EXPBrain: Exponential Integrators for Glioblastoma Brain Tumor Simulations

Magdalena Pabisz[1], Dominika Ciupek[2][0000−0002−1411−3060], Askold Vilkha[1][0000−0001−9272−9082], and Maciej Paszyński[1][0000−0001−7766−6052]

[1] AGH University of Krakow, Poland
maciej.paszynski@agh.edu.pl***
[2] Sano Centre for Computational Medicine, Kraków, Poland †

Abstract. In this paper, we discuss MATLAB implementation of the exponential integrators method employed for simulations of brain tumor progression. As the input data, we utilize the publicly available T1-weighted magnetic resonance imaging dataset ds003826, representing healthy individuals. The data is originally stored using NIfTI format. We randomly select one anonymized individual from the considered dataset. We normalize the brain scan data using min-max normalization to a range of 0 to 255. In the data from ds003826, the voxel resolution is not isotropic in all directions, so we interpolate the data from dimensions $176 \times 248 \times 256$ to $194 \times 248 \times 256$ in order to have proper proportions of the human brain. We set the data as a sequence of 256 PNG files with the resolution of 194×248. Having the MRI scan data, we run the exponential integrators method simulating the glioblastoma tumor growth using the Fisher-Kolmogorov diffusion-reaction model with logistic growth. We assume the initial tumor location and run the simulation predicting the tumor growth two years forward. For the spatial discretization, we employ the finite difference method, and for the temporal discretization, we use the ultra-fast exponential integrators method. Our simulator generates results suitable for visualization using the ParaView tool.

Keywords: T1-weighted magnetic resonance, MATLAB code, brain tumor simulations, exponential integrators, Fisher-Kolmogorov diffusion-reaction model with logistic growth, ParaView visualization

1 Introduction

In this paper, we present the EXPBrain code that performs simulations of the glioblastoma brain tumor on the patient's MRI scan data using the exponential integrators method [1]. Glioblastoma is a malignant brain tumor with a high mortality rate [2]. This tumor is highly aggressive, and it generates a microvascular proliferation that is not visible on MRI scans [3]. Thus, computer-based

*** home.agh.edu.pl/paszynsk
† https://sano.science/

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

simulations of the evolution of the brain tumor are essential in planning treatment and surgery. The state-of-the-art three-dimensional finite element or finite difference simulators are computationally intense [4–7], and the Physics Informed Neural Networks simulators can deal well with two-dimensional simulations only [8]. The exponential integrators considered in our code method allow for high-speed and accurate simulations of time-dependent problems. We employ this method to simulate the growth of a three-dimensional glioblastoma brain tumor. Our exponential integrators method is based on the Fisher-Kolmogorov diffusion-reaction equation with logistic growth [9].

The EXPBrain simulator presented here is a useful tool for modeling the future growth of the brain tumor. Here we discuss some of the possible use cases. One of the use cases of this tool is to produce compelling visualizations of the tumor growth that can help patients understand the dangers of not undergoing professional medical treatment. Another use case is to assist medical professionals in determining the possible directions of intense tumor growth. While it should not be used as a determining factor in surgical decisions, it can highlight the areas where the tumor is likely to grow more aggressively, thus indicating the areas that should be studied more closely. In the future, after some modifications of the model, it can be used to incorporate treatment effects and improve prediction accuracy.

In our research, we utilized a publicly available T1-weighted magnetic resonance imaging dataset, named *ds003826* [10], which consists of the collected data from 136 healthy individuals. Their ages vary from 18 to 35 years. All the subjects were scanned using a 3T scanner called Magnetom Skyra provided by Siemens [11]. It uses a 20-channel or 64-channel head/neck coil. An MPRAGE sequence was employed to obtain the T1-weighted images, using 176 sagittal slices with $1 \times 1 \times 1.1$ millimeter cube [mm^3] voxel size, with TR = 2300 millisecond [ms] and TE = 2.98 millisecond [ms]. Data from the dataset were originally saved in NIfTI format. For our analysis, one anonymized subject was randomly selected from the dataset. The data were then normalized using min-max normalization to a range of 0 to 255, which enabled proper saving in PNG format. In the case of our dataset *ds003826*, the voxel resolution was not isotropic in all directions, so it was necessary to interpolate the data from dimensions $176 \times 248 \times 256$ to $194 \times 248 \times 256$ in order to maintain the brain proportions consistent with reality. Finally, the randomly selected individual's data was saved as a set of PNG files, with each file corresponding to one axial slice of the brain.

We use the exponential Euler method, which is of the first order. We also employ the routine presented in [12] for computing the action of the corresponding φ-functions over the vector. The output from the simulation is stored in the ParaView format for visualization. We test our code using the 256 slices with a resolution of 194×248 each, selecting the initial location of the tumor and simulating the tumor growth prediction two years forward. The numerical results show that we can perform 100 iterations over $128 \times 128 \times 128$ computational mesh in less than 5 minutes on a single computing node from Athena computer [13]. Thus, this simulator can be employed "on the fly".

