KAZIMOV J.K.

SYNCHRONOUS PARALLEL ALGORITHM FOR AN ITERATED NEWTONIAN PROCESS

Abstract

In this work the development of a parallel synchronized algorithm of an iteration process for SIMD multiprocessor computational systems was considered. An r-parallel algorithm is described by means of control structures and convergence issues studied. And also the acceleration of parallel algorithm is considered.

§1. The construction of problem.

Let $f(x)\in C^d[a,b]$, $f'(x)\neq 0$ for all $x\in[a,b]$. Let's suppose, that algorithm given below is realized on type IBM SIMD with r-processors. At each iteration step by given approximations $x_{01}, x_{02},..., x_{0r}$ of the root z it is required to compute r-numbers of the next approximations:

$$x_{k+1,i} = \varphi_{k,i}\left(x_{k,1}; x_{k-1,1},..., x_{0,r}\right), \tag{1.1}$$

where $i=\overline{1,r}$.

For the parallel computation of the formula (1.1) we use one of local iteration methods with the degree of parallelism which equals to r.

Definition 1. *The local iteration methods with the degree of parallelism consists of iteration functions* $\left\{\Phi_k : R^{(k+1)r} \to R^r,\ k\geq 0\right\}$ *which has the following properties:*

a) $\Phi_k(z,...,z)=(z,...,z)$;

b) for all $k\geq 0$ *there exists open interval* $I\subseteq[a,b]$ *such that*

i) $z\in I$;

ii) $\Phi_k\left(I^{(k+1)r}\right)\subseteq I^r$;

iii) $\left(X_0, X_1,..., X_k\in I^r \wedge X_{k+1}=\Phi(X_k,..., X_0)\right)$

$\Rightarrow \lim\limits_{k\to\infty} x_{k,i}=z$,

where $X_j := \left(x_{j,1},..., x_{j,r}\right)$.

Let $\sigma_k := \max\limits_i \left|z-x_{k,i}\right|$.

Definition 2. *If* $\lim\limits_{k\to\infty}\dfrac{\sigma_{k+1}}{\sigma_k^{\lambda}}=c>0$, *then we'll say that the method has an order of convergence that equals to* λ.

At each step of iteration on the given approximations of the $(k+1)$-th iteration it is required to calculate r-number of the next approximations:

$$x_{k+1,i} = x_{k,i} - \frac{x_{k,i}-x_{k,i+1}}{f(x_{k,i})-f(x_{k,i+1})}\cdot f(x_{k,i}). \tag{1.2}$$

Here $i=1,2,...,r-1$; $k=0,1,2,...$

If $i=r$ then instead of the formula (1.2) we can use the next formulas:

$$x_{k+1,r} = x_{k,r} - \frac{x_{k,r}-x_{k,1}}{f(x_{k,r})-f(x_{k,1})}\cdot f(x_{k,r}). \tag{1.3}$$

The method determined by the formulas (1.2) and (1.3) is called the parallel method of r-sections. It is a discrete analogue of Newton's method from which we get it by the substitution of derivatives by the relation of differences. It is easy to see that this method belongs to local iteration method with the degree of parallelism. Note that we can show the convergence order of the given method is bounded and equals to two.

§2. Description of parallel algorithm. The convergence of the method.

For the description of the parallel method of r-sections lets consider the case when the computational system has three processors $(r=3)$. Then from (1.2) and (1.3) we have

$$x_{k+1,1} = x_{k,1} - \frac{x_{k,1} - x_{k,2}}{f(x_{k,1}) - f(x_{k,2})} f(x_{k,1}), \tag{2.1}$$

$$x_{k+1,2} = x_{k,2} - \frac{x_{k,2} - x_{k,3}}{f(x_{k,2}) - f(x_{k,3})} f(x_{k,2}), \tag{2.2}$$

$$x_{k+1,3} = x_{k,3} - \frac{x_{k,3} - x_{k,1}}{f(x_{k,3}) - f(x_{k,1})} f(x_{k,3}). \tag{2.3}$$

Now we describe the structure of control of parallel algorithm in order to calculate formulas (2.1) and (2.3) by means of controlling expressions. The controlling expressions are the expressions on a special language, describing the structure of control of parallel program [1].

Using the control operators, we can give the structure of parallel program in the following form:

begin

process $p;q;r;q_1;r_1;q_2$ end

path $r;q;r_1;q_1;p;q_2$ end

$p: Y_k := f(x_{k,1})$,

$q: x_{k+1,1} := x_{k,1} - \dfrac{x_{k,1} - x_{k,2}}{f(x_{k,2}) - z_k} f(x_{k,1})$,

$r: z_k := f(x_{k,2})$,

$q_1: x_{k+1,2} := x_{k,2} - \dfrac{x_{k,2} - x_{k,3}}{f(x_{k,2}) - z1_k} f(x_{k,2})$,

$r_1: z1_k := f(x_{k,3})$,

$q_2: x_{k+1,3} := x_{k,3} - \dfrac{x_{k,3} - x_{k,1}}{z1_k - Y_k} Y_k$

end.

