

Towards a Parallel, 3D Simulation of Platelet Aggregation and Blood Coagulation

Oral Exam — Elijah Newren

January 7, 2004

B.6.2 Summarized Formulas	82
B.6.3 Why dividing by zero does not occur	85

1 Introduction

Coupled, intricate systems exist to maintain the fluidity of the blood in the vascular system while allowing for the rapid formation of a solid clot to prevent excessive blood loss subsequent to vessel injury [10]. These systems can be invoked as part of the body's normal defense mechanism against blood loss (a process referred to as hemostasis), but these same systems are also invoked during unwanted, pathological and perhaps life threatening clot formation known as thrombosis. Indeed, these systems can be seen as a delicate balancing act continually occurring to control clot formation and lysis in order to prevent hemorrhage without causing thrombosis [2]. Despite more than a century of research in blood biochemistry, platelet and vascular wall biology, and fluid dynamics, the complexity of blood clotting under flow has prevented quantitative and predictive modeling [13]. Yet quantitative modeling of blood function under flow could have numerous diagnostic and therapeutic uses.

When the wall of a blood vessel is injured, a variety of embedded molecules become exposed. This initiates two interacting processes known as platelet aggregation and blood coagulation. Both of these processes involve multiple subprocesses. Platelet aggregation starts when platelets suspended in the blood, which normally circulate in an inactive state, adhere to damaged tissue and undergo an activation process. During the activation of a platelet, the platelet changes from its rigid discoidal shape to a more deformable spherical form with several long, thin pseudopodia; the platelet's surface membrane becomes sticky to other activated platelets; and the platelet begins to

Outline

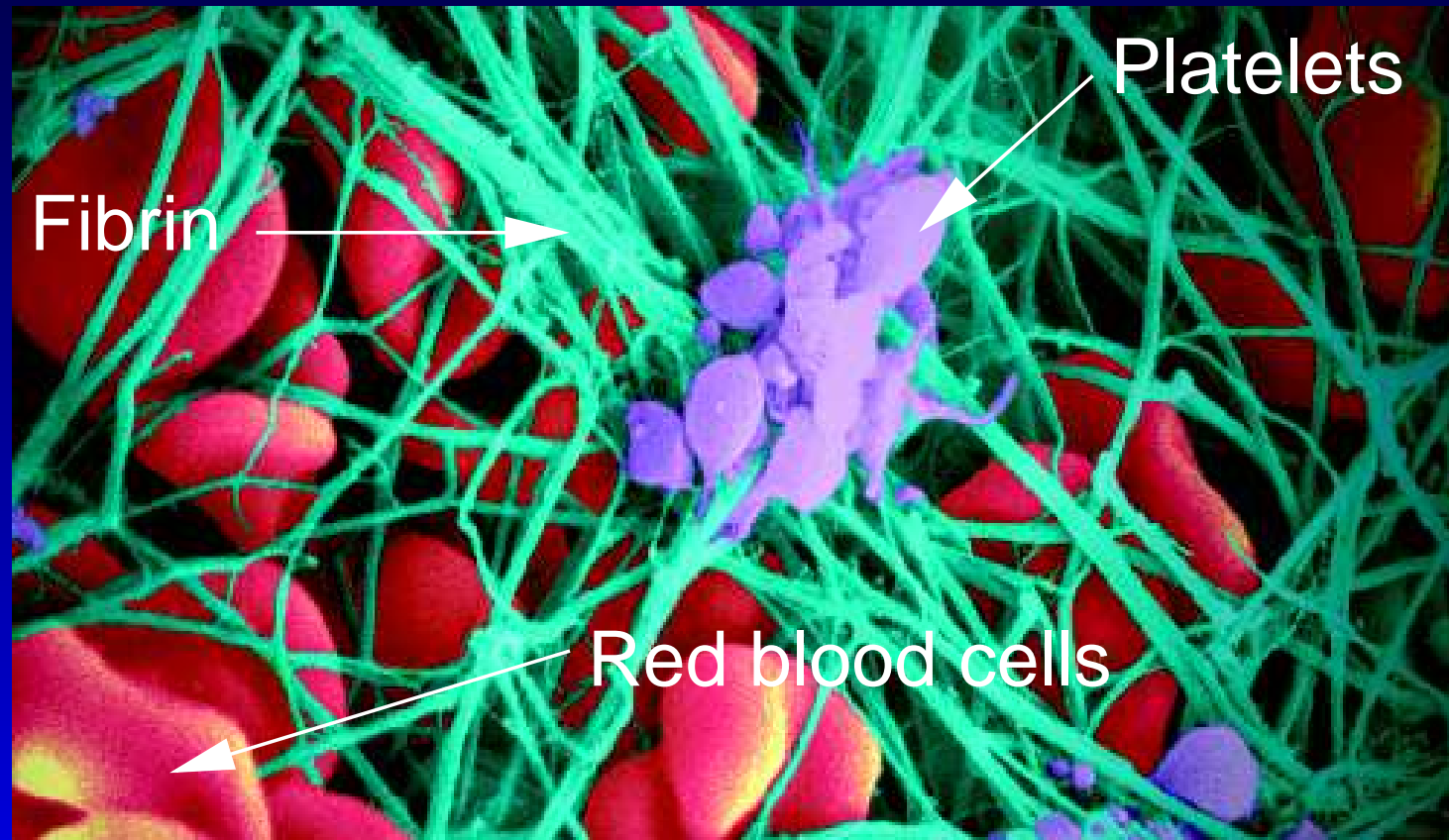
- Biological Problem
- Model Components
- Numerical and Communication Methods
- Future Work

Outline

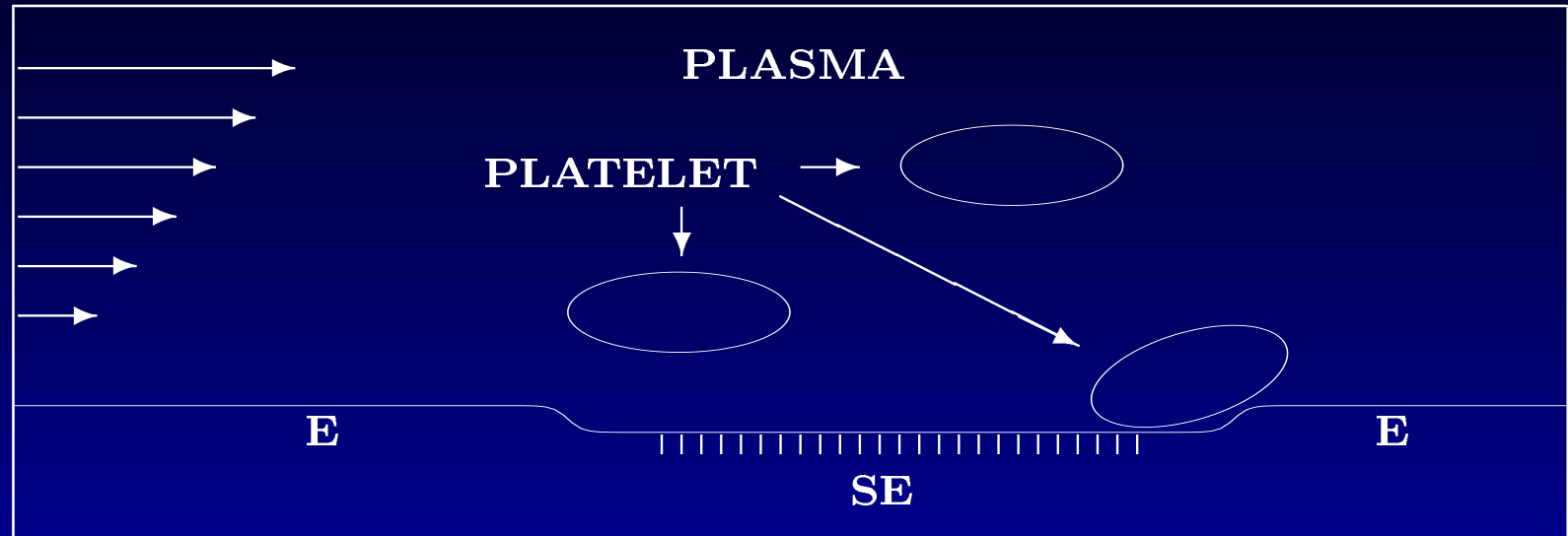
- Biological Problem
- Model Components
- Numerical and Communication Methods
- Future Work

