C1 - Assignment 2 Report

Student Number: u1858921

Submission Deadline: 6pm November 21st, 2018

Contents

1 Introduction

In this report we study numerical solutions to boundary value problems, in particular for the advection-reaction-diffusion equation under given boundary conditions. In general, we define a boundary value problem as follows

> Let $\Omega \subseteq \mathbb{R}^d$, $d = 1, 2$, be a bounded simply connected open domain. Find, for f, v given, a function u such that
>
> $$\mathcal{L}[u] = f, \quad \text{in } \Omega, \qquad u = v, \quad \text{on } \partial\Omega,$$
>
> where $\mathcal{L} = -\Delta u + \mathbf{p} \cdot \nabla u + qu$, with $\mathbf{p}, q$ given smooth, bounded functions.

In this report, we deal with a *Dirichlet problem*, in which we ask to find the solution of the equation in the interior of Ω while prescribing its value on the boundary $\partial\Omega$.

We obtain numerical solutions of the advection-diffusion-reaction equation through the *finite-difference* (FD) method. The approach of the FD method is to evaluate the given operator $\mathcal{L}$ at a set of discretisation points $(x_j)_{j=1}^J$ in Ω with the derivatives within replaced by difference quotients of the approximated solution, U. These difference quotients (e.g. $\partial^-, \partial^+, \partial^+\partial^-$) are obtained by truncating the Taylor expansion of the exact solution about the discretisation points x_j. Then after this discretisation, one should obtain a system of equations, which can be rewritten as a linear matrix problem, i.e. $A\mathbf{U} = \mathbf{f}$ for the approximate solution $\mathbf{U}$ and given data $\mathbf{f}$. There are a number of ways in which linear problems can be solved, one that we shall use is the *Gauss-Seidel* method discussed in Assigment 1 of this course.

In section 2 we go through the discretisation of the advection-diffusion-reaction equation through the finite difference method and present its numerical implementation. In section 3, we test the numerical scheme obtained in section 2 on the advection-diffusion-reaction equation with known analytical solutions as to deduce the error of the scheme and some analyical results. In section 4, we provide a conclusion to the analysis performed in the previous sections.

2 Problem

The aim of this report is to numerically deduce (through the method of finite differences) the solution of the advection-diffusion-reaction equation

$$-\alpha u'' + \beta u' + \gamma u = 0,$$

for given parameter values α, β and γ. In particular, we seek to find the the solution of the linear advection-diffusion equation

$$-\alpha u'' + \beta u' = 0, \quad u(0) = 0,\ u(L) = 1, \tag{1}$$

and the linear diffusion-reaction equation

$$-\alpha u'' + \gamma u = 0, \quad u(0) = 0,\ u(L) = 1, \tag{2}$$

in the domain $\Omega = [0, 1]$. For equation (1), define the so-called *Péclet* number, Pe, through

$$Pe = \frac{|\beta| L}{2\alpha},$$

and for the equation (2), the so-called Damköhler number

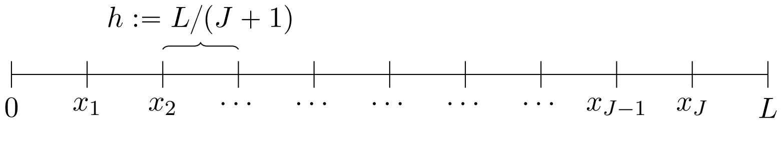
$$Da = \frac{\gamma}{\alpha}.$$

2.1 Discretisation Through Finite-Differences

We begin by discretising the general boundary value problem

$$-\alpha u'' + \beta u' + \gamma u = 0, \quad u(0) = 0,\ u(L) = 1. \tag{3}$$

When we require equation (1) or (2), we choose $\gamma = 0$ or $\beta = 0$ respectively. We discretise equation (3) by finite differences with simple centred schemes for second and first order derivatives. We begin by letting $(x_j)_{j=0}^{J+1}$ be a partition of the interval $\Omega = [0, L]$ such that $0 = x_0 < x_1 < x_2 < \cdots < x_J < x_{J+1} = L$. We denote the jth interval by $I_j = [x_{j-1}, x_j]$ with the mesh size $h_j = |I_j|$. Within this report, we consider a uniform mesh size and therefore set $h := h_j$ for all j, hence $x_j = jh$. Now let $U_j := u(x_j)$ for each $j \in \{1, \ldots, J\}$ and define