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

The alternative approach to this problem may utilize Physics Informed Neural Networks (PINNs), originally proposed by Karniadakis [14]. For instance, in [8], PINNs instantiated for the brain tumor simulation can predict the future behavior of the glioblastoma tumor evolution within one hour for the patient-specific case. We employ the exponential time integrators [15–17] designed to solve semilinear problems. While there are many brain tumor simulation methods, the originality of our method lies in the novel implementation using the exponential integrators method. As far as the authors know, there is only one work in the literature using the exponential integrators method to simulate the tumor growth [18].

The efficiency comes from the fact that the exponential routines are extremely efficient on operators coming from a semi-discretization in space employing finite differences. Moreover, the exponential time integrators [1, 16, 17] are designed to solve semilinear problems presented in this article. They are suitable for long-time simulations, and they are unconditionally stable. The computational model and numerical methods have already been described in detail in [19]. In this paper, we focus on MATLAB implementation, its performance, and its limitations.

The structure of the paper is the following. In Section 2, we introduce the finite difference spatial discretization for the Fisher-Kolmogorov brain tumor model. In Section 3, we derive the exponential integrators method for temporal discretization. The following Section presents the MATLAB code description, including the software architecture and the installation manual. We summarize the paper in Section 5, presenting exemplary numerical experiments. The conclusions are presented in Section 6. In the Appendix section, we provide some code snippets (Appendix A) and a user guide to visualize the simulation results in ParaView (Appendix B).

2 Finite differences

The simulation is based on the Fisher-Kolmogorov diffusion-reaction equation with logistic growth,

$$\begin{cases} \dfrac{\partial u}{\partial t} = \underbrace{\nabla \cdot (D(\mathbf{x})\nabla u)}_{\text{Tumor cell diffusion}} + \underbrace{\rho u(1-u)}_{\text{Tumor cell proliferation}}, & \text{in } \Omega \times I, \\ \nabla u \cdot n = 0, & \text{on } \partial\Omega \times I, \\ u(\mathbf{x}, 0) = u_0, & \text{on } \Omega \times \{0\}, \end{cases} \tag{1}$$

where $\mathbf{x} = (x, y, z)$, $u(x, y, z; t)$ represents the tumor cell density, $D(x, y, z)$ is the diffusion coefficient estimated for different materials based on the MRI scan data, and ρ is the tumor cells proliferation rate (patient-specific).

We first semi-discretize (1) in space employing finite differences. Namely, we introduce a regular grid with equidistant points in each spatial direction:

$$\{x_{i,j,k} = ((i-1)h, (j-1)h, (k-1)h)\}_{i=1,\ldots,N_x; j=1,\ldots,N_y; k=1,\ldots,N_z}, \tag{2}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

and we represent the values of the tumor cell densities at these points and time moment t as

$$\{u^t_{i,j,k} = u(x_{i,j,k}; t)\}_{i=1,\ldots,N_x; j=1,\ldots,N_y; k=1,\ldots,N_z}. \tag{3}$$

We discretize the equation at time moment t using finite differences as follows:

$$\begin{aligned}\frac{\partial u^t_{i,j,k}}{\partial t} =& \frac{\partial D(x_{i,j,k})}{\partial x_1}\frac{\partial u^t_{i,j,k}}{\partial x_1} + D(x_{i,j,k})\frac{\partial^2 u^t_{i,j,k}}{\partial x_1^2} + \\ & \frac{\partial D(x_{i,j,k})}{\partial x_2}\frac{\partial u^t_{i,j,k}}{\partial x_2} + D(x_{i,j,k})\frac{\partial^2 u^t_{i,j,k}}{\partial x_2^2} \\ & \frac{\partial D(x_{i,j,k})}{\partial x_3}\frac{\partial u^t_{i,j,k}}{\partial x_3} + D(x_{i,j,k})\frac{\partial^2 u^t_{i,j,k}}{\partial x_3^2} + \rho u^t_{i,j,k}(1 - u^t_{i,j,k}),\end{aligned} \tag{4}$$

with

$$\begin{aligned}\frac{\partial u^t_{i,j,k}}{\partial x_1} &= \frac{u^t_{i+1,j,k} - u^t_{i,j,k}}{h}, \\ \frac{\partial u^t_{i,j,k}}{\partial x_2} &= \frac{u^t_{i,j+1,k} - u^t_{i,j,k}}{h}, \\ \frac{\partial u^t_{i,j,k}}{\partial x_3} &= \frac{u^t_{i,j,k+1} - u^t_{i,j,k}}{h}, \\ \frac{\partial^2 u^t_{i,j,k}}{\partial x_1^2} &= \frac{u^t_{i+1,j,k} - 2u^t_{i,j,k} + u^t_{i-1,j,k}}{h^2}, \\ \frac{\partial^2 u^t_{i,j,k}}{\partial x_2^2} &= \frac{u^t_{i,j+1,k} - 2u^t_{i,j,k} + u^t_{i,j-1,k}}{h^2}, \\ \frac{\partial^2 u^t_{i,j,k}}{\partial x_3^2} &= \frac{u^t_{i,j,k+1} - 2u^t_{i,j,k} + u^t_{i,j,k-1}}{h^2},\end{aligned} \tag{5}$$

and we obtain the following system of semilinear Ordinary Differential Equations

$$\begin{cases}\dot{U}(t) = AU(t) + F(U(t)), & \text{in } I, \\ U(0) = U_0,\end{cases} \tag{6}$$

