

Solving a non-linear partial differential equation for the simulation of tumour oxygenation

*Julian Köllermeier (julian.koellermeier@rwth-aachen.de), Lisa Kusch (lisa.kusch@rwth-aachen.de),
Thorsten Lajewski (thorsten.lajewski@rwth-aachen.de)*

Center for Computational Engineering Science - mathematics division,
RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany

Advisor: *Prof. Dr. Martin Frank (frank@mathcces.rwth-aachen.de), Center for computational engineering science - mathematics division*

1. INTRODUCTION

The research on tumors is a significant scientific challenge of our time. Most insights into the micro environment of a tumor are gained via invasive methods, which can change the outcomes of the experiments. Other non-invasive methods like PET imaging cannot provide resolutions needed for detailed quantification of the tumour microenvironment (see DASU, 2004 [1], p.9f). As a consequence it is crucial to come up with a suitable mathematical model for the theoretical analysis of tumor tissue.

One important aspect which can already be modeled using a partial differential equation is the oxygenation of a tumor which influences the success of radiation therapy (see also BRIZEL ET AL, 1997 [5]). Therefore, a valid mathematical model can be helpful to predict the outcome of a therapy and may help to improve therapy methods.

First simulations using a finite element method software generated promising results (compare DASU, 2004 [1]). But due to limitations in computational power, the calculable area is bounded when non-specialized tools are used. Hence, adapting simple mathematical methods to lessen the overhead of memory during the calculation is essential.

The aim of this project is the development and analysis of a program which can calculate larger domains in two space dimensions, while needing only a minimum amount of storage space and reasonable computation time.

Our approach towards this task is to use a finite difference method for the given partial differential equation. We describe different ways to generate a suitable domain for a tissue and develop and analyze a discretization of this domain with a structured grid. One significant aspect in terms of memory is the data format we developed as an output for the discretization. To solve the emerging system of equations we employ suitable iterative methods.

Following this approach we were able to obtain satisfying results. These show that our program manages to solve the equation on large domains with a user chosen resolution. The accuracy of the solution can be estimated using an a-priori error estimator.

2. APPLICATION

Considering tumor radiative treatment one has to take into account the special structure and characteristics of the tumor. The following sections show briefly how the physiology of the tissue influences radiotherapy (compare also DASU 2004 [1]). These points all lead to the development of theoretical models for tumor oxygenation as described in the next section and they were also the reason for us to build up a new simulation software adapted to the problem.

2.1. Tumour micro-environment and oxygenation. The reason tumor treatment is difficult is the special micro-environment of its tissue. Altogether this results in a vascular network of the tumor that is highly irregular, which is different from normal tissue. Another important point is the

fact that tumors do not have any lymphatic vessels. Without these lymphatic vessels the pH may decrease which leads to a lack of oxygen in some regions of the tissue away from the vessels.

Additionally tumor cells consume much more glucose than normal cells. An important link to radiotherapy is that the DNA repair mechanisms do not work properly in that case of a glucose deprivation, which is interesting for acute and chronic hypoxia (DENEKAMP AND DASU 1999 [4]).

2.2. Acute and chronic hypoxia. Hypoxia means a lack of oxygen in a cell. This may result from a large distance of the cell to the next blood vessel or a low oxygen level inside the blood vessel itself.

Chronic hypoxia describes a permanent oxygen defect that does not depend on time. In contrast to that, acute hypoxia is the result of an instantaneous oxygen deprivation that does not remain for such a long time. The typical time scale for acute hypoxia in tumour tissue varies between minutes and hours (compare [10]). Hence, the oxygen diffusion itself can still be modelled as a stationary process.

The reason for perfusion limited hypoxia, a type of acute hypoxia, is the temporary occlusion or collapse of certain vessels inside the tumor. This happens because of the rigidity of red blood cells induced by the lack of glucose and a low pH.

2.3. Influences on radiotherapy. In general, one can say that there is an increase in radiosensitivity of the tumor cells if the level of oxygenation rises. It is supposed that the different forms of hypoxia also have different impacts on the success of radiative treatment. According to DENEKAMP AND DASU 1999 [3] chronically hypoxic cells that lack both oxygen and glucose are not able to activate their DNA repair mechanisms, so that they are more sensitive to radiation than normal cells. Depending on the conditions a chronically hypoxic cell may also be more sensitive than an oxygenated and glucose-fed cell. In contrast to that, acute hypoxia leads only to a temporary lack of oxygen, so that the repair mechanisms might still be working due to the short time of lacking oxygen. Therefore, acute hypoxic cells are radioresistant to a certain extent (compare DASU 2000 [4]). This is one of the reasons why tumor oxygenation is considered one of the most important factors that determine the failure of radiation treatment (see e.g. BRIZEL ET AL 1997[5]). Thus, it is necessary to distinguish between these effects by predicting for example the amount of oxygen in different regions of the tissue.

In order to get information about the oxygenation of certain parts of a tumor one could try to measure oxygen values experimentally. The problem is that those measurement methods are mostly invasive and can therefore spoil the quality of the measurement itself or the resolution is very poor and not at all sufficient. Additionally, the methods are most often not able to give any information about whether they measure acute or chronic hypoxia. For more information see DASU 2004 [1].

At this point theoretical modeling of tumor oxygenation seems the only possibility to overcome those problems because it can quickly give very good results depending on the model. Due to this we will describe a model of a virtual tumor that enables the efficient simulation of tumor oxygenation.

3. THEORETICAL MODELING

3.1. Mathematical modeling of tumor oxygenation. The problem of tumor oxygenation can be described as a diffusion equation for the partial pressure p of oxygen in the tissue. The partial pressure serves as a representative value for the oxygen content.

The stationary diffusion equation for p reads

$$(1) \quad -D\Delta p(x) = f(p, x), \quad x \in G \subset \mathbb{R}^n.$$

G is the spatial domain representing a tumor tissue with a specific number of blood vessels inside. The term on the left describes the diffusion caused by the concentration difference in the domain. The standard form ($f = 0$) of the diffusion equation is a linear partial differential equation. Therefore, a linear numeric solver would be sufficient to solve that equation.

For describing the process of tumor oxygenation the diffusion equation is supplemented with the loss term $f = -q(p)$, which describes the oxygen consumption of the tissue.

$$(2) \quad q(p) = q_{max} \frac{p}{p+k}$$

In this equation q_{max} is the maximum consumption rate and k is the partial pressure at which the consumption rate decreases to one half of the maximum value.

Because of the fact that this term is non-linear, the whole PDE has to be solved with a non-linear solver.

The diffusion coefficient D in (3) implies the assumption of being independent of x . This is only valid if the tissue is homogeneous, which is a valid simplification in this project. This finally leads to equation (3).

$$(3) \quad -D\Delta p + q_{max} \frac{p}{p+k} = 0$$

This equation can be classified as an elliptic equation. Hence, the problem is a boundary value problem. As the partial pressure inside the blood vessels is typically constant, the vessels serve as boundaries additionally to the border of the domain. Here lies the challenge because of the difficult approximation of the circular vessel boundaries.

For the project we have considered two different boundary models. Both have in common that the variable p at the boundary of the blood vessels inside the tissue is set to a user-specified value which serves as a boundary condition. Therefore the area of interest contains a huge number of boundaries. The simulation with a common finite element method software using an unstructured grid therefore results in a grid with very high resolution and many grid points near the vessels. This leads to a higher memory consumption. An alternative to the finite element approach is to use a Cartesian grid with properly discretized vessels (see Section 3.3).

In the first boundary model we have considered the boundary of the tissue to be a Dirichlet boundary. That means the pressure at the outer boundary is set to a certain value.

Concerning the second boundary model we have considered the domain to be periodic. So one does not need to impose values for p on the boundary of the domain.

Parameters. The given equation (3) is composed of the different parameters D , q_{max} , k and the variable p . In order to include these in the model one has to take a closer look at their influence in the equation and their units. In the given model the used measurement unit for length is μm .

All blood vessels have a constant specific value for their partial pressure. The unit of the pressure is $1 \text{ mmHg} \approx 133 \text{ Pa}$ corresponding to the pressure occurring in a 1 mm mercury column. A representative value for the pressure in blood vessels is 40 mmHg (THOMLINSON AND GRAY, 1955 [3]).