$\mathbf{U} = (U_1, \ldots, U_J)$, which denotes the approximation of u on the interior of the partitioning of $\Omega = [0, L]$. Then the discrete problem approximating the continuous problem (3) is given by

$$-\alpha \partial^- \partial^+ U_j + \beta \partial^0 U_j + \gamma U_j = 0, \tag{4}$$

for $j = 1, \ldots, J$ with $U_0 = u(0) = 0$ and $U_{J+1} = u(L) = 1$, where

$$\partial^- \partial^+ U_j = \frac{U_{j-1} - 2U_j + U_{j+1}}{h^2}, \quad \partial^0 = \frac{U_{j+1} - U_{j-1}}{2h}.$$

Equation (4) can be rewritten as

$$\left(-\frac{\alpha}{h^2}-\frac{\beta}{2h}\right)U_{j-1}+\left(\frac{2\alpha}{h^2}+\gamma\right)U_j+\left(-\frac{\alpha}{h^2}+\frac{\beta}{2h}\right)U_{j+1}=0,$$

for $j = 1, \ldots, J$, which gives rise to the following linear equation

$$A\mathbf{U} = \mathbf{f}, \tag{5}$$

where $A = (a_{ij}) \in \mathbb{R}^{J\times J}$ is such that

$$a_{ij} = \begin{cases} -\frac{\alpha}{h^2}-\frac{\beta}{2h} & \text{if } j = i-1, \\ \frac{2\alpha}{h^2}+\gamma & \text{if } j = i, \\ -\frac{\alpha}{h^2}+\frac{\beta}{2h} & \text{if } j = i+1, \\ 0 & \text{otherwise,} \end{cases}$$

and $\mathbf{f} = (f_i) \in \mathbb{R}^J$ is such that

$$f_i = \begin{cases} 0 & i \neq J, \\ \frac{\alpha}{h^2}-\frac{\beta}{2h} & i = J. \end{cases}$$

We make note of the fact that the matrix A is sparse and tridiagonal. This completes the finite difference discretisation, and we may now apply numerical inversion techniques to the matrix A to deduce the numerical solution $\mathbf{U}$ of the problem (3). If A is made diagonally dominant, then we may use the Gauss-Seidel numerical scheme implemented in Assignment 1 to solve the linear system.

2.2 Numerical Implementation

To implement the above discretisation into a numerical scheme, we begin by creating a class called `FiniteDifference` shown below.

```
class FiniteDifference
{
public:
   FiniteDifference(); // Default Constructor
   FiniteDifference(int J, double L, double alpha, double beta, double gamma, double
       b_0, double b_L);
   FiniteDifference(const FiniteDifference& scheme); // Copy Constructor
   ~FiniteDifference(); // Destructor

   int getJ();
   double getL();
   double getAlpha();
   double getBeta();
   double getGamma();
   double getb_0();
   double getb_L();

   SparseMatrix constructMatrix(); // Constructs the "differential operator" matrix A

private:
   int J_; // The number of points in the discretisation
   double L_; // Length of interval
   double b_0_, b_L_; // Boundary conditions
   double alpha_, beta_, gamma_; // Equation parameters
};
```

To carry out the finite difference method described above, we define the member function `constructMatrix` of the class `FiniteDifference` to constuct the operator matrix A defined in equation (5).

```
SparseMatrix FiniteDifference::constructMatrix()
{
   double h = L_/(double) (J_ + 1); // Setting the mesh size
   SparseMatrix A = SparseMatrix(J_,J_);
   double D = 2*alpha_/(double) (h*h) + gamma_; // Diagonal terms of A
   double UD = -(alpha_ - h*beta_/2.0)/(double) (h*h); // Upper-diagonal terms of A
   double LD = -(alpha_ + h*beta_/2.0)/(double) (h*h); // Lower-diagonal terms of A
   for (int i = 0; i < J_; ++i)
   {
      for (int j = 0; j < J_; ++j)
      {
         if(j == i + 1) // Upper-diagonal entries
         {
            A.addEntry(i, j, UD);
         }
         else if(j == i) // Diagonal entries
         {
            A.addEntry(i, j, D);
         }
         else if (j == i - 1) // Lower-diagonal entries
         {
         A.addEntry(i, j, LD);
         }
      }
   }
   return A;
}
```

We then proceed to solve the linear system in equation (5) by inverting the constructed matrix, A, against $\mathbf{f}$ using the Gauss-Seidel numerical scheme (described in the previous assignment) to obtain the approximate solution to the problem (3) in the form of a STL vector $\mathbf{U}$.