where $U(t) = \{u^t_{i,j,k}\}_{i=1,\ldots,N_x; j=1,\ldots,N_y; k=1,\ldots,N_z}$ is the time-dependent vector of the degrees of freedom in space. To derive the A operator, we simplify the derivation, assuming $\frac{\partial D_{i,j,k}}{\partial x_i} = 0$,

$$\begin{aligned}\frac{\partial u^t_{i,j,k}}{\partial t} =& D(x_{i,j,k})\frac{u^t_{i+1,j,k} - 2u^t_{i,j,k} + u^t_{i-1,j,k}}{h^2} \\ & + D(x_{i,j,k})\frac{u^t_{i,j+1,k} - 2u^t_{i,j,k} + u^t_{i,j-1,k}}{h^2} \\ & + D(x_{i,j,k})\frac{u^t_{i,j,k+1} - 2u^t_{i,j,k} + u^t_{i,j,k-1}}{h^2} + \rho u^t_{i,j,k}(1 - u^t_{i,j,k}),\end{aligned} \tag{7}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

and we group the terms

$$\begin{aligned} h^2 \frac{\partial u^t_{i,j,k}}{\partial t} &= -6D(x_{i,j,k})u^t_{i,j,k} + D(x_{i,j,k})u^t_{i-1,j,k} \\ &+ D(x_{i,j,k})u^t_{i,j-1,k} + D(x_{i,j,k})u^t_{i,j,k-1} \\ &+ D(x_{i,j,k})u^t_{i+1,j,k} + D(x_{i,j,k})u^t_{i,j+1,k} \\ &+ D(x_{i,j,k})u^t_{i,j,k+1} + h^2 \rho u^t_{i,j,k}(1 - u^t_{i,j,k}), \end{aligned} \tag{8}$$

The entries of matrix A are

$$A_{i,j,k;l,m,n} = \begin{cases} -6D(x_{i,j,k}), & (i,j,k) == (l,m,n), \\ D(x_{i-1,j,k}), & (l,m,n) \in \{(i-1,j,k), (i+1,j,k), \\ & \qquad (i,j-1,k), (i-1,j+1,k), \\ & \qquad (i-1,j,k-1), (i-1,j,k+1)\}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

We employ $\{1,...,N_x\} \times \{1,...,N_y\} \times \{1,...,N_z\} \ni (i,j,k) \rightarrow global(i,j,k) = i + (j-1)N_y + (j-1)(k-1)N_yN_z$ as the mapping from the integer coordinates into the global rows / columns numbering. The F operator is given by $F(U(t)) = \rho U(t)(1 - U(t))$.

3 Exponential integrators

For the time discretization, we consider a uniform partition of the time interval as

$$0 = t_0 < t_1 < \ldots < t_{N-1} < t_N = T, \tag{10}$$

we define $I_n = (t_n, t_{n+1})$ and $\tau = t_{n+1} - t_n$, $\forall n = 0, \ldots, N-1$. Let U_n be the numerical approximation of the solution of (6) at t_n, we know that the integral representation of the solution of (6) at t_{n+1}, also known as the *variation-of-constants* formula, reads

$$U_{n+1} = e^{\tau A} U_n + \tau \int_0^1 e^{(1-\theta)\tau A} F(U(t_n + \tau\theta)) d\theta. \tag{11}$$

Different approximations of the nonlinear term in (11) lead to different exponential time integration methods [15]. All these methods are expressed in terms of the so-called φ-*functions* defined as

$$\begin{cases} \varphi_0(z) = e^z, \\ \varphi_p(z) = \displaystyle\int_0^1 e^{(1-\theta)z} \frac{\theta^{p-1}}{(p-1)!} d\theta, \ \forall p \geq 1, \end{cases} \tag{12}$$

which satisfy the following recurrence relation

$$\varphi_{p+1}(z) = \frac{1}{z}\left(\varphi_p(z) - \frac{1}{p!}\right). \tag{13}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

Here, we will focus on the simplest exponential integrator method, the *Exponential Euler* method, which is first order in time. For higher-order methods, we refer to [15, 16]. For that, we approximate in (11) the non-linear term with its value at t_n that is known, i.e., $F(U(t_n + \tau\theta)) \approx F(U_n)$. Integrating exactly in (11), we obtain

$$U_{n+1} = \varphi_0(\tau A)U_n + \tau\varphi_1(\tau A)F(U_n), \tag{14}$$

which is given in terms of the φ-functions (12). Finally, employing the recurrence formula (13), we rewrite (14) as

$$U_{n+1} = U_n + \tau\varphi_1(\tau A)(F(U_n) + \tau AU_n). \tag{15}$$

For the numerical results, we employ the *MATLAB* routines from [12] for computing the action of φ-functions over vectors. In these routines, the authors employ the scaling and squaring method together with a truncated Taylor series approximation to the exponential of a matrix. As we show in the numerical results, these routines applied to operator τA coming from finite difference semi-discretization in space are extremely efficient. Moreover, exponential integrators are suitable for long-time simulations as they are unconditionally stable.

