Miron Pavluš; Igor Podlubny

Parallel method for initial value problems for linear ordinary differential equations: speed up estimation

Kybernetika, Vol. 33 (1997), No. 3, 279--290

Persistent URL: `http://dml.cz/dmlcz/124713`

KYBERNETIKA — VOLUME 33 (1997), NUMBER 3, PAGES 279-290

PARALLEL METHOD FOR INITIAL VALUE PROBLEMS FOR LINEAR ORDINARY DIFFERENTIAL EQUATIONS: SPEED UP ESTIMATION

MIRON PAVLUŠ AND IGOR PODLUBNY

Speed up of a new parallel algorithm for solving initial value problems for linear ODEs on large parallel MIMD computers is determined. Arithmetical opreations are considered and a time for information interchange is neglected. The determined speed up is evaluated as a ratio between the number of arithmetical operations needed for serial algorithm and the number of arithmetical operations needed for parallel algorithm on large parallel MIMD computers. The used numerical method for solving initial value problems of linear ODEs is the Runge–Kutta method. An optimal number of subintervals (or processors) and an optimal number of equidistant points for one processor are determined if a total interval is subdivided into N equal parts. It is proved the speed up is proportional to $N^{1/2}$. An elementary example, illustrating the idea of the speed up estimation, is given.

1. INTRODUCTION

This paper is a continuation of a research started by Podlubny [3]. In addition to the references mentioned there, we would like to give here a brief additional survey of related approaches, some of which are not widely known.

In the past, several authors paid specific attention to the parallel solution of ordinary differential equations. In 1964, Nievergelt [2] suggested the so-called branch method. In 1967, Miranker and Liniger [1] developed a certain class of numerical integration formulas of linear multi-step type. In 1976, Worland [7] studied so-called block implicit methods. All the mentioned methods are approximate methods and can be used on MIMD machines for the approximate solution of the problem $y' = f(x, y)$, $y(x_0) = y_0$ to speed-up the process of computation.

Recently Podlubny [3] suggested a parallel algorithm for solving linear ODE and their systems with non-constant coefficients; the method is based on the solutions of the so-called local problems. Among other methods, the Nievergelt's approach is the closest to Podlubny's method. The common feature is that in both methods the entire interval $[a, b]$, in which we look for the solution of the global problem, is divided into subintervals, and certain auxillary (local) problems must be solved in those subintervals.

Podlubny's approach even allows construction of an exact closed-form analytical solution of the global problem, if one knows exact closed-form solutions of local problems. However, Nievergelt's approach allows only approximate numerical solution of the global problem. The linearity of the problem allowed Podlubny to obtain analytical solution of a global problem as a linear combination of solutions of the local problems. On the other hand, it is always possible to use any known numerical method for the solution of the local problems. Two independent approaches developed in [2] and [3] shows wide possibilities for the potential development of similar ideas in the future. A modification of Podlubny's method for linear systems of ODEs with constant coefficients was given by Török [6].

One of the most important characteristics of each parallel algorithm — the speed up — was not investigated in the papers [3] and [6]. The first estimate for Podlubny's algorithm speed up for a particular case (a sample problem $y' - y = x$, $y(x_0) = y_0$) was given by Podlubny, Pavluš and Purcz [4] under the assumption that the Runge–Kutta method was used for the solution of local problems. The estimates of speed up of Podlubny's algorithm for other seven known numerical methods for the same sample problem were derived by Purcz [5].

This paper is devoted to the further study of the speed up of Podlubny's algorithm for solving an initial-value problem for a general linear ordinary differential equation of the mth order with non-constant coefficients and an arbitrary right-hand side.

2. THE STUDIED METHOD

Let us briefly remind the parallel algorithm for solving Cauchy's problem for a non-homogeneous linear ordinary differential equations (ODEs) which has been described in detail in [3].