Since the calculation of the values $f(x)$ and $\dfrac{x_{k,j} - x_{k,j+1}}{f(x_{k,j}) - f(x_{k,j+1})}$ can be divided then there are two interactive parallel process. The first part of the program contains all control part given in the form of controlling expressions, the second part is the description of operators denoted by names as by labels.

If the number of processors equals to r then the structure of the control of the parallel program is determined by analogous way.

Now let's consider the convergence of the method. Let ξ be exact and $\bar{x}$-approximated roots of the equation $f(x)=0$, being on the same interval $[a,b]$. Let's substitute equation $f(x)=0$ by the equivalent equation $x=\varphi(x)$, where $\varphi(x)$ is defined and differentiated on the interval $[a,b]$, moreover all its values $\varphi(x)\in[a,b]$, $|\varphi'(x)|\le q<1$. Under these computations the iterations (16'') from [2, p.139] for every processor then we shall get

$$|\xi - x_{k,i}| \le \frac{q}{1-q}|x_{k,i} - x_{k-1,i}|.$$

Here i is the number of the processor, k is the number of iteration. If $|q|<1$, then $|x_{k,i} - x_{k-1,i}| \to 0$, when $k \to \infty$. So at $n \ge N$ we have

$$|x_{k,i} - x_{k-1,i}| < \varepsilon \quad \text{and} \quad |\xi - x_{k,i}| < \frac{q}{1-q}\varepsilon.$$

From which in particular it follows that if

$$q=\frac{1}{2}, \text{ then } |\xi - x_{k,i}| < \varepsilon.$$

So, we can say if for the i-th processor two sequential approximations $x_{k-1,i}$ and $x_{k,i}$ coincide with the given exactness ε then we can stop iteration process and $x_{k,i}$ is approximated solution of the problem.

§3. Acceleration of parallel algorithm.

The acceleration of parallel algorithm is the value showing how fast can be solved the given problem while using the parallel algorithm in comparison with the sequential realization of algorithm.

Let N be a number of initial problem and $T_r(N)$ is a minimal step of parallel algorithm (1.2)-(1.3) on the parallel IBM with the number of processor $r>1$.

Let $T_1(N)$ be a minimal step of sequential best algorithm. Then

$$S_r(N)=\frac{T_1(N)}{T_r(N)} \tag{3.1}$$

is called the acceleration of parallel algorithm. Here we assume that at the estimation of the acceleration in r-processor they all work with the same velocity.

Let's consider the parallel algorithm for the iteration (1.2)-(1.3).

It is known that (see [1]) for parallel computation of a result of the problem with N input data by means of r processors it is required not more than $m(r,N)$ steps where

$$m(r,N)=\min(p,n)+\max\left(0,\left[\left(N-2^p\right)/r\right]\right), \qquad p=[\log_2 r],\ n=[\log_2 N]. \tag{3.2}$$

The next estimation is true.

Theorem. *If* $r<N$ *then*

$$S_r(N)=\frac{rT_1(N)}{r\log_2 r+(N-r)},$$

if $r=N$, *then*

$$S_r(N)=\frac{T_1(N)}{\log_2 r},$$

if $r > N$, *then*

$$S_r(N)=\frac{T_1(N)}{\log_2 N}.$$

Proof. By means of (3.1) and (3.2) we can get:

$$S_r(N)=\frac{T_1(N)}{\min(\log_2 r,\log_2 N)+\max\left[0,\frac{N-r}{r}\right]} \tag{3.3}$$

If $r < N$, then $\min(\log_2 r,\log_2 N)=\log_2 r$,

$$\max\left[0,\frac{N-r}{r}\right]=\frac{N-r}{r}.$$

If $r = N$, then $\min(\log_2 r,\log_2 N)=\log_2 r$

$$\max\left[0,\frac{N-r}{r}\right]=0.$$

If $r > N$, then $\min(\log_2 r,\log_2 N)=\log_2 N$

$$\max\left[0,\frac{N-r}{r}\right]=0.$$

Taking into account these expressions in (3.3) we can get:

$$S_r(N)=\begin{cases}\dfrac{rT_1(N)}{r\log_2 r+(N-r)}, & if \quad r<N\\ \dfrac{T_1(N)}{\log_2 r}, & if \quad r=N,\\ \dfrac{T_1(N)}{\log_2 r}, & if \quad r>N.\end{cases}$$

Hence we see that the attained pay off in time is proportional to $\log_2 r$, i.e. one can accelerate this problem on the quality $\log_2 r$.

References

[1]. Под ред. Ершова А.П. *Алгоритмы, математическое обеспечение и архитектура многопроцессорных вычислительных систем.* М.: Наука, 1982, 336 с.

[2]. Демидович Б.П., Марон А.И. *Основы вычислительной математики.* М.: Наука, 1970, 664 с.

Kazimov J.K.
Baku State University.
23, Z.I.Khalilov str., 370148, Baku, Azerbaijan.

Received November 12, 2000; Revised February 18, 2001.
Translated by Mamedova V.A.