligned}
\tag{3.37}
$$

Finally, insertion of the approximate solution U into the equation (3.37), while neglecting the truncation error, and making use of the coefficients defined in (3.36), gives the difference scheme

$$
U_m^{n+1} = U_m^n + \frac{k}{2}\left(-aU_{m+3}^{n+1} - bU_{m+1}^{n+1} + bU_{m-1}^{n+1} + aU_{m-3}^{n+1} - aU_{m+3}^n - bU_{m+1}^n + bU_{m-1}^n + aU_{m-3}^n\right). \tag{3.38}
$$

Von Neumann stability

Next in line is to examine if these two discretization methods are Von Neumann stable. The discretization based on Euler's method (3.35) is examined first. Assume solutions of the form $U_m^n = \xi^n \mathrm{e}^{i\beta x_m}$ with $x_m = -1 + mh$. Insertion into the discretization formula yields an equation where all the terms contain the expression $\mathrm{e}^{-i\beta}$. Dividing the equation by this exponential expression yields

$$
\xi^{n+1}\mathrm{e}^{i\beta mh} = \xi^n \mathrm{e}^{i\beta mh} + k\xi^n(-a\mathrm{e}^{i\beta(m+3)h} - b\mathrm{e}^{i\beta(m+1)h} + b\mathrm{e}^{i\beta(m-1)h} + a\mathrm{e}^{i\beta(m-3)h}).
$$

Moreover, dividing the equation by $\xi^n \mathrm{e}^{i\beta mh}$ leads to

$$
\begin{aligned}
\xi &= 1 + k(-a\mathrm{e}^{3i\beta h} - b\mathrm{e}^{i\beta h} + b\mathrm{e}^{-i\beta h} + a\mathrm{e}^{-3i\beta h}) \\
&= 1 - 2ki(a\sin(3\beta h) + b\sin(\beta h)).
\end{aligned}
$$

Now, using the fact that $\sin(3x) = 3\sin(x) - 4\sin^3(x)$ and that $b = \frac{1+\pi^2}{2h} - 3a$, yields

$$
\begin{aligned}
\xi &= 1 - 2ki\left(3a\sin(\beta h) - 4a\sin^3(\beta h) + \left(\frac{1+\pi^2}{2h} - 3a\right)\sin(\beta h)\right) \\
&= 1 - 2ki\left(\left(\frac{1+\pi^2}{2h}\right)\sin(\beta h) - 4a\sin^3(\beta h)\right) \\
&= 1 + i\frac{k}{h}\sin(\beta h)\left(\frac{\sin^2(\beta h)}{h^2} - (1+\pi^2)\right),
\end{aligned}
\tag{3.39}
$$

where $a = \frac{1}{8h^3}$ has been inserted in the last step. To achieve Von Neumann stability, the condition $|\xi| \leq 1 + \mu k$, where $\mu \geq 0$ is a constant independent of h and k, needs to be satisfied. The Von Neumann stability criterion is equivalently stated as

$$|\xi|^2 \leq 1 + 2\mu k + (\mu k)^2.$$

The squared modulus of equation (3.39) is given as

$$\begin{aligned} |\xi|^2 &= 1 + \frac{k^2}{h^2}\sin^2(\beta h)\left(\frac{\sin^2(\beta h)}{h^2} - (1+\pi^2)\right)^2 \\ &= 1 + k^2\beta_h^2\left(\beta_h^2 - C^2\right)^2, \end{aligned}$$

where the quantities $C := \sqrt{1+\pi^2}$ and $\beta_h := \frac{\sin(\beta h)}{h}$ are defined. Assuming (by contradiction) that the Von Neumann criterion is satisfied gives

$$\begin{aligned} & 2\mu k + \mu^2 k^2 - k^2\beta_h^2\left(\beta_h^2 - C^2\right)^2 \geq 0 \\ & \Rightarrow \mu^2 k^2 + 2\mu k - k^2\beta_h^2\left(\beta_h^2 - C^2\right)^2 = D,\ D \geq 0 \\ \text{(abc-formula)}\ & \Rightarrow \mu = \frac{1}{k}\left(-1 \pm \sqrt{1 + D + k^2\beta_h^2(\beta_h^2 - C^2)^2}\right). \end{aligned}$$

Observe that it is possible for μ to be a non-negative constant if the positive solution of the quadratic formula is chosen. Furthermore, independence of μ from h and k is only possible if $\mu = 0$. These observations yield

$$\begin{aligned} & \mu = 0 \Rightarrow -1 + \sqrt{1 + D + k^2\beta_h^2(\beta_h^2 - C^2)^2} = 0 \\ & \Rightarrow D = -k^2\beta_h^2(\beta_h^2 - C^2)^2. \end{aligned}$$

The last equation is clearly a contradiction to the criterion that $D \geq 0$. Hence, there exists no positive constant μ independent of h and k that satisfies the Von Neumann criterion. Conclusively, the discretization based on Euler's method (3.35) is not Von Neumann stable.

Next, the discretization based on Crank-Nicolson (3.38), is examined. In a similar manner, it is assumed that $U_m^n = \xi^n \mathrm{e}^{i\beta x_m}$ and $x_m = -1 + mh$. Insertion into the discretization formula and dividing the equation by $\xi^n \mathrm{e}^{i\beta mh}\mathrm{e}^{-i\beta}$ yields

$$\begin{aligned} \xi &= 1 + \frac{k}{2}\left(-a\xi\mathrm{e}^{3i\beta h} - b\xi\mathrm{e}^{i\beta h} + b\xi\mathrm{e}^{-i\beta h} + a\xi\mathrm{e}^{-3i\beta h} - a\mathrm{e}^{3i\beta h} - b\mathrm{e}^{i\beta h} + b\mathrm{e}^{-i\beta h} + a\mathrm{e}^{-3i\beta h}\right) \\ &= 1 - ki\left(a\xi\sin(3\beta h) + b\xi\sin(\beta h) + a\sin(3\beta h) + b\sin(\beta h)\right) \\ &\Rightarrow \xi + ki\xi\left(a\sin(3\beta h) + b\sin(\beta h)\right) = 1 - ki\left(a\sin(3\beta h) + b\sin(\beta h)\right) \\ &\Rightarrow \xi = \frac{1 - ki\left(a\sin(3\beta h) + b\sin(\beta h)\right)}{1 + ki\left(a\sin(3\beta h) + b\sin(\beta h)\right)}. \end{aligned}$$

Inserting values for the coefficients a and b and, once more, making use of the identity $\sin(3x) = 3\sin(x) - 4\sin^3(x)$, yields

$$\xi = \frac{1 - ki\left(\frac{1+\pi^2}{2h}\sin(\beta h) - \frac{4}{8h^3}\sin^3(\beta h)\right)}{1 + ki\left(\frac{1+\pi^2}{2h}\sin(\beta h) - \frac{4}{8h^3}\sin^3(\beta h)\right)} = \frac{1 + i\frac{k}{2h}\sin(\beta h)\left(\frac{\sin^2(\beta h)}{h^2} - (1+\pi^2)\right)}{1 - i\frac{k}{2h}\sin(\beta h)\left(\frac{\sin^2(\beta h)}{h^2} - (1+\pi^2)\right)}.$$

The modulus of ξ is thus

$$|\xi| = \frac{|1 + i\frac{k}{2h}\sin(\beta h)\left(\frac{\sin^2(\beta h)}{h^2} - (1+\pi^2)\right)|}{|1 - i\frac{k}{2h}\sin(\beta h)\left(\frac{\sin^2(\beta h)}{h^2} - (1+\pi^2)\right)|} = \left[\frac{1 + \frac{k^2}{4h^2}\sin^2(\beta h)\left(\frac{\sin^2(\beta h)}{h^2} - (1+\pi^2)\right)^2}{1 + \frac{k^2}{4h^2}\sin^2(\beta h)\left(\frac{\sin^2(\beta h)}{h^2} - (1+\pi^2)\right)^2}\right]^{\frac{1}{2}} = 1.$$

Hence, the stability criterion is fulfilled and the Crank-Nicolson method is Von Neumann stable.

b)

The analytical solution of the problem (3.31) is given by $u(x,t) = \sin(\pi(x-t))$. In this section, the complete difference schemes, including the boundary conditions, are shown for both discretizations based on the Euler method and the Crank-Nicolson method. Results from the numerical implementation with the Crank-Nicolson method (3.38) will be shown. The discretization based on the Euler method has been implemented numerically for practical testing, but, as the theory predicts, it cannot be used to approximate the analytical solution in any reasonable manner.

When studying the discretization based on Euler's method, given in equation (3.35), it becomes apparent that there are fictitious nodes in the system of equations. Using the periodic boundary condition $u(x+2,t) = u(x,t)$, the fictitious nodes can be eliminated. In particular,

$$U_0 = U_M,\ U_{-1} = U_{M-1},\ U_{-2} = U_{M-2},\ U_{-3} = U_{M-3}$$
$$\text{and likewise } U_{M+1} = U_1,\ U_{M+2} = U_2,\ U_{M+3} = U_3.$$

The terms inside the parentheses in (3.35), known as the spatial discretization, can be represented with a cyclic matrix Q. The matrix has the form

$$Q := \begin{pmatrix} 0 & -b & 0 & -a & 0 & \dots & a & 0 & b & 0 \\ b & 0 & -b & 0 & -a & 0 & \dots & a & 0 & 0 \\ 0 & b & 0 & -b & 0 & -a & 0 & \dots & a & 0 \\ a & 0 & b & 0 & -b & 0 & -a & 0 & \dots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & \dots & a & 0 & b & 0 & -b & 0 & -a \\ 0 & -a & 0 & \dots & a & 0 & b & 0 & -b & 0 \\ 0 & 0 & -a & 0 & \dots & a & 0 & b & 0 & -b \\ 0 & -b & 0 & -a & 0 & \dots & a & 0 & b & 0 \end{pmatrix},$$

and the linear system of equations (3.35) becomes

$$\boldsymbol{U}^{n+1} = (I + kQ)\boldsymbol{U}^n,$$

where

$$\boldsymbol{U}^n = [U_0^n, U_1^n, \dots, U_{M-1}^n, U_M^n]^T,$$

and I is the identity matrix matching the dimensions of Q.

Similarly, a rearrangement of the discretization based on Crank-Nicolson's method (3.38), yields the system

$$\begin{aligned} &U_m^{n+1} - \frac{k}{2}\left(-aU_{m+3}^{n+1} - bU_{m+1}^{n+1} + bU_{m-1}^{n+1} + aU_{m-3}^{n+1}\right) \\ &= U_m^n + \frac{k}{2}\left(-aU_{m+3}^n - bU_{m+1}^n + bU_{m-1}^n + aU_{m-3}^n\right),\ 0 \le m \le M, \end{aligned}$$

where the spatial discretization for $\boldsymbol{U}^{n+1}$ and $\boldsymbol{U}^n$ can be written in terms of the matrix Q. Here the periodic boundary conditions are once again used to eliminate fictitious nodes. The linear system of equation (3.38) then becomes

$$(I - \frac{k}{2}Q)\boldsymbol{U}^{n+1} = (I + \frac{k}{2}Q)\boldsymbol{U}^n.$$

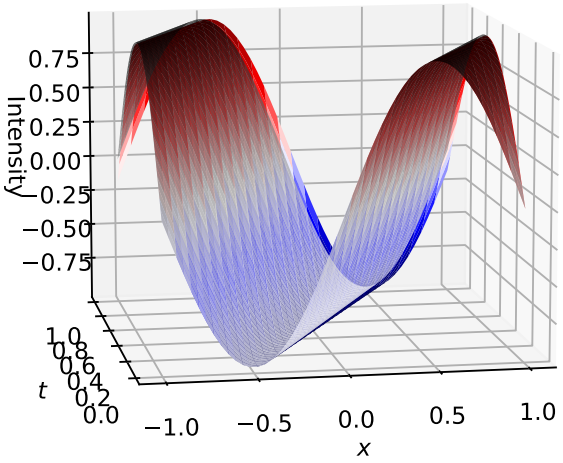
The t-axis is discretized on an equidistant grid of points, in a similar manner to the discretization of the x-axis. This reads

$$t_0 = 0,\ t_1 = \frac{T}{N}, \dots,\ t_{N-1} = \frac{T(N-1)}{N},\ t_N = T,$$

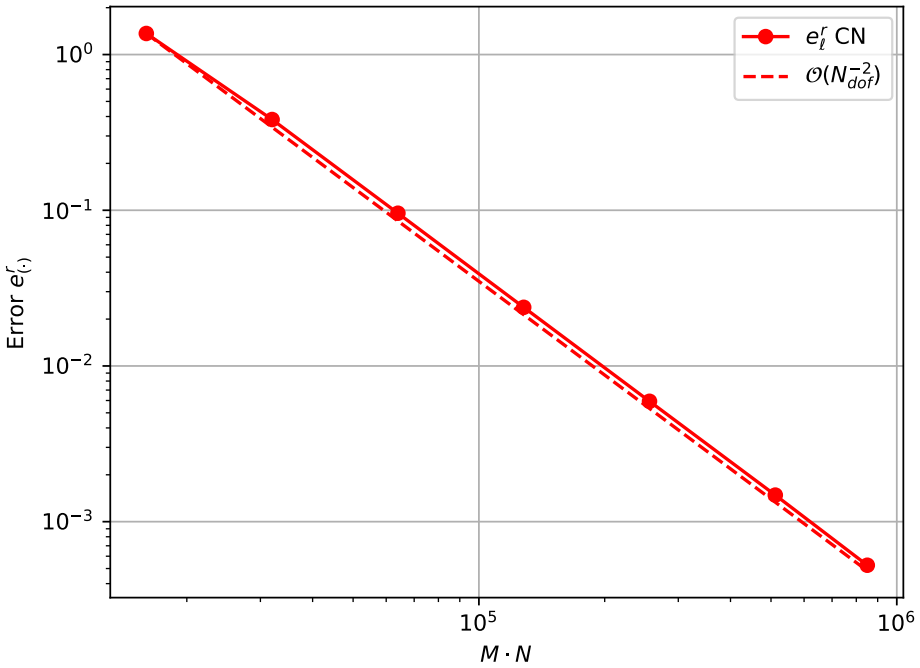
where the step length is defined as $k = \frac{T}{N}$.

