BOUNDARY CONDITIONS AND PARTICLES
Boundary conditions and particle methods: a non-standard issue, because particles make sense only as a collection of overlapping points.

A single particle on a boundary not enough to enforce a given boundary condition at that point.
IMPORTANT to allow the flow to:

Boundary conditions and particle methods: a non-standard issue, because particles make sense only as a collection of overlapping points.

A single particle on a boundary not enough to enforce a given boundary condition at that point.
**IMPORTANT** to allow the flow to:

- leave the computational box without non-physical artifacts \(\text{(numerical issue)}\)

Boundary conditions and particle methods: a non-standard issue, because particles make sense only as a collection of overlapping points.

A single particle on a boundary not enough to enforce a given boundary condition at that point.
IMPORTANT to allow the flow to:

- leave the computational box without non-physical artifacts (numerical issue)
- inject material in computational box

Boundary conditions and particle methods: a non-standard issue, because particles make sense only as a collection of overlapping points.

A single particle on a boundary not enough to enforce a given boundary condition at that point.
IMPORTANT to allow the flow to:

- leave the computational box without non-physical artifacts (numerical issue)
- inject material in computational box
- generate vorticity and to impart forces at interfaces/solid boundaries (fluid-structure interaction, physical and numerical issue)

Boundary conditions and particle methods: a non-standard issue, because particles make sense only as a collection of overlapping points.

A single particle on a boundary not enough to enforce a given boundary condition at that point.
BCs using Ghost Particles

SPH methods often use the concept of ghost particles to overcome this difficulty.

Example where one wants to enforce zero flow at the boundary by using ghost particles:

\[ h \]

\[ h \] is the radius of the regularization kernel which allows to recover local values from particle strengths.
BCs using **Ghost Particles**

SPH methods often use the concept of ghost particles to overcome this difficulty.

Example where one wants to enforce zero flow at the boundary by using ghost particles:

**DIFFICULTY:** accurate definitions of ghost particles need local mappings around interface onto half-space geometries

$h$ is the radius of the regularization kernel which allows to recover local values from particle strengths.
The case of vortex methods for incompressible flows even more delicate, because vorticity boundary values in general not known.
The case of vortex methods for incompressible flows is even more delicate, because the vorticity boundary values in general are not known. Like for all numerical methods, one can deal with boundary conditions in two ways:
The case of vortex methods for incompressible flows even more delicate, because vorticity boundary values in general not known.

Like for all numerical methods, can deal with boundary conditions in two ways:

• using body-fitted grids (the boundary is made of specific points of a given grid used to solve for the flow)
The case of vortex methods for incompressible flows even more delicate, because vorticity boundary values in general not known.

Like for all numerical methods, can deal with boundary conditions in two ways:

- **using body-fitted grids** (the boundary is made of specific points of a given grid used to solve for the flow)
- **seeing the boundary as an immersed boundary**
Vorticity BCs for No-slip Incompressible Flows

Boundary conditions appear at two levels:

- **KINEMATICS**: velocity from vorticity: \( \text{div } \mathbf{u} = 0, \ \text{curl } \mathbf{u} = \omega \) in \( \Omega \) and \( \mathbf{u} \cdot \mathbf{n} = 0 \) on \( \partial \Omega \) (no-through condition)

- **DYNAMICS**: advection-diffusion equation for vorticity:
  \[
  \omega_t + (\mathbf{u} \cdot \nabla) \omega = (\omega \cdot \nabla) \mathbf{u} + \nu \Delta \omega \quad \text{in } \Omega , \ \omega = ? \ \text{on } \partial \Omega
  \]
Kinematic Boundary Condition

Classical way to deal with the first boundary condition \((\mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \partial \Omega)\) is to look for a decomposition of the velocity field into a rotational and a potential part:

\[
\mathbf{u} = \nabla \varphi + \nabla \times \psi
\]

IN PRACTICE: compute first \(\omega\), without bothering about boundary conditions, then fix boundary conditions with \(\varphi\)
Kinematic Boundary Condition

Classical way to deal with the first boundary condition \((\mathbf{u}.\mathbf{n} = 0 \text{ on } \partial\Omega)\) is to look for a decomposition of the velocity field into a rotational and a potential part:

\[
\mathbf{u} = \nabla \varphi + \nabla \times \psi
\]

\[\text{div } \mathbf{u} = 0 \Rightarrow \Delta \varphi = 0 \text{ in } \Omega\]

IN PRACTICE: compute first \(\mathbf{\omega}\), without bothering about boundary conditions, then fix boundary conditions with \(\varphi\)
Kinematic Boundary Condition

Classical way to deal with the first boundary condition \((u \cdot n = 0\) on \(\partial \Omega\)) is to look for a decomposition of the velocity field into a rotational and a potential part:

\[
 u = \nabla \varphi + \nabla \times \psi
\]

\[
 \text{div } u = 0 \Rightarrow \Delta \varphi = 0 \text{ in } \Omega
\]

\[
 \nabla \times u = \omega \Rightarrow \Delta \psi = \omega , \text{div } \psi = 0 \text{ in } \Omega.
\]

IN PRACTICE: compute first \(\omega\), without bothering about boundary conditions, then fix boundary conditions with \(\varphi\).
Kinematic Boundary Condition

Classical way to deal with the first boundary condition \((u \cdot n = 0 \text{ on } \partial \Omega)\) is to look for a decomposition of the velocity field into a rotational and a potential part:

\[
\mathbf{u} = \nabla \varphi + \nabla \times \mathbf{\psi}
\]

\[\text{div } \mathbf{u} = 0 \implies \Delta \varphi = 0 \text{ in } \Omega \]

\[\nabla \times \mathbf{u} = \mathbf{\omega} \implies \Delta \mathbf{\psi} = \mathbf{\omega} \text{, div } \mathbf{\psi} = 0 \text{ in } \Omega.\]

Boundary Condition \(u \cdot n = 0\) gives \(\partial \varphi / \partial n = -\partial (\nabla \times \mathbf{\psi}) / \partial n\) on \(\partial \Omega\)

**IN PRACTICE**: compute first \(\mathbf{\omega}\), without bothering about boundary conditions, then fix boundary conditions with \(\varphi\).
Kinematic Boundary Condition

Classical way to deal with the first boundary condition ($\mathbf{u} \cdot \mathbf{n} = 0$ on $\partial \Omega$) is to look for a decomposition of the velocity field into a rotational and a potential part:

$$\mathbf{u} = \nabla \varphi + \nabla \times \psi$$

$$\text{div } \mathbf{u} = 0 \Rightarrow \Delta \varphi = 0 \text{ in } \Omega$$

$$\nabla \times \mathbf{u} = \omega \Rightarrow \Delta \psi = \omega, \text{ div } \psi = 0 \text{ in } \Omega.$$ 

Boundary Condition $\mathbf{u} \cdot \mathbf{n} = 0$ gives $\partial \varphi / \partial n = -\partial (\nabla \times \psi) / \partial n$ on $\partial \Omega$

**IN PRACTICE**: compute first $\omega$, without bothering about boundary conditions, then fix boundary conditions with $\varphi$.

In a grid-free vortex method, this results in:

$$\mathbf{u}(x) = \int_{\Omega_f} K(x - y) \omega(y) \, dy + \int_{\partial \Omega} \nabla G(x - y) q(y) \, dy$$

where $q$ is a potential to be determined from an integral equation on $\partial \Omega$.
Next, enforce that *tangential velocities are also zero* at the boundary.

**traditional** numerical recipe for vortex methods *mimics the physical mechanism*: vorticity produced at the boundary to prevent any slip velocity at the boundary.
Next, enforce that *tangential velocities are also zero* at the boundary.

**Traditional** numerical recipe for vortex methods *mimics the physical mechanism*: vorticity produced at the boundary to prevent any slip velocity at the boundary.

---

**FRACTIONAL STEP ALGORITHM** [Chorin 1978]:

---
Next, enforce that \textit{tangential velocities are also zero} at the boundary.

\textbf{traditional} numerical recipe for vortex methods \textit{mimics the physical mechanism}: vorticity produced at the boundary to prevent any slip velocity at the boundary.

\textbf{FRACTIONAL STEP ALGORITHM} [Chorin 1978]:
1) first substep without vorticity creation.
Next, enforce that *tangential velocities are also zero* at the boundary.

**Traditional** numerical recipe for vortex methods *mimics the physical mechanism*: vorticity produced at the boundary to prevent any slip velocity at the boundary.