Thus, in the exponential integrators method we compute the sequence

$$\begin{aligned} U_1 &= U_0 + \tau \int_0^1 e^{(1-\theta)(\tau A)} d\theta(\rho U_0(1-U_0) + \tau AU_0), \\ U_2 &= U_1 + \tau \int_0^1 e^{(1-\theta)(\tau A)} d\theta(\rho U_1(1-U_1) + \tau AU_1), \\ &\dots \\ U_{n+1} &= U_n + \tau \int_0^1 e^{(1-\theta)(\tau A)} d\theta(\rho U_n(1-U_n) + \tau AU_n). \end{aligned} \tag{16}$$

4 Software description

The EXPBrain code runs the exponential integrator simulations of the glioblastoma tumor growth within the 3D domain. The code parameters are described in Table 1. The dimensions of the domain $x2-x1, y2-y1, z2-z1$ [mm] define the human head size. The code allows the simulation to be performed within a prescribed time interval, starting from $t0$ and ending at the final time moment T. It assumes initial brain tumor location at point xic, yic, zic [mm]. The exponential integrators simulation in time is based on the finite difference approximation in space using the computational mesh with $nelx, nely, nelz$ elements.

The simulation is performed within the human head model based on MRI scan data loaded into the directory brain_scan out_*.png.

The simulation output is a sequence of ParaView files generated into the directory paraview_files tumor_*.vti.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

Parameter	Description
$rho = 0.025$;	tumor proliferation rate
$x1 = 0$; $y1 = 0$; $z1 = 0$;	left-front-bottom domain corner
$x2 = 193$; $y2 = 193$; $z2 = 193$;	right-rear-top domain corner
$t0 = 150$;	initial time moment
$T = 750$;	final time moment
steps=100;	number of time steps
$tau = (T - t0)/steps$;	time step size
$xic = 102$; $yic = 138$; $zic = 96$;	initial location of tumor
$nelx = 32$; $nely = 32$; $nelz = 32$;	number of mesh elements
hx, hy, hz	element diameters
$hx2, hy2, hz2$	element diameters squared
$nx = nelx + 1$; $ny = nely + 1$; $nz = nelz + 1$;	number of grid points

Table 1: Code parameters

4.1 Software architecture

The software metadata are summarized in Table 2. The primary executable is Tumor_growth_3D.m file. First, in the *Initializing* step, the simulation parameters are initialized as described in Table 1.

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v1
C2	Permanent link to code/repository used for this code version	https://github.com/Magdamini/EXPBrain
C4	Legal Code License	GNU General Public License (GPL)
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Matlab, ParaView.
C7	Compilation requirements, operating environments & dependencies	Matlab, https://github.com/higham/expmv/expmv.m, normAm.m, select_taylor_degree.m, select_taylor_degree.m, theta_taylor.mat, theta_taylor_half.mat, theta_taylor_single.mat
C9	Support email for questions	maciej.paszynski@agh.edu.pl

Table 2: Code metadata

Next, the *Reading MRI scan* step calls the get_diffusion3d(nx,ny,nz) routine that reads the MRI scan data and sets the diffusion coefficient for white matter, gray matter, cerebrospinal fluid, air, and bones. The routine assumes that the MRI scan provides 256 bitmaps of 194×248 pixels (this step can be adjusted to another MRI scan resolution in the routine if needed).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

The *Generating finite difference matrix* step constructs a finite difference discretization in space with nx, ny, nz elements. The finite difference matrix A is created. The initial tumor configuration is set up with xc, yc, and zc coordinates.

Next, the *Setting up exponential integrators* step decides on the number of time *steps*, and their time length *tau*.

The *Computing exponential integrators* step runs the numerical simulation, followed by pumping out the ParaView files in write_solution_to_files routine.

4.2 Code installation and usage

The code can be downloaded from repository
https://github.com/Magdamini/EXPBrain
(available under GNU General Public License (GPL))
The code is implemented in Matlab and uses the ParaView tool for visualizations. It requires Matlab and the following libraries (included in the GitHub repository)
https://github.com/higham/expmv/expmv.m,
https://github.com/higham/expmv/normAm.m,
https://github.com/higham/expmv/select_taylor_degree.m,
https://github.com/higham/expmv/select_taylor_degree.m,
https://github.com/higham/expmv/theta_taylor.mat,
https://github.com/higham/expmv/theta_taylor_half.mat,
https://github.com/higham/expmv/theta_taylor_single.mat

5 Illustrative examples

The exemplary MRI scan data files are presented in Figure 1. There are 256 scans with a resolution of 194×248 pixels. Running the EXPBrain code starting from $t0 = 150$ for 100-time steps until time moment $T = 750$, using $128 \times 128 \times 128$ finite difference mesh with the initial location of tumor defined as $xic = 102; yic = 138; zic = 96$; produces a sequence of output ParaView files, illustrated in Figures 2-3. The exponential integrators simulation takes 287 [s] (less than 5 minutes) on a single node from Athena supercomputer [13]. It predicts two years of tumor evolution.