A Blood Clot

A colorized scanning electron micrograph of a blood clot formed
in vitro without fbw

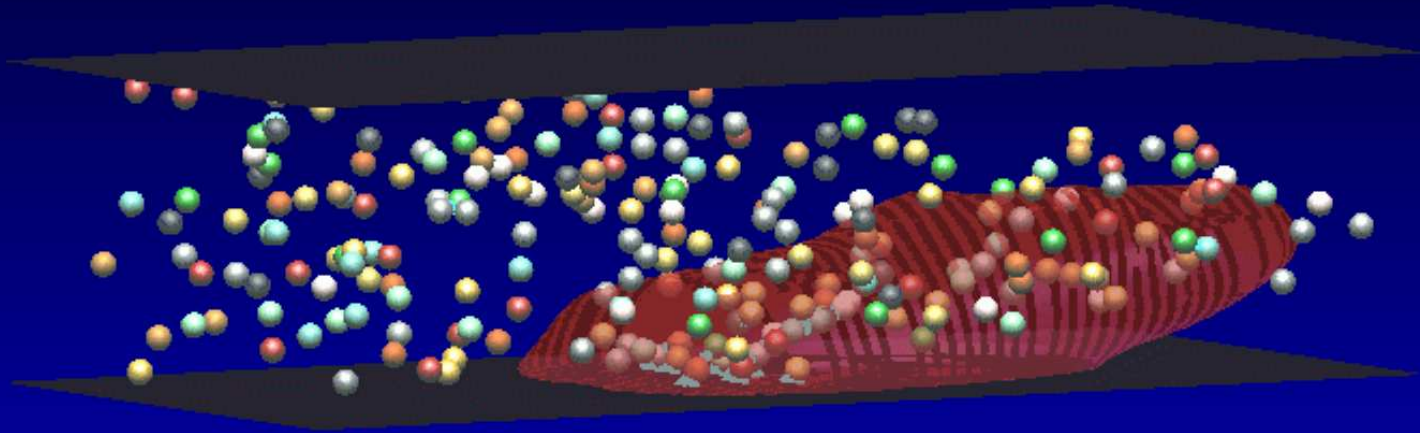


Platelet Aggregation

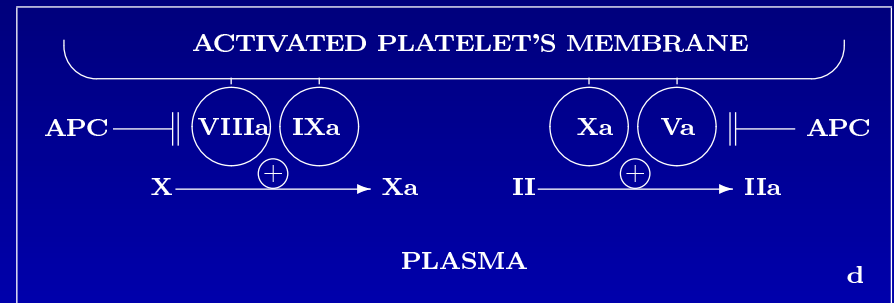
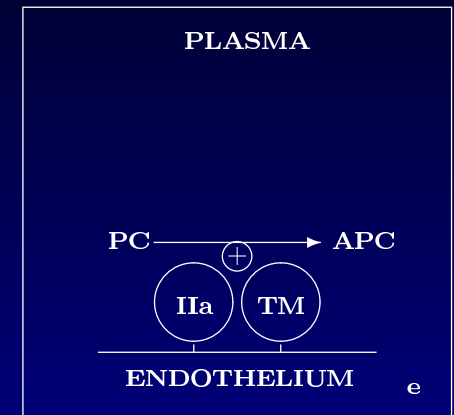
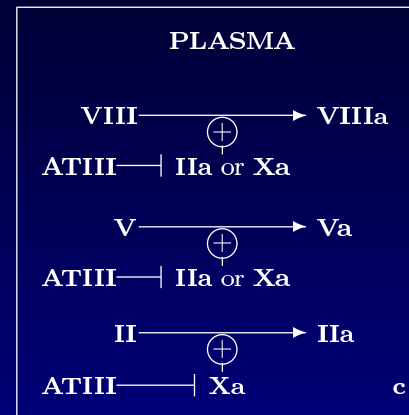
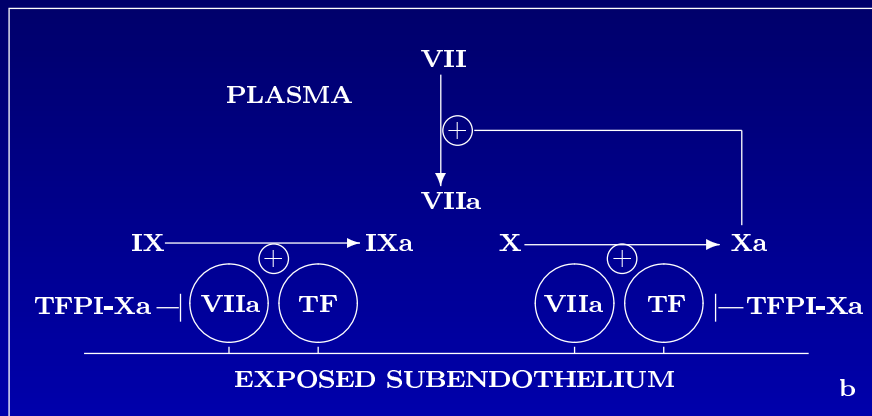
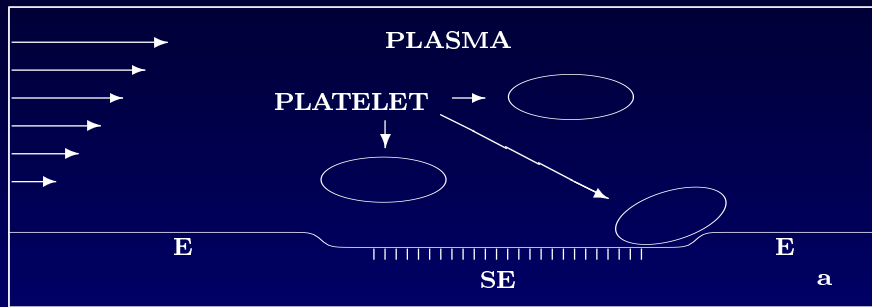


Platelet Aggregation

Simulation of Platelet Aggregation by H. Yu and A. Fogelson



Blood Coagulation



Reactions: (a) schematic of injured site. SE—exposed subendothelium, E—endothelium; (b) TF-VIIa system on subendothelium; (c) plasma-phase reactions; (d) VIIIa:IXa and Va:Xa complexes on activated platelet surface; (e) TM:IIa complex on endothelial surface.

→ indicates enzymatically-promoted reaction. ⊖ indicates inhibition. ⊎ indicates inactivation.