**FRACTIONAL STEP ALGORITHM** [Chorin 1978]:

1) first substep without vorticity creation
2) compute resulting slip
Next, enforce that \textit{tangential velocities are also zero} at the boundary.

\textbf{traditional} numerical recipe for vortex methods \textit{mimics the physical mechanism}: vorticity produced at the boundary to prevent any slip velocity at the boundary.

\textbf{FRACTIONAL STEP ALGORITHM} \cite{Chorin1978}:

1) first substep without vorticity creation
2) compute resulting slip
3) remove this slip by injecting in the flow the appropriate sheet of vorticity
DYNAMICS - Lighthill’s Algorithm

Next, enforce that tangential velocity are also zero at the boundary.

**Traditional** numerical recipe for vortex methods *mimics the physical mechanism*: vorticity produced at the boundary to prevent any slip velocity at the boundary.

[Koumoutsakos-Leonard 1992]
Next, enforce that tangential velocity are also zero at the boundary

**traditional** numerical recipe for vortex methods mimics the physical mechanism: vorticity produced at the boundary to prevent any slip velocity at the boundary

\[ \frac{\partial \omega}{\partial t} - \nu \Delta \omega = 0 \]

\[ \nu \frac{\partial \omega}{\partial n} = - \frac{1}{\Delta t} u \cdot \tau \partial \Omega \]

[Koumoutsakos-Leonard 1992]
### 3D Vorticity Flux BCs

In 3D, need boundary conditions for 3 vorticity components

After advection step computation of slip

**Vorticity flux onto flow particles**

$$\begin{align*}
\frac{\partial \omega}{\partial t} - \nu \Delta \omega &= 0 \quad \text{sur } \Omega \times ]t_0, t_0 + \delta t[
\\
\omega(t_0) &= 0 \quad \text{sur } \Omega
\\
\nu \frac{\partial \omega_z}{\partial n} &= - \frac{\partial u_\theta}{\partial t} \quad \text{sur } \partial \Omega \times ]t_0, t_0 + \delta t[
\\
\nu \left( \frac{\partial \omega_\theta}{\partial n} + \kappa \omega_\theta \right) &= - \frac{\partial u_z}{\partial t} \quad \text{sur } \partial \Omega \times ]t_0, t_0 + \delta t[
\\
\omega_r &= 0 \quad \text{sur } \partial \Omega \times ]t_0, t_0 + \delta t[
\end{align*}$$

Case of flow past a cylinder [Cottet-Poncet 2003]
3D Vorticity Flux BCs

In 3D, need boundary conditions for 3 vorticity components

After advection step computation of slip

Vorticity flux onto flow particles

\[
\begin{align*}
\frac{\partial \omega}{\partial t} - \nu \Delta \omega &= 0 & \text{sur } \Omega \times ]t_0, t_0 + \delta t[ \\
\omega(t_0) &= 0 & \text{sur } \Omega \\
\nu \left( \frac{\partial \omega}{\partial n} + \kappa \omega \right) &= -\frac{\partial u_z}{\partial t} & \text{sur } \partial \Omega \times ]t_0, t_0 + \delta t[ \\
\omega_r &= 0 & \text{sur } \partial \Omega \times ]t_0, t_0 + \delta t[ 
\end{align*}
\]

difficulty: requires local coordinate system on the surface

case of flow past a cylinder [Cottet-Poncet 2003]
Obstacles, walls, objects .. are part of the flow

The case of a rigid body with prescribed velocity
Obstacles, walls, objects .. are part of the flow

• rigid body with prescribed velocity or interacting with flow

The case of a rigid body with prescribed velocity
Obstacles, walls, objects .. are part of the flow

- rigid body with prescribed velocity or interacting with flow
- elastic membrane

The case of a rigid body with prescribed velocity
Obstacles, walls, objects .. are part of the flow

- rigid body with prescribed velocity or interacting with flow
- elastic membrane
- visco-elastic body ..

The case of a rigid body with prescribed velocity
The case of a rigid body with prescribed velocity
COUPLING AND BOUNDARY CONDITIONS

Atomistic:
Molecular Dynamics

\[ m \frac{d^2 r}{dt^2} = F \]

Continuum:
Navier-Stokes Eqs.

\[ \frac{Du}{Dt} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 u \]
\[ \frac{D\rho}{Dt} = \rho \nabla \cdot u \]

Boundary Conditions

+ Fa
+ Fc

\[ m : \text{mass} \quad r : \text{position} \quad F : \text{force} \]

\[ u : \text{velocity} \quad P : \text{pressure} \quad \rho : \text{density} \quad \nu : \text{viscosity} \]
Boundary Conditions = Coupling

\[ \rho \frac{Du}{Dt} = \nabla \cdot \sigma + f \text{ (enforces b.c.)} \]
Boundary Conditions = Coupling

\[ \rho \frac{Du}{Dt} = \nabla \cdot \sigma + f \text{ (enforces b.c.)} \]
Boundary Conditions = Coupling

\[ \rho \frac{Du}{Dt} = \nabla \cdot \sigma + f \text{ (enforces b.c.)} \]
Penalization Method: \[ f(x) = \lambda \chi_S (u_S - u) \]
I. IMMERSED BOUNDARY METHOD for SPH

- Enforce boundary velocity by a bodyforce $f$ in Momentum Equation

$$\rho \ \frac{Du}{Dt} = -\nabla p + \nabla \tau + f$$

$$= f_{i,\text{part}} + f_{i,\text{boundary}}$$
I. IMMERSED BOUNDARY METHOD for SPH

- Enforce boundary velocity by a bodyforce $f$ in Momentum Equation

$$\rho \frac{Du}{Dt} = -\nabla p + \nabla \cdot \tau + f$$

- Approximate Material Derivative at time step $i$ and solve for $f$

$$\rho_i \frac{u_{i+1} - u_i}{\Delta t} = -\nabla p_i + \nabla \cdot \tau_i + f_i \Rightarrow f_i = \rho_i \frac{u_{i+1} - u_i}{\Delta t} - (-\nabla p_i + \nabla \cdot \tau_i)$$

- Desired Velocity field on the boundary

$$u_{i+1} = u_{\text{desired}}$$

$$u_{i+1} = u_{\text{desired}} \Rightarrow f_i = \rho_i \frac{u_{\text{desired}} - u_i}{\Delta t} - (-\nabla p_i + \nabla \cdot \tau_i) = f_{i,\text{part}} + f_{i,\text{boundary}}$$
Boundary Conditions: A Particle-Mesh Operation
Boundary Conditions: A Particle-Mesh Operation

- Compute part of forcing term on the particles

\[ f_{i, \text{part}} = \rho_i \frac{-u_i}{\Delta t} - (- \nabla p_i + \nabla \tau_i) \]
Boundary Conditions: A Particle-Mesh Operation

- Compute part of forcing term on the particles

\[ f_{i,\text{part}} = \rho_i \frac{-u_i}{\Delta t} - (-\nabla p_i + \nabla \tau_i) \]

- Particles to Boundary (Particle to Mesh Interpolation)

\[ f_{i,\text{boundary}} = \rho_i \frac{u_{\text{desired}}}{\Delta t} + f_{i,\text{part,interpolated}} \]
Boundary Conditions: A Particle-Mesh Operation

- Compute part of forcing term on the particles

\[ f_{i,\text{part}} = \rho_i \frac{-u_i}{\Delta t} - \left( -\nabla p_i + \nabla \tau_i \right) \]

- Particles to Boundary (Particle to Mesh Interpolation)

\[ f_{i,\text{boundary}} = \rho_i \frac{u_{\text{desired}}}{\Delta t} + f_{i,\text{part,interpolated}} \]

- Force \(\rightarrow\) Boundary to Particles (Mesh - Particle Interpolation)
Boundary Conditions: A Particle-Mesh Operation

- Compute part of forcing term on the particles

\[ f_{i, \text{part}} = \rho_i \frac{-u_i}{\Delta t} - (-\nabla p_i + \nabla \tau_i) \]

- Particles to Boundary (Particle to Mesh Interpolation)

\[ f_{i, \text{boundary}} = \rho_i \frac{u_{\text{desired}}}{\Delta t} + f_{i, \text{part}, \text{interpolated}} \]

- Force -> Boundary to Particles (Mesh - Particle Interpolation)

Lattice Boltzmann and Impulsively Started Cylinders

Results on Swimming

Finite Volume  (Kern & Koumoutsakos, J. Exp. Biology, 2007)

Particle + IBM  (Hieber & Koumoutsakos, JCP, 2008)

