

Об ускорении расчета турбулентных течений путем одновременного моделирования его нескольких реализаций

Б.И. Краснопольский

НИИ механики МГУ

Высокоточное моделирование турбулентных течений в рамках прямого численного моделирования или метода крупных вихрей предполагает проведение длительных нестационарных расчетов для достоверного определения осредненных и пульсационных характеристик течения. Для несжимаемых течений основную вычислительную сложность представляет решение систем линейных алгебраических уравнений, в том числе полученных после аппроксимации эллиптического уравнения для давления. Суммарно, решение таких систем уравнений может занимать более 90% общего времени расчета. Известно, что при одновременном решении системы уравнений с несколькими правыми частями, удельное время решения на каждый из векторов правой части оказывается существенно меньше за счет повышения вычислительной интенсивности операций типа произведения разреженной матрицы на вектор. В настоящей работе обсуждается попытка использования данного факта для общего ускорения расчета нестационарных течений за счет одновременного расчета нескольких реализаций одного и того же турбулентного течения.

Ключевые слова: турбулентные течения, прямое численное моделирование, системы линейных алгебраических уравнений с несколькими правыми частями, ускорение

1. Введение

Высокоточное моделирование инженерных и научных задач вычислительной гидродинамики и расчета турбулентных течений является типичной задачей для многопроцессорных вычислительных систем. Применение наиболее достоверных подходов, как то прямое численное моделирование или использование модели крупных вихрей, предполагает получение характеристик турбулентного течения за счет осреднения нестационарных полей в течение некоторого промежутка времени, что требует проведения длительных расчетов. Для интегрирования по времени в таких задачах обычно применяются явные или полу-явные схемы 3-го или более высоких порядков, например [1, 2], для которых подавляющее время расчета шага приходится на решение уравнений для давления.

Для решения систем линейных алгебраических уравнений (СЛАУ) общего вида, полученных при разностной аппроксимации эллиптического уравнения для давления, широко применяются итерационные методы подпространства Крылова и/или многосеточные методы [3, 4]. С алгоритмической точки зрения, данные методы включают в себя базовые операции с плотно-заполненными векторами, а также операции умножения разреженной матрицы на плотно-заполненный вектор (SpMV), которые на практике оказываются доминирующими. Если достаточно эффективная реализация операций с плотно-заполненными векторами на сегодняшний день не представляет больших сложностей, то эффективность реализации операций с разреженными матрицами зависит от большого количества различных факторов. К таковым следует отнести выбор формата представления разреженной матрицы, ее размер и топологию расположения ненулевых элементов.

Несмотря на многообразие предложенных в литературе форматов представления разреженных матриц и соответствующих работ по оптимизации операций типа SpMV,

$$y = Ax,$$

одним из наиболее востребованных в полноценных вычислительных приложениях является простой и универсальный формат CSR (Compressed Sparse Row). Реализация умножения разреженной матрицы на вектор в этом случае оказывается операцией, ограниченной пропускной способностью канала доступа к памяти (“memory-bound”), то есть время доступа к данным доминирует над вычислениями. Вычислительную интенсивность операций типа SpMV можно повысить, если одновременно выполнять умножение разреженной матрицы на несколько векторов (SpMMV),

$$\mathbf{Y} = \mathbf{AX},$$

где матрицы \mathbf{X} и \mathbf{Y} содержат nv столбцов (например, [5]). Однако, итерационные методы, основанные на построении подпространства Крылова, предполагают последовательное построение базисных векторов и операции типа SpMV только с одним вектором. Обобщение методов для одновременной работы с несколькими правыми частями является нетривиальной задачей, и, фактически, требует разработки новых вычислительных алгоритмов.

Целью настоящей работы является апробация подхода к модификации алгоритма расчета турбулентных течений вязкой несжимаемой жидкости для более эффективного использования современных вычислительных систем. Во второй главе обсуждаются предпосылки и теоретические оценки потенциального ускорения расчета за счет проведения нескольких одновременных расчетов. В третьей главе приводятся экспериментальные результаты, подтверждающие результаты приведенных оценок и потенциальную работоспособность предлагаемого подхода. Обсуждаются перспективы и ближайшие планы в рамках данного исследования.