Interactions

- Platelets accelerate Coagulation

Interactions

- Platelets accelerate Coagulation
- Coagulation accelerates Platelet Aggregation

Interactions

- Platelets accelerate Coagulation
- Coagulation accelerates Platelet Aggregation
- Inhibitory Effects

Interactions

- Platelets accelerate Coagulation
- Coagulation accelerates Platelet Aggregation
- Inhibitory Effects
- Fibrinolysis

Interactions

- Platelets accelerate Coagulation
- Coagulation accelerates Platelet Aggregation
- Inhibitory Effects
- Fibrinolysis
- Advection & Diffusion

Problem Summary

- Little is known

Problem Summary

- Little is known
- Experimental difficulty

Problem Summary

- Little is known
- Experimental difficulty
- Other computational work

Problem Summary

- Little is known
- Experimental difficulty
- Other computational work
- My goals

Outline

- Biological Problem
- **Model Components**
- Numerical and Communication Methods
- Future Work

Model Components

- Fluid Solver

Model Components

- Fluid Solver
- Method for representing the surface of platelets and vessel walls and for tracking the connections between the various structures

Model Components

- Fluid Solver
- Method for representing the surface of platelets and vessel walls and for tracking the connections between the various structures
- Fluid-structure interactions

Model Components

- Fluid Solver
- Method for representing the surface of platelets and vessel walls and for tracking the connections between the various structures
- Fluid-structure interactions
- Advection-diffusion-reaction equations throughout fluid

Model Components

- Fluid Solver
- Method for representing the surface of platelets and vessel walls and for tracking the connections between the various structures
- Fluid-structure interactions
- Advection-diffusion-reaction equations throughout fluid
- Presence of platelets and walls

Model Components

- Fluid Solver
- Method for representing the surface of platelets and vessel walls and for tracking the connections between the various structures
- Fluid-structure interactions
- Advection-diffusion-reaction equations throughout fluid
- Presence of platelets and walls
- Reactions on surfaces

Model Components

- Fluid Solver
- Method for representing the surface of platelets and vessel walls and for tracking the connections between the various structures
- Fluid-structure interactions
- Advection-diffusion-reaction equations throughout fluid
- Presence of platelets and walls
- Reactions on surfaces
- Coupling of diffusion and binding/unbinding of surface chemicals

Outline

- Biological Problem
- Model Components
- Numerical and Communication Methods
 - Fluid Solver
 - Parallel Programming
 - Immersed Boundary Method
- Future Work

Navier Stokes Solver

$$\mathbf{u}_t + \nabla p = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu\Delta\mathbf{u} + \mathbf{f}$$

$$\nabla \cdot \mathbf{u} = 0$$

Navier Stokes Solver

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+\frac{1}{2}} = -[(\mathbf{u} \cdot \nabla)\mathbf{u}]^{n+\frac{1}{2}} + \frac{\nu}{2}\Delta(\mathbf{u}^{n+1} + \mathbf{u}^n) + \mathbf{f}^{n+\frac{1}{2}}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0$$

Navier Stokes Solver

$$\frac{\mathbf{u}^{*,k} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+\frac{1}{2},k} = -[(\mathbf{u} \cdot \nabla)\mathbf{u}]^{n+\frac{1}{2}} + \frac{\nu}{2}\Delta(\mathbf{u}^{*,k} + \mathbf{u}^n) + \mathbf{f}^{n+\frac{1}{2}}$$

$$\frac{\mathbf{u}^{n+1,k} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+\frac{1}{2},k+1} = \frac{\mathbf{u}^{*,k} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+\frac{1}{2},k}$$

$$\nabla \cdot \mathbf{u}^{n+1,k} = 0$$

Outline

- Biological Problem
- Model Components
- Numerical and Communication Methods
 - Fluid Solver
 - **Parallel Programming**
 - Immersed Boundary Method
- Future Work

Parallel Programming

- Shared memory vs. distributed memory

Parallel Programming

- Shared memory vs. distributed memory
- MPI

Parallel Programming

- Shared memory vs. distributed memory
- MPI
- Explicit send and receives

Parallel Programming

- Shared memory vs. distributed memory
- MPI
- Explicit send and receives
- Easy to cause program to hang

Parallel Programming

- Shared memory vs. distributed memory
- MPI
- Explicit send and receives
- Easy to cause program to hang
- Communication performance crucial

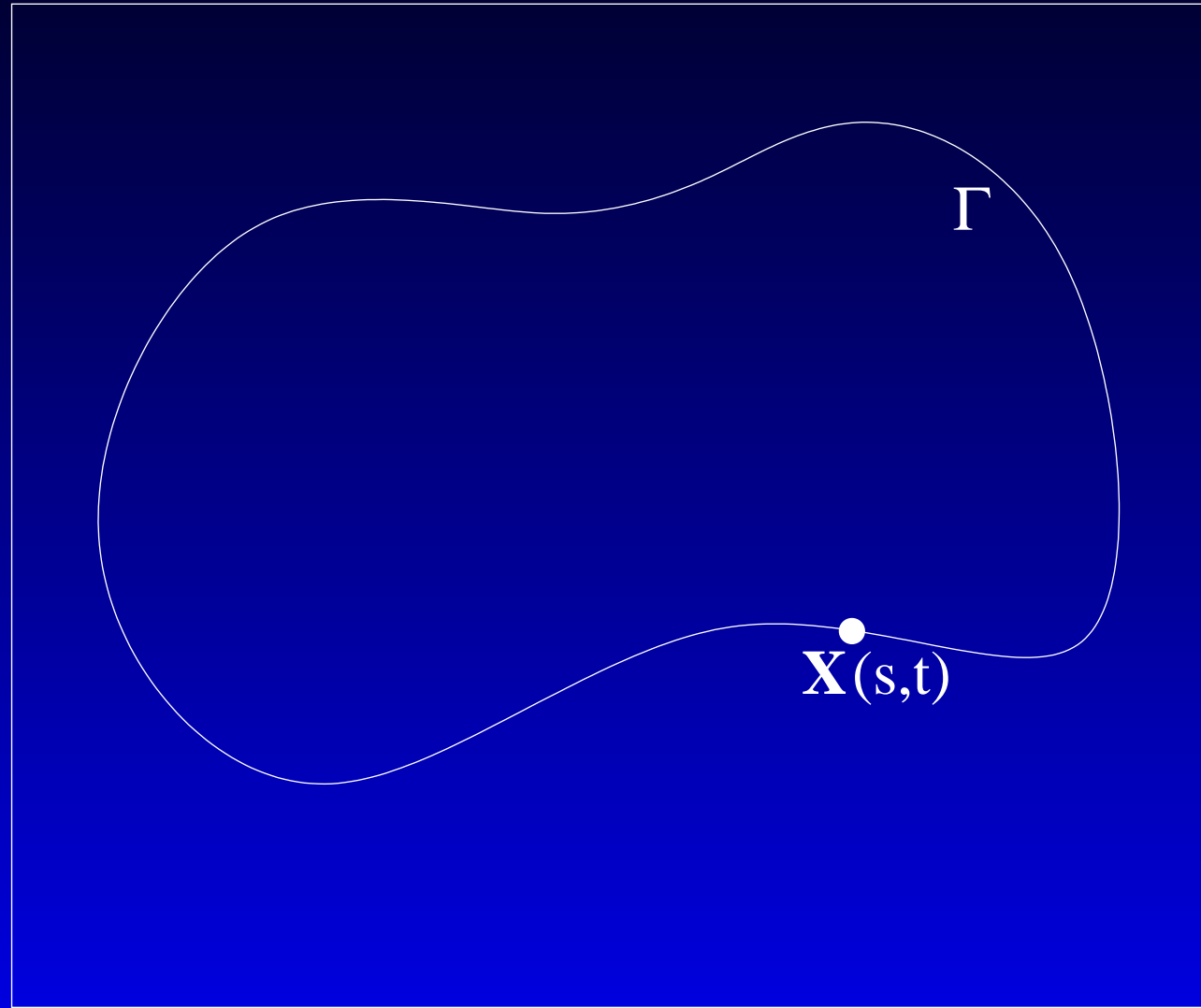
Parallelizing the NS Solver



Outline

- Biological Problem
- Model Components
- Numerical and Communication Methods
 - Fluid Solver
 - Parallel Programming
 - Immersed Boundary Method
- Future Work

Immersed Boundary Method



Immersed Boundary Equations

$$\mathbf{f}(x, t) = \int_{\Gamma} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds$$

$$\frac{d\mathbf{X}}{dt} = \mathbf{u}(\mathbf{X}(s, t), t) = \int_{\Omega} \mathbf{u}(x, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) dx$$

