Porous Medium Equation

C.Lehre, E.A.Myhre

March 31, 2019

Abstract

The porous medium equation is one of the simplest examples of evolutionary non-linear parabolic partial differential equations. The equation arise in various natural phenomena, and is closely related to the heat equation, both in terms of theory and properties. Areas of applicability are e.g gas flow through porous medium, heat transfer and population dynamics.

In this report we solve the one-dimensional porous medium equation with homogeneous Dirichlet boundary conditions numerically by implementing two finite difference schemes. We consider an explicit FTCS and an implicit BTCS scheme, and look into order of convergence for both methods. Although there are many solutions to the porous medium equation, we consider a special solution in which the initial value is a point source, namely the Barenblatt solution. The explicit scheme initialized with the Barenblatt solution performs well when the discrete grid in time is sufficiently fine compared to the grid in space. As for the implicit scheme, we observe unexpected results when initializing with the Barenblatt solution when the grid in time is coarser than the grid in space. However, the scheme perfoms as expected when initializing with a smooth function, independent on the ratio of step-sizes in time and space.

As expected theoretically, both methods converge with order 1 in time and 2 in space.

1. Introduction

In this report we consider the one-dimensional porous medium equation, a non-linear parabolic partial differential equation. We solve the problem numerically with appropriate initial values and boundary conditions by implementing two different finite difference schemes, and compare the numerical results with theoretical expectations in terms of order of convergence.

The problem itself is stated in section 2, and we present some theory regarding the problem and the difference schemes in section 3. In section 4 we present some of the most common applications of the equation, before we present and discuss numerical results in section 5. Finally, we summarize our findings and discuss further studies in section 6.

2. Problem description

In this report we are concerned with the porous medium equation. From now on we will refer to the equation with the abbreviation PME.

We consider homogeneous Dirichlet boundary conditions in one spatial dimension, and we end up with the following model problem.

$$\boxed{\begin{aligned} &u_t = \nabla_x^2(u^s) \quad s > 1 \\ &x \in [-4, 4], \quad T > 0 \\ &u(x,0) = f(x) \\ &u(-4,t) = u(4,t) = 0 \end{aligned}} \tag{1}$$

The equation is an example of a non-linear parabolic partial differential equation, and occurs in numerous physical phenomena which we will discuss in a later section.
Observe that the PME corresponds to the heat equation $u_t = \nabla_x^2(u)$ for $s = 1$. In fact, one can observe similar properties for the PME with $s > 1$ and the heat equation.

3. Theory

We will focus on a special solution to the PME, namely the Barenblatt solution. The solution occurs in the situation in which the initial value, $f(x)$, is a point-source. The analytical solution, as derived in [1] is

$$\boxed{B(x,t) = \max\left(0, \frac{1}{t^\alpha}\big(C - \frac{s-1}{2s}\beta\frac{|x|^2}{t^{2\beta}}\big)\right)} \tag{2}$$

$$\alpha = \frac{d}{d(s-1)+2}, \quad \beta = \frac{\alpha}{d},$$

where d is the number of spatial dimensions (here $d = 1$). Observe that the Barenblatt solution in (1) has compact support for $t > 0$. The solution is a so-called self-similar solution, i.e, when appropriately scaling the variables, the solution becomes stationary[2]. Observe from (2) that the solution is not differentiable everywhere, in particular where $C - \frac{s-1}{2s}\beta\frac{|x|^2}{t^{2\beta}}$ changes sign. This might cause some problems that will be further discussed in a later section.

3.1. Difference schemes

We will now propose two finite difference schemes to solve the problem (1) numerically.

First, lets rewrite the PME in (1).

$$\begin{aligned} u_t &= \nabla_x^2(u^s) = (su^{s-1}\cdot u_x)_x \\ &= su^{s-1}\cdot u_{xx} + s(s-1)u^{s-2}\cdot u_x^2, \end{aligned} \tag{3}$$

where $(\cdot)_x = \frac{\partial(\cdot)}{\partial x}$ and $(\cdot)_{xx} = \frac{\partial^2(\cdot)}{\partial x^2}$.

To obtain finite difference schemes to (1), we have to discretize the temporal and spatial domains as

$$\begin{aligned} x_m &= mh,\ m = 1,\dots,M+1, \quad h = \frac{x_{M+1}-x_1}{M} \\ t_n &= nk,\ n = 0,\dots,N, \quad k = \frac{T}{N}, \end{aligned}$$

where h and k are the stepsizes in space and time, respectively.

