

Supermodeling of a Tumor with Isogeometric Analysis Solvers

Maciej Paszyński

Marcin Łoś, Leszek Siwik, Adrian Kłusek, Witold Dzwiniel



Department of Computer Science
AGH University of Science and Technology, Kraków, Poland
home.agh.edu.pl/paszynsk

Oden Institute, The University of Texas at Austin
August 6, 2019

- [1] Marcin Łoś, Maciej Paszyński, Adrian Kłusek, Witold Dzwiniel, [Application of fast isogeometric L2 projection solver for tumor growth simulations](#), **Computer Methods in Applied Mechanics and Engineering** 316 (2017), 1257-1269
- [2] Marcin Łoś, Adrian Kłusek, Muhammad Amber Hassaan, Keshav Pingali, Witold Dzwiniel, Maciej Paszyński, [Parallel fast isogeometric L2 projection solver with GALOIS system for 3D tumor growth simulations](#), **Computer Methods in Applied Mechanics and Engineering**, 343 (2019) 1-22
- git clone -b tumor --single-branch
<https://github.com/marcinlos/iga-ads>
- [3] Leszek Siwik, Marcin Łoś, Adrian Kłusek, Keshav Pingali, Witold Dzwiniel, Maciej Paszyński, [Supermodeling of Tumor with Isogeometric Analysis Solvers](#) in preparation. (2019)

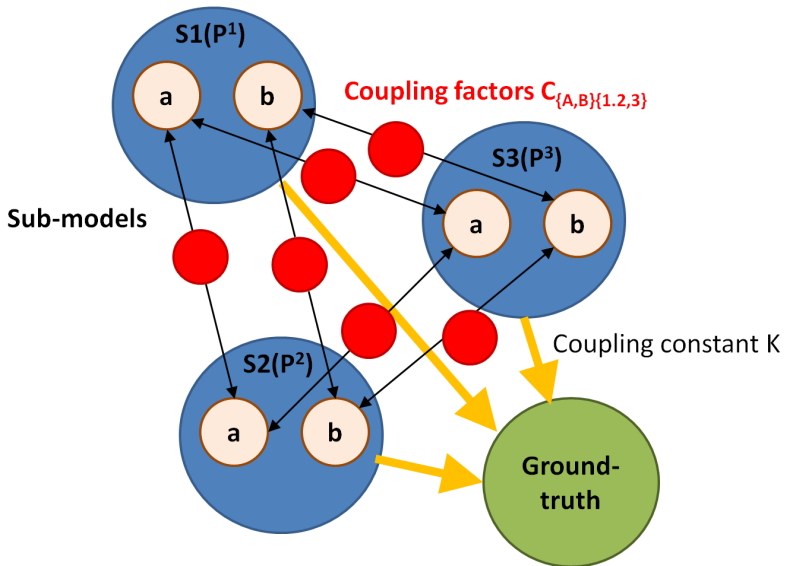
- Introduction
- Critical issues of supermodeling
- Tumor model
- Sensitivity analysis
- Supermodeling algorithm
- Numerical results
- Conclusions

- Data assimilation is the key component of computer simulations
- Computer models of a tumor with several dozen of parameters
- Solution space explodes with a number of parameters
- Classical data assimilation algorithms result in prohibitively long computations
- The single tumor model itself may not be able to match the reality well
- We propose supermodeling as a second abstraction layer to classical data assimilation procedures, which can improve their performance

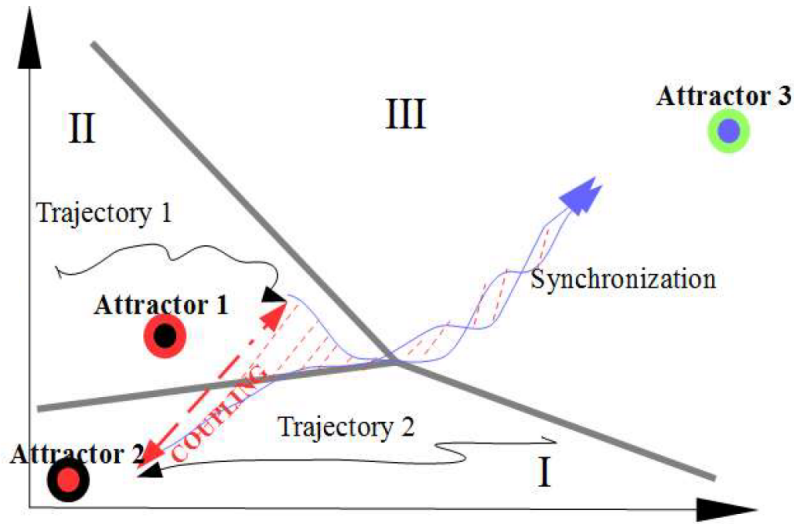
Frank M. Selten, Francine J. Schevenhoven, and Gregory S. Duane, [Simulating climate with a synchronization based supermodel](#), **Chaos** 27, 126903 (2017)

