4. *Ромм Я.Е., Дзюба А.С.* Метод распознавания рукописных символов на основе сортировки полярных координат. – Таганрог: Изд-во ТГПИ, 2012. – 42 с.
5. *Tou J.T., Gonzalez R.C.* Pattern recognition principles. – Addison-Wesley publishing company. – 1974.
6. *Ян Д.Е.* Исследование, развитие и реализация методов автоматического распознавания рукописных текстов в компьютерных системах: Дисс. … канд. физ.-мат. наук. – М., 2003. – 179 с.
7. *Горошкин А.Н.* Применение векторного подхода к распознаванию рукописных символов: научная статья // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2006. – С. 15-17.
8. *Рюмин О.Г.* Разработка и исследование алгоритмов распознавания изображений на основе определения экстремальных признаков замкнутых контуров с помощью сортировки: Автореф. дисс. … канд. техн. наук. – Таганрог: Изд-во ТТИ ЮФУ, 2008, – 16 с.

Статью рекомендовал к опубликованию д.т.н., профессор Д.П. Фельдман.

Ромм Яков Евсеевич – ФГБОУ ВПО "Таганрогский государственный педагогический институт имени А.П. Чехова"; e-mail: romm@list.ru; 347926, г. Таганрог, ул. Инициативная, 48; тел.: 88634601753, 88634601812, 88634601807; 89094081126; д.т.н.; профессор; член Европейской Академии Естествознания (EuANH); зав. кафедрой информатики.

Дзюба Андрей Сергеевич – e-mail: dzuba_ni@aaanet.ru; тел.: 89094080776; аспирант.

Romm Yakov Evseevich – Taganrog State Pedagogical Institute; e-mail: romm@List.ru; 48, Initsiativnaya street, 347926, Russia; phones: +78634601753, +78634601812, +78634601807, +79094081126; dr. of eng. sc.; professor; the member of European Academy of Natural History (EuANH); head of the chair of information science.

Dzuba Andrew Sergeevich – e-mail: dzuba_ni@aaanet.ru; phone: +79094080776; postgraduate student.

УДК 519.6:681.3

Я.Е. Ромм, И.И. Стаховская

ВРЕМЕННАЯ СЛОЖНОСТЬ ПАРАЛЛЕЛЬНОГО КУСОЧНО-ИНТЕРПОЛЯЦИОННОГО ВЫЧИСЛЕНИЯ ФУНКЦИЙ

Представлены алгоритмы и оценки временной сложности параллельного кусочно-полиномиального вычисления функций одной действительной переменной на основе интерполяционного полинома Ньютона. Для рассматриваемой схемы учитывается вариация степени полинома и количества подынтервалов, различные границы погрешности, предложена максимально параллельная форма вычислительного алгоритма. На модели неветвящихся параллельных программ достигается оценка временной сложности $O(\log_2 N)$, *где* N *– заданная граница вариации степени полинома, количество процессоров соответствует максимальному числу подынтервалов и проверочных точек. Для стандартных функций достигается временная сложность вычисления* $O(1)$. *Синхронно со всеми полиномиальными приближениями функций выполнимо параллельное вычисление их производных и определенных интегралов.*

Кусочно-полиномиальное вычисление функций; параллельные вычисления; временная сложность; интерполяционный полином Ньютона.

Ya.E. Romm, I.I. Stakhovskaya

TIME COMPLEXITY OF PARALLEL PIECEWISE POLYNOMIAL CALCULATION OF FUNCTION EVALUATION

In the article the algorithms and the time complexity evaluation for parallel piecewise polynomial calculation of univariate functions on a base of Newton's interpolating polynomials are resented. A polynomial degree and a number of subintervals are algorithmically variated for an a priori different given error bound; the massive parallel form of the computational algorithm was proposed. It was depicted in a model of nonbranching parallel program that we can achieve evaluation of time complexity $T(R) = O(\log_2 N)$, *where N is a prescribed limit of polynomial degree variation and processor number complies a maximum number of subintervals and checkpoints. For standard functions it is* $T(1) = O(1)$. *Parallel approximate computing of derivatives and integrals is available synchronously along with the polynomial approximation of a function.*

Piecewise polynomial calculating of functions; parallel computation; time complexity; Newton's interpolating polynomials.

