Параллельные интерполяционные алгоритмы численного решения задачи Коши для обыкновенных дифференциальных уравнений на SIMD-компьютере

Фельдман Л.П.
Кафедра ПМиИ ДонГТУ
E-mail: feldman@r5.dgtu. donetsk.ua

Abstract

Feldman L.P. Parallel interpolation algorithms of a numerical solution of a Cauchy problem for the ordinary differential equations on SIMD the computer
The parallel block algorithms of a numerical solution of a Cauchy problem for the ordinary differential equations are considered. Output of design formulas for one-step and multistage four dot methods and rating of their accuracy are reduced. For the obtained algorithms of parallel solution the execution times and accelerations are defined in view of time of data exchanges in vector SIMD. It is showed, that parallel algorithm of a type the predictor-proofer as compared to one-step four by dot parallel algorithm at the same acceleration and complexity has accuracy on three order above.

Введение

Опыт эксплуатации современных высокопроизводительных компьютерных систем убедительно показал, что эффективность их характеристик существенно зависит от согласования структуры параллельных вычислительных систем с классом решаемых задач и используемых численных методов. Отсюда следует, что распараллеливание существующих алгоритмов и разработка новых параллельных алгоритмов является машинно-зависимой задачей. Поэтому рассмотрим методы построения и оценки точности параллельных алгоритмов, эффективных для процессоров SIMD-структуры.

Для упрощения изложения рассмотрим случай одного обыкновенного дифференциального уравнения первого порядка:

$$\frac{dx}{dt} = f(t,x), \quad x(t_0) = x_0. \tag{1}$$

Обозначим через $x(t)$ точное решение дифференциального уравнения, а через $u_n = u(t_n)$ – приближенное значение решения в точке t_n .Множество k точек равномерной сетки t_{n+j} ,$j=1,2,...k$ с шагом τ назовем *блоком,* а точку t_n началом блока. Алгоритм решения задачи Коши (1), позволяющий параллельно в каждой точке блока вычислять значение приближенного решения, построим на основе формул *с забеганием вперед* [1].

$$u_{n+i} = u_n + \tau \sum_{j=-p}^{k} b_{i,j} f(t_{n+j}, u_{n+j}), \quad i = \overline{1,k}. \tag{2}$$

Блочные методы особенно удачно реализуются на параллельных вычислительных системах. Известны два типа блочных методов: одношаговый блочный метод и блочный метод типа предиктор-корректор [2]. В первом методе только последняя точка блока используется в следующем блоке, тогда как во втором методе используются все точки предшествующего блока. Обозначим через

$F_{n+i} = f(t_{n+i}, u_{n+i})$ для $i = 0,1,2,\ldots,k$

и построим по этим значениям интерполяционный многочлен $\mathbf{L_k(t)}$. Тогда формулы (2) можно получить с помощью следующих интегралов:

$$\mathbf{u_{n+i} = u_n + \int_{t_{n-p}}^{t_{n+i}} L_k(t)dt, i = 1,2,...,k.} \tag{3}$$

При этом, как будет показано ниже, общее количество точек **k+p**, используемых в формулах (3), определяет порядок точности метода, а число точек в блоке **k** – ускорение при параллельном выполнении алгоритма. Ограничимся здесь рассмотрением лишь четырехточечных блочных методов, т.к. вывод расчетных формул и оценка их точности для других случаев могут быть получены аналогично.

1. Одношаговый четырехточечный блочный метод

Выведем формулы, позволяющие вычислять значения приближенного решения параллельно в четырех узлах блока, используя при этом только одно известное значение в последней точке предыдущего блока. С помощью системы *Mathematica 3* получим формулы, соответствующие (3) при **p=0 и k=4.**

```
T4 = Table[{tn + i * τ, F_{n+i}}, {i, 0, 4}];
P4[t_] = InterpolatingPolynomial[T4, t];
U = {"u_{n+1}=", "u_{n+2}=", "u_{n+3}=", "u_{n+4}="};
Do[u_{n+i} = u_n + Simplify[Integrate[P4[t], {t, tn, tn + i * τ}]];
 Print[U[[i]], u_{n+i}], {i, 4}]
```

$$u_{n+1} = \frac{1}{720}\tau\,(251\,F_n + 646\,F_{1+n} - 264\,F_{2+n} + 106\,F_{3+n} - 19\,F_{4+n}) + u_n$$

$$u_{n+2} = \frac{1}{90}\tau\,(29\,F_n + 124\,F_{1+n} + 24\,F_{2+n} + 4\,F_{3+n} - F_{4+n}) + u_n \tag{4}$$

$$u_{n+3} = \frac{3}{80}\tau\,(9\,F_n + 34\,F_{1+n} + 24\,F_{2+n} + 14\,F_{3+n} - F_{4+n}) + u_n$$

$$u_{n+4} = \frac{2}{45}\tau\,(7\,F_n + 32\,F_{1+n} + 12\,F_{2+n} + 32\,F_{3+n} + 7\,F_{4+n}) + u_n$$