Critical issues of supermodeling

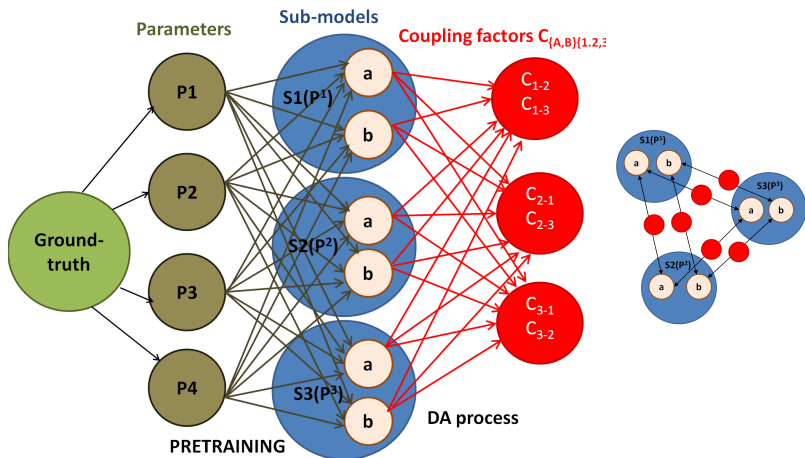
Supermodeling



How does it work?



Supermodeling



Critical issues in supermodeling

- Which sub-models: heterogeneous, homogeneous
- How many sub-models ($M=?$) and # teaching samples
- How to select sub-models?
- Number of and which dynamic variables are coupled ($N=?$)
- Strong or weak coupling?
- Training procedure?

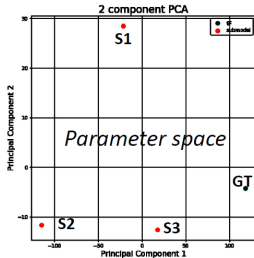
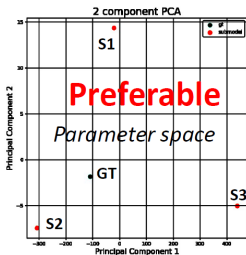
The general idea is to speed up data assimilation for a complex multi-parameter dynamical process by adding the **supermodeling** abstraction layer.

- The supermodel consists of **homogeneous M** sub-models with various parameter sets $\mathbf{P}_1 = (p_1^1, \dots, p_n^1), \dots, \mathbf{P}_M = (p_1^M, \dots, p_n^M)$
- We use supermodeling for prediction of the system **trajectory**
- As a **ground truth**, we use the results produced by another simulation.

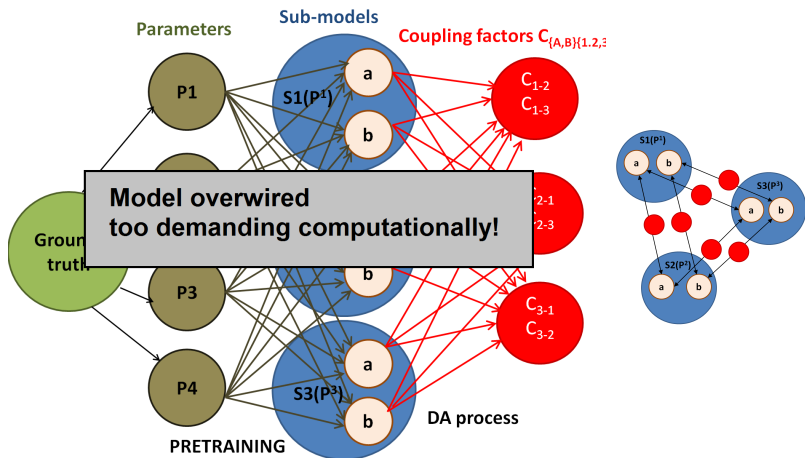
Adrian Kłusek, Marcin Łoś, Maciej Paszyński, Witold Dzwiniel, [Efficient model of tumor dynamics simulated in multi-GPU environment](#), **The International Journal of High Performance Computing Applications**, 33(3) (2019) 1-18

How to select sub-models ?

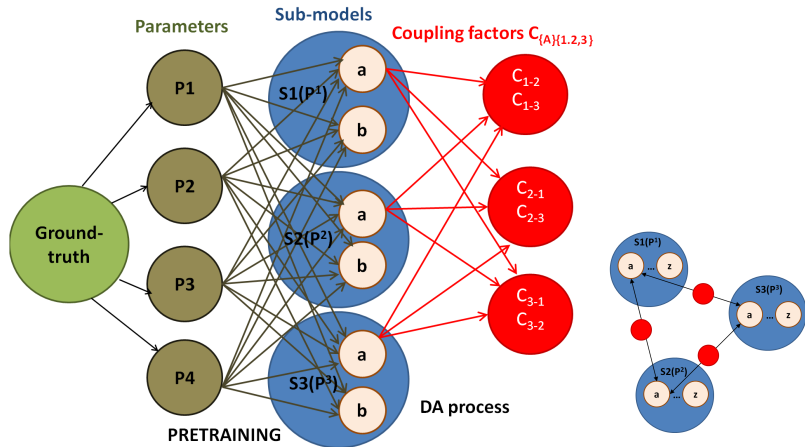
- Randomly selected set of parameters P_1, P_2, \dots, P_M for each sub-model
- Pretrained models as sub-models (using classical DA procedures, e.g., resulting from inverse modeling, sensitivity analysis)
- Rule of thumb: close to the GT, should be closed to different „good” local minima surrounding GT



Coupling all dynamic variables?