Постановка вопроса. Схемы кусочно-полиномиальной аппроксимации изначально рассматривались [1, 2] в качестве оптимальных машинных алгоритмов вычисления функций. Аппроксимирующий полином при этом строится на каждом подынтервале объединения, покрывающего основной интервал. Длину подынтервала для заданной степени полинома можно выбрать столь малой, чтобы приближение на нем не превышало априори заданной границы погрешности. На этой основе возможна высокая точность приближения полиномом малой степени.

Ниже оценивается кусочно-полиномиальная аппроксимация функций на основе интерполяционного полинома Ньютона, которая позволяет сохранить общность интерполяции, и отличается сравнительно высокой точностью.

С целью ускоренного вычисления функций синтезируются параллельные алгоритмы на модели неветвящихся параллельных программ. Модель понимается как последовательность параллельных арифметических команд [3]. Временная сложность (время выполнения) алгоритма измеряется количеством последовательных шагов представляющей его неветвящейся параллельной программы, обозначается $T(R)$, где R – количество процессоров. Модель абстрагируется от архитектуры и обмена [3], интерпретируется как идеальный параллельный процессор [4].

Для рассматриваемой схемы вычисления функций в условиях вариации степени полинома и числа подынтервалов ставится задача дать максимально параллельную форму вычислительного алгоритма и оценить его временную сложность.

Рассматривается функция

$$y = f(x),\ x \in [a,b],\ \ f(x) \in R^2, \tag{1}$$

где $[a,b]$ – произвольно фиксированный отрезок из области допустимых значений. Выбирается система подынтервалов равной длины:

$$[a,b] = \bigcup_{i=0}^{P-1}[x_i, x_{i+1})\,, P = 2^k,\ k \in \{0,1,\ldots\}\,. \tag{2}$$

Таким образом,

$$x_{i+1} - x_i = (b-a)/P,\ \ i = 0, 1, \ldots, P-1.$$

Для каждого подынтервала с номером i строится интерполяционный полином Ньютона с равноотстоящими узлами степени n, где n – выбирается минимальной для достижения априори заданной точности приближения на всех подынтервалах. При этом полином Ньютона на i-м подынтервале преобразуется к виду:

$$\Psi_{ni}(x) = \tilde{a}_{0if} + \tilde{a}_{1if}x + \tilde{a}_{2if}x^2 + \ldots + \tilde{a}_{nif}x^n \quad x \in [x_i, x_{i+1}),\ n = \text{const},\ i = 0, 1, \ldots, P-1. \quad (3)$$

В (3) f соответствует аппроксимируемой функции (1), на каждом из подынтервалов не должна превышаться априори заданная граница погрешности

$$\left| f(x) - \Psi_{ni}(x) \right| \le \varepsilon, \quad x \in [x_i, x_{i+1}), \quad i = 0, 1, \ldots, P-1. \quad (4)$$

Проверку (4) можно выполнить по системе проверочных точек, число которых на каждом подынтервале равно m, неравенство проверяется в каждой из них на каждом подынтервале. Минимальность степени n обеспечивается следующим образом. Построение (3) и проверка (4) начинается с $n = 1$ и $k = 0$. При нарушении неравенства (4) хоть в одной проверочной точке значение k увеличивается на единицу, и так – до априори заданной границы $k \le K$. Если в результате заданная точность не достигнута, то снова полагается $k = 0$, при этом степень полинома увеличивается на единицу, затем, при нарушении (4), снова выполняется переход к минимальному k, и так – до априори заданной границы $n \le N$. Фиксируется наименьшее n, при котором (4) выполняется во всех проверочных точках всех 2^k подынтервалов, в соответствии с этим фиксируется k.

Рассматриваемый полином Ньютона на подынтервале $[x_i, x_{i+1}]$ записывается в виде $\Psi_{ni}(x) = y_{i0} + \sum_{j=1}^{n} \Delta^j y_{i0} / (j! h^j) \prod_{k=0}^{j-1} (x - x_{ik})$, где $y_{ir} = f(x_{ir})$, $\Delta^j y_{i0}$ – конечная разность j-го порядка в точке x_{i0}, h – расстояние между узлами интерполяции:

$$h = (x_{i+1} - x_i) / n. \quad (5)$$

Узлы примут вид: $x_{ij} = x_i + jh,\ j = 0, 1, \ldots, n-1.$ Обозначив $t = (x - x_{i0}) / h$, полином $\Psi_{ni}(x)$ можно записать в виде

$$\Psi_{ni}(t) = y_{i0} + \sum_{j=1}^{n} b_{ij} \prod_{k=0}^{j-1} (t - k), \quad b_{ij} = \Delta^j y_{i0} / j!. \quad (6)$$

Вариационные свойства излагаемых алгоритмов заимствуются из [5, 6].

