



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. М. В. ЛОМОНОСОВА

Физический факультет

КУРСОВАЯ РАБОТА ПО ДИСЦИПЛИНЕ

"ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ ДЛЯ РЕШЕНИЯ РЕСУРСОЁМКИХ ЗАДАЧ ФИЗИКИ"

Использование технологий параллельного
программирования
в моделировании тромбоза

Канева Валерия Николаевна

2 курс, группа 219

Научный руководитель:

_____ Д. Ю. Нечипуренко

« ____ » _____ 2015 г.

Москва, 2015 г.

Содержание

1	Аннотация	2
2	Введение	2
3	Теория	3
3.1	Физическая постановка задачи	3
3.2	Описание модели	4
3.3	Параллельная версия программы	4
4	Результаты и их анализ	6
4.1	Результаты распараллеливания	6
4.2	Физические результаты	8
5	Заключение	10
6	Вывод	10

1 Аннотация

Работа посвящена созданию параллельной версии программы моделирования тромбоза. Анализируется корректность физических результатов и полученное ускорение. Оцениваются дальнейшие перспективы.

2 Введение

Кровеносная система млекопитающих имеет сложное устройство: она включает в себя сердце и разветвлённую сеть сосудов – артерий, вен, капилляров, по которым кровь течёт с большими скоростями и давлениями. Благодаря этому кислород и питательные вещества доставляются к органам и тканям тела, а продукты метаболизма удаляются. Нарушение целостности сосудов может привести к кровопотерям и попаданию посторонних микроорганизмов в систему кровообращения. Проблема остановки кровотечения является одной из жизненно важных в организме. Обычно в гемостазе выделяют следующие звенья: сосудистый гемостаз, тромбоцитарный гемостаз, свёртывание и фибринолиз. Данная работа посвящена моделированию явлений тромбоцитарного гемостаза.

Тромбоциты - безъядерные клеточные фрагменты 2-4 мкм диаметром, способные активироваться в присутствии коллагена, тромбина, АДФ, тромбоксана А₂ и даже при любом значительном возмущении окружающей среды. [3] При повреждении стенки сосуда они способны образовывать тромбоцитарную пробку, прикрепляясь к пострадавшим участкам (адгезия) и друг к другу (агрегация), таким образом останавливая кровотечение. Однако при определённых нарушениях в сердечно-сосудистой системе агрегат может образовываться внутри сосуда - его обычно называют тромбоцитарным тромбом. Эти сгустки препятствуют свободному току крови и могут иметь тяжёлый исход для организма к примеру, они являются центральными факторами в таких клинических случаях как инфаркт и инсульт.

Инфаркт - некроз органа, вызванный недостатком кровообращения. Наиболее важную роль тромбоз играет в таком частном случае инфаркта как инсульт.

Инсульт - острое нарушение мозгового кровообращения. Причиной ишемического инсульта является непосредственно закупорка сосуда тромбом или сужение артерий. Геморрагический инсульт - кровоизлияние в головной мозг, например, из-за разрыва сосуда, причиной которого может быть именно образование тромба. В России заболеваниями, вызывающим тромбоз, подвержены более 30 млн. человек, при этом от тромбоза ежегодно умирают более 200 000 человек. В среднем же по миру тромбоз встречается у 1-2 человек на 1000 населения в год [2].

Тромбоцитарный тромбоз является серьёзной проблемой современной медицины и активно изучается. Методов исследования *in vivo* и *in vitro* недостаточно. Компьютерное моделирование позволяет проверить предполагаемую модель явления, предсказать поведение системы в тех или иных условиях, сложно воспроизводимых в лаборатории.

3 Теория

3.1 Физическая постановка задачи

Опишем процесс образования тромбоцитарного тромба. Если бы тельца были распределены равномерно по высоте сосуда, их адгезия к месту повреждения была бы затруднена, а значит должна существовать сила, которая смещает частицы к стенкам. Эту функцию выполняют эритроциты - красные клетки крови, значительно превосходящие тромбоциты как по концентрации, так и по массе. Их постоянные столкновения приводят к перераспределению концентрации тромбоцитов - её увеличению на периферии - и перемешиванию частиц. Экспериментально доказано, что присутствие эритроцитов увеличивает скорость роста агрегата в несколько раз.