Coupling one dynamic variable

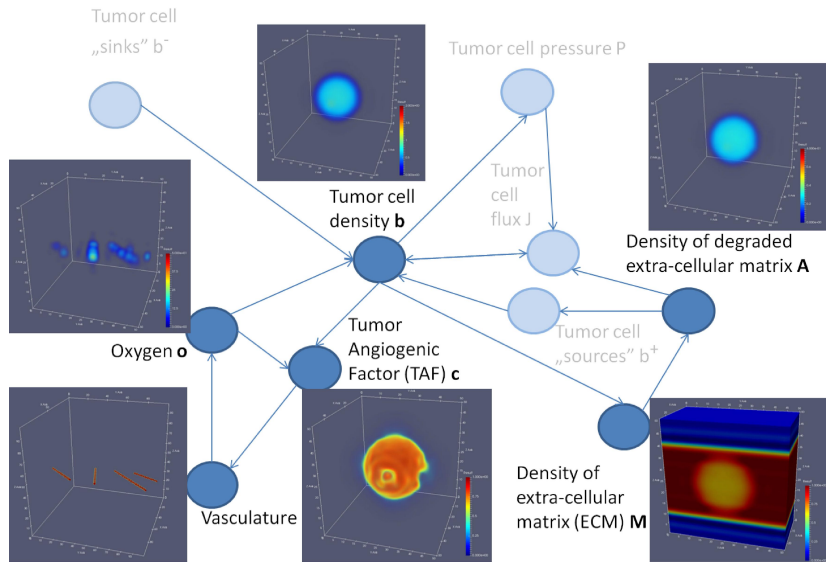


Supermodel construction

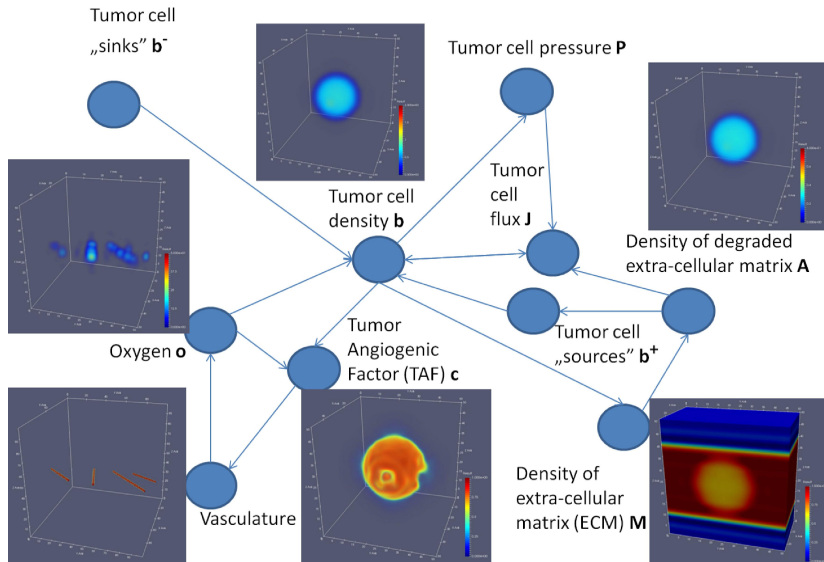
- Selection of homogenous model with $\mathbf{P} = (p_1, \dots, p_n)$ parameters
- Sensitivity analysis: find the most sensitive parameters and dynamical variables
- Classical data assimilation: find M sets of parameters $\mathbf{P}_1 = (p_1^1, \dots, p_n^1), \dots, \mathbf{P}_M = (p_1^M, \dots, p_n^M)$
- Create the supermodel by coupling the submodels via the most sensitive dynamical variable
- Train the supermodel: estimate $M!$ coupling coefficients by using classical data assimilation procedure
- Validate the supermodel on a test data

Tumor model

Tumor growth model



Tumor growth model



$$\left\{ \begin{array}{l} \frac{\partial b}{\partial t} = -\nabla \cdot J - \frac{b}{T^{death}} [o < o^{death}] + \\ \quad \frac{b}{T^{prol}} \left(1 + \frac{\tau_b A}{\tau_b A + 1} P_b \right) \left(1 - \frac{b}{b^M} \right) [o > o^{prol}] \\ \frac{\partial c}{\partial t} = \chi_c \Delta c - \gamma_c o c + c^+ \\ \frac{\partial o}{\partial t} = \alpha_o \Delta o - \gamma_o b o + \delta_o (o^{max} - o) \\ \frac{\partial M}{\partial t} = -\beta_M M b \\ \frac{\partial A}{\partial t} = \gamma_A M b + \chi_{OA} \Delta A - \gamma_{OAA} \end{array} \right.$$

Dynamic variable used for coupling: **tumor cell density b**

Most sensitive model parameters:

tumor cell proliferation threshold o^{prol} and hypoxia threshold o^{death} ,
tumor cell proliferation time T^{prol} and survival time T^{death}