2. Предпосылки и теоретические оценки

2.1. Умножение разреженной матрицы на блок из нескольких векторов

Для формата CSR можно получить простые оценки объема данных, к которым необходимо обратиться на чтение или запись для умножения разреженной матрицы на один вектор. Не вдаваясь в различные стратегии оптимизации операций типа SpMV для формата CSR, активно обсуждаемые в литературе (например, [6, 7]), в простейшем случае код на языке Си для умножения матрицы на вектор представлен на рис. 1. Обозначая n и

```
void SpMV(int *row, int *col, double *val, double *x, double *y, int n)
{
    int i, indx, j1, j2;
    double sum;

    j1 = row[0]; // (1)
    for(i = 0; i < n; i++)
    {
        j2 = row[i+1]; // (1)
        sum = 0.;
        for(j = j1; j < j2; j++)
        {
            indx = col[j]; // (2)
            sum += val[j] * x[indx]; // (3) & (4)
        }
        y[i] = sum; // (5)
        j1 = j2;
    }
}
```

Рис. 1: Реализация умножения разреженной матрицы, представленной в формате CSR, на плотно-заполненный вектор.

nnz – количество строк и ненулевых элементов матрицы соответственно, для умножения

матрицы на вектор имеем: (1) $(n+1)$ обращений к массиву *row* типа *int*¹; (2) nnz обращений к массиву *col* типа *int*; (3) nnz обращений к массиву *val* типа *double*; (4) nnz обращений по индекс-вектору к массиву x типа *double*; (5) n обращений к массиву y типа *double*. Суммарно, для выполнения операции умножения матрицы на вектор необходимо обратиться на чтение или запись к $\sum_1 = 12n + 20nnz$ байт данных.

При реализации операции SpMMV матрицы \mathbf{X} и \mathbf{Y} целесообразно хранить построчно в виде длинных векторов. В этом случае, для приведенных выше оценок, объем данных в п. (4) и (5) увеличивается пропорционально количеству столбцов в матрицах \mathbf{X} и \mathbf{Y} и составляет $\sum_{nv} = 4n(2nv + 1) + 12nnz + 8nv \cdot nnz$ байт, где nv – количество столбцов в матрицах \mathbf{X} и \mathbf{Y} . Причем, если в первом случае обращение к $8nnz$ байт вектора \mathbf{x} выполняется по индекс-вектору с нерегулярным доступом к памяти по 8 байт, то для nv векторов данная операция выглядит как nnz обращений к памяти по индекс-вектору с последовательным чтением по $8nv$ байт. Данное изменение шаблона обращения к памяти может способствовать улучшению эффективности использования кэш-памяти за счет “регуляризации” доступа к данным, а также существенно более эффективно использованию длинных регистров.

Игнорируя потенциальный дополнительный выигрыш за счет “регуляризации” доступа к памяти, можно оценить ожидаемый выигрыш в удельном объеме зачитываемых данных для nv векторов в сравнении с nv -кратным выполнением умножения для одного вектора:

$$P_{nv} = \frac{nv \cdot \sum_1}{\sum_{nv}} = \frac{nv(12n + 20nnz)}{4n(2nv + 1) + 12nnz + 8nv \cdot nnz}.$$

Обозначив $C = nnz/n$, характеризующее среднее число ненулевых элементов в строке матрицы, приведенное выше выражение можно свести к виду:

$$P_{nv} = \frac{nv(3 + 5C)}{2nv(C + 1) + 3C + 1}. \tag{1}$$

Соответствующие оценки для нескольких типичных значений параметров C и nv приведены в табл. 1. Например, для $C = 7$ и $nv = 4$ выигрыш в объеме считываемых данных при умножении на блок векторов составляет 1.77 раза, что является ориентиром ожидаемого ускорения расчета соответствующей операции.