The diffusive term of the equation contains the diffusion coefficient D . As the unit $\frac{\mu\text{m}^2}{\text{s}}$ depicts, it describes how much oxygen diffuses across an area in a certain time along the pressure gradient. It can also be described as the mobility of oxygen particles. The diffusion coefficient in the tumor tissue is of the order 10^3 . A higher diffusion coefficient means a higher diffusion of oxygen from the blood vessels into the tissue. For example it can be set to $2 \cdot 10^3 \frac{\mu\text{m}^2}{\text{s}}$ (compare TANNOCK, 1972 [6]) depending on the temperature of the body.

The consumption term (2) stands for the consumption of oxygen in a part of the tissue with a certain pressure. The expression $\frac{p}{p+k}$ is a value between 0 and 1. Therefore, the maximum consumption q_{max} in front of the expression makes sense. A representative value in a tumor tissue is a maximum consumption of $15 \frac{\text{mmHg}}{\text{s}}$ (THOMLINSON AND GRAY, 1955[3], TANNOCK, 1972 [6]) and k can be set to 2.5 mmHg (DASU, 2004 [1]).

3.2. Tissue generation. The task of tissue generation is to specify the domain on which the PDE (3) will be solved. In order to generate a meaningful tumor tissue one has to choose the size of the domain and then decide on a certain distribution of blood vessels inside the domain afterwards. Here the size can be specified by the length and width of the domain as we only consider 2D domains.

As in DASU, 2004 [1] for the distribution of the vessel in the area we use a normal distribution of the minimal distance between two vessels. A normal distribution can be described by two parameters mean μ and variance σ^2 .

The calculation of the minimal distances between two vessels $d_{i_{min}}$ is done in two different ways. In the case of Dirichlet boundaries it is the distance to another vessel or to the boundary. For periodic boundaries $d_{i_{min}}$ has to be computed over the boundaries in order to get the right distance to the next vessel.

To assign the centers of the vessels we have investigated two methods, described in the following sections.

Scattered grid. By using the scattered grid method n vessels will be placed in the domain so that the distance in x- and y-direction is equal for each vessel. Afterwards each vessel will be moved a certain distance into a random direction. The choice of the direction is done using a uniform distribution, while the distance is chosen from a normal distribution with $\mu = 0$ and σ^2 equal to the value specified by the user. An example for the application of this method is given by the following Figure (1) which can be compared to images of vascular structures (see DASU 2004 [1] and also KONERDING ET AL. 1999 [9]).

In applications, one can see that a higher variance results in a more unstructured grid, whereas a small variance leads to an almost equidistant distribution of blood vessels.

An advantage of this method is that the number of vessels is only dependent on the mean and is therefore the same for different values of σ^2 . A drawback of this is that the mean distance between the vessels will be significantly smaller than specified by the user because of the scattering.

Dart throwing. In the dart throwing method for each vessel a normal-distributed distance will be chosen. Afterwards the vessel is placed at a random position and it is tested if no other vessel lies inside a radius of the chosen distance. If this test fails, the vessel is removed and placed at another random position and so on.

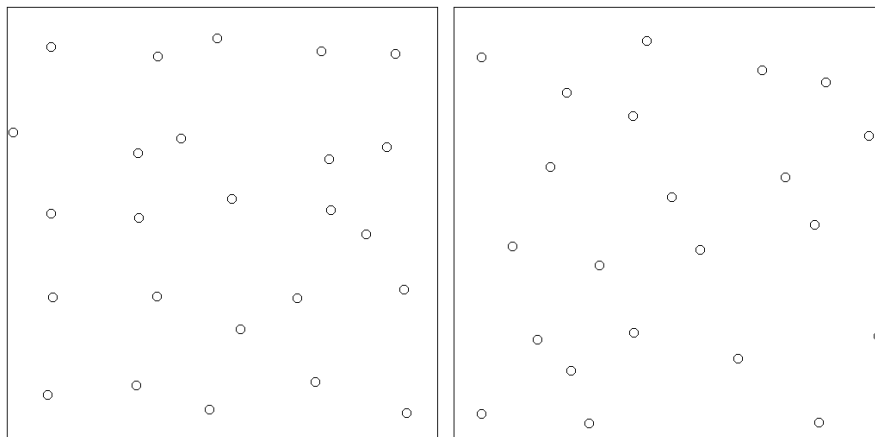


Figure 1: Vessel distribution, left: scattered grid, right: dart throwing

The number of vessels inserted by this method is smaller than the number of vessels placed by the scattered grid method. Therefore, the amount of placed vessels can vary in a large range so the results are less comparable. In contrast to that the value for the variance and especially the mean is closer to the ones the user specified.

3.3. Discretization. The first step when solving a non-linear partial differential equation is the discretization of the related domain and the differential operator, which will be discussed in the following sections.

Domain discretization. In this project, a regular grid is used for the discretization of the domain of interest. The granularity can be chosen by a certain grid spacing, or resolution, that represents the distance between two adjacent points on the grid. For the sake of simplicity, the grid spacing $\Delta x = \Delta y = h$ is equal in both directions and has a constant value in the whole domain, therefore adaptivity is not taken into account. The motivations are the reduced implementation effort and the large memory consumption of an adaptive grid. The total number of points of the domain is therefore given as $n = (\text{length}/h + 1) \cdot (\text{width}/h + 1)$. Note that n grows quadratically if the resolution h gets smaller, which is important for memory consumption throughout the whole development process of our software project.

Vessel discretization. In order to translate the continuous model into a discrete model, one must choose a way to discretize the vessels, which are modeled as disks in the continuous model and serve as boundaries for the domain. In the mathematical model a vessel is represented by the coordinates of its center, its diameter and its internal pressure. Different pressures in the vessels correspond to different oxygen distributions, so that one could also use a certain pressure distribution in an advanced model. The center is placed somewhere in the domain and does not necessarily have to be a grid point. The discretization of the vessels is now the task to identify those grid points that belong to the vessel. These points later have a constant pressure value defined by the pressure value of the vessel they belong to. Such a point is called a *vessel point*. There are many different ways of choosing the correct vessel points. During our project we investigated three methods, but we will only present here the method that proved the best suited.

For the vessel points one chooses every grid point that is geometrically inside the vessel and adds all those grid points that are adjacent to one of the others and are also not more than $\frac{\Delta x}{2}$ away from the circle, either in x- or in y-direction. This means that in addition to the points geometrically inside the vessel, the distance from the intersections of the grid lines with the circle and the grid points has to be computed and this distance has to be smaller than $\frac{\Delta x}{2}$ in order to exclude points that are quite far away from the vessel.

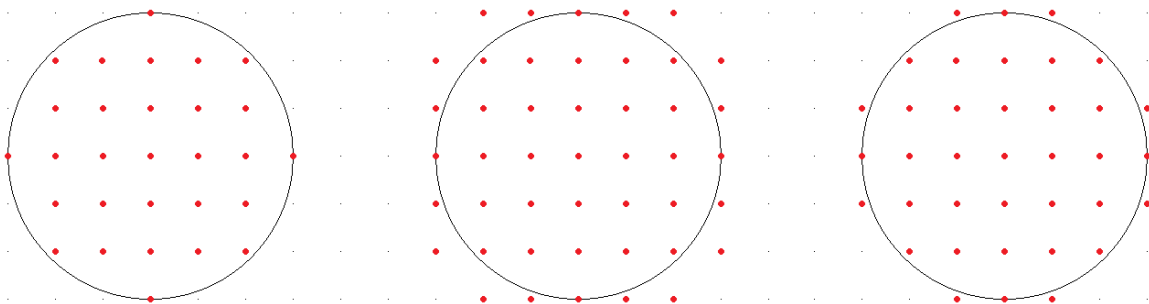


Figure 2: Discretization results of two standard methods (left and middle) and our new method (right)

Figure 2 shows a vessel of diameter 6 with $\Delta x = 1$ where the center of the vessel is exactly at a grid point. When using the above discretization method, one can see that it does a good job in this example, because it only chooses a grid point outside the vessel if it is sufficiently near to the vessel. The two standard methods on the left and in the middle do not show similarly good results.

The behavior of this discretization method is investigated in detail in Section 4. Several tests have been done and the error between the exact and numerical solution has been computed in order to characterize the method with respect to the discretization error and rate of convergence.