In the first scheme we take use of a first order, forward difference in time and second order, central differences in space, i.e

$$\begin{aligned} u_t(x,t) &= \frac{u(x,t+k)-u(x,t)}{k} + \mathcal{O}(k), \\ u_x(x,t) &= \frac{u(x+h,t)-u(x-h,t)}{2h} + \mathcal{O}(h^2), \\ u_{xx}(x,t) &= \frac{u(x+h,t)-2u(x,t)+u(x-h,t)}{h^2} + \mathcal{O}(h^2). \end{aligned} \tag{4}$$

Let $U_m^n \approx u(x_m,t_n)$ be the numerical approximation to $u(x,t)$ at the gridpoint $(x = x_m, t = t_n)$. If we use the this approximation and insert the difference formulas in (4) into (3), we get

$$\begin{aligned} \frac{U_m^{n+1}-U_m^n}{k} &= \frac{s(U_m^n)^{s-1}}{h^2}(\delta_x^2 U_m^n) \\ &\quad + \frac{s(s-1)}{4h^2}(\delta_x U_m^n)^2. \end{aligned}$$

The discretization results in the following scheme:

$$\boxed{\begin{aligned} U_m^{n+1} &= U_m^n + sr(U_m^n)^{s-1}(U_{m+1}^n - 2U_m^n + U_{m-1}^n) \\ &\quad + \frac{sr(s-1)}{4}(U_m^n)^{s-2}(U_{m+1}^n - U_{m-1}^n)^2, \end{aligned}} \tag{5}$$

where $r = k/h^2$. This is an explicit method, meaning that the method uses information at the current time-step to calculate the approximation at the next time-step.

Another approach to a finite difference scheme is to interchange the the first order, forward difference formula in time, with a first order, backward difference formula, i.e

$$u_t(x,t) = \frac{u(x,t)-u(x,t-k)}{k} + \mathcal{O}(k). \tag{6}$$

Combining this with the central differences in (4) and inserting into (3) we get the following scheme:

$$\boxed{\begin{aligned} &U_m^{n+1} = U_m^n + sr(U_m^{n+1})^{s-1}(U_{m+1}^{n+1} - 2U_m^{n+1} + U_{m-1}^{n+1}) \\ &+ \frac{sr(s-1)}{4}(U_m^{n+1})^{s-2}(U_{m+1}^{n+1} - U_{m-1}^{n+1})^2. \end{aligned}} \tag{7}$$

Notice that this scheme is implicit, meaning that the method uses unknown information in the current time-step to compute the approximation. In order to implement this, we have to solve a non-linear system of equations in each time step, causing the method to be computationally more expensive than the explicit scheme.

Based on the difference formulas used to derive the schemes, we can expect first-order convergence in time and second-order convergence in space for both methods.

As the problem (1) is non-linear, von Neumann's stability criterion can no longer be applied. However, one can approximate the stability of the schemes by linearizing the non-linear terms. We chose to not go forward with this analysis, but we will however analyze the stability by numerical experiments in a later section.

A method that depends on a step-size h is convergent of order p if

$$\|e_h\| = Ch^p,$$

for some constant $C > 0$. If we now take the logarithm on each side of the above equation, we get

$$\log(\|e_h\|) = \log(C) + p\log(h).$$

Thus, if we plot the norm of the global error against the step-size in a loglog fashion, the slope of the resulting curve gives the order of convergence.

4. Applications

The porous media equation is closely related to the heat equation, and appears naturally in several physical applications. The deep mathematical knowledge which has been developed on this equation, is strongly due to the interesting prospects and the diversity of usage this equation offers in many different fields. We will discuss some of the most common applications, to get a better understanding of the nature of the PME.

4.1. Gas flow through porous mediums

One major use of the equation, is when studying gas flow through a porous medium. By using and combining relevant laws in physics considering such a problem, one can obtain a general relationship given by

$$\rho_t = c\nabla(\rho^s) \tag{8}$$

where ρ denotes the density, and c depends on physical quantities such as pressure, viscosity, porosity and permeability. This equation might be extended to a more generalized situation, where the mentioned physical quantities might depend on space and time. The equation (8) extends to

$$\epsilon(x,t)\rho_t = \nabla\left(c(x,t)\nabla\rho^s\right)$$

for functions ϵ and c.