Supermodeling of tumor

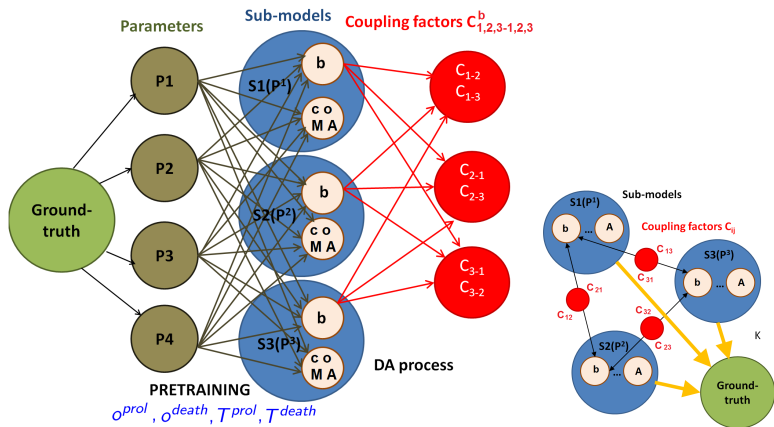


Figure: Dynamic variable used for coupling: tumor cell density b
 Most sensitive model parameters:
 tumor cell proliferation threshold o^{prol} and hypoxia threshold o^{death} ,
 tumor cell proliferation time T^{prol} and survival time T^{death}

Single submodel: Numerical formulation

Explicit time discretization:

$$\begin{cases} b_{t+1} = b_t + \Delta t (-\nabla \cdot J_t + b_t^- + b_t^+) \\ c_{t+1} = c_t + \Delta t (\chi_c \Delta c_t - \gamma_c o_t c_t + c_t^+) \\ o_{t+1} = o_t + \Delta t (\alpha_0 \Delta o_t - \gamma_o b_t o_t + \delta_o (o^{max} - o_t)) \\ M_{t+1} = M_t + \Delta t (-\beta_M M_t b_t) \\ A_{t+1} = A_t + \Delta t (\gamma_A M_t b_t + \chi_{OA} \Delta A_t - \gamma_{OA} A_t) \end{cases}$$

$$J = -D_b b (\nabla P + r_b \nabla A)$$

$$P = \begin{cases} 0 & \text{for } b < b^N \\ \frac{b-b^N}{b^M-b^N} & \text{for } b^N \leq b \leq b^M \end{cases}$$

$$b^+ = \frac{b}{T^{prol}} \left(1 + \frac{\tau_b A}{\tau_b A + 1} P_b \right) \left(1 - \frac{b}{b^M} \right) \quad \text{for } o > o^{prol}$$

$$b^- = -\frac{b}{T^{death}} \quad \text{for } o < o^{death}$$

Initial state:

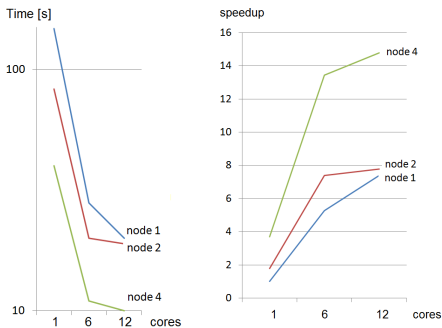
- tumor concentrated in the center of the domain
- constant ECM in each skin layer
- no TAF, no degraded ECM

Isogeometric analysis (IGA-FEM with B-splines basis functions)
with Alternating Directions Solver (ADS)

Parameters:

- $120 \times 120 \times 120$ elements
- quadratic B-splines ($p = 2$)
- $\Delta t = 1$ (60 minutes of reality)
- 300 time steps

Single submodel: Numerical results 3D



$300 \times 10[s] = 3000[s] = 50$ minutes of simulation using GLUON*
on 4 nodes of PROMETHEUS cluster from CYFRONET

Roshan Dathathri, Gurbinder Gill, Loc Hoang, Hoang-Vu Dang, Alex Brooks, Nikoli Dryden, Marc Snir, Keshav Pingali,

[GLUON: A Communication - Optimizing Substrate for Distributed Heterogeneous Graph Analytics](#), **Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) June 2018**

Sensitivity analysis

Model parameters

Symbol	Value	Description
b_m	0	min tumor cell density
b_M	2	max tumor cell density
b^{norm}	1	normal tumor cell density
D_b	varies	tumor cell diffusion rate
r_b	0.3	tumor cells chemoattractant sensitivity
σ^{prol}	10	tumor proliferation threshold
σ^{death}	2	tumor cell hypoxia threshold
T^{prol}	10	tumor cell proliferation time
T^{death}	100	tumor cell survival time
P_b	0.001	maximum stimulated mitosis rate
τ_b	0.5	instantaneous reaction rate
β_M	0.0625	ECM decay rate
γ_A	0.032	production rate of attractants
χ_{aA}	0.000641	decay rate of digested ECM
γ_{oA}	0.000641	diffusion rate of digested ECM
χ_c	0.0000555	TAF diffusion rate
γ_c	0.01	TAF decay rate
α_o	0.0000555	oxygen diffusion rate
γ_o	0.01	oxygen consumption rate
δ_o	0.4	oxygen delivery rate
σ^{max}	60	maximal oxygen concentration

Table: Continuous model parameters

Sensitivity analysis

The IGA-ADS tumor solver is a stand-alone code, executed with the input parameters provided from the command line, e.g.:

```
./tumor 2 80 10000 0.1 1000 0.5 10 2 10 100 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555  
0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4
```

We perform the sensitivity analysis of the model using the following method. We start with the above reference values of the parameters. We pick one parameter, and we run 20 simulations varying its values +/- 10 percent over the range presented in Tables, while keeping other parameters fixed.