3 Results

We can test the finite-difference approximation by analysing the error (using the L^∞ norm) between the numerical and analytical solution, that is by taking the L^∞ norm of the difference of these two solutions. For this, we consider the cases of equations (1) and (2) separately.

3.1 Linear Advection-Diffusion Equation

We can in fact solve equation (1) analytically with the given boundary conditions to obtain a unique solution of the following form

$$u(x) = \frac{1 - \exp\left(\frac{\beta x}{\alpha}\right)}{1 - \exp\left(\frac{\beta L}{\alpha}\right)}, \quad x \in [0, L], \tag{6}$$

for $\alpha \neq 0$ and $\beta \in \mathbb{R}$. We note that when $\beta = 0$, $u(x) = x$, which is the limiting behaviour of the solution (6). To be able to use the Gauss-Seidel algorithm, we require that the discretisation matrix A is diagonally dominant. We can guarantee this as long as

$$\left|\frac{2\alpha}{h^2}\right| > \left|-\frac{\alpha}{h^2} - \frac{\beta}{2h}\right| + \left|-\frac{\alpha}{h^2} + \frac{\beta}{2h}\right| \geqslant \left|\frac{\beta}{h}\right|,$$

where the first inequality follows by diagonal dominance and the second inequality follows by the triangle inequality. That is, for A to be diagonally dominant, we require that $2|\alpha| > h|\beta|$. Since $h := \frac{L}{J+1}$, where J denotes the number of discretisation points, it follows that

$$J + 1 > \frac{|\beta|L}{2|\alpha|} \geqslant Pe,$$

since $Pe = \frac{|\beta|L}{2\alpha}$. Thus if the number of disretisation points J is greater that the Péclet number minus one ($J > Pe - 1$), then we are guaranteed to have convergence of the approximate solution to the correct solution through the use of Gauss-Seidel. We have therefore obtained a sufficient condition for the numerical scheme to converge. Taking this into account, along with knowing the analytical solution (6), we can deduce the error, $\|u_h - u\|_\infty$, of the numerical solution, u_h, with respect to the analytical solution, u, for various values of α, β and number of discretisation points J. The error plots for this can be seen in Figure 1, which have been computed for Péclet numbers up to 10.5. We can compute the error for even larger Péclet numbers if desired by simply increasing the number of discretisation points (to guarantee convergence). From Figure 1, we see that the error between the analytical solution and the numerical solution decreases exponentially for Peclet numbers of order at least 1, but when Pe is very close to zero, we observe that the error is very small and decreases extremely slowly as mesh size h decreases. To explain the small constant error for when the Peclet number is very small, we use stability of the finite-difference method, which states that