The analytical and numerical solution to the linearized KdV-equation (3.31) is shown in figure 3.20a. Once again, only the numerical solution based on the Crank-Nicolson method is shown due to the fact the numerical solution with Euler is unstable.

In order to quantify the convergence of the numerical solution to the analytical solution, the relative error, defined as in equation (2.5), is calculated. The convergence plot in figure 3.20b shows that e_ℓ^r goes as $\mathcal{O}(N_{\text{dof}}^{-2})$, as expected, given the truncation error term in (3.37).



(a) Numerical and analytical solution



(b) Relative error e_ℓ^r

Figure 3.20: Linearized Korteweg-deVries equation on $x \in [-1, 1]$ and $t \in [0, 1]$, with a periodic boundary condition and initial condition $u(x, 0) = \sin(\pi x)$. In (a), the analytical solution is shown in grey and the numerical solution is plotted with a *seismic* color map. The numerical solution is calculated using Crank-Nicolson with $M = N = 50$. In (b), the relative error e_ℓ^r at $t = 1$ is plotted with the x-axis as the number of degrees of freedom. The numerical solution is calculated with Crank-Nicolson where $N = 1000$ and M increases exponentially.

c)

The following statement is proved in this section.

Statement. *The continuous L_2 norm of the analytical solution of problem* (3.31) *is conserved over time, as long as the periodic boundary condition is imposed.*

Proof. When the periodic boundary condition is imposed, the analytical solution can be expressed in a Fourier series on the form

$$u(x,t) = \sum_{k\in\mathbb{Z}} \widehat{u}(k,0)\exp\left(-i\pi k(1+\pi^2)t + i\pi^3k^3t\right)\exp\left(i\pi kx\right).$$

Using the notation $u(x,t)^*$ for the complex conjugate of $u(x,t)$, the absolute value of $u(x,t)$ can be expressed with a double summation on the form

$$\begin{aligned}
|u(x,t)|^2 &= u(x,t)^*u(x,t) \\
&= \left(\sum_{k\in\mathbb{Z}} \widehat{u}(k,0)^*\mathrm{e}^{i\pi k(1+\pi^2)t - i\pi^3k^3t}\mathrm{e}^{-i\pi kx}\right)\left(\sum_{k\in\mathbb{Z}} \widehat{u}(k,0)\mathrm{e}^{-i\pi k(1+\pi^2)t + i\pi^3k^3t}\mathrm{e}^{i\pi kx}\right) \\
&= \sum_{k\in\mathbb{Z}}\sum_{l\in\mathbb{Z}} \widehat{u}(k,0)^*\widehat{u}(l,0)\mathrm{e}^{i\pi k(1+\pi^2)t - i\pi^3k^3t}\mathrm{e}^{-i\pi l(1+\pi^2)t + i\pi^3l^3t}\mathrm{e}^{-i\pi kx}\mathrm{e}^{i\pi lx}.
\end{aligned}$$

Integrating over $x \in [-1,1]$ and using the orthogonality of Fourier basis functions, namely that $\int_{-1}^{1}\exp\left(-i\pi kx\right)\exp\left(i\pi lx\right)\mathrm{d}x = 2\delta_{k,l}$, yields

$$\begin{aligned}
\int_{-1}^{1}|u(x,t)|^2\mathrm{d}x &= \sum_{k\in\mathbb{Z}}\sum_{l\in\mathbb{Z}} \widehat{u}(k,0)^*\widehat{u}(l,0)\mathrm{e}^{i\pi k(1+\pi^2)t - i\pi^3k^3t}\mathrm{e}^{-i\pi l(1+\pi^2)t + i\pi^3l^3t}\int_{-1}^{1}\mathrm{e}^{-i\pi kx}\mathrm{e}^{i\pi lx}\mathrm{d}x \\
&= 2\sum_{k\in\mathbb{Z}} \widehat{u}(k,0)^*\widehat{u}(k,0)\mathrm{e}^{(i\pi k(1+\pi^2)t - i\pi^3k^3t)}\mathrm{e}^{-(i\pi k(1+\pi^2)t - i\pi^3k^3t)} \\
&= 2\sum_{k\in\mathbb{Z}} \widehat{u}(k,0)^*\widehat{u}(k,0).
\end{aligned}$$

In a similar manner, the integral of $|u(x,0)|^2$ can be calculated as

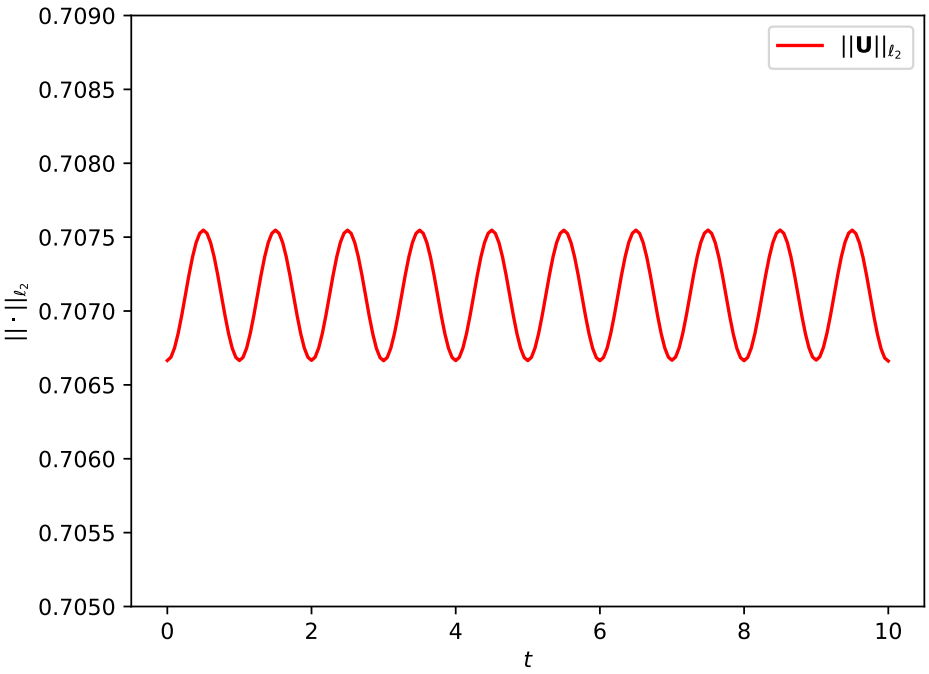
$$\begin{aligned}
\int_{-1}^{1}|u(x,0)|^2 &= \sum_{k\in\mathbb{Z}}\sum_{l\in\mathbb{Z}} \widehat{u}(k,0)^*\widehat{u}(l,0)\int_{-1}^{1}\mathrm{e}^{-i\pi kx}\mathrm{e}^{i\pi lx}\mathrm{d}x \\
&= 2\sum_{k\in\mathbb{Z}} \widehat{u}(k,0)^*\widehat{u}(k,0).
\end{aligned}$$

Observe that the two integrals are identical to each other. The definition of the L_2 norm gives

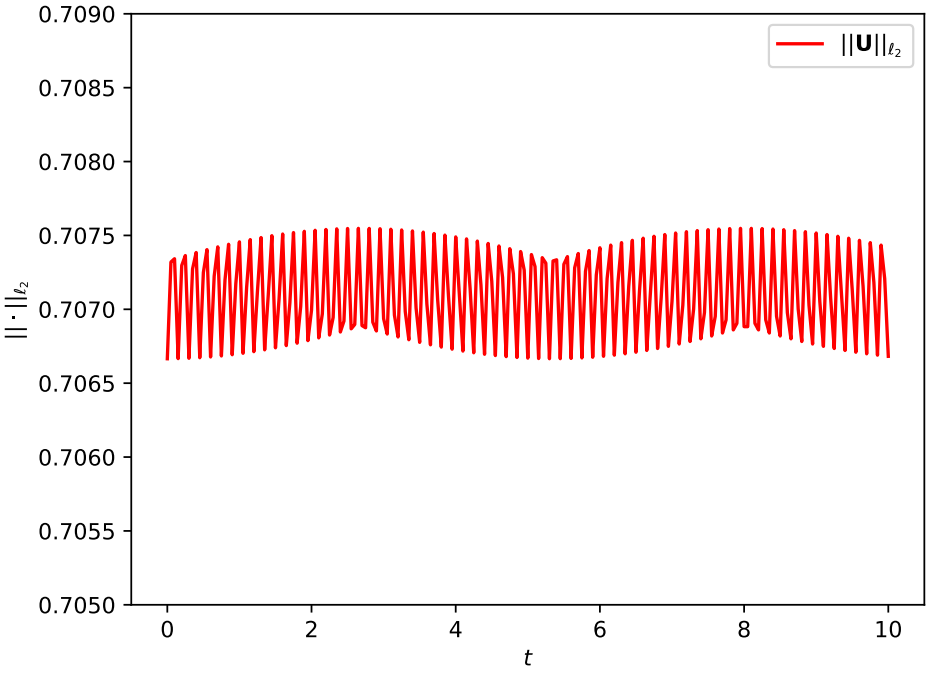
$$\begin{aligned}
||u(x,t)|| &:= \sqrt{\frac{1}{2}\int_{-1}^{1}|u(x,t)|^2\mathrm{d}x} = \sqrt{\frac{1}{2}\int_{-1}^{1}|u(x,0)|^2\mathrm{d}x} \\
&= \sqrt{\sum_{k\in\mathbb{Z}}|\widehat{u}(k,0)|^2}.
\end{aligned}$$

Hence, the L_2 norm is conserved for any time $t > 0$. □

Numerically, the conservation of the L_2 norm can be observed by calculating the discrete ℓ_2 norm and using it as an approximation of the continuous L_2 norm. The ℓ_2 norm is calculated as a function of time for two different initial conditions, by using the discretization method based on the Crank-Nicolson method (3.38). The first case is with $u(x,0) = \sin(\pi x)$, as in the original problem (3.31), and the second case is with $u(x,0) = \sin(2\pi x)$. The plots of the discrete ℓ_2 norm against time are shown in the figures 3.21a and 3.21b. Observe that the ℓ_2 norm is oscillating around a fixed value and not drifting with time, which is in agreement with the discussion above.



(a) Initial condition $u(x, 0) = \sin(\pi x)$



(b) Initial condition $u(x, 0) = \sin(2\pi x)$

Figure 3.21: Linearized Korteweg-deVries equation on $x \in [-1, 1]$ and $t \in [0, 10]$, with a periodic boundary condition with period 2. The discrete ℓ_2 norm of the numerical solution as a function of time, with initial condition $u(x, 0) = \sin(\pi x)$, is shown in (a) and with initial condition $u(x, 0) = \sin(2\pi x)$ in (b). The numerical solution is calculated using the Crank-Nicolson method.