Let us consider the following Cauchy problem:

$$L_m[y] = f(x), \quad x \in [a,b] \tag{1}$$

$$y^{(j-1)}(a) = y_{j-1}, \quad (j = 1, \dots, m) \tag{2}$$

where $L_m[y] = \sum_{k=0}^{m} p_k(x) y^{(k)}(x)$; $p_k(x)$ $(k = 0, \dots, m)$ and $f(x)$ are continuous in the closed interval $[a,b]$. We look for the solution of the problem (1)–(2) in the interval $[a,b]$.

Let us divide this interval in n subintervals:

$$a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b$$

and look for the solution of the problem (1)–(2) in the form:

$$y(x) = \begin{cases} \sum_{k=1}^{m} a_{ik} u_{ik}(x) + v_i(x), & x \in I_i, \\ I_i = [x_{i-1}, x_i], & (i = 1, \dots, n) \end{cases} \tag{3}$$

The function $u_{ik}(x)$ is equal to 0 outside I_i and it is equal to the solution of the problem

$$\begin{cases} L_m[u_{ik}] = 0, & x \in I_i \\ u_{ik}^{(j-1)}(x_{i-1}) = \delta_{j-1,k-1}, & j = 1, \dots, m, \quad (k = 1, \dots, m) \end{cases} \tag{4}$$

when $x \in I_i$ ($\delta_{j,k}$ is Kronecker's delta).

Similarly, $v_i(x)$ is equal to 0 outside I_i and it is equal to the solution of the problem

$$\begin{cases} L_m[v_i] = f(x), & x \in I_i \\ v_i^{(j-1)}(x_{i-1}) = 0, & j = 1, \dots, m, \end{cases} \tag{5}$$

when $x \in I_i$.

The constant a_{ik} must be determined by satisfying the initial conditions (2) and the condition of continuity of $y^{(j-1)}(x), (j = 1, \dots, m)$ in $[a, b]$. The second condition leads to satisfying continuity of $y^{(j)}(x)$ at the points $x = x_i$, $(i = 1, \dots, n-1)$.

The above considerations along with the use of Picard's theorem lead to the following result:

Theorem 1. If $p_\nu(x)$, $(\nu = 0, \dots, m)$ and $f(x)$ are continuous in $[a, b]$, then the solution of the Cauchy problem (1) - (2) for the interval $[a, b]$ can be obtained in the form (3), where functions $u_{ik}(x)$ and $v_i(x)$ are defined above and the coefficients a_{ik} are given by the recurrence relations:

$$a_{1k} = y_{k-1},$$

$$a_{ik} = \sum_{j=1}^{m} a_{i-1,j} u_{i-1,j}^{(k-1)}(x_{i-1}) + v_{i-1}^{(k-1)}(x_{i-1}), \quad (i = 2, \dots, n; \quad k = 1, \dots, m) \tag{6}$$

These are main results given in the article [3] regarding to the new parallel algorithm for initial value problems for linear ODEs on large parallel MIMD computers.

In this paper we determine the speed up for the Cauchy problem (1) - (2) if the Runge–Kutta (RK) method is applied for the solution. We neglect time of information interchange between processors. The idea of the speed up estimation is illustrated by the following simple example.

3. EXAMPLE

Let us consider the problem

$$\begin{cases} y'(x) + p_0(x)y(x) = f(x) \\ y(a) = y_a \end{cases} \tag{7}$$

to find its numerical solution in the interval $[a, b]$.

We divide the interval $[a, b]$ into $N = nq$ equal parts by equidistant points

$$a = x_0 < x_1 < \ldots < x_{1.q} < \ldots < x_{2q} < \ldots < x_{nq-1} < x_{nq} = b$$

with the step $h = (b-a)/(nq)$. We have $m = 1$ (the order of the equation) for the problem (7).