$\mathbf{F}(s, t)$ = Some function of $\mathbf{X}(s, t)$ and its derivatives

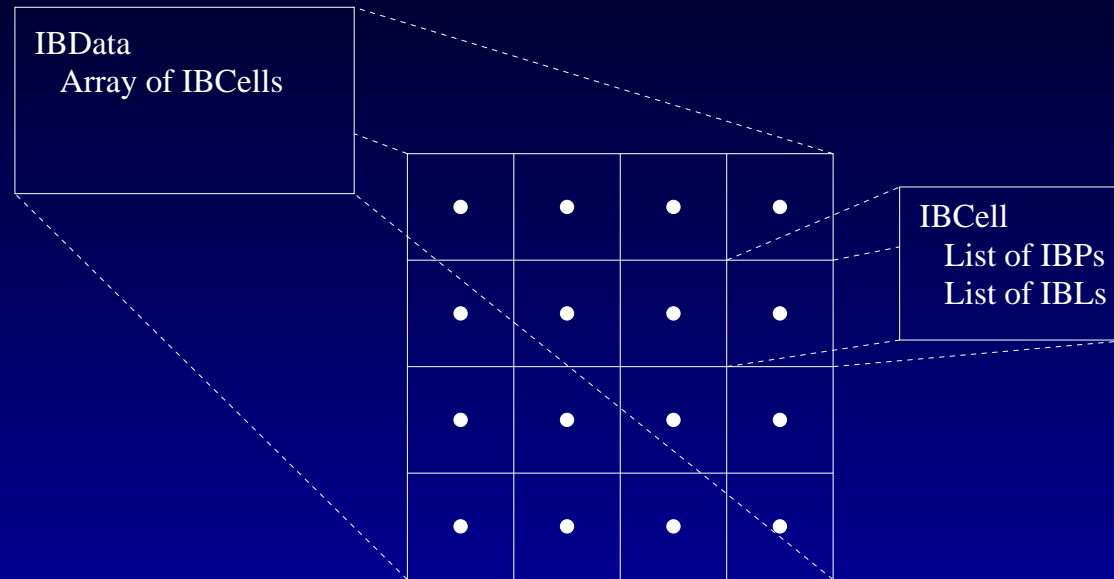
Immersed Boundary Equations

$$\mathbf{f}_{ij} = \sum_k \mathbf{F}_k \delta_h(\mathbf{x}_{ij} - X_k)$$

$$\frac{d\mathbf{X}_k}{dt} = \mathbf{U}_k = \sum_{ij} \mathbf{u}_{ij} \delta_h(\mathbf{x}_{ij} - X_k) h^2$$

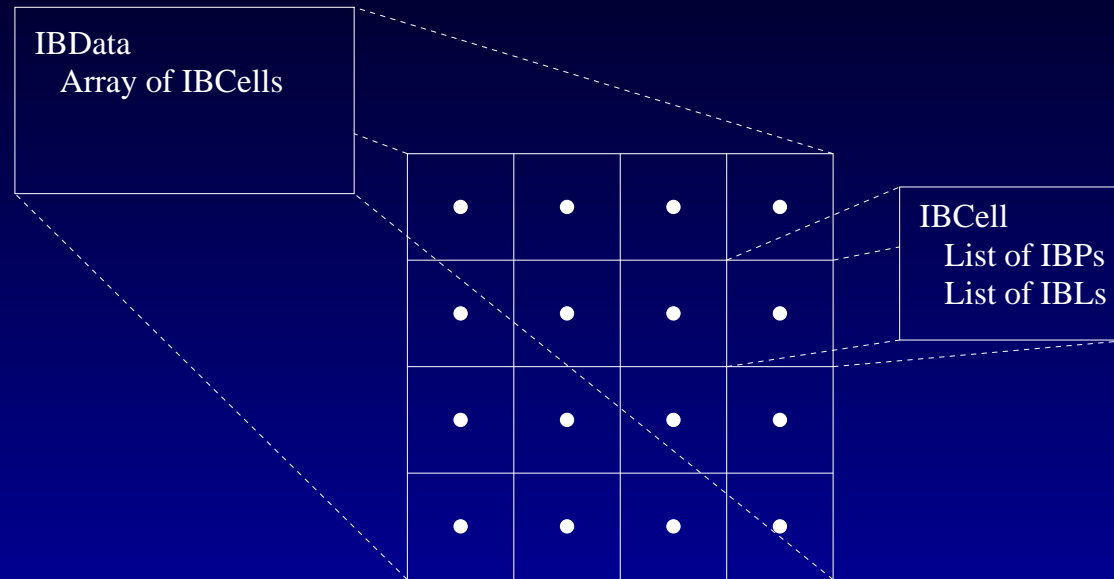
$$\mathbf{F}_k = \sum_i T_0 (\|X_i - X_k\| - \ell_0) \frac{X_i - X_k}{\|X_i - X_k\|}$$

Data Structures



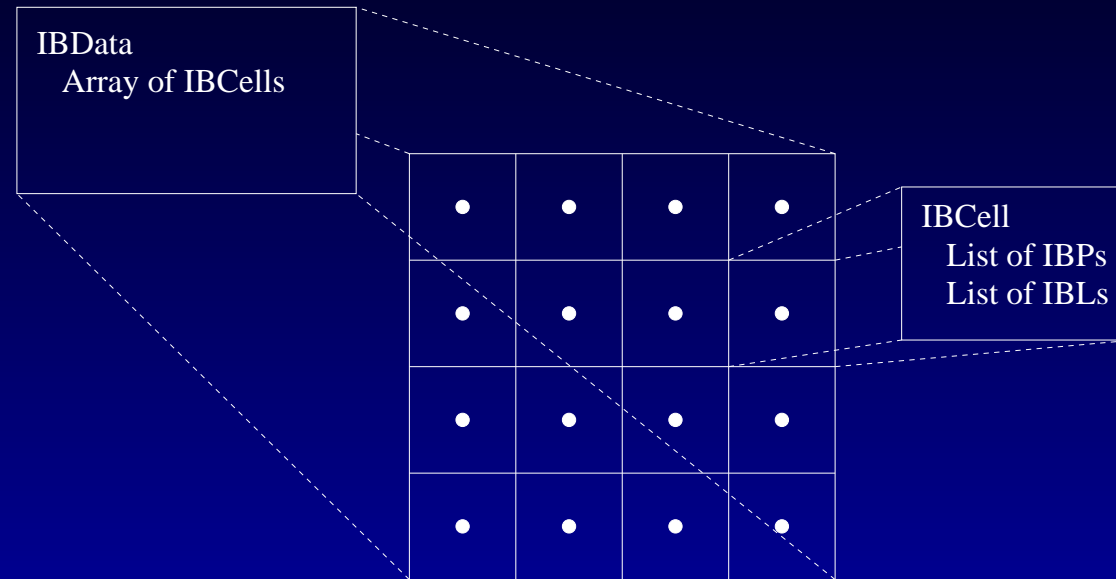
- SAMRAI

Data Structures



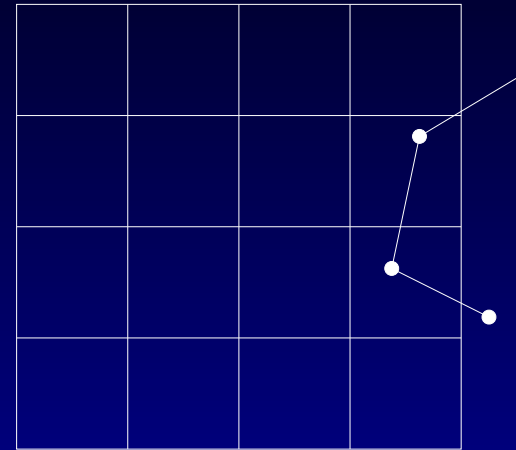
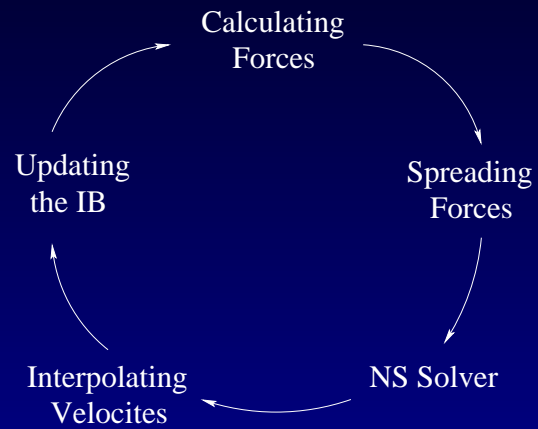
- SAMRAI
- IBP and IBL access

