

**ИСПОЛЬЗОВАНИЕ МЕТОДОВ ОБЪЕКТНО -
ОРИЕНТИРОВАННОГО
ПРОГРАММИРОВАНИЯ ПРИ ОПЕРАЦИЯХ
СО СЛАБО - ОБУСЛОВЛЕННЫМИ МАТРИЦАМИ**

к.т.н. С.В. Маловица, С.И. Березина
(представил д.т.н., проф. Д.В. Голкин)

Проведен анализ причин возникновения вычислительных ограничений при математических операциях со слабо - обусловленными корреляционными матрицами. Предлагаются алгоритмы разрешения этих затруднений с применением методов объектно - ориентированного программирования.

При решении различных задач космической динамики таких как моделирование движения космического аппарата (КА), прогноз движения КА, определения вероятности нахождения КА в заданном районе, приходится оперировать с корреляционными матрицами [1]. Свойства корреляционных матриц соответствуют свойствам квадратных диагонально симметричных матриц [2]. Поэтому с ними проводятся такие же математические операции. Однако зачастую возникают события, когда определитель этих матриц близок к нулю (слабо - обусловленные матрицы).

Существует большое количество способов представления чисел в ПЭВМ. Числа, составляющие элементы корреляционных матриц, являются вещественными (непрерывными). Множество вещественных чисел бесконечно и несчетно [3]. Этот факт и создает затруднения при операциях со слабо - обусловленными корреляционными матрицами. Дело в том, что для абсолютно точного представления любого вещественного числа в ПЭВМ необходимо использовать бесконечное число бит, что очевидно невозможно.

В современных ПЭВМ наиболее часто используются 32, 64 и 80 разрядные форматы хранения значений вещественных чисел [4], в языке C++ – **float**, **double** и **extended** [5]. Относительная точность представления этих чисел (отношение неточности представления к значению самой величины) определяется количеством разрядов мантииссы. Например, для формата **double** мантиисса состоит из 52 бит. Следовательно, относительная точность составляет $2^{-52}=10^{-16}$. Такая точность удовлетворяет большинство практических задач космической баллистики. Кроме того, представленные ранее форматы вещественных чисел поддерживаются на аппаратном уровне [4], что существенно ускоряет вычислительный процесс.

Однако, даже такая точность может оказаться неприемлемой при решении практических задач, если в этих задачах проводятся операции со слабо -

обусловленными корреляционными матрицами, например, если в результате вычислительных ограничений становится неверным известное равенство матричной алгебры

$$\mathbf{M} \times \mathbf{M}^{-1} = \mathbf{E}, \quad (1)$$

где \mathbf{M} – слабо - обусловленная матрица; \mathbf{M}^{-1} – обратная матрица; \mathbf{E} – единичная матрица.

Самые разнообразные попытки преодолеть эти затруднения с помощью изменения способа вычисления обратной матрицы не приводят к требуемому результату. Единственным способом, применяющимся на практике, в этом случае был переход к более точному представлению вещественных чисел, например, от **double** к **extended**. Однако, иногда даже 80 разрядов будет не хватать для того, чтобы было верным выражение (1).

Таким образом, актуальной является научно - техническая задача, состоящая в разработке алгоритмов операций со слабо - обусловленными матрицами и реализации их с помощью программных средств. Эти алгоритмы должны быть таковы, чтобы точность представления вещественных чисел не ухудшалась от операции к операции, а реализация их с помощью программных средств не должна использовать объем памяти больше необходимого.

Для разрешения вышеуказанных затруднений целесообразно вещественные числа представить в форме исчисляемых чисел, ибо неисчисляемость вещественных чисел и приводит к этим затруднениям. При обращении квадратных матриц в основном используются такие математические операции как сложение, вычитание умножение и, в некоторых случаях, деление [8]. Поэтому числа, форма представления которых заменит форму представления вещественных чисел, должны быть замкнуты по операциям сложения, вычитания, умножения и деления [6]. Рациональные числа как раз и обладают свойством замкнутости по перечисленным выше математическим операциям.