4.2. Heat transfer

Another important application is related to heat propagation with temperature-dependent thermal conductivity. In general, this process can be described by the following equation

$$c\rho\frac{\partial T}{\partial t} = \mathrm{div}(\kappa\nabla T) \tag{9}$$

considering the temperature (T), specific heat (c), density of the medium (ρ) and thermal conductivity (κ). This equation can be written as a generalized PMEs in the cases where some of the variables c, ρ, κ are not constant, but rather dependent on the temperature. It has been shown that several heat transfer problems can be modelled with (9). For instance, one can describe heat propagation by mainly electromagnetic radiation in plasmas at very high temperatures by assuming $\kappa = \kappa(T)$. This becomes a PME with parameter $s = 4$. Another example is electron heat conduction in a plasma, which may be described by the porous media equation with $s = 3.5$.

4.3. Population dynamics

The PME can also be used to model the spread of biological populations. By discounting the presence of several species, we can model the population spread in the following way

$$\partial_t u = div(\kappa\nabla u) + f(u)$$

where u is the concentration of species, κ is the diffusivity and $f(u)$ is a reaction term considering symbiotic interaction between species. Assuming κ is an increasing function of the density, u, and disregarding the reaction term, $f(u)$, we obtain the PME with exponent $s = 2$.

5. Numerical results

In order to find numerical solutions to the model problem given by (1), we used the Barenblatt solution at t_0 as initialization for $f(x)$ with $s = 2$. The choice $s = 2$ corresponds to multiple applications, e.g groundwater infiltration and population dynamics, and we found this case particularily interesting to study.
The Barenblatt solution for $s = 2$ is shown in figure 1. Since the Barenblatt solution diverges for very low t-values, we decided to consider the problem on the domain $x \in [-4, 4]$ and $T \in [0.1, 2]$. Hence $u(x, t_0) = f(x) = B(x, t_0)$, with $t_0 = 0.1$ and $B(x, t_0)$ given by the exact solution in (2).

The explicit scheme given by (5) is the simplest method to implement, as all the terms on the right hand side of the equation is known. This yields a linear system of equations which is easy to solve. However, this method offers some stability concerns if r gets too big. The behaviour of the method was roughly analyzed numerically, and we found that stability issues often occurred for $r > 0.4$. Thus we made sure to choose a sufficiently small stepsize in time, to control the value of r well below 0.4. The numerical solution with this method is presented in figure 2, and it looks to be very similar to the exact solution. The evolution of the solution with time can be viewed in the animation in figure 3.

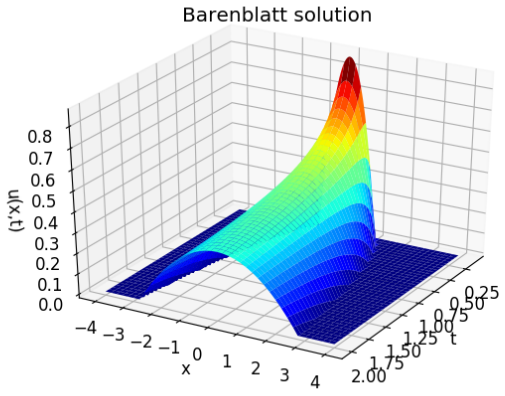

Figure 1: Barenblatt solution with $M = 200$ and $N = 200$.

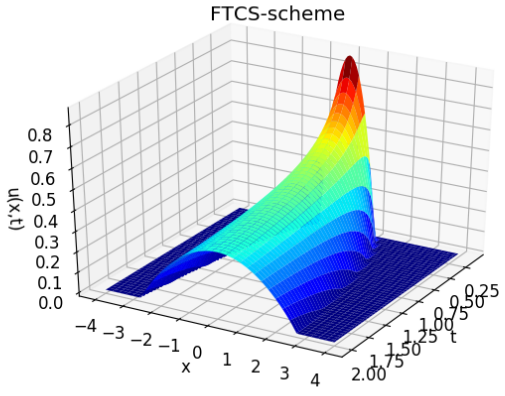

Figure 2: Numerical solution using the explicit FTCS scheme with $M = 100$ and $N = 800$

Figure 3: Animation FTCS

The implicit scheme given by (7) is a bit more complicated to implement. In order to calculate the approximation at the next step in time, we have to solve a non-linear system of equations. This was done using a numerical solver, namely