Вначале рассматривается случай часто встречающейся функции, например, стандартной. В этом случае коэффициенты аппроксимирующего полинома вычисляются априори для всех подынтервалов, хранятся в виде массива в памяти компьютера. При вычислении функции по значению аргумента дешифрируется номер подынтервала в качестве адреса коэффициентов аппроксимирующего полинома: $i = \text{int}(x - a/\rho)$, int – целая часть числа, $\rho = x_{i+1} - x_i$, $i = \overline{0, P-1}$, $x \in [x_i, x_{i+1})$. В [5, 6] приводятся программы аппроксимации функций на данной основе.

При вычислении полинома по схеме Горнера время вычисления функции составит $T(1) = n\,(t_y + t_c)$, где t_y, t_c – время бинарного умножения и сложения. Отсюда $T(1) = O(1)$, поскольку степень полинома минимальна и постоянна.

При вычислении функции общего вида минимизация времени может быть достигнута за счет преобразования алгоритмов к параллельной форме. Вычислить $\Delta^j y_{i0}$ по максимально параллельной форме можно на основе выражения [7]:

$$\Delta^j y_{i0} = \sum_{\ell=0}^{j} (-1)^{\ell} \; C_j^{\ell} y_{i(j-l)}, \tag{7}$$

где $y_{ir} = f(x_{ir})$, биноминальные коэффициенты C_j^{ℓ} можно считать априори вычисленными и хранимыми в памяти компьютера для всех $j \le n$ и $n \le N$. Отсюда $\Delta^j y_{i0} = \sum_{\ell=0}^{j} \alpha_{\ell} y_{i(j-\ell)}$, где α_{ℓ} – известны. Можно априори вычислить сразу b_{ij} из (6): $b_{ij} = \sum_{\ell=0}^{j} \tilde{\alpha}_{\ell} \; y_{i(j-\ell)}$, где $\tilde{\alpha}_{\ell} = (-1)^{\ell} C_j^{\ell} / j!$. Правая часть (7) вычисляется одновременным умножением $\tilde{\alpha}_{\ell} \times y_{i(j-\ell)}$, $\ell = 0, 1, ..., j$, за время t_y на $j+1$-процессорах. С применением схемы сдваивания для дальнейшего вычисления (7) потребуется не более $\lceil \log_2(j+1) \rceil$ последовательных шагов, искомая оценка примет вид

$$T(j+1) = t_y + \lceil \log_2(j+1) \rceil t_c .$$

Значения $\Delta^j y_{i0}$ можно получить параллельно по всем $j = 1, 2, ..., n$. С учетом времени t_f вычисления функции в одном узле из правой части (7) имеет место.

Лемма 1. Все конечные разности $\Delta^j y_{i0}$ и коэффициенты из (7) для полинома (6) на каждом подынтервале можно вычислить параллельно за время

$$T((n^2 + 3n)/2) = t_f + t_y + \lceil \log_2(n+1) \rceil t_c .$$

Каждое произведение $P_{nj}(t) = \prod_{k=0}^{j-1} (t-k)$ представляет собой разложение полинома по корням $k = 0, 1, \ldots, j-1$, поэтому коэффициенты полинома

$$P_{nj}(t) = d_{0j} + d_{1j}\, t + d_{2j}\, t^2 + ... + d_{nj}\, t^j$$

можно вычислить априори и хранить в памяти компьютера для всех требуемых значений j [8], ниже они предполагаются известными. Поиск окончательных коэффициентов полинома (3) с переходом от x к t выполняется путем приведения подобных и приравнивания коэффициентов в (6):

$$y_{i0} + \sum_{j=1}^{n} b_{ij} \sum_{l=1}^{j} d_{\ell j} t^{\ell} = y_{i0} + \sum_{\ell=1}^{n} \sum_{j=\ell}^{n} (b_{ij}\, d_{\ell j}) t^{\ell} .$$

Отсюда

$$\Psi_{ni}(t) = a_{0if} + \sum_{\ell=1}^{n} a_{\ell if} t^{\ell}, \; a_{0if} = y_{i0}, a_{\ell if} = \sum_{j=\ell}^{n} b_{ij} d_{\ell j} . \tag{8}$$

Лемма 2. Коэффициенты полинома из (8) на i-м подынтервале можно вычислить параллельно с временной сложностью

$$T((n^2 + 3n)/2) = t_f + 2t_y + \left(\lceil \log_2(n+1) \rceil + \lceil \log_2(n-1) \rceil \right) t_c, \tag{9}$$

что влечет $T(n^2/2) \sim 2\log_2 n \cdot t_c = O(\log_2 n)$.