For example, possible modifications of parameter p_6 are as follows:

```
./tumor 2 80 10000 0.1 1000 0.45 10 2 10 100 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555  
0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4
```

```
./tumor 2 80 10000 0.1 1000 0.455 10 2 10 100 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555  
0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4
```

...

```
./tumor 2 80 10000 0.1 1000 0.55 10 2 10 100 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555  
0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4
```

Sensitivity analysis

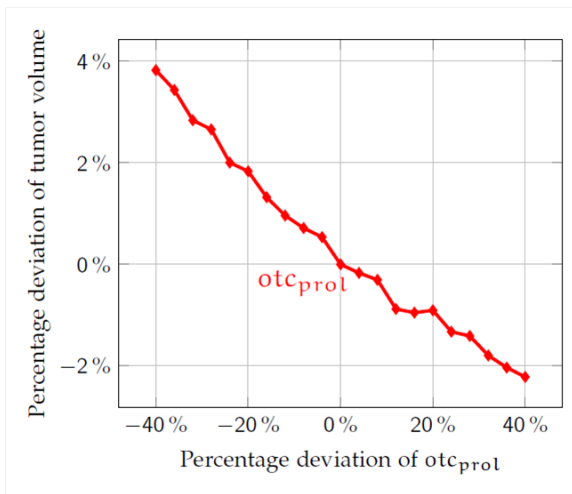


Figure: Sensitivity of the tumor model with respect to tumor proliferation threshold.

Sensitivity analysis

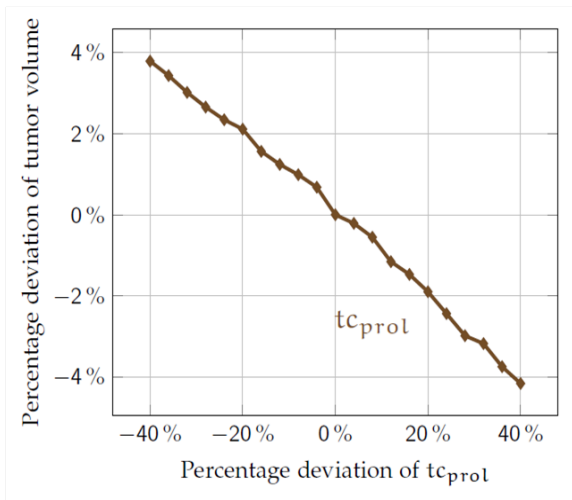


Figure: Sensitivity of the tumor model with respect to tumor cell proliferation time.

Sensitivity analysis

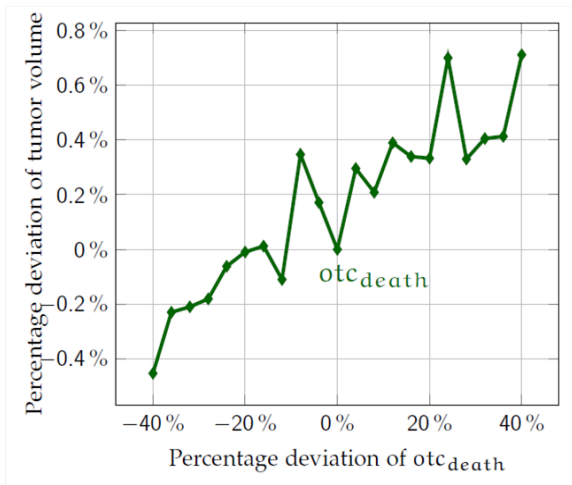


Figure: Sensitivity of the tumor model with respect to tumor cell hypoxia threshold.

