УДК 519.6

Е.А. НИКОЛАЕВСКАЯ, Т.В. ЧИСТЯКОВА

ПРОГРАММНО-АЛГОРИТМИЧЕСКИЕ МЕТОДЫ ПОВЫШЕНИЯ ТОЧНОСТИ КОМПЬЮТЕРНЫХ РЕШЕНИЙ

Ключевые слова: *приближенно заданные исходные данные, ошибки округления, достоверность решения, машинная точность.*

Введение. Использование параллельных компьютеров дает возможность намного увеличить размерности дискретных моделей. Однако получаемые компьютерные решения не всегда имеют физический смысл, во-первых, из-за приближенного характера исходных данных и ошибок, возникающих вследствие представления данных в памяти компьютера, т.е. при переводе их из десятичной в двоичную систему счисления; во-вторых, из-за ошибок округления в процессе вычислений, обусловленных конечной величиной разрядной сетки компьютера; в третьих, из-за ошибок, возникающих в результате замены бесконечного итерационного процесса конечным. Поэтому после постановки математической задачи и ввода ее в компьютер необходимо исследовать корректность постановки компьютерной модели задачи, ее обусловленность и достоверность получаемых результатов [1–4].

Акцентируем внимание на решении проблемы получения достоверного решения программно-алгоритмическими методами в условиях переменной разрядности. Экспериментально исследуем возникающие проблемы и пути их решения.

Достоверность получаемых результатов в значительной мере зависит от чувствительности задач к малым ошибкам округлений, в том числе при переводе чисел из десятичной системы в систему счисления компьютера. Например, для системы линейных алгебраических уравнений (СЛАУ)

$$Ax = b\,, \tag{1}$$

где A и b имеют вид

$$A = \begin{matrix} 0.1348531574394464 & 0.1878970588235294 & 0.1909117647058824 & 0.1779264705882353 \\ 0.1878970588235294 & 0.262 & 0.265 & 0.247 \\ 0.1909117647058824 & 0.265 & 0.281 & 0.266 \\ 0.1779264705882353 & 0.247 & 0.266 & 0.255, \end{matrix}$$

$$b = 0.3516,\ 0.4887,\ 0.5105,\ 0.4818\,,$$

а ее решение

$$x = \begin{bmatrix} 0.66621621621616067387980644904305 72\ldots e13 \\ -0.40168918918907235069521662982454 77\ldots e13 \\ -0.16655405405399700518940186397842 41\ldots e13 \\ 0.97972972973027970725749406963011 15\ldots e12 \end{bmatrix}.$$

После ввода в компьютер получаем систему

$$A_1 x_1 = b_1 \tag{2}$$

с матрицей A_1 и правой частью b_1 вида

$$A_1 = \begin{matrix} 0.134853157439446\mathbf{397} & 0.187897058823529\mathbf{389} & 0.190911764705882\mathbf{391} & 0.177926470588235\mathbf{297} \\ 0.187897058823529\mathbf{389} & 0.2620000000000000\mathbf{10} & 0.265\ 0000000000000\mathbf{13} & 0.24\mathbf{699999999999999} \\ 0.190911764705882\mathbf{391} & 0.2650000000000000\mathbf{13} & 0.2810000000000000\mathbf{27} & 0.2660000000000000\mathbf{14} \\ 0.177926470588235\mathbf{297} & 0.2469999999999999\mathbf{97} & 0.2660000000000000\mathbf{14} & 0.2550000000000000000\mathbf{4} \end{matrix}$$

© Е.А. Николаевская, Т.В. Чистякова, 2009

$$b_1 = 0.3516,\ 0.4887,\ 0.5105,\ 0.4818,$$

точное решение которой $x_1 = 3.547\ldots e13,\ -2.138\ldots e13,\ -8.867\ldots e13,\ 5.216\ldots e12$.

Продолжая компьютерное исследование системы на удвоенной разрядности алгоритмами методов Банча и Гаусса из библиотеки Linpack [5], получаем решения

$$x_{\text{Banch}} = 2.810\ldots e12, -1.694\ldots e12,\ \ldots -7.027\ldots e11,\ 4.133\ldots e11,$$

$$x_{\text{Gauss}} = 3.164\ldots e12, -1.908\ldots e12, \ldots -7.911\ldots e11,\ 4.653\ldots e11,$$

весьма далекие от решения как компьютерной модели задачи, так и математического результата.

Получение неправильного решения объясняется тем, что оценка числа обусловленности матрицы системы $\operatorname{cond}(A) = 2.089436217259268e^{16}$ и, следовательно, удвоенной разрядности недостаточно для получения достоверного решения. С позиций математики проблема заключается в том, что вместо системы уравнений с точно заданными исходными данными вида (3)

$$\widetilde{A}\widetilde{x} = \widetilde{b} \tag{3}$$

необходимо исследовать СЛАУ с приближенно исходными данными вида

$$Ax = b \tag{4}$$

и указать допуски погрешности в исходных данных

$$||\widetilde{A} - A|| = ||\Delta A|| \le \varepsilon_A,\ ||\widetilde{b} - b|| = ||\Delta b|| \le \varepsilon_b, \tag{5}$$

где $||\cdot||$ — одна из используемых норм.