6 Conclusions

– The EXPBrain code performs ultra-fast simulations of the glioblastoma brain tumor on the patient's MRI scan data. It can predict two years of future brain tumor growth within 5 minutes on a single computing node, including the generation of output Paraview files. In comparison, alternative highly efficient simulators using finite difference or finite element method for the tumor growth simulations take several minutes to compute a single time step [4–7]. The high performance of our method comes from the fact that the exponential routines are very efficient on operators coming from a semi-discretization in space employing finite differences.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

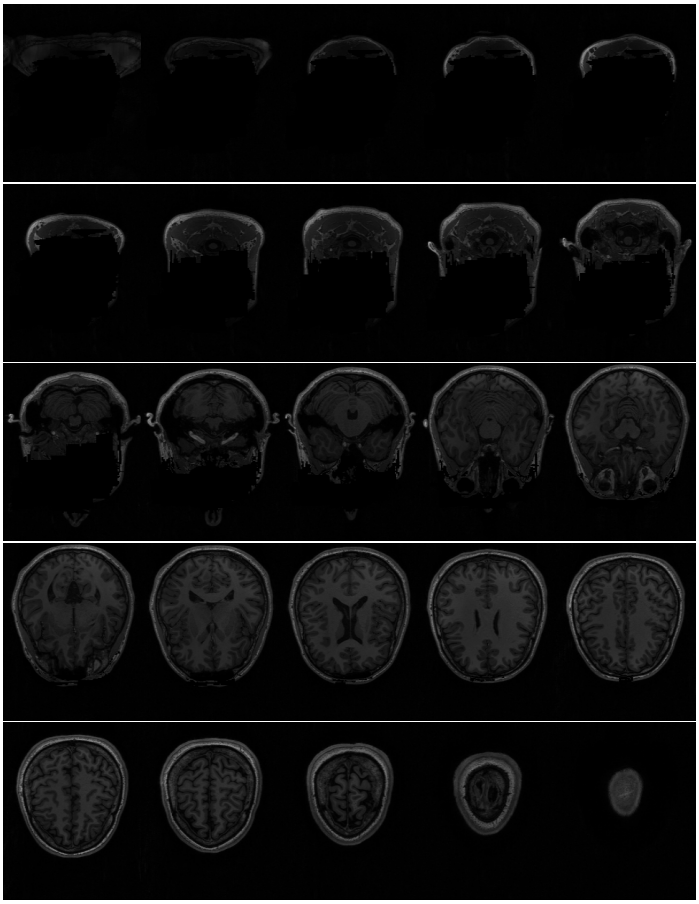

Fig. 1: MRI scans of the human head.

– Fast tumor growth simulators are crucial for patient-specific modeling. Tumor growth models include several patient-specific parameters. Hence, accurate data assimilation techniques fitting model parameters to patient-specific data would require several runs of the simulator [20, 21]. The EXPBrain simulator can be employed in data assimilation algorithms [20, 21] to assimilate the model parameters, such as the diffusion coefficient D specific for each kind of tissue (describing how the tumor cells expand in a tissue, similarly to the diffusion phenomena), and the proliferation rate of the tumor cells ρ (how fast they multiply).
– The EXPBrain simulator could be used by medical doctors to predict the future progression of brain tumor cells for up to two years.
– The future work will attempt to develop the adaptive version of the exponential integrators software [22–25].

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

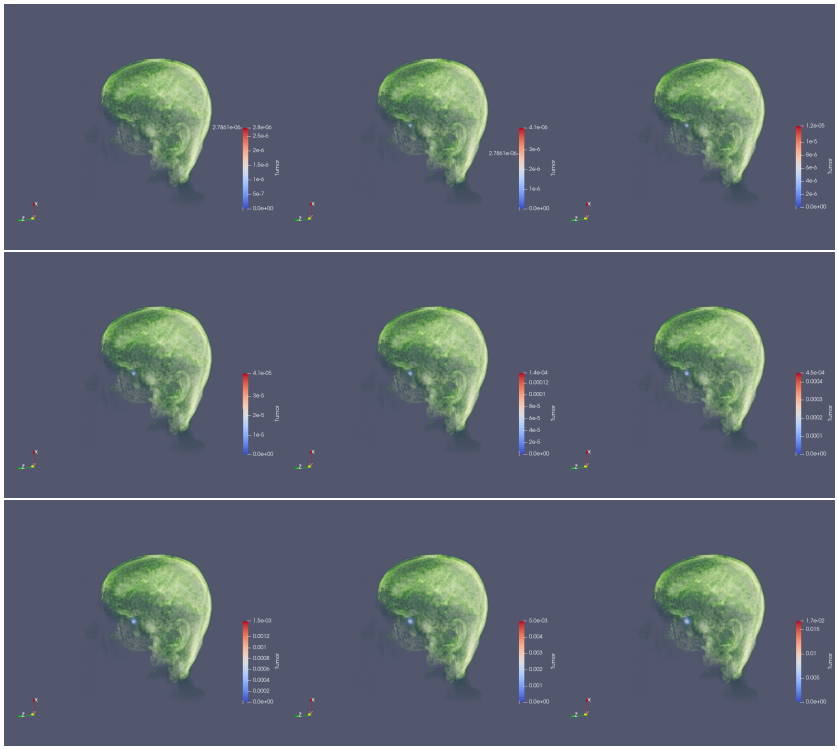

Fig. 2: Glioblastoma brain tumor simulation with exponential integrators method.

