

УДК 004.94:004.942

SIBERNETIC: ПРОГРАММНЫЙ КОМПЛЕКС НА БАЗЕ АЛГОРИТМА PCI SPH, ОРИЕНТИРОВАННЫЙ НА ЗАДАЧИ МОДЕЛИРОВАНИЯ В ОБЛАСТИ БИОМЕХАНИКИ ЖИВЫХ СИСТЕМ

© 2014 г. А.Ю. Пальянов, С.С. Хайрулин

Федеральное государственное бюджетное учреждение науки Институт систем информатики им. А.П. Ершова Сибирского отделения Российской академии наук, Новосибирск, Россия,
e-mail: palyanov@iis.nsk.su

Поступила в редакцию 16 сентября 2014 г. Принята к публикации 31 октября 2014 г.

Описан программный комплекс *Sibernetic*, разрабатываемый нами в рамках работы над международным проектом *OpenWorm*, нацеленным на создание биологически обоснованной компьютерной модели нематоды *Caenorhabditis elegans*, включая ее нервную систему. Для моделирования тела *C. elegans* и фрагмента внешней среды, в которой оно могло бы перемещаться под действием мышечных сокращений, понадобился алгоритм, способный рассчитывать динамику жидкости, эластичного тела, водонепроницаемых эластичных пленок и мышечных волокон. За основу был взят алгоритм *PCI SPH* – модификация известного алгоритма «гидродинамики сглаженных частиц», позволяющая моделировать несжимаемую жидкость, который мы реализовали в параллельной форме (*OpenCL*) и существенно дополнили функционалом, ориентированным на задачи моделирования в области биомеханики живых систем. В данной работе описываются возможности *Sibernetic*, проиллюстрированные на примере разработанной нами модели тела нематоды *C. elegans*, снабженного мышечной системой, чрезвычайно актуальной ввиду неизменно высокого уровня интереса к исследованию и моделированию этого организма среди нейробиологов и специалистов по биомеханике беспозвоночных. С помощью предложенного подхода эта задача впервые была решена со столь высокой степенью реалистичности и детализации.

Ключевые слова: математические модели в науках о живом, программное обеспечение для научных исследований, *PCI SPH*, метод частиц, гидродинамика, биомеханика беспозвоночных, *C. elegans*, высокопроизводительные параллельные вычисления, *OpenCL*.

ВВЕДЕНИЕ

Симулятор *Sibernetic* (sibernetic.org) был создан в рамках работы над международным проектом *OpenWorm*, целью которого является создание биологически обоснованной компьютерной модели (виртуального организма) нематоды *Caenorhabditis elegans*, включающей тело организма, сенсорную, нервную и мышечную системы, а также фрагмент внешней среды, в которой действуют необходимые физические законы. Центральный объект интереса *OpenWorm* – изучение фундаментальных принципов, лежащих в основе работы нервной системы *C. elegans*, однако ограничиться моделирова-

нием одной лишь нервной системы в рамках данной задачи не представляется возможным. Необходимы и сенсорная система, чтобы обеспечить поток входящих сигналов, и мышечная система, дающая телу способность передвигаться, подчиняясь командам нервной системы; и фрагмент внешней среды, в котором будет происходить перемещение нематоды, приводя, в свою очередь, к изменению сенсорных сигналов (Palyanov *et al.*, 2012; Cohen, Sanders, 2014).

Интерес к моделированию *C. elegans* понятен – это самый изученный на сегодня многоклеточный организм, состоящий из 959 клеток, с нервной системой, насчитывающей всего 302 нейрона, образующих около 7000 межней-

ронных соединений. Несмотря на кажущуюся простоту, эта нематода воспринимает мир посредством хемо-, термо-, механо- и фоточувствительных рецепторов, имеет довольно широкий спектр поведенческих паттернов, обладает зачатками памяти и способна к формированию условных рефлексов (Rankin, 2004). Многие исследователи разделяют мнение, что именно *C. elegans* может оказаться ключом к пониманию фундаментальных принципов работы нервной системы. Сенсорная система *C. elegans* способна воспринимать механические воздействия различного характера, концентрации широкого спектра химических веществ, температуру окружающей среды и степень освещенности в видимом диапазоне. Все эти факторы будет необходимо учесть при создании виртуального организма и изначально принимать во внимание при разработке его модели.

На первом этапе мы взяли за решение задачи, связанной с моделированием тела червя, оснащенного мышечной системой, а также внешней среды. В настоящее время создан работающий прототип, имеющий высокий уровень детализации и точности моделирования физических характеристик тела нематоды и его окружения, включающий в себя следующие компоненты: 1) водонепроницаемая оболочка тела, заполненная жидкостью под давлением (гидростатический скелет); 2) полный набор из 95 мышечных клеток тела, каждая из которых расположена на своем месте и представлена десятками мышечных волокон; 3) реалистичная модель фрагмента внешней среды, позволяющая сочетать твердые, эластичные и жидкие объекты и учитывающая все необходимые силы:

вязкость, трение, поверхностное натяжение, силу тяжести и др.