При исследовании математических свойств систем линейных алгебраических уравнений с приближенно заданными исходными данными, связанных с компьютерной реализацией, в качестве приближенной модели в (4) будем понимать именно компьютерную модель задачи. Предположим, что погрешность исходных данных ΔA, Δb в этом случае содержит погрешность, возникающую при записи коэффициентов матрицы в память компьютера или их вычислении.

Специфика компьютерной модели задачи состоит в следующем:

• свойства компьютерной модели, в силу самой ее природы, необходимо исследовать компьютерными алгоритмами;

• свойства машинной модели задачи могут отличаться от свойств математической задачи;

• критерий плохой или хорошей обусловленности машинной модели задачи, если исходные данные математической задачи заданы точно, зависит от математических свойств машинной модели и арифметических свойств компьютера и появляется принципиальная возможность достичь любой заданной точности машинного решения.

Погрешность в решении системы (4), например, с квадратной невырожденной матрицей, вызванная неточным заданием исходных данных (5), оценивается по формуле

$$\frac{||x - \widetilde{x}||}{||x||} \le \sum \frac{h(A)}{1 - \varepsilon_A h(A)} (\varepsilon_A + \varepsilon_B). \tag{6}$$

Из (6) следует, что для получения достоверного решения погрешность задания исходных данных и погрешность вычислений должны согласовываться с числом обусловленности матрицы системы, другими словами, система должна быть хорошо обусловленной. Определяющим при этом является число обусловленности матрицы $h(A)$: $h(A) = ||A|| \times ||A^{-1}||$. Заметим, что сказанное имеет место для машинно-невырожденной задачи $(1.0 + 1/h(A) \ne 1.0)$.

Понятия хорошо и плохо обусловленная матрица тесно связаны с вычислительными возможностями конкретного компьютера и длины мантиссы машинного слова. В результате одна и та же система может квалифицироваться для одной длины мантиссы машинного слова как «машинно плохо обусловленная», или «почти вырожденной», а для другой — «машинно хорошо обусловленная».

Итак, одно из средств получения достоверного решения для плохо обусловленных задач — повышение точности представления чисел и выполнения операций.

Использование библиотеки GMP в вычислениях с переменной разрядностью чисел. Для повышения точности вычислений используем функции библиотеки GMP [6], большой набор которых позволяет организовать вычислительный процесс с разной разрядностью. Как свободная библиотека GMP для произвольной разрядности работает с целыми и рациональными числами, а также с числами с плавающей точкой.

Чтобы проводить вычисления с желаемой точностью, необходимо в Си/С++-программы вставить обращения на соответствующие функции из библиотеки GMP для преобразования типов данных и арифметических действий, а также подключить файл *gmp.h*, где описаны прототипы этих функций.

Эксперименты по применению библиотеки GMP выполнены на интеллектуальной рабочей станции Инпарком-256 [7]. Исследовано применение библиотеки GMP для получения точного решения в случае СЛАУ с плохо обусловленными матрицами. Зная обусловленность матрицы системы и точность вычислений на компьютере, можно определить необходимую разрядность для получения достоверного решения. В табл. 1 приведены значения *macheps*, характеризующие точность плавающей арифметики на Инпарком-256 с различной разрядностью.

Таблица 1

Язык программирования	Длина мантиссы	*macheps*
C++	double (53)	1.110e-16
C++ с использованием GMP	64 128 256	2.71050543121376108502e-20 1.46936793852785938496092…e-39 4.31808427754722231269317…e-78

В табл. 2 представлены результаты решения СЛАУ (1) методом Гаусса, полученные с удвоенной разрядностью С++-программой, а также полученные с повышенной разрядностью с использованием функций библиотеки GMP. Как видим, с увеличением разрядности получаемое компьютерное решение приближается к точному решению.

Таблица 2

Язык программирования	Длина мантиссы	Компьютерное решение системы
C++	double (53)	3.60239e+12 – 2.17203e+12 – 9.00599e+11 5.29764e+11
C++ с использованием GMP	64	**0.666**199107066943565109e13 – **0.4016**7887337851497738e13 – **0.1665**49776766692724716e13 **0.9797**04569216725111902e12
	128	**0.66621621621616067387980б**7836677102584254e13 – **0.401689189189072350695216**8315835297097735e13 – **0.166554054053997005189401**9476345873995266e13 **0.9797297297302797072574945**617251937251362e12
	256	**0.6662162162161606738798064490430572**168030…e13 – **0.4016891891890723506952166298245477**287559…e13 – **0.1665540540539970051894018639784241**392354…e13 **0.979729729730279707257494069630115**72355973…e12

Эксперименты показали, что время решения СЛАУ методом Гаусса с разрядностью 64 C++-программой с использованием функций GMP значительно увеличивается по сравнению со временем решения с удвоенной разрядностью (табл. 3), поскольку потребовалось дополнительное время на вызов и инициализацию функций GMP для каждой арифметической операции. С увеличением размерности задачи эта разница уменьшается. Дальнейшее увеличение разрядности в C++-программе с использованием функций GMP ненамного увеличивает время выполнения C++-программы.

