

Разработка алгоритмических методов ускорения вычислений

Е.В.Криворуков, И.Н.Шаповал

ИФВЭЯФ ННЦ ХФТИ, г. Харьков

1. ВВЕДЕНИЕ

Ускорение вычислений при выполнении сложных научно–технических расчётов являлось актуальным на всех этапах развития вычислительной техники. Основными методами решения данной задачи являются совершенствование элементной базы и основных компонентов ЭВМ; внедрение прогрессивных архитектурных решений; разработка оригинальных алгоритмов и программных комплексов решения задач.

В данной работе приводятся некоторые результаты именно по последнему направлению оптимизации вычислений — применению таблично–алгоритмических методов ускорения вычислений (ТАМ), суть которых состоит в использовании эффекта более быстрой аппроксимации функции по опорным значениям по сравнению с прямыми вычислениями. Данной проблематике было уделено достаточно внимания исследователями [1–4], однако результаты были получены в основном для обладающих большой памятью универсальных ЭВМ. В связи со значительным ростом объёмов оперативной памяти у персональных ЭВМ последних поколений (у Pentium MMX и Pentium II этот показатель, как правило, превышает 32Мб) представляет интерес исследовать возможности ПЭВМ в зависимости от наличных ресурсов и с учётом особенностей их технической и системной архитектуры и весьма существенного фактора отдалённости большинства исследователей от мест концентрации высокопроизводительной вычислительной техники, в силу ряда причин (финансово–организационных, ограниченности доступа и т.п.) не преодолеваемой существующими сетевыми структурами.

Рассмотрим вкратце постановку задачи; подробная постановка приведена в работе авторов [4].

2. ПОСТАНОВКА ЗАДАЧИ И МЕТОД РЕШЕНИЯ

Прямое вычисление широкого класса функций, как хорошо известно [1–4], в окрестности точки $x_0 \in [a, b]$ может быть заменено выполнением аппроксимации функции по её опорным значениям, например, полиномами специального вида или усечёнными рядами разложения этой функции. Должны быть

изучены и переработаны все те существенно времяёмкие вычисления, в которых такая замена способа вычисления функции приводит к значительному уменьшению времени их выполнения.

Вычисление функции $f(x)$ реализуется набором операций $\{/, *, \pm, \text{выборка из памяти}\}$, расположенных в порядке убывания времен выполнения $t_1 > t_2 > \dots \geq t_n$ одной соответствующей операции и, следовательно, время вычисления $f(x)$ $t_f = \sum_{i=1}^n n_i t_i$ для выбранной расчётной среды определяется набором $n_i = \{n_1, n_2, \dots, n_n\}$, где целое n_1 равно количеству операций деления, n_2 – умножения и т.д. Поэтому сокращение времени вычисления может быть достигнуто за счёт уменьшения или всех, или первых за счёт последних компонент набора n_i .

При выборе алгоритма ускоренного вычисления функции такая оптимизация по набору n_i в распространённых реализациях ТАМ приводит к столь значительному росту объёмов опорных таблиц (ОТ) функций, а значит, и к увеличению затрат времени на поиск наилучших узловых значений функций в этих таблицах, что это может приводить к эффекту, прямо противоположному ожидаемому ускорению. Стремление получить приемлемую точность аппроксимаций без увеличения объёмов таблиц, в свою очередь, также увеличивает затраты времени, сопоставимые с ожидаемым выигрышем. Так, например, хорошо известно, что использование наилучшего чебышевского приближения обеспечивает наилучшее равномерное приближение среди полиномов фиксированной степени N , но при этом расходуется значительное время на нахождение точек альтернанса и вычисление коэффициентов полинома. Тем самым стоит задача найти реализацию ТАМ, при которой эффект ускорения непосредственно аппроксимирующего вычисления не приводит к конкурирующему росту затрат времени на поддержку такой аппроксимации, нивелирующему ускорение. В качестве решающего фактора здесь выступают адаптивные аспекты использования ОТ.

Введём набор $n^* = \{n_1^*, n_2^*, \dots, n_n^*\}$, представляющий операции аппроксимации функций оптимально подобранным рядом; этот набор и время его реализации $t^* = \sum_{i=1}^n n_i^* t_i$ при однократном вычислении, естественно, не должны зависеть от выбора функции.