В статье рассматриваются возможности симулятора *Sibernetic*, достигнутые результаты и дальнейшие перспективы.

МЕТОДЫ

Моделирование динамики жидкости

В качестве базового метода моделирования динамики жидкости был выбран метод SPH (гидродинамика сглаженных частиц) (Gingold *et al.*, 1977), который базируется на описании динамики сплошных сред, введенном Лагранжем. Жидкость представляется набором дискретных элементов – частиц, которые характеризуются координатами в трехмерном пространстве, скоростью и массой. Эти частицы имеют пространственное расстояние (известное как «длина сглаживания», обычно обозначаемая как h), на котором их свойства «сглаживаются» функцией ядра. Любая скалярная физическая величина в точке \mathbf{r} рассчитывается как взвешенная сумма вкладов от всех частиц, находящихся в пределах длины сглаживания:

$$A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h),$$

где A_j – значение величины A для частицы j , ρ_j – плотность, связанная с частицей j , h – радиус сглаживания и W – сглаживающая функция ядра.

В табл. 1 представлены широко используемые, хорошо зарекомендовавшие себя ядра сглаживания, которые мы используем для расчета ряда физических величин.

Таблица 1

Ядра сглаживания, используемые в *Sibernetic*

Величина	Ядро сглаживания
Плотность	$W_{poly6}(x_{ij}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - x_{ij}^2)^2, & 0 \leq x_{ij} \leq h \\ 0, & \text{иначе} \end{cases}$
Давление	$W_{spiky}(x_{ij}, h) = \frac{15}{\pi h^6} \begin{cases} (h - x_{ij})^3, & 0 \leq x_{ij} \leq h \\ 0, & \text{иначе} \end{cases}$
Вязкость	$W_{viscous}(x_{ij}, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{x_{ij}^3}{2h^3} + \frac{x_{ij}^2}{h^2} + \frac{h}{2x_{ij}^2} - 1, & 0 \leq x_{ij} \leq h \\ 0, & \text{иначе} \end{cases}$

Движение несжимаемой жидкости описывается системой дифференциальных уравнений Навье-Стокса:

$$\rho \left(\frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \right) \mathbf{v} = -\nabla p + \mu \nabla \cdot (\nabla \mathbf{v}) + \mathbf{f}$$

(сохранение момента),

$$\frac{\partial}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \text{ (сохранение массы),}$$

где μ – коэффициент вязкости; \mathbf{f} – сумма всех внешних сил, действующих на частицу; \mathbf{v} – скорость частицы, p – давление. Использование частиц существенно упрощает эти уравнения. Число частиц постоянно, и масса каждой из них неизменна, поэтому сохранение массы гарантировано и второе уравнение становится ненужным. Поскольку частицы движутся вместе с жидкостью, полная производная поля скоростей обращается в частную производную скорости по времени, т. е. для системы частиц конвективный член $\mathbf{v} \cdot \nabla \mathbf{v}$ равен нулю (Müller *et al.*, 2003). Для i -й частицы уравнение движения можно записать следующим образом:

$$m_i \frac{\partial \mathbf{v}_i}{\partial t} = \mathbf{F}_i^{\text{pressure}} + \mathbf{F}_i^{\text{external}} + \mathbf{F}_i^{\text{viscosity}}.$$

Сила давления и сила вязкости рассчитываются по следующим формулам (Müller *et al.*, 2003):

$$\mathbf{F}_i^{\text{pressure}} = -\frac{m_i}{\rho_i} \frac{p_i + p_j}{2} \nabla W(\mathbf{r}_{ij}, h),$$

$$\mathbf{F}_i^{\text{viscosity}} = \frac{m_i}{\rho_i} \sum_j \frac{m_j}{\rho_j} \frac{\mu_i + \mu_j}{2} (\mathbf{v}_j - \mathbf{v}_i) \nabla^2 W(\mathbf{r}_{ij}, h).$$

Базовый алгоритм SPH не предполагает несжимаемости моделируемой жидкости, однако для решения этой проблемы было предложено несколько модификаций стандартного метода SPH: WCSPH (Becker *et al.*, 2007), ISPH (Shao *et al.*, 2003) и PCI SPH (Solenthaler *et al.*, 2009a). При выборе оптимального варианта мы ориентировались как на минимизацию вычислительных затрат в единицу времени и возможность работы симуляции при относительно большом временном шаге, так и на приемлемый уровень точности получаемого решения. Всем этим требованиям отвечает метод PCI SPH (predictor-corrector incompressible smoothing particle hydrodynamics), при котором несжимаемость жидкости обеспечивается путем использования схемы «предиктор–корректор» для определения давления в точках пространства, занимаемых частицами. Для этого информация о флуктуаци-

ях плотности активно распространяется через жидкость, и значения давления обновляются до тех пор, пока значения плотности во всей системе не станут удовлетворительными. При таком подходе исчезает необходимость решать уравнение Пуассона для давления, требующее значительных вычислительных затрат, и в то же время остается возможность корректной работы симуляции с большим временным шагом, более чем на порядок большим, чем при использовании метода WCSPH.

