РЕАЛИЗАЦИЯ ПРОДОЛЬНО-ПОПЕРЕЧНЫХ ПРОГОНОК
НА ВС "МИНСК-222"

Н.Н. Миленков

Описывается реализация метода расщепления с использованием продольно-поперечных прогонок на примере решения задачи обтекания тела несжимаемой вязкой жидкости. Приводятся оценки времени счета для произвольного числа машин (C) и точек (N).

I. В осесимметрическом случае движение несжимаемой вязкой жидкости описывается системой [1,2]:
\[
\frac{\partial u}{\partial t} + \left[ u \frac{\partial u}{\partial x} + u \frac{\partial u}{\partial y} + v \left( \frac{\partial v}{\partial x} + \frac{\partial v}{\partial y} \right) \right] f_1(y) = 0,
\]
\[
\frac{\partial v}{\partial t} + \left[ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + v \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \right) \right] f_2(y) = 0,
\]
\[
\frac{\partial (\rho + u^2 + v^2)}{\partial t} + \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) f_3(y) = 0,
\]
где \(u\) - \(x\)-компонента скорости, \(\rho\) - давление, \(\nu\) - коэффициент вязкости, \(f_s(y)\) (\(s=1,2,3\)) - специальные функции, связанные с численным методом решения; производные по времени добавлены для построения релаксационного процесса, при котором присоединяется определенный начальный и краевой условия.

Методом расщепления двухмерная задача сводится к последовательности одномерных [1,2].

На каждом дробном шаге \(t=(n+\frac{1}{2})\epsilon\) решается система:
\[
\frac{1}{2} \frac{\partial u}{\partial t} + \left( \frac{\partial f_1}{\partial x} + \frac{\partial f_1}{\partial y} \right) f_1(y) = 0,
\]
\[
\frac{1}{2} \frac{\partial v}{\partial t} + \left( \frac{\partial f_2}{\partial x} + \frac{\partial f_2}{\partial y} \right) f_2(y) = 0,
\]
\[
\frac{1}{2} \frac{\partial f_3}{\partial t} + \frac{\partial f_3}{\partial x} + \frac{\partial f_3}{\partial y} = 0,
\]
где \(\rho = \rho + u^2 / 2\).

На каждом целом шаге \(t=n\epsilon\) решается система:
\[
\frac{1}{2} \frac{\partial u}{\partial t} + \left[ u \frac{\partial u}{\partial x} - v \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \right) \right] f_1(y) = 0,
\]
\[
\frac{1}{2} \frac{\partial v}{\partial t} + \left[ u \frac{\partial v}{\partial x} - v \left( \frac{\partial v}{\partial x} + \frac{\partial v}{\partial y} \right) \right] f_2(y) = 0,
\]
\[
\frac{1}{2} \frac{\partial f_3}{\partial t} + \left( \frac{\partial f_3}{\partial x} + \frac{\partial f_3}{\partial y} \right) f_3(y) = 0,
\]
где \(Q = \rho + u^2 / 2\).

В качестве начальных условий берутся функции \(u, v, \rho\), согласные на предыдущем шаге. Аппроксимация (I,a) и (I,0) приводит к системам разностных уравнений линейного и явного вида. Первые решаются прогонкой, вторые - явным методом. Расчетные формулы имеют вид:

\[
\dot{Q}_i^{K} = \rho_i^{K} + \frac{(Q_i^{K})^2}{2};
\]
\[
\rho_i^{K} = \frac{1}{\left[ \tau_i^{(m)} - \Delta_i^{K} \right]};
\]
\[
B_i^{K} = \frac{B_i^{K} + \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - \dot{Q}_i^{K}}{Q_i^{(m)} - \Delta_i^{K}};
\]
\[
C_i^{K} = \frac{C_i^{K}}{\left[ \left( Q_i^{(m)} - \Delta_i^{K} \right) (Q_i^{K} + \dot{Q}_i^{K}) - \dot{Q}_i^{K} \right]};
\]
\[
D_i^{K} = \frac{D_i^{K} + \tau_i^{(m)} \left( Q_i^{K} + \dot{Q}_i^{K} \right)}{Q_i^{(m)} - \Delta_i^{K}};
\]
\[
\dot{Q}_i^{K} = \frac{Q_i^{K} - \left( Q_i^{K} \right)^2}{2};
\]
\[
\dot{Q}_i^{(m)} = \frac{Q_i^{K} - \dot{Q}_i^{K}}{2} - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2 - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2 - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2;
\]
\[
\dot{Q}_i^{(m)} = \frac{Q_i^{K} - \dot{Q}_i^{K}}{2} - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2 - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2 - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2;
\]
\[
\dot{Q}_i^{K} = \frac{Q_i^{K} - \dot{Q}_i^{K}}{2} - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2 - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2 - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2;
\]
\[
\dot{Q}_i^{(m)} = \frac{Q_i^{K} - \dot{Q}_i^{K}}{2} - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2 - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2 - \tau_i^{(m)}(Q_i^{K} + \dot{Q}_i^{K}) - (Q_i^{K})^2/2;
где \( N = 3 \times N + 4 \times 4000 \) — число точек прямоугольной области, в которой решается задача;