Оценим порядок точности полученных для четырехточечного блока приближенных формул (4). Для этого найдем погрешность аппроксимации каждого из уравнений (4) на решении исходного уравнения (1). Разложим в ряды в окрестности точки $\mathbf{t_n}$ функции $\mathbf{x_{n+i}}$ и производные $\mathbf{x'_{n+i}}$, входящие в формулы для невязок:

```
x_n = x[t_n]; x'_n = x'[t_n];
Do[x_{n+i} = Series[x[t_n + i * τ], {τ, 0, 6}], {i, 4}];
Do[x'_{n+i} = Series[x'[t_n + i * τ], {τ, 0, 5}], {i, 4}];
```

$$r_{n+1} = \frac{1}{720}(251\,x'_n + 646\,x'_{1+n} - 264\,x'_{2+n} + 106\,x'_{3+n} - 19\,x'_{4+n}) + (x_n - x_{n+1})\,/\,\tau;$$

$$r_{n+2} = \frac{1}{180}(29\,x'_n + 124\,x'_{1+n} + 24\,x'_{2+n} + 4\,x'_{3+n} - x'_{4+n}) + (x_n - x_{n+2})\,/\,\tau\,/\,2;$$

$$r_{n+3} = \frac{1}{80}(9\,x'_n + 34\,x'_{1+n} + 24\,x'_{2+n} + 14\,x'_{3+n} - x'_{4+n}) + (x_n - x_{n+3})\,/\,\tau\,/\,3;$$

$$r_{n+4} = \frac{1}{90}(7\,x'_n + 32\,x'_{1+n} + 12\,x'_{2+n} + 32\,x'_{3+n} + 7\,x'_{4+n}) + (x_n - x_{n+4})\,/\,\tau\,/\,4;$$

```
R = {"r_{n+1}=", "r_{n+2}=", "r_{n+3}=", "r_{n+4}="}; Do[Print[R[[i]], r_{n+i}], {i, 4}];
```

$$r_{n+1} = -\frac{3}{160}x^{(6)}[t_n]\,\tau^5 + O[\tau]^6 \qquad r_{n+2} = -\frac{1}{180}x^{(6)}[t_n]\,\tau^5 + O[\tau]^6$$

$$r_{n+3} = -\frac{1}{160}x^{(6)}[t_n]\,\tau^5 + O[\tau]^6 \qquad r_{n+4} = O[\tau]^6$$

В последних формулах значения x'_{n+i}, $i=0,1,\dots,4$ вычисляются на точном решении, т. е. $x'_{n+i}=f(t_{n+i},x_{n+i})$. Таким образом, порядок аппроксимации первых трех уравнений (4) равен $O(\tau^5)$, в то же время последнее уравнение имеет порядок аппроксимации $O(\tau^6)$.

Систему уравнений (4) можно решить с помощью следующего алгоритма:

$$u_{n+i,0}=u_n+i\tau F_n,\ i=1,2,3,4,$$
$$u_{n+1,s+1}=\frac{1}{720}\tau\,(251F_n+646F_{1+n,s}-264F_{2+n,s}+106F_{3+n,s}-19F_{4+n,s})+u_n,$$
$$u_{n+2,s+1}=\frac{1}{90}\tau\,(29F_n+124F_{1+n,s}+24F_{2+n,s}+4F_{3+n,s}-F_{4+n,s})+u_n,$$
$$u_{n+3,s+1}=\frac{3}{80}\tau\,(9F_n+34F_{1+n,s}+24F_{2+n,s}+14F_{3+n,s}-F_{4+n,s})+u_n,$$
$$u_{n+4,s+1}=\frac{2}{45}\tau\,(7F_n+32F_{1+n,s}+12F_{2+n,s}+32F_{3+n,s}+7F_{4+n,s})+u_n,\ s=\overline{0,4}. \quad (5)$$