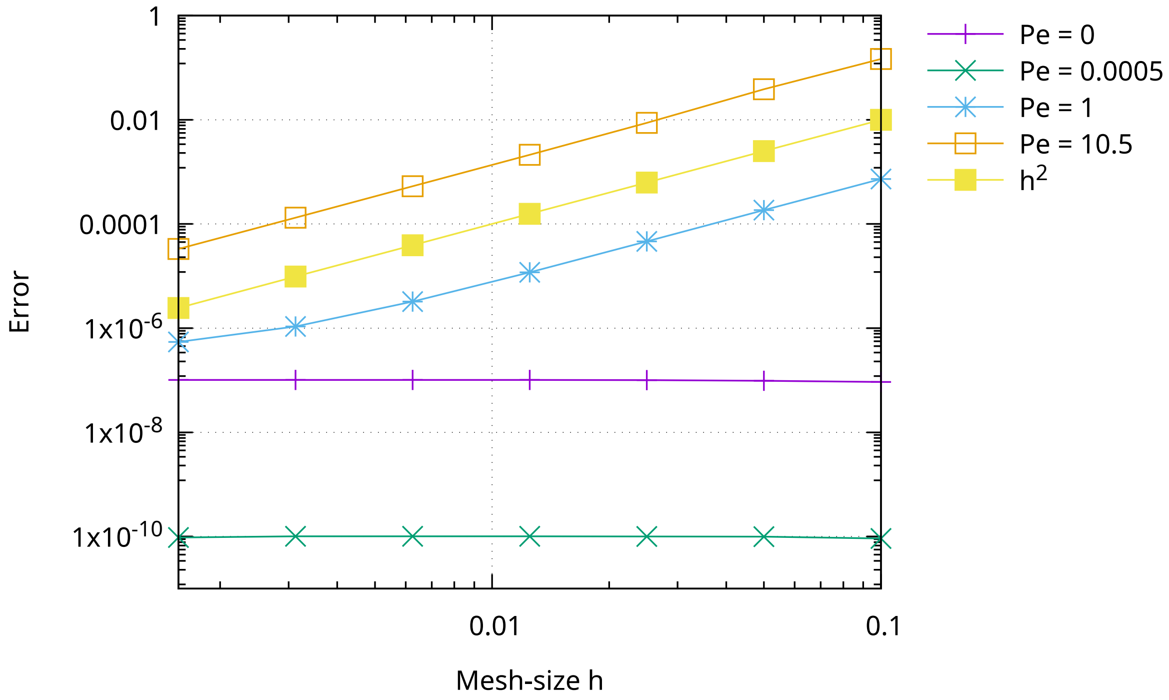


Figure 1: Log-log plot of the error between the analyical and numerical solution of equation (1) for variaous Peclét numbers Pe and mesh sizes h.

$$\|r_h u - u_h\|_\infty \leqslant C\|\mathcal{L}_h(r_h u) - \mathcal{L}_h u_h\|_\infty,$$

for some constant C, where $r_h : C(\overline{\Omega}) \to \overline{\mathbb{U}}_h$ is the restriction operator that allows us to compare continuous functions with grid functions[1] and u_h is the approximate solution U obtained on the

[1] Where $\overline{\mathbb{U}}_h := \{ U : \overline{\Omega}_h \to \mathbb{R} \}$ and $\overline{\Omega}_h$ is the set of all grid points $(x_j)_{j=0}^{J+1}$.

mesh size h. By performing a Taylor expansion about the point $x+h$ and $x-h$ for some $x \in \Omega_h = (x_j)_{j=1}^J$, we deduce that if $\beta \neq 0$

$$\|r_h u'' - u_h''\|_\infty \leqslant \frac{h^2}{12}\|u^{(4)}\|_\infty, \quad \|r_h u' - u_h'\|_\infty \leqslant \frac{h^2}{6}\|u^{(3)}\|_\infty,$$

where $u^{(k)}$ denotes the kth derivative of u with respect to x. Therefore, we can obtain the following bound on the error

$$\begin{aligned}\|r_h u - u_h\|_\infty &\leqslant C\| - \alpha r_h u'' + \beta r_h u' + \alpha u_h'' - \beta u_h'\|_\infty \\ &\leqslant C\left(|\alpha|\|r_h u'' - u_h''\|_\infty + |\beta|\|r_h u' - u_h'\|_\infty\right) \\ &\leqslant C\left(\frac{|\alpha|h^2}{12}\|u^{(4)}\|_\infty + \frac{|\beta|h^2}{6}\|u^{(3)}\|_\infty\right).\end{aligned}$$

Since we know the analytical solution, for $k = 3, 4$

$$u^{(k)}(x) = -\frac{\left(\frac{\beta}{\alpha}\right)^k}{1 - \exp\left(\frac{\beta L}{\alpha}\right)}\exp\left(\frac{\beta x}{\alpha}\right),$$

therefore, for $k = 3, 4$

$$\|u^{(k)}\|_\infty = \left|\frac{\beta}{\alpha}\right|^k \left|\frac{\exp\left(\frac{\beta L}{\alpha}\right)}{1 - \exp\left(\frac{\beta L}{\alpha}\right)}\right| = \left|\frac{\beta}{\alpha}\right|^k g(\alpha, \beta),$$

where $g(\alpha, \beta) = \left|\frac{\exp\left(\frac{\beta L}{\alpha}\right)}{1 - \exp\left(\frac{\beta L}{\alpha}\right)}\right|$. Then

$$\|r_h u - u_h\|_\infty \leqslant C\frac{h^2}{4}\frac{|\beta|^4}{|\alpha|^3}g(\alpha, \beta).$$