Data Structures



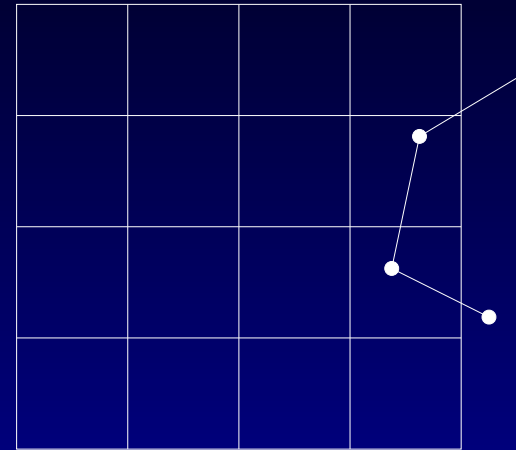
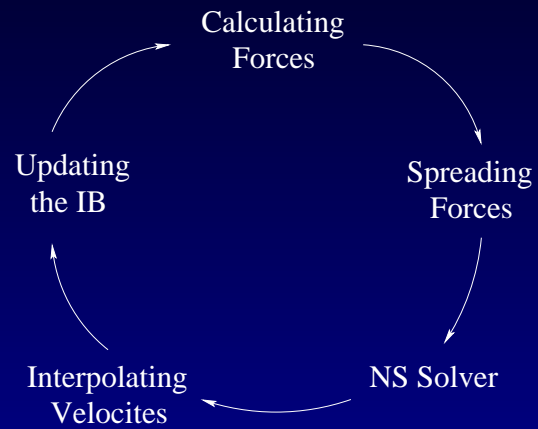
- SAMRAI
- IBP and IBL access
- Non-uniqueness of IBLs

Calculating Forces



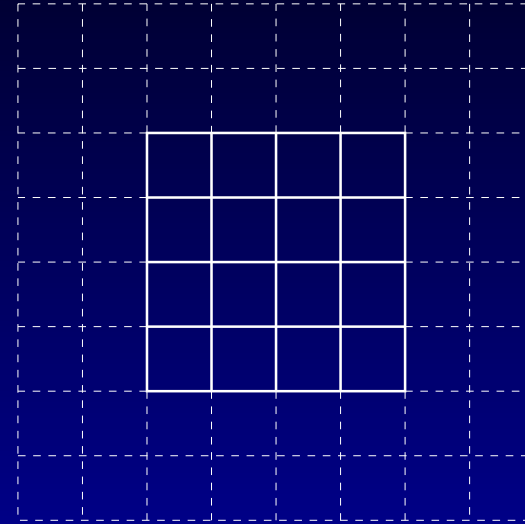
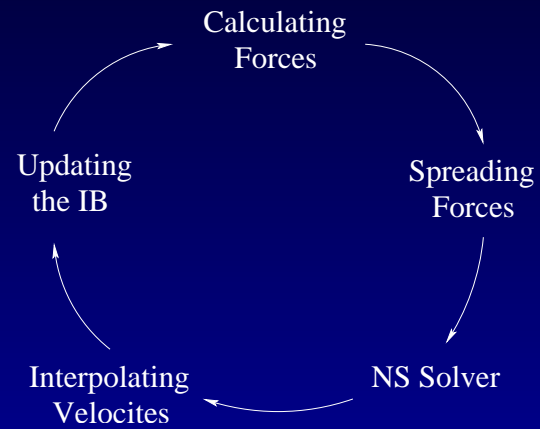
- Obtaining non-local IBPs

Calculating Forces



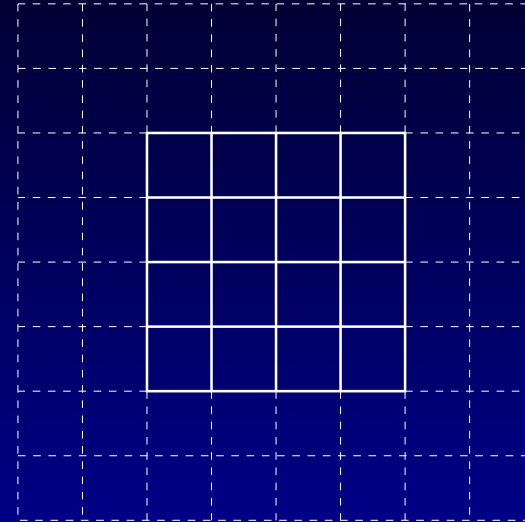
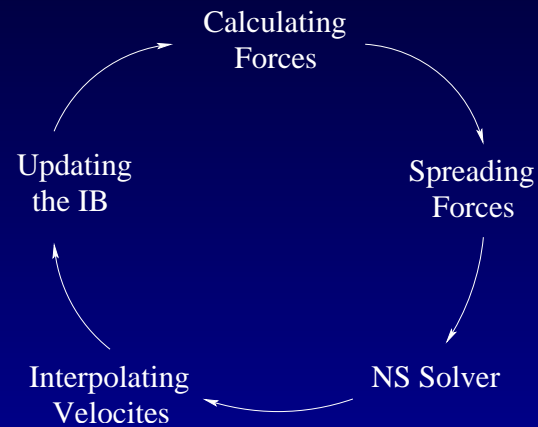
- Obtaining non-local IBPs
- Smarter dummies

Spreading Forces



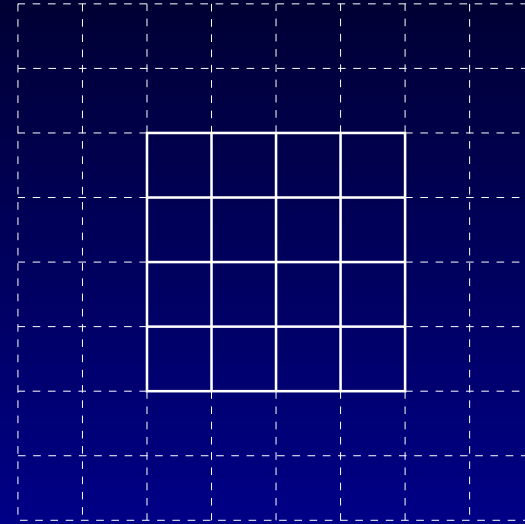
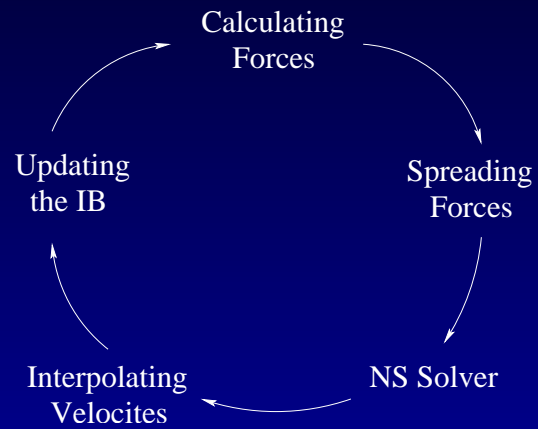
- Delta function has four width stencil

Spreading Forces



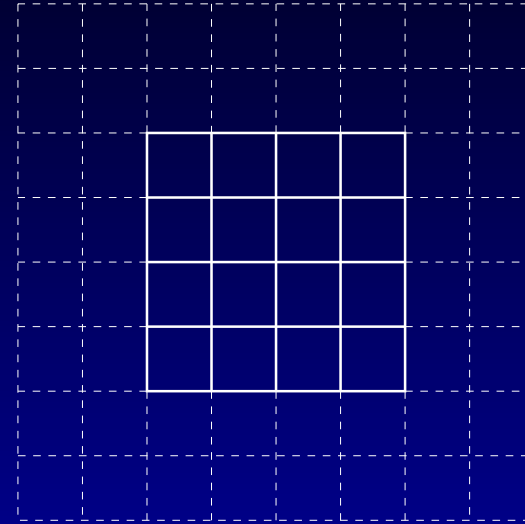
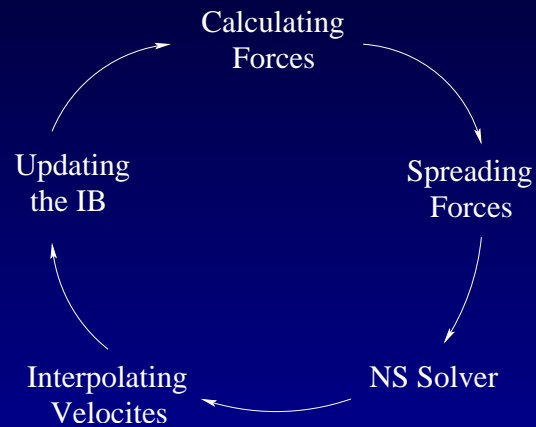
- Delta function has four width stencil
- Need two ghost cells of IBPs with corresponding forces