`scipy.optimize.fsolve`

The implicit numerical solution is shown in figure 4, and the evolution with time can be viewed in the animation in figure 5.
The benefit with this method is that you do not have any stability considerations to make, as with the explicit method, as the method is unconditionally stable. Thus, one can allow for a coarser grid in time while still maintaining stability when using the implicit scheme.
However, the computations are much more expensive due to the non-linear system of equations that needs to be solved at each time-step, which makes the algorithm drastically slower.
In figure 9 and 10 we have plotted the absolute error for both methods, compared to the exact solution. We observe that the implicit error is more accurate, giving a smaller error. So by doing more expensive calculations, one obtains a stable and more accurate method.

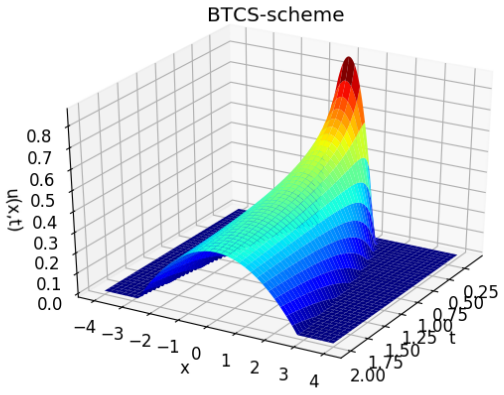

Figure 4: Numerical solution using the implicit BTCS scheme with $M = 200$ and $N = 200$

Figure 5: Animation BTCS

Looking at figures 1 and 4, we see that the numerical approximation using the implicit scheme is very similar to the Barenblatt solution for step-sizes corresponding to $M = 200$ and $N = 200$.
However, when increasing the number of gridpoints in both space and time, we get unexpected results.
In figure 6 we have plotted the numerical solution of (1) witg $M = 300$ and $N = 300$. Notice how the solution decays as expected for a short amount of time, but becomes stationary for the remaining time of integration.

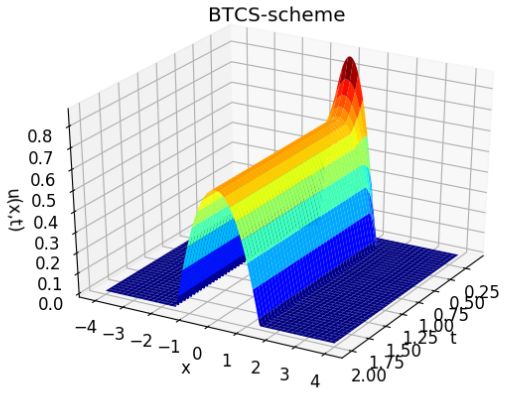

Figure 6: Numerical solution initialized with the Barenblatt solution using the implicit BTCS scheme with $M = 300$ and $N = 300$

In figure 7 we increase the number of gridpoints even further, using $M = 800$ and $N = 800$. Notice how the solution decays with time, but stays constant rather than spreading out in space as we expect. It is also apparent that the solution decays in a greater extent than the Barenblatt solution in figure 1.

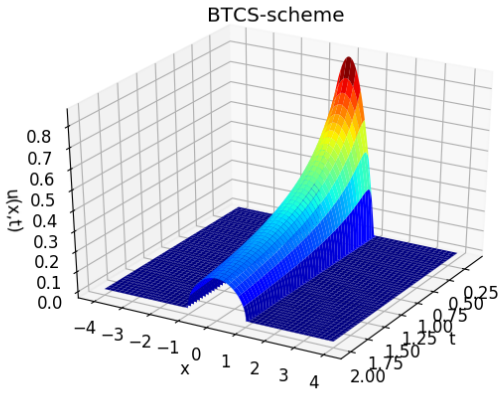

Figure 7: Numerical solution initialized with the Barenblatt solution using the implicit BTCS scheme with $M = 800$ and $N = 800$

We also observe the same unexpected behaviour when increasing the integration time while keeping the amount of gridlines in both time and space as when the scheme performs well, or by reducing the domain in space. Again, these are factors that contribute to the step-length in time being greater than the step-length in space.
In both cases above, we solve the unexpected behavior of the scheme by increasing the step-length in space relative to the step-length in time.
Thus, it is apparent that the step-length in time needs to be sufficiently lower than the step length in space for the scheme to perform as wanted. That is, a sufficiently finer grid in time than in space.
However, when initializing the numerical approximation with a smooth function the scheme works as expected. This is illustrated in figure 8, where the scheme was applied on an initialization given by $f(x) = e^{-x^2}$, with the exact same grid as in figure 7.

