УДК 004.222:519.65

ТОЧНОСТЬ МАТЕМАТИЧЕСКИХ ВЫЧИСЛЕНИЙ КЛАССИЧЕСКИХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Лобов Д.В., Лепко А.Э., Луговская Л.А.

ГОУ ВПО «Петрозаводский государственный университет», Петрозаводск, e-mail: ldenis@psu.karelia.ru

В настоящее время основным инструментом, накапливающим и обрабатывающим данные, является ЭВМ. Большинство входных и выходных данных результатов экспериментов, независимо от того, получены ли они человеком или ЭВМ, имеют конечную точность. Анализ ошибок в численном результате должен являться непременной составной частью любого серьезного вычисления на ЭВМ. Исходная информация для проведения численного эксперимента очень редко является точной, так как исходные величины являются экспериментальными данными или основаны на приблизительных оценках. Рассмотрение ошибок в вычислениях позволяет дать определенную оценку точности результатов. В случае привычного натурального эксперимента наиболее полным должен быть анализ погрешностей составных частей экспериментальной установки и анализ погрешностей ЭВМ в качестве основного инструмента обработки полученных данных. В данной статье представлен анализ возможной точности математических расчетов на ЭВМ. Дана статистика влияния количества и типа математических операций, типов данных на точность обрабатываемых результатов с помощью ЭВМ в классических языках программирования.

Ключевые слова: точность математических вычислений, анализ погрешностей ЭВМ, типы данных, классические языки программирования

ACCURACY OF MATHEMATICAL CALCULATIONS OF CLASSICAL PROGRAMMING LANGUAGE

Lobov D.V., Lepko A.E., Lugovskaya L.A.

Petrozavodsk State University, Petrozavodsk, e-mail: ldenis@psu.karelia.ru

Currently, the main tool collects and processes data is a computer. Most of the input and output data of the experimental results, regardless of whether they received by person or a computer, have a finite accuracy. Error analysis in numerical result should be an indispensable part of any serious computer calculations. Source information for the numerical experiment is seldom accurate, since the initial values are experimental data or based on rough estimates. Review errors in the calculation allows to give a specific estimate of the result accuracy. Error analysis of the components of the experimental setup and analysis of computer errors as the main instrument of the data processing must be more complete in the case of the usual natural experiment. In this article we present an analysis of the possible accuracy of mathematical calculations on a computer. The statistics of influence the amount and type of mathematical operations, data types on accuracy of the results processed by the computer in classical programming languages.

Keywords: accuracy of mathematical calculations, computer error analysis, data types, classic programming languages

Погрешность результатов при использовании математических вычислений может рассматриваться как следствие погрешности данных, округления или усечения. Аналогично, погрешность вычисления с помощью библиотечных процедур может быть следствием этих ошибок. Проблема анализа погрешности численных результатов – одна из фундаментальных в проблеме надежности вычислений [3].

Для получения предсказуемых и точных результатов важны следующие факторы: неточность внутреннего представления, неравномерность распределения вещественных чисел, вычитание. При операциях сложения и вычитания могут возникнуть такие ошибки как: потеря значащих цифр мантиссы у меньшего из чисел при выравнивании порядков, потеря крайней справа значащей цифры результата при сложении или вычитании мантисс, выход за границу допустимого диапазона значения того или иного вещественного типа при нормализации результата, большие и маленькие числа.

Для получения результата косвенного измерения выполняют математические операции, а потом округляют результат. Очевидно, что нет необходимости выполнять математические операции с результатами прямых измерений, как с точными числами, а нужно ограничивать их, получив определенное количество цифр в результате. Это облегчает работу, но при таких вычислениях возможны дополнительные погрешности. Чтобы погрешности вычислений не вносили искажений в конечный результат, необходимо брать их на порядок меньше погрешностей косвенных измерений. Поэтому все вычисления следует проводить с количеством значащих цифр, превышающим, как минимум, на единицу количество значащих цифр результатов измерений.