Покажем, что выполнение четырех шагов вычислений по формулам (5) обеспечат точность приближенного решения порядка $O(\tau^5)$. Обозначим через $u_n^{[k]}$ значение решения в точке t_n, найденное с точностью $O(\tau^k)$, а через

$$F_n^{[k]}=f(t_n,u_n^{[k]}) \quad (6)$$

вычисленное при этом значение правой части уравнения. Допустим, что значение u_n вычислено с точностью $O(\tau^6)$, тогда и значение правой части уравнения может быть определено с такой же точностью, т. е. имеем $F_n^{[6]}$. Выполнив первый шаг вычислений по первой формуле (5), получим

$$u_{n+i,0}^{[1]}=u_n^{[6]}+i\tau F_n^{[5]},\ i=\overline{1,4},$$

так как точность формулы Эйлера имеет порядок $O(\tau)$. Для сокращения записей дальнейшие оценки приведем только для второй формулы из (5), так как оценки для остальных получаются аналогично.

$$u_{n+1,1}^{[2]}=u_n^{[6]}+\frac{1}{720}\tau\left(251F_n^{[5]}+646F_{n+1,0}^{[1]}-264F_{n+2,0}^{[1]}+106F_{n+3,0}^{[1]}-19F_{n+4,0}^{[1]}\right).$$

Вычисления на шаге s =1 дадут следующий результат

$$u_{n+1,2}^{[3]}=u_n^{[6]}+\frac{1}{720}\tau\left(251F_n^{[5]}+646F_{n+1,1}^{[2]}-264F_{n+2,1}^{[2]}+106F_{n+3,1}^{[2]}-19F_{n+4,1}^{[2]}\right).$$

Шаг s =2 приведет к следующей оценке

$$u_{n+1,3}^{[4]}=u_n^{[6]}+\frac{1}{720}\tau\left(251F_n^{[5]}+646F_{n+1,2}^{[3]}-264F_{n+2,2}^{[3]}+106F_{n+3,2}^{[3]}-19F_{n+4,2}^{[3]}\right).$$

При s = 3 достигается предельно возможная точность, определяемая порядком аппроксимации уравнения (1), для рассматриваемой формулы. В самом деле,

$$u_{n+1,4}^{[5]}=u_n^{[6]}+\frac{1}{720}\tau\left(251F_n^{[5]}+646F_{n+1,3}^{[4]}-264F_{n+2,3}^{[4]}+106F_{n+3,3}^{[4]}-19F_{n+4,3}^{[4]}\right).$$

т.е. получено приближенное решение с точностью $O(\tau^5)$. Последующие итерации по формулам (5) не приведут к повышению порядка точности получаемых результатов в первых трех узлах блока. Они необходимы лишь в том случае, если требуется получить приближенное решение в последнем узле с точностью $O(\tau^6)$. Выполнив шаг $s=4$, получим в первых трех узлах приближенное решение по-прежнему с точностью порядка $O(\tau^5)$, а для приближенного значения в последнем узле блока получим оценку

$$u_{n+4,4}^{[6]}=u_n^{[6]},+\frac{1}{45}\tau\left(7F_n^{[5]}+32F_{n+1,4}^{[5]}+12F_{n+2,4}^{[5]}+32F_{n+3,4}^{[5]}+7F_{n+4,4}^{[5]}\right),$$

Найдем время выполнения алгоритма, определяемого формулами (5), на одном процессоре. Обозначим через $\mathbf{t_f}$ время вычисления значения функции $\mathbf{f(t,x)}$, а через $\mathbf{t_{ad}}$, $\mathbf{t_{mul}}$ время выполнения операции сложения и умножения соответственно. Время вычисления приближенных значений решения с точностью $\mathbf{O}(\tau^5)$ во всех четырех узлах блока составит

$$T_s=17\, t_f+72\, t_{ad}+24\, t_{mu}.$$

Для параллельного выполнения вычислений по формулам (5) закрепим за каждым узлом блока процессор. Объединим процессоры в кольцо, чтобы иметь возможность одновременной передачи данных соседним процессорам. Обозначим через $\mathbf{t_{ta}}$ – время передачи числа соседнему процессору. Время параллельного вычисления приближенных значений решения с той же точностью для всех четырех узлов блока составит

$$T_p=5\, t_f+18\, t_{ad}+6\, t_{mu}+12\, t_{ta}.$$

Таким образом, ускорение параллельного одношагового четырехточечного алгоритма будет равно

$$W(4)= T_s / T_p=(17\, t_f+72\, t_{ad}+24\, t_{mu})/(5\, t_f+18\, t_{ad}+6\, t_{mu}+12t_{ta}) \qquad (7)$$