Therefore, the order of convergence is at least 2, which was expected by the Fundamental Theorem of Numerical Analysis for FD methods, since, from above 2 is clearly the order of consistency. We note that since $Pe = \frac{|\beta|L}{2\alpha}$, and if we assume $\alpha > 0$ and $\beta \neq 0$, then

$$\|r_h u - u_h\|_\infty \leqslant C\frac{2h^2\alpha}{L^4}(Pe)^4 g(\alpha, \beta). \tag{7}$$

Hence, the fourth order behaviour of the Peclet number in equation (7) may explain why the error in Figure 1 is extremely small for when Pe is small, since this order is significantly greater than that of h.

3.1.1 Determining the Order of Convergence

Assuming the same notation as in the previous subsection, we say that a numerical scheme has order of convergence p if there exists a constant C independent of h such that

$$\|r_h u - u_h\|_\infty \leqslant Ch^p,$$

where u denotes the exact solution of the required BVP and u_h denotes the approximate solution obtained through said numerical scheme under mesh size h. We can assume that we may write

$$\|r_h u - u_h\|_\infty = Ch^p + \mathcal{O}(h^{p+1}),$$

then by taking the log of each side, we obtain

$$\log(\|r_h u - u_h\|_\infty) = \log C + p \log h + \mathcal{O}(h),$$

therefore, we may simply determine the order of convergence, p, by plotting the curve h^q alongside the log-log plot of error against mesh size h and see if the error curves line up with h^q, in which case it is safe to assume that $p \approx q$. Therefore, by analysing the plots of Figure 1, we can state that the order of convergence, when the Péclet number is large enough, is approximately 2 by comparing it to the the curve h^2 plotted alongside the data, although is slightly smaller than 2 for $Pe = 1$ when the mesh size is small. To instead dertmine p more precisely, we can proceed as follows:

$$\frac{\|u - u_h\|_\infty}{\|u - u_{h/2}\|_\infty} = \frac{Ch^p + \mathcal{O}(h^{p+1})}{C(h/2)^p + \mathcal{O}((h/2)^{p+1})} = 2^p + \mathcal{O}(h).$$

Then taking the $\log_2$ of both sides yields

$$\log_2\left(\frac{\|u - u_h\|_\infty}{\|u - u_{h/2}\|_\infty}\right) = p + \mathcal{O}(h),$$

which for h small yields a good approximation of the order of convergence, p, of the numerical scheme. By performing this analysis on the errors obtained for different mesh sizes (see Figure 1), the precise order of convergence is outlined in Table 1 for various different Péclet numbers. In this table, the *guess* made by looking at the data in Figure 1 agrees with the precise order of convergence calculated.

Péclet number	Order of convergence
0	-0.002694509
0.0005	0.001483022
1	1.872685377
10.5	2.007963453

Table 1: Precise values of the order of convergence of the Gauss-Seidel numerical scheme when performed on the linear system (5) for various values of Péclet numbers. The order of convergence was calculated by applying the method described in section 3.1.1 using mesh sizes $h = \frac{1}{80}$ and $\frac{h}{2} = \frac{1}{160}$.

In Figures 2, 3, 4, 5, for small mesh sizes, it is difficult differentiate between the plots made in each figure, but one can, if one zooms close enough, see the difference between the numerical solutions generated by different mesh sizes.

3.2 Linear Diffusion-Reaction Equation

We can also solve equation (2) analytically with the given boundary conditions to obtain a unique solution of the following form

$$u(x) = \frac{\sinh\left(\sqrt{\frac{\gamma}{\alpha}}x\right)}{\sinh\left(\sqrt{\frac{\gamma}{\alpha}}L\right)}, \quad x \in [0, L],$$

and then apply a similar analysis to that done to equation (1) in Section 3.1. We omit the analysis in this report since the method is almost identical.