Поскольку аппаратно ЭВМ позволяет использовать лишь конечное подмножество чисел, то разумно соблюдать ряд правил [1], присущих арифметическим операциям. Эти правила более важны, чем требования обеспечения максимально возможной близости

полученного результата к точному результату некоторой операции. Большое количество итерационных алгоритмов не требуют слишком высокой точности выполнения операций, но они оказываются неприменимыми при нарушении описанных выше правил.

На сегодняшний момент существует два способа проводить вычисления с большей точностью. Первый способ – точные значения заменяются приближенными, и проводится оценка погрешности исходных данных и округлений. У этого метода имеет ряд существенных недостатков. Второй способ – это интервальный анализ.

Интервальные вычисления – это достаточно разработанная область. Запись численного алгоритма производится на языках программирования, специально спроектированных с учетом требований интервальной арифметики. Такие языки называются SC-языками – от английского словосочетания «Scientific Computations», то есть «Научные Вычисления». За 70–90-е гг. было разработано целое семейство таких языков: PASCAL-SC и -XSC, FORTRAN-SC и -XSC, OBERON-XSC, MODULA-SC [4]. С помощью SC-языков разработаны различные пакеты численных методов, а написанные на этих языках программы используются для решения научно-технических задач.

Кроме того, существует несколько экспертных систем для математических расчетов посредством ЭВМ. Одно из них – приложение для математических и инженерных вычислений «Mathcad» корпорации PTC. При проведении численных расчетов (в отличие от символьных расчетов) Mathcad использует аппаратно реализованные арифметические инструменты самого компьютера, оперируя 64-битными числами с плавающей точкой. Именно отсюда проистекают все особенности и недостатки численных расчетов, которые на первый взгляд могут показаться следствием несовершенства самой программы [2].

Анализ точности вычислений на примерах вычислительной математики

В данном разделе рассмотрено несколько примеров математических вычислений, иллюстрирующих качество результатов расчетов в классических системах программирования («Turbo Pascal», «Free Pascal» и «Fortran G95»). Проанализируем точность проведенных вычислений, сравнивая их с экспертной системой «MathCAD».

Вычисления посредством системы «MathCAD»

Векторы и матрицы рассматриваются в системе «Mathcad» как одномерные и двумерные массивы данных.

Пример 1: A – матрица размера 2x3, B – матрица размера 3x3.

$$A=\begin{bmatrix}1.100000 & 2.700000 & 3.300000\\ 2.400000 & 7.300000 & 2.800000\end{bmatrix}\quad B=\begin{bmatrix}7.200000 & 6.700000 & 5.400000\\ 4.100000 & 9.200000 & 8.600000\\ 5.100000 & 7.800000 & 9.300000\end{bmatrix}\quad A\cdot B=\begin{bmatrix}35.820000 & 57.950000 & 59.850000\\ 61.490000 & 105.080000 & 101.780000\end{bmatrix}$$

Пример 2: C – матрица размера 3x4, D – матрица размера 4x2.

$$C=\begin{bmatrix}7.125000 & 1.019000 & 5.168000 & 6.951000\\ 2.155000 & 3.208000 & 8.197000 & 9.998000\\ 7.127000 & 8.154000 & 6.318000 & 7.511000\end{bmatrix}\quad D=\begin{bmatrix}1.167000 & 2.287000\\ 7.111000 & 8.151000\\ 1.896000 & 6.998000\\ 7.318000 & 8.322000\end{bmatrix}\quad C\cdot D=\begin{bmatrix}76.226930 & 118.612630\\ 114.033849 & 171.642855\\ 133.244729 & 189.482609\end{bmatrix}$$

Пример 3: E – матрица размера 7x5, F – матрица размера 5x6.