Но тромбоциты движутся с большими скоростями - порядка 1000 мкм /сек - для того, чтобы частица прикрепилась к месту повреждения требуется быстро и аккуратно её остановить. Этой цели служат расположенный на тромбоцитах рецептор - гликопротеин *Ib-V-IX* и циркулирующий в крови фактор фон Виллебранда, способный связываться с коллагеном и тромбоцитами тромба, покрывая их. Пролетающие мимо тромбоциты при помощи рецептора цепляются за слабо связанный фактор и плавно, но быстро останавливаются.

Обратимо связанный тромбоцит может оторваться, однако активация стабилизирует его. Частицы первого слоя активируются коллагеном и прочно с ним связываются. Тромбоциты следующих слоёв активируются растворёнными активаторами, секретитируемыми другими активированными тромбоцитами или образованными в процессе каскада свёртывания. Они связываются с частицами тромба при помощи фибриногеновых мостиков.[3]

Фибрин, образующийся в ходе реакций плазменной системы свертывания, полимеризуясь, формирует сеть, в которой запутываются тромбоциты. Образуется красный кровяной сгусток, содержащий в том числе другие клетки крови (например, эритроциты). Этот процесс обеспечивает плотную закупорку повреждённого сосуда.[4]

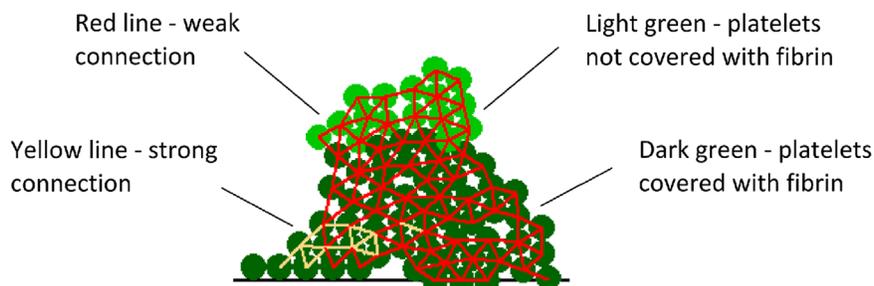


Рис. 1: Структура тромба. Воспроизведено из [5]

3.2 Описание модели

В данной программе моделируется процесс тромбообразования: расчёт ведётся в течение определённого количества итераций - "программных единиц времени". На каждом шаге вычисляются силы, действующие на каждую из частиц, рассчитываются скорости, координаты и некоторые другие параметры.

Модель образования агрегата в программе имеет значительные упрощения, однако получаемая картина соответствует тому, что наблюдается в лабораторных опытах. В ней учитываются следующие силы, действующие на тельца:

- Упругие силы отталкивания тромбоцитов от стенок сосуда
- Упругие силы отталкивания тромбоцитов друг от друга
- Силы вязкости, действующие со стороны потока (в этой модели поле скоростей квадратичное)
- Эффективная сила со стороны эритроцитов, смещающих тромбоциты к стенкам (от координат и времени не зависит)
- Силы связей между частицами в тромбе

Последний пункт заслуживает особого внимания. В данной модели предполагается, что тромбоциты слабо активированы, образуются фибриногеновые мостики, однако связи обратимы. Лишь первый слой частиц прикреплён к зоне повреждения так, что их уже нельзя оторвать от стенки (на момент начала работы программы этот слой уже образован).

Фибриногеновые мостики имитируются жесткими пружинами. Считается, что каждой частице принадлежит две пружины и если они (частицы) находятся достаточно близко (на расстоянии трёх радиусов и меньше), то их пружины с определённой вероятностью могут соединиться друг с другом, образуя связь между тельцами. С другой стороны, пружины могут и разрываться - это случайный процесс.

Концентрация свободных тромбоцитов поддерживается постоянной.

3.3 Параллельная версия программы

Было решено распараллелить программу при помощи стандарта OpenMP - выбор был обусловлен наличием сильной зависимости по данным в вычислениях. Параллельная область включает в себя весь цикл по времени, в том числе вывод данных в файл.

Создающие сложности для распараллеливания программы детали:

- В программе используются переменные, актуальные значения которых должны быть доступны всем потокам:

- Число тромбоцитов, находящихся правее расчётной зоны: они могут вернуться в расчёт слева, если концентрация свободных частиц будет меньше должной
 - Количество свободных тромбоцитов в расчётной области
 - Полное количество тромбоцитов в расчётной области
 - Элементы массива с номерами соединённых пружин
- В блоке расчёта взаимодействий каждый поток изменяет не только силу, действующую на частицу, соответствующую данному потоку, но и силу, действующую на тромбоцит с ним взаимодействующий.