3.5 Problem 5 - Poisson Equation in One Dimension

a)

Consider the Poisson equation in 1D

$$-u_{xx} = f(x), \;\; u(a) = d_1, \; u(b) = d_2. \tag{3.40}$$

Assume a uniform partition, i.e. that $x_m = a + mh$, $h = \frac{b-a}{M}$ and $m \in [0, M]$. By discretizing the problem with linear finite elements, a linear system $\mathbf{Au} = \mathbf{f}$ can be created. Since the problem has inhomogeneous Dirichlet boundary conditions, the approach outlined in section 2.5 of [1] will be followed, in order to create the system. As is done in Curry's note, the inhomogeneous problem is related to the corresponding homogeneous problem. The weak form is therefore written with test functions satisfying $v \in H_0^1([a, b])$, i.e. $v(a) = v(b) = 0$. The weak form becomes

$$\begin{aligned} -\int_a^b u''(x)v(x)\mathrm{d}x &= \int_a^b f(x)v(x)\mathrm{d}x, \\ \int_a^b u'(x)v'(x)\mathrm{d}x - [u'(x)v(x)]_a^b &= \int_a^b f(x)v(x)\mathrm{d}x, \\ \int_a^b u'(x)v'(x)\mathrm{d}x &= \int_a^b f(x)v(x)\mathrm{d}x, \end{aligned} \tag{3.41}$$

where u is an element of $H^1([a, b])$. Next, $R_h \in H^1([1, b])$ is introduced, satisfying $R_h(a) = d_1$ and $R_h(b) = d_2$. Then $\hat{u} := u - R_h \in H_0^1([a, b])$ is defined, so the variational (3.41) form can be rewritten as

$$\int_a^b \hat{u}'(x)v'(x)\mathrm{d}x = \int_a^b f(x)v(x)\mathrm{d}x - \int_a^b R_h'(x)v'(x)\mathrm{d}x. \tag{3.42}$$

Using linear finite elements, the functions in (3.42) are now restricted to lie in the finite dimensional subspace X_h^1 of $H^1([a, b])$, i.e. the space of piece-wise linear functions with basis $\{\varphi_i\}$, as defined in [1]. The grid is partitioned into M elements, such that $v = \sum_{i=0}^{M} v_i\varphi_i(x)$ and likewise for u and $\hat{u}$. R_h takes the natural form $R_h = d_1\varphi_0(x)+d_2\varphi_M(x)$. Introducing the stiffness matrix $A_{ij} = \int_a^b \varphi_i'(x)\varphi_j'(x)\mathrm{d}x$ such that

$$A = \frac{1}{h}\begin{pmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \ddots & & \vdots \\ 0 & -1 & 2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(M+1)\times(M+1)},$$

and removing v, equation (3.42) becomes

$$A\hat{u} = F - A(d_1, 0, \ldots, 0, d_2)^T, \tag{3.43}$$

where F is initially zero and then constructed by adding

$$\frac{1}{x_{k+1}-x_k}\begin{pmatrix}\int_{x_k}^{x_{k+1}}(x_{k+1}-x)f(x)\mathrm{d}x\\ \int_{x_k}^{x_{k+1}}(x-x_k)f(x)\mathrm{d}x\end{pmatrix}$$

to the sub-vector $[F_j]_{j=k,k+1}$, where $k=0,\ldots,M-1$. F is approximated with Gaussian quadrature, which results in

$$\begin{aligned}A\hat{u} &= F - A(d_1,0,\ldots,0,d_2)^T \\ &\approx \frac{1}{h}\begin{pmatrix} Q_0[(x_1-x)f(x)] \\ Q_0[(x-x_0)f(x)]+Q_1[(x_2-x)f(x)] \\ \vdots \\ Q_{M-2}[(x-x_{M-2})f(x)]+Q_{M-1}[(x_M-x)f(x)] \\ Q_{M-1}[(x-x_{M-1})f(x)]\end{pmatrix} - \frac{1}{h}\begin{pmatrix} d_1\\ -d_1\\ \vdots\\ -d_2\\ d_2\end{pmatrix},\end{aligned} \tag{3.44}$$

where $Q_k[g(x)]$ denotes the Gaussian quadrature of the function g over the interval $[x_k, x_{k+1}]$. In the numerical implementation, we have used Gauss-Legendre quadrature with 5 sample points and weights. The boundary conditions are implemented by removing the first and last row and column from the stiffness matrix A on the left side of equation (3.44), in addition to the first and last entries of the vector $F - A(d_1,\ldots,d_2)^T$. The resulting matrix and vector are given as

$$\begin{aligned}\mathbf{A} &:= \frac{1}{h}\begin{pmatrix}-2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1\\ & & -1 & 2\end{pmatrix} \in \mathbb{R}^{(M-1)\times(M-1)}, \\ \mathbf{f} &:= \begin{pmatrix} Q_0[(x-x_0)f(x)]+Q_1[(x_2-x)f(x)]+d_1/h \\ \vdots \\ Q_{M-2}[(x-x_{M-2})f(x)]+Q_{M-1}[(x_M-x)f(x)]+d_2/h\end{pmatrix} \in \mathbb{R}^{(M-1)\times 1},\end{aligned} \tag{3.45}$$

respectively. $\mathbf{u} := \hat{u}$ is also defined, so the system can be written as $\mathbf{Au} = \mathbf{f}$. Finally, when this system is solved, the vector $\hat{u}$ is expanded from dimension $M-1$ to $M+1$ by including d_1 as first entry and d_2 as the last, i.e. let $u(x) = \hat{u}(x) + R_h(x)$.

b)

Now consider $0 \le x \le 1$ with

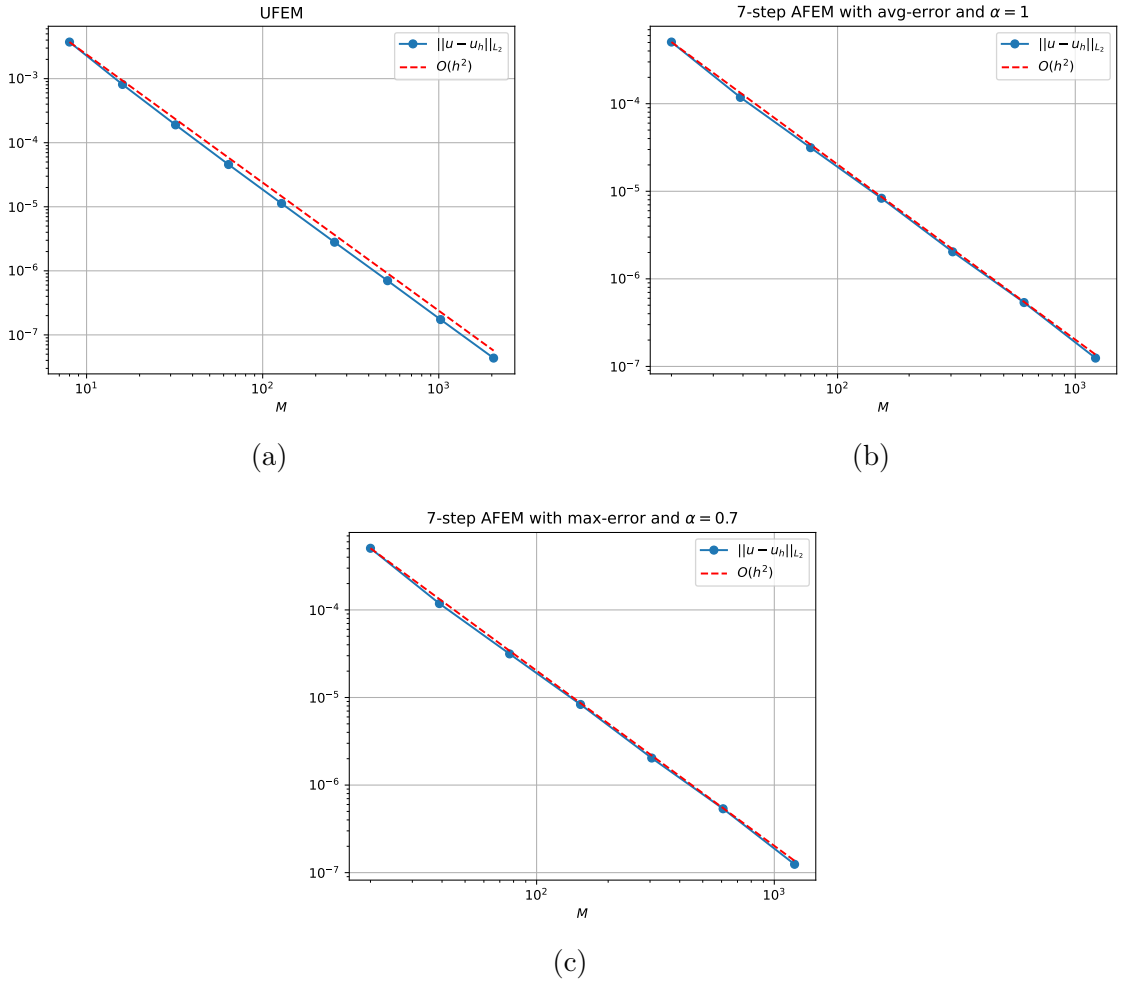
$$f(x) = -2, \quad d_1 = 0,\ d_2 = 1. \tag{3.46}$$

First, the analytical solution is calculated. The Poisson equation reads $-u_{xx} = -2$, which yields

$$u(x) = x^2 + c_1 x + c_2,$$

where $c_1, c_2 \in \mathbb{R}$ are constants. The boundary conditions give $c_1 = c_2 = 0$, resulting in

$$u(x) = x^2.$$



(a) (b)

(c)

Figure 3.22: One dimensional Poisson equation $-u_{xx} = -2$ with $x \in [0, 1]$ and $u(0) = 0, u(1) = 1$. The L^2-error is displayed in "log-log" plots. u_h is the numerical solution while u is the analytical solution. (a) UFEM. (b) AFEM with average-error and $\alpha = 1$. (c) AFEM with maximum-error and $\alpha = 0.7$.

When using the finite element method, both uniform mesh refinement (hereafter UFEM) and adaptive mesh refinement (hereafter AFEM) is performed. In the rest of the problem, when performing UFEM, $M = \{8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$ will be used. On the contrary, AFEM will be performed using *average*-error with $\alpha = 1$, and *max*-error with $\alpha = 0.7$ as it is defined in section 2.4.

Convergence plots, where the difference between analytical and numerical solution in L_2 norm is calculated, for both UFEM and AFEM, are depicted in figure 3.22. It is apparent that all methods converge with order $\mathcal{O}(h^2)$ in L_2 norm. It can be shown that this is the expected theoretical result [1].

c)

Next, consider $-1 \leq x \leq 1$ with

$$f(x) = -(40000x^2 - 200)e^{-100x^2}, \quad d_1 = e^{-100}, \; d_2 = e^{-100}. \tag{3.47}$$

First, the analytical solution is calculated. The Poisson equation reads

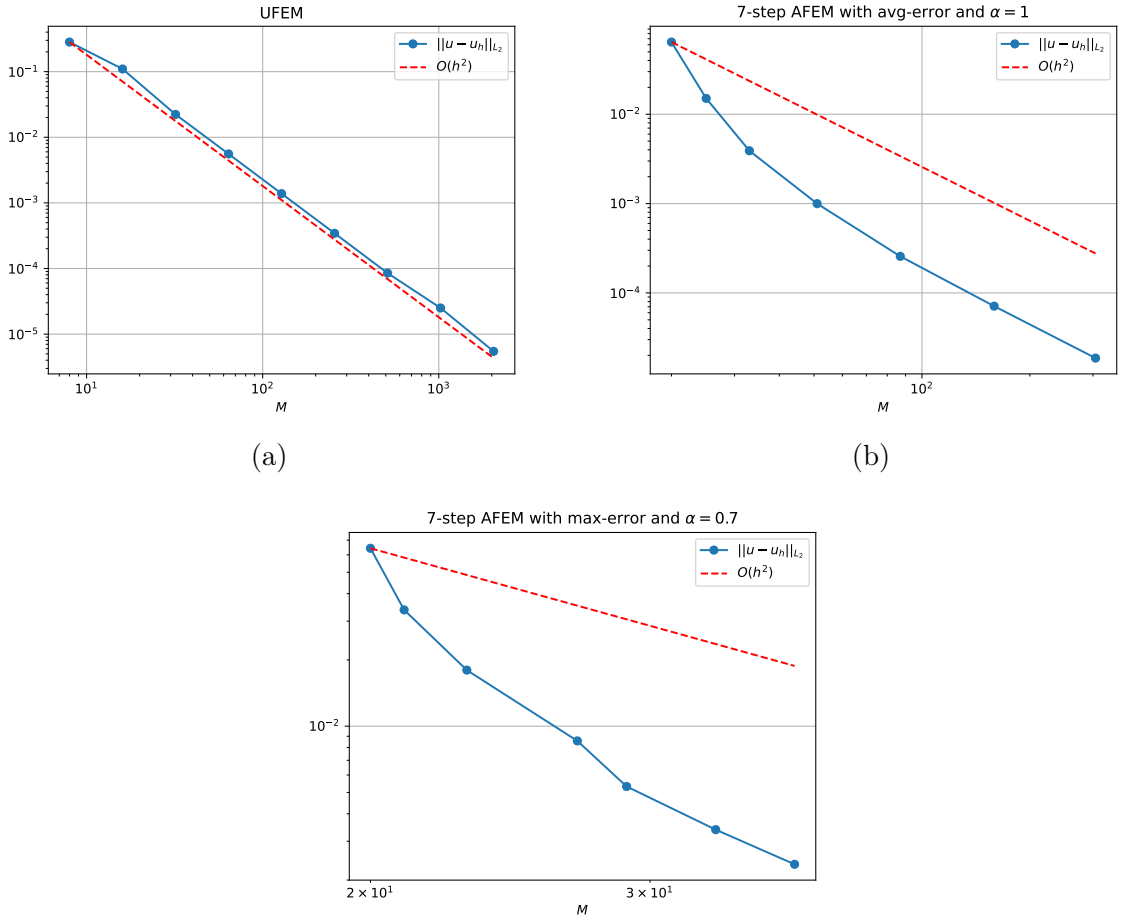


Figure 3.23: One dimensional Poisson equation $-u_{xx} = -(40000x^2 - 200)e^{-100x^2}$ with $x \in [-1, 1]$ and $u(-1) = u(1) = e^{-100}$. The L^2-error is displayed in "log-log" plots. u_h is the numerical solution while u is the analytical solution. (a) UFEM. (b) AFEM with average and $\alpha = 1$. (c) AFEM with maximum and $\alpha = 0.7$.