Таблица 1: Оценки выигрыша удельного объема зачитываемых данных P_{nv} для операции типа SpMMV при выполнении операции с nv векторами относительно соответствующей операции типа SpMV с одним вектором

nv	$C = 7$	$C = 11$	$C = 15$
2	1.41	1.41	1.42
4	1.77	1.78	1.79
8	2.03	2.05	2.07
16	2.19	2.22	2.24

¹В дальнейшем одним дополнительным обращением пренебрегается и используется только асимптотика по количеству строк или ненулевых элементов матрицы.

2.2. Оценка ускорения расчета турбулентных течений за счет одновременного моделирования нескольких реализаций

Типичный высокоточный расчет турбулентного течения в рамках прямого численного моделирования или модели крупных вихрей состоит из двух этапов расчета: (1) выработки по некоторому начальному полю скоростей турбулентного течения, и (2) набора статистики и осреднения полей этого турбулентного течения. Общее время интегрирования задачи зависит, вообще говоря, от конкретной постановки задачи, и в том числе, от наличия однородных пространственных направлений, вдоль которых возможно дополнительное осреднение результатов. Конкретная реализация турбулентного течения зависит от выбранных начальных данных, и различные начальные поля скоростей будут приводить к различным реализациям этого течения. Со статистической точки зрения, однако, данные поля будут эквивалентными и могут быть осреднены между собой. Таким образом, вместо проведения одного длительного расчета турбулентного течения возможно провести несколько более коротких расчетов для различных начальных данных, а результаты этих расчетов осреднить между собой. Недостатком данного подхода является определенный объем “накладных расходов” в виде дополнительного времени, затрачиваемого на расчет перестроения течения и выработки турбулентного режима для дополнительных реализаций.

Следующие оценки позволяют определить потенциальный выигрыш от ускорения расчета турбулентного течения с осреднением по нескольким реализациям. Обозначим p – количество одновременно рассчитываемых реализаций турбулентного течения, T_I – характерное время перестроения начального поля скоростей до выработки турбулентного режима течения и T_T – характерное время осреднения для получения искомых характеристик турбулентного течения (рис. 2). Тогда в случае проведения одного расчета общее физиче-