Решения:

- Директива *atomic* позволяет корректно изменять значение переменной, глобальной для всех потоков. *flush()*, по умолчанию присутствующий на входе и выходе в *atomic* обеспечивает доступность актуального значения для любого потока[6]
- Во втором случае отсутствует необходимость в том, чтобы было известно актуальное значение переменных: они используются лишь после отработки блока расчёта сил. Эту ситуацию можно сравнить с задачей о нахождении числа π методом Монте-Карло: каждый поток увеличивает переменную отвечающую за количество точек, попавших в окружность. В такой ситуации мы использовали бы *reduction(+:sum)*. При использовании *reduction(+:sum)* для каждого потока создаётся копия переменной, он с ней работает, и при завершении работы цикла значения суммируются в глобальную переменную, а локальные(поточковые) уничтожаются. Однако информация о силах и моментах сил хранится в одномерных массивах, а значит директиву использовать нельзя. Но можно реализовать нечто подобное для массивов. Так как расчёт сил выполняется на каждом шаге по времени, создавать и уничтожать массивы не имеет смысла. Создадим для каждого одномерного массива двумерный в начале программы: первый индекс пробегает те же значения, что и индекс одномерного массива, а второй - от нуля до количества потоков.

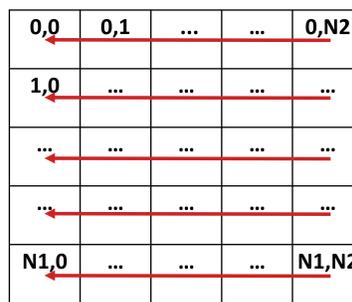


Рис. 2: Редукция массива. $N1$ - длина исходного массива, $N2$ - количество потоков

4 Результаты и их анализ

4.1 Результаты распараллеливания

В программе есть множество параметров, от которых зависит её работа в общем и ускорение параллельной версии в частности. Приведём здесь несколько графиков и проанализируем их.

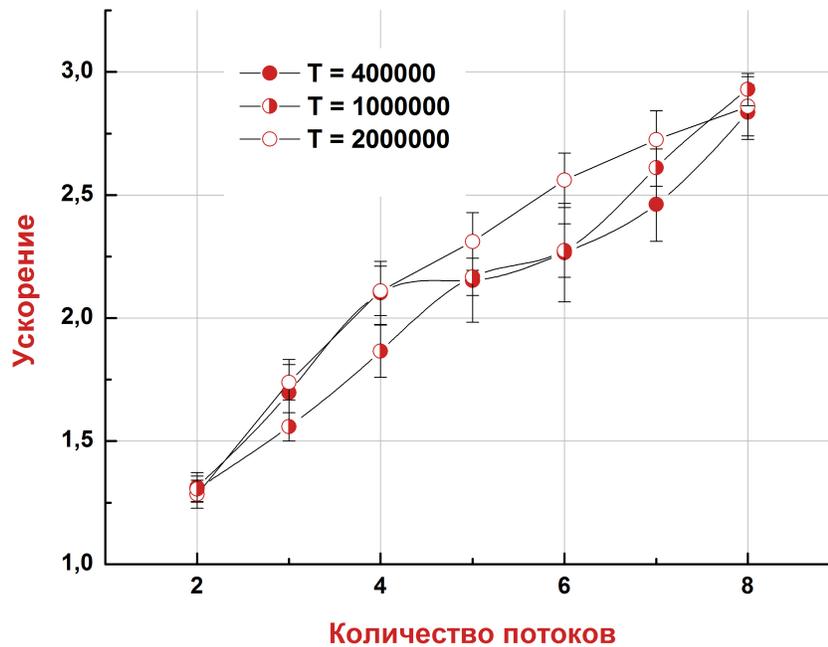


Рис. 3: Графики ускорения при различных количествах временных итераций. Программная единица соответствует 1 мс физического времени. Максимальные времена работы: 4, 12, 27 минут соответственно

Можно обратить внимание на большие погрешности по оси ускорений: время работы оценивалось по пяти измерениям. Редкие результаты, сильно отличающиеся от остальных (как в меньшую, так и в большую сторону), были исключены из расчёта, потому что необходимо было оценить именно среднее ускорение. В программе значительную роль играют случайные процессы, поэтому нельзя предсказать, как ускорится программа при одновременном запуске. Однако глядя на график можно заметить общую тенденцию: ускорение тем больше, чем больше количество временных итераций. С чем это связано? Это связано с тем, что чем дольше наблюдение, тем больше вырастает в размерах тромб (см. подраздел "Физические результаты"). А так как концентрация свободных частиц поддерживается постоянной, возрастает количество тромбоцитов, участвующих в расчёте. Таким образом увеличивается вычислительная сложность, а значит распределение работы по потокам становится более эффективным.