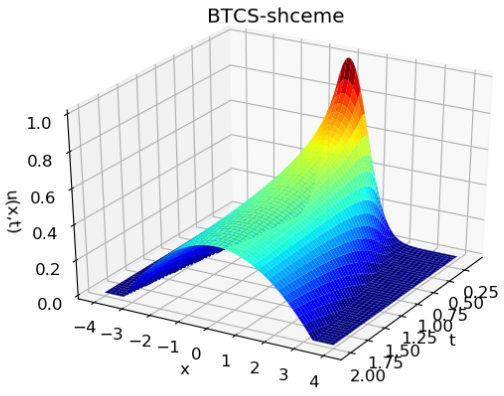

Figure 8: Numerical solution with smooth initialization using the implicit BTCS scheme with $M = 800$ and $N = 800$

This weird behavior is most likely due to the method that solves the non-linear system of equations arising from the implicit scheme. The method approximates the Jacobian of the function that is evaluated, and will not necessarily converge properly for a function that is not differentiable. The fsolve function is not an ideal choice for solving the non-linear system of equations when initializing the numerical approximation with the Barenblatt solution, as it is not differentiable.

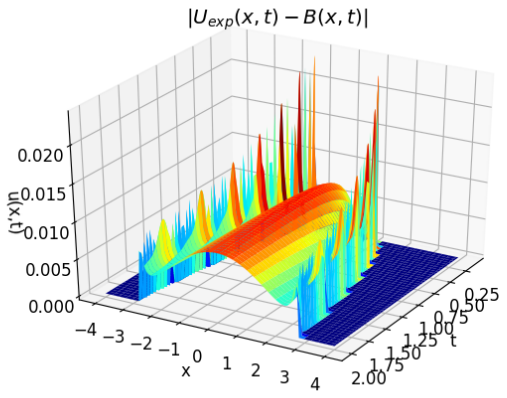

Figure 9: Absolute error between the numerical solution using the explicit scheme and the Barenblatt solution with $M = 100$ and $N = 800$

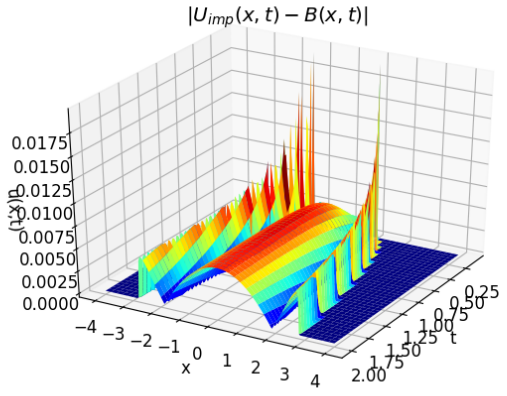

Figure 10: Absolute error between the numerical solution using the implicit scheme and the Barenblatt solution with $M = 200$ and $N = 200$

As mentioned above, we expect the order of convergence to be one in time and two in space for both methods. To check whether the schemes work as expected, we construct convergence plots in space and time, where the error in some norm is plotted versus the step-size in a logarithmic fashion. We would then expect a linear relationship in both cases, with slope 1 in time and slope 2 in space.

In figure 9 and 10 we have plotted the absolute error between the Barenblatt solution and numerical approximations using the explicit and implicit methods, respectively. These plots are illustrating the motivation for producing the convergence plots the way we did. As mentioned above, the exact solution is not differentiable everywhere, hence it can not be Taylor-expanded which is problematic when considering the error. Observe that the error where the solution is non-differentiable is large and irregular.

Due to this, we can not use the exact solution as reference when considering the error in space. Instead, we choose another initialization, which is smooth over all x. We decided to use a function somewhat similar to the Barenblatt initialization, that is

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

We then calculated a reference solution numerically with this initialization, using a very fine grid in space, and considered the the error between this reference solution and other numerical approximations with varying step-sizes. The error at $T = 2$ was measured in the 1st norm, and plotted versus the step-size in a log-log fashion, as shown in figure 11. We obtained the theoretically expected slope for both methods, indicating that the schemes works well.