Interpolating Velocities



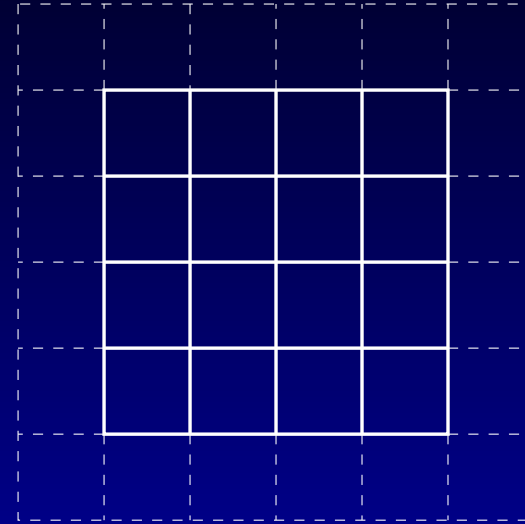
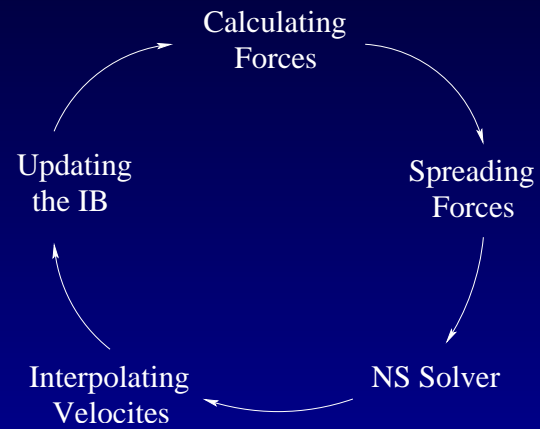
- Delta function has four width stencil

Interpolating Velocities



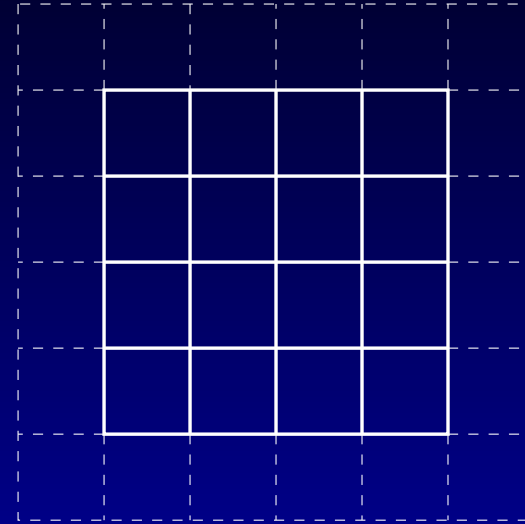
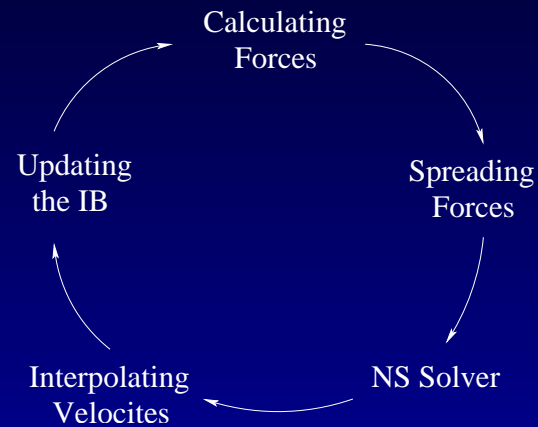
- Delta function has four width stencil
- Need two ghost cells of cell centered velocities

Updating the IB Location



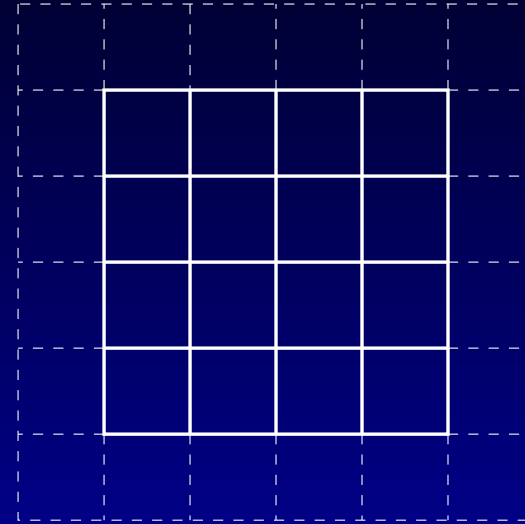
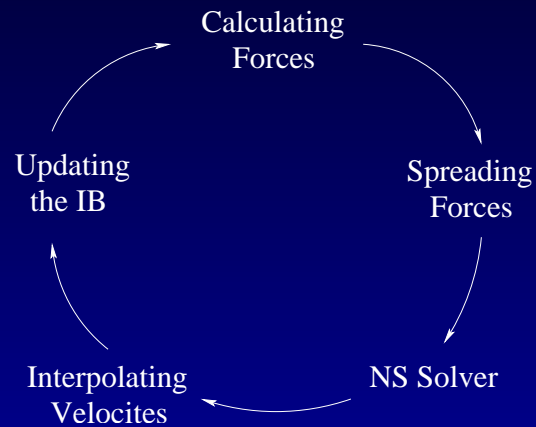
- CFL condition

Updating the IB Location



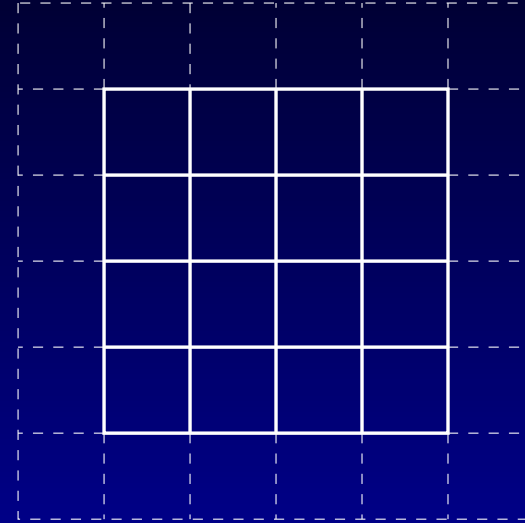
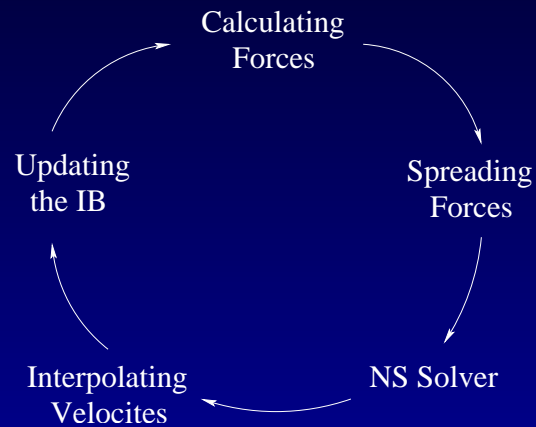
- CFL condition
- Connecting IBLs to IBPs