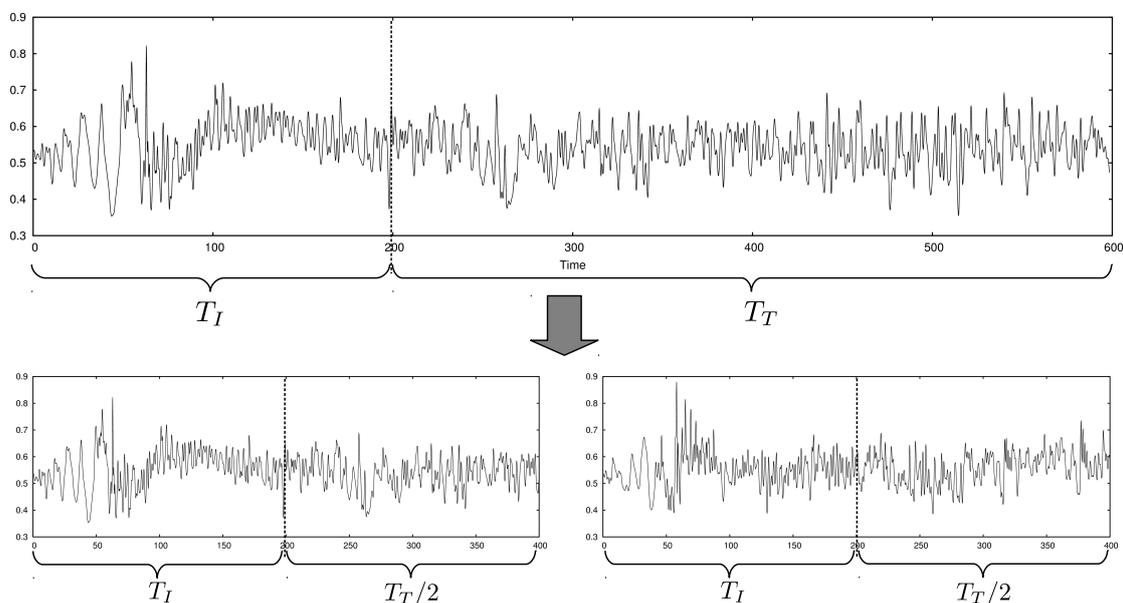


Рис. 2: Типичное поведение мгновенных значений компонент скорости как функции времени для турбулентного режима течения. Расчет турбулентного течения в круглой трубе, продольная компонента скорости. Сверху: перестроения потока T_I и осреднение результатов за время T_T ; снизу: расчет перестроения потоков T_I и осреднение результатов за время $T_T/2$ для двух реализаций течения.

ское время интегрирования задачи составит $T_1 = T_I + T_T$, а при осреднении результатов p расчетов $T_p = T_I + T_T/p$. Формально, эти p расчетов могут выполняться независимо на различных вычислительных ресурсах, однако в рамках данной статьи представляет интерес

вопрос ускорения расчета при использовании заданного количества ресурсов и повышение эффективности их использования для расчета турбулентных течений. Обозначим $\mathcal{T}_p(t)$ – процессорное (астрономическое) время расчета p реализаций турбулентного течения до момента времени t . Тогда потенциальное ускорение за счет осреднения по нескольким одновременно выполняемым расчетам будет определяться как

$$K_p = \frac{\mathcal{T}_1(T_1)}{\mathcal{T}_p(T_p)}.$$

Предполагая, для простоты, что процессорное время расчета $\mathcal{T}_i(t)$ прямо пропорционально времени расчета t , а γ_p – есть функция определяющая замедление для одновременного расчета p состояний турбулентного течения по сравнению с единичным набором, т.е. $\mathcal{T}_p(t) = \gamma_p \mathcal{T}_1(t)$, имеем:

$$K_p = \frac{T_I + T_T}{\gamma_p(T_I + T_T/p)}.$$

Введя параметр $\beta = T_T/T_I$, характеризующий отношение интервала времени интегрирования для набора статистики ко времени перестроения потока с начального поля скоростей, окончательно можно получить:

$$K_p^\beta = \frac{1 + \beta}{\gamma_p(1 + \beta/p)}. \quad (2)$$

Данная величина будет тем выше, чем меньше будет доля времени на формирование турбулентного потока в сравнении с размером интервала времени интегрирования для набора статистики. Если из постановки задачи следует, что $\beta \ll 1$, то, очевидно, обсуждаемый подход оказывается неприменимым.

Как упоминалось ранее, существенную часть времени расчета течений несжимаемой жидкости обычно занимает решение СЛАУ для давления, где доминирующей является операция типа SpMV. Используя полученные ранее выражения для обобщенной операции типа SpMMV как оценку для коэффициента замедления расчета нескольких реализаций $\gamma_p = p/P_p$, возможно получить окончательные оценки для потенциального ускорения при расчете нескольких реализаций турбулентного течения. Например, для расчета 4 реализаций течения, $p = 4$, и параметра β , равного 5 и 10, имеем оценки для ускорения расчета $K_4^5 = 1.18$ и $K_4^{10} = 1.39$. Таким образом, простейшие оценки показывают, что возможно достижение общего ускорения расчета турбулентного течения за счет одновременного проведения расчета нескольких состояний вместо одного длительного расчета, а также позволяют оценить рамки применимости предлагаемого подхода и получить размер ожидаемого ускорения для конкретной задачи.

3. Экспериментальная проверка результатов

3.1. Умножение разреженной матрицы на несколько правых частей

Для проверки изложенных выше оценок первоначально был реализован прототип алгоритма умножения разреженной матрицы, представленной в формате CSR, на плотно-заполненную матрицу, состоящую из нескольких столбцов. В качестве тестовой здесь была использована 7-диагональная матрица размером 1.024 млн. строк (дискретный аналог уравнения Пуассона с 7-точечным шаблоном на прямоугольной сетке 32x32x1000 ячеек). Количество столбцов nv матриц \mathbf{X} и \mathbf{Y} варьировалось от 1 до 16. Тестовые измерения проводились в последовательном режиме на одном узле СК “Ломоносов-2”. Времена выполнения соответствующих операций SpMMV приведены в табл. 2. В таблице также указаны параметры, характеризующие ускорение расчета данной операции по сравнению с nv -кратным выполнением единичного умножения (по аналогии с данными из табл. 1). Как видно из сравнения, расчет операции SpMMV оказывается существенно быстрее, чем последовательное исполнение нескольких операций SpMV. Полученные расчетные данные для

