
Proceedings Paper:
Yang, FW, Goodyer, CE, Hubbard, ME et al. (2015) Parallel Implementation of an Adaptive, Multigrid Solver for the Implicit Solution of Nonlinear Parabolic Systems, with Application to a Multi-Phase-Field Model for Tumour Growth. In: Ivanyi, P and Topping, BHV, (eds.) Civil Comp Proceedings (CCP:107). Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, 24-27 Mar 2015, Dubrovnik. Civil-Comp Press. Article no: 39. ISSN: 1759-3433.

https://doi.org/10.4203/ccp.107.39

Parallel Implementation of an Adaptive, Multigrid Solver for the Implicit Solution of Nonlinear Parabolic Systems, with Application to a Multi-Phase-Field Model for Tumour Growth

F.W. Yang[1], C.E. Goodyer[2], M.E. Hubbard[3] and P.K. Jimack[4]

[1]Department of Mathematics, University of Sussex, Brighton, UK
[2]NAG Ltd., Wilkinson House, Jordan Hill Road, Oxford, UK
[3]School of Mathematical Sciences, University of Nottingham, UK
[4]School of Computing, University of Leeds, Leeds, UK

Abstract

The first half of the paper provides an overview of a new engineering software tool that is designed for the efficient solution of problems that may be modeled as systems of linear and nonlinear partial differential equations (PDEs) of parabolic type. Our tool is built upon the PARAMESH library, [15], which provides hierarchical mesh adaptivity in parallel in two and three dimensions. Our discretizations are based upon cell-centred finite difference schemes in space and implicit multi-step methods in time (primarily the second order backward differential formula (BDF2)). This results in the need to solve a nonlinear algebraic system at each time step, and we have implemented an optimal nonlinear multigrid method based upon full approximation scheme (FAS). The second half of the presentation illustrates the application of this new software framework to a challenging application, namely a multi-phase-field model of tumour growth [18]. We show some typical simulations for growth of the model tumours, and these results demonstrate second-order convergence in both space and time. We conclude with a discussion of the challenges of obtaining highly scalable parallel performance for a software tool that combines both local mesh adaptivity (requiring efficient dynamic load-balancing) and a multigrid solver (requiring careful implementation of coarse grid operations and inter-grid transfer operations in parallel).

Keywords: parallel, adaptive mesh refinement, finite difference, implicit, multigrid.

1 Introduction

Many problems in computational engineering are based upon the use of complex mathematical models and their numerical approximations. These models often consist of highly nonlinear, time-dependent and coupled PDEs. Accurate, efficient and reliable numerical algorithms (and, frequently, great computational power) are necessary

in order to obtain robust computational solutions. This work is concerned with the novel application of advanced numerical methods to the efficient solution of nonlinear time-dependent systems of PDEs. Specifically, the focus is on parabolic systems. This type of system may be applied to a plethora of different applications, ranging from solidification [3], computational fluid dynamics [10, 11] to tumour growth [18].

The multigrid method is commonly accepted as being one of the fastest numerical methods for solving algebraic equations arising from mesh-based discretizations of PDEs. Brandt in his 1977 paper [5] systematically describes the first multigrid methods, and some of their applications. The subsequent publications e.g. [6, 17] suggest further combinations of multigrid methods with spatial adaptivity and adaptive time-stepping, for applications in which physical effects occur at multiple length and time scales. Often these problems tend to be very stiff, thus requiring fully-implicit (strongly stable) temporal discretization. Alternative existing software packages which offer adaptive multigrid solvers are, for instance, DEAL.II [2] and DUNE [7]. However, to our best knowledge, generally the systems implemented in these packages are preferably discretized using explicit schemes in a parallel setting due to their simplicity of implementation.

The software that we have implemented aims to efficiently solve the algebraic systems arising from each *implicit* time step. We combine FAS multigrid with dynamic adaptivity and adaptive time-stepping in a parallel setting and address the challenge in parallel performance from using this combination of schemes. In Section 2, we introduce our software framework, from its fundamental mesh generator to multigrid and adaptivity. In Section 3, an application of multi-phase-field model of tumour growth is presented, along with our results and novel findings. We conclude this paper in Section 4 with suggestions for possible future work.

2 Software Framework

The software that is used in this work, Campfire [3, 13, 19], is dependent upon an open source software library, which is called PARAMESH [15]. This software library generates structured, cell-centred, Cartesian meshes with the use of a block partitioning strategy, and obtains spatial adaptivity by having multiple layers of mesh refinements. Providing two user-defined, problem-specific tolerances (i.e. one for coarsening and another for refining), the mesh can be dynamically adapted through adaptive mesh refinement (AMR). The mesh hierarchies supported by PARAMESH are quad-tree and oct-tree meshes in two and three dimensions, respectively. We show a typical example of such a 2-D adaptive grid in Figure 1 (a) with a globally user-defined block size of 2×2, and its corresponding quad-tree mesh structure in Figure 1 (b). In this example, instead of the original depth-first Morton ordering 1 that is implemented in PARAMESH, an alternative level-by-level ordering is used in Campfire because the partition of the mesh structure that is induced by this ordering is more fitting to the multigrid solver in parallel. The parallel distribution example shown in Figure 1 (b) is modified from the original PARAMESH implementation, so the workload is balanced

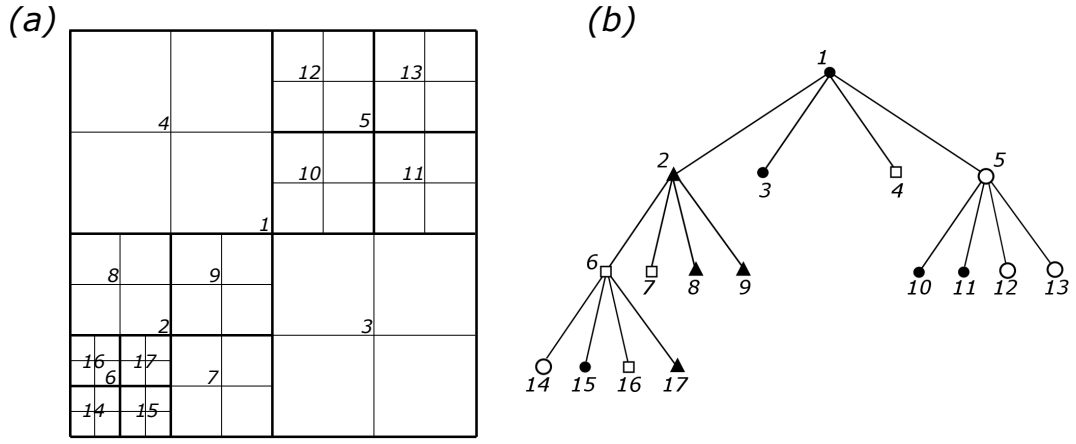

Figure 1: (a) A 2-D adaptive Cartesian mesh: the heavy lines indicate the boundaries of each block, and the lighter lines indicate individual cell-centred grid points. (b) The corresponding quad-tree mesh structure of the mesh in (a); four shapes (i.e. □, ▲, ● and ○) are used to illustrated a possible distribution to four MPI processes in a parallel environment (ensuring that the work is equally distributed within each level of the tree).

on each mesh level, as multigrid operates from one level to another in a sequential order, instead of globally distributing the whole mesh tree.

