Tumor growth simulation using isogeometric L^2-projections solver

Marcin Łoś **Witold Dzwinel** **Maciej Paszyński**

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

Department of Computer Science
AGH University, Kraków, Poland

Agenda

- Isogeometric L2 projections algorithm

Proposed by prof. Victor Calo: L. Gao, V.M. Calo, *Fast Isogeometric Solvers for Explicit Dynamics*, Computer Methods in Applied Mechanics and Engineering, (2014).

- Tumor growth model

Obtained from prof. Witold Dzwinel: W. Dzwinel, A. Kłusek, O.V. Vasilyev, *Supermodeling in Simulation of Melanoma Progression*, Procedia Computer Science, 80 (2016) 999–1010

- Numerical results
- Conclusions
- Further research

Isogeometric L2 projections

The Alternating Direction Implicit (ADI) method

G. Birkhoff, R.S. Varga, D. Young, *Alternating direction implicit methods*, **Advanced Computing** (1962)

Isogeometric L2 projections proposed by prof. Victor Manuel Calo

L. Gao, V.M. Calo, *Fast Isogeometric Solvers for Explicit Dynamics*, **Computer Methods in Applied Mechanics and Engineering**, (2014).

Parallel version for shared memory parallel machines (GALOIS) (collaboration with prof. Keshav Pingali (ICES))

Paper under construction (CPC)

Isogeometric L2 projections

In general: non-stationary problem of the form

$$\partial_t u - \mathcal{L}(u) = f(x, t)$$

with some initial state u_0 and boundary conditions

$\mathcal{L}$ – well-posed linear spatial partial differential operator

Discretization:

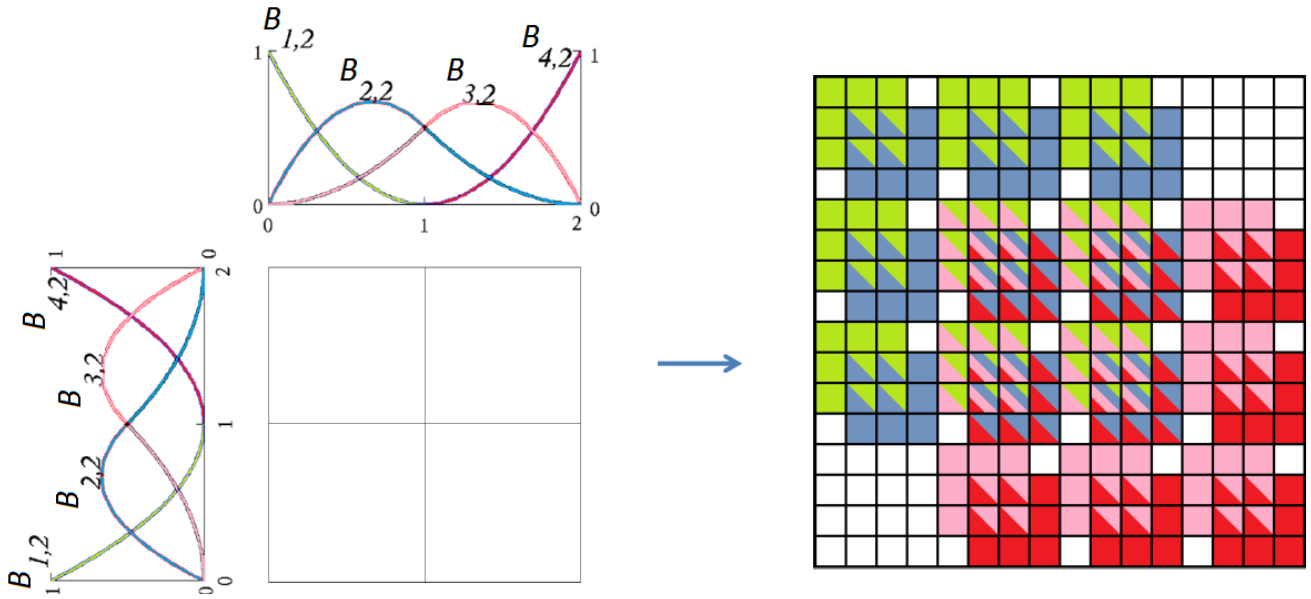
- spatial discretization: isogeometric FEM

  Basis functions: $\phi_1, \ldots, \phi_n$ (tensor product B-splines)

- time discretization with explicit method

- implies isogeometric L2 projections in every time step

L^2 projections – tensor product basis



Isogeometric basis functions:

- 1D B-splines basis $B_1(x), \ldots, B_n(x)$
- higher dimensions: tensor product basis
  $B_{i_1 \cdots i_d}(x_1, \ldots, x_d) \equiv B_{i_1}^{x_1}(x_1) \cdots B_{i_d}^{x_d}(x_d)$