Если данное построение полиномов (8) выполнено одновременно для всех 2^k подынтервалов, то правая часть (9) не изменится, а число процессоров согласно (2) возрастет в 2^k раз. Отсюда и из леммы 2 вытекает:

Теорема 1. Коэффициенты полиномов (8) на всех 2^k подынтервалах можно параллельно вычислить с временной сложностью

$$T(((n^2+3n)/2)\times 2^k)=t_f+2t_y+\left(\left\lceil \log_2(n+1)\right\rceil+\left\lceil \log_2(n-1)\right\rceil\right)t_c, \quad (10)$$

эквивалентная форма (10): $T(n^2\times 2^{k-1})\sim 2\log_2 n\cdot t_c=O(\log_2 n)$.

С учетом числа m проверочных точек и схемы Стоуна [3, 9] имеет место.

Теорема 2. Коэффициенты полиномов (8) на 2^k подынтервалах можно вычислить параллельно и выполнить проверки неравенств (4) по всем подынтервалам за время

$$T(R)\sim \log_2 n\times(t_y+3t_c)+t_f=O(\log_2 n), \quad (11)$$

где $R=\max(n\times m\times 2^k,\ (n^2+3n)\times 2^{k-1})$.

Пусть в дополнение к теореме 2 требуется определить условия истинности всех проверочных неравенств в пределах изменения всех $k\le K$. Проверку можно выполнить одновременно для всех $k=0,1,\ldots,K$. Отсюда вытекает

Следствие 1. В рассматриваемых условиях имеет место утверждение теоремы 2 и оценка (11), где $R=\max(n\times m\times(2^{K+1}-1),\ (n^2+3n)\times(2^K-1))$.

Предположим, что построение интерполяционных полиномов (8) выполнено одновременно для всех 2^k подынтервалов при каждом $k=0,1,\ldots,K$, при этом степень полинома принимает одновременно все возможные значения $n=1,2,\ldots,N$, $N=\text{const}$. Если искомую проверку выполнять максимально параллельно по всем рассматриваемым n, то правая часть оценки теоремы 2 примет вид:

$$T(R)\sim\log_2 N\times(t_y+t_c)+t_f+2t_y+\left(\left\lceil \log_2(N+1)\right\rceil+\left\lceil \log_2(N-1)\right\rceil\right)t_c,$$

$$R\sim\max((N^2/2)\times m\times(2^{K+1}-1),\ (N^3/3+3N^2/2)\times(2^K-1)).$$

Полином вида (8) влечет табличный вид производной: $f'(x)\approx\frac{1}{h}\sum_{\ell=1}^{n}\ell a_{i\,\ell}t^{\ell-1}$, $t=(x-x_i)/h$. Вместе с коэффициентами аппроксимирующих полиномов получаются коэффициенты полиномов, аппроксимирующих производные тех же функций. В [5] на основании аналогичного формирования коэффициентов первообразных выводится формула $\int_\alpha^\beta f(x)\,dx\approx h\sum_{i=0}^{P-1}\int_0^n\Psi_{in}(t)\,dt$, где $P=2^k$ – число подынтервалов разбиения промежутка интегрирования, h из (5). Окончательно, $\int_\alpha^\beta f(x)\,dx\approx h\sum_{i=0}^{P-1}\sum_{\ell=0}^{n}\frac{a_{\ell i\,f}\,n^{\ell+1}}{\ell+1}$. Чтобы параллельно вычислить правую часть, достаточно применить схему сдваивания к готовым коэффициентам, что повлечет время $O(\log_2 P+\log_2 n)$, или, с учетом границ числа подынтервалов и степеней, – $O(K+\log_2 N)$. Таким образом, наряду с функцией одновременно могут быть вычислены производные и интегралы с дополнительным количеством процессоров.