Моделирование эластичного тела

Эластичный материал представляется в виде сложной упорядоченной системы частиц, связанных между собой эластичными неразрывными соединениями (пружинами) (Terzopoulos *et al.*, 1988). Каждая частица в такой системе обладает свойствами жидкости (на нее действуют все внутренние силы, характерные для частиц жидкости, такие как вязкость, давление и проч.), но при этом на них дополнительно действует сила упругости (в соответствии с законом Гука):

$$\mathbf{F}_i^{\text{elastic}} = -k \sum_j \frac{\mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|} (\|\mathbf{r}_{ij}\| - \|\mathbf{r}_{ij}^0\|),$$

где k – коэффициент упругости, $\|\mathbf{r}_{ij}\|$ – расстояние между частицами i и j , $\|\mathbf{r}_{ij}^0\|$ – равновесная длина пружины, j – индекс j -й эластичной частицы, с которой i имеет эластичное соединение. Коэффициент упругости может быть задан независимо для каждой пары частиц i и j , соединенных пружиной.

Также на основе обычного эластичного тела создан дополнительный тип «материи», способный сокращаться под действием проходящего внешнего сигнала в соответствии с его уровнем. В этом случае на пару частиц, i и j , будет дополнительно воздействовать соответствующая внешняя сила, сближающая эти частицы, направленная вдоль соединяющей их прямой.

Неподвижные объекты и границы

Механизм обработки взаимодействия движущихся объектов с неподвижными границами базируется на методе, описанном в работе M. Ihmsen с соавт. (2010). Граница представляется в виде набора особых частиц, расположенных

в узлах решетки с заданным шагом; для каждой такой частицы используются те же характеристики, что и для обычных частиц, но в отличие от них граничные частицы неподвижны в течение всей симуляции и для каждой из них задан вектор нормали. При попадании i -й частицы в поле действия граничных частиц (это происходит в случае, если $\|\mathbf{r}_i - \mathbf{r}_b\| \leq r_0$, где $r_0 = h/2$) на нее начинают влиять силы отталкивания, пропорциональные расстояниям от частицы до границы; координаты и скорость частицы в этом случае корректируются следующим образом:

$$\mathbf{v}_i(t + \Delta t) = \varepsilon [\mathbf{v}_i^*(t + \Delta t)]_t - \delta [\mathbf{v}_i(t)]_n$$

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i^*(t + \Delta t) + \|\mathbf{r}_i^*(t + \Delta t) - \mathbf{r}_b\| \cdot \mathbf{n}_b,$$

где $[\mathbf{v}_i(t)]_n = (\mathbf{v}_i(t) \cdot \mathbf{n}_b) \cdot \mathbf{n}_b$ определяет компоненту скорости, нормальную к границе, а $[\mathbf{v}_i^*(t + \Delta t)]_t = \mathbf{v}_i^*(t + \Delta t) - [\mathbf{v}_i^*(t + \Delta t)]_n$ – тангенциальную. ε и δ , лежащие в интервале от 0 до 1, контролируют трение и эластичность столкновения с границей. Величины со звездочкой «*» – исходное значение (до учета столкновения). Индекс « b » – граничная частица (boundary), t – тангенциальная и n – нормальная компоненты вектора.

Возможно, что при контакте с поверхностью частица i оказывается одновременно взаимодействующей с несколькими граничными частицами. В этом случае их вектора нормалей образуют усредненный вектор нормали, в который входят с различным «весом» в зависимости от расстояния между i -частицей и данной граничной частицей (Ihmsen *et al.*, 2010):

$$\mathbf{n}_i^c = \sum_b w_{ib}^c \mathbf{n}_b,$$

$$\text{где } w_{ib}^c = \max\left(0, \frac{\mathbf{r}_0 - \|\mathbf{r}_{ib}^*\|}{r_0}\right),$$

$\|\mathbf{r}_{ib}^*\| = \|\mathbf{r}_i^*(t + \Delta t) - \mathbf{r}_b\|$. Затем координаты и скорость частицы i корректируются следующим образом:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i^*(t + \Delta t) + \frac{1}{\sum_b w_{ib}^c} \sum_b w_{ib}^c \|\mathbf{r}_0 - \mathbf{r}_{ib}^*\| \cdot \frac{\mathbf{n}_i^c}{\|\mathbf{n}_i^c\|}, \quad (1)$$

$$\mathbf{v}_i(t + \Delta t) = \varepsilon [\mathbf{v}_i^*(t + \Delta t)]_t \quad (2)$$

Данный подход позволяет корректно обрабатывать контакт с границей, а кроме того, обеспечивает возможность создавать границы различных форм.