Gram matrix of B-spline basis on 2D domain $\Omega = \Omega_x \times \Omega_y$:

$$\mathcal{M}_{ijkl} = (B_{ij}, B_{kl})_{L^2} = \int_\Omega B_{ij} B_{kl} \, \mathrm{d}\Omega$$

Standard multi-frontal solver: $O(N^{1.5})$ in 2D, $O(N^2)$ in 3D

L^2 projections – tensor product basis

Isogeometric basis functions:

- 1D B-splines basis $B_1(x), \ldots, B_n(x)$
- higher dimensions: tensor product basis
  $B_{i_1 \cdots i_d}(x_1, \ldots, x_d) \equiv B_{i_1}^{x_1}(x_1) \cdots B_{i_d}^{x_d}(x_d)$

Gram matrix of B-spline basis on 2D domain $\Omega = \Omega_x \times \Omega_y$:

$$\mathcal{M}_{ijkl} = (B_{ij}, B_{kl})_{L^2} = \int_\Omega B_{ij} B_{kl} \, \mathrm{d}\Omega$$

$$= \int_\Omega B_i^x(x) B_j^y(y) B_k^x(x) B_l^y(y) \, \mathrm{d}\Omega$$

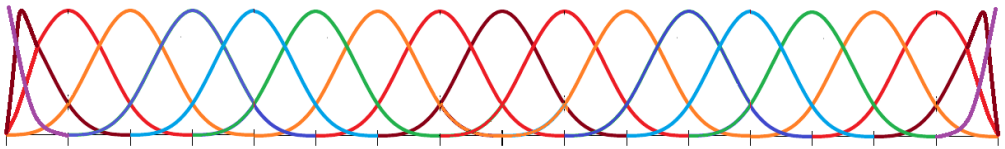
$$= \int_\Omega (B_i B_k)(x) \, (B_j B_l)(y) \, \mathrm{d}\Omega$$

$$= \left(\int_{\Omega_x} B_i B_k \, \mathrm{d}x \right) \left(\int_{\Omega_y} B_j B_l \, \mathrm{d}y \right)$$

$$= \mathcal{M}_{ik}^x \mathcal{M}_{jl}^y$$

$$\mathcal{M} = \mathcal{M}^x \otimes \mathcal{M}^y \quad \text{(Kronecker product)}$$

Gram matrix of tensor product basis



B-spline basis functions have **local support** (over $p+1$ elements)

$\mathcal{M}^x$, $\mathcal{M}^y$, ... – banded structure

$\mathcal{M}^x_{ij} = 0 \iff |i-j| > 2p+1$

Exemplary basis functions and matrix for cubics

$$\begin{bmatrix}
(B_1, B_1)_{L^2} & (B_1, B_2)_{L^2} & (B_1, B_3)_{L^2} & (B_1, B_4)_{L^2} & 0 & 0 & \cdots & 0 \\
(B_2, B_1)_{L^2} & (B_2, B_2)_{L^2} & (B_2, B_3)_{L^2} & (B_2, B_4)_{L^2} & (B_2, B_5)_{L^2} & 0 & \cdots & 0 \\
(B_3, B_1)_{L^2} & (B_3, B_2)_{L^2} & (B_3, B_3)_{L^2} & (B_3, B_4)_{L^2} & (B_3, B_5)_{L^2} & (B_3, B_6)_{L^2} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & (B_n, B_{n-3})_{L^2} & (B_n, B_{n-2})_{L^2} & (B_n, B_{n-1})_{L^2} & (B_n, B_n)_{L^2} &
\end{bmatrix}$$

Alternating Direction Solver

Idea exploit Kronecker product structure of $\mathcal{M} = \mathcal{M}^x \otimes \mathcal{M}^y$

Generally, consider

$$\mathbf{M}\mathbf{x} = \mathbf{b}$$

with $\mathbf{M} = \mathbf{A} \otimes \mathbf{B}$, where $\mathbf{A}$ is $n \times n$, $\mathbf{B}$ is $m \times m$

Definition of Kronecker (tensor) product:

$$\mathbf{M} = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}\, B_{11} & \mathbf{A}\, B_{12} & \cdots & \mathbf{A}\, B_{1m} \\ \mathbf{A}\, B_{21} & \mathbf{A}\, B_{22} & \cdots & \mathbf{A}\, B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}\, B_{11} & \mathbf{A}\, B_{12} & \cdots & \mathbf{A}\, B_{mm} \end{bmatrix}$$

Alternating Direction Solver – 2D

RHS and solution are partitioned into m blocks of size n each

$$\mathbf{x}_i = (x_{i1}, \ldots, x_{in})^{\mathsf{T}}$$
$$\mathbf{b}_i = (b_{i1}, \ldots, b_{in})^{\mathsf{T}}$$