Рассмотрим следующий график.

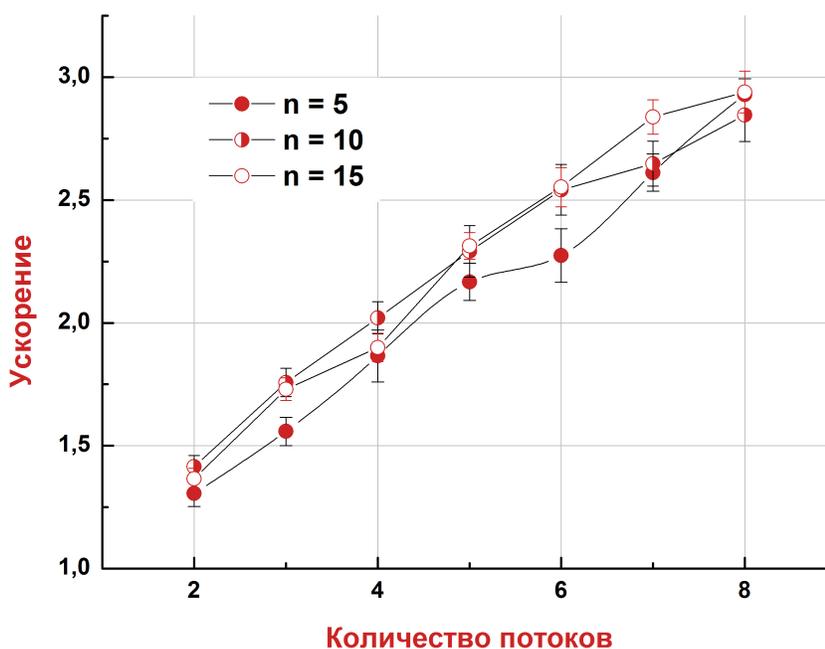


Рис. 4: Графики ускорений при разных концентрациях свободных частиц. Максимальные времена работы: 12, 18, 16 минут соответственно

В целом можно заметить тенденцию: ускорение тем больше, чем больше концентрация свободных частиц. Причина этому, опять же, в увеличении объёма вычислений при росте числа тромбоцитов.

В идеальном случае ускорение должно быть пропорционально или практически пропорционально количеству потоков. Однако видно, что ускорение на восьми thread-х примерно равно трём. В чем причина?

Как упоминалось ранее, параллельная область включает в себя ту часть программы, в которой происходит запись в файл. Необходимо на протяжении всего расчёта (а если конкретнее - на каждой тысячной временной итерации) записывать все координаты, скорости, и другие параметры частиц. Это необходимо для последующей визуализации. Тем не менее, существуют такие задачи, для которых в фиксирование всех этих величин не является необходимым. Пример такой задачи рассмотрен в разделе "Физические результаты".

Будет ли отличаться ускорение в таком случае? Ведь запись происходит последовательно. Проведя измерения времени работы, можно прийти к выводу, что данный фактор не даёт значительный вклад в уменьшение ускорения.

Значит, должны существовать другие причины. Вернёмся к тем способам, которые мы избрали для решения проблем распараллеливания.

- **Использование директивы *atomic*.** Выражение в *atomic* выполняется атомарно, а на входе и выходе выполняется *flush()* для этой переменной.[6] Это не может не замедлить программу, тем более, что директива используется 7-10 раз за временную итерацию.
- **Вспомогательные массивы.** На каждой временной итерации перед областью расчёта взаимодействий обнуляются двумерные вспомогательные массивы, а после - значения из них определённым образом складываются и экспортируются в основные одномерные массивы. Процесс этого распределения происходит последовательно, что отрицательно влияет на эффективность распараллеливания.

Однако несмотря на все недостатки, эти способы работают и дают ускорение. Заменить их на что-то другое в нашем представлении не является возможным.