$$E=\begin{bmatrix}2.315000 & 2.365000 & -5.215000 & 8.956000 & -8.958000\\ 9.998000 & 5.235000 & 6.898000 & -9.215000 & 12.365000\\ -15.289000 & 9.658000 & 14.895000 & 7.236000 & -15.236000\\ 18.523000 & 16.235000 & 3.256000 & -8.145000 & 7.569000\\ 20.365000 & 23.365000 & -5.369000 & 12.111000 & 10.359000\\ 8.256000 & 9.325000 & 14.236000 & -7.989000 & 23.256000\\ 12.368000 & 8.369000 & -23.236000 & 3.369000 & 12.368000\end{bmatrix}\quad F=\begin{bmatrix}23.365000 & 2.369000 & -8.256000 & 23.369000 & 2.898000 & -23.569000\\ 4.125000 & 11.256000 & -17.085000 & 5.968000 & 32.256000 & 2.398000\\ -5.978000 & 16.587000 & 7.268000 & 6.256000 & -1.365000 & 5.268000\\ 11.068000 & 20.488000 & -13.636000 & 6.325000 & 7.236000 & 8.975000\\ 12.368000 & 9.266000 & 35.268000 & 3.236000 & 1.516000 & 5.236000\end{bmatrix}$$

$$E\cdot F=\begin{bmatrix}83.353334 & 46.089170 & -535.476045 & 63.247127 & 141.338073 & -42.887573\\ 264.900101 & 122.804718 & 439.895761 & 289.767895 & 140.484194 & -204.712153\\ -514.781345 & 326.628564 & -566.537340 & -210.002573 & 276.151171 & 447.140589\\ 483.759434 & 183.889013 & -28.627543 & 523.100162 & 525.448758 & -413.954540\\ 866.469392 & 566.302184 & -406.150741 & 691.887340 & 923.347335 & -289.301358\\ 345.472213 & 412.464660 & 805.118699 & 362.372471 & 282.728640 & -47.161925\\ 652.660769 & -78.288316 & -23.718881 & 254.941341 & 380.638040 & -298.844155\end{bmatrix}$$

Пример 4: Пусть задана матрица T размером 5*5 и матрица Q размером 5*5. Пусть элементы матриц – дробные числа с точностью 6 знаков после запятой. Выбор шести значимых цифр после запятой обоснован достаточной точностью для большинства реально получаемых экспериментальных данных.

$$\mathbf{T}=\begin{bmatrix} 2.035687 & 5.236589 & 1.025455 & 4.587569 & 7.854666 \\ 8.586999 & 5.555555 & 8.256365 & 4.525685 & 2.365563 \\ 2.050505 & 9.987987 & 5.563214 & 4.147852 & 2.456654 \\ 8.989898 & 7.777777 & 3.363636 & 5.265655 & 2.323232 \\ 4.444444 & 5.587458 & 5.698745 & 5.585858 & 8.878789 \end{bmatrix} \quad \mathbf{Q}=\begin{bmatrix} 7.589654 & 8.741236 & 5.236589 & 2.353535 & 5.555555 \\ 5.256366 & 1.203568 & 5.147852 & 8.888777 & 6.236985 \\ 5.252525 & 1.036985 & 1.254254 & 5.236666 & 7.777777 \\ 8.520360 & 1.123321 & 1.111111 & 7.475869 & 6.545454 \\ 8.741369 & 3.210210 & 5.566998 & 6.555666 & 4.150659 \end{bmatrix}$$

Однако результат перемножения матриц представим в виде таблицы чисел с 12 выводимыми разрядами.

$$T \cdot Q = \begin{pmatrix} 156.110089664341 & 55.528832828368 & 87.727630635868 & 142.496529508245 & 114.575420435205 \\ 196.979869815448 & 102.986950668644 & 102.11896112689 & 162.168987202013 & 186.012932573732 \\ 154.099770751908 & 48.259883625418 & 87.416926081989 & 169.85341929655 & 154.302363799542 \\ 191.954144494982 & 104.805027058699 & 110.118254860711 & 162.502883187442 & 168.724408375165 \\ 218.24060901215 & 86.261821456861 & 114.819515986622 & 189.933764681652 & 177.278614671898 \end{pmatrix}$$

Вычисления посредством систем программирования Pascal и Fortran

Для сравнения падения точности расчетов при большом количестве арифметических операций были написаны программы, реализующие перемножение матриц.