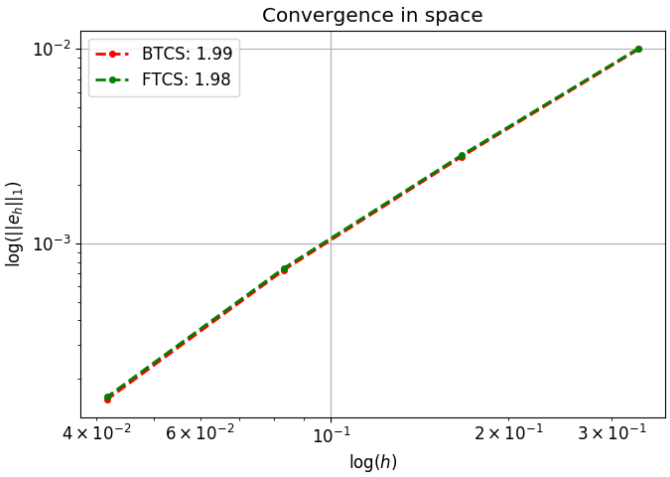

Figure 11: Convergence in space. The green curve corresponds to the explicit scheme, while the red curve corresponds to the implicit scheme

For making a convergence plot in time, the non-differentiability is not an issue as we consider the error at a suitable x-value. In this case we used the exact solution with fixed parameters as a reference, and compared it to numerical approximations initialized by the Barenblatt solution at t_0 with varying step-sizes in time. The error was measured in the 1st norm at some appropriate value for x where the solution is smooth. The resulting convergence plot is

presented in figure 12. We observe that the slopes are approximately equal to 1 for both methods. This is a good indication that our implementation works as expected.

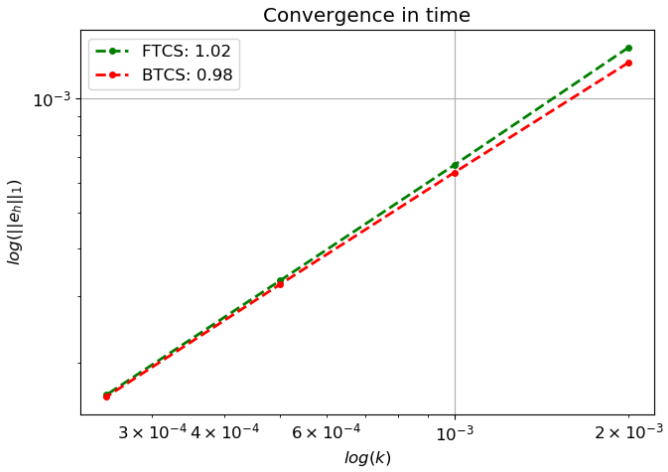

Figure 12: Convergence in time. The green curve corresponds to the explicit scheme, while the red curve corresponds to the implicit scheme

6. Summary

The explicit scheme works well and is stable under certain restrictions on r, while the implicit scheme is unconditionally stable. We observe some unexpected behaviour for the implicit scheme when initializing with the exact Barenblatt solution, and this weird behaviour is most likely due to the method that solves the non-linear system of equation at each time-step. However, when the step-length in time is sufficiently small compared to the step-length in space, the implicit scheme works well with this initialization. The implicit scheme is more computationally expensive than the explicit, but the resulting global error is less for the implicit scheme. As expected from the theory, both methods converge with order 2 in space and 1 in time

For further studies it would be interesting to look more into the effect of the ratio of the step-length in time and space to the performance on the implicit scheme we have implemented, as discussed in section 5. It would also be interesting to implement a method for solving the non-linear system of equations arising in the implicit scheme without using the Jacobian, and see if the resulting scheme performs well unconditionally. To get a more accurate understanding of the effect of r on the stability of the explicit scheme, one could also do an analytical stability analysis by linearizing the non-linear terms in the equation.

Both group members contributed with a equal amounts of work for this project.

References

[1] L. C. Evans, *Partial Differential Equations*. American Mathematical Society, 1999.

[2] J. L. Vázquez, *The Porous Medium Equation, Mathematical Theory*. Oxford Science Publications, 2007.

[3] B. Owren, *TMA4212 Numerical solution of partial differential equations with finite difference methods*. 2017.