С точки зрения практической реализации очень удобным в этом случае оказывается использование объектно-ориентированного программирования, а именно, его реализации на языке C++, так как на языке C++ есть методика перегрузки математических и логических операций. Теперь, если идентификатор **double** заменить на имя класса рациональных чисел (например, **Racional**), программисту не нужно будет менять остальной текст программы, так как математические и логические операции аппаратной реализации вещественных чисел будут, с математической точки зрения, соответствовать таким же операциям с рациональными числами. Следовательно, для программиста введение нового класса (объекта) является «прозрачным», а значит, не будет снижена скорость разработки программных средств.

Возможность преобразования вещественного формата хранения чисел к формату рациональных чисел не вызывает сомнений, так как вследствие ограничения количества бит все вещественные числа, хранимые в ПЭВМ, в какой то мере являются рациональными, например, для случая **double** [7]:

$$ВЧ = \frac{1 m_{51} m_{51} \dots m_1 m_0}{100 \dots 000} \cdot 2^{E_{11} E_{10} \dots E_1 E_0 - 1023}, \quad (2)$$

где $m_0 - m_{51}$ – значения битов соответствующих разрядов мантииссы; $E_0 - E_{11}$ – значения битов соответствующих разрядов показателя степени числа.

Для исключения последовательного «нарастания» размерности составляющих рационального числа (числителя и знаменателя) необходимо: во-первых, постоянно находить наибольший общий делитель числителя и знаменателя и делить их на него; во-вторых, динамически распределять ОЗУ, чтобы не хранить неиспользуемые данные. Это предусмотрено в приемах объектно-ориентированного программирования, например, с помощью операторов **new** и **delete[]**, хотя, с системной точки зрения в уже разработанной версии реализации класса **Racional** использовался «естественный» формат натуральных чисел, т.е. хранение числа происходит в форме его текстового десятичного написания в формате **AnsiString**. Несмотря на то, что в этом формате хранение менее эффективно, распределение памяти происходит более надежно, чем с операторами **new** и **delete[]**.

Таким образом, реализовав алгоритмы работы с рациональными числами с помощью объектно-ориентированного программирования, удалось преодолеть затруднения при операциях со слабо - обусловленными корреляционными матрицами. Разработанные средства могут применяться и в других областях применения вычислительных средств, когда точность представления вещественных чисел не определена, но играет существенную роль. Однако, разработанные средства, работая с исчисляемыми форматами представления чисел, не используют специальных аппаратных средств. Поэтому целесообразно использовать стандартные программные средства, а при работе с потенциально слабо - обусловленными матрицами проверять их с помощью (1) и если оно невыполнимо, то использовать предлагаемые средства.

EMAIL авторов – cybermen@gomail.com.ua.

ЛИТЕРАТУРА

1. Жданюк Б. Ф. Основы статистической обработки траекторных изменений. – М.: Сов. радио, 1978. – 384 с.
2. Эльясберг П. Е. Определение движения по результатам измерения. – М.: Наука, 1976, – 416 с.
3. Коршунов Ю.М. Математические основы кибернетики. – М.: Энергия, 1980. – 424 с.
4. . Руководство по архитектуре IBM PC AT / Под ред. М. Л. Мархасина. – Минск: ООО «Консул», 1993. – 949 с.
5. Язык программирования C++ . – М.: И.В.К. - Софт, 1991. – 314 с.
6. Виноградов И.М. Основы теории чисел. – М.: Госиздат, 1949. – 180 с.
7. МК К1810. Структура, программирование, применение / Под ред. Ю. М. Казаринова. – М.: Высшая школа, 1990. – 269 с.

8. Гантмахер Ф.Р. Теория матриц. – М.: Наука, 1967. – 575 с.

Поступила в редколлегию 23.02.2001