Моделирование непроницаемой мембраны

Одной из оригинальных возможностей симулятора *Sibernetic* является поддержка моделирования непроницаемых эластичных материалов, в том числе однослойных пленок. Многие биологические объекты обладают водонепроницаемыми оболочками, однако при их моделировании посредством обычного эластичного тела наблюдается значительная утечка жидкости сквозь них. Это особенно актуально для оболочки тела *C. elegans*, форма которого поддерживается благодаря «гидростатическому скелету» – внутренней среде, находящейся под избыточным давлением 2–30 кПа (Harris *et al.*, 1957).

В основе моделирования непроницаемых мембран лежит описанный выше метод, предложенный в работе М. Ihmsen с соавт. (2010) для расчетов взаимодействия движущихся объектов с границей, представленной неподвижными частицами. Мы реализовали его в *Sibernetic* в исходном виде для неподвижных границ, а также разработали его модификацию для взаимодействия с границами подвижных эластичных объектов. Для ее реализации был введен дополнительный класс физических примитивов, поддерживаемых симулятором, названный «мембранами». Каждая элементарная мембрана представляет собой объект, составленный из трех соседних частиц эластичного тела (находящихся на его поверхности), образующих треугольник. Тройка индексов частиц, образующих мембрану, указывается явным образом. Окрестность треугольника, образованного тройкой частиц, отталкивает любую частицу жидкости, подошедшую достаточно близко. На каждом шаге симуляции при контакте частицы жидкости с окрестностью мембраны для частицы вычисляется сила отталкивания, которая корректирует ее координаты и скорость.

На каждой итерации при контакте частицы жидкости i с частицей эластичного тела j (т. е. при $\|\mathbf{r}_i - \mathbf{r}_j\| \leq r_0$) определяется число элементарных мембран, в состав которых она входит (т. е. имеющих один из трех своих индексов, j). Если это число больше нуля, то для каждой из этих элементарных мембран вычисляется сначала проекция частицы i на мембрану, затем – вектор нормали к плоскости

мембраны и наконец результирующий вектор нормали \mathbf{n}_i^m , усредненный по всем мембранам. Далее скорость и позиция частицы i корректируются в соответствии с (1) и (2) с использованием \mathbf{n}_i^m вместо \mathbf{n}_i^c .

Параллельная реализация на OpenCL

Один из недостатков симуляции на основе системы частиц по сравнению с сеточными методами состоит в том, что необходимо использовать большое количество частиц для симуляций с эквивалентной разрешающей способностью. Одним из возможных решений этой проблемы является использование параллельного алгоритма вместо последовательного, поскольку даже современные настольные ПК, как правило, располагают многоядерными процессорами или мощными видеокартами, которые было бы разумно использовать «в полную силу».

Моделируемая система состоит из N «независимых» частиц; для каждой отдельной частицы последовательно вычисляются все действующие на нее силы, на основе которых затем вычисляются новые координаты и скорость; все эти действия для каждой частицы происходят независимо и теоретически могут быть выполнены параллельно, так как входными данными для них служат координаты и скорости, полученные на предыдущей ите-

рации. Отсюда следует, что алгоритм хорошо распараллеливается по данным.

Для решения этой задачи мы решили остановиться на платформе OpenCL, предназначенной для создания приложений, связанных с параллельными вычислениями на гетерогенных вычислительных системах (Munshi *et al.*, 2011). Основными преимуществами OpenCL является открытость стандарта, поддержка большинства основных производителей как комплектующих, так и программного обеспечения, например Intel, AMD, NVIDIA (более подробный список – на сайте <http://www.khronos.org/opencl/>), а также возможность исполнять программный код на различных устройствах: GPU, CPU, FPGA.

В табл. 2 представлены результаты сравнения производительности различных реализаций SPH и PCI SPH. Мы использовали данные о Bullet-Fluid и параллельной реализации SPH. Bullet-Fluid реализован на OpenCL и имеет ограничение на максимальное число частиц в симуляции (65536). Исходный код реализации SPH (Goswami *et al.*, 2010) является закрытым, и для сравнения мы использовали табличные данные, представленные в этой статье. Для сравнения производительности с Bullet-Fluid одинаковые конфигурации тестировались на нем и на Sibernetic, а затем вычислялось среднее время, затрачиваемое симуляцией на одну итерацию.