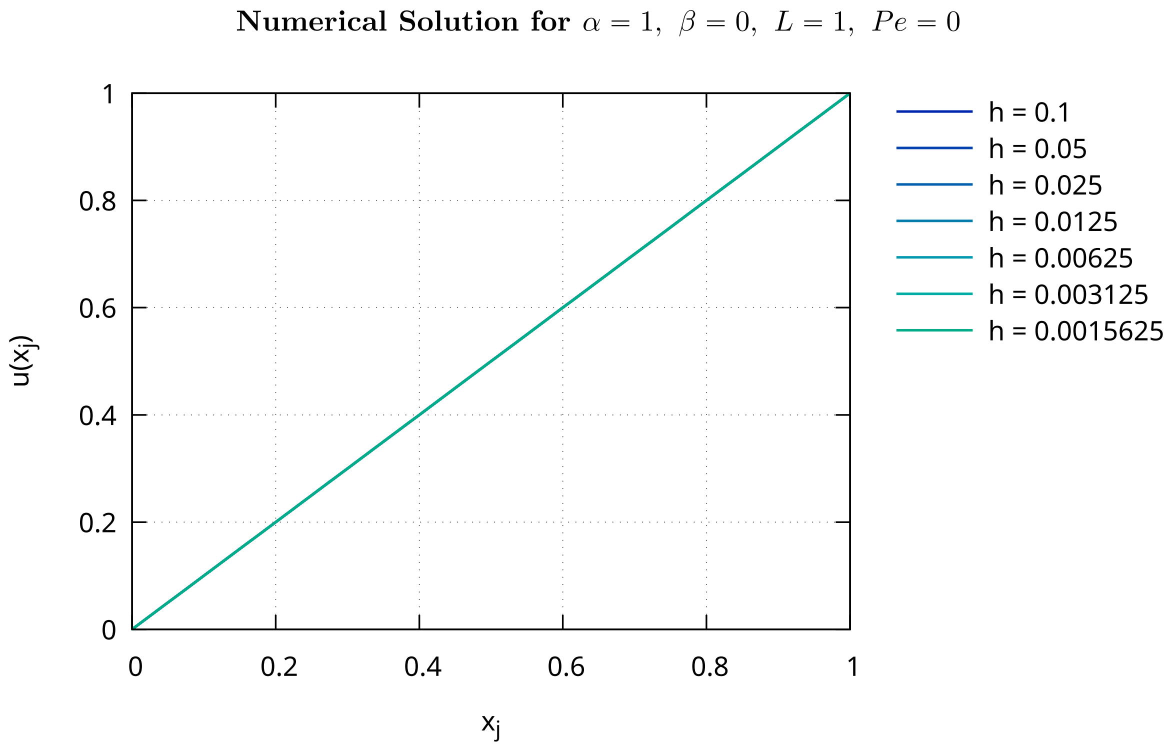


Figure 2: Solution plots of the BVP (1) for $\alpha = 1,\ \beta = 0,\ L = 1$ and various mesh sizes h.