Sensitivity analysis

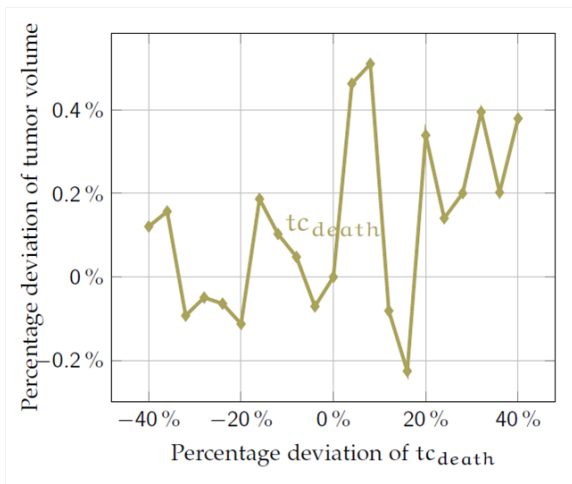
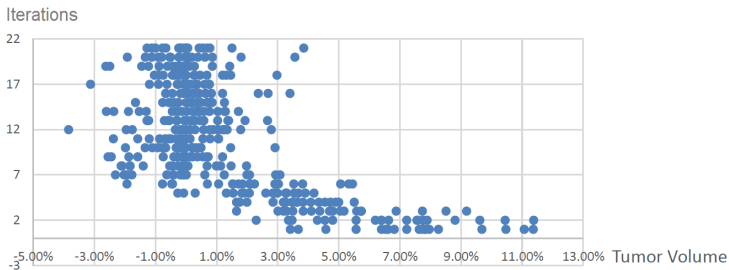
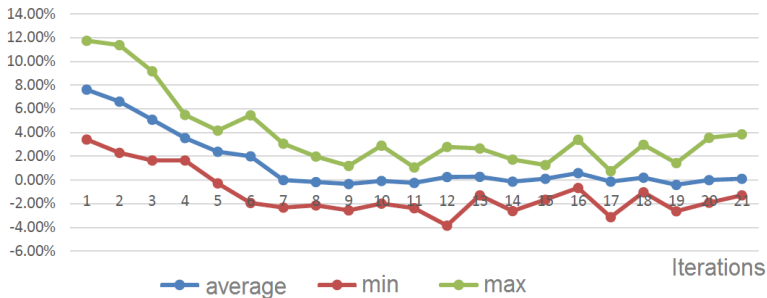


Figure: Sensitivity of the tumor model with respect to tumor cell survival time.

Inverse problem solution with GA



Tumor Volume

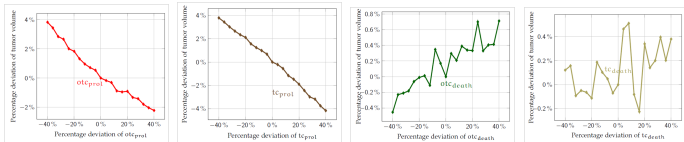


Supermodeling algorithm

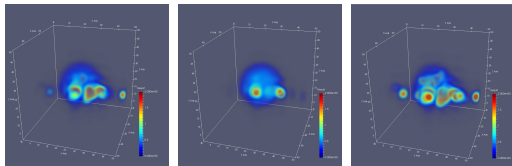
Supermodeling algorithm: Initialization

INITIALIZATION

- 1 Perform sensitivity analysis to find most sensitive parameters



- 2 Optionally solve inverse problem to find local minima
- 3 Setup three submodels *sim1*, *sim2*, *sim3* with different parameters, resulting in different tumor progressions



TRAINING

- 1 Setup identical initial states in each submodel,
- 2 Setup coupling weights C_{ij}^b for $i, j = 1, 2, 3$, setup K coefficient
- 3 For STEP=1,300
 - A Run 1 step in each simulator (sim1, sim2, sim3) and „reality”
 - B Modify obtained fields using the coupling constants

$$b_i(x, y) + = \sum_{i=1,2,3} C_{ij}^b (b_j(x, y, z) - b_i(x, y, z)) + \sum_{i=1,2,3} K (b_i(x, y, z) - b_{meas}(x, y, z))$$

- C Correct the coupling parameter

$$C_{ij}^b + = \int_{\Omega} (b_i(x, y, z) - b_{meas}(x, y, z)) (b_i(x, y, z) - b_j(x, y, z))$$

EXECUTION

- 1 Setup identical initial states in each submodel,
- 2 Use coupling weights C_{ij}^b for $i, j = 1, 2, 3$, and K coefficient as obtained from training stage
- 3 For STEP=1,300
 - A Run 1 step in each simulator (sim1, sim2, sim3) and „reality”
 - B Modify obtained fields using the coupling constants

$$b_i(x, y)_{+} = \sum_{i=1,2,3} C_{ij}^b (b_j(x, y, z) - b_i(x, y, z)) + \sum_{i=1,2,3} K (b_i(x, y, z) - b_{meas}(x, y, z))$$