As for the storage of the vessel points, we use a specialized data structure. Instead of storing every vessel point, for each row of a vessel we store the index of the first and last vessel point of that vessel in that row. In addition, we only need the number of the vessel, to be able to distinguish between the vessels and assign individual pressure values or delete vessels from the data format afterwards. Due to this, we have developed a new approach to simulate acute hypoxia.

It is important to state, that our special data structure reduces the memory consumption overhead from quadratic to linear.

Equation discretization. When working on a Cartesian grid with equal distances in each direction a finite difference approach is simple and efficient because one can easily exploit information on adjacency and the structure of the emerging matrices.

The terms of the equation (3) have to be evaluated at the individual grid points. In order to approximate the differential Laplacian, we use the standard, centered five point stencil. For an equidistant grid with $\Delta x = \Delta y = h$, this simplifies to

$$(4) \quad -\Delta p|_{i,j} \approx \frac{-p_{i-1,j} - p_{i,j-1} + 4p_{i,j} - p_{i,j+1} - p_{i+1,j}}{h^2} = -\Delta_h p|_{i,j}.$$

3.4. Structure of discrete system. An important part of the whole problem is the assembly and solution of the system of equations that has to be solved numerically. In order to understand the difficulties and our solution approach better we now explain the structure of the equation system.

As an example for the structure of the system matrix A and the right hand vector b , we consider the problem (3). By using the discrete Laplacian from (4), one can write the resulting linear system of equations in matrix vector form $Ap + q(p) = b$ after assembling all the unknowns on the left hand side and all the known boundary values on the right hand side. Let the oxygen pressure value inside the vessel be denoted as p_0 and the distance between adjacent grid points be h . Furthermore, we assume homogeneous Dirichlet boundary conditions.

The consumption term reads

$$(5) \quad (q(p))_i = 0, \text{ index corresponds to a vessel point,}$$

$$(6) \quad (q(p))_i = \frac{p_i}{p_i + k} \text{ otherwise.}$$

For the explanation of the structure of the matrix, we first consider the so called *vessel points*, which are located inside the vessel. These points have a constant predefined oxygen pressure value p_0 , so the equation for those points is just

$$(7) \quad p_i = p_0$$

Therefore, the matrix has only a 1 on the diagonal in the *vessel* point-rows and the right hand side vector b has a p_0 in the corresponding entry.

As the pressure at these vessel points is already known, the contributions from these points to the other equations are moved to the right hand side vector b . So for every vessel point neighbor of an arbitrary point, one has to add p_0 to the corresponding entry in b . The other values emerge from the standard five point stencil for the Laplacian (4).

3.5. Numerics. When using finite differences for the Laplacian the non-linear, partial differential equation (3) becomes a non-linear system of equations

$$(8) \quad f : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad f(p) = DA p + q(p) - Db = 0$$

with n as the number of points. Here, A and b belong to the linear system of the Laplacian.

Note that it is crucial to set $q(p) = 0$ for all points in the vessels to guarantee that equation (7) holds for these points. To solve this equation numerically we use Newton's algorithm as it is a

simple and low memory-consuming algorithm with a good speed of convergence.

Newton's algorithm is a special form of a fixed-point iteration described by the equation

$$(9) \quad p_{k+1} = p_k - (f'(p_k))^{-1} f(p_k)$$

with a start value p_0 and

$$(10) \quad f' : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n} \quad f'(p) = D \cdot A + q'(p).$$

In (10) $q'(p)$ describes the differentiated consumption term (2):

$$(11) \quad q'(p) = q_{max} \cdot \frac{k}{(p+k)^2}.$$

As (10) is a convex function with values of $q'(p)$ between 0 and q_{max} , a good convergence of Newton's algorithm is guaranteed. In general its convergence order is quadratic (see Section 4.3).

Now the task is to solve a linear system of equations

$$(12) \quad Ax = b \quad \text{with } A \in \mathbb{R}^n \times \mathbb{R}^n, b \in \mathbb{R}^n$$

corresponding to $f'(p_k)y = f(p_k)$ and update the solution vector p for every iteration step.

To handle the challenge of a low memory consumption we have to use appropriate methods for the solution of the linear system. Therefore, we especially do not want to store the large matrix A of the equation system. This can be achieved by using methods working on the evaluation of the matrix by multiplying a vector so that the matrix-vector product can be directly evaluated. The direct evaluation of these matrix-vector products without storing the matrix takes advantage of the good structure of the matrix introduced in Section 3.4. On top of that, the boundary values of the vessels are directly included in the matrix-vector product which is a benefit for the issue of memory consumption. The evaluated product is stored in a vector. So our approach is to use suitable numerical methods that only use operations on matrices resulting in vectors. There are several ways to treat linear equation systems (12) numerically. We choose two different iterative methods instead of a matrix decomposition. These are the Conjugate Gradient Method and the Conjugate Residual method which are both Krylov subspace methods. Both methods guarantee convergence for symmetric positive definite matrices. As the derivative of the consumption term added to the diagonal of the system matrix discussed in Section 3.4 is nonnegative, the resulting matrix is symmetric positive definite. For more information on the convergence see Section 4.3 and MEISTER 2008 [8].

4. RESULTS

Having modeled the problem and developed a specialized software we can now present our first simulation results of tumor oxygenation. Additionally, we first validate the numerical solution against a simplified model problem and derive a simple but efficient a priori error estimator. We investigate a novel approach of modeling acute hypoxia and take a look at the convergence and runtime of the numerical methods used during the simulation. As memory consumption has been identified as the main bottleneck in recent applications concerning tumour oxygenation (see DASU 2004 [1]), we show that our implementation is able to reduce memory storage significantly. The following tests were carried out on standard personal computers with between 3 and 8 GB of random access memory and a CPU frequency of 2.67 GHz.

4.1. Validation.

Exact solution of simplified problem. In order to validate the program and to investigate the differences between various settings, especially with respect to the discretization methods, one needs an exact solution of the problem to compare it to the numerical results. However, deriving an exact solution for the non-linear partial differential equation is not trivial. Therefore, we restrict ourselves to a simpler problem.

The non-linear PDE reads

$$(13) \quad -D\Delta p + q(p) = 0$$

Now we assume $q(p) \equiv 0$, which means that we have no oxygen consumption. This simplification is valid for very small values of p , as the consumption term $q(p) = q_{max} \cdot \frac{p}{p+k}$ goes to zero for vanishing values of p .

Notice that even though the PDE is simplified compared with the non-linear equation, it still contains the important Laplacian. By neglecting the non-linear consumption term the type - and therefore also the behavior of the equation - is the same. Thus, by the investigation of different discretization methods, one can get results that are also valid for the general case.

Furthermore, we need to prescribe boundary conditions on a domain in order to make the derivation of an exact solution possible. For the domain, we choose the area between two concentric circles. As boundary values we choose constant values p_0 for the inner boundary and P_0 for the outer boundary.

With the help of polar coordinates, the solution reads as follows

$$(14) \quad p(x, y) = \frac{p_0 - P_0}{\ln(r_0) - \ln(R_0)} (\ln(\sqrt{(x - x_0)^2 + (y - y_0)^2}) - \ln(r_0)) + p_0$$

Here (x_0, y_0) denotes the center of the vessel. As expected p decreases from p_0 on the inner circle to P_0 on the outer circle.

Comparison of exact and numerical solution. With the help of the exact solution we can now investigate the quality of the numerical solution. In order to obtain the same PDE we set $D = 1$ and the parameter q_{max} is set to zero, therefore the consumption term in the equation vanishes as well. Now we embed a quadratic numerical domain into a disk and place a vessel in the middle. With the help of the exact solution it is now possible to compute the values of p at the boundary of the numerical domain, since the numerical domain is enclosed by the circle with radius R_0 . Using those values as Dirichlet boundary conditions for the numerical simulation, we can compare the exact solution with the numerical solution afterwards.

In order to examine the convergence rate as a function of Δx and the discretization method, we compare different test cases. For a first comparison between different discretization methods, we use a grid of size $200 \times 200 \mu\text{m}$ and let the vessel in the center have radius $r_0 = 20 \mu\text{m}$. The exact solution on the domain can be computed using the formula above, resulting in the oxygenation map in Figure 3 on the left. The pointwise error on the whole domain can be seen in Figure 3 on the right.