Each of the mesh blocks is surrounded by a layer of guard cells (not illustrated in Figure 1 for clarity), which may be expanded to multiple layers for schemes with larger stencils. The guard cells at the actual domain boundary contain information which allows the specified boundary conditions to be implemented, and others are used to store values of corresponding grid points on the neighbouring blocks. In PARAMESH, parallelism is achieved through distributing mesh blocks to multiple MPI processes, and using MPI to communicate between individual MPI processes to exchange data held by these guard cells. A possible workload-balanced parallel distribution is shown in Figure 1 (b) across four MPI processes. If AMR is carried out dynamically, then dynamic load-balancing is applied in order to maintain the parallel performance.

A nonlinear multigrid with full approximation scheme (FAS) [5, 6, 17] is built in Campfire for the algebraic systems arising from every time step. This multigrid method exploits iterative methods which are known to have a smoothing property, and smooth the nonlinear algebraic systems on each mesh level. More specifically, when all Fourier components (up to the resolution of the grid) are present in the initial error, smoothing methods (such as Jacobi and Gauss-Seidel) damp out the highest frequency components of the error in a very small number of iterations. Therefore, by applying a few sweeps of such an iterative method on a fine grid, a large reduction of the high frequency components of the error is achieved. In order to remove the remaining low frequency components of the error on the fine grid, the algebraic system is moved

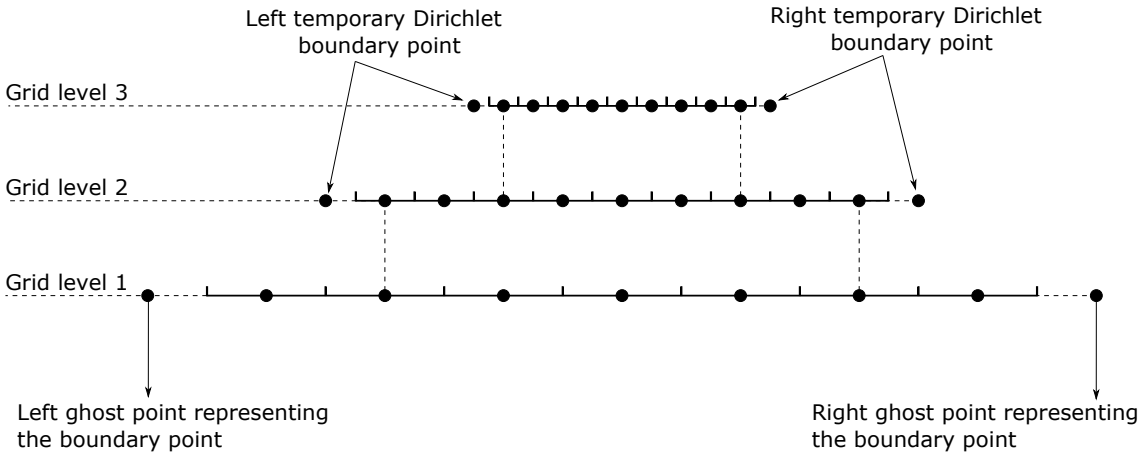

Figure 2: A three-level hierarchy of 1-D adaptive cell-centred grids, with the use of temporary boundary points, the nonlinear FAS multigrid with MLAT can be applied on these grids.

down to a coarser grid in the grid hierarchy. In order to form a problem on the coarser grid that is connected with the problem on the fine grid, a modified right-hand side (RHS) is given to the coarse grid problem, which depends upon the restricted residual and the fine grid solution itself. Since the number of grid points is reduced on such a coarser grid, part of the remaining error has a high frequency on this grid. The iterative method can again quickly remove the high frequency components of the error within a small number of sweeps. This is repeated until the coarsest grid (with the least number of grid points) is reached. A large amount of iterations can be performed to obtain an "exact" solution on the coarsest grid, with a reasonable time cost because of the small amount of work per iteration. This "exact" solution can then be interpolated to the next finer grid and used to improve the solution. Once the correction reaches the finest mesh, it completes a full cycle, and this is termed a V-cycle. A user-defined stopping criterion is required here (typically imposed on the residual) to indicate if the solution has converged with acceptable errors, or another V-cycle is required.

The FAS multigrid method is designed for uniform grids. With the given spatial adaptivity from PARAMESH (e.g. Figure 1 (a)), one may combine the FAS multigrid with a multi-level adaptive technique (MLAT) in a straightforward manner. That is, by fixing the guard cells as temporary Dirichlet boundary points, the problem can be smoothed on local grids, and only the coarsest grid is required to capture the whole domain. For the purpose of demonstration, the use of temporary boundary points is illustrated on 1-D adaptive cell-centred grids in Figure 2.

For the grid transfer operators, used to move between grid levels, a cell-averaging restriction and a bilinear interpolation are employed in Campfire. We illustrate these two operators on a simple 2-D cell-centred grid in Figure 3. A 2-D version of the

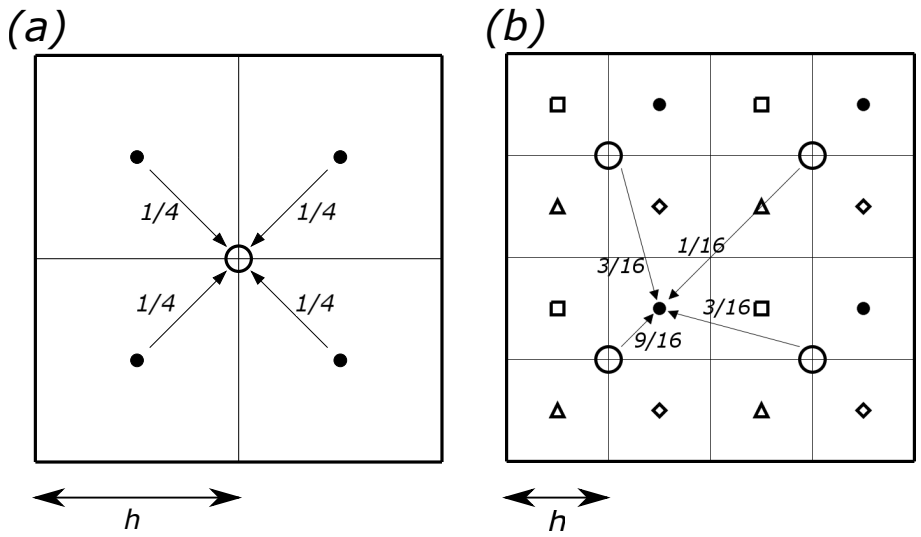

Figure 3: (a) The cell-averaging restriction operator in Equation (1): this process reduces a group of four points (marked as •) on the fine mesh level to one point (marked as ○) on the coarse mesh level. (b) The bilinear interpolation operator in Equation (2): arrows indicate an example of this process from points (marked as ○) on the coarse mesh level to a point (marked as •) on the fine mesh level.