Longitudinal and lateral velocity
Results on Swimming

Finite Volume  (Kern & Koumoutsakos, J. Exp. Biology, 2007)

Particle + IBM  (Hieber & Koumoutsakos, JCP, 2008)

Longitudinal and lateral velocity
Boundary Conditions = Coupling

\[ \rho \frac{Du}{Dt} = \nabla \cdot \sigma + f \text{ (enforces b.c.)} \]

Penalization: \[ f(x) = \lambda \chi_S (u_S - u) \]

Immersed Boundary: \[ f(x) = \kappa \delta_S (x_S - x) \]


FISH SCHOOLING

Monday, July 23, 12
Penalization Method: \( f(x) = \lambda \chi_S (u_S - u) \)

Immersed Boundary Method: \( f(x) = \kappa \delta_S (x_S - x) \)
Boundary Conditions = Coupling

Penalization Method: \( f(x) = \lambda \chi_S(u_S - u) \)

Immersed Boundary Method: \( f(x) = \kappa \delta_S(x_S - x) \)

\[ \rho \frac{Du}{Dt} = \nabla \cdot \sigma + f \text{ (enforces b.c.)} \]
Re 9500: Multiresolution + Multicores + (multi)GPUs
Benchmark: The Impulsively Started Cylinder
Benchmark: The Impulsively Started Cylinder

Re=15000 on [1,4]x[0,π] – global formula

- analytic
- 2048x4096 nofgp
- 4096x4096 nofgp
- 2048x2048 fgp
- 4096x4096 fgp
- petros

CD vs T graph
Lattice Boltzmann + Immersed Boundaries

Re = 550

Lattice Boltzmann + Immersed Boundaries

Re = 550

Multiphysics/Multiscale

\[
\rho \frac{Du}{Dt} = \nabla \cdot \sigma + f \text{ (enforces b.c.)}
\]

Boundary Conditions = Coupling

Multiphysics/Multiscale

\[
f(x) \approx F \text{(Atomistic Simulations)}
\]
Boundary Conditions = Coupling

\[ \rho \frac{Du}{Dt} = \nabla \cdot \sigma + f \text{ (enforces b.c.)} \]

Multiphysics / Multiscale

\[ f(\mathbf{x}) \approx F \text{(Atomistic Simulations)} \]
Schwarz DD for Liquids

Iterate, until the solution in the overlap region converges.

Conservative scheme - transport coefficients in A and C match.
Bridging FLUX & SCHWARZ DD Algorithms


Diagram:
- Overlap domain
- Imposition of the velocity from MD to continuum (VG-coupling)
- MD subdomain
- Imposition of the velocity from MD to continuum (V-coupling)
- LB domain
- MD domain = MD subdomain + Overlap domain
- $\delta_s$
Bridging FLUX & SCHWARZ DD Algorithms


Overlap domain

Imposition of the velocity from MD to continuum (VG-coupling)

MD subdomain

Imposition of the velocity from MD to continuum (V-coupling)

LB domain

MD domain = MD subdomain + Overlap domain

\( \delta_s \)

Imposition of the velocity from continuum to MD
Bridging FLUX & SCHWARZ DD Algorithms


- Overlap domain
- MD subdomain
- LB domain
- MD domain = MD subdomain + Overlap domain

Imposition of the velocity from MD to continuum (V-coupling)
Imposition of the velocity from continuum to MD

$\delta_s$
Bridging FLUX & SCHWARZ DD Algorithms


**Diagram Description**

- **Overlap domain**
- **MD subdomain**
- **LB domain**

- Imposition of the velocity from MD to continuum (VG-coupling)
- Imposition of the velocity from MD to continuum (V-coupling)
- Imposition of the velocity from continuum to MD

**Equation**

MD domain = MD subdomain + Overlap domain
TEST 1 : EQUILIBRIUM  Non-Periodic MD

Boundary force

Specular wall
How can we account for the particles in the red domain?
How can we account for the particles in the red domain?

Take fluid structure into account: \( g(r) \)

\[
\rho(r) = \int_0^r 4\pi r'^2 \rho g(r') \, dr'
\]

\[
F_m(z) = -2\pi \rho \int g(r) \frac{\partial U(r)}{\partial r} \frac{z}{r} \, x \, dx \, dz_{\text{red}}
\]
A comparison of Forces

No force

Uniform distribution (O’Connell 1995)

Repulsive (Nie et al. 2004)

Use fluid structure (Werder et al. 2005)


A comparison of Forces


A comparison of Forces

- No force
- Uniform distribution (O’Connell 1995)
- Repulsive (Nie et al. 2004)
- Use fluid structure (Werder et al. 2005)


A comparison of Forces

- No force
- Uniform distribution (O’Connell 1995)
- Repulsive (Nie et al. 2004)
- Use fluid structure (Werder et al. 2005)


A comparison of Forces

- No force
- Uniform distribution (O’Connell 1995)
- Repulsive (Nie et al. 2004)
- Use fluid structure (Werder et al. 2005)


A comparison of Forces

- No force
- Uniform distribution (O’Connell 1995)
- Repulsive (Nie et al. 2004)
- Use fluid structure (Werder et al. 2005)


A comparison of Forces


A comparison of Forces

- **No force**
- **Uniform distribution (O’Connell 1995)**
- **Repulsive (Nie et al. 2004)**
- **Use fluid structure (Werder et al. 2005)**

---


A comparison of Forces

No force

Uniform distribution (O’Connell 1995)

Repulsive (Nie et al. 2004)

Use fluid structure (Werder et al. 2005)


MD vs Hybrid scheme

Hybrid solution

Reference MD solution

The hybrid scheme is \( \sim (L/R)^3 \) times faster for a computational domain of size \( L \) and a MD subdomain of size \( R \).

Relative Error \( \sim 1.3\% \)

Errors for Different Couplings

- V
- MD
- VG
- MD - V
- MD - VG
The problem with density variations

- Density variations depend on liquid state
- Amplitude proportional to **structural correlations** in the liquid

\[ T = 84K, \rho = 1.5 \, g/cm^3 \]
\[ T = 131K, \rho = 1.35 \, g/cm^3 \]
\[ T = 215K, \rho = 1.0 \, g/cm^3 \]
• Controlling of the external boundary force

• measured density $\rho^m \Rightarrow$ target density $\rho^t$

\[ e(r) = \rho^t(r) - \rho^m(r) \]

\[ \text{Force} = k \nabla e(r) \]
Results with Control Approach I

- at equilibrium (no flow)
- $T = 84K$, $\rho = 1.5 g cm^{-3}$

AC interface = Elastic Boundary + External Force

Controller deduces the boundary force

\[
\text{Controller deduces the boundary force}
\]
- uniform flow
- $T = 131K, \rho = 1.35 g cm^{-3}$
Results with Control Approach II

- uniform flow
- $T = 131K, \rho = 1.35\text{g/cm}^{-3}$

![Graphs showing density and velocity over distance to interface with and without control.](image-url)
AC Interface for Water at Equilibrium

1 nm
1D embedding

Force [kJ/(mol nm)]

Distance to interface [nm]

Red. density

RESULT WITHOUT CONTROL

RESULT WITH CONTROL
Water Couette Flow

CFD

MD

CFD

-\( v \)

\( v \)

Velocity [nm/ps]

\(-0.10\)

\(-0.05\)

0.00

0.05

0.10

-\( x [\text{nm}] \)

-6

-5

-4

-3

-2

-1

0

1

2

3

4

5

6

CFD

MD

CFD

Monday, July 23, 12
MULTISCALE METHODS + GPU

MD - Lattice-Boltzmann
MULTISCALE METHODS + GPU

MD - Lattice-Boltzmann
MULTIPHYSICS PARTICLE SIMULATIONS
Particles for Biology @ CSE Lab

Diffusion inside/ on Real Cells
Particles for Biology @ CSE Lab

Diffusion inside/ on Real Cells

Mesenchymal Motion
Particles for Biology @ CSE Lab

Growth and Form

Diffusion inside/on Real Cells

Mesenchymal Motion
Particles for Biology @ CSE Lab

- Growth and Form
- Virtual Surgery
- Diffusion inside/on Real Cells
- Mesenchymal Motion
Particles for Biology @ CSE Lab

- Growth and Form
- Virtual Surgery
- Diffusion inside/on Real Cells
- Mesenchymal Motion
- Cancer Modeling
Particles for Biology @ CSE Lab

- Growth and Form
- Virtual Surgery
- Diffusion inside/on Real Cells
- Mesenchymal Motion
- Cancer Modeling