При вводе элементов матриц из вышеприведенных примеров 1-3 результирующие матрицы совпадают с матрицами, полученными в системе «MathCad».

Сравнивая результаты расчетов примера 4 с результатом, полученным при перемножении этих матриц в программе MathCAD, можно сделать следующие выводы:

1) При перемножении матриц T и Q с использованием компилятора Turbo Pascal в случае, когда элементы матриц – дробные числа типа single, в результирующей матрице точность элементов – 4 знака после запятой;

2) Результат перемножения матриц T и Q с использованием компилятора Turbo Pascal (элементы матрицы – дробные числа типа real. Вывод осуществлялся с точностью 11 знаков после запятой):

```
156.11008966000   55.52883282800   87.72763063600  142.49652951000  114.57542044000
196.97986982000  102.98695067000  102.11896113000  162.16898720000  186.01293257000
154.09977075000   48.25988362600   87.41692608200  169.85341930000  154.30236380000
191.95414450000  104.80502706000  110.11825486000  162.50288319000  168.72440838000
218.24060901000   86.26182145700  114.81951599000  189.93376468000  177.27861467000
```

Видно, что восьмой знак после точки уже округлен. Для матриц с размерами меньше, чем 2*3 и 3*3, результирующая матрица на выходе имеет 9 знаков после запятой. Однако, для матриц этих же размеров, но с элементами с большим числом знаков после запятой, элементы результирующей матрицы имеют округление в девятом знаке после запятой;

3) При перемножении матриц T и Q с использованием компилятора Turbo Pascal в случае, когда элементы матриц – дробные числа типа double, в результирующей матрице точность получаемых элементов – 8 знаков после запятой.

4) Результат перемножения матриц T и Q с использованием компилятора Turbo Pascal (элементы матрицы – дробные числа типа extended. Вывод осуществляется с точностью 11 знаков после запятой):

```
156.11008966434   55.52883282837   87.72763063587  142.49652950825  114.57542043
196.97986981545  102.98695066864  102.11896112689  162.16898720201  186.01293257
154.09977075191   48.25988362542   87.41692608199  169.85341929655  154.30236379
191.95414449498  104.80502705870  110.11825486071  162.50288318744  168.72440837
218.24060901215   86.26182145686  114.81951598662  189.93376468165  177.27861467
```

Одиннадцатый знак округлен!

5) Результат перемножения матриц T и Q с использованием компилятора Free Pascal (элементы матрицы – дробные числа типа real, вывод осуществляется с точностью 11 знаков после запятой) аналогичен результату перемножения матриц T и Q с использованием компилятора Turbo Pascal при использовании типа данных extended для элементов матриц. При использовании элементов матрицы типа extended , результат не меняется.

Можно сделать вывод о том, что при использовании дробных чисел типа extended или при использовании компилятора Free Pascal, а не Turbo Pascal, точность расчета результирующей матрицы повышается.

6) Результат перемножения матриц T и Q при использовании компилятора Fortran G95 (элементы матрицы – дробные числа типа real*4 – 4 байта):

```
156.110092163086  55.528835296631  87.727630615234 142.496536254883 114.575424194336
196.979858398438 102.986946105957 102.118957519531 162.168991088867 186.012939453125
154.099761962891  48.259883880615  87.416923522949 169.853424072266 154.302352905273
191.954132080078 104.805023193359 110.118263244629 162.502883911133 168.724411010742
218.240615844727  86.261817932129 114.819519042969 189.933761596680 177.278625488281
```

Видно, что 5-6 знак после запятой округлен.

7) Результат перемножения матриц T и Q при использовании компилятора Fortran G95 (элементы матрицы – дробные числа типа real*8 – 8 байт):

```
156.110089664   55.5288328284   87.7276306359   142.496529508   114.575420435
196.979869815   102.986950669   102.118961127   162.168987202   186.012932574
154.099770752   48.2598836254   87.4169260820   169.853419297   154.302363800
191.954144495   104.805027059   110.118254861   162.502883187   168.724408375
218.240609012   86.2618214569   114.819515987   189.933764682   177.278614672
```

Элементом результирующей матрицы является число с точностью восемь знаков после запятой. Девятый знак округлен.