restriction can be written as

$$
\begin{aligned}
I_h^{2h} u_h(x,y) = \\
&\frac{1}{4}\Bigg[u_h\left(x-\frac{h}{2},y-\frac{h}{2}\right)+u_h\left(x-\frac{h}{2},y+\frac{h}{2}\right) \\
&\quad u_h\left(x+\frac{h}{2},y-\frac{h}{2}\right)+u_h\left(x+\frac{h}{2},y+\frac{h}{2}\right)\Bigg],
\end{aligned}
\tag{1}
$$

and a 2-D version of the interpolation is given as

$$
I_{2h}^{h} u_{2h}(x,y) = \begin{cases}
\frac{1}{16}\Big[9u_{2h}\left(x-\frac{h}{2},y-\frac{h}{2}\right)+3u_{2h}\left(x-\frac{h}{2},y+\frac{3h}{2}\right) & \\
+3u_{2h}\left(x+\frac{3h}{2},y-\frac{h}{2}\right)+u_{2h}\left(x+\frac{3h}{2},y+\frac{3h}{2}\right)\Big] & \text{for } \bullet; \\
\frac{1}{16}\Big[3u_{2h}\left(x-\frac{3h}{2},y-\frac{h}{2}\right)+u_{2h}\left(x-\frac{3h}{2},y+\frac{3h}{2}\right) & \\
+9u_{2h}\left(x+\frac{h}{2},y-\frac{h}{2}\right)+3u_{2h}\left(x+\frac{h}{2},y+\frac{3h}{2}\right)\Big] & \text{for } \square; \\
\frac{1}{16}\Big[3u_{2h}\left(x-\frac{h}{2},y-\frac{3h}{2}\right)+9u_{2h}\left(x-\frac{h}{2},y+\frac{h}{2}\right) & \\
+u_{2h}\left(x+\frac{3h}{2},y-\frac{3h}{2}\right)+3u_{2h}\left(x+\frac{3h}{2},y+\frac{h}{2}\right)\Big] & \text{for } \diamond; \\
\frac{1}{16}\Big[u_{2h}\left(x-\frac{3h}{2},y-\frac{3h}{2}\right)+3u_{2h}\left(x-\frac{3h}{2},y+\frac{h}{2}\right) & \\
+3u_{2h}\left(x+\frac{h}{2},y-\frac{3h}{2}\right)+9u_{2h}\left(x+\frac{h}{2},y+\frac{h}{2}\right)\Big] & \text{for } \triangle,
\end{cases}
\tag{2}
$$

where the array u stores values at the grid points, (x, y) are the Cartesian coordinates, $h, 2h$ are the grid spacing on fine and coarse grids respectively and the geometric symbols are indicated in Figure 3 (b). The 3-D version of these two operators (i.e. 3-D cell averaging and trilinear interpolation) are straightforward.

Our software tool is intended for mathematical models that consist of a system of multiple coupled equations. From our experience, it is better for a smoother to update the whole system (all variables), at each visited grid point, simultaneously. Hence, we utilise a point-wise, nonlinear block Jacobi method of the local relaxation-type, in which Newton's method is used to solve the small nonlinear system corresponding to the unknowns on each grid point [16]. In order to demonstrate the use of this nonlinear block Jacobi method, consider a finite difference discretization of a system of elliptic nonlinear PDEs: $\mathcal{F}(u) = 0$, where u is a vector contains all unknowns. Let $u_{i,k}$ be the approximate solution on grid point i for unknown variable k, where we assume $\mathcal{K}$ unknowns at each grid point i. The system $\mathcal{F}(u) = 0$ is made up of $N \times \mathcal{K}$ coupled nonlinear algebraic equations,

$$\mathcal{F}_{i,k}(u) = 0, \tag{3}$$

where $i = 1, \ldots, N$ and $k = 1, \ldots, \mathcal{K}$ (to clarify the notation $u_{i,k}$ is the k^{th} component of $u_i \in \mathbb{R}^{\mathcal{K}}$ and $\mathcal{F}_{i,k}$ is the k^{th} component of $\mathcal{F}_i \in \mathbb{R}^{\mathcal{K}}$). On one grid point i, all $\mathcal{K}$ variables may be updated simultaneously as

$$u_i^{\ell+1} = u_i^{\ell} - C_i^{-1}\mathcal{F}_i(u^{\ell}), \tag{4}$$

where ℓ is the number of iteration and C_i^{-1} is the inverse of the $\mathcal{K} \times \mathcal{K}$ Jacobian matrix C_i, which is given as

$$C_i = \begin{pmatrix} \frac{\partial \mathcal{F}_{i,1}}{\partial u_{i,1}} & \frac{\partial \mathcal{F}_{i,1}}{\partial u_{i,2}} & \cdots & \frac{\partial \mathcal{F}_{i,1}}{\partial u_{i,\mathcal{K}}} \\ \frac{\partial \mathcal{F}_{i,2}}{\partial u_{i,1}} & \frac{\partial \mathcal{F}_{i,2}}{\partial u_{i,2}} & \cdots & \frac{\partial \mathcal{F}_{i,2}}{\partial u_{i,\mathcal{K}}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{F}_{i,\mathcal{K}}}{\partial u_{i,1}} & \frac{\partial \mathcal{F}_{i,\mathcal{K}}}{\partial u_{i,2}} & \cdots & \frac{\partial \mathcal{F}_{i,\mathcal{K}}}{\partial u_{i,\mathcal{K}}} \end{pmatrix}. \tag{5}$$

The Gauss-Seidel form of this method is straightforward as it uses the most up-to-date values of u on the RHS of Equation (4), rather than only using u^{ℓ}. Although for the purpose of demonstration, an elliptic system is chosen, a discretised, time-dependent parabolic system of PDEs may also be written in the form of Equation (3) at each time step when an implicit temporal discretization scheme is used.

Having described multigrid and AMR, for problems in which physical effects occur at multiple time scales, fully implicit time-stepping schemes are usually appropriate (such problems tend to be stiff), often combined with adaptive step size selection. In Campfire, we use the number of V-cycles within each time step as an indicator to control the selection of δt. Typically, when a time step size δt is relatively small, the multigrid solver converges very quickly, so δt has the potential to be increased. On the contrary, the multigrid solver struggles if a time step size δt is too large, and may even fail to converge. Thus δt has to be reduced, and for the failed attempt, that time

step has to be restarted. δt is kept the same if the solver converges within a reasonable number of V-cycles. Together with the use of adaptive BDF2 [9], we have applied this adaptive time-stepping to the long-wave models of thin film flows [10, 11] (see [19] for detailed discussions) and phase-field models of binary alloy solidification [3, 13]. Within our framework, it is also possible and straightforward for one to implement an indicator that is based upon local ordinary differential equation error estimation, such as the one used in [12].

Having provided a brief overview of the software that we have developed, in the following section, we demonstrate our software framework with a multi-phase-field model of tumour growth [18].

3 An Application Base Upon Multi-Phase-Field Model of Tumour Growth

Over the last few decades, the understanding of tumourigenesis (the birth and growth of tumours) has developed dramatically and the contribution of mathematical tumour modelling cannot be neglected. The biological experiments and clinical observations have been complemented by the mathematical models that have been developed. Even simple tumour models and simulations can help to support or deny the hypotheses made from observations. Therefore, the two, mathematical modelling and experimental work, when interwoven together, expand our knowledge on tumour growth and eventually contribute to therapies. The review papers [1, 8, 14] describe a number of examples in detail to illustrate how theory can drive experiments and vice versa.

On the other hand, notwithstanding the advances in scientific and medical research, the process of tumourigenesis and further growth still remains elusive. To study the complex procedures of tumour growth itself and its interactions with the host, a continuum modelling technique which consists of a set of PDEs can be used to model the morphology of tumours. In our work presented here, a multi-phase-field model of tumour growth is considered, from Wise et al. [18]

There are in total four independent phase-field variables in this model, namely ϕ_W, ϕ_H, ϕ_V and ϕ_D which represent volume fractions of extracellular fluid, healthy cells, viable tumour cells and dead tumour cell, respectively. In addition, there are three assumptions amongst these volume fractions. Firstly, it is assumed the extracellular fluid volume fraction is everywhere constant, $\phi_W(x, y, z, t) = \phi_{W,0} =$ constant. After this assumption, the tumour model only consists of multiple solid cell fractions. Secondly, cells are assumed to be close-packed, and this leads to the sum of the healthy cell volume fraction ϕ_H, the viable tumour cell volume fraction ϕ_V and the dead tumour cell volume fraction ϕ_D to be equal to 1 (i.e. $\phi_H + \phi_V + \phi_D = 1$). Furthermore, the range of values of these phase-field variables is from 0 to 1. Thirdly, it is further assumed that inside the tumour there are only two types of cells: viable and dead. This indicates the total tumour cell volume fraction ϕ_T is the sum of ϕ_V and ϕ_D (i.e. $\phi_T = \phi_V + \phi_D$). Based upon these three assumptions, there are only two phase-field

variables that are required to be solved, and they are ϕ_T and ϕ_D. Once these two variables are obtained, other variables may be derived from the assumptions made.