Acknowledgements

The work of Askold Vilkha and Maciej Paszyński has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Sklodowska-Curie grant agreement No 101119556. The work of Dominika Ciupek has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857533 and the International Research Agendas Programme of the Foundation for Polish Science No MAB PLUS/2019/13. The work of Dominika Ciupek has also received funding from the project of the Minister of Science and Higher Education "Support for the activity of Centers of Excellence established in Poland under Horizon 2020" on the basis of the contract number MEiN/2023/DIR/3796. Data collection and sharing for this project was provided by the Human Connectome Project (HCP; Principal Investigators: Bruce Rosen, M.D., Ph.D., Arthur W. Toga, Ph.D., Van J. Weeden, MD). HCP funding was provided by the National Institute of Dental and Craniofacial Research (NIDCR), the National Institute of Mental Health (NIMH), and the National Institute of Neurological Disorders and Stroke (NINDS). HCP data are disseminated by the Laboratory of Neuro

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

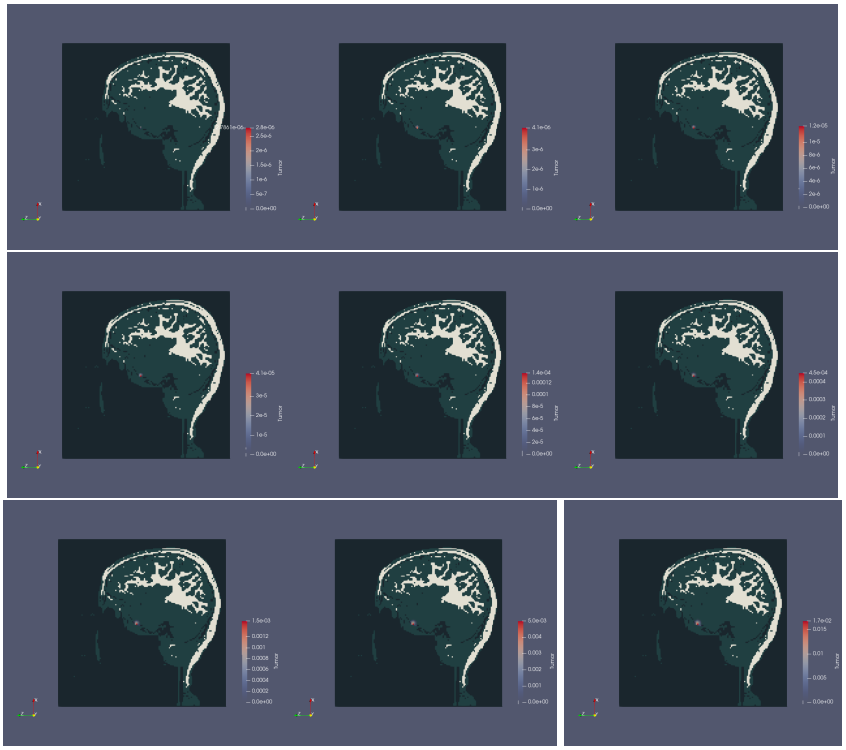

Fig. 3: Glioblastoma brain tumor simulation with exponential integrators method. Cross-section of the head.

Imaging at the University of Southern California. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2024/017764

Appendix

A Sample code snippets analysis

The simulation starts from the initially assumed configuration $u_0(x, y, z)$ of the brain tumor cells. It generates a sequence of ParaView files, allowing for the generation of a sequence of pictures or a movie from the simulation.
The spatial discretization utilizes the finite difference method.

```
    for i = 2:nx-1
        for j = 2:ny-1
            for k = 2:nz-1
                l = i + (j-1)*nx + (k-1)*nx*ny;
```

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

```
                values(idx) = (diff(i+1,j,k)-diff(i,j,k))/
                    hx2
                  + 2.0 * diff(i,j,k)/hx2
                  + (diff(i,j+1,k)-diff(i,j,k))/hy2
                  + 2.0 * diff(i,j,k)/hy2
                  + (diff(i,j,k+1)-diff(i,j,k))/hz2
                  + 2.0 * diff(i,j,k)/hz2;
                row_idx(idx) = l; col_idx(idx) = l;
                values(idx + 1) = - diff(i,j,k)/hx2;
                row_idx(idx + 1) = l; col_idx(idx + 1) = l
                    -1;
                values(idx + 2) =
                  (- diff(i+1,j,k)+diff(i,j,k))/hx2
                  - diff(i,j,k)/hx2;
                row_idx(idx + 2) = l; col_idx(idx + 2) = l
                    +1;
                values(idx + 3) = - diff(i,j,k)/hy2;
                row_idx(idx + 3) = l; col_idx(idx + 3) = l
                    -nx;
                values(idx + 4) = (-diff(i,j+1,k)+diff(i,j
                    ,k))/hy2
                  - diff(i,j,k)/hy2;
                row_idx(idx + 4) = l; col_idx(idx + 4) = l
                    +nx;
                values(idx + 5) = - diff(i,j,k)/hz2;
                row_idx(idx + 5) = l; col_idx(idx + 5) = l
                    -nx*ny;
                values(idx + 6) =
                 ( -diff(i,j,k+1)+diff(i,j,k))/hz2
                 - diff(i,j,k)/hz2;
                row_idx(idx + 6) = l; col_idx(idx + 6) = l
                    +nx*ny;
                idx = idx + 7;
            end
        end
    end
    % Create the sparse matrix A
    A = sparse(row_idx(1:idx-1), col_idx(1:idx-1),
               values(1:idx-1), nx*ny*nz, nx*ny*nz);
```