Monday, July 23, 12
The main **biosynthetic organelle** in Eukaryotes: Protein and lipid synthesis. Enclosed by a **contiguous** membrane.
The main **biosynthetic organelle** in Eukaryotes: Protein and lipid synthesis. Enclosed by a **contiguous** membrane.
FRAP: Fluorescence Recovery After Photobleaching

- Tag protein fluorescently
- Laser Bleach region of interest
- Monitor influx of unbleached protein

\[ D = ? \]
FRAP: Fluorescence Recovery After Photobleaching

- Tag protein fluorescently
- Laser Bleach region of interest
- Monitor influx of unbleached protein

FRAP: Fluorescence Recovery After Photobleaching

\[ D = ? \]

Helenius group (ETHZ)

Monday, July 23, 12
Recall: Diffusion in CFD (relatively easy)

\[
\frac{D\omega}{Dt} = \omega \cdot \nabla u + \nu \nabla^2 \omega
\]

\[
\frac{dx_p}{dt} = u
\]
Recall: Diffusion in CFD (relatively easy)

\[
\frac{\partial u}{\partial t} + u \cdot \nabla u = -\nabla P + \nu \nabla^2 u
\]

\[
\frac{D\omega}{Dt} = \omega \cdot \nabla u + \nu \nabla^2 \omega
\]

\[
\frac{dx_p}{dt} = u
\]
Recall: Diffusion in CFD (relatively easy)

\[ \nabla \times \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) = -\nabla P + \nu \nabla^2 u \]

\[
\frac{D\omega}{Dt} = \omega \cdot \nabla u + \nu \nabla^2 \omega
\]

\[
\frac{dx_P}{dt} = u
\]
Recall: Diffusion in CFD (relatively easy)

\[ \nabla \times \left( \frac{\partial u}{\partial t} + u \cdot \nabla u = -\nabla P + \nu \nabla^2 u \right) \]

\[ \omega = \nabla \times u \quad \nabla^2 u = -\nabla \times \omega \]

\[ \frac{D\omega}{Dt} = \omega \cdot \nabla u + \nu \nabla^2 \omega \]

\[ \frac{dx_p}{dt} = u \]
Recall: Diffusion in CFD (relatively easy)

\[ \nabla \times \left( \frac{\partial u}{\partial t} + u \cdot \nabla u - \nabla P + \nu \nabla^2 u \right) \]

\[ \omega = \nabla \times u \quad \nabla^2 u = -\nabla \times \omega \]

\[ \frac{D\omega}{Dt} = \omega \cdot \nabla u + \nu \nabla^2 \omega \]

\[ \frac{dx_p}{dt} = u \]
Recall: Diffusion in CFD (relatively easy)

\[ \nabla \times \left( \frac{\partial \mathbf{u}}{\partial t} \right) \quad \mathbf{u} \cdot \nabla \mathbf{u} = \nabla P \quad \nu \nabla^2 \mathbf{u} \]

\[ \omega = \nabla \times \mathbf{u} \quad \nabla^2 \mathbf{u} = \nabla \times \omega \]

\[ \frac{D\omega}{Dt} = \omega \cdot \nabla \mathbf{u} + \nu \nabla^2 \omega \quad \frac{dx_p}{dt} = \mathbf{u} \]
Recall: Diffusion in CFD (relatively easy)

\[ \nabla \times \left( \frac{\partial u}{\partial t} \right) + u \cdot \nabla u = \nabla P + \nu \nabla^2 u \]

\[ \nabla \times u \quad \nabla^2 u - \nabla \times \omega \]

\[ \frac{D\omega}{Dt} = \omega \cdot \nabla u + \nu \nabla^2 \omega \]

\[ \frac{dx_p}{dt} = u \]
Recall: Diffusion in CFD (relatively easy)

\[ \nabla \times \left( \begin{array}{ccc} \frac{\partial u}{\partial t} & u \cdot \nabla u = \nabla P & \nu \nabla^2 u \end{array} \right) \]

\[ \omega = \nabla \times u \quad \nabla^2 u = \nabla \times \omega \]

\[ \frac{D\omega}{Dt} = \omega \cdot \nabla u + \nu \nabla^2 \omega \]

\[ \frac{dx_p}{dt} = u \]
Recall: Diffusion in CFD (relatively easy)

\[ \nabla \times \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) = \nabla P + \nu \nabla^2 u \]

\[ \omega = \nabla \times u \]

\[ \nabla^2 u = \nabla \times \omega \]

\[ \frac{D \omega}{Dt} = \omega \cdot \nabla u + \nu \nabla^2 \omega \]

\[ \frac{d x_p}{dt} = u \]

“Vorticity” becomes “Concentration”
Diffusion in the Endoplasmic Reticulum

Sbalzarini, Mezzacasa, Helenius, Koumoutsakos, Biophysical J., 2006
...but, can you do this on a surface?

'...but, can you do this on a surface?' - A. Helenius

\[ \frac{\partial u}{\partial t} = \nabla_M \cdot (D \nabla_M u) \]

**Projection operator**

\[ T = \left(1 - \frac{\nabla \Phi \times \nabla \Phi}{|\nabla \Phi|^2}\right)|\nabla \Phi| \quad \Lambda = T \cdot D \]

Multiresolution particles

- Particles in a band around the surface
- Surfaces have to be at least one band thickness apart
Multiresolution particles

- Particles in a band around the surface
- Surfaces have to be at least one band thickness apart
Multiresolution particles

- Particles in a band around the surface
- Surfaces have to be at least one band thickness apart

Use adapted particles around the surface to improve resolution
Multiresolution particles

- Particles in a band around the surface
- Surfaces have to be at least one band thickness apart

Use adapted particles around the surface to improve resolution.
Multiresolution particles

- Particles in a band around the surface
- Surfaces have to be at least one band thickness apart

Use adapted particles around the surface to improve resolution
Diffusion on reconstructed ER of VERO cells

Diffusion/Reaction on Surfaces

Gray Scott system

\[ U + 2V \rightarrow 3V, \]
\[ V \rightarrow P \]

Lagrangian particle level set method + reaction-diffusion on implicit surface
Gray Scott system

\[ U + 2V \rightarrow 3V, \]
\[ V \rightarrow P \]
“Well, the stripes are easy, but what about the horse part?" - Turing
“Well, the *stripes* are easy, but what about the *horse* part?" — Turing
“Well, the stripes are easy, but what about the horse part?”

GROWTH: Reaction-Diffusion on Deforming Geometries
"Well, the stripes are easy, but what about the horse part?" - Turing

**GROWTH**: Reaction-Diffusion on **Deforming** Geometries
Diffusion/Reaction + **Growth** on Surfaces
Diffusion/Reaction + Growth Surfaces
Diffusion/Reaction + Growth Surfaces
Diffusion/Reaction + **Growth** Surfaces
Diffusion/Reaction + **Growth** Surfaces

Monday, July 23, 12
Diffusion/Reaction + **Growth** Surfaces

---

**Monday, July 23, 12**
Reaction-diffusion coupled with growth

**APPLICATION**: Solid tumor model

- Tissue incompressible
- Porous medium
- Sharp interface

Proliferation $\Rightarrow$ Pressure $\Rightarrow$ Velocity
i.e. proliferation has **global** effect

\[-\nabla \cdot \mathbf{u} = S\]
\[
\mathbf{u} = -\nabla p
\]

**mass conservation**
**Darcy’s Law**
Solid tumor model

\[ \Delta p = \begin{cases} 
-\mu (c - A) & \text{if } c \geq N, \\
\mu \mu_N & \text{if } c < N,
\end{cases} \]

\[ \frac{\partial c}{\partial t} = \Delta c - \lambda c \]

\[ [p]_{\Gamma} = \gamma \kappa \]

\[ c_{\Gamma} = 1 \]

\[ \Delta p = 0 \]

\[ c = 1 \]

Cristini, Lowengrub, Nie, Friedman
Solid tumor model

\[
\Delta p = \begin{cases} 
-\mu (c - A) & \text{if } c \geq N, \\
\mu \mu_N & \text{if } c < N, \text{ necrosis}
\end{cases}
\]

\[
\frac{\partial c}{\partial t} = \Delta c - \lambda c
\]

\[
[p]_\Gamma = \gamma \kappa \\
c_\Gamma = 1
\]

\[
\Delta p = 0 \\
c = 1
\]