Таблица 2

Результаты сравнения производительности различных реализаций SPH и PCI SPH

Название	Алгоритмы	Устройство, платформа	Среднее время на 1 шаг, мс	Количество частиц
Bullet-Fluid	SPH	GPU NVIDIA Tesla C1060, OpenCL	70	65 536
Goswami <i>et al.</i> , 2010	SPH	GPU NVIDIA GeForce GTX280, CUDA	89	75 200
Solenthaler <i>et al.</i> , 2009b	PCISPH	Intel Core2 Quad, 2.66 ГГц (1 поток)	722	100 000
Solenthaler <i>et al.</i> , 2009b	PCISPH	Intel Core2 Quad, 2.66 ГГц (OpenMP)	218	100 000
Sibernetic	PCI SPH, эластичное тело, мембраны	GPU NVIDIA Tesla C1060, OpenCL	125	65 536
Sibernetic	PCI SPH, эластичное тело, мембраны	CPU Intel Core i7-2700K (3.5 ГГц), OpenCL	139	65 536
Sibernetic	PCI SPH, эластичное тело, мембраны	GPU AMD Radeon R9 290X, OpenCL	82,8	65 536

РЕЗУЛЬТАТЫ

Результатом данной работы, во-первых, является сам программный код *Sibernetic*. Завершив этап разработки программного кода, мы приступили к проверке соответствия результатов, получаемых в симуляции, с точными аналитическими решениями или результатами экспериментов. Ниже приведены несколько проведенных тестов:

1. Тест на корректность поведения частицы в гравитационном поле. Одна частица или эластичное тело свободно падает в пространстве, находясь под действием силы гравитации; на каждой итерации детектируется смещение, которое проделала отдельная частица или центр масс эластичного тела от начала движения до текущей позиции, и сравнивается с точным решением. При этом тесте, в частности, проверяется точность численного интегратора системы уравнений движения.

2. Тест на возникновение вязкого трения. Симулируется стационарное течение жидкости внутри трубки. На некотором ее участке детектируется распределение скоростей в зависимости от удаленности от стенок трубки. Предполагается, что поток жидкости стабилизирован вдоль координаты y , так что $\frac{\partial \mathbf{v}}{\partial y} = 0$ (труба стоит вертикально, имеет гладкие стенки, на жидкость из внешних сил действует только гравитация). Поэтому общие уравнения Навье-Стокса упрощаются до одного двумерного эллиптического уравнения (Лойцянский, 1978; Ландау, Лифшиц, 2001):

$$\frac{\partial^2 \mathbf{v}}{\partial x^2} + \frac{\partial^2 \mathbf{v}}{\partial y^2} = \frac{1}{\mu} \frac{dp}{dy},$$

где \mathbf{v} – скорость потока, μ – вязкость жидкости, p – давление. Предполагается, что на границе с трубой скорость жидкости равна 0. Точное решение показывает, что профиль скоростей описывается параболической кривой.

3. Тест на затухание колебаний пружины. Одиночная эластичная частица или тело, состоящее из множества частиц, подвешено на пружине к неподвижной верхней стенке. Из внешних сил на объект действует только сила гравитации, которая тянет объект вниз. При смещении в пружине, в соответствии с законом Гука, воз-

никает сила, действующая в противоположном направлении, в результате возникают осцилляции. При неправильной работе интегратора амплитуда колебаний пружины будет со временем неограниченно возрастать, тогда как при правильной – колебания должны либо затухнуть через некоторое время при наличии трения в системе, либо осциллировать бесконечно при его отсутствии. Мы реализовали два различных симплектических метода интегрирования, известных своей стабильностью и хорошей точностью для своего класса: полуневяный метод Эйлера первого порядка и Leap-Frog второго порядка.

Проведенные тесты показали, что результаты расчетов *Sibernetic* хорошо согласуются с известными результатами экспериментов или аналитическими данными для рассмотренных конфигураций. Результаты представлены на графиках (рис. 1, 2).

Завершив разработку и тестирование *Sibernetic*, мы смогли приступить к одной из главных целей проекта – проектированию детальной реалистичной модели тела *C. elegans*, представляющей из себя водонепроницаемую эластичную оболочку (воспроизводящую геометрические размеры реальной нематоды), наполненную жидкостью под давлением для поддержания формы. К оболочке крепятся продольные тяжи мышечных клеток, расположенных в четырех боковых квадрантах тела. Результат представлен на рис. 3, а также доступен в форме видеозаписи (<http://www.youtube.com/watch?v=r2OBznK6HkI>).

ЗАКЛЮЧЕНИЕ

Большинство существующих реализаций алгоритмов SPH и PCISPH либо являются независимыми коммерческими продуктами, либо входят в состав платных программных пакетов для создания анимированных сцен, моделирования гидродинамики жидкости и т. д. *Sibernetic* (sibernetic.org) на сегодняшний день является единственной реализацией алгоритма PCI SPH с открытым исходным кодом. Ряд дополнительных оригинальных функций и типов «материи», ориентированных на задачи моделирования в области биофизики, в частности биомеханики беспозвоночных (эффективность которых продемонстрирована на примере моделирования тела