The component ϕ_T is assumed to obey the following Cahn-Hilliard-type advection-reaction-diffusion equations:

$$\frac{\partial \phi_T}{\partial t} = M\nabla \cdot (\phi_T \nabla \mu) + S_T - \nabla \cdot (\underline{u}_S \phi_T), \tag{6}$$

$$\mu = f'(\phi_T) - \epsilon^2 \nabla^2 \phi_T, \tag{7}$$

where $M > 0$ is the mobility constant, $f(\phi_T) = \phi_T^2(1-\phi_T)^2/4$ is the quartic double-well potential, $\underline{u}_S$ is the tissue velocity (and is substituted out of the equation later), and $\epsilon > 0$ is an interface thickness parameter between healthy and tumour tissue. S_T is the net source of tumour cells which depends on ϕ_T, ϕ_V and ϕ_D, and it is given as

$$S_T = nG(\phi_T)\phi_V - \lambda_L \phi_D, \tag{8}$$

where n is the concentration of nutrient, which is specified later, $\phi_V = \phi_T - \phi_D$, and $\lambda_L \geq 0$ is the rate of tumour cell proliferation. $G(\phi_T)$ is a continuous cut-off function and it is defined as

$$G(\phi_T) = \begin{cases} 1 & \text{if } \frac{3\epsilon}{2} \leq \phi_T \\ \frac{\phi_T}{\epsilon} - \frac{1}{2} & \text{if } \frac{\epsilon}{2} \leq \phi_T < \frac{3\epsilon}{2} \\ 0 & \text{if } \phi_T < \frac{\epsilon}{2}. \end{cases} \tag{9}$$

A similar dynamical equation for predicting the volume fraction of dead tumour cells ϕ_D is used:

$$\frac{\partial \phi_D}{\partial t} = M\nabla \cdot (\phi_D \nabla \mu) + S_D - \nabla \cdot (\underline{u}_S \phi_D), \tag{10}$$

where S_D is the net source of dead tumour cells, which depends on ϕ_V and ϕ_D. This source term is defined as

$$S_D = \left(\lambda_A + \lambda_N \mathcal{H}\left(n_N - n\right)\right)\phi_V - \lambda_L \phi_D, \tag{11}$$

where λ_A is the death rate of tumour cells from apoptosis, λ_N is the death rate of tumour cells from necrosis, n_N is the necrotic limit (when nutrient is below this threshold, dead tumour cells occur), and $\mathcal{H}$ is a Heaviside function. This Heaviside function is discontinuous and thus prevents us from obtaining a higher order convergence rate, so we instead use the smoother approximation given by

$$\mathcal{H}(n_N - n) = \begin{cases} 1 & \text{if } n_N - n \geq \epsilon^s \\ -\frac{1}{4(\epsilon^s)^3}\left(n_N - n\right)^3 + \frac{3}{4\epsilon^s}\left(n_N - n\right) + \frac{1}{2} & \text{if } -\epsilon^s \leq n_N - n \leq \epsilon^s \\ 0 & \text{if } n_N - n < -\epsilon^s, \end{cases} \tag{12}$$

where ϵ^s controls the steepness of the smooth transition between 0 and 1.

The tissue velocity $\underline{u}_S$ is assumed to obey Darcy's law, and is defined as

$$\underline{u}_S = -\kappa(\phi_T, \phi_D)(\nabla p - \frac{\gamma}{\epsilon}\mu\nabla\phi_T), \tag{13}$$

where $\kappa > 0$ is the tissue motility function and $\gamma \geq 0$ is a measure of the excess adhesion. An additional assumption made by Wise et al. [18] is that there is no proliferation or death of the host tissue, thus the velocity is constrained to satisfy

$$\nabla \cdot \underline{u}_S = S_T. \tag{14}$$

Instead of solving for the tissue velocity, Equations (13) and (14) are combined together, and a resulting Poisson-like equation for the cell pressure p can be constructed:

$$-\nabla \cdot (\kappa(\phi_T, \phi_D)\nabla p) = S_T - \nabla \cdot (\kappa(\phi_T, \phi_D)\frac{\gamma}{\epsilon}\mu\nabla\phi_T). \tag{15}$$

A quasi-steady equation is given for the nutrient concentration through diffusion:

$$0 = \nabla \cdot (D(\phi_T)\nabla n) + T_c(\phi_T, n) - n(\phi_T - \phi_D), \tag{16}$$

where $D(\phi_T) = D_H(1-Q(\phi_T)) + Q(\phi_T)$ is the diffusion coefficient, D_H is the nutrient diffusivity in the healthy tissue, $Q(\phi_T)$ is an interpolation function, and $T_c(\phi_T, n) = (v_P^H(1 - Q(\phi_T)) + v_P^T Q(\phi_T))(n_C - n)$ is the nutrient capillary source term. Furthermore, $v_P^H \geq 0$ and $v_P^T \geq 0$ are constants specifying the degree of pre-existing uniform vascularization, $n_C \geq 0$ is the nutrient level in capillaries and the interpolation function, $Q(\phi_T)$, is defined as

$$Q(\phi_T) = \begin{cases} 1 & \text{if } 1 \leq \phi_T \\ 3\phi_T^2 - 2\phi_T^3 & \text{if } 0 < \phi_T < 1 \\ 0 & \text{if } \phi_T \leq 0. \end{cases} \tag{17}$$

To sum up, this multi-phase-field model of tumour growth consists of a coupled system of five equations, and they are Equations (6), (7), (10), (15) and (16). There are five dependent variables in total in this system: two phase-field variables, ϕ_T and ϕ_D; and three supplementary variables, μ, p and n. These PDEs are valid throughout a domain Ω, there are no internal boundary conditions for the solid tumour, the necrotic core or other variables. Therefore, only one set of outer boundary conditions is required and this set is the following mixture of Neumann and Dirichlet boundary conditions:

$$\mu = p = 0, \quad n = 1, \quad \frac{\partial\phi_T}{\partial\nu} = \frac{\partial\phi_D}{\partial\nu} = 0 \ \text{ on } \ \partial\Omega, \tag{18}$$

where ν denotes the outward-pointing normal to the boundary $\partial\Omega$.

In order to obtain the desired second order convergence rate, we modified this model with additional penalty terms which are associated with the phase-field Equations (6) and (10). This slightly relaxes the constraint imposed to the range of values of the phase-field variables (i.e. $\phi \in [0, 1]$). The penalty terms added to Equation (6) are

$$\frac{1}{\delta}\min(\phi_T, 0) \quad \text{and} \quad \frac{1}{\delta}\max(\phi_T - 1, 0). \tag{19}$$

These terms have no impact when $0 \leq \phi_{T\,i,j,k} \leq 1$, but create a large correction to the system whenever ϕ_T tries to take a value outside of this interval. The larger the choice of the penalty parameter δ the larger this correction becomes, forcing the values of ϕ_T to be close to this range but at the expense of adding to the nonlinearity of the resulting system. The default value of δ used in this work is 10^{-4}. Similar penalty terms which take ϕ_D into account are also added to Equation (10).