As a first observation, the overall error is quite small compared with the other methods which we tested separately. Furthermore, the error occurs in the region near the vessel boundary, where certain areas show a larger error than others. Here the error is large near the upmost part of the vessel, for example. The large error in those regions occurs due to the distances of the vessel points to the real vessel boundary. This is also comparable to the discretization results in Figure 2.

The overall relative error can additionally be derived and is 0.00108 for our discretization method and the specific test case. In fact, a relative error of about 0.1% is quite a good result for a resolution of only $1 \mu\text{m}$.

In order to improve the significance of these first results we extended the test using different settings. Here we present a second test that used a $200 \times 200 \mu\text{m}$ domain and $r_0 = 30 \mu\text{m}$ in order to have larger vessels and to have more boundary effects in the results. We computed the absolute

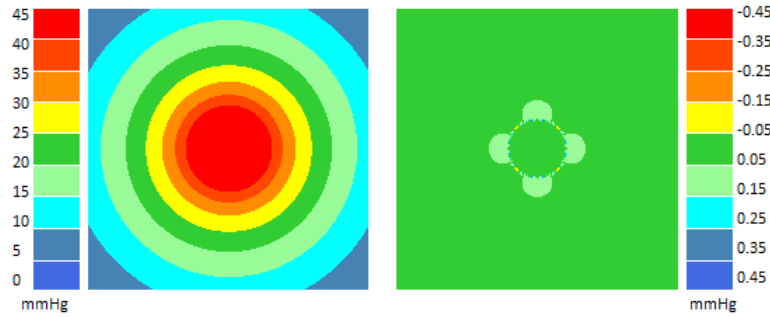


Figure 3: Left: Exact solution for the first test. Right: Error distribution for first test.

error for Δx varying from $8\mu\text{m}$ to $\frac{1}{32}\mu\text{m}$ and present the results in a double log-plot in order to obtain a rate of convergence easily. The results of this test can be seen in Figure 4.

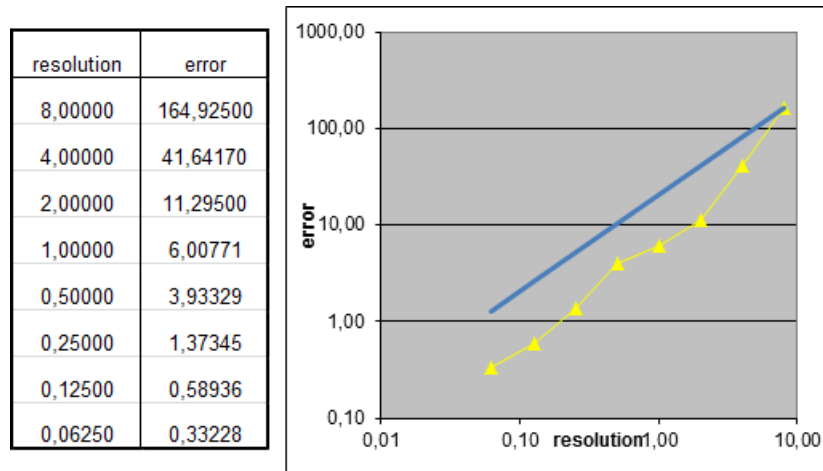


Figure 4: Error for test on $200 \times 200\mu\text{m}$ domain with vessel of radius 30 (yellow line), first order convergence for comparison (blue line)

Concerning the rate of convergence one can easily compute a curve fitting by linear regression of the double log data from 4. After that the slope of the line is the rate of convergence for the specific test case and method. This yields a numerical convergence rate of 1.2277. We also obtained similar results with other test cases. For comparison, Figure 4 also shows a line with slope 1. One can see, that the numerical method yields a better rate of convergence than one.

In summary, it is shown that the numerical solution of our software converges to the exact solution. This has been verified by comparison with the exact solution of a simplified problem. We were able to obtain a convergence rate of 1.2, which is a quite good result compared with a minimum memory storage overhead.

4.2. A priori error estimator. In practical applications one is interested in a priori error estimates in order to choose the right resolution. The choice of the resolution is very important, because it determines the number of grid points and therefore also the computational effort needed to solve the problem. So if one wants to get a numerical solution that is sufficiently accurate, it makes no sense to choose the smallest resolution possible because one then needs a lot of time to get this solution. It is reasonable to look for the largest resolution that will give the numerical solution that has the desired accuracy.

For the derivation of an a priori error estimator, we assume linear convergence as shown by the comparison of the exact and numerical solution in Section 4.1. As we are concerned with calculating the number of hypoxic cells inside a tumor tissue, a possible criterion for the accuracy of the solution could be, how well the fraction of these hypoxic cells is approximated by the numerical solution. Similarly, one could take the fraction of vessels that have a certain oxygen pressure value. So we take a function $\tilde{f} : \mathbb{R} \rightarrow \mathbb{R}$ that depends on the resolution. The value $\tilde{f}(h)$ represents the percentage of values within between 20 mmHg and 25 mmHg. The numerical solution gives this approximate value. The percentage of the exact solution is denoted by $f = \tilde{f}(0)$. Now, one wants to estimate the error

$$(15) \quad |\tilde{f}(h) - f|.$$

For the a priori error estimator, we employ the following ansatz for the calculation of the difference between two resolutions

$$(16) \quad |\tilde{f}(h_1) - \tilde{f}(h_2)| \approx C \cdot |h_1 - h_2|^k$$

where C is a constant and k is the rate of convergence. The error in (15) can then be derived as

$$(17) \quad |\tilde{f}(h) - f| = |\tilde{f}(h) - \tilde{f}(0)| \approx C \cdot |h - 0|^k$$

The value of C still has to be computed by comparison of different resolutions. Therefore we follow the estimation ansatz (16) and calculate C using two different resolutions as follows

$$(18) \quad C \approx \frac{|\tilde{f}(h_1) - \tilde{f}(h_2)|}{|h_1 - h_2|^k}$$

In order to get a value for C we will further assume $k = 1$. Nevertheless, the following derivation is also possible without this assumption. After having derived a value for C the a priori error estimation reads

$$(19) \quad |\tilde{f}(h) - f| = |\tilde{f}(h) - \tilde{f}(0)| \approx C \cdot |h - 0| \approx C \cdot h.$$

The values $\tilde{f}(h_1)$, $\tilde{f}(h_2)$ and $\tilde{f}(h_3)$ have to be computed by numerical simulations. Therefore we consider the following test case: Take a domain of size $400 \times 400 \mu\text{m}$ with vessels of radius $5 \mu\text{m}$ and a mean distance of $120 \mu\text{m}$ using the scattered grid distribution. Then we simulate the oxygen distribution by running the software and calculate the percentage of values within certain ranges of oxygen pressure values.

In our simulation, one can see that most of the cells have values of about 5 mmHg – 10 mmHg and only very few cells have very high values. But the most important fact is that the percentage of values seem to stabilize for finer meshes. The values of the numerical solution get nearer to the exact values, so one can assume that the exact value is approximated in the limit $h \rightarrow 0$, corresponding to $f = \tilde{f}(0)$. Furthermore, the values are still changing a lot for resolutions $8 \mu\text{m}$ and $4 \mu\text{m}$. When the resolution is about $0.5 \mu\text{m}$ the values are almost the same as for finer grids. So a reasonable value for the resolution seems to be $0.5 \mu\text{m}$ because smaller values do not increase the accuracy of the simulation very much and lead to higher computational costs.

It is now possible to estimate the value of C using the values from the simulation in order to compute an error bound and choose the resolution according to that. Here we choose $\tilde{f} = \widetilde{f_{0-5}}$ to be the percentage of hypoxic cells, specifically the percentage of cells that have an oxygen value of 0 mmHg – 5 mmHg.

Deriving C from equation (18), one obtains different values depending on the choice of h_1, h_2 . The values are different from each other, so we take the maximum of the relevant values for our estimate:

$$(20) \quad C = 7998 \approx 8000.$$

So, for a relative accuracy of $1\% = 0.01$, it is approximately sufficient to use a resolution of $0.25 \mu\text{m}$ according to the following derivation; here we assume the exact value $f_{0-5} \approx 0.25$ which seems reasonable when compared to our simulations where f_{0-5} stabilizes at about 0.24:

$$(21) \quad \frac{|\widetilde{f_{0-5}}(1) - f_{0-5}|}{|f_{0-5}|} \approx C \cdot \frac{|0.25 \mu|^1}{|f_{0-5}|} = 8000 \cdot \frac{0.00000025}{0.25} = 0.008 < 1\%$$

Note that, as this is only an error *estimate*, it does not necessarily hold for every application, but it can give useful information on how to choose the resolution in general. It is also important to say that the values of C differ, when considering different functions for \tilde{f} , for example the percentage of oxygenated cells (in our example the other values were smaller, which means that the choice of $0.25 \mu\text{m}$ is still quite conservative).