Таблица 3

Порядок матрицы	C++		C++, GMP	
	double (53), с	64, с	128, с	256, с
200	0. 04	0. 75	0. 84	1. 04
1000	6. 55	96.44	14. 72	107.7
4500	688. 94	7320. 06	8050. 12	10104. 56

Функции библиотеки GMP позволяют задавать разрядность в начале программы и выполнять вычисления с этой разрядностью, а также изменять разрядность по мере необходимости в процессе вычислений, т.е. различные фрагменты алгоритма выполнять с различной разрядностью. Этим можно воспользоваться для улучшения точности решения СЛАУ одним из прямых методов, организуя итерационное уточнение решения на повышенной разрядности по сравнению с основными вычислениями.

Реализуя итерационное уточнение решения системы (1) с повышенной разрядностью, за 10 итераций получаем достаточно хорошее приближение к точному решению, а именно с разрядностью 128 получаем 24 верных цифры в решении, а с разрядностью 256 получим 40 верных цифр решения (табл. 4).

Таблица 4

Язык программирования	Длина мантиссы	Компьютерное решение системы
C++	double (53)	3.60239e+12 – 2.17203e+12 –9.00599e+11 5.29764e+11
C++ с использованием GMP	64	**0.666**199107066943565109e13 – **0.4016**7887337851497738e13 – **0.1665**49776766692724716e13 **0.9797**04569216725111902e12
C++ с использованием GMP для итерационного уточнения решения	128	**0.666216216216160673879806**783…e13 – **0.401689189189072350695216**831…e13 – **0.1665540540539970051894019**47…e13 **0.9797297297302797072574945**617…e12
	256	**0.6662162162161606738798064490430572168030** …e13 – **0.4016891891890723506952166298245477287559** …e13 – **0.1665540540539970051894018639784241392354** …e13 **0.97972972973027970725749406963011572355973** …e12

Умножение матрицы на вектор и матрицы на матрицу — базовые макрооперации для многих задач вычислительной математики, например итерационных методов решения систем линейных алгебраических уравнений и т.п. Именно на этих макрооперациях проведены исследования по использованию функций GMP в программах с организацией параллельных вычислений с помощью MPI [8], написанных как C++-программы. В табл. 5 приведена сравнительная характеристика времени выполнения с помощью параллельной C++-программы без использования функций GMP и с использованием функций GMP в виде одного MPI-процесса, а также в виде нескольких MPI-процессов.

По результатам видно, что время решения задачи с разрядностью 64 с помощью параллельной C++-программы с использованием функций GMP значительно увеличивается по сравнению со временем решения с удвоенной разрядностью с помощью параллельной C++-программы без использования функций GMP, а с ростом порядка матрицы эта разница уменьшается аналогично рассмотрению в табл. 3.

Таблица 5

Порядок матриц	Количество процессов	C++, MPI, double(53), с	C++, GMP, MPI, 64, с
1000	1	17. 31	174. 9
	4	0. 5	57
	8	0. 22	30
	16	0. 10	15. 5
	32	0. 05	5. 8
2000	1	228. 69	1000. 1
	4	14. 8	468. 6
	8	3. 33	242. 2
	16	1. 86	124. 6
	32	1. 2	64. 9
3000	4	66. 8	1631
	8	29. 2	824
	16	19. 3	410. 2
	32	11. 1	218. 3

Заключение. Для плохо обусловленных систем с точно заданными исходными данными, используя функции библиотеки GMP для повышения разрядности вычислений, можно получить решение с необходимой точностью.

Время решения задач с помощью программ, использующих функции библиотеки GMP, уменьшается с ростом объема задачи.

Функции библиотеки GMP целесообразно использовать как в традиционных последовательных, так и в параллельных программах.

СПИСОК ЛИТЕРАТУРЫ

1. Молчанов И.Н. Машинные методы решения прикладных задач. Алгебра, приближение функций, обыкновенные дифференциальные уравнения. — Киев: Наук. думка, 2007. — 550 с.
2. Химич А.Н. Оценки возмущений для решения задачи наименьших квадратов // Кибернетика и системный анализ. — 1996. — № 3. — С. 142–145.
3. Химич А.Н. Оценки полной погрешности решения систем линейных алгебраических уравнений для матриц произвольного ранга // Компьютерная математика. — 2002. — № 2. — С. 41–49.
4. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений. — М.: Мир, 1980. — 279 с.
5. Dongarra J.J., Moler C.B. Bunch J.R., Steward G.W. LINPACK user's guide. — Philadelphia: SIAM, 1979. — 307 p.
6. http://gmplib.org/
7. Химич А.Н., Молчанов И.Н., Мова В.И. и др. Численное программное обеспечение интеллектуального компьютера Инпарком. — Киев: Наук. думка, 2007. — 221 с.
8. http://www.mpiforum.org/

Поступила 07.07.2009