The exponential integrators simulation is very elegant

```
    %Time grid and step size
    fprintf("Setting up exponential integrators...\n");
    steps = 100;
    tau = (T-t0)/steps;
    t = t0:tau:T;

    %Exponential Euler method
    fprintf("Computing exponential integrators...\n");
```

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

```
U = zeros(dimx,steps+1);
U(:,1) = U0;
for i = 1:steps
    Fu = rho*U(:,i).*(1-U(:,i));
    U(:,i+1) = U(:,i)+tau*phiB(-tau*A,Fu-A*U(:,i));
    show_progress(i, steps);
end
write_solution_to_files;
```

it includes phiB routine from https://github.com/higham/expmv library, performing fast exponential integrators operations.

B Paraview software user guide

After running the computations, the code generates a sequence of *.vti files [26]. To visualize computed data, open the ParaView application [27, 28]. Then click "Open" from the "File" menu in the upper left corner. Choose the file brain.vti. It should be in the paraview_files directory. Once the file is selected, click the "OK" in the dialog box. To display the data, in the "Properties" tab, click the green "Apply" button and set the representation to "Volume". After this step, the brain image should be visible on the screen. Then, repeat the above instructions to open files containing tumor data. When opening those files, Paraview should see them as a group of files named tumor_..vti and open them all at once. After setting up those files, choose the brain.vti file in the "Pipeline Browser". Then focus on the right tab, named "Color Map Editor". Find the bar "Select a color map from default presets" or the small icon with a heart and choose one of the proposed color maps. Use the slider to decrease the opacity. Now, both tumor and brain data should be visible on the screen. Use the mouse to rotate the image. To display (or hide) the scale, select the chosen object and click on the colorful "Toggle Color Legend Visibility" icon in the upper left corner. To start the animation, choose the tumor_1.vti file, and on the right bar, set the "Automatic Rescale Range Mode" to "Grow and update every timestep". Then click the play button in the top part of the window. The tumor image can also be seen at a particular time. To visualize it, change the time option next to the play button and adjust the scale by clicking the "Rescale to Data Range" button.

References

1. M. Hochbruck and A. Ostermann. Explicit exponential Runge–Kutta methods for semilinear parabolic problems. SIAM Journal on Numerical Analysis, 43(3):1069–1090, 2005
2. Caroline Hertler, Jörg Felsberg, Dorothee Gramatzki, Emilie Le Rhun, Jennifer Clarke, Riccardo Soffietti, Wolfgang Wick, Olivier Chinot, François Ducray, Patrick Roth, Kerrie McDonald, Peter Hau, Andreas F. Hottinger, Jaap Reijneveld, Oliver Schnell, Christine Marosi, Michael Glantz, Amélie Darlix, Giuseppe Lombardi, Dietmar Krex, Martin Glas, David A. Reardon, Martin van den Bent, Florence Lefranc, Ulrich Herrlinger, Evangelia Razis, Antoine F. Carpentier, Samuel

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

Phillips, Roberta Rudà, Antje Wick, Emeline Tabouret, David Meyronet, Claude-Alain Maurage, Elisabeth Rushing, Robert Rapkins, Elisabeth Bumes, Monika Hegi, Astrid Weyerbrock, Dawit Aregawi, Christian Gonzalez-Gomez, Alessia Pellerino, Martin Klein, Matthias Preusser, Martin Bendszus, Vassilis Golfinopoulos, Andreas von Deimling, Thierry Gorlia, Patrick Y. Wen, Guido Reifenberger, Michael Weller, Long-term survival with IDH wildtype glioblastoma: first results from the ETERNITY Brain Tumor Funders' Collaborative Consortium (EORTC 1419), *European Journal of Cancer Research*, 189 (2023) 112913