$$u_{xx} = 200e^{-100x^2} - 40000x^2 \cdot e^{-100x^2},$$

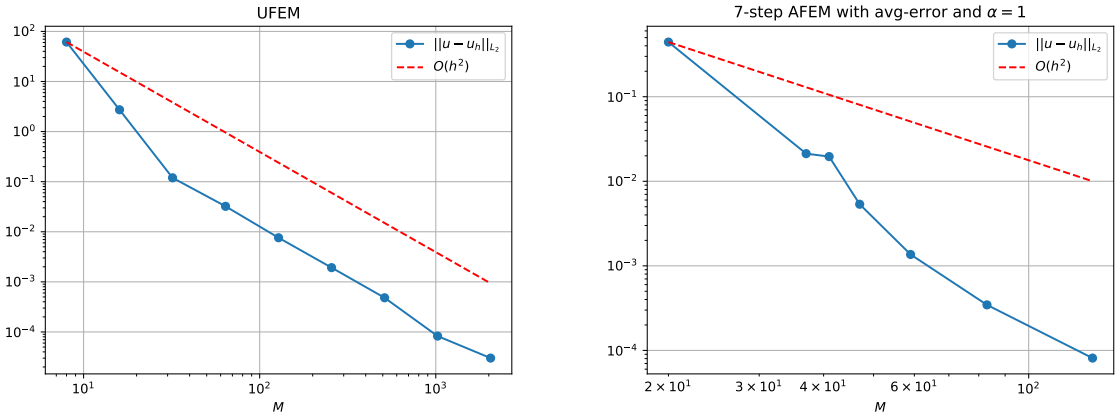
which gives

$$u(x) = e^{-100x^2} + c_1 x + c_2,$$

where $c_1, c_2 \in \mathbb{R}$ are constants. The boundary conditions readily give that $c_1 = c_2 = 0$, which means the analytical solution is

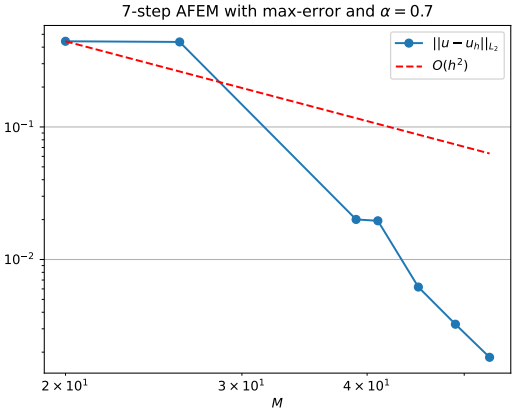
$$u(x) = e^{-100x^2}.$$

The convergence plots for UFEM and AFEM, in this case, are depicted in figure 3.23. It is apparent that all methods converge with order $\mathcal{O}(h^2)$ in L_2 norm, where (c) approximately reaches $\mathcal{O}(h^2)$ towards the end of the plotted interval.



(a)

(b)



(c)

Figure 3.24: One dimensional Poisson equation $-u_{xx} = -(4000000x^2 - 2000)e^{-1000x^2}$ with $x \in [-1, 1]$ and $u(-1) = u(1) = e^{-1000}$. The L^2-error is displayed in "log-log" plots. u_h is the numerical solution while u is the analytical solution. (a) UFEM. (b) AFEM with average and $\alpha = 1$. (c) AFEM with maximum and $\alpha = 0.7$.

d)

Now consider $-1 \leq x \leq 1$ with

$$f(x) = -(4000000x^2 - 2000)e^{-1000x^2}, \quad d_1 = e^{-1000},\ d_2 = e^{-1000}. \tag{3.48}$$

The derivation of the analytical solution is entirely analogous the one in **c)**. Thus,

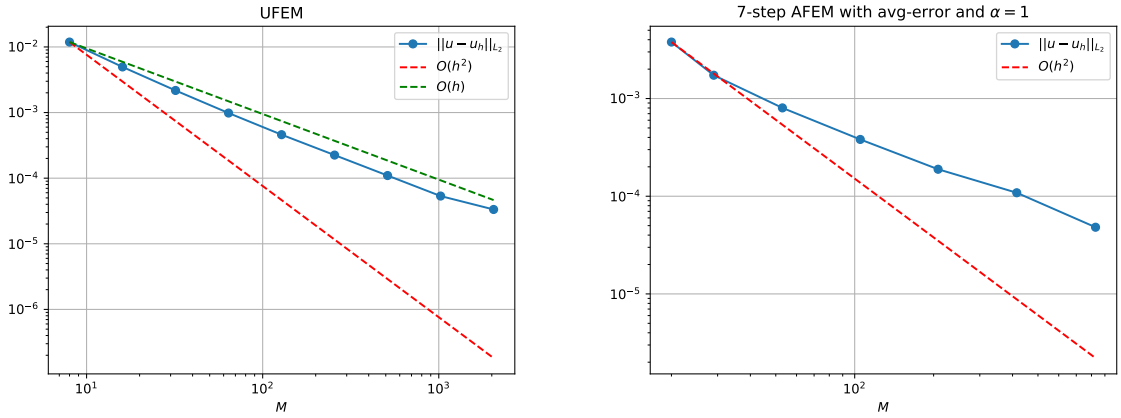
$$u(x) = e^{-1000x^2}.$$

The convergence plots for UFEM and AFEM are depicted in figure 3.24. It is apparent that all methods converge with order of at least $\mathcal{O}(h^2)$ in L_2 norm.

e)

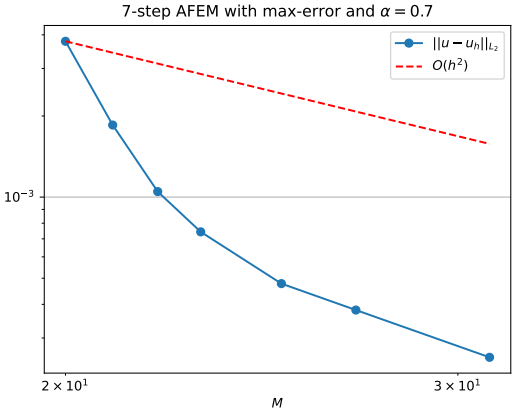
Finally, consider $0 \leq x \leq 1$, with

$$f(x) = \frac{2}{9}x^{-4/3}, \quad d_1 = 0,\ d_2 = 1 \tag{3.49}$$



(a)

(b)



(c)

Figure 3.25: One dimensional Poisson equation $-u_{xx} = -\frac{2}{9}x^{-4/3}$ with $x \in [0, 1]$ and $u(0) = 0, u(1) = 1$. The L^2-error is displayed in "log-log" plots. u_h is the numerical solution while u is the analytical solution. (a) UFEM. (b) AFEM with average and $\alpha = 1$. (c) AFEM with maximum and $\alpha = 0.7$.

Integration yields that u must be on the form $u(x) = x^{2/3} + c_1 x + c_2$, where $c_1, c_2 \in \mathbb{R}$ are constants. The boundary conditions give $c_1 = 0$ and $c_2 = 0$. Hence, the analytical solution is

$$u(x) = x^{2/3}.$$

The convergence plots for UFEM and AFEM are depicted in figure 3.25. Observe that UFEM yields a convergence of order $\mathcal{O}(h)$ in this case. One possible explanation for this is that the derivative of the analytical solution $u(x)$ are not defined for $x = 0$. For example, it is assumed that the variational form, written as

$$\int_0^1 u'(x)v'(x)\mathrm{d}x = \int_0^1 f(x)v(x)\mathrm{d}x \tag{3.50}$$

should hold for all $v \in H_0^1([0, 1])$, where $v(0) = v(1) = 0$. Letting $v(x) = \sin(\pi x)$, the right hand side of (3.50) becomes

$$\int_0^1 f(x)v(x)\mathrm{d}x = \int_0^1 \frac{2}{9}x^{-\frac{4}{3}} \cdot \sin(\pi x)\mathrm{d}x$$
$$= \left[\frac{2}{9}x^{-\frac{4}{3}} \cdot (-\frac{1}{\pi}\cos(\pi x))\right]_0^1 - \int_0^1 \frac{8}{27}x^{-\frac{7}{3}} \cdot \frac{1}{\pi}\cos(\pi x)\mathrm{d}x,$$

where it is readily seen that the first term is not defined for $x = 0$. AFEM with average error also seems to have a lower convergence rate compared to the other examples, while AFEM with max-error seems to uphold the convergence of rate $\mathcal{O}(h^2)$.

Chapter 4

Part 2

4.1 Problem 2 - Sine-Gordon Equation

Consider the Sine-Gordon equation on $\Omega \times I$, where $\Omega = [a, b]$ and $I = [0, T]$

$$
\begin{aligned}
&u_{tt} - u_{xx} + \sin(u) = 0, && (x,t) \in \Omega \times I, && (4.1)\\
&u(a,t) = f_1(t),\ u(b,t) = f_2(t), && (x,t) \in \partial\Omega \times I, && (4.2)\\
&u(x,0) = u_0(x),\ u_t(x,0) = u_1(x), && x \in \Omega \times \{t = 0\}. && (4.3)
\end{aligned}
$$

a)

One possible analytical solution to the Sine-Gordon equation is derived in the following. By introducing the change of variables $s = x - ct$, equation (4.1) can be expressed differently, because

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \frac{\partial u}{\partial s}\frac{\partial s}{\partial t} = -c\frac{\partial u}{\partial s},\\
\frac{\partial^2 u}{\partial t^2} &= \frac{\partial}{\partial t}\left(-c\frac{\partial u}{\partial s}\right) = -c\frac{\partial}{\partial s}\frac{\partial u}{\partial t} = c^2\frac{\partial^2 u}{\partial s^2},\\
\frac{\partial u}{\partial x} &= \frac{\partial u}{\partial s}\frac{\partial s}{\partial x} = \frac{\partial u}{\partial s},\\
\frac{\partial^2 u}{\partial x^2} &= \frac{\partial}{\partial x}\frac{\partial u}{\partial s} = \frac{\partial}{\partial s}\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial s^2}.
\end{aligned}
$$

This leads to the Sine-Gordon equation (4.1) on the following form

$$\left(1 - c^2\right) u_{ss} = \sin(u).$$

Multiplying the equation by u_s and performing integration by parts yields

$$
\begin{aligned}
(1-c^2)\int u_{ss}u_s \mathrm{d}s &= (1-c^2)\left((u_s)^2 - \int u_{ss}u_s \mathrm{d}s\right) + D_1,\\
\implies (1-c^2)\int u_{ss}u_s \mathrm{d}s &= (1-c^2)\frac{1}{2}(u_s)^2 + D_2,
\end{aligned}
$$

where $D_2 = \frac{1}{2}D_1$, for the left hand side and

$$\int \sin(u)u_s \mathrm{d}s = -\cos(u) + D_3,$$

for the right hand side. Hence,

$$\frac{1}{2}(1-c^2)u_s^2 = -\cos(u) + D,$$

where $D = D_3 - D_2$. The constant D determines one unique solution to the problem. Let $D = 1$ in this case. Thus, note that

$$\begin{aligned} u_s^2 &= \frac{2(1-\cos(u))}{(1-c^2)} \\ u_s^2 &= \frac{4\sin^2\left(\frac{u}{2}\right)}{(1-c^2)} \\ \frac{u_s}{\sin\left(\frac{u}{2}\right)} &= \pm\frac{2}{\sqrt{1-c^2}}, \end{aligned}$$

which, by integrating again, yields

$$\begin{aligned} \frac{\mathrm{d}u}{\sin\left(\frac{u}{2}\right)} &= \pm\frac{2}{\sqrt{1-c^2}}\mathrm{d}s \\ \ln\tan\frac{u}{4} &= \pm\frac{s}{\sqrt{1-c^2}} + C \\ u(s) &= 4\arctan\left(\exp\left\{\pm\frac{s}{\sqrt{1-c^2}}\right\}\right), \end{aligned}$$

where the integration constant C is set to 0. This means that one possible analytical solution to the Sine-Gordon equation (4.1) is

$$u(x,t) = 4\arctan\left(\exp\left\{\pm\frac{x-ct}{\sqrt{1-c^2}}\right\}\right), \tag{4.4}$$

when the integration constants are set to 0 and 1.

b)

The following statement is proved in this section.

Statement. *The energy is conserved when* $u_x(a,t) = u_x(b,t) = 0$.