8) Результат перемножения матриц T и Q при использовании компилятора Fortran G95 (элементы матрицы – дробные числа типа real*10 – 10 байт):

Элементом результирующей матрицы является число с точностью четырнадцать знаков после запятой.

Таким образом, можно сделать следующие выводы:

– В экспертной системе MathCad результат выводится с точностью 12 знаков после запятой без ошибок и округлений;

– Компилятор Turbo Pascal делает округление вещественных чисел типа real в восьмом знаке после запятой;

– В результате перемножения двух матриц с элементами типа extended с помощью компилятора Turbo Pascal округление происходит в одиннадцатом знаке после запятой;

– При использовании компилятора Free Pascal округление элементов результирующей матрицы происходит в одиннадцатом знаке после запятой, независимо от типа элементов исходных матриц – real или extended. Точность не зависит от типа данных, т.к. во Free Pascal тип real занимает 8 байт, а не 6, как в Turbo Pascal. (Free Pascal использует математический сопроцессор (или эмуляцию) для всех вычислений с плавающей точкой. Размер стандартного типа Real зависит от процессора и является либо Single, либо Double);

– В результате перемножения двух матриц с элементами типа Real*4 с помощью программы, написанной на языке Fortran, округление происходит в пятом или шестом знаке после запятой;

– При использовании компилятора Fortran G95 в результате перемножения двух матриц с элементами типа Real*8 округление происходит в девятом знаке после запятой;

– При использовании Fortran G95 в результате перемножения двух матриц с элементами типа Real*10 округления не происходит вплоть до пятнадцатого знака после запятой.

Метод наименьших квадратов

МНК – один из базовых методов регрессионного анализа для оценки неизвестных параметров регрессионных моделей по выборочным данным. Метод основан на минимизации суммы квадратов остатков регрессии.

Рассмотрим следующую задачу: аппроксимировать степенным полиномом набор из восьми экспериментальных точек (таблица).

x	1.123456	2.258741	2.423658	2.725369	3.123655	3.128795	4.259965	5.302547
y	24.861238	33.568903	34.112533	36.555926	47.012618	60.024471	129.145676	195.688218

Показать, что результат вычислений будет также определяться типом входных и выходных переменных в рассматриваемых языках программирования.

Так как условие минимального отклонения значений функции от экспериментально полученных точек есть равенство нулю частных производных функции, то, решив систему уравнений, составленную из этих частных производных, можно найти приближенную зависимость в виде полинома.

Проведем аппроксимацию полиномом 5-й степени.

а) Результат вычислений в виде вектора коэффициентов C в программе MathCAD:

$$C = \begin{pmatrix} -603.73655760829 \\ 1272.277408568 \\ -925.652613184033 \\ 309.246283129287 \\ -47.653938149911 \\ 2.778427946251 \end{pmatrix}$$

Величина среднеквадратичного отклонения значений полинома и таблично заданной функции: 9.079724068764.

б) Результат вычислений в Turbo Pascal, тип входных данных – single, тип выходных данных (погрешности) – real:

```
A1 = -603.887968625873
A2 = 1272.589841049165
A3 = -925.886554166675
A4 =  309.327651709784
A5 =  -47.667253405554
A6 =    2.779255152189
Inaccuracy=9.0797291401
```

Отклонение с точностью 5 знаков после запятой.

в) Результат вычислений в Turbo Pascal, тип входных данных – double, тип выходных данных (погрешности) – real:

```
A1 = -604.250166568905
A2 = 1273.337654709816
A3 = -926.446875346825
A4 =  309.522672114894
A5 =  -47.699186705460
A6 =    2.781240094275
Inaccuracy=9.0797235253
```

Отклонение с точностью 5 знаков после запятой.