We can rewrite the system as a block matrix equation:

$$\begin{cases} \mathbf{A}B_{11}\mathbf{x}_1 + \mathbf{A}B_{12}\mathbf{x}_2 + \cdots + \mathbf{A}B_{1m}\mathbf{x}_m = \mathbf{b}_1 \\ \mathbf{A}B_{21}\mathbf{x}_1 + \mathbf{A}B_{22}\mathbf{x}_2 + \cdots + \mathbf{A}B_{2m}\mathbf{x}_m = \mathbf{b}_2 \\ \vdots \qquad\qquad \vdots \qquad \vdots \qquad\qquad \vdots \\ \mathbf{A}B_{m1}\mathbf{x}_1 + \mathbf{A}B_{m2}\mathbf{x}_2 + \cdots + \mathbf{A}B_{mm}\mathbf{x}_m = \mathbf{b}_m \end{cases}$$

Alternating Direction Solver – 2D

Factor out $\mathbf{A}$:

$$\begin{cases} \mathbf{A}(B_{11}\mathbf{x}_1 + B_{12}\mathbf{x}_2 + \cdots + B_{1m}\mathbf{x}_m) = \mathbf{b}_1 \\ \mathbf{A}(B_{21}\mathbf{x}_1 + B_{22}\mathbf{x}_2 + \cdots + B_{2m}\mathbf{x}_m) = \mathbf{b}_2 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ \mathbf{A}(B_{m1}\mathbf{x}_1 + B_{m2}\mathbf{x}_2 + \cdots + B_{mm}\mathbf{x}_m) = \mathbf{b}_m \end{cases}$$

Wy multiply by $\mathbf{A}^{-1}$ and define $\mathbf{y}^i = \mathbf{A}^{-1}\mathbf{b}^i$

$$\begin{cases} B_{11}\mathbf{x}_1 + B_{12}\mathbf{x}_2 + \cdots + B_{1m}\mathbf{x}_m = \mathbf{y}_1 \\ B_{21}\mathbf{x}_1 + B_{22}\mathbf{x}_2 + \cdots + B_{2m}\mathbf{x}_m = \mathbf{y}_2 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ B_{m1}\mathbf{x}_1 + B_{m2}\mathbf{x}_2 + \cdots + B_{mm}\mathbf{x}_m = \mathbf{y}_m \end{cases}$$

Alternating Direction Solver – 2D

Consider each component of $\mathbf{x}_i$ and $\mathbf{y}_i \Rightarrow$ family of linear systems

$$\begin{cases} B_{11}x^{1i} + B_{12}x^{2i} + \cdots + B_{1m}x^{mi} = y_{1i} \\ B_{21}x^{1i} + B_{22}x^{2i} + \cdots + B_{2m}x^{mi} = y_{2i} \\ \vdots \qquad\quad \vdots \qquad\quad \vdots \qquad\quad \vdots \\ B_{m1}x^{1i} + B_{m2}x^{2i} + \cdots + B_{mm}x^{mi} = y_{mi} \end{cases}$$

for each $i = 1, \ldots, n$

$\Rightarrow$ linear systems with matrix $\mathbf{B}$

Alternating Direction Solver – 2D

Two steps – solving systems with **A** and **B** in different *directions*

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & 0 \\ A_{21} & A_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{nn} \end{bmatrix} \begin{bmatrix} y_{11} & y_{21} & \cdots & y_{m1} \\ y_{12} & y_{22} & \cdots & y_{m1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1n} & y_{2n} & \cdots & y_{mn} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{21} & \cdots & b_{m1} \\ b_{12} & b_{22} & \cdots & b_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1n} & b_{2n} & \cdots & b_{mn} \end{bmatrix}$$

$$\begin{bmatrix} B_{11} & B_{12} & \cdots & 0 \\ B_{21} & B_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{mm} \end{bmatrix} \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix}$$

Two one dimensional problems with multiple RHS:

- $n \times n$ with m right hand sides $\rightarrow O(n * m) = O(N)$
- $m \times m$ with n right hand sides $\rightarrow O(m * n) = O(N)$

Linear computational cost $O(N)$

Isogeometric L2 projections

The computational cost of the solver is so low, that most of the time is spent on the integration

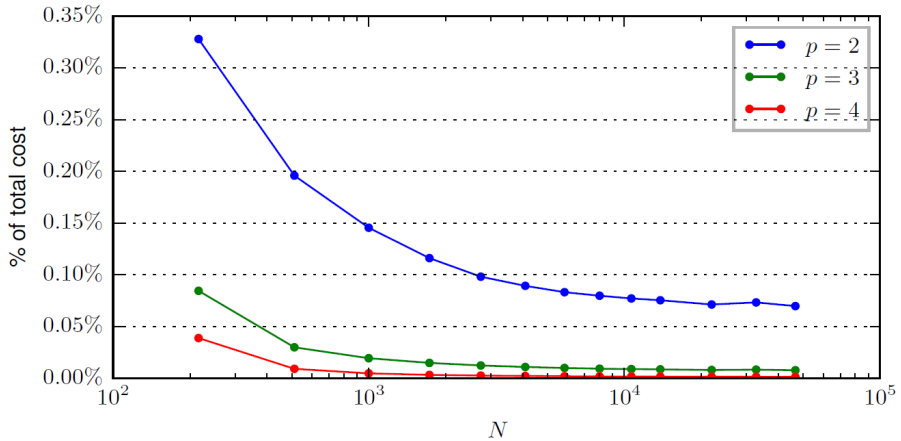


Figure: Time spent on integration with respect to time spent on factorization (below 1 percent of the total time for 2D problems, for all p and N)