4.2 Физические результаты

Один из наиболее важных результатов этой программы - выходные файлы, описывающие движение частиц. С помощью них можно визуализировать процесс. Во многом именно визуально определяется корректность результата.

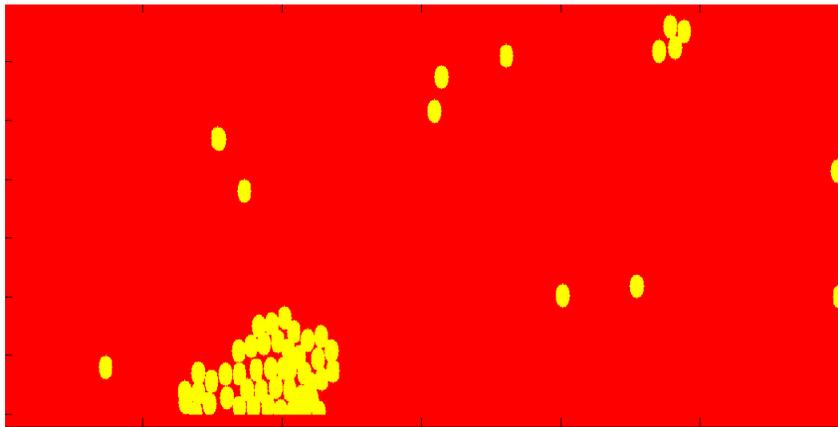


Рис. 5: Визуализация

Программа позволяет исследовать поведение системы в зависимости от различных параметров, например, констант, отвечающих за взаимодействие тромбоцитов друг с другом и со стенками сосуда, коэффициентов вязкости жидкости, скорости потока и так далее.

В качестве примера рассмотрим задачу о среднем и максимальном количествах тромбоцитов в агрегате в зависимости от времени наблюдения за системой.

Видно, что скорость роста как среднего, так и максимального количества тромбоцитов в агрегате, падает со временем, и сами величины стремятся к константе. В первую очередь это связано с распределением скорости: она убывает от центра к краям, поэтому когда высота тромба значительно вырастает, он размывается потоком. По графику можно оценить: закупорит ли агрегат весь сосуд в данных условиях?

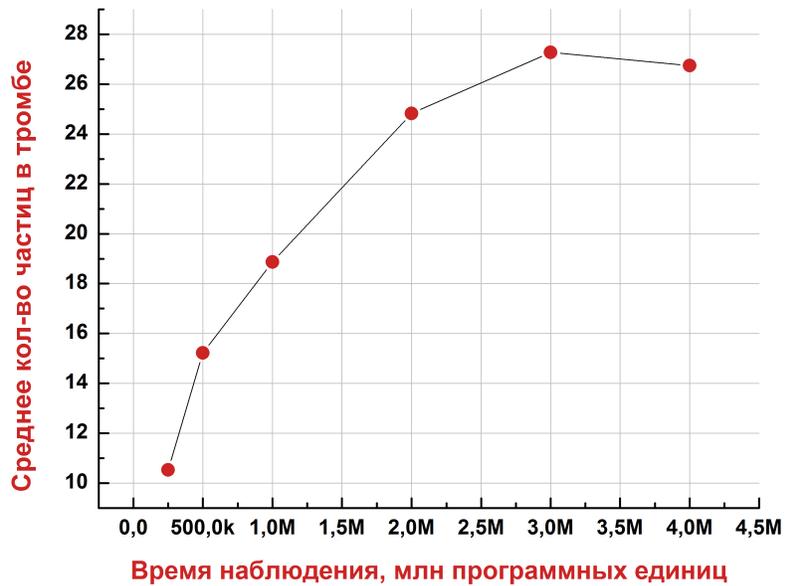


Рис. 6: График зависимости среднего размера тромба от времени наблюдения. Одна единица программы соответствует 1 мс физического времени.