Proof. The energy is the quantity which is obtained after multiplying equation (4.1) by u_t and integrating with respect to both x and t. Before integrating, note the following equivalence relation

$$\begin{aligned} & u_t u_{tt} - u_t u_{xx} + u_t \sin(u) = 0 \\ \iff & \frac{\partial}{\partial t}\left(\frac{1}{2}\left(\frac{\partial u}{\partial t}\right)^2 + \frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2 - \cos(u)\right) - \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\frac{\partial u}{\partial x}\right) = 0. \end{aligned} \tag{4.5}$$

This can be verified by noting that

$$\frac{\partial}{\partial t}\left(\frac{1}{2}\left(\frac{\partial u}{\partial t}\right)^2+\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2-\cos(u)\right)-\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\frac{\partial u}{\partial x}\right)=0$$
$$\frac{\partial^2 u}{\partial t^2}\frac{\partial u}{\partial t}+\frac{\partial^2 u}{\partial t\partial x}\frac{\partial u}{\partial x}+\frac{\partial u}{\partial t}\sin(u)-\left(\frac{\partial u}{\partial t}\frac{\partial^2 u}{\partial x^2}+\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial t\partial x}\right)=0.$$

Hence, integration of (4.5) yields

$$\begin{aligned}
0 &= \int_0^{t'}\int_a^b (u_t u_{tt}-u_t u_{xx}+u_t\sin(u))\,\mathrm{d}x\mathrm{d}t\\
&= \int_0^{t'}\int_a^b \left\{\frac{\partial}{\partial t}\left(\frac{1}{2}\left(\frac{\partial u}{\partial t}\right)^2+\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2-\cos(u)\right)-\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\frac{\partial u}{\partial x}\right)\right\}\mathrm{d}x\mathrm{d}t,\\
&= \int_0^{t'}\int_a^b \frac{\partial}{\partial t}\left(\frac{1}{2}\left(\frac{\partial u}{\partial t}\right)^2+\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2-\cos(u)\right)\mathrm{d}x\mathrm{d}t-\int_0^{t'}\int_a^b \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\frac{\partial u}{\partial x}\right)\mathrm{d}x\mathrm{d}t.
\end{aligned} \tag{4.6}$$

Note that

$$\int_a^b \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\frac{\partial u}{\partial x}\right)\mathrm{d}x = u_t(b,t^*)u_x(b,t^*)-u_t(a,t^*)u_x(a,t^*)=0,$$

when $u_x(b,t^*)=u_x(a,t^*)=0$, $\forall t^*\in I$ holds. This means that the second integral in equation (4.6) evaluates to zero. Hence, the first integral must also evaluate to zero, which yields

$$\begin{aligned}
0 &= \int_0^{t'}\int_a^b \frac{\partial}{\partial t}\left(\frac{1}{2}u_t^2+\frac{1}{2}u_x^2-\cos(u)\right)\mathrm{d}x\mathrm{d}t\\
&= \oint_{\mathcal{C}}\left(-\frac{1}{2}u_t^2-\frac{1}{2}u_x^2+\cos(u)\right)\mathrm{d}x \quad \text{(Green's Theorem)}\\
&= -\left[\int_a^b\left(-\frac{1}{2}u_t^2-\frac{1}{2}u_x^2+\cos(u)\right)\mathrm{d}x\right]_{t=t'}+\left[\int_a^b\left(-\frac{1}{2}u_t^2-\frac{1}{2}u_x^2+\cos(u)\right)\mathrm{d}x\right]_{t=0}\\
&= \left[\int_a^b E_x\mathrm{d}x\right]_{t=t'}-\left[\int_a^b E_x\mathrm{d}x\right]_{t=0}\\
&= E(t')-E(0),
\end{aligned}$$

where $E_x(x,t):=\frac{1}{2}u_t^2+\frac{1}{2}u_x^2-\cos(u)$, $E(t'):=\left[\int_a^b E_x\mathrm{d}x\right]_{t=t'}$ and $\mathcal{C}$ is the positively (anticlockwise) oriented boundary of the region $\{(x,t)\mid a\le x\le b, 0\le t\le t'\}$. This shows that the energy E is conserved, since $E(t')=E(0)$. □

In the literature, e.g. [8], the energy density is usually defined as $E_x(x,t)=\frac{1}{2}u_t^2+\frac{1}{2}u_x^2+1-\cos(u)$. This does not contradict our derivation, because an arbitrary constant could have been included inside the parenthesis in the equivalence relation (4.5). Therefore, this convention will be used in the following, in order to ensure that the energy is a positive quantity.

c)

The Sine-Gordon equation (4.1) is discretized using the principle of semi-discretization [7]. Two systems of ODEs are developed; one of first order, and one of second order, in time.

To begin with, the x-axis is discretized in the following manner

$$x_0 = a,\ x_1 = a + \frac{b-a}{M+1}, \ldots, x_M = a + \frac{M(b-a)}{M+1}, x_{M+1} = b,$$

letting $h = \frac{b-a}{M+1}$. A central finite difference approximation is used to discretize the spatial derivative. Let $u_m := u(x_m, t)$, where the dependence on time of u_m is implicit. The stencil is then on the form

$$(u_m)_{xx} = \frac{1}{h^2}\delta^2 u_m = \frac{1}{h^2}(u_{m-1} - 2u_m + u_{m+1}) + \mathcal{O}(h^2), \quad 1 \leq m \leq M,$$

Inserting the stencil for the spatial derivative into the Sine-Gordon equation (4.1) gives

$$\ddot{u}_m = \frac{1}{h^2}\delta^2 u_m - \sin(u_m) + \mathcal{O}(h^2),$$

where the quantity $\ddot{u}_m$ denotes the second order derivative of u_m with respect to time. Note that this is a set of ordinary differential equations (ODE) along lines parallel to the t-axis and across the x-axis at the grid points x_m. Next, the approximate solutions $v_m := v_m(t) \approx u_m(t)$ are introduced, where the truncation error is neglected, by requiring that

$$\ddot{v}_m = \frac{1}{h^2}\delta^2 v_m - \sin(v_m),\ 1 \leq m \leq M.$$

Considering the Dirichlet boundary conditions (4.2), namely that $u(a,t) = v_0(t) = f_1(t)$ and $u(b,t) = v_{M+1}(t) = f_2(t)$, the ODEs can be reformulated in the following manner

$$\ddot{\boldsymbol{v}} = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \ldots & 0 \\ 1 & -2 & 1 & 0 & \ldots \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ldots & 1 & -2 & 1 \\ 0 & \ldots & 0 & 1 & -2 \end{pmatrix}\boldsymbol{v} + \begin{pmatrix} \frac{1}{h^2}f_1(t) - \sin(v_1) \\ -\sin(v_2) \\ \vdots \\ -\sin(v_{M-1}) \\ \frac{1}{h^2}f_2(t) - \sin(v_M) \end{pmatrix} \tag{4.7}$$

where $\boldsymbol{v} = [v_1, v_2, \ldots, v_M]^T$ and $\ddot{\boldsymbol{v}} = [\ddot{v}_1, \ddot{v}_2, \ldots, \ddot{v}_M]^T$. In compact notation, the system of equations (4.7) can be written as

$$\ddot{\boldsymbol{v}} = \frac{1}{h^2}A\boldsymbol{v} + \boldsymbol{g}(t, \boldsymbol{v}). \tag{4.8}$$

This is a system of equations of second order in time and will be used later, in combination with Runge-Kutta-Nyström methods defined in **e)**, to solve the Sine-Gordon equation numerically.

By introducing $\boldsymbol{w} = \dot{\boldsymbol{v}}$ and the vector $\boldsymbol{y} = [\boldsymbol{v}^T, \boldsymbol{w}^T]^T$, equation (4.8) can be reformulated as a first order system of differential equations in time

$$\dot{\boldsymbol{y}} = \begin{pmatrix} \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{w}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{w} \\ \frac{1}{h^2}A\boldsymbol{v} + \boldsymbol{g}(t, \boldsymbol{v}) \end{pmatrix}. \tag{4.9}$$

This system of equations will be used in combination with Runge-Kutta methods defined in **d)** to find numerical solutions to the Sine-Gordon equation.

d)

Runge-Kutta (RK) integrators can be used to solve an initial value problem on the form

$$\dot{\boldsymbol{y}} = f(t, \boldsymbol{y}),\ \boldsymbol{y}(t_0) = \boldsymbol{y}^0.$$

Comparing to the system of equations (4.9) and using the initial conditions (4.3) yields

$$\dot{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{w} \\ \frac{1}{h^2} A\boldsymbol{v} + \boldsymbol{g}(t, \boldsymbol{v}) \end{pmatrix} =: F(t, \boldsymbol{y}),\ \boldsymbol{y}(0) = \boldsymbol{y}^0 = \begin{pmatrix} \boldsymbol{v}^0 \\ \boldsymbol{w}^0 \end{pmatrix},$$

where $\boldsymbol{v}^0 = \boldsymbol{v}|_{t=0} = [u_0(x_1), \ldots, u_0(x_M)]^T$ and $\boldsymbol{w}^0 = \boldsymbol{w}|_{t=0} = [u_1(x_1), \ldots, u_1(x_M)]^T$, where $u_0(x) = u(x, 0)$ and $u_1(x) = u_t(x, 0)$.

The explicit Runge-Kutta methods RK2, RK3 and RK4 are used in our implementation. It is given that these methods are of second, third and fourth order, in accordance with their names. This means that the total accumulated error after using the iterative method along the time grid is of order $\mathcal{O}(k^2)$, $\mathcal{O}(k^3)$ and $\mathcal{O}(k^4)$ for RK2, RK3 and RK4, respectively. Let $\boldsymbol{Y}^n \approx \boldsymbol{y}(t_n)$ be the numerical solutions produced by the integrators at time $t = t_n$. A step with RK2 takes the form

$$\begin{aligned} s_1 &= F(t, \boldsymbol{Y}^n) \\ s_2 &= F(t + k, \boldsymbol{Y}^n + k s_1) \\ \boldsymbol{Y}^{n+1} &= \boldsymbol{Y}^n + \frac{k}{2}(s_1 + s_2), \end{aligned}$$

where the calculations are repeated iteratively for $0 \leq n \leq N - 1$. The iterative scheme is run until $t_N = T$, so the step length in the time direction may be defined as $k = \frac{T}{N}$. Similarly, a step with RK3 and RK4 take the form

RK3	RK4
$s_1 = F(t, \boldsymbol{Y}^n)$	$s_1 = F(t, \boldsymbol{Y}^n)$
$s_2 = F(t + \frac{k}{2}, \boldsymbol{Y}^n + \frac{k}{2} s_1)$	$s_2 = F(t + \frac{k}{2}, \boldsymbol{Y}^n + \frac{k}{2} s_1)$
$s_3 = F(t + k, \boldsymbol{Y}^n - k s_1 + 2k s_2)$	$s_3 = F(t + \frac{k}{2}, \boldsymbol{Y}^n + \frac{k}{2} s_2)$
	$s_4 = F(t + k, \boldsymbol{Y}^n + k s_3)$
$\boldsymbol{Y}^{n+1} = \boldsymbol{Y}^n + \frac{k}{6}(s_1 + 4s_2 + s_3)$	$\boldsymbol{Y}^{n+1} = \boldsymbol{Y}^n + \frac{k}{6}(s_1 + 2s_2 + 2s_3 + s_4).$

In order to perform h- and k-refinement, the analytical solution, as well as a domain, needs to be specified for the three RK-integrators. The analytical solution is chosen to be equation (4.4) with a minus sign in the exponential function. Moreover, the constant in the change of variables is set to $c = \frac{1}{2}$. Furthermore, the domain that is studied is $\Omega \times I = [-5, 5] \times [0, 5]$. Hence, the analytical solution is

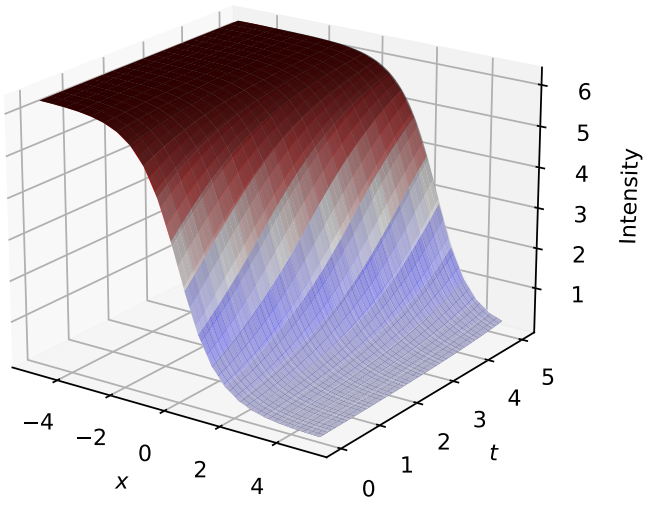


Figure 4.1: Sine-Gordon equation solved on $x \in [-5, 5]$ and $t \in [0, 5]$ with Dirichlet boundary conditions and initial condition $u(x, 0) = 4 \arctan\left(\exp\left\{-\frac{2x}{\sqrt{3}}\right\}\right)$. The numerical solution is calculated using RK4 with $M = N = 20$ and plotted in a *seismic* color map. The analytical solution is plotted in grey.

$$u(x,t) = 4 \arctan\left(\exp\left\{-\frac{2x-t}{\sqrt{3}}\right\}\right), \tag{4.10}$$

where the associated boundary and initial conditions, (4.2) and (4.3), are

$$\begin{aligned} f_1(t) &= 4 \arctan\left(\exp\left\{\frac{10+t}{\sqrt{3}}\right\}\right), \quad f_2(t) = 4 \arctan\left(\exp\left\{-\frac{10-t}{\sqrt{3}}\right\}\right), \\ u_0(x) &= 4 \arctan\left(\exp\left\{-\frac{2x}{\sqrt{3}}\right\}\right), \quad u_1(x) = \frac{4 \exp\left\{-\frac{2x}{\sqrt{3}}\right\}}{\sqrt{3}\left(1 + \exp\left\{-\frac{4x}{\sqrt{3}}\right\}\right)}. \end{aligned} \tag{4.11}$$

The numerical solution to the Sine-Gordon equation, in addition to the analytical solution (4.10), is shown in figure 4.1. Here, the numerical solution is calculated using RK4 on a mesh grid with $M = N = 20$.