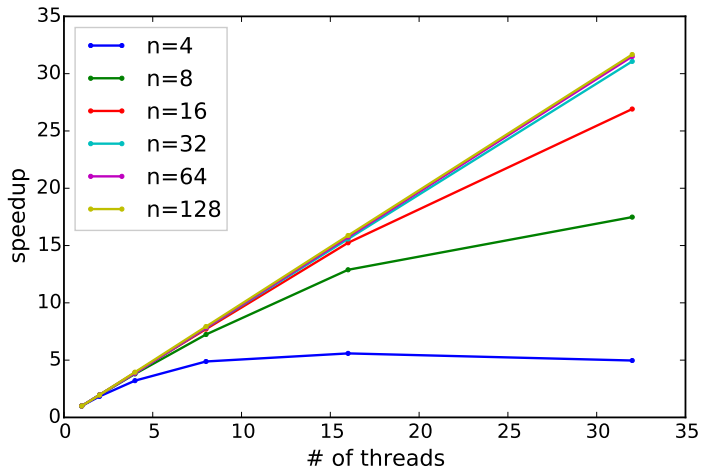
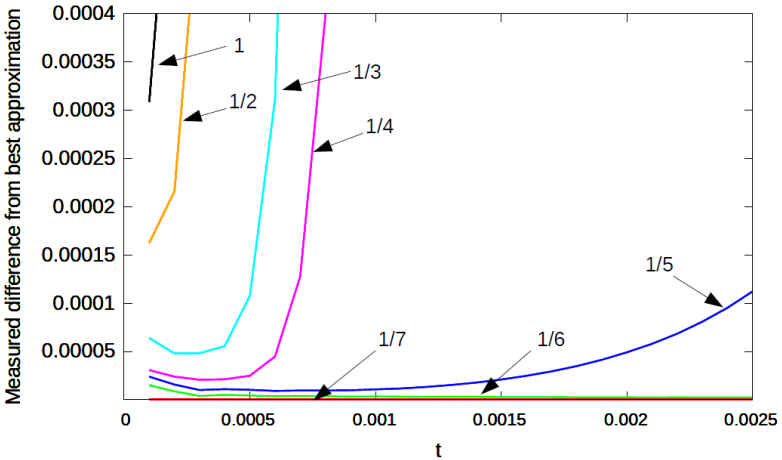


Figure: Speedup of parallel integration with GALOIS cubics, 2D problem different mesh sizes

Expensive isogeometric integration that can be speeded-up on multi-core machines

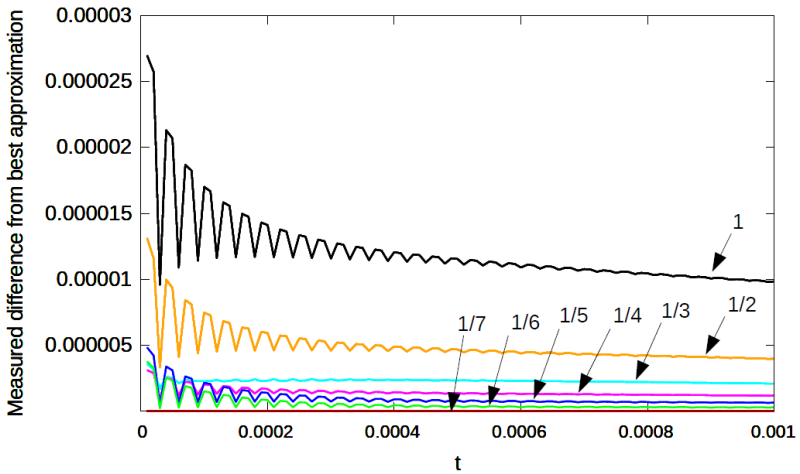
Isogeometric L2 projections

Time step size limited by Courant-Friedrichs-Levy (CFL) condition



Relative errors for the time step $Dt = 10^{-4}$

Figure: Lack of convergence for $Dt = 10^{-4}, \frac{10^{-4}}{2}, \ldots, \frac{10^{-4}}{5}$