Numerical results

Numerical results: Experiment 1

$$C_{ij} = 0.5, K = 2.0$$

sim1 with tumor proliferation threshold $\sigma_1^{prol} = 3.0$,

sim1 with tumor proliferation threshold $\sigma_2^{prol} = 5.0$,

sim1 with tumor proliferation threshold $\sigma_3^{prol} = 15.0$,

reality

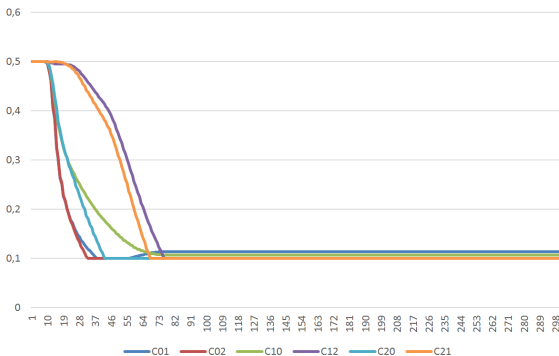


Figure: Convergence of coupling coefficients C_{ij}

Numerical results: Experiment 1

$$C_{ij} = 0.5, K = 2.0$$

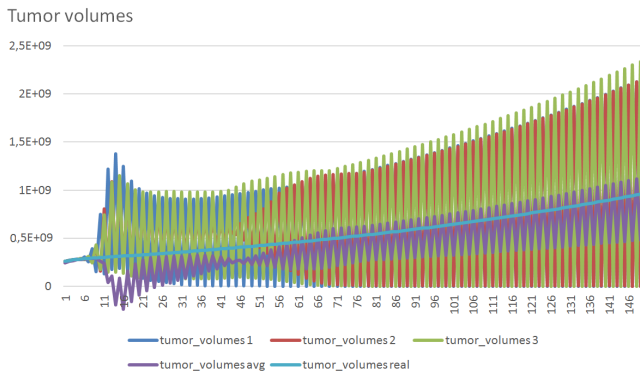


Figure: Convergence of tumor volumes for different submodels *sim1*, *sim2*, *sim3*, for the averaged model $(sim1+sim2+sim3)/3$, for the supermodel, with respect to the "reality".

To obtain better fitting, we will change reality coupling constant K

Numerical results: Experiment 2

$$C_{ij} = 0.5, K = 0.9$$

sim1 with tumor proliferation threshold $\sigma_1^{prol} = 3.0$,

sim1 with tumor proliferation threshold $\sigma_2^{prol} = 5.0$,

sim1 with tumor proliferation threshold $\sigma_3^{prol} = 15.0$,

reality

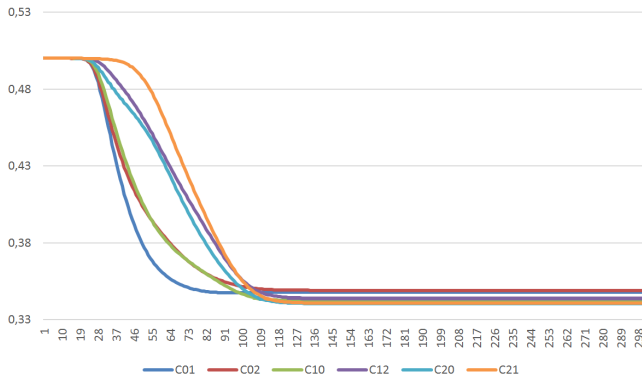


Figure: Convergence of coupling coefficients C_{ij}

Numerical results: Experiment 2

$$C_{ij} = 0.5, K = 0.9$$

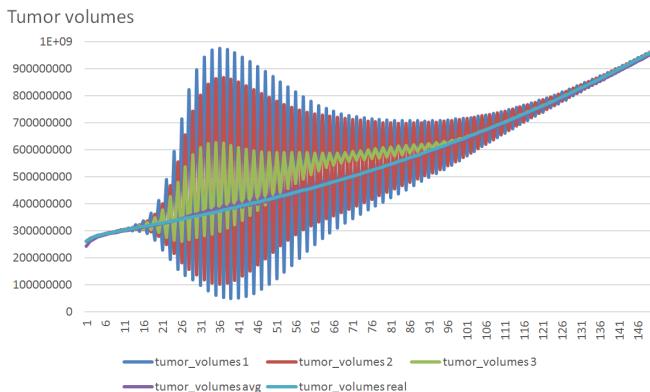


Figure: Convergence of tumor volumes for different submodels $sim1$, $sim2$, $sim3$, for the averaged model $(sim1+sim2+sim3)/3$, for the supermodel, with respect to the "reality".

Numerical results: Experiment 2

$$C_{ij} = 0.5, K = 0.9$$

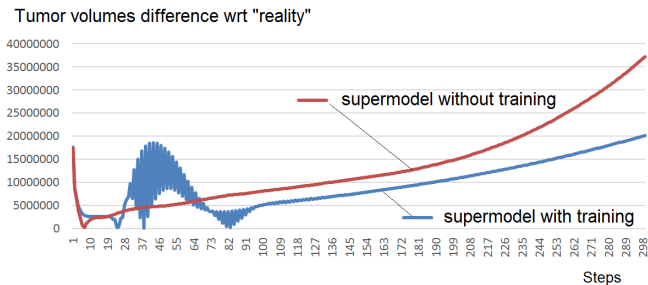


Figure: Difference between supermodel with respect to the "reality", for the supermodel before and after the training phase.