выбранной тестовой матрицы несколько превосходят оценочные значения, что в том числе объясняется дополнительным выигрышем за счет “регуляризации” шаблона доступа к памяти и увеличения эффективности работы кэш-памяти процессора.

Таблица 2: Времена выполнения операции SpMMV для nv правых частей и соответствующий выигрыш по сравнению с nv -кратным исполнением операции SpMV.

nv	Time, ms.	Performance gain
1	7.97	-
2	10.65	1.50
4	13.63	2.34
8	25.58	2.49
16	43.09	2.96

3.2. Решение СЛАУ для систем с несколькими правыми частями

Несмотря на то, что возможность решения СЛАУ с несколькими правыми частями реализована во многих библиотеках прямых численных методов, насколько можно судить по публикациям, в полной мере такой функционал не поддерживается ни в одной из широко-распространенных библиотек итерационных методов. С целью оценки перспектив изложенного подхода одновременного моделирования нескольких турбулентных полей, в библиотеке SparseLinSol [8] был создан прототип соответствующего функционала, реализующий возможность решения СЛАУ с несколькими правыми частями. Эффективность реализации проверена экспериментально для тестовой матрицы размером 1 млн. ячеек (разностный аналог уравнения Пуассона на равномерной сетке 100^3) и различного количества правых частей. Для решения СЛАУ использован стабилизированный метод би-сопряженных градиентов с алгебраическим многосеточным предобуславливателем и симметричным методом Гаусса-Зейделя для сглаживания решения после интерполяции между уровнями. Полученные результаты для 20 итераций методов приведены в табл. 3. Хотя реализация данных

Таблица 3: Времена решения СЛАУ с nv правыми частями и соответствующий выигрыш по сравнению с nv -кратным решением для единичной правой части.

nv	Time, s.	Performance gain
1	7.2	-
2	10.24	1.41
4	15.05	1.91
8	28.45	2.02

алгоритмов предполагает как заметное количество операций с векторами, так и операции

типа SpMV с матрицами всех уровней иерархии многосеточного метода, для которых в значительной степени варьируется размер, шаблон расположения и число ненулевых элементов матрицы, полученные ранее базовые оценки для операций типа SpMMV оказываются сопоставимыми. Так, решение СЛАУ с $p = 4$ правыми частями оказывается в 1.91 раза быстрее, чем решение 4 СЛАУ с единичной правой частью.

3.3. Одновременный расчет нескольких реализаций течения

В рамках проводимых работ по исследованию турбулентных течений вязкой несжимаемой жидкости ранее было разработано приложение для прямого численного моделирования турбулентных течений в трубах с кривизной и кручением [9], использующее библиотеку SparseLinSol для решения СЛАУ для давления. В ходе настоящего исследования созданное ранее приложение было обобщено на случай произвольного количества одновременно рассчитываемых реализаций течения. Важной особенностью используемого метода расчета [10] в этом случае является тот факт, что матрица СЛАУ для давления определяется только геометрией расчетной области и расчетной сеткой, но не зависит от параметров течения, что позволяет решать одну и ту же СЛАУ с несколькими правыми частями.

Для сетки размером $128 \times 64 \times 64$ ячеек, заданной в цилиндрической системе координат, проведены тестовые расчеты течения в прямой трубе. Расчет шага по времени при моделировании одной реализации течения занял 32.8 секунды, тогда как при расчете 4 реализаций течения на этой же сетке это время составило 69.3 секунды. Общий выигрыш в данном случае составляет 1.9 раза, или $\gamma_4 = 4/1.9 = 2.11$, что полностью согласуется с приведенными выше результатами. Это соответствует коэффициентам ускорения полного расчета $K_4^5 = 1.26$ и $K_4^{10} = 1.49$, что даже несколько превосходит изначальные теоретические оценки.