Cristini, Lowengrub, Nie, Friedman
Solid tumor model

\[\Delta p = \begin{cases} 
-\mu (c - A) & \text{if } c \geq N, \\
\mu \mu N & \text{if } c < N, \text{ necrosis}
\end{cases}\]

\[\frac{\partial c}{\partial t} = \Delta c - \lambda c\]

surface tension \[\Gamma = \gamma \kappa\]

\[c_\Gamma = 1\]

\[\Gamma(t)\]

Cristini, Lowengrub, Nie, Friedman
Solid tumor model with necrosis

\[ \gamma = 1 \quad N = \frac{1}{2} \quad \lambda = 1 \quad \mu = 20 \quad \mu_G = 1 \quad \mu_N = 1 \]
Solid tumor model with necrosis

\[ \gamma = 1 \quad N = \frac{1}{2} \quad \lambda = 1 \quad \mu = 20 \quad \mu_G = 1 \quad \mu_N = 1 \]
Tumor Growth:

Simulations with a Necrotic Core

\[ \bar{c} = 0.5 \quad \mu = 20 \quad N = 0.5 \quad G_{\nu} = 1.0 \quad \gamma = 1.0 \]
COMPUTATION : Exploring Possibilities (and bridging disciplines)

Monday, July 23, 12
COMPUTATION : Exploring Possibilities (and bridging disciplines)

Monday, July 23, 12
COMPUTATION: Exploring Possibilities (and bridging disciplines)

Monday, July 23, 12
CROWN DROPLET BREAKUP
marangoni instability of a drop impact onto an ethanol sheet

**VASCULOGENESIS**

blood vessel formation in embryonic development

---

**CROWN DROPLET BREAKUP**

marangoni instability of a drop impact onto an ethanol sheet

---

**COMPUTATION : Exploring Possibilities** (and bridging disciplines)

---


Vortex Rings: $Re = 3000$
The Fluid Mechanics of Cancer: Angiogenesis

CRANIAL VESSEL ANGIOGENESIS IN ZEBRAFISH
HTTP://ZFISH.NICHHD.NIH.GOV/ZFATLAS/FLI-GFP/FLI_MOVIES.HTML

Monday, July 23, 12
The Fluid Mechanics of Cancer: Angiogenesis
Multiscale modeling of Angiogenesis

Vasculature

Multiscale modeling of Angiogenesis

Tip Cell

Vasculature

ECM

Scale

Multiscale modeling of Angiogenesis

Tip Cell

- Selection and Signaling
- Migration
- Extending Filopodia

Vasculature

Growth Factors

ECM

Scale

Elements of Cellular Dynamics

- **cells stick to cells**
  - transmembrane CAMs: cadherin, ICAM-1, ...
  - formation of clusters, cords

- **cells guided by the extracellular matrix**
  - transmembrane CAMs: integrins, ...
  - facilitates migration

- **cells secrete proteinases**
  - Matrix metalloproteinases: degrade matrix,
  - free matrix-bound growth factors

- **cells sense chemical gradients**
  - gradients of “chemoattractant” serve as migratory cues

- **cells proliferate**
Tip Cell Migration - Chemotaxis

**Tip Cell Migration:**

\[
\frac{\mathbf{x}_p}{\partial t} = \mathbf{u}_p, \quad \frac{\mathbf{u}_p}{\partial t} = a_p - \lambda \mathbf{u}_p
\]

**VEGF Reaction-Diffusion:**

\[
\frac{\partial [dV]}{\partial t} = k_V \nabla^2 [dV] + \gamma_V [T][V] + \alpha_V [bV] - d_V [dV] - \beta_V [dV] ([ECM] - [bV]) - v_V [dV][EC].
\]

**Concentrations:**
- \([bV]\) - bound VEGF
- \([dV]\) - soluble VEGF
- \([V]\) - total VEGF

**Densities:**
- \([T]\) - Tumor Cells
- \([ECM]\) - ECM
- \([EC]\) - Endothelial Cells

\(\mathbf{x}_p\) - Particle location
\(a_p\) - Migration acceleration
\(\mathbf{u}_p\) - Migration velocity
\(\lambda\) - Drag coefficient

Tip Cell Migration: 

\[
\frac{\mathbf{x}_p}{\partial t} = \mathbf{u}_p, \quad \frac{\mathbf{u}_p}{\partial t} = \mathbf{a}_p - \lambda \mathbf{u}_p
\]

VEGF Reaction-Diffusion: 

\[
\frac{\partial [\text{dV}]}{\partial t} = k_V \nabla^2 [\text{dV}] + \gamma_V [T][V] + \alpha_V [\text{bV}] - d_V [\text{dV}] \\
- \beta_V [\text{dV}] ([\text{ECM}] - [\text{bV}]) - \nu_V [\text{dV}][\text{EC}].
\]

**Concentrations:** 
- \([\text{bV}]\) - bound VEGF 
- \([\text{dV}]\) - soluble VEGF 
- \([V]\) - total VEGF 

**Densities:** 
- \([T]\) - Tumor Cells 
- \([\text{ECM}]\) - ECM 
- \([\text{EC}]\) - Endothelial Cells 

**Variables:** 
- \(\mathbf{x}_p\) - Particle location 
- \(\mathbf{a}_p\) - Migration acceleration 
- \(\mathbf{u}_p\) - Migration velocity 
- \(\lambda\) - Drag coefficient

Tip Cell Migration - Chemotaxis

**Tip Cell Migration:**

\[ \frac{\mathbf{x}_p}{\partial t} = \mathbf{u}_p, \quad \frac{\mathbf{u}_p}{\partial t} = a_p - \lambda \mathbf{u}_p \]

**VEGF Reaction-Diffusion:**

\[
\frac{\partial [dV]}{\partial t} = k_V \nabla^2 [dV] + \gamma_V [T][V] + \alpha_V [bV] - d_V [dV] \\
- \beta_V [dV] ([ECM] - [bV]) - v_V [dV][EC].
\]

**Concentrations:**
- \([bV] - \text{bound VEGF}\)
- \([dV] - \text{soluble VEGF}\)
- \([V] - \text{total VEGF}\)

**Densities:**
- \([T] - \text{Tumor Cells}\)
- \([ECM] - \text{ECM}\)
- \([EC] - \text{Endothelial Cells}\)

**Symbols:**
- \(\mathbf{x}_p - \text{Particle location}\)
- \(a_p - \text{Migration acceleration}\)
- \(\mathbf{u}_p - \text{Migration velocity}\)
- \(\lambda - \text{Drag coefficient}\)

Tip Cell Migration - Chemotaxis

Tip Cell Migration:

\[
\frac{\mathbf{x}_p}{\partial t} = \mathbf{u}_p, \quad \frac{\mathbf{u}_p}{\partial t} = a_p - \lambda \mathbf{u}_p
\]

VEGF Reaction-Diffusion:

\[
\frac{\partial [dV]}{\partial t} = k_V \nabla^2 [dV] + \gamma_V [T] [V] + \alpha_V [bV] - d_V [dV] \\
- \beta_V [dV] ([ECM] - [bV]) - v_V [dV] [EC].
\]

Concentrations:
- \([bV]\) - bound VEGF
- \([dV]\) - soluble VEGF
- \([V]\) - total VEGF

Densities:
- \([T]\) - Tumor Cells
- \([ECM]\) - ECM
- \([EC]\) - Endothelial Cells

Symbols:
- \(\mathbf{x}_p\) - Particle location
- \(a_p\) - Migration acceleration
- \(\mathbf{u}_p\) - Migration velocity
- \(\lambda\) - Drag coefficient

References:
Tip Cell Migration - Chemotaxis

Tip Cell Migration:

\[ \frac{\mathbf{x}_p}{\partial t} = \mathbf{u}_p, \quad \frac{\mathbf{u}_p}{\partial t} = \mathbf{a}_p - \lambda \mathbf{u}_p \]

VEGF Reaction-Diffusion:

\[ \frac{\partial [dV]}{\partial t} = k_V \nabla^2 [dV] + \gamma_V [T][V] + \alpha_V [bV] - d_V [dV] \]
\[- \beta_V [dV] ([ECM] - [bV]) - v_V [dV][EC]. \]

Concentrations:
- \([bV]\) - bound VEGF
- \([dV]\) - soluble VEGF
- \([V]\) - total VEGF

Densities:
- \([T]\) - Tumor Cells
- \([EC]\) - Endothelial Cells
- \([ECM]\) - ECM

\(\mathbf{x}_p\) - Particle location
\(\mathbf{a}_p\) - Migration acceleration
\(\mathbf{u}_p\) - Migration velocity
\(\lambda\) - Drag coefficient