Если учитывать только время вычислений правой части уравнения, т.к. времена выполнения арифметических операций и обмена значительно меньше времени вычисления правой части, то ускорение четырехточечного параллельного алгоритма можно считать приближенно равным

$$W(4)_{pot}=4.$$

Для **k** – точечного блока имеет место оценка

$$W(k)_{pot}=k.$$

2. Многошаговый четырехточечный блочный метод

Блок состоит из четырех узлов, при построении вычислительных формул используются найденные на предыдущем этапе приближенные значения решения в четырех узлах предшествующего блока. Формулы для вычислений на первом шаге получим, использовав экстраполяционную формулу Адамса, узлами интерполяции которой являются точки предшествующего блока

```
T4 = Table[{tn + i * τ, Fn+i}, {i, -3, 0}];
P4[t_] = InterpolatingPolynomial[T4, t];
U0 = {"un+1,0=", "un+2,0=", "un+3,0=", "un+4,0="};
Do[un+i = un + Simplify[Integrate[P4[t], {t, tn, tn + i * τ}]];
 Print[U0[[i]], un+i], {i, 4}]
```

$$u_{n+1,0}=\frac{1}{24}\tau\,(-9\,F_{-3+n}+37\,F_{-2+n}-59\,F_{-1+n}+55\,F_n)+u_n$$

$$u_{n+2,0}=\frac{1}{3}\tau\,(-8\,F_{-3+n}+31\,F_{-2+n}-44\,F_{-1+n}+27\,F_n)+u_n$$

$$u_{n+3,0}=\frac{-3}{8}\tau\,(25\,F_{-3+n}-93\,F_{-2+n}+123\,F_{-1+n}-63\,F_n)+u_n \qquad (8)$$

$$u_{n+4,0}=\frac{-4}{3}\tau\,(18\,F_{-3+n}-65\,F_{-2+n}+82\,F_{-1+n}-38\,F_n)+u_n$$

Формулы (8) имеют точность порядка $\mathbf{O}(\tau^4)$, поскольку являются четырехточечными экстраполяционными формулами Адамса. Поэтому полученное после первого шага $\mathbf{s = 0}$ приближенное решение имеет локальную погрешность $\mathbf{O}(\tau^4)$, если в узлах предшествующего блока решение было получено с такой же или лучшей точностью.

Формулы для следующих итераций получим, использовав (3) при **p=3 и k=4**.

```
T44 = Table[{tn + i * τ, If[i < 1, F_{n+i}, F_{n+i,s}]}, {i, -3, 4}];
P44[t_] = InterpolatingPolynomial[T44, t];
    U = {"u_{n+1,s+1}=", "u_{n+2,s+1}=", "u_{n+3,s+1}=", "u_{n+4,s+1}="};
    Do[u_{n+i} = u_n + Simplify[Integrate[P44[t], {t, tn, tn + i * τ}]];
        Print[U[[i]], u_{n+i}], {i, 4}]
```

$$u_{n+1,s+1}=u_n+\frac{1}{120960}(\tau(-191F_{-3+n}+1879F_{-2+n}-9531F_{-1+n}+68323F_n+68323F_{1+n,s}-9531F_{2+n,s}+1879F_{3+n,s}-191F_{4+n,s}))$$

$$u_{n+2,s+1}=u_n+\frac{1}{3780}(\tau(5F_{-2+n}-72F_{-1+n}+1503F_n+4688F_{1+n,s}+1503F_{2+n,s}-72F_{3+n,s}+5F_{4+n,s}))$$

$$u_{n+3,s+1}=u_n+\frac{1}{4480}(\tau(-13F_{-3+n}+117F_{-2+n}-513F_{-1+n}+2777F_n+3897F_{1+n,s}+5535F_{2+n,s}+1685F_{3+n,s}-45F_{4+n,s})) \qquad (9)$$

$$u_{n+4,s+1}=u_n+\frac{2}{945}\tau(4F_{-3+n}-32F_{-2+n}+108F_{-1+n}-53F_n+892F_{1+n,s}+108F_{2+n,s}+724F_{3+n,s}+139F_{4+n,s})$$