We expect the results to be valid for other applications with larger domains and more vessels. Therefore, the user is now able to choose an appropriate value for the resolution without calculating a numerical solution first.

4.3. Convergence results for numerical methods. Memory consumption is not the only consideration, we should also keep in mind the runtime of the simulation. Therefore it is important to analyze the assumptions made on the convergence of the different numerical methods used.

Convergence of Newton's algorithm. The inequality

$$(22) \quad \|p_{k+1} - p^*\| \leq C \cdot \|p_k - p^*\|^2$$

holds for Newton's algorithm under several conditions. First, the Jacobian $f'(p)$ should be invertible and Lipschitz continuous. Furthermore, the initial values should be chosen in an appropriate way which is discussed later on. Looking at the Jacobian $f'(p)$ given by (10), we can directly state that it is invertible as the matrix occurring from the discretization has full rank. Additionally we state for a $p \geq 0$ and $s \geq 0$,

$$(23) \quad \|f'(p) - f'(s)\| = \|q'(p) - q'(s)\| \leq \|q''(\xi)\|_\infty \cdot \|p - s\| = L\|p - s\|.$$

As $q''(\xi)$ is bounded in the domain of the solution, the Jacobian is Lipschitz continuous. Therefore we expect local quadratic convergence for special start values.

To analyze the convergence for the model one first must think about appropriate start values. As the exact solution should show the chosen pressure values for the points belonging to the vessels, the idea is to assign this pressure already to these points in the vector of start values. Our tests validate this idea.

In order to analyze the occurring residuals for each step we make use of the ∞ -norm of the evaluation of the right hand side of the non-linear system as it is independent of the size of the calculated area.

Table 1 shows the decrease of the residuals for different domain sizes or matrix dimensions. Here the values in the first row stand for the length of the the quadratic area in mm^2 . One can get an impression of how quick Newton's method converges for even larger systems. It becomes clear that the number of iterations does not depend on the dimension of the system to be solved. But for the case of the area of $1 mm^2$ corresponding to a dimension of 1002001 there are more iterations needed than for the other dimensions. This is due to the fact that in the case of a number of 64 vessels with a radius of $5 \mu m^2$ the resulting area has a lower mean intervacular distance. Therefore the full range of the different pressure values is given in the whole area. In all other cases there are more

Iterations	1000	2000	3000	4000
	180000	180000	180000	180000
1	254.843	242.998	243.065	243.064
2	21.2117	3.99768	3.55857	3.55476
3	8.25828	0.768127	0.176955	0.176499
4	1.04148	0.096785	0.00931269	0.000825185
5	0.038891	0.000894985	9.87851e-05	1.56672e-06
6	0.000124126	3.05714e-07		
7	4.82274e-07			

Table 1: Decrease of the residual for different domain sizes or matrix dimensions

areas of low pressure so that the values for all points do not differ that much. This explains why the change, as compared with the starting vector is smaller for these cases. As a result, the number of iterations depends on the mean intervascular distance in the area denoted by the variance and, in particular, the mean of the used distributions. In DASU 2004 [1], a representative value of $100 \mu\text{m}$ for the mean intervascular distance is given.

For an order of convergence higher than 1, the estimate $\|p_k - p^*\| \approx \|p_{k+1} - p_k\|$ holds.

In table (2) we look at the same setup as in (1) for the area of 1 mm^2 . One notices that, for the area densely filled with vessels, the quadratic convergence can be seen after 6 iterations as the number of correct digits doubles.

Iterations	Norm of y_k
1	43.4475
2	18.337
3	11.4244
4	3.10141
5	0.3243
6	0.00404526
7	0.00000847614

Table 2: Norms of the update vectors for Newton's algorithm

Convergence of Conjugate Gradients and Conjugate Residual. In the following we look at the convergence of the linear numerical methods. Theoretically, both used methods should converge to the exact solution in n iterative steps where n , as usual, denotes the dimension of the system of equations. As already mentioned, the solution should be in the range of the approximation behavior that can be reached by the discretization. So one gets a lower number of iterations than n . Table (3) shows different numbers of iterations for different dimensions and different means for the distribution of the vessels. They were calculated by using a residual of 0.0001 as a stopping criterion.

Dimension	1002001 (*)	9006001	9006001 (*)
Mean Iterations	780	435	802

Table 3: Mean number of iterations for the CG-method

The (*) denotes an area with a lower mean intervascular distance. Again, one notices that the number of iterations to reach a certain residual does not depend on the dimension but on the mean intervascular distance. This is the same result as for the number of iterations of Newton's algorithm. Note that the start value for each iteration is consistent with the solution y of the linear system

for the old iteration so that the change of the old Newton step does not get lost and the starting residual of the used linear method decreases during the iteration of the non-linear solver.

From experience, we know that other iterative methods, such as the Jacobi method, are not as efficient as the CG-method and the CR-method since they need a much higher number of iterations.

To investigate the speed of convergence of the two methods used one can have a look at Figure (5).

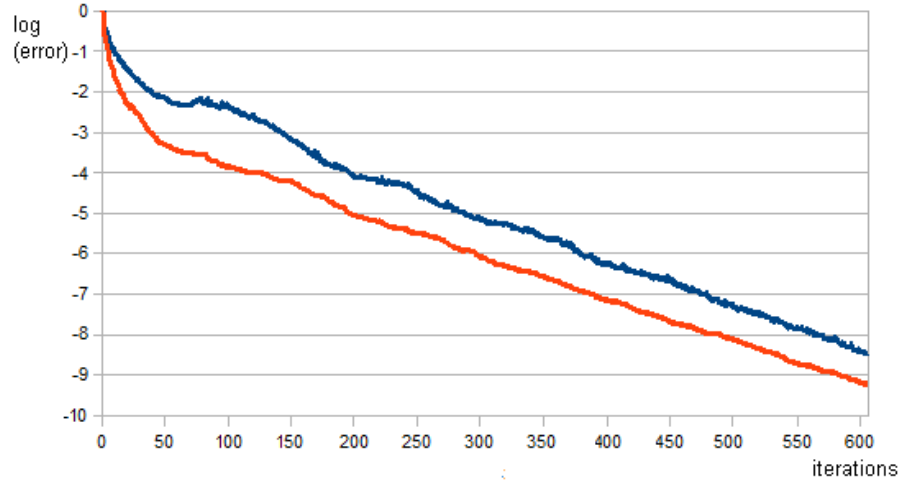


Figure 5: Development of residual of CG (blue) and CR (red)

The graph plots the normed error

$$(24) \quad \log_{10}\left(\frac{\|r_k\|_{\infty}}{\|r_0\|_{\infty}}\right)$$

over the number of iterations. Both methods show a linear progression of -0.01 normed errors per iteration denoting that they have the same speed of convergence. But one can see that for the first iterations the Conjugate Residual method converges faster than the Conjugate Gradient method so that it needs fewer iterations. Note that the convergence is measured in the ∞ -norm. There will be different results when measuring in the 2-norm or in the A -norm, as the Conjugate Gradient method minimizes the A -norm of the error, whereas the Conjugate Residual method minimizes the 2-norm of the residual. These results can be validated by similar results in RABENSEIFNER 2010 [8].

4.4. Memory requirements and runtime.

Memory requirements. Now it is important to check whether the goal of the project to minimize memory requirements to ensure the calculation of large areas of tumor tissue is successfully met. As the information on the discretization needs a small amount of memory that depends on the area of interest and the resolution, the solution of the non-linear system is the most memory-intensive part of the calculation. In the following, we will look more closely at the memory required by the non-linear and linear numerics.