Next, h- and k-refinement for the three RK-integrators is considered. Due to the CFL-condition, there is a constraint on the fraction k/h. This constraint depends on the chosen integrator. It prevents us from performing k-refinement in the usual way, by setting M to a large value and increasing N from a small value. To avoid this problem, a reference solution is used, namely a numerical solution calculated on a mesh grid with parameters M_{ref} and N_{ref}. Note that this reference solution replaces the analytical solution when computing the relative error. The refinement along the time direction is

thus performed by finding numerical solutions with $M = M_{ref}$ and increasing N while ensuring that $N << N_{ref}$. Then, the relative error with respect to the reference solution can be computed. We set $M = M_{ref}$ so that the error along the x-axis for the numerical solutions and the reference solution cancel each other out. This way, the observed error is a consequence of error in time and the CFL-condition is still fulfilled, which means that the solutions are stable. This approach is similar to what has been done in [5]. The plot of the k-refinement, with $M_{ref} = 400, N_{ref} = 10000$, is shown in figure 4.2b for the three RK-integrators. As is apparent, the relative errors match the expected orders for all three methods. More specifically, the relative error decreases as $\mathcal{O}(k^2)$, $\mathcal{O}(k^3)$ and $\mathcal{O}(k^4)$ for RK2, RK3 and RK4, respectively.

The h-refinement is executed in the usual manner, i.e. the relative error is calculated with respect to the analytical solution (4.10). Figure 4.2a shows a plot of the h-refinement for all three RK-integrators, when $N = 1000$. Since the central difference approximation is used along the x-axis when deriving the difference scheme, second order convergence of the relative error is expected. As seen in the figure, the results from the implementation match these expectations.

e)

Runge-Kutta-Nyström (RKN) integrators can be used to solve initial value problems of second-order ODEs on the form

$$\ddot{\boldsymbol{y}} = f(t, \boldsymbol{y}, \dot{\boldsymbol{y}}),\ \boldsymbol{y}(t_0) = \boldsymbol{y}^0,\ \dot{\boldsymbol{y}}(t_0) = \dot{\boldsymbol{y}}^0.$$

The system of second-order ODEs (4.8)

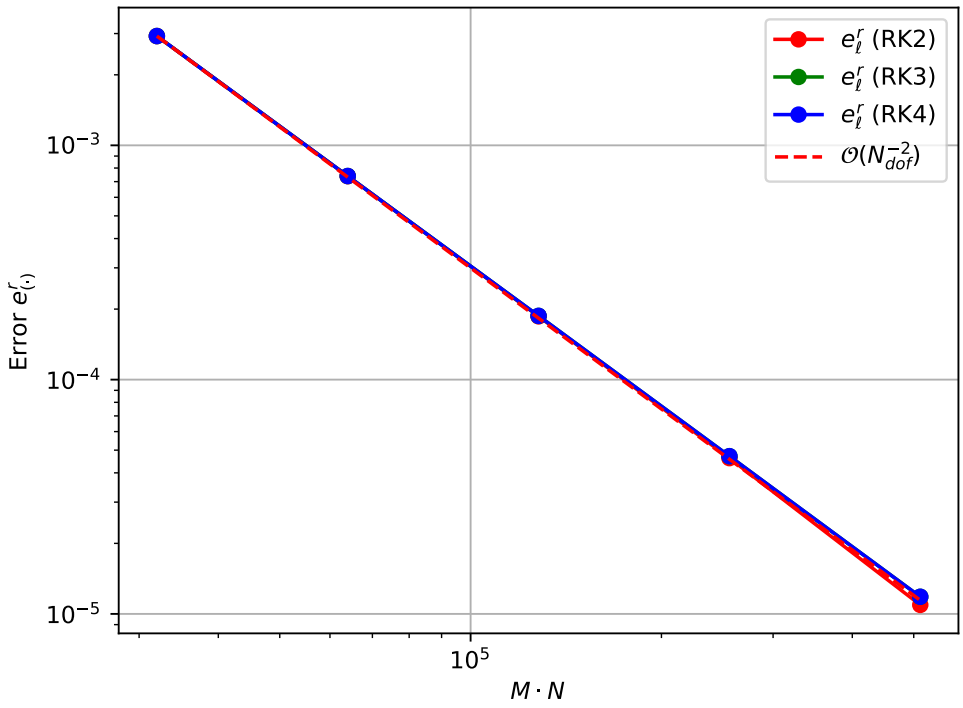
$$\begin{aligned} \ddot{\boldsymbol{v}} &= \frac{1}{h^2} A\boldsymbol{v} + \boldsymbol{g}(t, \boldsymbol{v}) := G(t, \boldsymbol{v}), \\ \boldsymbol{v}|_{t=0} &= \boldsymbol{v}^0,\ \dot{\boldsymbol{v}}|_{t=0} = \dot{\boldsymbol{v}}^0, \end{aligned}$$

where the initial conditions are $\boldsymbol{v}^0 = [u_0(x_1), \ldots, u_0(x_M)]^T$ and $\dot{\boldsymbol{v}}^0 = [u_1(x_1), \ldots, u_1(x_M)]^T$, can be solved with RKN-integrators.

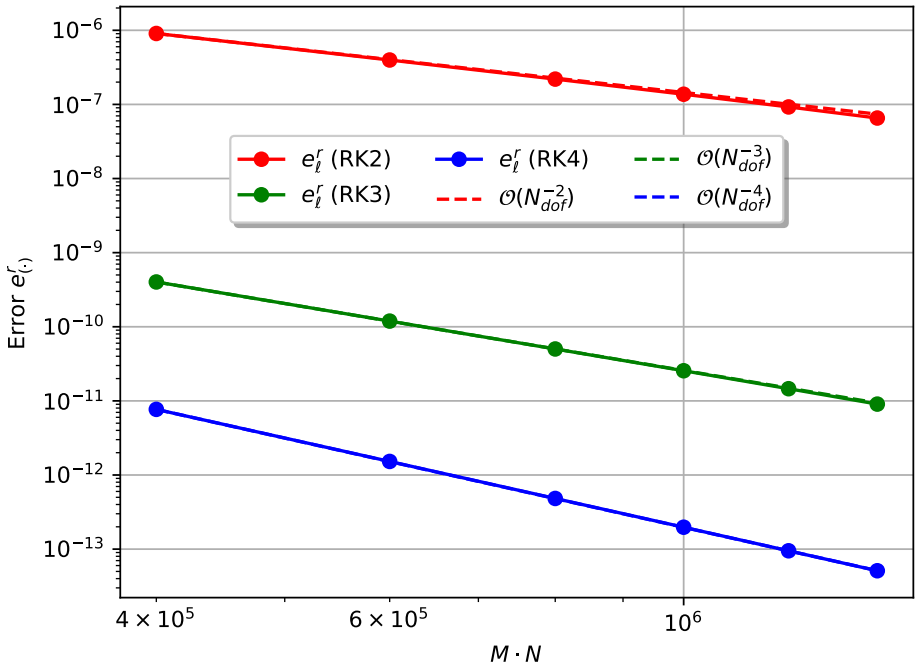
In our implementation, the explicit and symplectic integrators RKN-12 and RKN-34 are used. It is given that RKN-12 is a second order method, while RKN-34 is a fourth order method, which means that their total accumulated error is of order $\mathcal{O}(k^2)$ and $\mathcal{O}(k^4)$, respectively. Let $\boldsymbol{V}^n \approx \boldsymbol{v}(t_n)$ and $\dot{\boldsymbol{V}}^n \approx \dot{\boldsymbol{v}}(t_n)$ be the numerical solutions produced by the integrators at time $t = t_n$. The RKN-12 integrator takes the form

$$\begin{aligned} s_1 &= G(t_n, \boldsymbol{V}^n) \\ \dot{\boldsymbol{V}}^{n+1} &= \dot{\boldsymbol{V}}^n + k s_1 \\ \boldsymbol{V}^{n+1} &= \boldsymbol{V}^n + k\dot{\boldsymbol{V}}^n + k^2 \frac{s_1}{2}, \end{aligned}$$

and the RKN-34 integrator looks like



(a) h-refinement



(b) k-refinement

Figure 4.2: Sine-Gordon equation solved on $x \in [-5, 5]$ and $t \in [0, 5]$ with Dirichlet boundary conditions and initial condition $u(x, 0) = 4 \arctan\left(\exp\left\{-\frac{2x}{\sqrt{3}}\right\}\right)$. The relative error obtained with h-refinement and $N = 1000$ is plotted in (a), while the relative error obtained with k-refinement and $M_{ref} = 400, N_{ref} = 10000$ is plotted in (b). This is done for the different RK-integrators in both cases.

$$
\begin{aligned}
s_1 &= G\left(t_n + (\frac{1}{2} - \delta)k, \boldsymbol{V}^n + (\frac{1}{2} - \delta)k\dot{\boldsymbol{V}}^n\right) \\
s_2 &= G\left(t_n + \frac{k}{2}, \boldsymbol{V}^n + \frac{k}{2}\dot{\boldsymbol{V}}^n + \frac{k^2}{24\delta}s_1\right) \\
s_3 &= G\left(t + (\frac{1}{2} + \delta)k, \boldsymbol{V}^n + (\frac{1}{2} + \delta)k\dot{\boldsymbol{V}}^n + k^2(\frac{1}{12\delta}s_1 + (\delta - \frac{1}{12\delta})s_2)\right) \\
\dot{\boldsymbol{V}}^{n+1} &= \dot{\boldsymbol{V}}^n + k\left(\frac{s_1 + s_3}{24\delta^2} + (1 - \frac{1}{12\delta^2})s_2\right) \\
\boldsymbol{V}^{n+1} &= \boldsymbol{V}^n + k\dot{\boldsymbol{V}}^n + k^2\left((\frac{1}{48\delta^2} + \frac{1}{24\delta})s_1 + (\frac{1}{2} - \frac{1}{24\delta^2})s_2 + (\frac{1}{48\delta^2} - \frac{1}{24\delta})s_3\right),
\end{aligned}
$$

where the generating coefficient is $\delta = \frac{1}{12}(2 - \sqrt[3]{4} - \sqrt[3]{16})$. Note that the generating coefficient δ should not be confused with the central difference operator. These calculations will be executed iteratively for $0 \leq n \leq N - 1$.

The same problem as in **d)** arises when executing k-refinement. By performing k-refinement as explained in **d)**, with $M_{ref} = 400, N_{ref} = 10000$, it is observed that the relative error when using RKN-34 decreases with $\mathcal{O}(k^4)$, while the relative error with RKN-12 is of order $\mathcal{O}(k^2)$, which is precisely as expected. This can be seen in figure 4.3b.