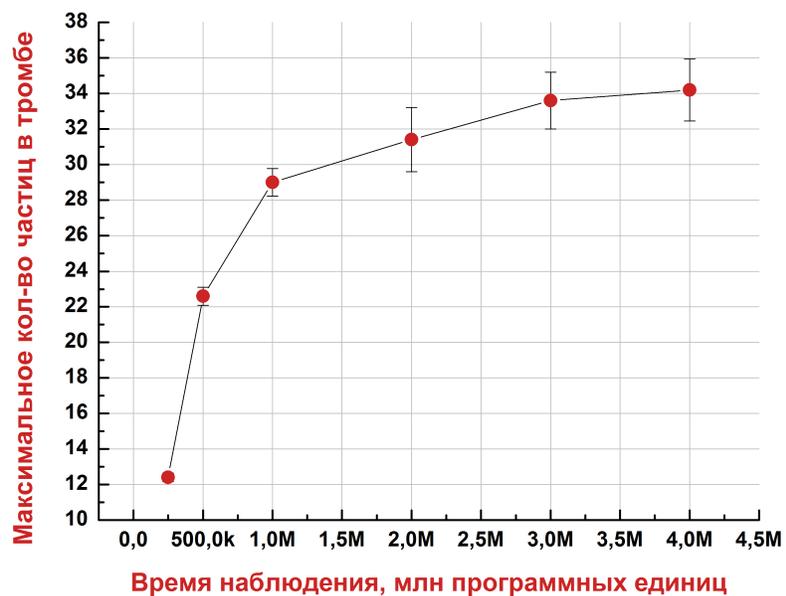


Рис. 7: График зависимости максимального размера тромба от времени наблюдения. Одна единица программы соответствует 1 мс физического времени.

Радиус частицы - 1 единица программы, высота сосуда - 35. Максимальное количество тромбоцитов оценим как 35 (см. Рис.5). Если выстроить тромбоциты в высоту, весь сосуд перекроют 18. Однако такой тромб - слишком неустойчивая конструкция. Опыт показывает, что агрегат из 35 частиц не может закупорить сосуд.

Такой результат совпадает с результатом последовательной программы.

Все вычисления были проведены с использованием ресурсов суперкомпьютерного комплекса МГУ имени М.В. Ломоносова.[7]

5 Заключение

Эта задача является не конечной целью, а промежуточным результатом, необходимым для проверки корректности методов распараллеливания и оценки ускорения. В результате обработки данных расчётов мы пришли к выводу, что результаты в необходимой степени соответствуют действительности.

Следующим этапом должна стать программа, моделирующая тромбоз в трёхмерном пространстве, что усложнит вычисления, а значит эффективность распараллеливания должна возрасти. То, что даже в двумерной модели было получено заметное ускорение, свидетельствует о том, что создание более сложной параллельной версии программы оправдано.

Также необходимо учитывать, что поле скоростей в реальности иное. На данный момент существует два способа решения этой проблемы: использование пакета для расчёта гидродинамики или написание программы вручную. В результате мы пришли к выводу, что первый способ не подходит для двумерной модели, так как все расчёты скоростей в таком случае будут последовательными, а значит ускорение будет ещё ниже. Это означает, что написание программы для расчёта гидродинамики в сосуде может оправдать себя, если подойти к этому с учётом параллельности всего процесса. В отношении трёхмерной модели данный вопрос требует дальнейших исследований.

Также планируется усложнение модели: необходимо в большей степени учитывать биохимические процессы, происходящие при повреждении стенки сосуда.

6 Вывод

В результате работы была написана корректно работающая параллельная версия программы, и получено заметное ускорение. Также на основании результатов можно предположить, что написание трёхмерной усложнённой версии программы будет оправдано.

Список литературы

- [1] М.А.Пантелеев, Ф.И.Атауллаханов. *Свёртывание крови: биохимические основы*. Клиническая онкогематология - том 1, номер 1 (2008)
- [2] <http://www.tearx.ru/>
- [3] М.А.Пантелеев, А.Н.Свешникова. *Тромбоциты и гемостаз*. Онкогематология, 2 (2014)
- [4] Физиология человека: в 3-х томах. Т.2.Пер. с англ/Под ред. Р.Шмидта и Г.Тевса - 3-е изд. - М.: Мир, 2005
- [5] Tosenberger A, Ataulakhanov F, Bessonov N, Panteleev M, Tokarev A, Volpert V *Modelling of thrombus growth in flow with a DPD-PDE method*. J Theor Biol. 2013 Nov 21.
- [6] *OpenMP Application Program Interface Version 3.1* July 2011
- [7] Воеводин Вл.В., Жуматий С.А., Соболев С.И., Антонов А.С., Брызгалов П.А., Никитенко Д.А., Стефанов К.С., Воеводин Вад.В. *Практика суперкомпьютера "Ломоносов" // Открытые системы*. - Москва: Издательский дом "Открытые системы N 7, 2012. С. 36-39.