Updating the IB Location



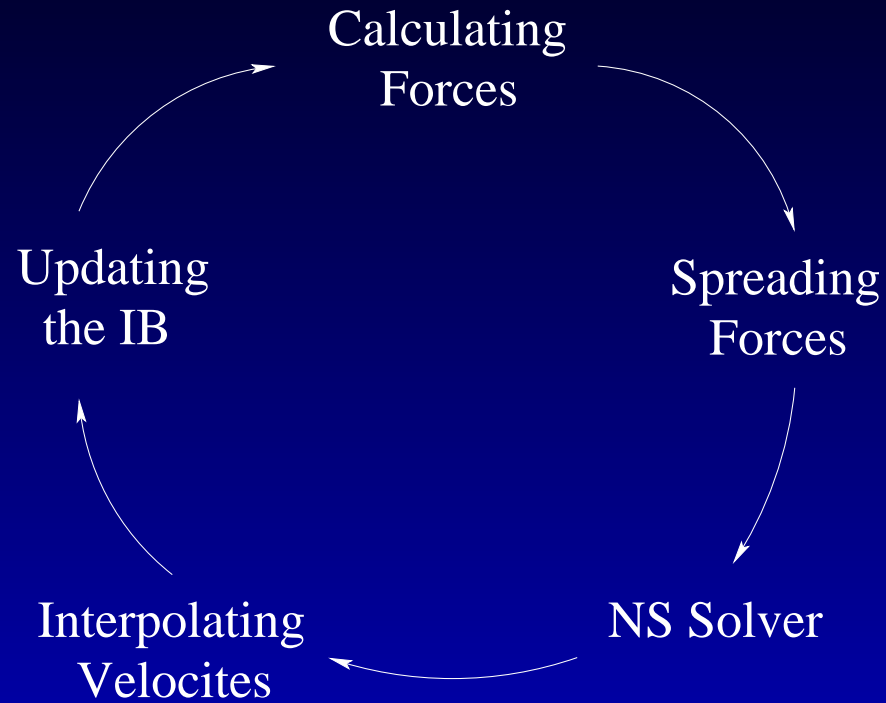
- CFL condition
- Connecting IBLs to IBPs
- “Long Distance” Communication (LDC)

Updating the IB Location



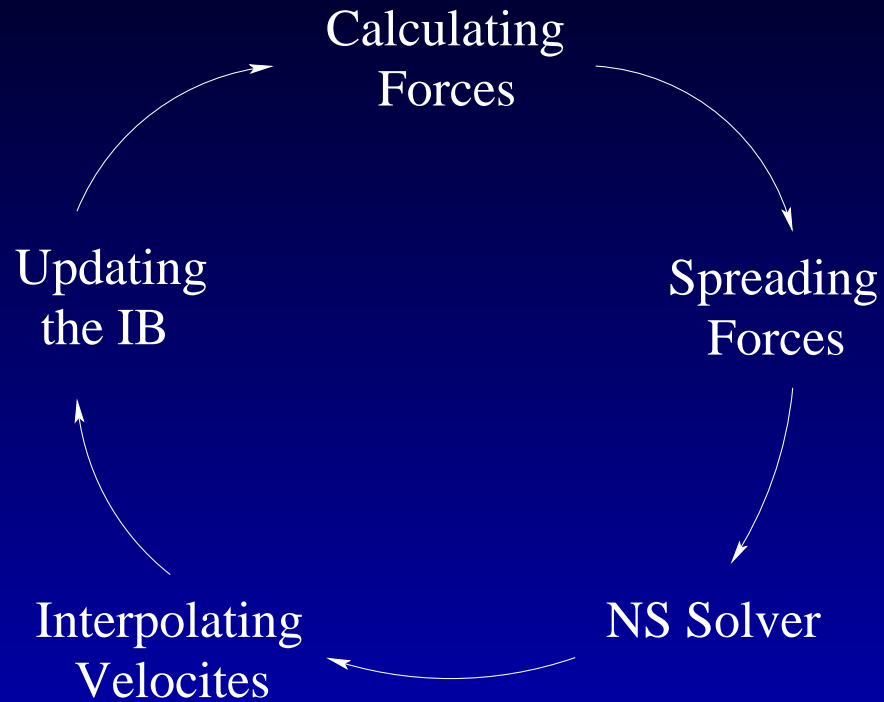
- CFL condition
- Connecting IBLs to IBPs
- “Long Distance” Communication (LDC)
- Deletion of extra IBLs and IBPs

Optimizations



- Minimizing size of the dummy list

Optimizations



- Minimizing size of the dummy list
- Coalescing of communication steps

Alternate Methods

- Make IBPs and IBLs global among processors
 - Easier to implement
 - Requires an all-to-all broadcast message
 - Allows use of II method (or hybrid IB/II method)
 - Reduces four messages to two

Alternate Methods

- Make IBPs and IBLs global among processors
- Make IBPs and IBLs “global per interface”
 - Allow use of II method (or hybrid IB/II method)
 - Requires a “reduced” all-to-all broadcast message
 - Still requires normal spreading and interpolation communication
 - Still requires LDC for links between interfaces
 - Increases number of messages required from four to five

Outline

- Biological Problem
- Model Components
- Numerical and Communication Methods
- Future Work

Future Work

Future Work

- Debugging

Future Work

- Debugging
- Immersed Interface Method

Future Work

- Debugging
- Immersed Interface Method
- Improving Stability of Implicit Solves

Future Work

- Debugging
- Immersed Interface Method
- Improving Stability of Implicit Solves
- Forces Between Connecting Interfaces

Future Work

- Debugging
- Immersed Interface Method
- Improving Stability of Implicit Solves
- Forces Between Connecting Interfaces
- Modeling of Red Blood Cells

Future Work

- Debugging
- Immersed Interface Method
- Improving Stability of Implicit Solves
- Forces Between Connecting Interfaces
- Modeling of Red Blood Cells
- Chemical Interactions

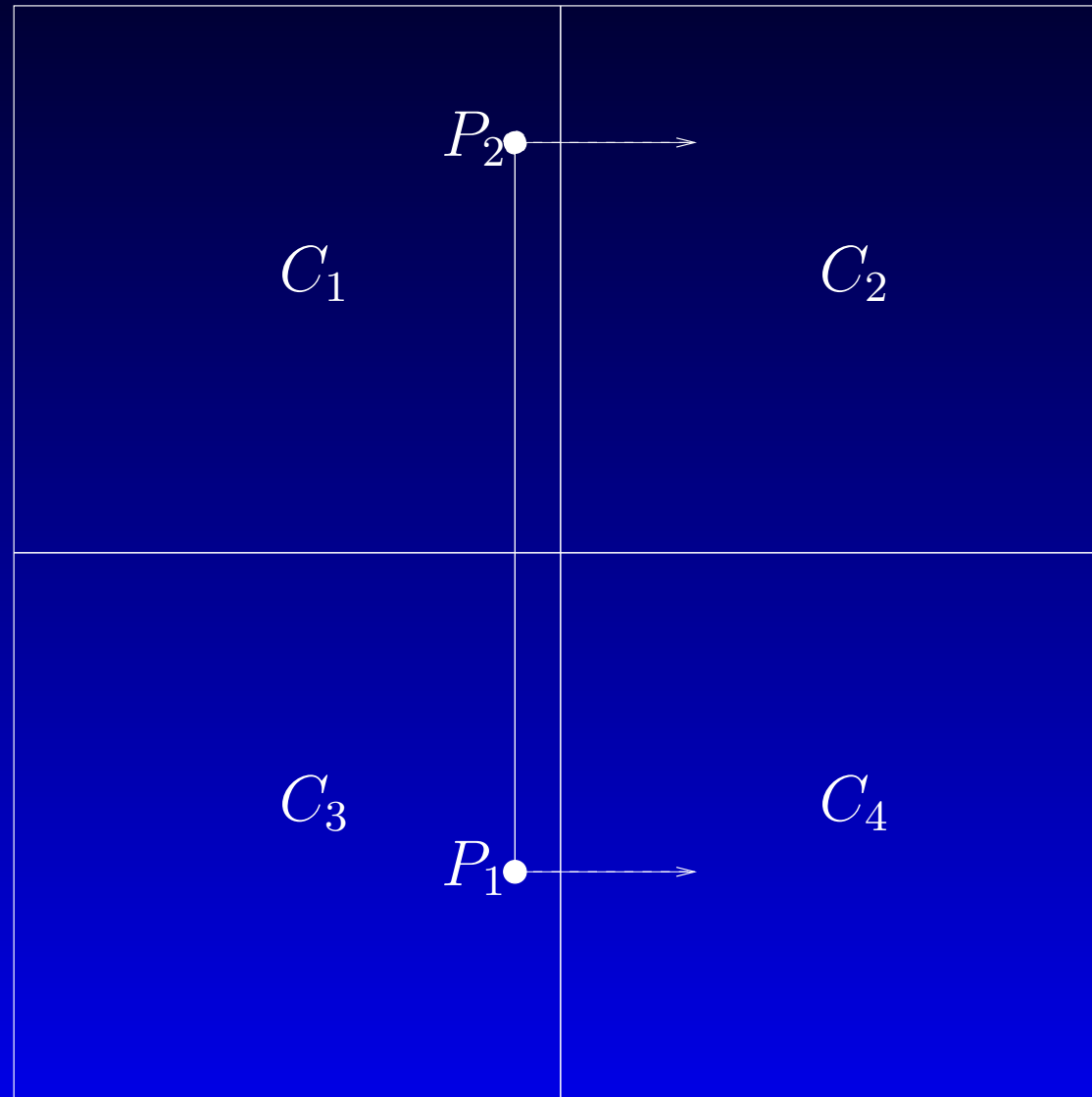
Future Work

- Debugging
- Immersed Interface Method
- Improving Stability of Implicit Solves
- Forces Between Connecting Interfaces
- Modeling of Red Blood Cells
- Chemical Interactions
- Adaptive Mesh Refinement