Relative errors for the time step $Dt = 10^{-5}$

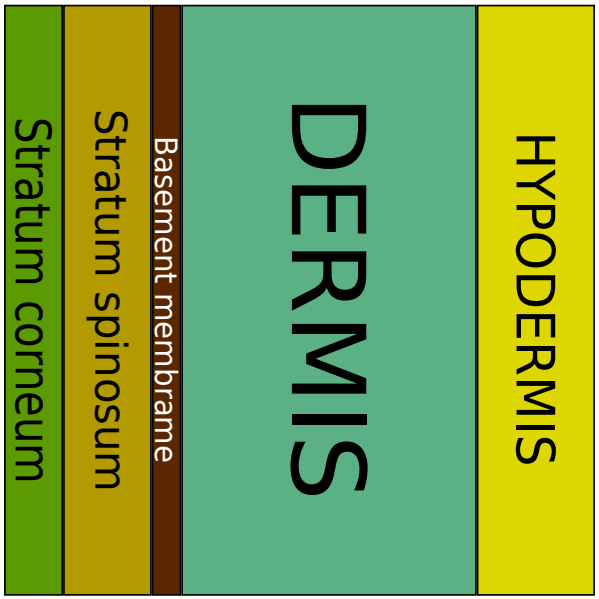
Figure: Convergence for $Dt = 10^{-5}$ and smaller time steps

Melanoma growth model

Hybrid approach – two components:

- continuous – concentration of various substances
  - cancer cells
  - extracellular matrix
  - tumor angiogenic factor (TAF)
- discrete – vasculature model
  - vasculature evolution
  - oxygen distribution

Domain – skin layers



Different tumor cell diffusion coefficient P_b:

- stratum corneum – $P_b = 0.05$
- stratum spinosum – $P_b = 0.3$
- basement membrame – $P_b = 0.002$
- dermis – $P_b = 0.15$
- hypodermis – $P_b = 0.05$

Tumor cell density

Tumor cell density – b

- main quantity of interest
- values between $b^m = 0$ (no cancer cells) and $b^M = 2$
- $b^N = 1$ – *normal* tumor cell density

$$\frac{\partial b}{\partial t} = -\nabla \cdot J + b^- + b^+$$

- b^+, b^- – tumor cell proliferation and apopthosis factors
- J – tumor cell flux

Tumor cell proliferation/death

b^+, b^- – governed by the oxygen concentration o

- $o > o^{prol}$ – tumor cells multiply ($b^+ > 0$)
- $o < o^{death}$ – tumor cells die ($b^- > 0$)

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

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

Tumor cell flux

J – induced by pressure of tumor and extracellular matrix

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

where

- P – tumor pressure, present for tumor cell density exceeding b^N

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

- A – (degraded) extracellular matrix
- D_b – cell diffusion coefficient

Extracellular matrix (ECM)

- provides support for the cell structures
- can be degraded by tumor cells

$$\frac{\partial M}{\partial t} = -\beta_M M b$$
$$\frac{\partial A}{\partial t} = \gamma_A M b + \chi_{aA} \Delta A - \gamma_{oA} A$$

where

- M – ECM density
- A – degraded ECM density

Tumor angiogenic factor (TAF)

- produced by oxygen-starved tumor cells
- signal to the vasculature – „more oxygen is needed here”
- influences vasculature evolution (discrete model)

$$\frac{\partial c}{\partial t} = \chi_c \Delta c - \gamma_c o\, c + c^+$$

where

- c – TAF concentration
- o – oxygen concentration
- c^+ – TAF production rate

$$c^+ = b(1 - c) \quad \text{for } o < o^{death}$$

Discrete model – vasculature

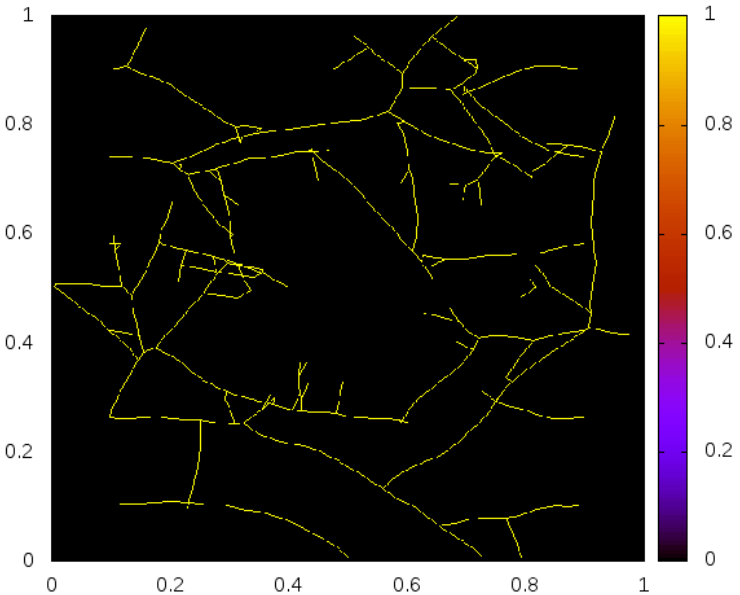
- network of vessels distributing oxygen to the cells
- coupled with the continuous model
  - oxygen concentration influences tumor cell development
  - TAF concentration influences vasculature evolution
- model – graph embedded in the domain
- vasculature evolution processes modifies the graph
  - sprout creation
  - sprout migration
  - degradation