The h-refinement plots with the RKN-integrators are shown in figure 4.3a. The relative error is, as in **d)**, calculated with respect to the analytical solution. As expected, due to the central difference approximation, the decrease in relative error is of order $\mathcal{O}(h^2)$. Note that $N = 15000$ in this case, which is much larger than the value used when performing h-refinement in **d)**, where $N = 1000$ was sufficient. This is the case because it was observed that the relative error with RKN-12 is unstable for large values of M, when N is chosen too small. In other words, we observed that RKN-12 has a strict CFL-condition, i.e. it needs a large N compared to M in order to produce a stable solution.

f)

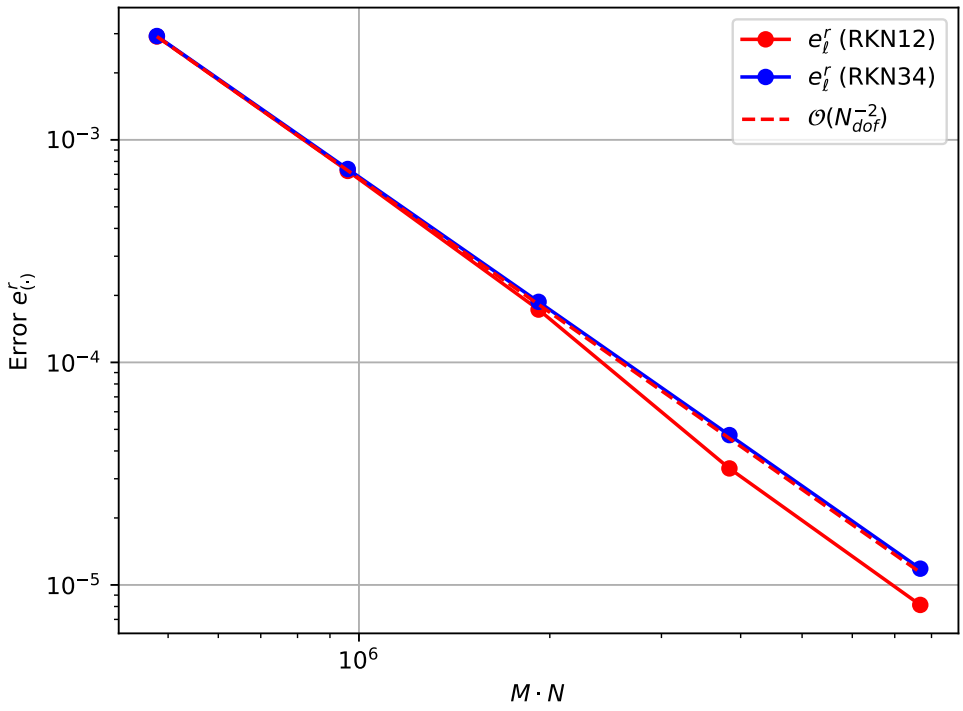
Consider the boundary value problem (BVP)

$$
\begin{aligned}
&u_{tt} - u_{xx} + \sin(u) = 0, && (x,t) \in \Omega \times I, && (4.12)\\
&u(-2,t) = 0,\ u(2,t) = 0, && (x,t) \in \partial\Omega \times I, && (4.13)\\
&u(x,0) = \sin(\pi x)^2 e^{-x^2}, && x \in \Omega \times \{t = 0\}, && (4.14)\\
&u_t(x,0) = \sin(\pi x)^4 e^{-x^2}, && x \in \Omega \times \{t = 0\}, && (4.15)
\end{aligned}
$$

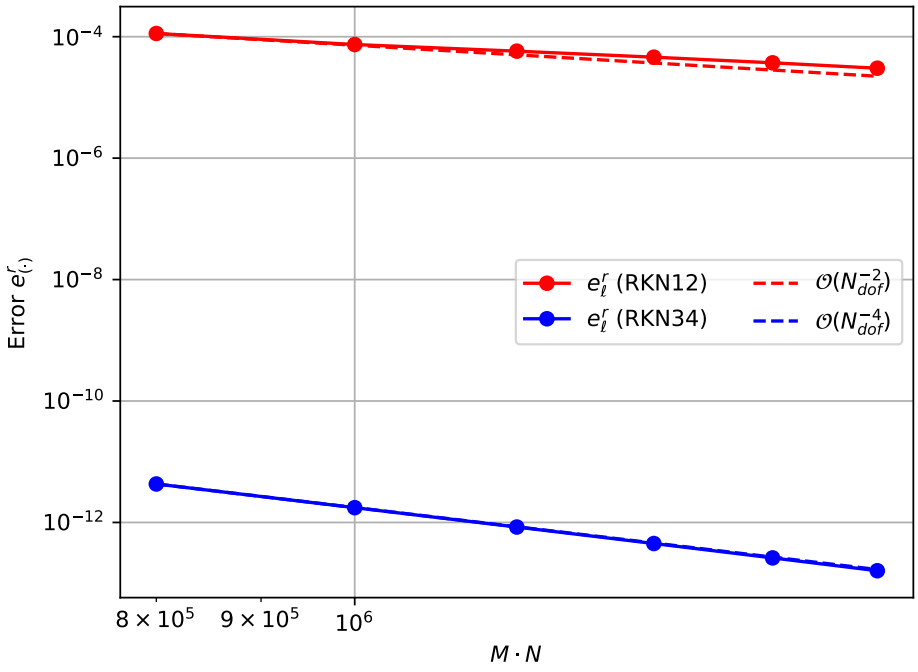
where $\Omega = [-2, 2]$ and $I = [0, 4]$. The RK4 and RKN-34 integrators are applied, in order to solve the BVP numerically. The numerical solution is shown in three dimensions in figure 4.4a. Figure 4.4b shows the numerical solution in the $x - z$-plane, at different times. Here, it is observed that the solution is anti-symmetric at $t = 4$ in relation to its initial condition.

From task **b)**, the energy is on the form

$$
E(t) = \int_{-2}^{2}\left(\frac{1}{2}u_t^2(x,t) + \frac{1}{2}u_x^2(x,t) + 1 - \cos(u(x,t))\right)\mathrm{d}x,
$$

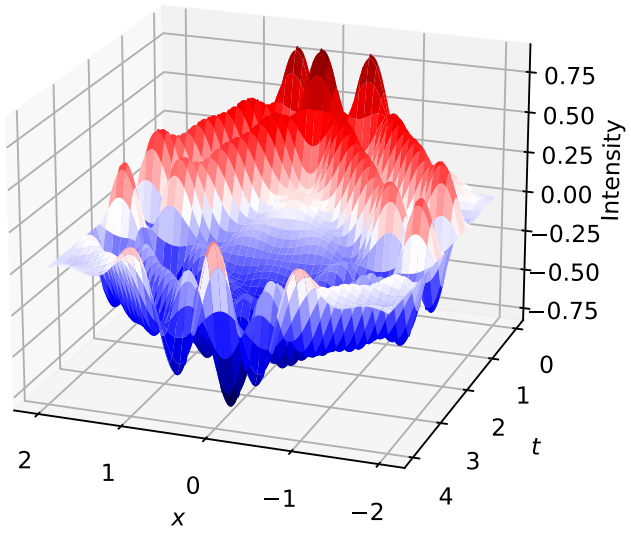


(a) h-refinement

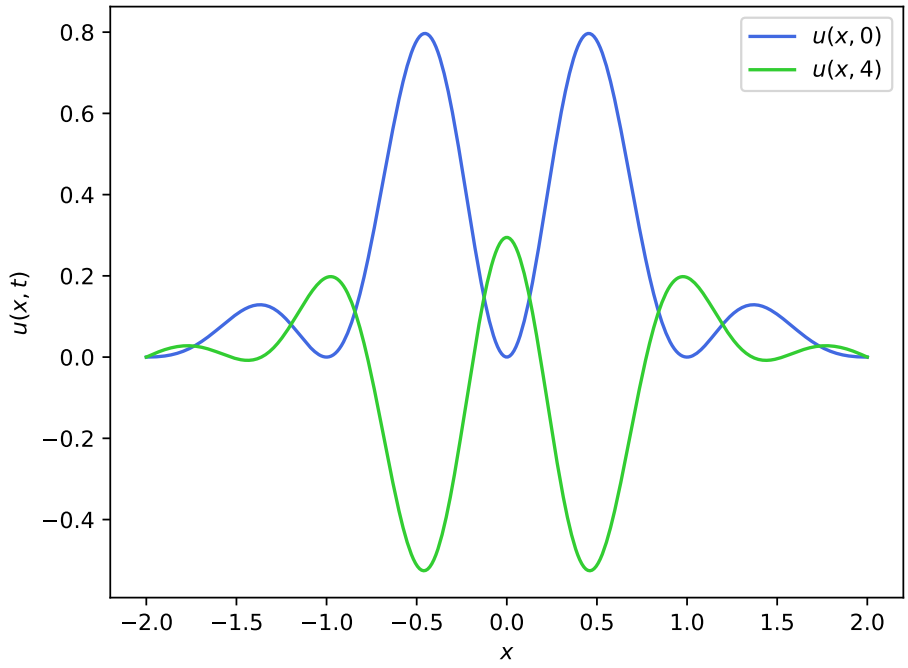


(b) k-refinement

Figure 4.3: Sine-Gordon equation solved on $x \in [-5, 5]$ and $t \in [0, 5]$ with Dirichlet boundary conditions and initial condition $u(x, 0) = 4 \arctan\left(\exp\left\{-\frac{2x}{\sqrt{3}}\right\}\right)$. The relative error obtained with h-refinement and $N = 15000$ is plotted in (a), while the relative error obtained with k-refinement and $M_{ref} = 400, N_{ref} = 10000$ is plotted in (b). This is done for RKN-12 and RKN-34 in both cases.



(a) 3d-plot of numerical solution



(b) Numerical solution at $t = 0$ and $t = 4$.

Figure 4.4: Sine-Gordon equation solved on $x \in [-2, 2]$ and $t \in [0, 4]$ with homogeneous Dirichlet conditions and initial condition $u(x, 0) = \sin(\pi x)^2 \mathrm{e}^{-x^2}, u_t(x, 0) = \sin(\pi x)^4 e^{-x^2}$. The numerical solution is calculated using RK4 with $M = 350$ and $N = 500$. Figure (a) shows a 3d-plot of the numerical solution and (b) shows the solution at time $t = 0$ and $t = 4$.

and fulfills the relation

$$E(t) = E(0) + \int_0^t (u_t(2,t')u_x(2,t') - u_t(-2,t')u_x(-2,t'))\,\mathrm{d}t'.$$

The Dirichlet boundary conditions yield $u_t(-2,t) = u_t(2,t) = 0$, which means that the energy is conserved in the system. From numerical calculations, shown in figure 4.5, it is clear that the energy oscillates around a fixed value until $t = 4$, when the solution is anti-symmetric with regards to the starting position and the energy reaches its initial energy $E(0)$. How precisely the initial energy is reached at $t = 4$ is quantified later.

The integral in the energy-relation at a given time step, is calculated with Gaussian quadrature. More precisely, after the quantities in the integrand are found, the values in the discrete grid points are interpolated using cubic interpolation, before the integral of the resulting function is approximated with Gauss-Legendre quadrature. The quantities u_t and u are found by integrating with RK4 or RKN-34, while the spatial derivative u_x is found by differentiation. By using a central difference approximation, as explained in section 2.2, the truncation error in x is of second order. In order to achieve second order accuracy at the boundaries, the formula (3.3), as introduced in Task 1 (section 3.1), is used. Hence, the approximate derivative is given by

$$(\boldsymbol{V}^n)' = \frac{1}{2h}\begin{pmatrix} -3 & 4 & -1 & 0 & \dots & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 0 & 1 \\ 0 & \dots & 0 & 1 & -4 & 3 \end{pmatrix} \boldsymbol{V}^n,$$

where $\boldsymbol{V} = [V_0, V_1, \dots, V_{M+1}]^T$ is the numerical solution at time step $t = t_n$.

The normalized energy difference $\Delta E = \frac{|E(4)-E(0)|}{E(0)}$, hereby denoted the *energy difference*, is calculated for different mesh grids, with both RK4 and RKN-34.

$(h = ck)$-refinement, or rather $(N = cM)$-refinement, is performed, for different values of the constant c. The c values chosen are $c = \{1.5, 4.0\}$ and the M values are doubled for each refinement in the range $M = 32$ to $M = 4096$. The corresponding plots are shown in figure 4.6. It is observed, in figure 4.6a, that the energy difference is smaller for RKN-34 than for RK4, at least for a small refinement constant $c = 1.5$ and for low values of M. Increasing c corresponds to a finer grid along the t-axis and the two methods will hence give a very similar energy difference, as seen in figure 4.6b.

When executing k-refinement, the same effect can be observed. k-refinement is performed such that $M = 200$ is a constant and N is increased from $N = 400$ to $N = 1500$. The resulting energy difference is shown in figure 4.7a. In this case, a clearer difference between the two methods is observed, compared to when performing $(h = ck)$-refinement, as shown in figure 4.6. The difference is small, but at its largest, the relative difference between the energy difference calculated with the two methods is roughly 2%.