According to (4), (5) we have two following problems

$$\begin{cases} u_{i1}'(x) + p_0(x)u_{i1}(x) = 0 \\ u_{i1}(x_{(i-1)q}) = 1 \end{cases} \tag{8}$$

$$\begin{cases} v_i'(x) + p_(x)v_i(x) = f(x) \\ v_i(x_{(i-1)q}) = 0 \end{cases} \tag{9}$$

for each interval $I_i = [x_{(i-1)q}, x_{iq}]$, $i = 1, \dots, n$. Problems (8) and (9) can be solved numerically by the well-known RK method:

$$z_{j+1} = z_j + \frac{h}{6}[K_1^{(j)} + 2(K_2^{(j)} + K_3^{(j)}) + K_4^{(j)}]$$

where

$$K_1^{(j)} = g(x_j, z_j)$$

$$K_2^{(j)} = g(x_j + \frac{h}{2}, z_j + \frac{h}{2}K_1^{(j)})$$

$$K_3^{(j)} = g(x_j + \frac{h}{2}, z_j + \frac{h}{2}K_2^{(j)})$$

$$K_4^{(j)} = g(x_{j+1}, z_j + h.K_3^{(j)}) \qquad (i-1)q \le j < iq.$$

For problem (8) we suppose $g(x,z) = -p_0(x)\,z(x)$ and $z_{(i-1)q} = 1$. For problem (9) we put $g(x,z) = f(x) - p_0(x)\,z(x)$ and $z_{(i-1)q} = 0$. Let us suppose the constants $\frac{h}{2}, \frac{h}{6}$ in RK method have been computed and stored in advance. Let r_0, r_f are integer non-negative numbers of arithmetical operations needed for computation of some single values of functions $p_0(x)$, $f(x)$. Now it is easy to calculate the number of arithmetical operations for one RK step. Results are given in the following table

problem	(8)	(9)
total number of operations	$4(r_0+2)+13$	$4(r_0+r_f+2)+13$

For example, according to (8), for the Cauchy problem $y'(x) - y(x) = x$, $y(a) = y_a$ we have $g(x,z) = -(-z)$ and $r_0 = 0$ i.e. $r_0 + 2 = 3$. According to (9), for the same problem we have $g(x,z) = -(-z) + x$ and $r_0 = 1$, $r_f = 0$ i.e. $r_0 + r_f + 2 = 3$. However, the same problem can be considered with $g(x,z) = z$ or $g(x,z) = z + x$ as it has been done in [4], where expressions $r_0 + 2$ and $r_0 + r_f + 2$ took the values of 0 and 1 respectively.

According to the table, $2(4r_0 + 2r_f + 21)\,q$ arithmetical operations are necessary for solving both (8) and (9) problems in the interval $I_i = [x_{(i-1)q}, x_{iq}]$, $i = 1, \dots, n$. If we suppose to use n processors for the simultaneous solution in n intervals I_i $i = 1, \dots, n$, then we have in total only $2(4r_0 + 2r_f + 21)\,q$ arithmetical operations for problems (8) and (9). To form the solution of the problem (7) we have to determine coefficients a_{ik} in expression (3), which are given by recurrence relations (6). In our particular example recurrence relations (6) take on the following form:

$$a_{11} = y_a$$
$$a_{i1} = a_{i-1,1}u_{i-1,1}(x_{(i-1)q}) + v_{i-1}(x_{(i-1)q}) \qquad i = 2, \dots, n \tag{10}$$

where $u_{i-1,1}(x_{(i-1)q})$, $v_{i-1}(x_{(i-1)q})$ must be replaced by the values of RK solutions of problems (8) and (9) in the interval $I_{i-1} = [x_{(i-2)q}, x_{(i-1)q}]$. We need $n-1$ multiplications and $n-1$ additions for the computations using the recurrence relation (10). In our case solution (3)

$$y(x) = a_{i1}u_{i1}(x) + v_i(x) \qquad x \in I_i = [x_{(i-1)q}, x_{iq}], \quad i = 1, \dots, n$$

requires $2q$ arithmetical operations.

Therefore, the problem (7) requires $2[2(2r_0 + r_f + 11)q + n - 1]$ arithmetical operations if n processors are used for the simultaneous solution in n subintervals I_i $i = 1, \dots, n$ and RK method is applied on each subinterval.