Tip Cell Migration - Chemotaxis

Tip Cell Migration:

\[
\frac{\mathbf{x}_p}{\partial t} = \mathbf{u}_p, \quad \frac{\mathbf{u}_p}{\partial t} = \mathbf{a}_p - \lambda \mathbf{u}_p
\]

VEGF Reaction-Diffusion:

\[
\frac{\partial [dV]}{\partial t} = k_V \nabla^2 [dV] + \gamma_V [T][V] + \alpha_V [bV] - d_V [dV] \\
- \beta_V [dV] ([ECM] - [bV]) - \nu_V [dV][EC].
\]

Concentrations:
- \([bV]\) - bound VEGF
- \([dV]\) - soluble VEGF
- \([V]\) - total VEGF

Densities:
- \([T]\) - Tumor Cells
- \([ECM]\) - ECM
- \([EC]\) - Endothelial Cells

- \(\mathbf{x}_p\) - Particle location
- \(\mathbf{a}_p\) - Migration acceleration
- \(\mathbf{u}_p\) - Migration velocity
- \(\lambda\) - Drag coefficient

Tip Cell Migration:

\[
\frac{\mathbf{x}_p}{\partial t} = \mathbf{u}_p, \quad \frac{\mathbf{u}_p}{\partial t} = a_p - \lambda \mathbf{u}_p
\]

VEGF Reaction-Diffusion:

\[
\frac{\partial [dV]}{\partial t} = k_V \nabla^2 [dV] + \gamma_V [T][V] + \alpha_V [bV] - d_V [dV] - \beta_V [dV] ([ECM] - [bV]) - \nu_V [dV][EC].
\]

Concentrations:
- \([bV]\) - bound VEGF
- \([dV]\) - soluble VEGF
- \([V]\) - total VEGF

Densities:
- \([T]\) - Tumor Cells
- \([ECM]\) - ECM
- \([EC]\) - Endothelial Cells

EC Uptake

- \(\mathbf{x}_p\) - Particle location
- \(a_p\) - Migration acceleration
- \(\mathbf{u}_p\) - Migration velocity
- \(\lambda\) - Drag coefficient

Tip Cell Migration - Chemotaxis

Tip Cell Migration:

\[ \frac{x_p}{\partial t} = u_p, \quad \frac{u_p}{\partial t} = a_p - \lambda u_p \]

VEGF Reaction-Diffusion:

\[ \frac{\partial [dV]}{\partial t} = k_V \nabla^2 [dV] + \gamma_V [T] [V] + \alpha_V [bV] - d_V [dV] \\
\quad - \beta_V [dV] ([ECM] - [bV]) - \nu_V [dV] [EC]. \]

Concentrations:
- [bV] - bound VEGF
- [dV] - soluble VEGF
- [V] - total VEGF

Densities:
- [T] - Tumor Cells
- [ECM] - ECM
- [EC] - Endothelial Cells

Parameters:
- \( x_p \) - Particle location
- \( a_p \) - Migration acceleration
- \( u_p \) - Migration velocity
- \( \lambda \) - Drag coefficient

Endothelial Cell Representation

Hybrid representation of ECs:

Tip cell particles $Q_p$:
- Discrete particle representation
- Particle location: $x_p$
- Migration acceleration: $u_p$
- Drag coefficient: $\lambda$

\[
\frac{d x_p}{dt} = u_p, \quad \frac{d u_p}{dt} = a_p - \lambda u_p
\]

Stalk cell density $\rho$:
- Continuum vessel representation
- Tip and stalk communicate through Particle-Mesh, Mesh-Particle interpolations

\[
\rho_{i}^{n+1} = \max \left( \rho_{i}^{n}, \sum_{p} B(ih - x_p) Q_p \right)
\]

\[
Q_p = \sum_{i} h^3 q_i M^\prime (x_p - ih)
\]

Tip Cell “deposes” endothelial cells

Monday, July 23, 12
Endothelial Cell Representation

Hybrid representation of ECs:

Tip cell particles $Q_p$:
- Discrete particle representation
- Particle location: $x_p$
- Migration acceleration: $u_p$
- Drag coefficient: $\lambda$

\[
\frac{x_p}{\partial t} = u_p, \quad \frac{u_p}{\partial t} = a_p - \lambda u_p
\]

Stalk cell density $\rho$:
- Continuum vessel representation
- Tip and stalk communicate through Particle-Mesh, Mesh-Particle interpolations

\[
\rho_{i}^{n+1} = \max\left(\rho_{i}^{n}, \sum_{p} B(i h - x_p) Q_p\right)
\]

\[
Q_p = \sum_{i} h^3 q_i M_A'(x_p - ih)
\]

Tip Cell “deposes” endothelial cells
Vascular Endothelial Growth Factors

Growth Factors
- Released By Cells, From Matrix
- React/Diffuse/Bind-Unbind

### Model equations

**Living tumor cells**

\[ \begin{align*}
\frac{\partial u_T}{\partial t} &= \omega_T \nabla \cdot (u_T \nabla u) \quad \text{convection} \\
&+ u_T u_N H (u_N - \bar{u}_N u_T) \quad \text{proliferation} \\
&- \delta_T H (\bar{u}_N u_T - u_N) u_T \quad \text{in } \Omega, \\
u_T &= \bar{u} - u_D - u_C - \hat{u}_C \quad \text{on } \delta\Omega.
\end{align*} \]

**TAF**

\[ \begin{align*}
\frac{\partial u_A}{\partial t} &= k_A \nabla^2 u_A \quad \text{diffusion} \\
&+ \gamma_A u_T H (\bar{u}_N u_T - u_N) \quad \text{secretion by hypoxic tumor cells} \\
&- \nu_A u_C u_A \quad \text{uptake by EC} \\
u_A &= 0 \quad \text{in } \delta D.
\end{align*} \]

**Nutrient**

\[ \begin{align*}
\frac{\partial u_N}{\partial t} &= v_N \nabla \cdot [(k_E + k_N (u_C + \hat{u}_C)) \nabla u_N] \quad \text{diffusion} \\
&- v_N u_T u_N \quad \text{consumption} \\
u_N &= \epsilon + \beta (u_C + \hat{u}_C) \quad \text{in } \Omega, \\
u_N &= 0 \quad \text{on } \delta\Omega.
\end{align*} \]

**Endothelial cells**

\[ \begin{align*}
\frac{\partial u_C}{\partial t} &= k_C \nabla^2 u_C \quad \text{diffusion} \\
&+ \gamma_C u_A (\bar{u}_C - u_C - u_T - u_D)_+ (u_C + \hat{u}_C) \quad \text{proliferation} \\
&- \delta_C u_C \quad \text{death} \\
u_C &= 0 \quad \text{outside } \Omega, \\
u_C &= 0 \quad \text{on } \delta D.
\end{align*} \]
Matrix-bound VEGF (bVEGF) - Assumptions

- Some VEGF isoforms express heparin-binding sites binding to domains in the ECM

- Local gradients of matrix bound VEGF influence sprout morphology

- Matrix bound VEGF is cleaved by Matrix Metalloproteinases (MMPs) released at endothelial sprout tips


Matrix-bound VEGF - Modeling

- Initially distributed in pockets
- Establishes local chemotactic gradient
- Cleaved VEGF (cVEGF) becomes soluble
  - bVEGF is cleaved by MMPs
  - Uptake of cVEGF by ECs ρ
  - cVEGF diffuses through ECM
  - cVEGF is subject to natural decay

\[
\frac{\partial [b\text{VEGF}]}{\partial t} = -C ([b\text{VEGF}], [\text{MMP}]) - U ([b\text{VEGF}], \rho)
\]

\[
C ([b\text{VEGF}], [\text{MMP}]) = \min ([b\text{VEGF}], \nu_b \text{MMP}[b\text{VEGF}])
\]

\[
\frac{\partial [c\text{VEGF}]}{\partial t} = k_V \nabla^2 [c\text{VEGF}] + C ([b\text{VEGF}], [\text{MMP}]) - U ([c\text{VEGF}], \rho) - \delta_V [c\text{VEGF}]
\]
Matrix-bound VEGF - Modeling