3. R. Stupp, M. Brada, M. J. van den Bent, J. C. Tonn, G. Pentheroudakis, High-grade glioma: ESMO clinical practice guidelines for diagnosis, treatment and follow-up, Annals of Oncology, 25 (2014) 93–101.
4. Jana Lipková, Panagiotis Angelikopoulos, Stephen Wu, Esther Alberts, Benedikt Wiestler, Christian Diehl, Christine Preibisch, Thomas Pyka, Stephanie Elisabeth Combs, Panagiotis E. Hadjidoukas, Koen Van Leemput, Petros Koumoutsakos, John S. Lowengrub, Bjoern H Menze, Personalized Radiotherapy Design for Glioblastoma: Integrating Mathematical Tumor Models, Multimodal Scans, and Bayesian Inference, *IEEE Transactions on Medical Imaging*, 38 (2018) 1875-1884
5. Marcin Łoś, Maciej Paszyński, Adrian Kłusek, Witold Dzwinel, Application of fast isogeometric L2 projection solver for tumor growth simulations, *Computer Methods in Applied Mechanics and Engineering, Special Issue on Isogeometric Analysis: Progress and Challenges*, 316 (2017) 1257–1269.
6. Marcin Łoś, Adrian Kłusek, Muhammad Amber Hassaan, Keshav Pingali, Witold Dzwinel, Maciej Paszyński, Parallel fast isogeometric L2 projection solver with GALOIS system for 3D tumor growth simulations, *Computer Methods in Applied Mechanics and Engineering*, 341 (2019) 1-22.
7. Adrian Kłusek, Marcin Łoś, Maciej Paszyński, Witold Dzwinel, Efficient model of tumor dynamics simulated in multi-GPU environment, *The International Journal of High Performance Computing Applications*, 33(3) (2019) 489–506.
8. Andy Zhu, Accelerating Parameter Inference in Diffusion-Reaction Models of Glioblastoma Using Physics-Informed Neural Networks, *SIAM Undergraduate Resources Online*, (2022)
9. K. R. Swanson, E. C. Alvord Jr, and J. D. Murray. A quantitative model for differential motility of gliomas in grey and white matter. Cell Proliferation, 33(5):317–329, 2000.
10. Zareba, M. R., Fafrowicz, M., Marek, T., Beldzik, E., Oginska, H., Domagalik, A. (2022). Late chronotype is linked to greater cortical thickness in the left fusiform and entorhinal gyri. Biological Rhythm Research, 53(10), 1626-1638.
11. MAGNETOM Skyra 3T MRI scanner *https://www.siemens-healthineers.com/pl/magnetic-resonance-imaging/3t-mri-scanner/magnetom-skyra* (accessed on December 20, 2024)
12. A. H. Al-Mohy and N. J. Higham. Computing the action of the ma trix exponential, with an application to exponential integrators. SIAM Journal on Scientific Computing, 33(2):488–511, 2011.
13. Athena supercomputer *https://www.cyfronet.pl/en/19073,artykul,athena.html* (accessed on December 20, 2024)
14. Maziar Raissi, Paris Perdikaris, George E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational physics*, 378 (2019) 686–707.
15. Marlis Hochbruck, Alexander Ostermann, Exponential integrators, *Acta Numerica*, 19 (2010) 209–286.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10

16. Judit Muñoz-Matute, Leszek Demkowicz, Multistage DPG time-marching scheme for nonlinear problems, *arXiv:2309.00069* (2023)
17. Matteo Croci, Judit Muñoz-Matute, Exploiting Kronecker structure in exponential integrators: Fast approximation of the action of ϕ-functions of matrices via quadrature, *Journal of Computational Science*, 67 (2023) 101966.
18. Lucia Maddalena, Stefania Ragni, Existence of solutions and numerical approximation of a non-local tumor growth model, *Mathematical Medicine and Biology: A Journal of the IMA*, 37(1) (2020) 58–82
19. Magdalena Pabisz, Judit Muñoz-Matute, Maciej Paszyński, Augmenting MRI scan data with real-time predictions of glioblastoma brain tumor evolution using faster exponential time integrators, *Journal of Computational Science*, 85 (2025) 102493.
20. Maciej Paszyński, Leszek Siwik, Witold Dzwinel, Keshav Pingali, Supermodeling, a convergent data assimilation meta-procedure used in simulation of tumor progression, *Computers & Mathematics with Applications*, 113 (2022) 214-224.
21. Leszek Siwik, Marcin Łoś, Adrian Kłusek, Anna Paszyńska, Keshav Pingali, Witold Dzwinel, Maciej Paszyński, Tuning three-dimensional tumor progression simulations on a cluster of GPGPUs, *Journal of Computational and Applied Mathematics*, 412 (2022) 114308.
22. Anna Paszyńska, Maciej Paszyński, Konrad Jopek, M Woźniak, Damian Goik, Piotr Gurgul, Hassan AbouEisha, Mikhail Moshkov, Victor Manuel Calo, Andrew Lenharth, Donald Nguyen, Keshav Pingali, Quasi-Optimal Elimination Trees for 2D Grids with Singularities, Scientific Programming, 1 (2015) 303024
23. Anna Paszyńska, Maciej Paszyński, Ewa Grabska, Graph Transformations for Modeling hp-Adaptive Finite Element Method with Triangular Elements Lecture Notes in Computer Science, 5103 (2008) 604-613.
24. Maciej Paszyński, Anna Paszyńska, Graph Transformations for Modeling Parallel *hp*-Adaptive Finite Element Method, Parallel Processing and Applied Mathematics: 7th International Conference, PPAM 2007, Gdańsk, Poland, September 9-12 (2007) 1313-1322.
25. Damin Goik, Konrad Jopek, Maciej Paszyński, Andrew Lenharth, Donald Nguyen, Keshav Pingali, Graph grammar based multi-thread multi-frontal direct solver with Galois scheduler Procedia Computer Science 29 (2014) 960-969
26. Will Schroeder, Ken Martin, Bill Lorensen, The Visualization Toolkit (4th ed.) Kitware (2006).
27. James Ahrens, Berk Geveci, Charles Law, ParaView: An End-User Tool for Large Data Visualization, *Visualization Handbook*, Elsevier (2005)
28. Utkarsh Ayachit, *The ParaView Guide: A Parallel Visualization Application*, Kitware, 2015

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_10