Выполняя итерации по формулам (9) для $s=0,1,2,3$, полученные по формулам (8) значения приближенного решения можно уточнить до достижения предельной точности формул (9) и после четырех шагов точность результатов будет иметь порядок $\mathbf{O(\tau^8)}$. Ниже приведены операторы *Mathematica*, позволяющие выполнить оценку точности формул (9).

```
x_n = x[t_n]; x'_n = x'[t_n]; Do[x_{n+i} = Series[x[t_n + i * τ], {τ, 0, 9}], {i, 4}];
Do[x'_{n+i} = Series[x'[t_n + i * τ], {τ, 0, 8}], {i, -3, 4}];
r_{n+1} = 1/120960 (-191 * x'_{n-3} + 1879 * x'_{n-2} - 9531 * x'_{n-1} +
      68323 * x'_n + 68323 x'_{1+n} - 9531 x'_{2+n} + 1879 * x'_{3+n} - 191 x'_{4+n}) +
   (x_n - x_{n+1}) / τ;
Print["r_{n+1}=", r_{n+1}];
```

$$r_{n+1}=-\frac{2497\,x^{(9)}[t_n]\,\tau^8}{3628800}+O[\tau]^9$$

Аналогично получаются оценки для остальных разностных уравнений

```
r_{n+2} = 1/7560
    (5 * x'_{n-2} - 72 * x'_{n-1} + 1503 * x'_n + 4688 * x'_{1+n} + 1503 x'_{2+n} - 72 * x'_{3+n} + 5 x'_{4+n}) +
   (x_n - x_{n+2}) / τ / 2;
Print["r_{n+2}=", r_{n+2}];
```

$$r_{n+2}=\frac{23\,x^{(9)}[t_n]\,\tau^8}{226800}+O[\tau]^9$$

$$r_{n+3} = \frac{1}{13440}(-13 * x'_{n-3} + 117 * x'_{n-2} - 513 * x'_{n-1} + 2777 * x'_{n} + 3897 x'_{1+n} + 5535 x'_{2+n} + 1685 * x'_{3+n} - 45 x'_{4+n}) + (x_n - x_{n+3}) / \tau / 3;$$

Print["r_{n+3}=", r_{n+3}];

$$r_{n+3} = -\frac{27 x^{(9)}[t_n]\, \tau^8}{44800} + O[\tau]^9$$

$$r_{n+4} = \frac{1}{1890}(4 * x'_{n-3} - 32 * x'_{n-2} + 108 * x'_{n-1} - 53 * x'_{n} + 892 x'_{1+n} + 108 x'_{2+n} + 724 * x'_{3+n} + 139 * x'_{4+n}) + (x_n - x_{n+4}) / \tau / 4;$$

Print["r_{n+4}=", r_{n+4}];

$$r_{n+4} = \frac{107 x^{(9)}[t_n]\, \tau^8}{56700} + O[\tau]^9$$

Если столь высокая точность не требуется, то можно сделать меньшее число итераций.

Заключение

Для получения характеристик последнего алгоритма, так же, как и в случае одношаговых параллельных методов, определим время вычисления приближенных значений решения по формулам (8) и (9) с точностью $O(\tau^8)$ во всех четырех узлах блока на одном процессоре, которое составит

$$T_s = 16\, t_f + 68\, t_{ad} + 64 t_{mu}.$$

Время параллельного вычисления приближенных значений решения с той же точностью для всех четырех узлов блока составит

$$T_p = 4\, t_f + 18\, t_{ad} + 20\, t_{mu} + 15\, t_{ta}.$$

Ускорение параллельного многошагового четырехточечного алгоритма будет равно

$$W_m(4) = (16\, t_f + 68\, t_{ad} + 64 t_{mu}) / (4\, t_f + 18\, t_{ad} + 20\, t_{mu} + 15\, t_{ta}). \qquad (10)$$

Если учитывать только время вычислений правой части уравнения, то ускорение четырехточечного параллельного алгоритма типа предиктор- корректор можно считать приближенно равным

$$W_m(4)_{pot} = 4.$$

Таким образом, параллельный алгоритм типа предиктор-корректор по сравнению с одношаговым четырехточечным параллельным алгоритмом при том же ускорении и трудоемкости имеет точность на три порядка выше.

Аналогично могут быть получены параллельные алгоритмы численного решения задачи Коши для системы обыкновенных дифференциальных уравнений.

Литература

1. Крылов В. И. и др. Вычислительные методы высшей математики - М.: Наука, 1977, 658 с.
2. Системы параллельной обработки -М.: Мир, 1985, 412 с.