In order to define the initial conditions for the 2-D simulations, firstly the 2-D domain Ω is defined which has Cartesian coordinates $(x, y) \in \Omega = [0, 40] \times [0, 40]$. The initial condition for ϕ_T can be defined as

$$\begin{aligned} \phi_T(x, y) =& 1 \quad \text{if} \quad \frac{(x-20)^2}{1.1} + (y-20)^2 \leq 2^2, \\ =& 0 \quad \text{otherwise.} \end{aligned} \tag{20}$$

This initial condition is discontinuous, so we employed a simple Jacobi iteration to smooth the initial conditions, both to allow second order accuracy to be seen and to avoid unnecessarily restrictive time steps at the start of the simulation. A 2-D version of such iteration is

$$\phi_{T\,i,j}^{l+1,t=0} = \frac{1}{4}\left(\phi_{T\,i+1,j}^{l,t=0} + \phi_{T\,i-1,j}^{l,t=0} + \phi_{T\,i,j+1}^{l,t=0} + \phi_{T\,i,j-1}^{l,t=0}\right). \tag{21}$$

In addition, $\phi_D(t = 0) = 0$ is assumed so that initially no dead tumour cells have occurred. $\mu(t = 0)$ is straightforward to calculate since μ is a function of ϕ_T as shown in Equation (7). The initial conditions for the pressure p and nutrient n require the application of a solver. Due to the increased computational cost in 3-D, an additional multigrid solver is implemented to solve first for the steady state solution of $n(t = 0)$ (since Equation (16) n is not dependent upon p), then $p(t = 0)$. A stopping criterion is used which is dependent upon the infinity norm of the residuals of n and p, and it terminates when $||r||_\infty \leq 1 \times 10^{-9}$.

The values of the parameters that are used in this paper for the multi-phase-field model of tumour growth are presented in Table 1.

Parameters	Values	Parameters	Values
M	10.0	ϵ	0.1
λ_L	1.0	λ_A	0.0
λ_N	3.0	γ	0.0
n_N	0.4	D_H	1.0
v_P^H	0.5	v_P^T	0.0
δ	0.0001	ϵ^s	0.2
n_C	1.0		

Table 1: The parameters of the multi-phase-field model of tumour growth. These were used by Wise et al. in [18], except for ϵ^s and δ.

There are two stopping criteria used in our multigrid solver, which depend upon the infinity norm of the residual from all five variables, and at least one of the stopping criteria must be satisfied in order to stop the iteration. The first is an absolute stopping criterion, which determines to stop the solver at the current time step if the infinity norm is smaller than 10^{-6}. The second is a relative stopping criterion, which takes the infinity norm after the first V-cycle in the current time step, and determines to stop if this infinity norm is reduced by a factor of 10^5 by subsequent V-cycles.

In order to demonstrate our dynamic AMR, results from a 2-D simulation are illustrated here. We present the solution of ϕ_T in Figure 4 with a starting time step size of $\delta t^1 = 1 \times 10^{-3}$. By using adaptive time-stepping, we are able to use a larger time step size later on, however, due to the nature of the problem, we found the time step size could not be increased too much. Our results in this figure show similar tumour evolution to [18] (for a detailed discussion on validation, see [19]). Our choice for the AMR strategy is conservative (not just considering the phase-field variables), because the pressure p obeys Darcy's law and is known, by Wise et al., to be difficult to resolve. Our AMR strategy takes into account the second derivative of the solutions of variables ϕ_T, ϕ_D, p and n, i.e. $|\nabla^2\phi_T|, |\nabla^2\phi_D|, |\nabla^2 p|$ and $|\nabla^2 n|$. Within each mesh block, on every grid point, (i, j), the adaptive assessment is computed via:

$$\text{adaptive assessment}_{i,j} = |u^{(1-4)}_{i+1,j} + u^{(1-4)}_{i-1,j} + u^{(1-4)}_{i,j+1} + u^{(1-4)}_{i,j-1} - 4u^{(1-4)}_{i,j}|. \tag{22}$$

Then the maximum value of adaptive assessment, $u^{\max}$ is selected from these four variables $u^{(1-4)}$ to represent this mesh block and compared against the user-defined refinement/coarsening criteria. The specific choices for these criteria must be obtained and evaluated from practice. In this case we define the refining criterion to be 0.01 and the coarsening criterion to be 0.001. Therefore, if adaptive assessment is greater than 0.01 (i.e. $dxdy|\nabla^2 u^{\max}| > 0.01$), this mesh block is marked for refining, or if it is less than 0.001 (i.e. $dxdy|\nabla^2 u^{max}| < 0.001$), then this mesh block is marked for coarsening. Due to the profile of p, the highest level of mesh refinement at $t = 0$ covers a much larger area than the initial seed of ϕ_T.

The adaptive meshes and the dynamic AMR are illustrated in Figure 5. The red color represent the highest mesh refinement level: 2048^2; the pink color is level 1024^2; the light blue shows 512^2 mesh level; finally, the dark blue, which only appears in the top two figures, indicates 256^2 mesh level.

Within a typical time step in the 2-D simulation, we illustrate our optimal multigrid convergence rate with five different grid hierarchies in Figure 6. The infinity norm of residuals from four variables (i.e. ϕ_T, μ, ϕ_D and n) is used to demonstrate the multigrid convergence rate. These results suggest the reduction in the residuals (at least for these four variables) is independent of grid sizes. The difficulty of resolving the pressure p has previously been identified, and a detailed discussion is presented in [19] which supports the observations of Wise et al. [18]

The spatial and the temporal discretization schemes that are used in this work are both second-order accurate. Therefore, we expect the overall convergence rate to be second order. This means by halving the time step size and doubling the number of

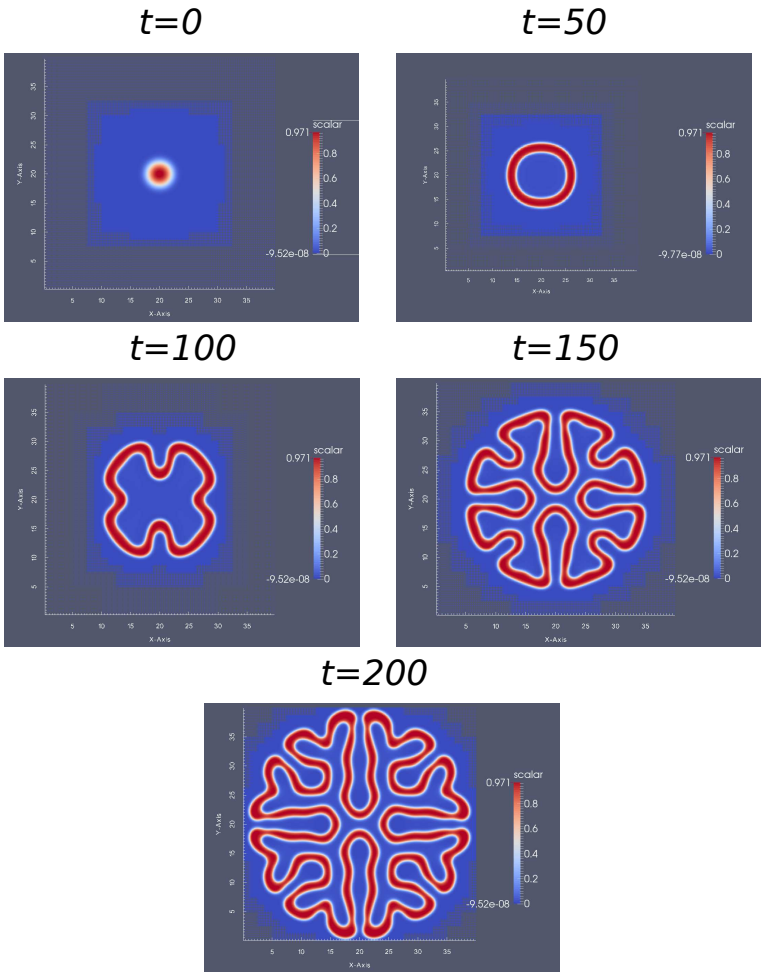

Figure 4: 2-D simulation, showing the evolution of ϕ_T. These results are generated from a grid hierarchy which has 8^2 as the coarsest grid and, if refined everywhere, 2048^2 as the finest grid.