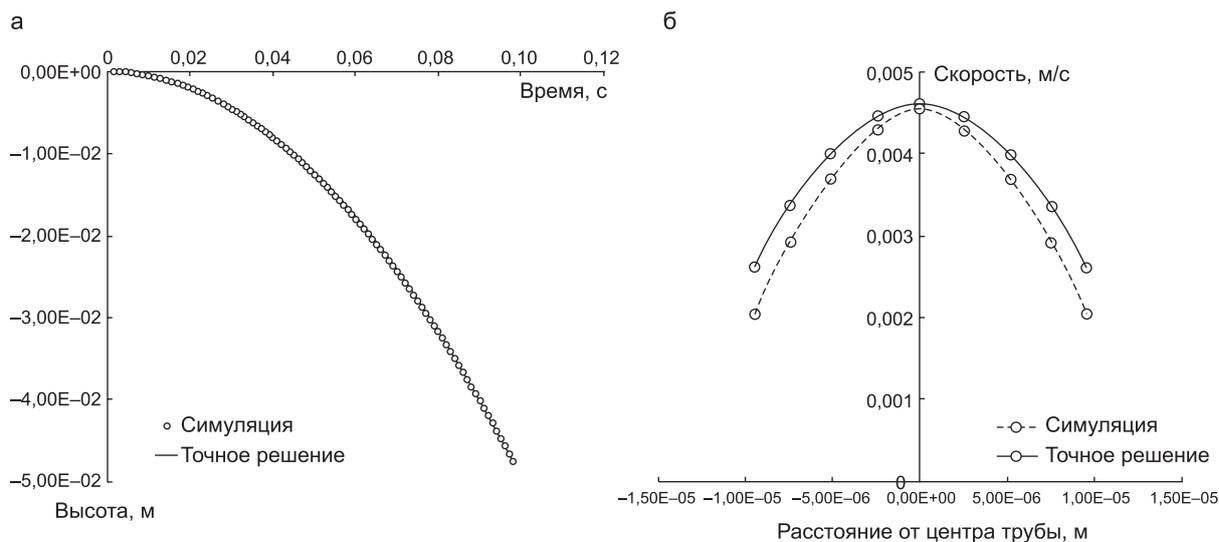


Рис. 1. Результаты тестов.

а – зависимость координаты свободно падающей точечной массы от времени; б – профиль распределения скоростей в трубе (в зависимости от расстояния от центра трубы).

Fig. 1. Test results. Diagram (a) shows the dependence of coordinate of free falling mass on time. Diagram (b) shows the profile of velocities distribution inside the pipe (depending on the distance from the center of the pipe).

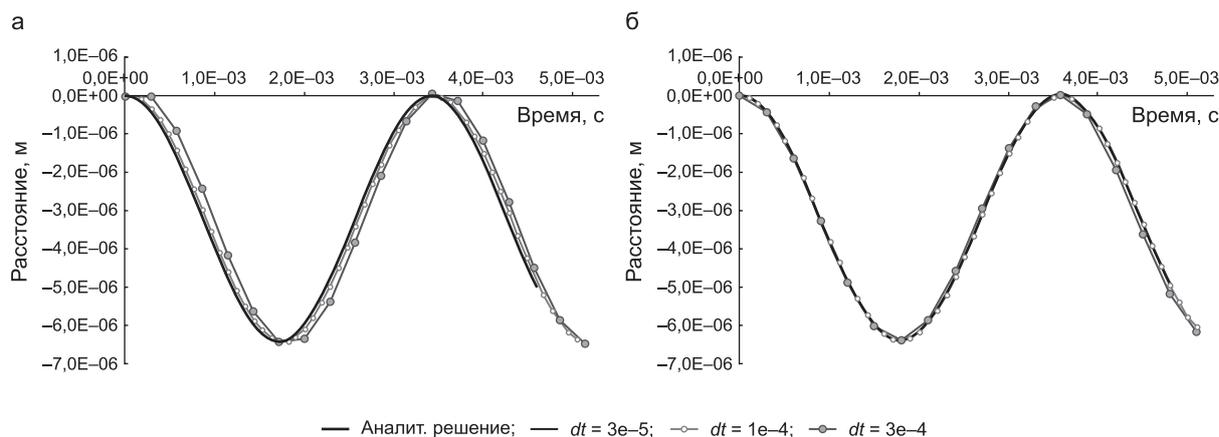


Рис. 2. Сравнение результатов численного моделирования с точным решением задач колебания массы на пружине.

а – результаты, полученные для полуявного метода Эйлера 1-го порядка точности; б – результаты для метода Leap-frog 2-го порядка точности. Черная линия – точное решение, остальные кривые – результаты, полученные при моделировании в Sibernetic при разных значениях временного шага dt .

Fig. 2. Comparison of numerical modeling results with precise solution for an oscillating mass connected to a spring.

а – results for the case of semi-implicit Euler's method (1-st order of accuracy); б – results for the case of using Leap-frog method (2-nd order of accuracy). Black line corresponds to exact solution, all the rest show the results of simulation using Sibernetic at various values of integration time step dt .

C. elegans), значительно увеличивает область возможных практических применений данной разработки. Кроме того, параллельные вычисления на основе OpenCL обеспечивают значительную производительность, а кросс-плат-

форменность (Windows/Linux/macOS), достигнутая благодаря использованию при разработке только стандартных библиотек языка C++, позволяет выполнять код на большинстве современных устройств и операционных систем.