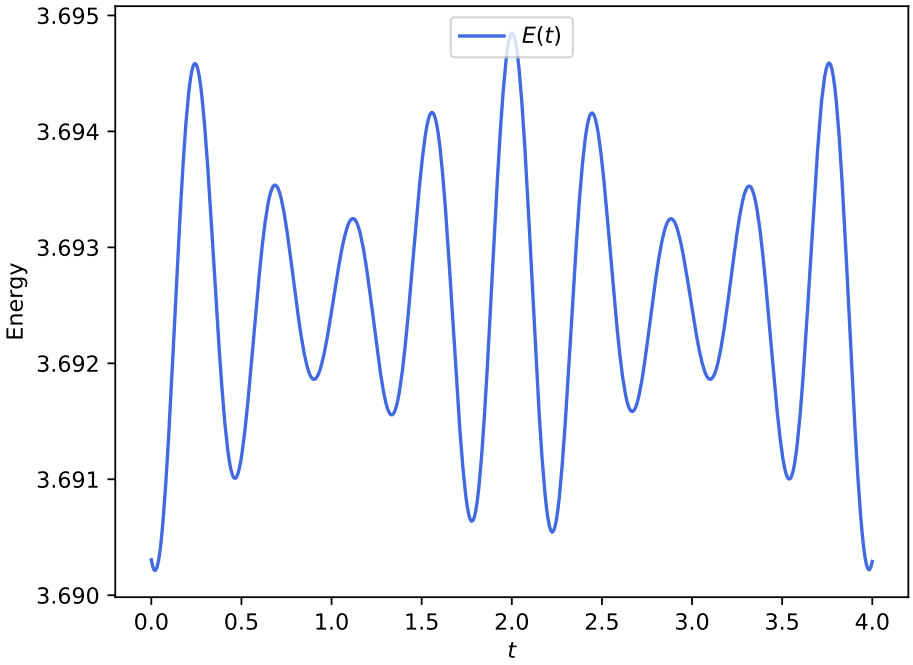
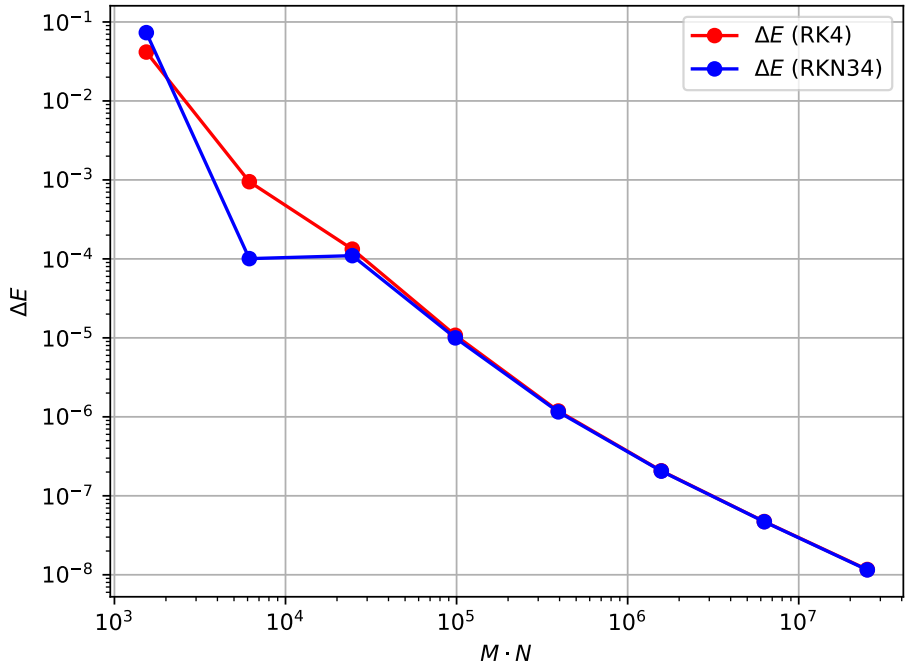


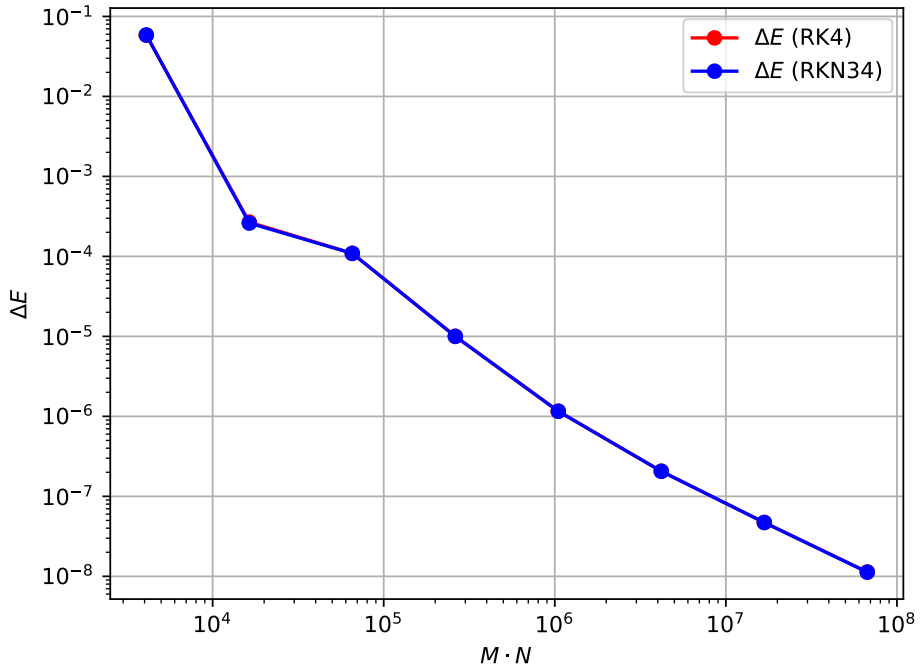
Figure 4.5: The energy of the Sine-Gordon equation solved on $x \in [-2, 2]$ and $t \in [0, 4]$ with homogeneous Dirichlet conditions and initial condition $u(x, 0) = \sin(\pi x)^2 \mathrm{e}^{-x^2}, u_t(x, 0) = \sin(\pi x)^4 e^{-x^2}$, is plotted as a function of time. The numerical solution is calculated using RK4 with $M = 350$ and $N = 500$.

Next, the computational times for RK4 and RKN-34 are compared. To this end, consider the same range of values for M as when performing $(h = ck)$-refinement, but with a large constant value for N. More precisely, set $N = 20000$. The computational times for both methods are depicted in figure 4.7b, as a function of the number of degrees of freedom in the system, i.e $M \cdot N$. Observe that, especially for large values of $M \cdot N$, there is a clear difference between the two methods, where RK4 is the slowest method of the two. The reason why RK4 is slower may be due to the increased number of function evaluations of F.

The results suggest that RKN-34 may be qualified as the superior method. This is due to its precise energy conservation and time efficiency. Its time usage seems to be its most important merit in this case, where a clear distinction between the two methods can be made. It is also appropriate to mention that RKN-34 is a symplectic integrator, which are known to have superior structural properties and long term capabilites on Hamiltonian systems, compared to the non-symplectic conventional methods [6]. We would therefore expect RKN-34 to significantly outperform RK4 for a larger time T.

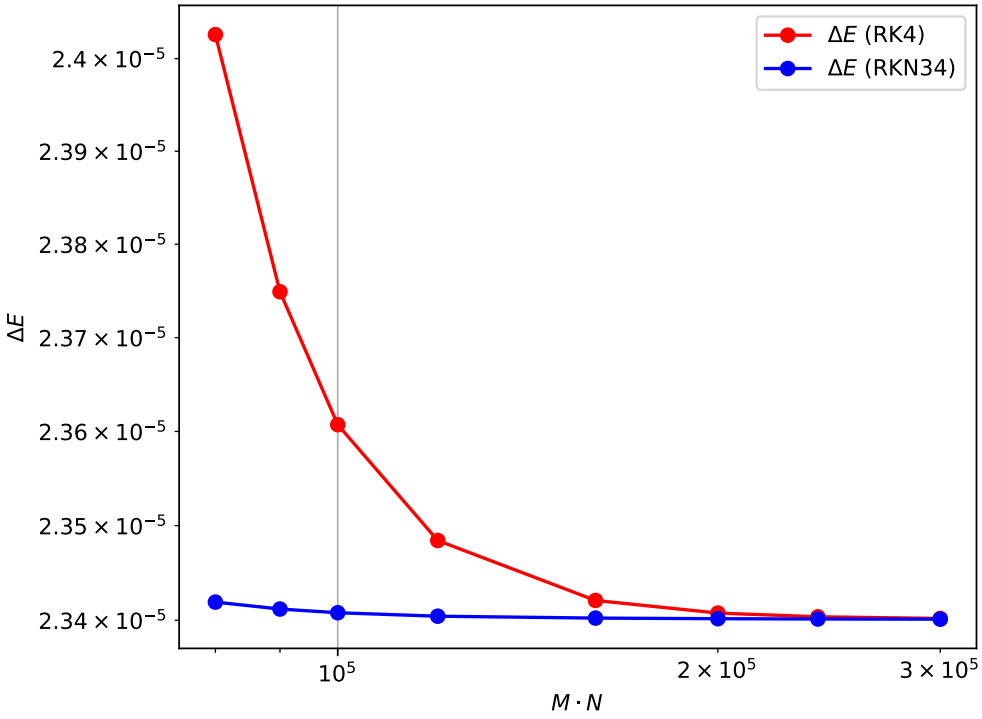


(a) $c = 1.5$

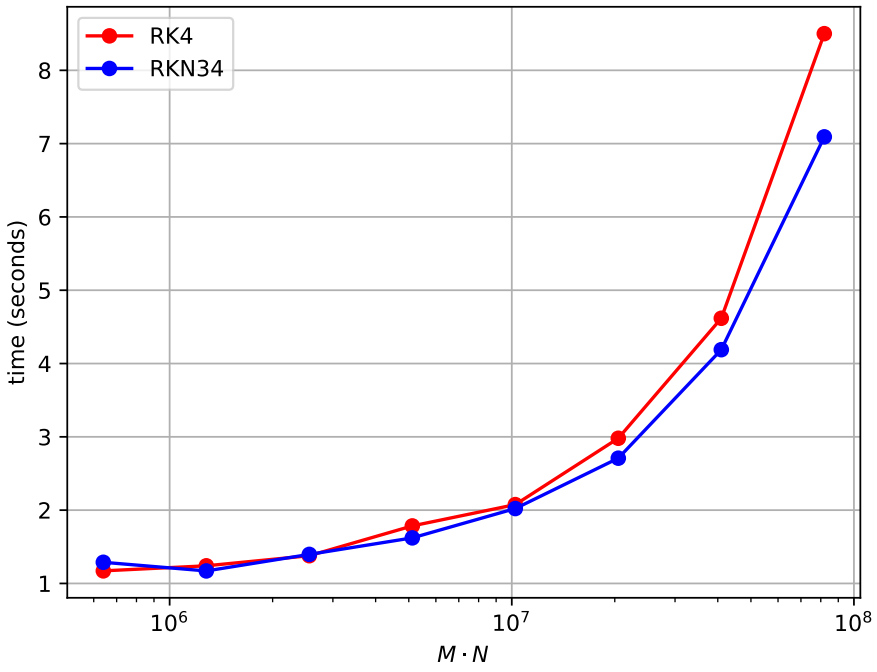


(b) $c = 4$

Figure 4.6: Sine-Gordon equation solved on $x \in [-2, 2]$ and $t \in [0, 4]$ with homogeneous Dirichlet conditions and initial condition $u(x, 0) = \sin(\pi x)^2 \mathrm{e}^{-x^2}$, $u_t(x, 0) = \sin(\pi x)^4 e^{-x^2}$. The energy difference is calculated for $(N = cM)$-refinement with $c = \{1.5, 4.0\}$ for both RK4 and RKN-34.



(a)



(b)

Figure 4.7: Sine-Gordon equation solved on $x \in [-2, 2]$ and $t \in [0, 4]$ with homogeneous Dirichlet conditions and initial condition $u(x, 0) = \sin(\pi x)^2 \mathrm{e}^{-x^2}$, $u_t(x, 0) = \sin(\pi x)^4 e^{-x^2}$. The energy difference, calculated with k-refinement with $M = 200$ and increasing N for both RK4 and RKN-34, is shown in (a). The computational times for both RK4 and RKN-34, calculated with with h-refinement with $N = 20000$, is shown in (b).

Bibliography

[1] C. Curry. *TMA4212 Part 2: Introduction to finite element methods*. 2018.

[2] L. R. Hellevik. *Numerical Methods for Engineers, 6.2 Finite Differences. Notation*. 2020. URL: https://folk.ntnu.no/leifh/teaching/tkt4140/._main055.html. (accessed: 01.03.2021).

[3] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, 2011, pp. 545–549.

[4] J. Liu, G. A. Pope, and K. Sepehrnoori. "A high-resolution finite-difference scheme for nonuniform grids". In: *Elsevier Applied Mathematics and Computation* 19 (3 1995), p. 164. DOI: https://doi.org/10.1016/0307-904X(94)00020-7. URL: https://www.sciencedirect.com/science/article/pii/0307904X94000207.

[5] E. Love and W. Rider. "On the convergence of finite difference methods for PDE under temporal refinement". In: *Computers Mathematics with Applications* 66.1 (2013), pp. 33–40. ISSN: 0898-1221. DOI: https://doi.org/10.1016/j.camwa.2013.04.019. URL: https://www.sciencedirect.com/science/article/pii/S0898122113002265.

[6] X. Lu and R. Schmid. "Symplectic integration of Sine–Gordon type systems". In: *Elsevier Mathematics and Computers in Simulation* 50 (1-4 1999), pp. 255–263. DOI: https://doi.org/10.1016/S0378-4754(99)00083-X. URL: https://www.sciencedirect.com/science/article/pii/S037847549900083X.

[7] B. Owren. *TMA4212 Numerical solution of partial differentialequations with finite difference methods*. 2017.

[8] J. Ramos. "The sine-Gordon equation in the finite line". In: *Elsevier Applied Mathematics and Computation* 124 (1 2001), p. 49. DOI: https://doi.org/10.1016/S0096-3003(00)00080-1. URL: https://www.sciencedirect.com/science/article/pii/S0096300300000801.

[9] Wikipedia®. *Burgers' equation*. 2021. URL: https://en.wikipedia.org/wiki/Burgers%5C%27_equation#Inviscid_Burgers'_equation. (accessed: 22.04.2021).