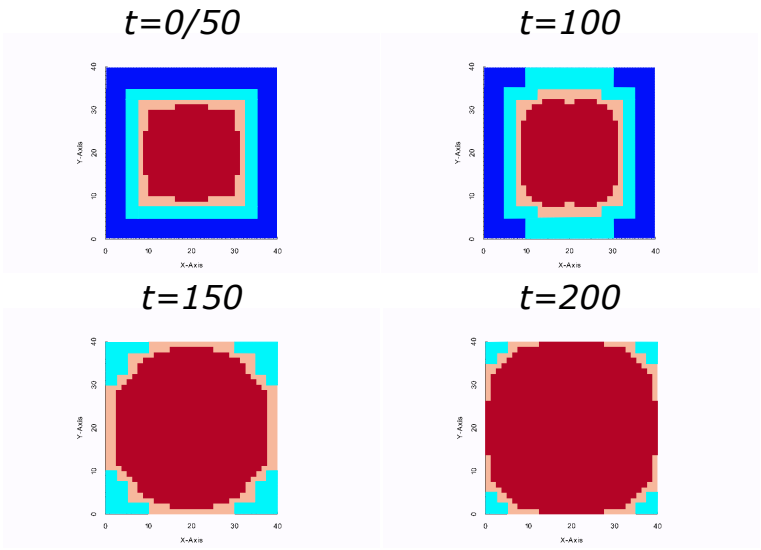

Figure 5: The adaptive meshes from the 2-D simulation shown in Figure 4. Since the mesh at $t = 0$ is relatively similar to the one at $t = 50$, these two are illustrated together here.

grid points in each direction, the error should reduce by a factor of four. In order to illustrate this, we conduct our convergence tests based upon solution restriction. That is, let's consider three example grid hierarchies using 2-D grids. They are $8^2 - 16^2$, $8^2 - 16^2 - 32^2$ and $8^2 - 16^2 - 32^2 - 64^2$. Each grid is associated with a δt: δt^{16^2}, $\delta t^{32^2} = \frac{\delta t^{16^2}}{2}$ and $\delta t^{64^2} = \frac{\delta t^{32^2}}{2}$, respectively. The solutions are obtained by solving the same problem on these three finest grids separately, with their corresponding δt, and with the assumption that the ending time T is exactly the same for all runs. To make a comparison between two solutions we restrict the fine grid solution to the coarse grid by using a restriction operator (e.g. four-point averaging shown in Equation (1)). Thus, the solution which is restricted from grid hierarchy $8^2 - 32^2$ can be compared to the solution from grid hierarchy $8^2 - 16^2$. Similarly, the restricted solution from hierarchy $8^2 - 64^2$ can be compared to the original solution from hierarchy $8^2 - 32^2$.

The infinity norm and the discrete two norm of the differences within the solutions are computed from these comparable solutions on the coarser grid. The infinity norm error measure is defined as the following:

$$||e||_\infty := max(|u_{i,j}^{restricted} - u_{i,j}|), \tag{23}$$

where $u^{restricted}$ is the restricted solution from the finer grid hierarchy, u is the solution from the coarser grid hierarchy, $i, j = 1, \ldots, N$ and N is the number of internal grid points in each axis direction on the finest grid of the coarser grid hierarchy. The discrete two norm error measure is given as

$$||e||_2 := \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(u_{i,j}^{restricted} - u_{i,j})^2}{N \times N}}. \tag{24}$$

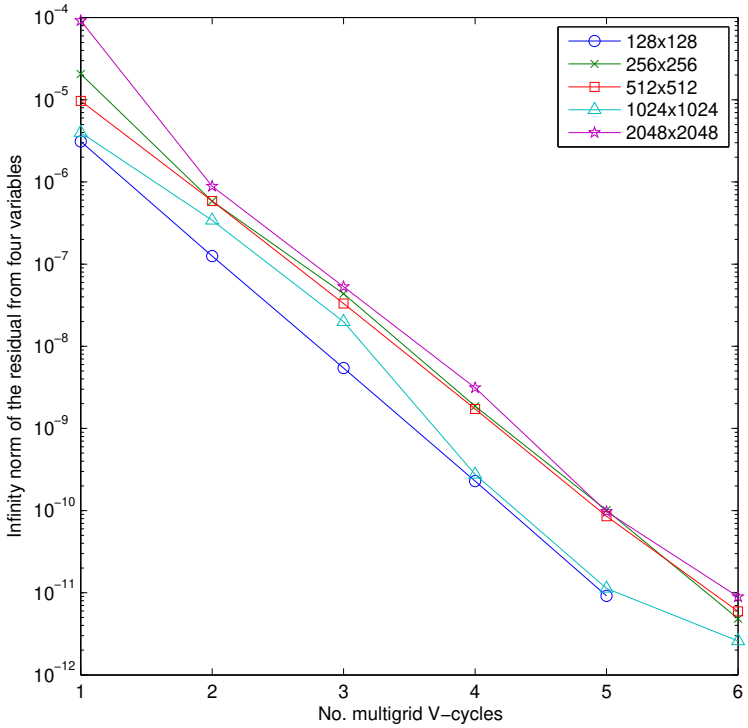

Figure 6: The optimal multigrid convergence rate within a typical time step. There are five different grid hierarchies and the same coarsest grid: 16^2 is used, but the number of grid points on the finest grid quadruples as the number of grids grows each time. For example, 1024×1024 indicates there are 7 grids from 16^2 to 1024^2. The infinity norm of residual is $\max(r(\phi_T), r(\mu), r(\phi_D), r(n))$.

When adaptive grids are used, we consider all grid points from the finest refinement level, as well as the next coarser one. For the tumour model, the infinity norm and the two norm are computed separately for all five variables using 5 different grid hierarchies. The finest grid used, if refined everywhere, has a grid resolution equivalent to 2048^2. The results for convergence tests are presented in Table 2. The evidence for having obtained second order convergence is compelling.

For 3-D simulations, the imposed initial condition of ϕ_T are defined by three ellipsoids as

$$\begin{aligned}\phi_T(x,y,z) = 1 \quad & \text{if} \quad \frac{(x-19)^2}{1.1} + (y-19)^2 + (z-19)^2 \leq 2^2, \\ & \text{or} \quad (x-20)^2 + \frac{(y-20)^2}{1.1} + (z-20)^2 \leq 2^2, \\ & \text{or} \quad (x-21)^2 + (y-19)^2 + \frac{(z-19)^2}{1.1} \leq 2^2, \\ = 0 \quad & \text{otherwise.}\end{aligned} \tag{25}$$

This model of tumour growth is solved in 3-D with the parameters stated in Table 1 and the 3-D initial condition in Equation (25) in a 3-D domain Ω which has the Cartesian coordinates $(x, y, z) \in \Omega = [0, 40] \times [0, 40] \times [0, 40]$. The finest grid resolution used, if refined everywhere, is 256^3. The solutions of ϕ_T at all grid points which have values in range of 0.5 to 1.0 are illustrated in Figures 7 and 8.