Naturally, the problem (7) can be solved on the entire interval $[a, b]$ by the RK method using step h, $g(x, z) = f(x) - p_0(x)\,z(x)$ and $z_0 = y_a$. The RK method is analogical to the problem (9) as to the one step of arithmetical operations and its characteristic numbers are given in the last column of the table above. Therefore, we can conclude that the total number of arithmetical operations for the problem (7) with serial RK method in the interval $[a, b]$ is

$$[4(r_0 + r_f + 2) + 13]\,rq \tag{11}$$

Finally, the ratio of the number of operations (11) for the serial RK method in the entire interval $[a, b]$ and the number of arithmetical operations $2[2(2r_0 + r_f + 11)\,q + n - 1]$ for suggested parallel version of the RK method in n intervals I_i, $i = 1, \dots, n$, gives the function

$$\tilde{F}(n, q) = F(N, n) = \frac{1}{2}\,\frac{[4(r_0 + r_f + 2) + 13]\,nq}{2(2r_0 + r_f + 11)\,q + n - 1} \tag{12}$$

characterizing the method's speed-up for problem (7). Under the obvious constraint $nq = N$, the function (12) takes on its maximum value at the point

$$(n^*, q^*) = ((NB)^{1/2}, (N/B)^{1/2})$$

where $B = 2(r_0 + r_f + 11)$. For large N the maximum value at this point is approximatelly equal

$$\tilde{F}(n^*, q^*) \approx \frac{A}{2(B)^{1/2}} N^{1/2} \tag{13}$$

where $A = [4(r_0 + r_f + 2) + 13]/2$. The proof of these statements is analogical to the proof of the Theorem 2 given below, and we omit the proof here.

To make these estimates more clear, let us consider the case of $r_0 = r_f = 1$, which means that only one arithmetical orepation is necessary for the computation of each function of $p_0(x)$ and $f(x)$.

The shape of the speed up function $F(N,n)$ is shown in Figure 1, in which the "square-root law" dependence of the maximal speed up given by relationship (13) is also clearly observed. Another view on the speed up is presented in Figure 2, in which each particular plot corresponds to a certain value of N.

In both figures we neglected the discrete nature of N and n to make the general enhancement more clear.

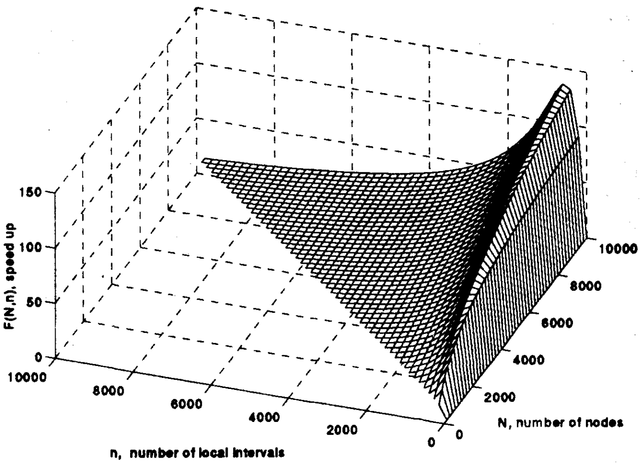

Fig. 1. Shape of $F(N,n)$.

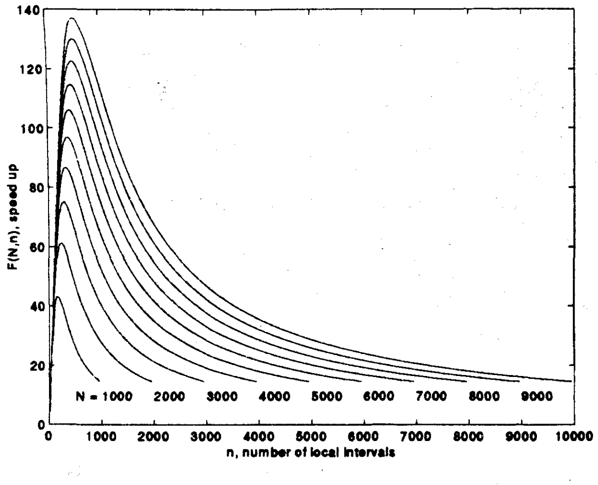

Fig. 2. Speed-up for various N as a function of n.