- Initially distributed in pockets
- establishes local chemotactic gradient
- cleaved VEGF (cVEGF) becomes soluble
  - bVEGF is cleaved by MMPs
  - Uptake of cVEGF by ECs ρ
  - cVEGF diffuses through ECM
  - cVEGF is subject to natural decay

\[
\frac{\partial [b\text{VEGF}]}{\partial t} = -C ([b\text{VEGF}], [\text{MMP}]) - U ([b\text{VEGF}], \rho)
\]

\[
C ([b\text{VEGF}], [\text{MMP}]) = \min ([b\text{VEGF}], v_b [\text{MMP}] [b\text{VEGF}])
\]

\[
\frac{\partial [c\text{VEGF}]}{\partial t} = k_V \nabla^2 [c\text{VEGF}] + C ([b\text{VEGF}], [\text{MMP}]) - U ([c\text{VEGF}], \rho) - \delta_V [c\text{VEGF}]
\]
Matrix-bound VEGF - Modeling

- Initially distributed in pockets
- Establishes local chemotactic gradient
- Cleaved VEGF (cVEGF) becomes soluble

  - bVEGF is cleaved by MMPs
  - Uptake of cVEGF by ECs ρ
  - cVEGF diffuses through ECM
  - cVEGF is subject to natural decay

\[
\frac{\partial [b\text{VEGF}]}{\partial t} = - C ([b\text{VEGF}], [\text{MMP}]) - U ([b\text{VEGF}], \rho)
\]

\[
C ([b\text{VEGF}], [\text{MMP}]) = \min ([b\text{VEGF}], v_{bV} [\text{MMP}] [b\text{VEGF}])
\]

\[
\frac{\partial [c\text{VEGF}]}{\partial t} = k_V \nabla^2 [c\text{VEGF}] + C ([b\text{VEGF}], [\text{MMP}]) - U ([c\text{VEGF}], \rho) - \delta_V [c\text{VEGF}]
\]
Matrix-bound VEGF - Modeling

- Initially distributed in pockets
- establishes local chemotactic gradient
- cleaved VEGF (cVEGF) becomes soluble
  - bVEGF is cleaved by MMPs
  - Uptake of cVEGF by ECs ρ
  - cVEGF diffuses through ECM
  - cVEGF is subject to natural decay

\[
\frac{\partial [b\text{VEGF}]}{\partial t} = -C ([b\text{VEGF}], [MMP]) - U ([b\text{VEGF}], \rho)
\]

\[
C ([b\text{VEGF}], [MMP]) = min ([b\text{VEGF}], v_{bV}[MMP][b\text{VEGF}])
\]

\[
\frac{\partial [c\text{VEGF}]}{\partial t} = k_V \nabla^2 [c\text{VEGF}] + C ([b\text{VEGF}], [MMP]) - U ([c\text{VEGF}], \rho) - \delta_V [c\text{VEGF}]
\]
Matrix-bound VEGF - Modeling

- Initially distributed in pockets
- Establishes local chemotactic gradient
- Cleaved VEGF (cVEGF) becomes soluble
  - bVEGF is cleaved by MMPs
  - Uptake of cVEGF by ECs ρ
  - cVEGF diffuses through ECM
    - cVEGF is subject to natural decay

\[
\frac{\partial [b\text{VEGF}]}{\partial t} = -C ([b\text{VEGF}], [\text{MMP}]) - U ([b\text{VEGF}], \rho)
\]

\[
C ([b\text{VEGF}], [\text{MMP}]) = \min ([b\text{VEGF}], \nu_b [\text{MMP}][b\text{VEGF}])
\]

\[
\frac{\partial [c\text{VEGF}]}{\partial t} = k_V \nabla^2 [c\text{VEGF}] + C ([b\text{VEGF}], [\text{MMP}]) - U ([c\text{VEGF}], \rho) - \delta_V [c\text{VEGF}]
\]
Matrix-bound VEGF leads to increased branching.

Vessel branching $\leftrightarrow$ capillary function

**BLOOD VESSEL FORMATION IN A MOUSE MODEL**

ONLY SOLUBLE VEGF
> THICKER VESSELS

SOLUBLE + MATRIX-BOUND VEGF
> INCREASED BRANCHING

RADIAL SOLUBLE VEGF GRADIENT AND LOCALIZED MATRIX-BOUND VEGF

Matrix-bound VEGF leads to increased branching. Vessel branching ↔ capillary function

Blood vessel formation in a mouse model

Matrix-bound VEGF leads to increased branching. vessel branching ↔ capillary function

BLOOD VESSEL FORMATION IN A MOUSE MODEL

The Extra-Cellular Matrix

- Fibrous Structure
- Guide Cell Migration
- Offer binding sites for VEGF and Fibronectin

Contractile fibroblasts secrete and organize *extracellular matrix fibers* (blue) that are loaded with *growth factor complexes* (green), resulting in a *turquoise* overlay color. The contractile fibers inside the cells are visualized by detecting a *smooth muscle protein* (red). The cells' nuclei are visualized in *yellow*. (Photo Credit: EPFL/LCB)
Extra-Cellular Matrix: Model

• Fibrous structures in ECM provide a guiding structure for migrating endothelial cells
• ECM fibers are subject of remodeling by migrating EC’s
• The ECM expresses binding sites for various growth factors and integrins

in vitro model 1: series of image slices illustrate the collagen fibril rearrangement around an early sprout in the axial dimension. The emboldened arrow signifies the center of the sprout.

Continuum Model Approach

Continuum Model of Mesenchymal Cell Migration

Considers:
• Cell-Cell Adhesion
• ECM-Cell Guidance and adhesion
• Pressure
• Chemotaxis
Continuum Model Approach

Continuum Model of Mesenchymal Cell Migration

Considers:
- Cell-Cell Adhesion
- ECM-Cell Guidance and adhesion
- Pressure
- Chemotaxis

Cell Density:

\[ \left\{ \rho_i \right\}_{i=1}^{\text{#CellTypes}} \]
Continuum Model Approach

Continuum Model of Mesenchymal Cell Migration

Considers:
- Cell-Cell Adhesion
- ECM-Cell Guidance and adhesion
- Pressure
- Chemotaxis

Cell Density:
\[
\{ \rho_i \}_{i=1}^{\#\text{CellTypes}}
\]

Evolution of Cell density:
\[
\frac{\partial \rho_i}{\partial t} + \nabla \cdot (a_i \rho) = \kappa_i \Delta \rho_i + R(\rho)
\]

- migrative response
- random fluctuation
- creation/death
Continuum Model of Mesenchymal Cell Migration

Considers:
- Cell-Cell Adhesion
- ECM-Cell Guidance and adhesion
- Pressure
- Chemotaxis

Cell Density:

\[ \{ \rho_i \} \quad \# \text{Cell Types} \]

Evolution of Cell density:

\[
\frac{\partial \rho_i}{\partial t} + \nabla \cdot (a_i \rho) = \kappa_i \Delta \rho_i + R(\rho) \]

Migrative Response:

\[ a_i = a_i^{c/c} + a_i^{\phi,e} + a_i^p. \]
Model Approach:

- Cells $\rho_i$ secrete adhesion signal $f_i$
- Short range diffusion establishing adhesion gradient
- Cells Respond to adhesion signal by migration upwards the adhesion signal.
- Inter and intra Cell-Type response can vary

Adhesion Signal:

$$\frac{\partial f_i}{\partial t} = D_i \Delta f_i + \alpha_i \left( 1 - \frac{f_i}{f_{i,\text{max}}} \right) \rho_i - \mu_i f_i$$

Cell Response:

$$a_{i,c/c} = \sum_j \kappa_{ij} L (f_j, df_j) \nabla f_j$$

Monday, July 23, 12
**Result:** Sprouting is non-linear

Given a chemotactic response strength $k_\phi$ how far do the cells migrate?