Следует отметить, что указанные выше теоретические оценки приведены для последовательного расчета и не учитывают коммуникационных накладных расходов в случае параллельных расчетов. При выполнении операций с несколькими правыми частями (SpMMV или решение СЛАУ) следует ожидать общего улучшения масштабируемости расчетов. В зависимости от количества векторов nv не меняется количество сообщений, передаваемых по коммуникационной сети, но лишь пропорционально увеличивается их размер. Как показывает практика, для реализаций итерационных методов решения СЛАУ общий размер пересылаемых сообщений является незначительным, а наиболее существенным параметром является латентность передачи сообщений. Увеличение же количества векторов, с которыми выполняются вычисления, приводит к увеличению доли локальных вычислений, и тем самым, повышению эффективности маскировки времени коммуникаций и дополнительного повышения эффективности предлагаемого в работе подхода. Данное направление, а именно параллельная реализация методов решения СЛАУ с несколькими правыми частями и всего приложения для расчета течений в целом является ближайшим планом работ, по результатам которого будет проведено детальное исследование эффективности и целесообразности представленного подхода.

4. Заключение

В настоящей работе обсуждается возможность ускорения расчета турбулентных течений за счет одновременного расчета нескольких состояний с последующим осреднением результатов. Приведены теоретические оценки потенциала ускорения таких расчетов за счет повышения эффективности базовой операции типа произведения разреженной матрицы на вектор, активно используемой в итерационных методах решения СЛАУ. Указанные теоретические оценки поэтапно экспериментально подтверждены для операций умножения разреженной матрицы на блок векторов, решения СЛАУ с несколькими правыми частями и для одновременного расчета нескольких течений. Полученные результаты подтверждают

возможность общего ускорения высокоточных расчетов турбулентных течений за счет одновременного моделирования нескольких реализаций течения с последующим осреднением результатов.

Следующим этапом данной работы является реализация параллельной версии прототипа решения СЛАУ с несколькими правыми частями на основе библиотеки SparseLinSol, в том числе с возможностью использования графических ускорителей, и апробация предложенной методики для проведения расчетов турбулентных течений.

Автор благодарит А.В. Медведева за ценные замечания в ходе обсуждения результатов и подготовки текста статьи.

Литература

1. Nikitin N. Third-order-accurate semi-implicit Runge–Kutta scheme for incompressible Navier–Stokes equations // International Journal for Numerical Methods in Fluids. 2006. Vol. 51. P. 221–233.
 2. Trias F.X., Lehmkuhl O. A Self-Adaptive Strategy for the Time Integration of Navier-Stokes Equations // Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology. 2011. Vol. 60. P. 116–134.
 3. Saad Y. Iterative Methods for Sparse Linear Systems, Second Edition. SIAM, 2003. 528 p.
 4. Trottenberg U., Oosterlee C.W., Schuller A. Multigrid. Academic Press, 2001. 631 p.
 5. Gropp W.D., Kaushik D.K., Keyes D.E., Smith B.F. Towards Realistic Performance Bounds for Implicit CFD Codes. Proceedings of parallel CFD'99. 1999. P. 233–240.
 6. Martone M. Efficient multithreaded untransposed, transposed or symmetric sparse matrix-vector multiplication with Recursive Sparse Blocks format // Parallel Computing. 2014. Vol. 40. P. 251–270.
 7. Buluc A., Fineman J.T., Frigo M., Gilbert J.R., Leiserson C.E. Parallel sparse matrix-vector and matrix-transpose-vector multiplication using Compressed Sparse Blocks. Proceedings of the 21-st ACM Symposium in Algorithms and Architectures (SPAA). P. 233–244.
 8. Krasnopolsky B., Medvedev A. Acceleration of Large Scale OpenFOAM Simulations on Distributed Systems with Multicore CPUs and GPUs // Parallel Computing: On the Road to Exascale. Series: Advances in Parallel Computing. 2016. Vol. 27. P. 93–102.
 9. Краснопольский Б.И., Парамонов В.П. Численное моделирование течений в трубах с кривизной и кручением. XI Всероссийский съезд по фундаментальным проблемам теоретической и прикладной механики. СБОРНИК ДОКЛАДОВ. Казань 20–24 августа 2015 г. 2015. С. 2062–2064.
 10. Nikitin N. Finite-difference method for incompressible Navier-Stokes equations in arbitrary orthogonal curvilinear coordinates // Journal of Computational Physics. 2006. Vol. 217. P. 759–781.
-