Vasculature updated every 10 time steps of the continuous model

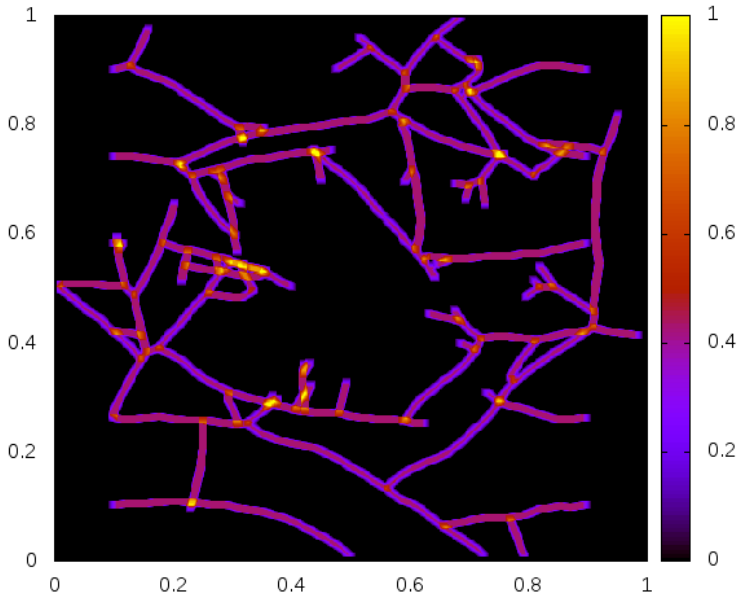
Based on: M. Welter, H. Rieger, *Physical determinants of vascular network remodeling during tumor growth*, The European Physical Journal E, 33(2), 149-163 (2010)

Oxygen distribution

Oxygen is concentrated in the vicinity of the vessels



(a) vasculature



(b) oxygen concentration

Sprout creation/migration

Creation

- new vessels are created by attaching *sprouts* to existing nodes
- sprout can be created at each node where TAF exceeds c_{min}
- sprout is created with probability $\Delta t / t^{sprout}$

Migration

- sprout expands until it merges with an existing vessel
- sprout grows in the direction of TAF source: $-\nabla c$

Numerical formulation

Forward Euler time discretization:

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

Spatial approximation – L^2-projections
Approximation space spanned by basis functions $B_1, \ldots, B_n$

$$\begin{cases} (b_{t+1}, B_i)_{L^2} = (b_t, B_i)_{L^2} + \Delta t \left(-\nabla \cdot J_t + b_t^- + b_t^+, B_i\right)_{L^2} \\ (c_{t+1}, B_i)_{L^2} = (c_t, B_i)_{L^2} + \Delta t \left(\chi_c \Delta c_t - \gamma_c o_t c_t + c_t^+, B_i\right)_{L^2} \\ (M_{t+1}, B_i)_{L^2} = (M_t, B_i)_{L^2} + \Delta t \left(-\beta_M M_t b_t, B_i\right)_{L^2} \\ (A_{t+1}, B_i)_{L^2} = (A_t, B_i)_{L^2} + \Delta t \left(\gamma_A M_t b_t + \chi_{OA} \Delta A_t - \gamma_{OA} A_t, B_i\right)_{L^2} \end{cases}$$

$u = (b, c, M, A) \Rightarrow (u_{t+1}, B_i)_{L^2} = (u_t, B_i)_{L^2} + F(u_t, B_i)$

Integration loop – sequential version

for *each element* $E = [\xi_{l_x}, \xi_{l_x+1}] \times \left[\xi_{l_y}, \xi_{l_y+1}\right] \times [\xi_{l_z}, \xi_{l_z+1}]$ **do**
  for *each quadrature point* $\boldsymbol{\xi} = \left(X_{k_x}, X_{k_y}, X_{k_z}\right)$ **do**
    $\mathbf{x} \leftarrow \Psi_E(\boldsymbol{\xi})$;
    $W \leftarrow w_{k_x} w_{k_y} w_{k_z}$;
    $u, Du \leftarrow 0$;
    for $I \in \mathcal{I}(E)$ **do**
      $u \leftarrow u + U_I^{(t)} \mathcal{B}_I(\boldsymbol{\xi})$;
      $Du \leftarrow Du + U_I^{(t)} \nabla \mathcal{B}_I(\boldsymbol{\xi})$;
    end
    for $I \in \mathcal{I}(E)$ **do**
      $v \leftarrow \mathcal{B}_I(\boldsymbol{\xi})$;
      $Dv \leftarrow \nabla \mathcal{B}_I(\boldsymbol{\xi})$;
      $U_I^{(t+1)} \leftarrow U_I^{(t+1)} + W\,|E|\,(uv + \Delta t\, F(u, Du, v, Dv))$
    end
  end
end