With the capability to run our solver in a parallel environment, here we present results from parallel efficiency tests up to 64 cores. We run the simulation of tumour growth for 10 time steps start from $t = 150$, and results of time cost are illustrated in Figure 9. The choice of mesh block size is 16^3, and within this simulation, the coarsest grid is 32^3, the finest grid, if refined everywhere, is 256^3. The actual number of leaf nodes is $8,060,928$ from $1,968$ mesh blocks. In this figure, the deterioration from 16 to 64 cores is caused, in part, by the fact that there is not enough workload on the coarsest grid (which only has 8 mesh blocks). Finding a robust solution to this issue is non-trivial, and there are a number of trade-offs that need to be considered. First of all, one may suggest to reduce the block size, in order to have more mesh blocks at the coarsest grid level. However, this results in the use of many more guard cells, thus causing a heavy burden on the memory as well as the parallel communication. Another logical suggestion would be using a finer coarsest grid (i.e. 64^3 in this case), however, the nonlinear multigrid with FAS solves the nonlinear problem on each level. Due to the nonlinearity, to obtain an "exact" solution on a finer coarsest grid may require many more iterations of the coarsest grid and this deteriorates the overall performance of our solver.

4 Conclusion

We have introduced our software framework, Campfire, which is based upon a mesh generator called PARAMESH. Our built-in nonlinear, optimal multigrid solver with

For variable ϕ_T						
Finest levels	δt	Time steps	Infinity norm	Ratio	Two norm	Ratio
5	8×10^{-3}	1250	-	-	-	-
6	4×10^{-3}	2500	0.719×10^{0}	-	4.969×10^{-2}	-
7	2×10^{-3}	5000	6.228×10^{-2}	11.5	4.876×10^{-3}	10.2
8	1×10^{-3}	10000	1.249×10^{-2}	4.99	1.142×10^{-3}	4.27
9	5×10^{-4}	20000	3.054×10^{-3}	4.09	2.806×10^{-4}	4.07
For variable μ						
5	8×10^{-3}	1250	-	-	-	-
6	4×10^{-3}	2500	1.367×10^{-2}	-	1.279×10^{-3}	-
7	2×10^{-3}	5000	1.205×10^{-3}	11.3	1.103×10^{-4}	11.6
8	1×10^{-3}	10000	3.241×10^{-4}	3.72	2.888×10^{-5}	3.82
9	5×10^{-4}	20000	8.226×10^{-5}	3.94	7.275×10^{-6}	3.97
For variable ϕ_D						
5	8×10^{-3}	1250	-	-	-	-
6	4×10^{-3}	2500	0.245×10^{0}	-	1.923×10^{-2}	-
7	2×10^{-3}	5000	1.663×10^{-2}	14.7	1.976×10^{-3}	14.7
8	1×10^{-3}	10000	4.303×10^{-3}	3.86	4.837×10^{-4}	4.08
9	5×10^{-4}	20000	1.076×10^{-3}	4.00	1.206×10^{-4}	4.01
For variable p						
5	8×10^{-3}	1250	-	-	-	-
6	4×10^{-3}	2500	4.918×10^{-2}	-	1.203×10^{-2}	-
7	2×10^{-3}	5000	5.940×10^{-3}	8.28	1.726×10^{-3}	6.97
8	1×10^{-3}	10000	1.469×10^{-3}	4.04	4.487×10^{-4}	3.85
9	5×10^{-4}	20000	3.673×10^{-4}	4.00	1.127×10^{-4}	3.98
For variable n						
5	8×10^{-3}	1250	-	-	-	-
6	4×10^{-3}	2500	0.102×10^{-0}	-	1.012×10^{-2}	-
7	2×10^{-3}	5000	7.385×10^{-3}	13.8	1.003×10^{-3}	10.1
8	1×10^{-3}	10000	1.508×10^{-3}	4.90	2.365×10^{-4}	4.24
9	5×10^{-4}	20000	3.696×10^{-4}	4.08	5.913×10^{-5}	4.00

Table 2: Results show the differences in consecutive solutions measured in the stated norm, followed by the ratio of consecutive differences.

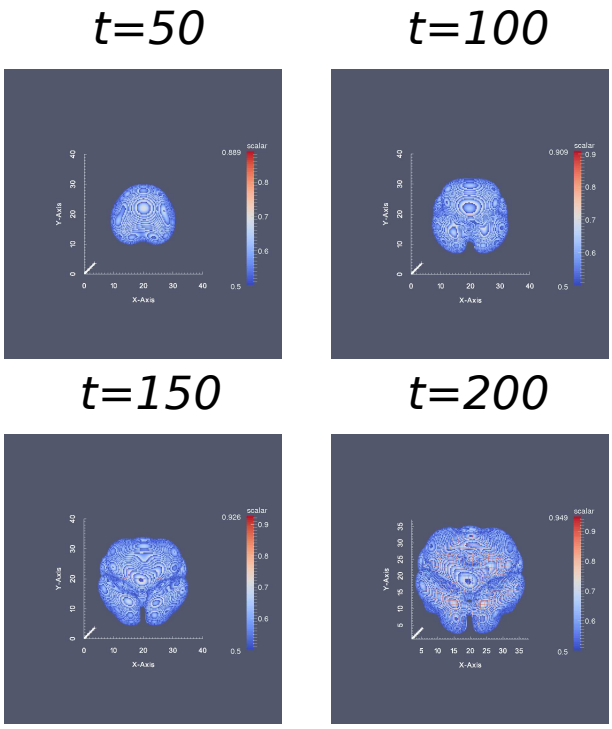

Figure 7: 3-D solutions of variable ϕ_T at $t = 50, 100, 150$ and 200. Images in this figure display the $[0.5 - 1.0]$ solutions of ϕ_T.

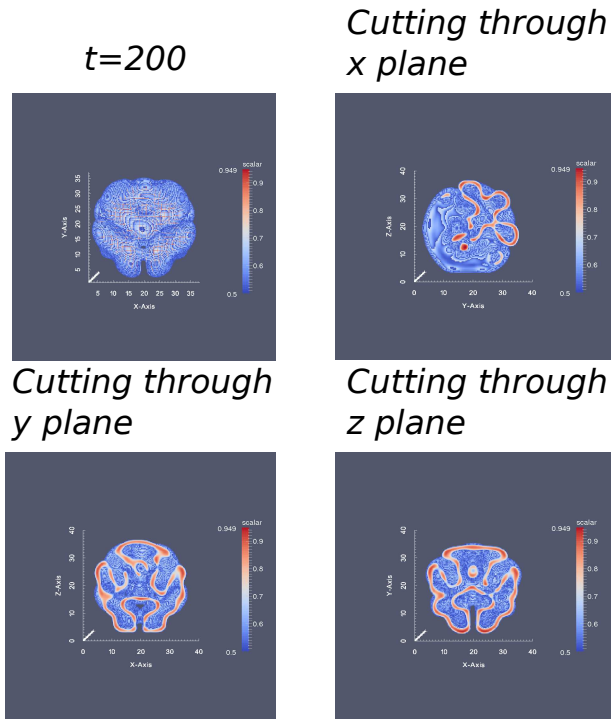

Figure 8: Images of three cross-sections through the middle of each plane for the solution of ϕ_T at all grid points which have values in range of 0.5 to 1.0 at $t = 200$.