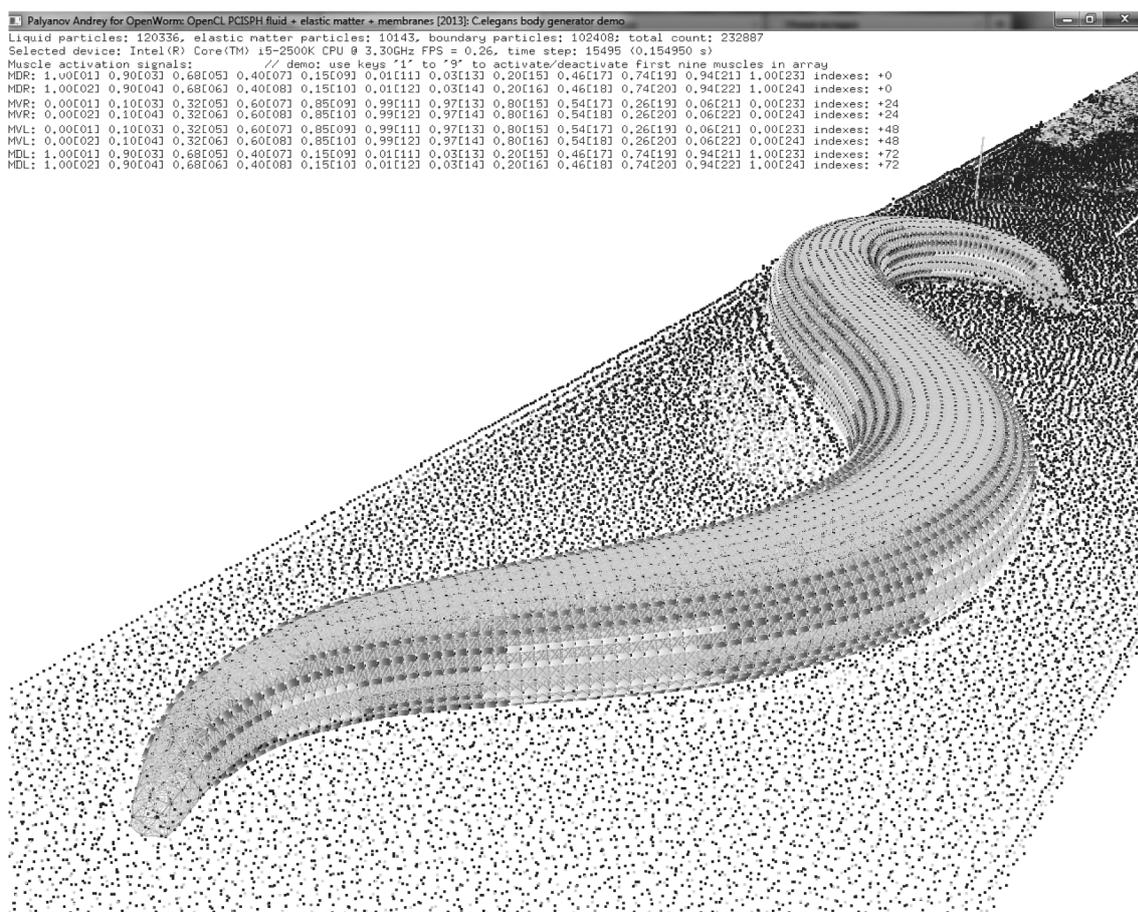


Рис. 3. «Скриншот», иллюстрирующий общий вид симулятора Sibernetic при расчете и визуализации движения модели тела *C. elegans* по поверхности, покрытой тонким слоем жидкости. Команды для мышечных сокращений левой и правой сторон задаются двумя синусоидальными сигналами, находящимися в противофазе.

Fig. 3. General view of a Sibernetic «screenshot» for case of calculation and visualization of *C. elegans* body prototype locomotion over a flat surface covered with a thin layer of water. Muscular contractions are driven by simple sinusoidal patterns opposite in phase between the left and right sides of the body.

БЛАГОДАРНОСТИ

Работа выполнена при поддержке интеграционного проекта СО РАН № 136 «Исследование информационных и молекулярно-генетических механизмов функционирования сетей нейронов на основе экспериментально-компьютерных подходов» и гранта РФФИ № 14-07-31039 «Разработка методологии и алгоритмической базы для задач моделирования в области биофизики живых систем».