To calculate a step of Newton's algorithm as described in Section 3.5, one needs two vectors of dimension n , which denotes the number of points occurring from the discretization. Therefore they depend on the size of the area of interest and not on the number of vessels in this area. This holds for all vectors used while solving the system of equations which can be seen in table (4). It shows the memory consumption of the solution methods on an area of 1 mm^2 for different numbers of vessels. Here one can observe that the used memory does not depend noticeably on the number of vessels in

the domain.

Vessels	1	64	225
Peak Memory	56148 KB	58042 KB	59988 KB

Table 4: Peak Memory for different numbers of vessels

As we expect it to be, Figure (6) depicts the dependency of the used memory on the dimension. Here, the red bars represent the memory used by the numerical methods and the blue bars the additional memory. The additional memory increases linearly with the grid constant, whereas the largest amount of the memory increases quadratically with the grid constant and, therefore, linearly with the dimension.

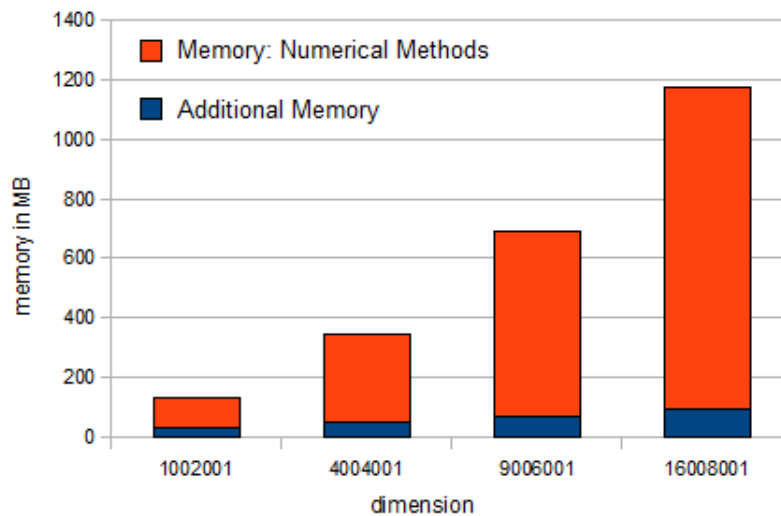


Figure 6: Peak memory for different dimensions

To understand the amount of memory needed one should account for all the vectors used. The numerical approaches to the linear system utilized are very low memory consuming. Both the CG and the CR method make use of several vectors of the length n . Instead of storing the matrices of the order of n^2 the needed matrix vector products are directly evaluated so that the costs of memory are reduced to the order of n .

The convenient algorithms for the Conjugate Gradient method and the Conjugate Residual Method can be improved by reusing the old values for the correction factors and for the matrix vector products of the last iteration step so that the number of needed vectors can be reduced from 5 to 3.

To calculate the residual, the linear and the non-linear solver need additional memory to store the residual vectors for a short time. Further memory storage is required by the evaluation of the matrix vector products. Due to this, one needs a maximum of 3 n -dimensional vectors in addition to the 2 of Newton's algorithm. When regarding the additional vectors for the linear solver, more vectors turn into account. Here one gets a maximum of 4 vectors additionally to the existing 5.

To summarize these results, after optimizing the algorithms one gets a maximum storage of 9 n -dimensional vectors from the numerical methods. This is a promising result, as we have a linear dependence of memory on the dimension. On top of that, the optimizations can be seen in the peak memory. When looking again at Figure (6) and comparing these results to the peak memories of

the program using the unoptimized algorithms for the numerical method one can clearly see the effects of optimization. Before, one needed 407 MB of main memory as a peak value for a dimension of 4004001, the peak memory of 297 MB for the optimized algorithm is a reduction of about 25 percent. Similar results occur for a dimension of 9006001. Here the old results for the peak memory were 760 MB compared to the new peak memory of 620 MB.

Runtime. As already discussed in Section 4.3 Newton’s algorithm converges quadratically. Also, the algorithms for the solution to the linear system ensure a good order of convergence as discussed in 4.3. For each step of these algorithms only up to two evaluations of the matrix vector product at the cost of $O(n)$ operations have to be computed. We achieve this by exploiting the sparsity of the system matrix as specified in Section 3.4. To get a better sense for the runtime of the program we will look at the runtime during solving the system of equations given in the table (4.4).

Dimension	1002001 (*)	4004001	9009001	9009001 (*)	16008001
Time in min	20	58	76	106	122

Table 5: Runtime for different dimensions

Here one can see the linear dependence on the dimension of the system as well as the influence of the number of iterations for Newton’s algorithm, as the values with (*) represent runtime calculations for areas with a dense structure of vessels.

To increase the speed of convergence of these iterative methods one has the possibility to use preconditioning algorithms. In the case of the Conjugate Gradients method, a possible method would be the Preconditioned Conjugate Gradient method. As the focus of the project is set to minimizing the used memory a preconditioning method is not optimal for several reasons. First, preconditioning means additional storage reserved for the preconditioning matrix and its evaluation. Another problem is the destruction of the simple matrix introduced in Section 3.3 so that the introduced data structure for the discretization becomes useless. The more simple and efficient way to decrease the overall runtime would be to parallelize the program. For the used iterative methods, this can be easily done as they are based on simple for-loops. Since memory-consumption is still an issue in the program, it can benefit from the use of distributed memory clusters and *MPI* would be the best approach. When using smaller cluster machines with large shared memory, the program could be parallelized using *OpenMP*.

5. APPLICATION RESULTS

5.1. Results of modeling hypoxia. Section 2.2 gives information on the two different forms of hypoxia. Our program provides several new ways to model hypoxia in an easy way due to the advantage that all parameters can be directly altered by the user and information on the radius and the pressure of each vessel is stored.

Chronic hypoxia results from limited diffusion. This is the form of hypoxia that is directly calculated in the model. To investigate the effects of chronic hypoxia, one can limit the diffusion even more by changing the diffusion coefficient D .

Looking at the results of Figure (7) with a diffusion coefficient of 2000 and Figure (8) with $D = 1000$ that were calculated in a 1 mm^2 tissue one can see the impacts of chronic hypoxia on the oxygenation of the tumor tissue. The left part of the Figure shows an oxygenation map showing the values of the oxygen pressure for each point of the discretization. The red values stand for the high pressure in the vessels whereas blue colors indicate areas of low pressure. On the right hand side of the Figure one can see a histogram showing the percentages for the different pressure values of the oxygenation map. In Figure (8) the histogram is shifted to the left since the oxygenations map contains more areas of low pressure.

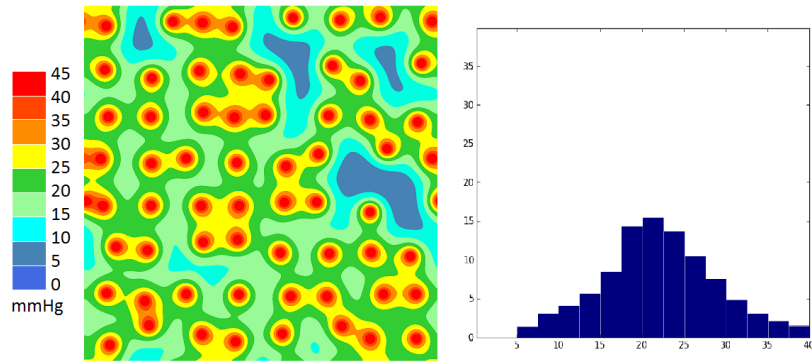


Figure 7: Tumour tissue ($D=2000$)

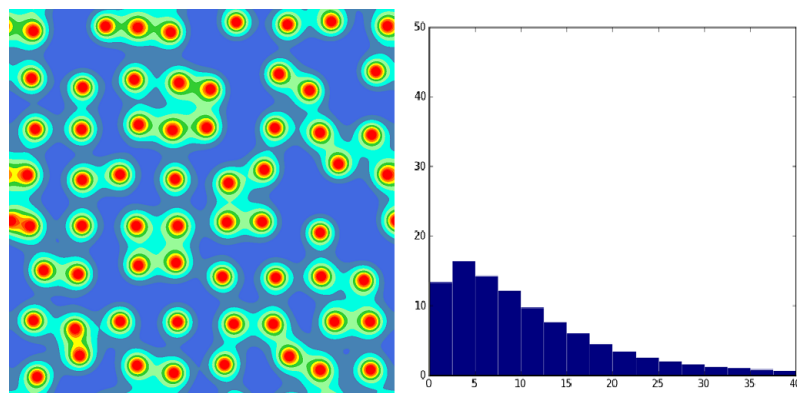


Figure 8: Tumour tissue ($D=1000$)

Another form of hypoxia is acute hypoxia, which was characterized in Section 2.2. In the model, there are different particular parameters for each vessel. The most apparent way to model less perfusion is to delete an arbitrary number of vessels in the tissue.