Conclusion

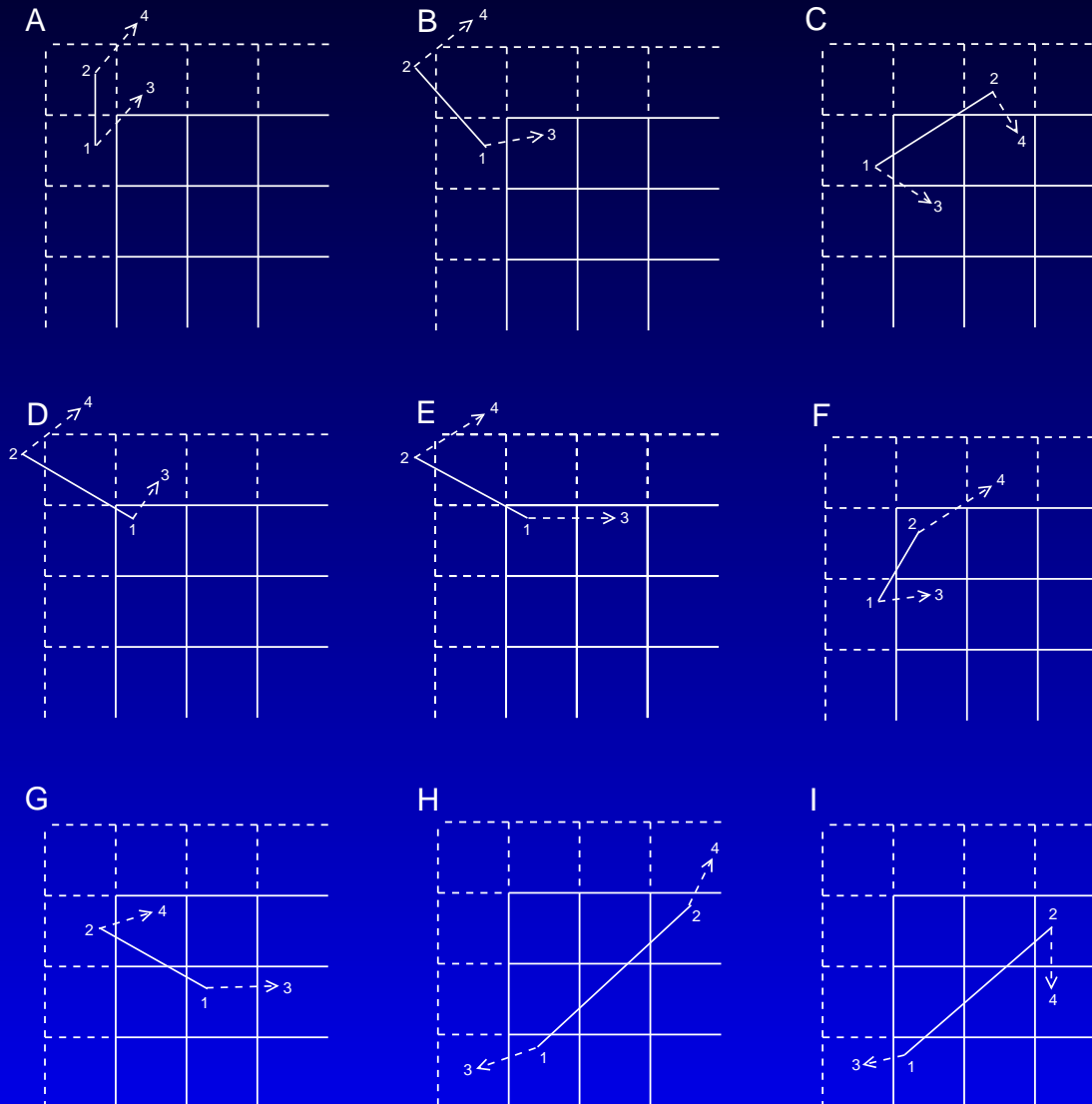
THE END

LDC



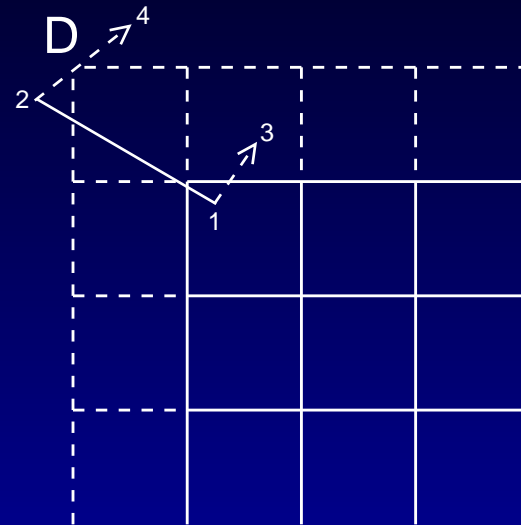
Return

LDC



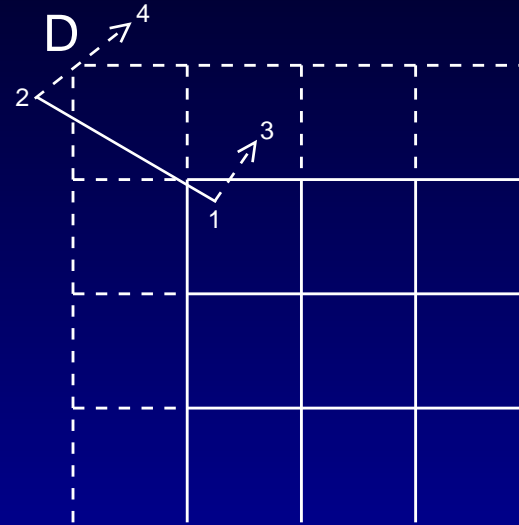
Return

LDC



[Return](#)

LDC



- 1 if O_2 does not update pt 1, then *me* needs to send him pt 3 (LDC₁)
- 2 if *me* does not update pt 2 and $O_3 \neq O_2$ then *me* needs to receive pt 4 from O_2 (LDC₁)
- 3 if O_3 does not update pt 2 and $O_4 \neq O_3$, then *me* needs to send pt 4 to O_3 . (LDC₂)

Return

Preliminary Data Structures

An IBData object contains

- Array of IBCells

An IBPoint is composed of

- Spatial coordinates (location)
- Whether or not IBPoint is tethered
- Force & Velocity vectors

An IBCell contains

- List of IBPoints
- List of IBLinks

An IBLink is composed of

- Resting Length
- Tension
- Whether a periodic boundary is crossed

[Return](#)

Final Data Structures

An IBData object contains

- Array of IBCells
- List of dummy IBPoints

An IBPoint is composed of

- Unique identifier
- Spatial coordinates (location)
- Force & Velocity vectors
- Whether or not IBPoint is tethered
- Owner & Local link count

An IBCell contains

- List of IBPoints
- List of IBLinks

An IBLink is composed of

- Unique identifier
- Resting Length
- Tension
- Whether a periodic boundary is crossed
- Pointers to two IBPoints

Return