\( \kappa \) — число кодов в считающейся с магнитной лентой (МЛ) порции;

\( N' = \kappa \) — число порций;

\( N' + 1 \) — число строк разностной сетки в порции;

начальные значения — массы \( U_l, U_r, D_r, P \) — находятся на МЛ;

массивы поперечных прогоночных коэффициентов насыщиваются порциями и хранятся на МЛ;

\( \Phi \) — формирует команды, подготавливающие константы, зависящие от параметров задачи, насыщающие массивы \( \gamma_l(t) \), которые постоянно находятся в МОЗУ;

\( d_{\alpha}(\alpha = 1, \ldots, 20) \) операторы условного перехода, осуществляющие переход к следующему оператору после повторения заданного числа циклов;

\( A_{i, \text{имк}} \) — вычисляет \( f^k \); 

\( B_{i, \text{имк}} \) — вычисляет прогоночные коэффициенты \( A_{l, \text{имк}} \); 

\( C_{l, \text{имк}} \) (при \( j = 1 \)) и \( D_{l, \text{имк}} \) (при \( j = 2 \)); 

\( g_{l, \text{имк}} \) — вычисляет \( U_{l, \text{имк}} + U_{r, \text{имк}} \); 

\( f_{l, \text{имк}} \) — вычисляет \( G_{l, \text{имк}} \); 

\( h_{l, \text{имк}} \) — вычисляет прогоночные коэффициенты \( e_{l, \text{имк}} \); 

\( \alpha_{l, \text{имк}} \) (при \( j = 1 \)) и \( \alpha_{l, \text{имк}} \) (при \( j = 2 \)); 

\( \beta_{l, \text{имк}} \) — вычисляет \( U_{l, \text{имк}} \) (при \( j = 1 \)), \( U_{r, \text{имк}} \) (при \( j = 2 \)); 

\( \gamma_{l, \text{имк}} \) — вычисляет \( P_{l, \text{имк}} \); 

\( \varphi_{l, \text{имк}} \) — вычисляет \( \varphi_{l, \text{имк}} \); 

\( \rho_{l, \text{имк}} \) (\( j = 1, 2, \ldots, 14 \)) — производит обмен с МЛ.

3. Объем памяти, необходимый для решения задачи,

\[ V = 6 	imes N + 20 - M + 800, \]

где первый член — число ячеек для массивов \( U_{l, \text{имк}}, U_{r, \text{имк}}, U_{r, \text{имк}}, U_{l, \text{имк}}, P_{l, \text{имк}}, D_{r, \text{имк}} \); второй — для массивов \( \gamma_l(t) \); третий — для программ и констант.

Для \( N < 1000 \) "Минск-22" решает задачу без использования МЛ. При увеличении \( N \) до 4000 время обмена с МЛ возрастает от нуля до максимума (здесь время приводится в расчете на одну точку). Практически время решения \( t \) составляет из времени счета \( t_{\text{от}} \) операторов \( A_{l, \text{имк}} \) (\( i = 1, 2, \ldots, 8 \)) и времени \( t_{\text{об}} \) обмена с МЛ.

\[ t = t_{\text{от}} + t_{\text{об}} = 0,036 + 0,024 = 0,060 \text{ сек.} \]
где в \( t_0 \) (для \( N > 4000 \)) входит запись \( \Pi 2 \) и сомножение 23 массивов, предполагается, что обмен производится оптимальными порциями [3] и простые, связанные с подводом зон, отсутствуют.

4. Распараллеливание вычислений при продольных прогонках не вызывает затруднений: область решения разделяется на подо- ся, обслуживаемые отдельными машинами.

Схема параллельных вычислений при параллельных прогонках была предложена Э.В. Ервеевским и Ю.Г. Косаревым. Ее реализацией занимался Е.Н. Хук, но он ушел из жизни. Согласно этой схеме, информация, необходимая для непрерывного счета (прогоночные коэффициенты, \( U_{thr}, C_{thr}, Q_{thr} \)), передается следующей или предыдущей машине в соответствии с рис. 1, при этом для запоминания прогоночных коэффициентов в каждой машине достаточно \[ 2(\ell-\ell+\ell)_M \] ячеек памяти, где \( L \) — номер машины.

Рис. 1. Схема параллельных вычислений при параллельной прогонке для системы из \( 3 \times 4 \) машин.

Случаи

1) первая машина вычисляет прогоночные коэффициен- ты, вторая и третья — проставляют;
2) первая передает информацию второй, после чего они ведут прогонку в прямом направлении;
3) вторая передает информацию второй, вторая — третьей, и все машины работают;
4) третья ведет прогонку в обратном направлении, вторая и первая проставляют;
5) все машины вычисляют прогоночные коэффициенты для соответствующих \( \ell \).