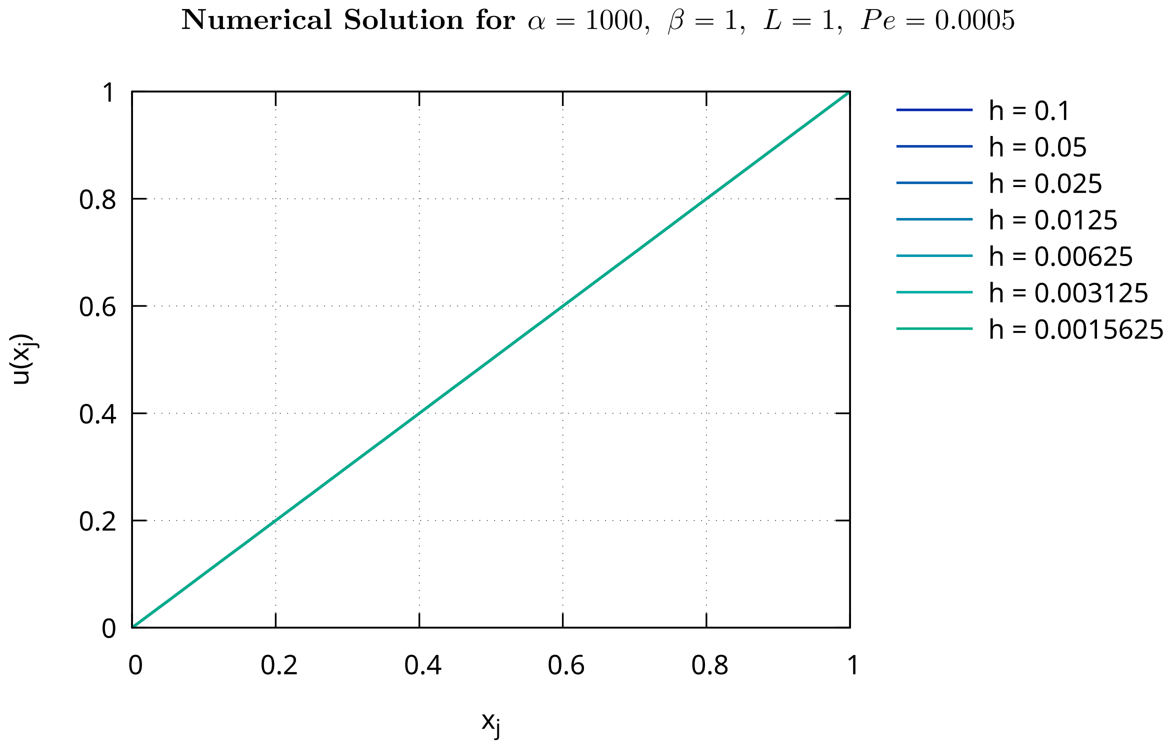


Figure 3: Solution plots of the BVP (1) for $\alpha = 1000,\ \beta = 1,\ L = 1$ and various mesh sizes h.

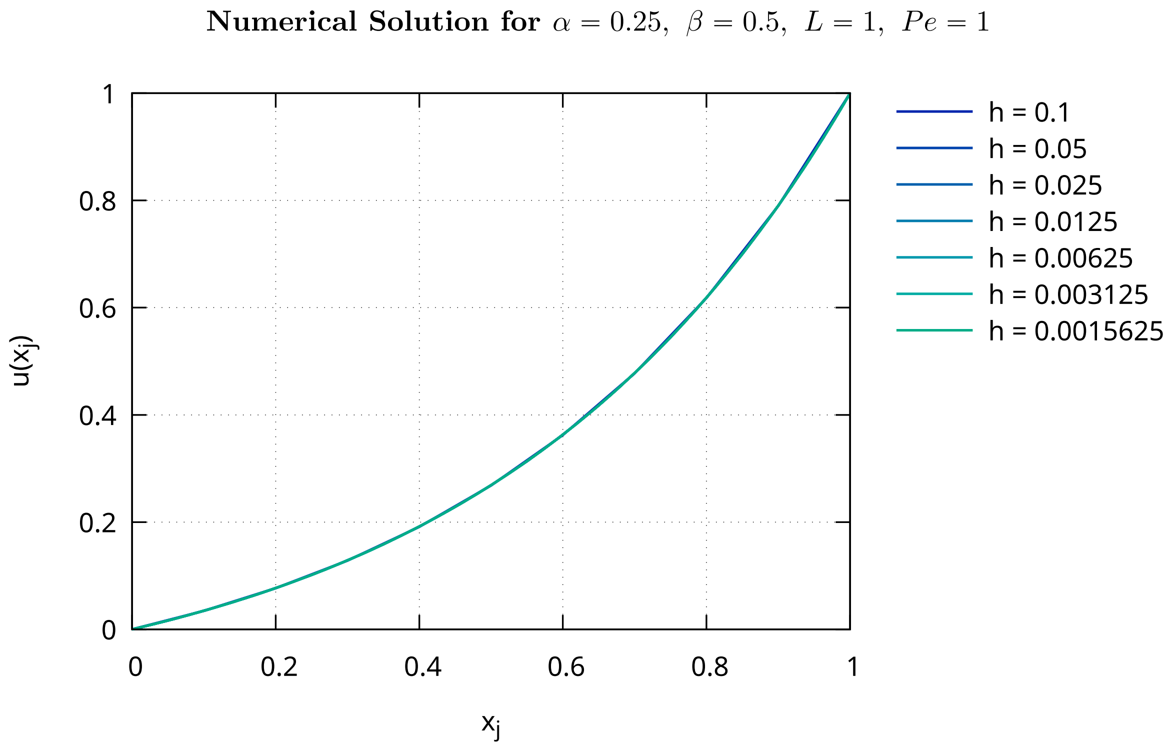


Figure 4: Solution plots of the BVP (1) for $\alpha = 0.25,\ \beta = 0.5,\ L = 1$ and various mesh sizes h.