Each element – independent computation
except for updating $U^{(t+1)}$ – shared state

- localize state, update once atomically
- execute element computations in parallel

Integration loop – parallel version

for *each element* $E = [\xi_{l_x}, \xi_{l_x+1}] \times \left[\xi_{l_y}, \xi_{l_y+1}\right] \times [\xi_{l_z}, \xi_{l_z+1}]$ *in parallel* **do**
  $U^{loc} \leftarrow 0$;
  for *each quadrature point* $\boldsymbol{\xi} = \left(X_{k_x}, X_{k_y}, X_{k_z}\right)$ **do**
    $\mathbf{x} \leftarrow \Psi_E(\boldsymbol{\xi})$;
    $W \leftarrow w_{k_x} w_{k_y} w_{k_z}$;
    $u, Du \leftarrow 0$;
    for $I \in \mathcal{I}(E)$ **do**
      $u \leftarrow u + U_I^{(t)} \mathcal{B}_I(\boldsymbol{\xi})$;
      $Du \leftarrow Du + U_I^{(t)} \nabla \mathcal{B}_I(\boldsymbol{\xi})$;
    end
    for $I \in \mathcal{I}(E)$ **do**
      $v \leftarrow \mathcal{B}_I(\boldsymbol{\xi})$;
      $Dv \leftarrow \nabla \mathcal{B}_I(\boldsymbol{\xi})$;
      $U_I^{loc} \leftarrow U_I^{loc} + W\,|E|\,(uv + \Delta t\, F(u, Du, v, Dv))$;
    end
  end
  synchronized
    for $I \in \mathcal{I}(E)$ **do**
      $U_I^{(t+1)} \leftarrow U_I^{(t+1)} + U_I^{loc}$
    end
  end
end

Implementation: `Galois::for_each`, `Galois::Runtime::LL::SimpleLock`

Numerical results

Initial state:

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

Parameters:

- 80×80 elements
- quadratic B-splines ($p = 2$)
- $\Delta t = 10^{-3}$
- 30,000 time steps
- 8 hours of sequential simulation (around 1s / time step)
- around 40 minutes (12 times faster)
  with parallel GALOIS solver on 16 cores

IGA-FEM simulation (1/3) TAF

Click in the middle

IGA-FEM simulation (2/3) Vasculature

Click in the middle

IGA-FEM simulation (3/3) Tumor

Click in the middle

Conclusions

- Isogeometric L2 projections applied for explicit solver of tumor growth
- 8 hours of sequential simulation (1 second per time step)
- The integration is almost perfectly parallelizable
- Time step size limited by CFL condition
  (may be a problem in 3D)
- Crank-Nicolson may be neccesary in 3D
  (direct solver with rIGA)

Current and future work

- GPGPU accelerators
- 3D melanoma growth simulations
- Smart average between different tumor growth models (supermodeling)
- Release of the open source parallel GALOIS based isogoemetric L2 projection package for explicit dynamics

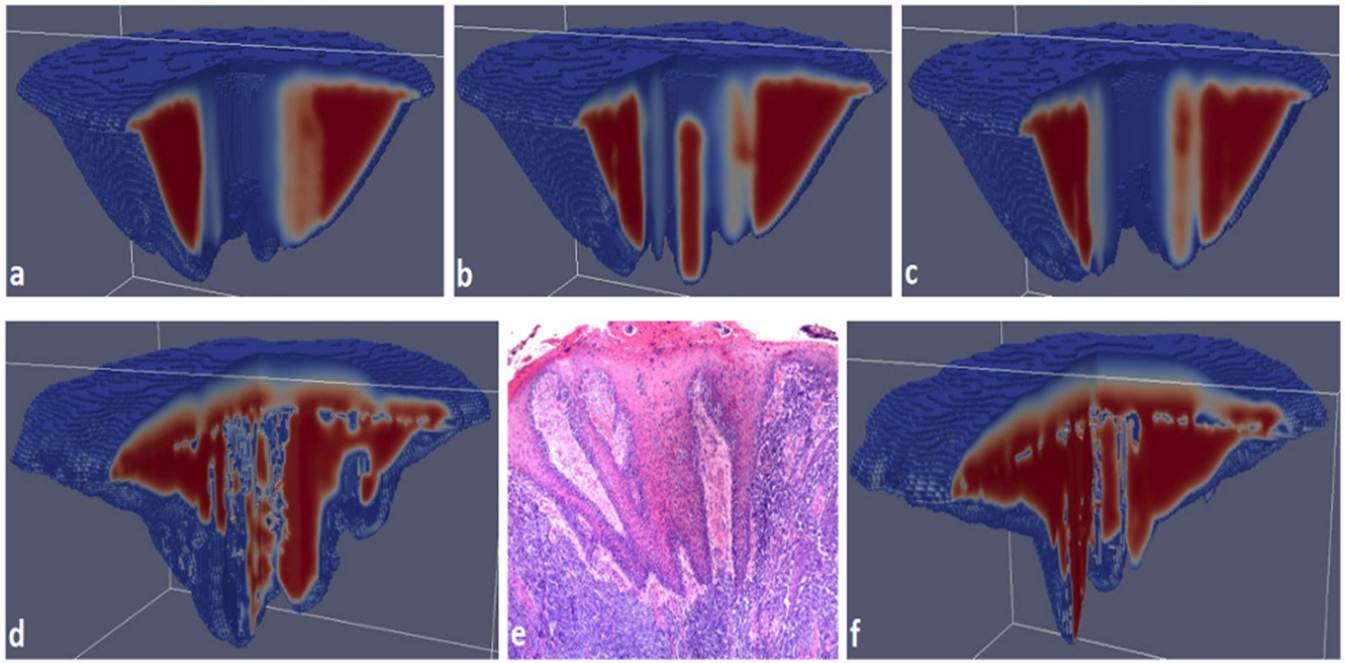
Marcin Łos, Maciej Woźniak, Maciej Paszyński, Andrew Lenharth, Keshav Pingali *IGA-ADS : Isogeometric Analysis FEM using ADS solver*, to be submitted to **Computer Physics Communications** (2016)