Заключение. Представлены максимально параллельные формы алгоритмов приближенного вычисления функций одной действительной переменной на основе варьируемого кусочно-полиномиального метода, использующего интерполяционные полиномы Ньютона. Показано, что вычисление произвольной функции осуществимо за время $T(R) = O\ (\log_2 N)$, где N – заданная граница вариации степени полинома. Наряду с рассматриваемыми полиномиальными приближениями функции одновременно выполнимо параллельное вычисление ее производных и определенных интегралов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Голубков Ю.А.* К правильному выбору алгоритмов аппроксимации функций для ЭВМ, работающих в реальном масштабе времени // Электронные вычислительные машины. – М.: ИТМ и ВТ АН СССР, 1965. – Вып. 3. – С. 115-154.
2. *Лебедев В.Н.* Введение в системы программирования. – М.: Статистика, 1975. – 312 с.
3. *Солодовников В.И.* Верхние оценки сложности, решения систем линейных уравнений. – В кн.: Теория сложности вычислений. Ч. I: Записки научных семинаров ЛОМИ АН СССР. – Л. – 1982. Т. 118. – С. 159-187.
4. *Котов В.Е.* О связи алгебраических и архитектурных аспектов параллельных вычислений. – В кн.: Вычислительные процессы и системы / Под ред. Г.И. Марчука. – М.: Наука, 1983. – С. 54-80.
5. *Аксайская Л.Н.* Разработка и исследование параллельных схем цифровой обработки сигналов на основе минимизации временной сложности вычисления функций. Автореф. дис. … канд. техн. наук. – Таганрог: ЮФУ, 2008. – 18 с.
6. *Джанунц Г.А.* Компьютерный метод кусочно-полиномиального приближения решений обыкновенных дифференциальных уравнений в применении к моделированию автоколебательных реакций. Автореф. дис. … канд. техн. наук. – Таганрог: ЮФУ, 2012. – 22 с.
7. *Пулькин С.П., Никольская Л.Н., Дъячков А.С.* Вычислительная математика. – М.: Просвещение, 1980. – 176 с.
8. *Ромм Я.Е.* Локализация и устойчивое вычисление нулей многочлена на основе сортировки. Ч. II // Кибернетика и системный анализ. – 2007. – № 2. – С. 161-174.
9. *Stone H.S.* Problems of parallel computation. – In: Complexity of Sequent. Paral Numer. Algor. // Ed. T.F. Traub. N.Y.: Acad. Press, 1973. – P. 1-16.

Статью рекомендовал к опубликованию д.т.н., профессор Л.П. Фельдман.

Ромм Яков Евсеевич – ФГБОУ ВПО "Таганрогский государственный педагогический институт им. А.П. Чехова"; e-mail: romm@list.ru; 347926, г. Таганрог, ул. Инициативная, 48; тел.: 88634601753, 88634601812, 88634601807; 89094081126; д.т.н.; профессор; член Европейской Академии Естествознания (EuANH); зав. кафедрой информатики.

Стаховская Ирина Илларионовна – e-mail: irihkast@gmail.ru; тел.: 89034604006; студентка.

Romm Yakov Evseevich – Taganrog State Pedagogical Institute; e-mail: romm@List.ru; 48, Initsiativnaya street, 347926, Russia; phones: +78634601753, +78634601812, +78634601807, +79094081126; dr. of eng. sc.; professor; the member of European Academy of Natural History (EuANH); head of the chair of information science.

Stakhovskaya Irina Illarionovna – e-mail: irihkast@gmail.ru; phone: +79034604006; student.

УДК 543.421:621.38

А.В. Вовна, А.А. Зори

СПОСОБ КОМПЕНСАЦИИ ТЕМПЕРАТУРНОГО ДРЕЙФА ОПТИЧЕСКОГО ИЗМЕРИТЕЛЯ КОНЦЕНТРАЦИИ ГАЗА

Рассмотрены вопросы повышения точности измерительного контроля концентрации метана оптико-абсорбционного газоанализатора. Разработан и исследован макетный образец быстродействующего оптического измерителя концентрации метана. Установлено: чувствительность по выходному напряжению составляет порядка 0,51 В/$^{об.}$%, абсолютная аддитивная погрешность измерения концентрации метана ±0,02$^{об.}$%, что на порядок меньше, чем абсолютная погрешность измерения концентрации метана (не более