ЛИТЕРАТУРА

- Ландау Л.Д., Лифшиц Е.М. Гидродинамика. 5-е изд. Физматлит, 2001. 731 с.
- Landau L.D., Lifshitz E.M. Fluid Mechanics, 5-th edition. M.: Fizmatlit, 2001. 731 p. (In Russian).
- Лойцянский Л.Г. Механика жидкости и газа. М.: Наука, 1978. 736 с.
- Loitsyanskiy L.G. Mechanics of Liquids and Gases. M.: Nauka, 1978. 736 p. (In Russian).
- Becker M., Teschner M. Weakly compressible SPH for free surface flows // Proc. ACM SIGGRAPH/Eurographics SCA. 2007. P. 63–72.
- Cohen N., Sanders T. Nematode locomotion: dissecting the neuronal-environmental loop // Curr. Opin. Neurobiol. 2014. V. 25. P. 99–106.
- Gingold R.A., Monaghan J.J. Smoothed particle hydrodynamics – theory and application to non-spherical stars // Mon. Not. R. Astr. Soc. 1977. P. 375–389.
- Goswami P., Schlegel P., Solenthaler B., Pajarola R. Interactive SPH Simulation and Rendering on the GPU // Proc. of the 2010 ACM SIGGRAPH/Eurographics Symp. on Computer Animation, 2010. P. 55–64.

- Harris J.E., Crofton H.D. Structure and Function in the Nematodes: Internal Pressure and Cuticular Structure in *Ascaris* // J. Exp. Biol. 1957. V. 34. P. 116–130.
- Hut P., Makino J. The Art of Computational Science (2007). Section 4.1 (electronic book). http://www.artcompsci.org/kali/vol/n_body_problem/title.html.
- Ihmsen M., Akinci N., Gissler M., Teschner M. Boundary Handling and Adaptive Time-stepping for PCISPH // Proc. VRIPHYS. Copenhagen, Denmark, 2010. P. 79–88.
- Munshi A., Gaster B., Mattson T., Fung T. OpenCL. Programming Guide. 2011.
- Müller M., Charypar D., Gross M. Particle-based fluid simulation for interactive applications // Proc. of the ACM Sigrgraph/Eurographics Symposium on Computer Animation. 2003. P. 154–159.
- Palyanov A., Khayrulin S., Larson S.D., Dibert A. Towards a virtual *C. elegans*: a framework for simulation and visualization of the neuromuscular system in a 3D physical environment // In Silico Biol. 2012. V. 11. P. 137–147.
- Rankin C.H. Invertebrate learning: what can't a worm learn? // Curr. Biol. 2000. V. 14. No. 15. R617–R618.
- Shao S., Lo E. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with free surface // Adv. Water Resour. 2003. P. 787–800.
- Solenthaler B., Pajarola R. Predictive-corrective incompressible SPH // SIGGRAPH'09: ACM SIGGRAPH 2009 papers. N.Y., 2009a. P. 1–6.
- Solenthaler B., Pajarola R. Performance Comparison of Parallel PCISPH and WCSPH // Technical Report, Department of Informatics, University of Zurich, December 17, 2009b.
- Terzopoulos D., Fleischer K. Modeling inelastic deformation: viscoelasticity, plasticity, fracture // Proc. of the 15th Ann. Conf. on Computer Graphics and Interactive Techniques. ACM Press, 1988. P. 269–278.
- (http://www.artcompsci.org/msa/web/vol_1/v1_web/node34.html)

SIBERNETIC: A SOFTWARE COMPLEX BASED ON THE PCI SPH ALGORITHM AND ORIENTED TO PROBLEMS OF SIMULATIONS IN BIOMECHANICS

A.Yu. Palyanov, S.S. Khayrulin

A.P. Ershov Institute of Informatics Systems SB RAS, Novosibirsk, Russia,
e-mail: palyanov@iis.nsk.su

Summary

Highly detailed realistic simulation of a biological neural network should include the organism's body for efficient work: a sensory system to provide a realistic stream of input data and a muscular system for body movement. The construction of a robotic system coupled with software simulation of a nervous system that would operate in the real world is quite challenging. Moreover, only a scientific group possessing such a robot can perform this research, while it is of worldwide interest. We see a promising alternative to this approach in the design of a computational model of a body. The international project called OpenWorm is aimed at building the first comprehensive computational model of *Caenorhabditis elegans*, a tiny 1-mm long roundworm, possessing a nervous system of 302 neurons, highly interesting to neurobiologists. As part of this project, we developed a special-purpose 3D physical engine, called Sibernetik, oriented to realistic computational simulation of invertebrate biomechanics, particularly *C. elegans*. In doing it, we developed and implemented a number of original features, which supplemented the PCI SPH algorithm, simulating incompressible liquid dynamics. The modified algorithm allowed computation of elastic matter, including its subtype able to contract in response to an incoming signal for simulation of muscle tissue, and liquid-impermeable 'membranes' preventing liquid particles from passing through their surface. The algorithm can be used for modeling a variety of objects from a single living cell membrane to a full body surface of invertebrates, such as mollusks or worms. A set of test scenarios illustrating Sibernetik capabilities was developed to check the correct work of the implemented mechanisms. Sibernetik is the first open source, cross-platform (Windows/Linux/macOS) parallel OpenCL/C++ PCISPH based high-performance software of the kind.

Key words: computer simulation, software for scientific research, PCI SPH, particle method, computational fluid dynamics, invertebrate biomechanics, *C. elegans*, high-performance parallel computing.