Пусть t_m^f – время, необходимое для построения ОТ из d_f опорных точек функции $f(x)$. ОТ должна строиться и использоваться с учётом обеспечения приемлемой точности аппроксимации вычислений.

Для оценки эффективности разрабатываемых реализаций ТАМ определим коэффициенты K_n - чистого и K_n - полного ускорения вычислений. Тогда для сеанса вычислений, в котором функция вычисляется M_f раз, $M_f \gg d_f$, введенные коэффициенты приобретают вид

$$K_n = t_f / t^*; K_n = M_f t_f / [t_m^f + (M_f - d_f)t^*], K_n \rightarrow K_n \quad (1)$$

вместе с ростом M_f при фиксированных t_f, t_m^f, d_f, t^* . Такая ситуация имеет место, очевидно, в сеансах вычислений, в которых количество обращений к функции сопровождается существенной локализацией значений аргумента этой функции.

Специфика универсальных ЭВМ, предназначенных для широкого круга задач, имеет следствием формирование унифицированного программного обеспечения. Поэтому программная реализация ТАМ на универсальной ЭВМ неизбежно увеличивает размеры ОТ по мере охвата классов задач или расширения областей определения функций. Это наблюдение подтверждают и проводимые в ведущих компьютерных фирмах реализации ТАМ для фиксированных функций. В [4] описана реализация, которая позволяет динамически в сеансе вычислений и адаптивно к области задания получить заметное ускорение вычислений. Построенная такая реализация означает, по существу, персонализацию универсальной ЭВМ.

В связи с широким распространением ПЭВМ и ростом их ресурсов памяти представляется естественным перенос и дальнейшее развитие ТАМ на ПЭВМ. Настоящая статья и имеет целью обоснование такого направления повышения скорости вычислений.

3. АДАПТИВНЫЕ РЕАЛИЗАЦИИ ТАМ

Основой ускорения является в соответствии с формулой (1) сведение вычислений для широкого (нефиксированного) класса функций к стандартной аппроксимации функции по её самозапоминаемым опорным значениям, так что эффект ускорения в пределе больших сеансов локализованных вычислений приближается к коэффициенту K_n . В таблице приведены полученные в численном эксперименте на ПЭВМ оценки для K_n .

Если в смеси вычислений по времени доминируют ускоряемые части, общее ускорение в пределе больших сеансов, приближаясь к K_n , может быть заметным. Степень приближения зависит, как показано в [4], от пространственной организации ОТ, порядка возникновения в сеансе вычислений дополняющих ОТ значений, объёма вычислений. Эффективны следующие адаптивные реализации

ТАМ: интервальная ОТ (строится непосредственно после определившегося интервала), дискретная ОТ (опорные узлы пополняют в ОТ по мере вычисления), ОТ с одной опорной точкой (при плавном изменении аргумента функции).

Функция	K_n
$x^{1/6}/(1-x^6)$	7.1
Интегральный синус	4.3
Гамма- функция	2.6
Полный эллиптический интеграл первого рода	6.7
Обобщённый полный эллиптический интеграл	8.2
Функция Бесселя	72.

Представительный банк результатов применения ТАМ для ускорения вычислений и эффективные алгоритмы их реализации формируются в процессе разработок и обобщения опыта их эксплуатации.

4. ВЫВОДЫ

Обоснована необходимость применения и развития в среде ПЭВМ выполненных разработок ТАМ с опытом их применения для универсальных ЭВМ. Показано на основе ожидаемого ускорения, что следует ожидать заметного ускорения времяёмких вычислений при локализации области их определения.

Литература

1. Попов Б.А., Теслер Г.С. Приближение функций для технических приложений.-Киев, Наук. думка, 1980.
2. Васильев А.Б., Прянников В.А., Семёновых В.И. Таблично-алгоритмический метод вычисления функций для универсальных микроЭВМ. Л.: Изд-во Ин-та точной механики и оптики, 1985. Т.3, с.621.
3. Потапов В.И., Флоренсов А.И. Таблично-алгоритмическое вычисление функций в ЭВМ. Иркутск: Изд-во ун-та, 1985, с. 107.
4. Криворуков Е.В., Шаповал И.Н. Программная реализация таблично-алгоритмического ускорения вычислений функций. Программирование, 1989г., №3, с. 78-87.

Статья поступила в редакцию 25 мая 1998 г.,
в издательство 1 июня 1998 г.