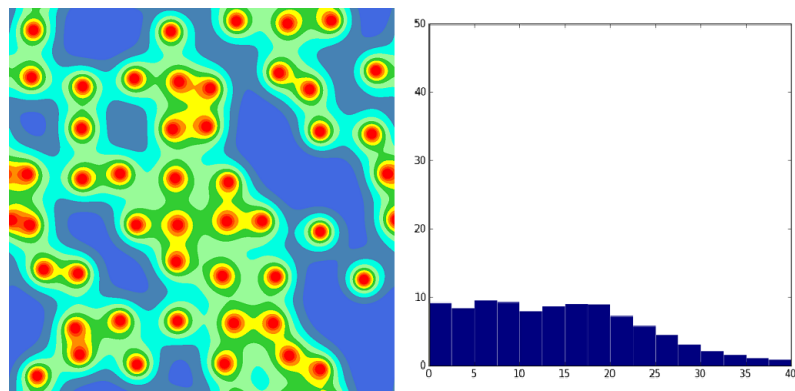


Figure 9: Tumour tissue with acute hypoxia for 25 percent of the vessels

In Figure (9) 25 percent of the blood vessels were randomly deleted. The results of this are larger areas of low pressure.

The program offers even more ways to simulate acute hypoxia. For example the radius of some vessels can be decreased possibly representing partially blocked vessels shown in Figure (10). Here,

instead of a radius of $10\ \mu\text{m}$, some vessels have only $2\ \mu\text{m}$. It is possible to prescribe different sizes for each vessel in the domain.

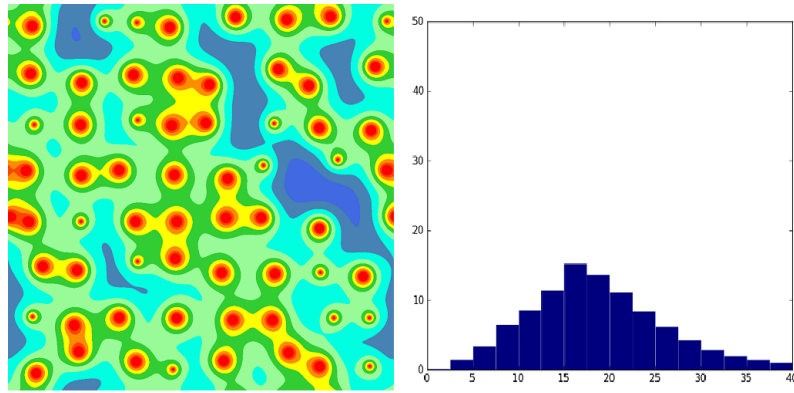


Figure 10: Tumour tissue with acute hypoxia modeled by radius

Less blood flow through the vessels can also be modeled by assigning lower pressure values to the affected vessels. Assigning $5\ \text{mmHg}$ instead of $45\ \text{mmHg}$ to some vessels, as done in Figure (11), leads to similar results as obtained by deleting these vessels. If one sets the pressure in the affected vessels to $27.5\ \text{mmHg}$, as has been done in Figure (12), one can observe results similar to Figure (10).

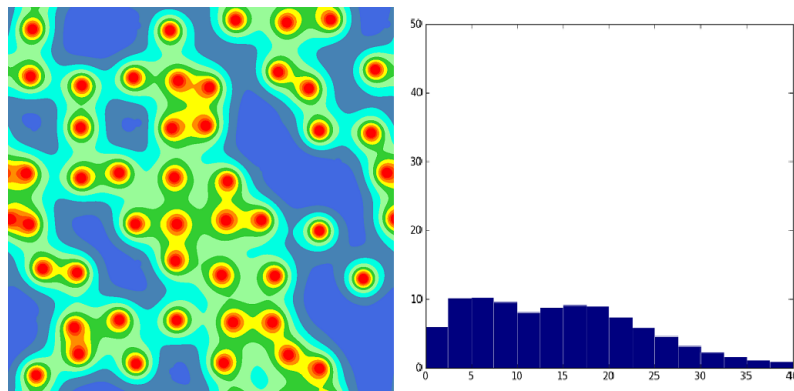


Figure 11: Tumour tissue with acute hypoxia modeled by pressure $p = 5\ \text{mmHg}$ in hypoxic vessels

Many tumours consist of necrotic cores in the center. This region is in general very different from normal tumour tissue. We have not covered this topic in this paper. General valid assumptions for modeling such necrotic regions are to alter the diffusion coefficient and to set the consumption term to zero, as there are no living cells consuming oxygen inside the core.

5.2. Application Example. As a final result, we present the simulation of tumor oxygenation in a large domain of interest. We consider a domain of length and width $3500\ \mu\text{m}$. Real tumors measure about $3 - 4\ \text{cm}$ in diameter. The resolution during simulation is $1\ \mu\text{m}$, corresponding to the error estimator from section 4.2. So the number of hypoxic cells should be accurately resolved. For the distribution of the vessels inside the domain we assume a mean intervascular distance of about $100\ \mu\text{m}$ together with a variance of circa $300\ \mu\text{m}^2$. This distribution results in a total number of 1296 vessels. The simulation on our desktop PC took approximately 30 minutes.

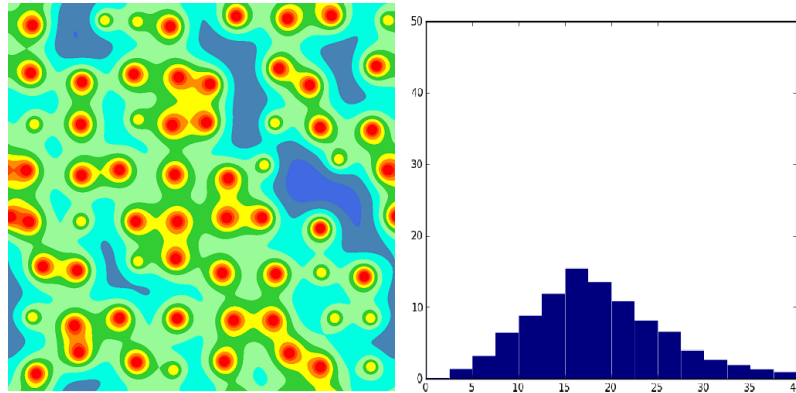


Figure 12: Tumour tissue with acute hypoxia modeled by pressure $p = 27.5$ mmHg in hypoxic vessels

The numerical solution calculated using our program is shown Figure (13). One can see that the tissue consists of large areas with hypoxic cells (represented by dark blue color). On the other hand, there are regions with many vessels resulting in a high oxygen value. The figure shows, that hypoxia only occurs in regions that show a large variation of intervascular distances due to the chosen distribution. In regions where the vessels are almost equidistantly placed, the oxygen values are above the hypoxic threshold.

This example demonstrates the good performance of our software program and verifies the benefits of our work, as other tools are not able to reproduce these result due to their disadvantages regarding memory consumption. We expect that the program can solve a domain measuring 4 cm^2 by using a peak main memory of about 30 gigabytes so that tumors of a size measured in cm^2 can be modeled. Using a third dimension would not disturb the linear increase of memory consumption with the dimension of the system. An area measuring 0.4 mm^2 could be calculated using 30 gigabyte by introducing a third dimension.

6. CONCLUSION

Summarizing everything, we have successfully shown that it can be reasonable to invest time to design numerical software for an explicit problem equation and problem domain. Our aim was to develop a software program that overcomes the problems of high memory consumption needed for storing the grid. Having achieved this, computations on larger domains can be started.

Our approach shows that very simple mathematical methods can be better than the more developed methods used in commercial tools. The impact of this can be seen in the discretization step very clearly. Instead of using a very irregular grid, a simple equidistant grid reduces the memory consumption greatly. One may need a finer grid resolution and therefore many more grid points, but for each grid point very little information has to be stored. Our investigation of the discretization of the vessels also achieves, that the circular borders can be represented accurately enough on such a regular grid.