To obtain better fitting, we will select different submodels

Numerical results: Experiment 3

$$C_{ij} = 0.5, K = 0.9$$

sim1 with tumor proliferation threshold $\sigma_1^{prol} = 0.1$,

sim1 with tumor proliferation threshold $\sigma_2^{prol} = 10.0$,

sim1 with tumor proliferation threshold $\sigma_3^{prol} = 30.0$,

reality

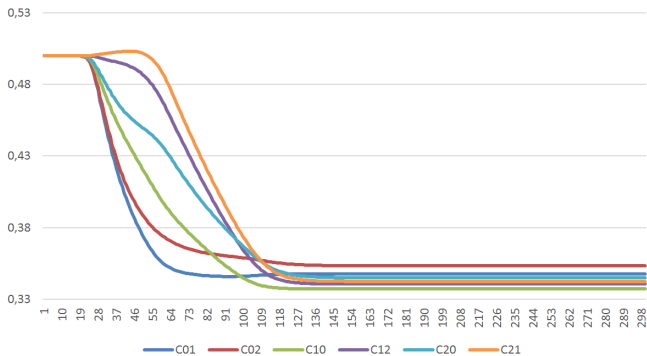


Figure: Convergence of coupling coefficients C_{ij}

Numerical results: Experiment 3

$$C_{ij} = 0.5, K = 0.9$$

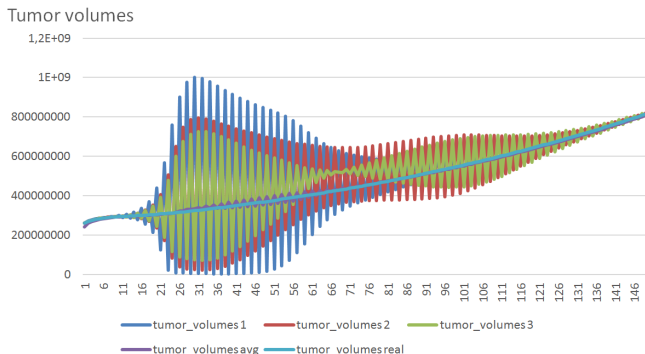


Figure: Convergence of tumor volumes for different submodels *sim1*, *sim2*, *sim3*, for the averaged model $(sim1+sim2+sim3)/3$, for the supermodel, with respect to the "reality".

Numerical results: Experiment 3

$$C_{ij} = 0.5, K = 0.9$$

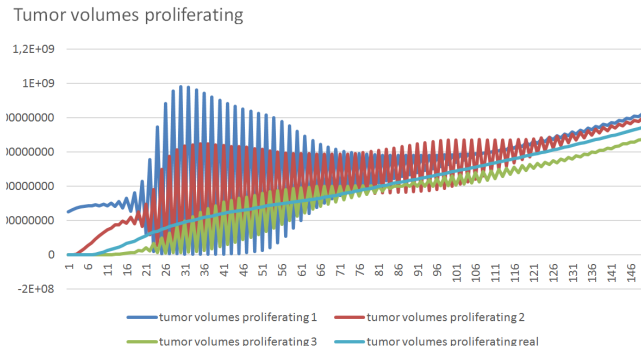


Figure: Convergence of tumor volumes for proliferating cells, for different submodels $sim1$, $sim2$, $sim3$, for the averaged model $(sim1+sim2+sim3)/3$, for the supermodel, with respect to the "reality".

Numerical results: Experiment 3

$$C_{ij} = 0.5, K = 0.9$$

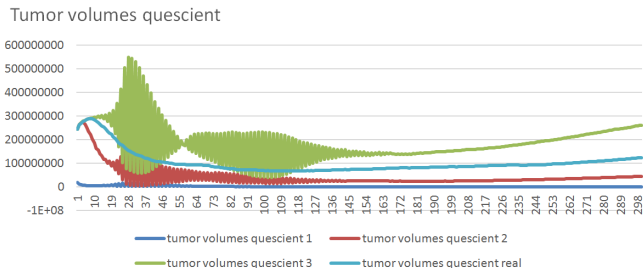


Figure: Convergence of tumor volumes for quiescent cells, for different submodels $sim1$, $sim2$, $sim3$, for the averaged model $(sim1+sim2+sim3)/3$, for the supermodel, with respect to the "reality".

Numerical results: Experiment 3

$$C_{ij} = 0.5, K = 0.9$$

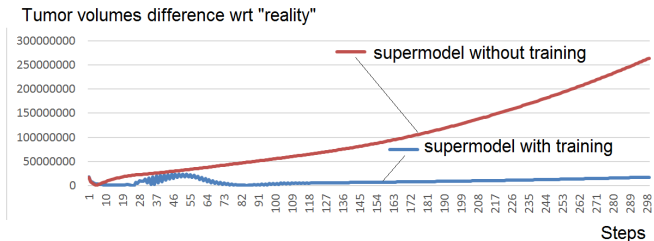


Figure: Difference between supermodel with respect to the "reality", for the supermodel before and after the training phase.

Conclusions

- Tumor growth model with 20+ parameters
- Linear cost $\mathcal{O}(N)$ IGA-ADS solver for tumor growth simulations coupled with discrete vasculature graph
- Sensitivity analysis
- Inverse analysis
- Supermodeling for intelligent coupling of several sub-models
- Training phase to find coupled coefficients C_{ij} , followed by the supermodel simulation phase
- Good agreement with "reality" when proper coupling of supermodel with reality and when we have a good selection of sub-models

The work has been supported by National Science Centre, Poland grant no. 2016/ 21/B/ST6/01539.

The visit of Maciej Paszyński at Oden Institute has been supported by J. T. Oden Research Faculty Fellowship.