4. SPEED UP ESTIMATION IN A GENERAL CASE

Let us consider problem (1) – (2) for a general natural m. Let us introduce the following substitutions

$$w_{m-1} = y^{m-1}(x), \ldots, w_1(x) = y'(x), w_0(x) = y(x)$$

Then problem (1) – (2) can be written in the matrix form

$$W'(x) = P(x)\,W(x) + F(x) \tag{14}$$

$$W(a) = Y_0 \tag{15}$$

where

$$P(x) = \begin{pmatrix} -p_{m-1}(x) & \cdots & -p_1(x) & -p_0(x) \\ 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 1 & 0 \end{pmatrix}$$

$$W(x) = \begin{pmatrix} w_{m-1}(x) \\ w_{m-2}(x) \\ \vdots \\ w_0(x) \end{pmatrix}; \quad F(x) = \begin{pmatrix} f(x) \\ 0 \\ \vdots \\ 0 \end{pmatrix}; \quad Y_0 = \begin{pmatrix} y_{m-1} \\ y_{m-2} \\ \vdots \\ y_0 \end{pmatrix}.$$

Let us solve the Cauchy problem (14) – (15) numerically by the RK method:

$$\mathcal{Z}_{j+1} = \mathcal{Z}_j + \frac{h}{6}[\mathcal{K}_1^{(j)} + 2(\mathcal{K}_2^{(j)} + \mathcal{K}_3^{(j)}) + \mathcal{K}_4^{(j)}]$$

where

$$\begin{aligned} \mathcal{K}_1^{(j)} &= \mathcal{G}(x_j, \mathcal{Z}_j) \\ \mathcal{K}_2^{(j)} &= \mathcal{G}(x_j + \frac{h}{2}, \mathcal{Z}_j + \frac{h}{2}\mathcal{K}_1^{(j)}) \\ \mathcal{K}_3^{(j)} &= \mathcal{G}(x_j + \frac{h}{2}, \mathcal{Z}_j + \frac{h}{2}\mathcal{K}_2^{(j)}) \\ \mathcal{K}_4^{(j)} &= \mathcal{G}(x_{j+1}, \mathcal{Z}_j + h\mathcal{K}_3^{(j)}) \end{aligned}$$

$\mathcal{Z}_{j+1}$, $\mathcal{Z}_j$, $\mathcal{K}_1^{(j)}$, $\mathcal{K}_2^{(j)}$, $\mathcal{K}_3^{(j)}$, and $\mathcal{K}_4^{(j)}$ are vectors of m components. Let us suppose the constants $\frac{h}{2}$, $\frac{h}{6}$ in RK method have been computed and stored in advance. We consider the addition $x_j + \frac{h}{2}$ as one arithmetical operation in one vector RK step.

Analizing one step of the vector RK method we can count $1 + 12m$ arithmetical operations if we exclude arithmetical operations needed for vector evaluation of the vector function $\mathcal{G}(x_j, \mathcal{Z}_j)$. One vector value of the function $\mathcal{G}(x_j, \mathcal{Z}_j) = P(x_j)\mathcal{Z}_j + F(x_j)$ requires

$$2m + r_f + \sum_{k=0}^{m-1} r_k$$

of arithmetical operations. This means that the total number of arithmetical operations for one RK step is

$$1+4\left(5m+r_f+\sum_{k=0}^{m-1} r_k\right). \tag{16}$$

After nq steps we have

$$\left[1+4\left(5m+r_f+\sum_{k=0}^{m-1} r_k\right)\right] nq \tag{17}$$

arithmetical operations. This is the number of operations needed for the serial RK method in the entire interval $[a,b]$. Note, if $m=1$ then the expression (17) is just the same like we have had for (11). m local problems (4) can be after substitutions