б) третья и вторая ведут прогонку в обратном направлении, первая — проставляет;
в) аналогично \( b \);
г) все машины ведут прогонку в обратном направлении, после чего выполняются, чередуясь, случай \( b \) и \( a \) для соответствующих \( \ell \).

Теперь нетрудно видеть, что отношение времени, связанного с особенностями \( BC \), к общему времени счета равно

\[
\theta = \frac{\ell - 1}{\ell_2} \frac{\ell_1}{\ell_2} + \frac{\ell}{M} \frac{18\ell_1}{\ell_2},
\]

где \( \ell \) — время, необходимое для параллельной прогонки (прямой и обратной) в одной точке;
\( \ell_2 \) — время передачи одного кода;
\( \ell_2 \) — время, затрачиваемое одной машиной для реализации шага по времени;
\( M_{max} = 1,66 \).

Первый член характеризует процент простот в обратной — передач. Для \( \ell = 0,012 \) сек, \( \ell_2 = 64 \cdot 10^{-6} \) сек, \( \ell_2 = 0,06 \) сек,
\( M_{max} = 1,66 \).

\[
\theta = 0,35 \ell^{\ell_1} + 0,032 \ell M \text{, то есть при } \ell_1/\ell \leq 0,1 \text{ и } \ell/M < 0,3 \text{ (что для системы всегда имеет место)}.
\]

\( \theta < 5 \% \).

Отсюда видно, что данный метод позволяет распараллеливать процесс вычислений практически на любое число ветвей. В конкретных реализациях при \( \ell < 10 \) можно использовать метод "транс- понирования" массивов вокруг запрограммированной диагонали (рис.2).

Рис. 2. "Транспонирование" массивов для случая 3 машин.

в результате которого в каждой машине вместо строки оказывается соответствующий столбец. При "транспонировании" несколько проще схема счета, пока число параллельных ветвей мало (\( \approx 10 \)), но с увеличением последовательного обмена информацией между ма-шинами становится значительным.
5 Операторная схема параллельного алгоритма

(8)

содержит ρ - операторы, состоящие из одноаковых компонент, последовательных с одноаковыми операторами схемы (2).

ρ - оператор δ₁ передает информацию последующей машине при ведении прогонки вперед, δ₉ - передает информацию предыдущей машине при ведении прогонки назад.

6. Как было показано в п. 4, дополнительные затраты времени, связанные с общностью БС, можно определить по формуле, аналогичной табл. 4.3 на с. 32. Таблица 4.3 показывает, что машина 1 позволяет реализовать алгоритм без использования МЛ, то время счета на ней равно (см. табл. 4.3):

(9)

Таблица

<table>
<thead>
<tr>
<th>N</th>
<th>𝜀</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>36</td>
<td>18</td>
<td>12</td>
<td>9</td>
<td>7,2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td>240</td>
<td>88</td>
<td>54</td>
<td>36</td>
<td>29</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>8000</td>
<td>480</td>
<td>240</td>
<td>123</td>
<td>88</td>
<td>71</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>12000</td>
<td>720</td>
<td>360</td>
<td>240</td>
<td>180</td>
<td>120</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>16000</td>
<td>960</td>
<td>480</td>
<td>320</td>
<td>240</td>
<td>192</td>
<td>123</td>
<td></td>
</tr>
</tbody>
</table>

а отношение времени, затрачиваемого одной машиной, к времени, затрачиваемому системой,

κ=α⋅\frac{ε}{ε_{пред}} = \frac{0,06}{0,036} = 1,66 \cdot \frac{ε}{ε_{пред}}. \quad \text{(10)}

Система может быть эффективно использована и в случае, когда часть информации (а именно значения массивов \( u, v, \rho \) в предыдущий момент времени) находится на МЛ. Например, при \( N = 4000, \, εₚ = 2, \, ε_{сч} = 0,036/2 = 0,018 \text{ сек}, \, ε₂ = 20 \cdot 200 \cdot 10^{-6} = 0,004 \text{ сек}, \) где 20 - число считываемых массивов, 200 масс - время считывания одного шага системой из 2 машин. Здесь коэффициент \( α = 0,06/2 - 0,022 = 1,36. \)

Если соотношение \( ε \) к \( N \) вынуждает систему использовать МЛ в полной мере, то \( α = 1. \) Заметим, что результаты получены в предположении идеальной работы МЛ и отсутствия простоев на подвод зон. Прaktически эффективность системы еще больше.

Выводы

На системе из \( ε \) машин реализуемый алгоритм реализуется в \( α \cdot ε \) раз быстрее, где \( α = (1 - 1,66). \) Программирование данной задачи для системы по сложности и объему практически такое же, как и для одноаковой ЭВМ.

Литература


Поступила в редакцию 30.10.1967 г.