**Cell progression**

$$R(x) = \int \rho(x, y) \, dy$$

**Total cell mass**

$$R = \int \rho(x, y) \, dx \, dy$$

*Statistics over n = 16 different matrices*
Modeling the ECM

Fibers:
• straight
• random direction
• distribution of lengths

Representation:
• indicator field: $E_X$
• density field: $E_\rho$
• directional field: $K$

\[
l = l_0 2^m z
\]
\[
\alpha \in \mathcal{U}([0, \pi])
\]
\[
z \in \mathcal{N}(0, 1)
\]
Modeling the ECM

Fibers:
- straight
- random direction
- distribution of lengths

\[ l = l_0 \cdot 2^m \cdot z \]
\[ \alpha \in U([0, \pi]) \]
\[ z \in \mathcal{N}(0, 1) \]

Representation:
- indicator field: \( E_X \)
- density field: \( E_\rho \)
- directional field: \( K \)
Matrix-aware Chemotaxis - Tip Cells

Cells are guided by extracellular matrix
- transmembrane CAMs: integrins, ...
- facilitates migration

Cells sense chemical gradients
- gradients of “chemoattractant” serve as migratory cues

Cells stick to cells
- gradient of “haptotactic” molecules serve as migration cues

Migration Speed

\[ \mathbf{a} = \alpha (\mathbf{E}_\rho) \mathbf{T} \left( w_V \nabla \Psi + w_F \nabla \Phi_b \right) \]
Cells will **attach** to fibers if they are aligned with the chemotactic cue $\nabla \phi$.
Angiogenesis: in silico
Angiogenesis: in silico

Milde F., Bergdorf M., Koumoutsakos P. A hybrid model of sprouting angiogenesis, Biophysical J., 2008
Matrix-bound VEGF - Simulation
Effect of Matrix structure on branching - Mesenchymal cells

Mean number of junctions identified with AngioQuant

Statistics over n = 50 different matrices
Results: Matrix bound VEGF perturbs Vasculature

DISTRIBUTED VEGF POCKETS:
A: 0, B: 500, C: 1'400, E: 10'000
The Extra-Cellular Matrix

Vasculature

- Proliferation
- Lumen Formation
- Blood Flow and Leakage

Tips Cell

Growth Factors

ECM

Scale

Lumen Formation and Maturation

Endothelial Cells:

\[
\frac{\partial c_{ec}}{\partial t} + v_{ec} \nabla c_{ec} = \begin{cases} 
\lambda c_{ec} c_{ec} c_{tot,max} - c_{tot} & \text{if } c_{tot} > c_{tot,rlx} \\
\lambda c_{ec} c_{ec} & \text{otherwise}
\end{cases}
\]

Proliferation Signal:

\[
\frac{\partial c_s}{\partial t} + v_{ec} \nabla c_s = c_{tot} D_s \Delta c_s + \alpha_{s,tc} c_{tc} - \beta_{s,ec} c_{ec} c_s - \gamma_s c_s
\]

Fibronectin:

\[
\frac{\partial c_f}{\partial t} = \begin{cases} 
\alpha_{f,ec} c_{ec} & \text{if } c_f < c_{f,max} \\
0 & \text{otherwise}
\end{cases}
\]

Matrix Degrading Enzymes:

\[
\frac{\partial c_m}{\partial t} = D_m \Delta c_m + \alpha_{m,tc} c_{tc} c_m - \gamma_m c_m
\]
Lumen Formation and Maturation: **Simulation**

Lumen Formation and Maturation: Simulation

**Blood Flow in Complex Geometries**

### Governing Equations:
- Navier-Stokes equation for incompressible flow
- Brinkmann penalization method

\[
\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} + \frac{1}{\eta} \chi (\mathbf{u}_{BD} - \mathbf{u}) + \frac{f}{\rho}
\]

\[
\nabla \cdot \mathbf{u} = 0
\]
**Blood Flow in Complex Geometries**

**Governing Equations:**
- Navier-Stokes equation for incompressible flow
- Brinkmann penalization method

\[
\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} + \frac{1}{\eta} \chi (\mathbf{u}_{BD} - \mathbf{u}) + \frac{f}{\rho} \\
\nabla \cdot \mathbf{u} = 0
\]

**Penalization Method:**
- Complex geometries
  - flow in vessels
- Porous media
  - flow through leaky vessels
  - flow into tissue
- Moving boundaries
  - flow in growing vessel network
Blood Flow in Complex Geometries: Simulation
Blood Flow in Complex Geometries: Simulation
“Blood Flow” in Image Extracted Networks
Governing Equations

Lagrangian Formulation

**Isothermal Compressible Viscous Fluid**

\[
\frac{D \rho_l}{Dt} = -\rho_l \nabla \cdot u_l
\]
\[
\rho_l \frac{Du_l}{Dt} = -\nabla p_l + \nabla \tau_l
\]
\[
p_l = RT_0 \rho_l
\]
\[
\tau_{l,ij} = \mu \left( \frac{\partial u_{l,i}}{\partial x_j} - \frac{\partial u_{l,j}}{\partial x_i} - \frac{2}{3} \delta_{i,j} \frac{\partial u_{l,k}}{\partial x_k} \right)
\]

**Elastic Solid**

**Continuity equation**

\[
\frac{D \rho_s}{Dt} = -\rho_s \nabla \cdot u_s
\]

**Momentum equation**

\[
\rho_s \frac{Du_s}{Dt} = \nabla \sigma_s = \nabla \left( -p_s I + S \right)
\]

**Linear**

\[
p_s = c_0^2 \left( \rho_s - \rho_0 \right)
\]
\[
\sigma_s = f(F)
\]

**Nonlinear**

\[
\frac{DS}{Dt} = 2\mu \left( \dot{\epsilon} \left( \frac{1}{3} \delta_{ij} \dot{\epsilon} \right) \right)
\]
\[
\ddot{\epsilon} = \frac{1}{2} \left( \nabla u + (\nabla u)^T \right)
\]
\[
\frac{DF}{Dt} = \frac{\partial u}{\partial x} F
\]
Particle Equations - Fluid

Set of ODEs

**Isothermal Compressible Viscous Fluid**

\[ \frac{dx_p}{dt} = u_p \]
\[ \frac{d\rho_p}{dt} = -\rho_p \langle \nabla \cdot u \rangle_p \]
\[ \rho_p \frac{du_p}{dt} = -\langle \nabla p \rangle_p + \langle \nabla \tau \rangle_p \]

\[ p_p = RT_0 \rho_p \]
\[ \tau_{ij,p} = \mu \left( \frac{\partial u_i}{\partial x_j} \right)_p - \left( \frac{\partial u_i}{\partial x_j} \right)_p - 2 \delta_{i,j} \left( \frac{\partial u_k}{\partial x_k} \right)_p \]

\[ \langle \rangle_p : Approximation on particle p \]
Particle Equations - Solid

Set of ODEs

Elastic Solid

\[ \frac{dx_p}{dt} = u_p \]
\[ \frac{d\rho_p}{dt} = -\rho_p \langle \nabla \cdot u \rangle_p \]
\[ \rho_p \frac{d\mathbf{u}_p}{dt} = \langle \nabla \sigma \rangle_p = -\langle \nabla p \rangle_p + \langle \nabla \mathbf{S} \rangle_p \]

Linear

\[ P_p = c_0^2 (\rho_p - \rho_0) \]
\[ \frac{dS_p}{dt} = 2\mu \left( \dot{\mathbf{e}}_p - \frac{1}{3} \delta_{ij} \mathbf{e}_{ij,p} \right) \]
\[ \dot{\mathbf{e}}_p = \frac{1}{2} \left( \langle \nabla \mathbf{u} \rangle_p + \langle \nabla \mathbf{u} \rangle_p^T \right) \]

Nonlinear

\[ \sigma_p = f(F_p) \]
\[ \frac{dF_p}{dt} = \left\langle \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \right\rangle_p F_p \]

\[ \left\langle \quad \right\rangle_p : \text{Approximation on particle } p \]
Particle Simulation of Elastic Solid

**Plane Strain Compression Test**

- Pistons move with constant velocity
- Elastic solid fixed to the pistons
- Highly dynamic deformation of large extent

$V = V_{piston}$

$V = -V_{piston}$
**Particle Simulation of Elastic Solid**

**Plane Strain Compression Test**

- Pistons move with constant velocity
- Elastic solid fixed to the pistons
- Highly dynamic deformation of large extent
Plane Strain Compression Test

**Linear Elasticity**

- Young's Modulus = 100
- Poisson ratio = 0.49
- ~2000 particles/nodes

**Nonlinear Elasticity**

- Hyperelastic Material
  - $C_{10} = 2.2$, $D = 0.001$
  - ~2000 particles/nodes

S.E. Hieber and P. Koumoutsakos. A Lagrangian particle method for the simulation of linear and nonlinear elastic models of soft tissue. *al., J. Comp. Physics, accepted*
Simulation of Liver Tissue

Aspiration Test

- Experiment to determine constitutive models for biological tissue
- A vacuum created in the aspiration devices causes the tissue to form a bubble
- The height of the tissue bubble determines the parameters of the nonlinear model

_Nava et al., Technology and Health Care, 2004, vol. 12, 269-280_
Particle Simulation of Aspiration Test

- Experiment and nonlinear model from Nava et al. (2004)

- 3D Particle simulation using \( \sim 10^5 \) particles

- Good agreement with experimental results in the tissue displacement