Simultaneous modelling of multiple turbulent flow states

B. Krasnopolsky

Institute of Mechanics, Lomonosov Moscow State University

The DNS or LES modelling of turbulent flows assumes long unsteady simulations in order to obtain correct mean velocity profiles and fluctuations. The main computational efforts to model incompressible turbulent flows are spent on solving systems of linear algebraic equations (SLAE), including the ones obtained from the discretization of elliptic equations for the pressure. Solution of these SLAEs usually takes up to 90% of the total computational time. It is known that simultaneous solution of the systems with multiple right hand sides is significantly more efficient in terms of the computational time related to each vector due to increasing the computational intensity and performance of SpMV-like operations. The present work discusses an attempt to speedup the whole turbulent simulation by the simultaneous calculations of several states of the same turbulent flow.

Keywords: turbulent flows, direct numerical simulation, systems of linear algebraic equations with multiple right hand sides, speedup

References

1. Nikitin N. Third-order-accurate semi-implicit Runge–Kutta scheme for incompressible Navier–Stokes equations // International Journal for Numerical Methods in Fluids. 2006. Vol. 51. P. 221–233.
2. Trias F.X., Lehmkuhl O. A Self-Adaptive Strategy for the Time Integration of Navier-Stokes Equations // Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology. 2011. Vol. 60. P. 116–134.
3. Saad Y. Iterative Methods for Sparse Linear Systems, Second Edition. SIAM, 2003. 528 p.
4. Trottenberg U., Oosterlee C.W., Schuller A. Multigrid. Academic Press, 2001. 631 p.
5. Gropp W.D., Kaushik D.K., Keyes D.E., Smith B.F. Towards Realistic Performance Bounds for Implicit CFD Codes. Proceedings of parallel CFD'99. 1999. P. 233–240.
6. Martone M. Efficient multithreaded untransposed, transposed or symmetric sparse matrix-vector multiplication with Recursive Sparse Blocks format // Parallel Computing. 2014. Vol. 40. P. 251–270.
7. Buluc A., Fineman J.T., Frigo M., Gilbert J.R., Leiserson C.E. Parallel sparse matrix-vector and matrix-transpose-vector multiplication using Compressed Sparse Blocks. Proceedings of the 21-st ACM Symposium in Algorithms and Architectures (SPAA). P. 233–244.
8. Krasnopolsky B., Medvedev A. Acceleration of Large Scale OpenFOAM Simulations on Distributed Systems with Multicore CPUs and GPUs // Parallel Computing: On the Road to Exascale. Series: Advances in Parallel Computing. 2016. Vol. 27. P. 93–102.
9. Krasnopol'skiy B.I., Paramonov V.P. Chislennoe modelirovanie techeniy v trubakh s kriviznoy i krucheniem [Numerical modelling of the flows in the pipes with curvature and torsion]. XI Vserossiyskiy s'ezd po fundamental'nym problemam teoreticheskoy i prikladnoy mekhaniki. SBORNIK DOKLADOV. Kazan' 20–24 avgusta 2015 [Proceedings of XI

All-Russian Congress on the Fundamental Theoretical and Applied Mechanics Problems. Kazan, Russia, August, 20–24, 2015]. 2015. P. 2062–2064.

10. Nikitin N. Finite-difference method for incompressible Navier-Stokes equations in arbitrary orthogonal curvilinear coordinates // *Journal of Computational Physics*. 2006. Vol. 217. P. 759–781.