г) Результат вычислений в Turbo Pascal, тип входных данных – single, тип выходных данных (погрешности) – single:

```
A1 =   14.696577662602
A2 =   -3.972552537918
A3 =   30.088212967850
A4 =  -23.218275838066
A5 =    6.757489985961
A6 =   -0.602229611312
Inaccuracy=9.7508990534
```

Отклонение с точностью 0 знаков. То есть, все знаки после разделителя целой и дробной частей не верны!

д) Результат вычислений в Turbo Pascal, тип входных данных – double, тип выходных данных (погрешности) – double:

```
A1 = -604.121005412191
A2 = 1273.071198556572
A3 = -926.247414892539
A4 =  309.453315097373
A5 =  -47.687839784718
A6 =    2.780535317950
Inaccuracy=9.0797233260
```

Отклонение с точностью 5 знаков после запятой.

е) Результаты вычислений при использовании компилятора Free Pascal полностью совпадают с результатами, полученными при использовании компилятора Turbo Pascal.

Таким образом, можно сделать следующие выводы:

– тип переменных для входных данных не влияет на точность результатов;

– тип данных размером 4 байта непригоден для выходных переменных в случаях расчетов, содержащих множественное возведение в степень и операции с матрицами;

– результат необходимо представлять только переменными, имеющими размер не ниже 6 байт, в этом случае точность результатов будет в пределах 5-6 знаков после запятой;

– при использовании входных данных с количеством знаков после запятой от 0 до 6 точность результата неизменна.

Заключение

На основании вышеизложенного можно сделать следующие выводы:

– Использование типа переменных размером 4 байта для научных расчетов нецелесообразно, так как результирующая точность при использовании переменных одинарной точности составляет не более 4 знаков после запятой. Тип данных размером 4 байта также непригоден для выходных переменных в случаях расчетов, содержащих множественное возведение в степень и операции с матрицами, так как происходит быстрое накопление погрешно-

сти и в некоторых случаях в результате будет верна лишь целая часть.

– Для большинства расчетов минимально и достаточно использовать для представления данных переменные с двойной точностью (8 байт), т.к. они дают точность не менее 5 знаков после запятой, что вполне достаточно для большинства научных исследований;

– Входные и выходные данные, а также все промежуточные данные, должны быть приведены к одному типу. Для получения результатов с двойной точностью все входные и выходные данные, а так же все константы, должны иметь двойную точность;

– Очевидно, что необходимо сводить к минимуму число необходимых арифметических операций. Например, умножение дробных чисел – часто самая неточная операция, сразу дающая ошибку в четвертом знаке результирующего числа. Также следует избегать вычитания. Особенно следует остерегаться вычитания очень близких друг к другу чисел;

– Чтобы погрешности вычислений не вносили искажений в конечный результат, необходимо брать их на порядок меньше погрешностей косвенных измерений. Поэтому все вычисления следует проводить с количеством значащих цифр, превышающим на единицу количество значащих цифр результатов измерений;

– При относительной погрешности результата косвенных измерений порядка 10-100% вычисление можно проводить с двумя значащими цифрами. При относительной погрешности порядка 1-10% – с тремя значащими цифрами. При относительной погрешности порядка 0.1-1% – с четырьмя значащими цифрами.

Список литературы

1. Бум Х., Э. Де Джонг. Критическое сравнение некоторых реализаций языков программирования. – Нидерланды. [Электронный ресурс]. – URL: http://www.az-design.ru/index.shtml?Support&SoftWare&Delphi/Pascal/001b4014 (дата обращения: 28.04.16).

2. Деммель Дж. Вычислительная линейная алгебра. Теория и приложения. – М.: Мир, 2001. – 430 с.

3. Институт технической кибернетики. Национальная академия наук Беларуси. [Электронный ресурс]. Минск, 2000. – URL: http://ice_diff.chat.ru (дата обращения: 23.04.16)

4. Юровицкий В.М. О компьютерной «вычислительной катастрофе». – МФТИ, РГСУ, Москва. – сор. 1999-2012 [Электронный ресурс]. – URL: http://www.yur.ru/science/computer/ Comcat.htm (дата обращения: 23.04.16).