- Adding adaptation to the alternating direction solver
- Addaptive dealing with CFL condition
- Application of rIGA ideas to ADI
- Extension to Crank-Nicolson type implicit schemes seems not possible so far

Thank you for attention

Questions...?

Verification of the supermodel



Picture obtained from prof. Witold Dzwinel
(3D finite difference + discrete models)
Supermodeling =
smart average from several simulations with different parameters

refined Isogeometric Analysis (rIGA)

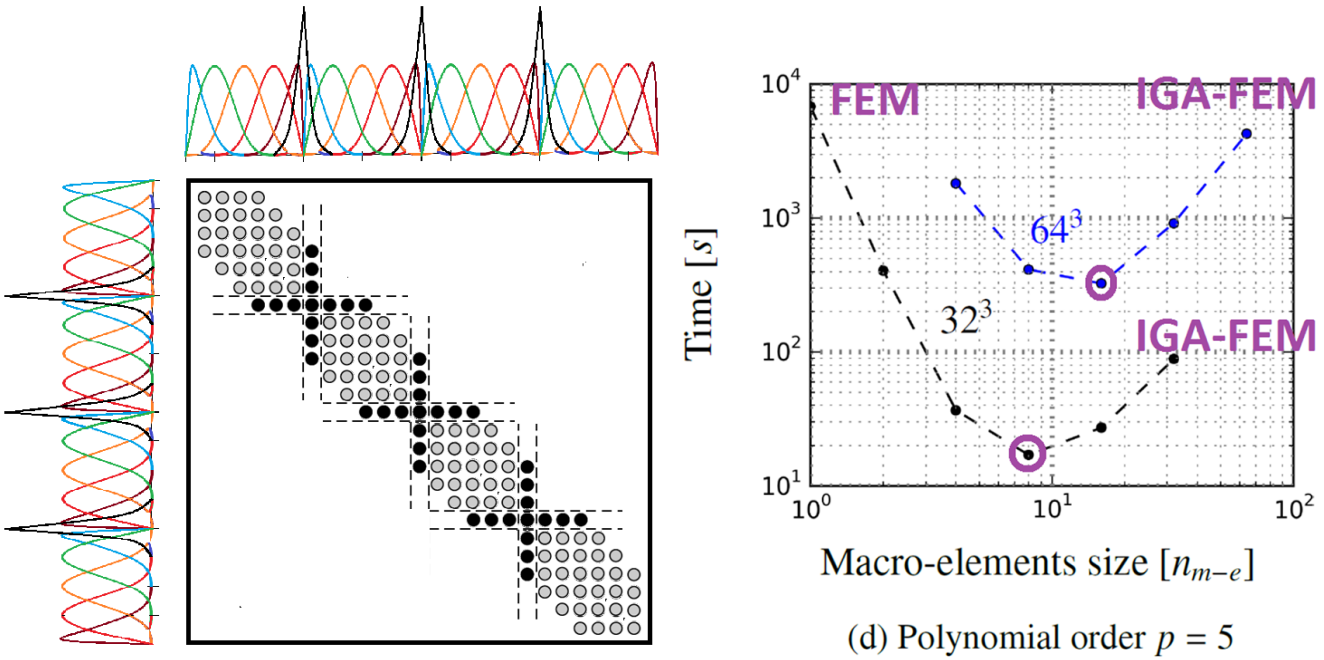


Figure: 1D intuition (left panel); 3D example (right panel)

Daniel Garcia, David Pardo, Lisandro Dalcin, Maciej Paszynski, Victor M. Calo, *Refined Isogeometric Analysis (rIGA): Fast Direct Solvers by Controlling Continuity*, accepted to **Computer Methods in Applied Mechanics and Engineering** (2016)