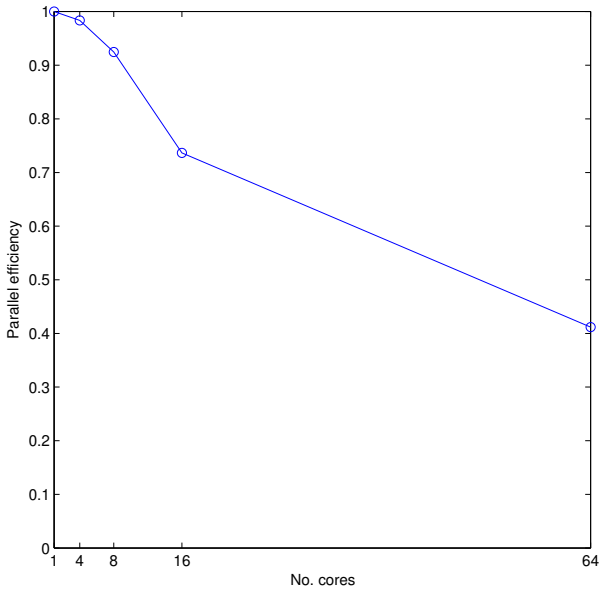

Figure 9: Parallel efficiency tested for 10 time steps with a middle developed tumour up to 64 cores.

FAS can be applied to general nonlinear, time-dependent parabolic systems. In addition, it is further coupled with dynamic AMR, adaptive time-stepping and parallelism through domain decomposition and parallel communication through MPI. We have briefly described the nonlinear multigrid method with FAS, its variation with MLAT and its grid transfer operators. For commonly coupled nonlinear systems, we have proposed a nonlinear block Jacobi method as the multigrid smoother.

This software framework is demonstrated for an application base upon a multi-phase-field model of tumour growth [18]. We have illustrated our dynamic evolving meshes, as well as the optimal multigrid convergence. Through the use of penalty terms and a smoothed Heaviside function, we are able to obtain, for the first time, an overall second order convergence rate (only first order solutions were obtained in [18]). This model is also solved in a computationally demanding 3-D space. We present our solution, as well as results from a parallel efficiency test. Although there are issues with our parallel scaling, these can be diminished as the amount of computational work on the finest grid level is increased (for example, in [3], satisfactory parallel efficiency is observed on up to 1024 processes for a sufficiently large problem - in a different application domain to that considered here). The reasons for the challenges associated with obtaining high parallel efficiency are multiple and have been discussed. The way in which the coarse grid problem is solved is of great importance, for example. Further work could therefore be undertaken to overcome this bottleneck on the coarsest grids, possibly through the use of only some of the cores at the coarsest levels. This technique, namely agglomeration, has been briefly mentioned in [17]. The dynamic load-balancing algorithm itself could also be improved: currently we focus only on the efficiency of the parallel smoothing, but this results in relatively inefficient grid transfer operators (in terms of data movement). Finally, one may seek a change in the multigrid algorithm, for example, using a Newton multigrid approach (see [4, 19] for detail). Due to the global Newton linearization, we may be able to afford a much finer coarsest grid (with its linear problem) and this may improve the parallel scaling.

References

[1] R.P. Araujo, D.L.S. McElwain, "A History of the Study of Solid Tumour Growth: The Contribution of Mathematical Modelling", Bulletin of Mathematical Biology, 66, 1039-1091, 2004.

[2] W. Bangerth, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, T.D. Young, "The Deal.II Library, Version 8.1", arXiv preprint at http://arxiv.org.abs.1312.2266v4, 2013.

[3] P. Bollada, C. Goodyer, P. Jimack, A. Mullis, F. Yang, "Thermalsolute Phase Field Three Dimensional Simulation of Binary Alloy Solidification", Journal of Computational Physics, Submitted August 2014.

[4] K.J. Brabazon, M.E. Hubbard, P.K. Jimack, "Nonlinear Multigrid Methods for Second Order Differential Operators with Nonlinear Diffusion Coefficient", Computers and Mathematics with Applications, Accepted 2014.

[5] A. Brandt, “Multi-Level Adaptive Solutions to Boundary-Value Problems”, Mathematics of Computation, 31, 333-390, 1977.

[6] W.L. Briggs, V.E. Henson, S.F. McCormick, “A Multigrid Tutorial”, Society for Industrial and Applied Mathematics, 2000.

[7] A. Burri, A. Dedner, R. Klfkorn, M. Ohlberger, “An Efficient Implementation of An Adaptive and Parallel Grid in DUNE”, in proceedings of the 2nd Russian-German Advanced Research Workshop on Computational Science and High Performance Computing, 2005.

[8] H.M. Byrne, T. Alarcon, M.R. Owen, S.D. Webb, P.K. Maini, “Modelling Aspects of Cancer Dynamics: A Review”, Philosophical Transactions of the Royal Society A, 364, 1563-1578, 2006.

[9] E. Emmerich, “Stability and Error of the Variable Two-Step BDF for Semilinear Parabolic Problems”, Journal of Applied Mathematics and Computing, 19, 33-55, 2005.

[10] P.H. Gaskell, P.K. Jimack, M.Sellier, H.M. Thompson, “Efficient and Accurate Time Adaptive Multigrid Simulations of Droplet Spreading”, International Journal for Numerical Methods in Fluids, 45, 1161-1186, 2004.

[11] P.H. Gaskell, P.K. Jimack, M. Sellier, H.M. Thompson, M.C.T. Wilson, “Gravity-Driven Flow of Continuous Thin Liquid Films on Non-Porous Substrates with Topography”, Journal of Fluid Mechanics, 509, 253-280, 2004.

[12] C.E. Goodyer, M. Berzins, “Adaptive Timestepping for Elastohydrodynamic Lubrication Solvers”, SIAM Journal on Scientific Computing, 28, 626-650, 2006.

[13] C.E. Goodyer, P.K. Jimack, A.M. Mullis, H.B. Dong, Y. Xie, “On the Fully Implicit Solution of a Phase-Field Model for Binary Alloy Solidification in Three Dimensions”, Advances in Applied Mathematics and Mechanics, 4, 665-684, 2012.

[14] J.S. Lowengrub, H.B. Frieboes, F. Jin, Y-L. Chuang, X. Li, P. Macklin, S.M. Wise, V. Cristini, “Nonlinear Modelling of Cancer: Bridging the Gap Between Cells and Tumours”, Nonlinearity, 23, R1-R91, 2010.

[15] K. Olson, P. MacNeice, “An Overview of the PARAMESH AMR Software and Some of Its Applications”, Adaptive Mesh Refinement-Theory and Applications, Lecture Notes in Computational Science and Engineering 41, eds. T. Plewa, T. Linde, G. Weirs (Springer), 2005.

[16] J.M. Ortega, W.C. Rheinbolt, “Iterative Solution of Non-linear Equations in Several Variables”, Academic Press, 1970.

[17] U. Trottenberg, C.Oosterlee, A.Schuller, “Multigrid”, Academic Press, 2001.

[18] S.M. Wise, J.S. Lowengrub, V. Cristini, “An Adaptive Multigrid Algorithm for Simulating Solid Tumor Growth Using Mixture Models”, Mathematical and Computer Modelling, 53, 1-20, 2011.

[19] F.W. Yang, “Multigrid Solution Methods for Nonlinear, Time-Dependent Systems”, PhD Thesis, Thesis Collection, School of Computing, University of Leeds, 2014.