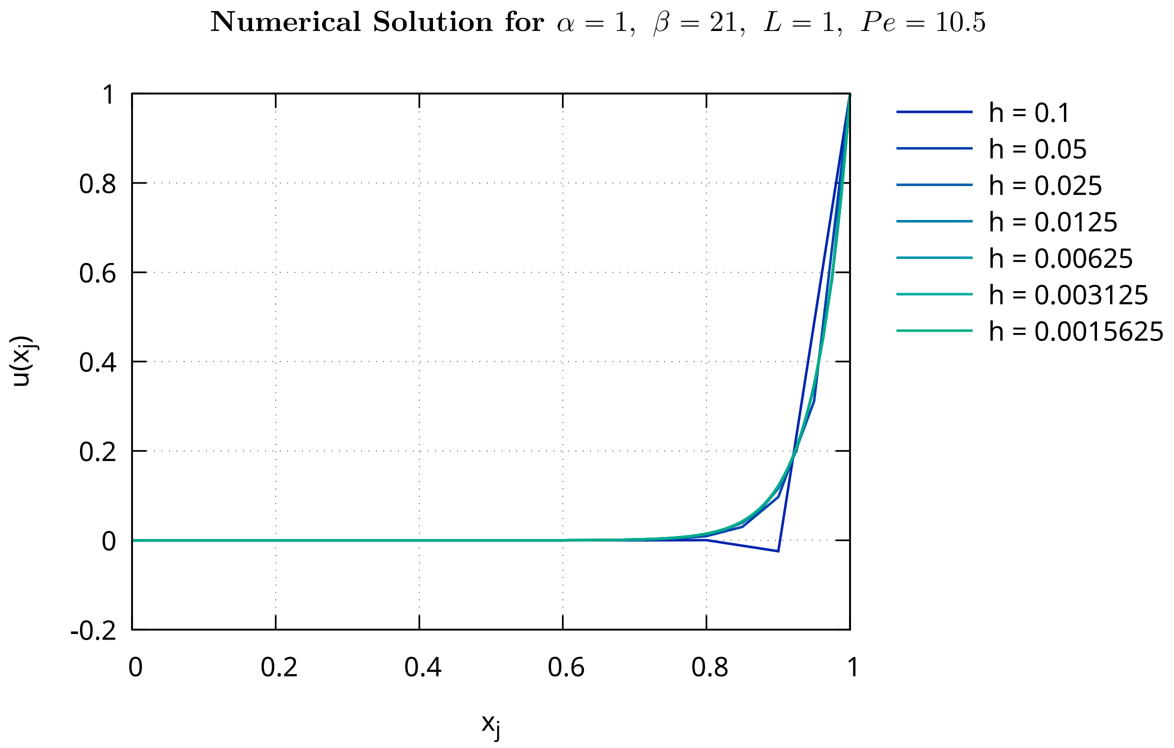


Figure 5: Solution plots of the BVP (1) for $\alpha = 1,\ \beta = 21,\ L = 1$ and various mesh sizes h.

4 Conclusion

In this report, we have described the finite-difference discretisation of the general advection-diffusion-reaction equation (3) for general parameter values α, β and γ. In section 2, we further described the numerical implementation of this discretisation which we used to obtain approximate numerical solutions to the equation (1) for various mesh sizes h in section 3. The numerical scheme, Gauss-Seidel, used to solve the linear system of equations (5) is accurate, as can be seen from the error plots in Figure 1, and we have determined both visually and numerically that the order of convergence of this scheme is approximately 2 (see Figure 1 and Table 1). This agrees (as we expect) with the Fundamental Theorem of Numerical Analysis for FD schemes since the order of consistency is 2 . Furthermore, we can see that the numerical solutions are accurate (to a certain degree of accuracy) by comparing the numerical solution plots of the BVP (1), shown in Figures 2, 3, 4, 5, with the corresponding analytical solution (6).