Using this simple but still flexible ansatz, we were able to design a new data format to save the boundary information efficiently. Thus, it was possible to reduce the memory consumption even more, so that the number of vessels inside the domain no longer dominates the memory consumption of the whole simulation, as was the case for commercial software packages.

Using a standard finite differences approach, we could exploit the structure of the emerging system of equations in order to solve the system quickly with the help of Newton's method and a CG or CR method.

For the validation of the model and the numerical solution we simplified the problem for the linear case and derived an exact solution. Our simulation results show the convergence of the numerical

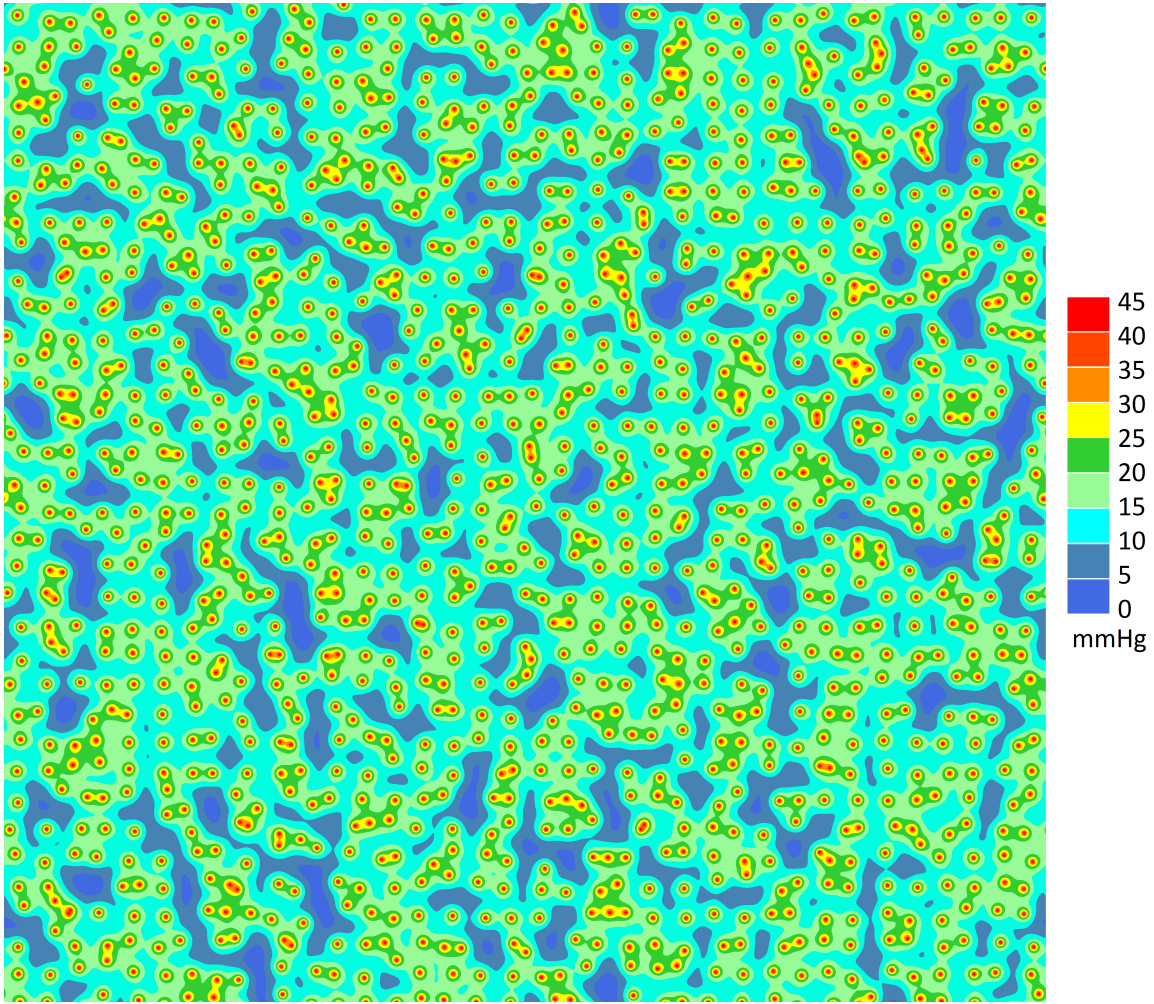


Figure 13: Oxygenation map for the simulation of tumor oxygenation on a large domain of size $3500 \times 3500 \mu\text{m}$ containing almost 1300 vessels

solution to the exact solution for vanishing resolution. In the context of this validation we obtained the desired linear convergence rate.

In order to ease practical application, we derived an a priori error estimator. With this, it is possible to choose a suitable resolution for a desired accuracy of the numerical solution. We showed that it can be enough to use a resolution of $1 \mu\text{m}$ for the accurate approximation of the number of hypoxic cells.

As for the numerics part of our project, we could verify the convergence order of Newton's algorithm for our nonlinear equation system and have investigated the CG and CR method with respect to memory consumption, runtime and convergence, too. Additionally we have shown that the choice of the initial guess has less influence on the behavior of the solvers.

It is important to note that we were able to introduce three novel approaches of modeling acute hypoxia that can possibly be used combined in order to get the most realistic results. Finally, with the help of the achievements above it was possible to simulate tumor oxygenation for tissues that have an area of several cm^2 . So far, there have only been successful simulations for tissues of about $1 \text{mm} \times 1 \text{mm}$. This means the outcome of our work on a specialized software is the step from the

mm scale to the cm scale. This result is quite meaningful because a real tumor also would have about 3 – 4 cm in diameter. In this sense, the simulation of tumor oxygenation can truly be used in practical applications.

Further work could be used to extend the model to a third dimension. The solution technique in that case would remain the same as in the two dimensional case but the generation of a domain would be more difficult as the vessels are no longer simple cylinders. Since our grid storage method can approximate arbitrary shapes in two dimensional space, it is possible to extend this to a third dimension. In three dimensions a second vessel point vector could be used, which stores a starting and end point of the vessel in the new dimension.

It is also possible to improve the model, for example, by treating the vessel boundary as a membrane. Due to the simplicity of the grid and the numerical methods our approach, an extend model can be fitted without much overhead. To gain the full advantage of the improved memory consumption of this solution, the program could also be useful to parallelize the program using *OpenMP* or *MPI*.

REFERENCES

- [1] I. DASU, Theoretical modelling of tumour oxygenation and influences on treatment outcome, Ph.D. thesis, Umeå University, 2004
- [2] J. DENEKAMP, A. DASU, Inducible repair and the two forms of tumour hypoxia - time for a paradigm shift, *Acta Oncologica*, 1999, 38, pp. 903-918
- [3] R.H. THOMLINSON, L.H. GRAY, The histological structure of some human lung cancers and the possible implications for radiotherapy, *British Journal of Cancer*, 1955, 9, pp. 539-549
- [4] A. DASU, Modelling the impact of two forms of hypoxia on novel radiotherapy approaches, Ph.D thesis, Umeå University, 2000
- [5] D.M. BRIZEL, G.S. SIBLEY, L.R. PROSNITZ, R.L. SCHER, M.W. DENWHIRST, Tumor hypoxia adversely affects the prognosis of carcinoma of the head and neck, *International Journal of Radiation Oncology Biology Physics*, 1997, 38, pp. 285-289
- [6] I.F. TANNOCK, Oxygen diffusion and the distribution of cellular radiosensitivity in tumours, 1972, 45, pp. 515-524
- [7] A. MEISTER, Numerik linearer Gleichungssysteme, 2008
- [8] R. RABENSEIFNER, Parallel Programming Workshop - Iterative Solvers and Parallelization, 2010
- [9] Konerding, M. A., Malkusch, W., Klapthor, B., van Ackern, C., Fait, E., Hill, S. A., Parkins, C., Chaplin, D. J., Presta, M. and Denekamp, J., Evidence for characteristic vascular patterns in solid tumours: quantitative studies using corrosion casts, *British Journal of Cancer*, 1999, 80, pp.724-732
- [10] A. Dasu, I. Dasu and M. Karlsson, Theoretical simulation of tumour oxygenation and results from acute and chronic hypoxia, *Physics in Medicine and Biology*, 2003, 48-17