$$w_{m-1}(x)=u_{ik}^{m-1}(x),\dots,w_1(x)=u_{ik}'(x),w_0(x)=u_{ik}(x)$$

rewritten as

$$W'(x)=P(x)\,W(x) \tag{18}$$

$$W(x_{(i-1)q})=e_k \tag{19}$$

where $k=1,\dots,m$ and

$$e_k=\begin{pmatrix}0\\0\\\vdots\\1\\\vdots\\0\end{pmatrix}\leftarrow(k)\,.$$

Problem (5) after substitutions

$$w_{m-1}(x)=v_i^{m-1}(x),\dots,w_1(x)=v_i'(x),w_0(x)=v_i(x)$$

can be rewritten as

$$W'(x)=P(x)\,W(x)+F(x) \tag{20}$$

$$W(x_{(i-1)q})=0 \tag{21}$$

If for m problems in (18)–(19) we suppose $\mathcal{G}(x_j,\mathcal{Z}_j)=P(x_j)\mathcal{Z}_j$, $\mathcal{Z}_{(i-1)q}=e_k$, $k=1,\dots,m$ then the number of arithmetical operations after q vector RK steps is equal to

$$qm\left[1+4\left(5m+\sum_{k=0}^{m-1} r_k\right)\right].$$

For problem (20)–(21) we put $\mathcal{G}(x_j,\mathcal{Z}_j)=P(x_j)\mathcal{Z}_j+F(x_j)$, $\mathcal{Z}_{(i-1)q}=0$, and according to (16) we receive

$$q\left[1+4\left(5m+r_f+\sum_{k=0}^{m-1} r_k\right)\right]$$

for q vector RK steps in a general interval $I_i = [x_{(i-1)q}, x_{iq}]$. Summation of two last expresions give us the total number of arithmetical operations

$$q\left[1+4r_f+4\sum_{k=0}^{m-1} r_k + m\left(21+4\sum_{k=0}^{m-1} r_k\right)+20m^2\right] \tag{22}$$

needed for both (18) - (19) and (20) - (21) problems. Note, if $m = 1$, then this general expression give us $2q(21 + 2r_f + 4r_0)$ arithmetical operations — the same number we had in the previous example.

According to Theorem 1 we need $m(m+1)(n-1)$ additional arithmetical operations for the computation of the coefficients a_{ik} and for obtaining the solution according to (3) we need $(m+1)\,q$ arithmetical operations in the interval $I_i = [x_{(i-1)q}, x_{iq}]$. Again if $m = 1$ obtaining the results coincide the results of the previous example.

The total number of arithmetical operations needed for parallel performance in n subintervals $I_i = [x_{(i-1)q}, x_{iq}]$, $i = 1, \dots, n$ (or on n processors) of the problem (1) - (2) is

$$q\left[2+4r_f+4\sum_{k=0}^{m-1} r_k + m\left(22+4\sum_{k=0}^{m-1} r_k\right)+20m^2\right]+(n-1)(m+1)\,m. \tag{23}$$

Now we can define the speed up function

$$\mathcal{F}(m,n,q) = \frac{nqA(m)}{B(m)\,q+n-1} \tag{24}$$

as a ratio of the number of arithmetical operations needed for serial (17) and parallel (23) version of the RK method for problem (1) - (2). We denote

$$A(m) = \left[1+4\left(5m+r_f+\sum_{k=0}^{m-1} r_k\right)\right] \Big/ \, ((m+1)\,m) \tag{25}$$

$$B(m) = \left[2+4r_f+4\sum_{k=0}^{m-1} r_k + m\left(22+4\sum_{k=0}^{m-1} r_k\right)+20m^2\right] \Big/ \, ((m+1)\,m). \tag{26}$$

Theorem 2. If the interval $[a, b]$ is divided into $N = nq$ equal parts then for parallel performance of the problem (1) - (2) by RK method an optimal number of intervals $I_i = [x_{(i-1)q}, x_{iq}]$, $i = 1, \dots, n$ (or processors) is equal to $n^* = (NB)^{1/2}$ and an optimal number of RK steps within each interval is equal to $q^* = (N/B)^{1/2}$. For large N the speed up function value in the optimal point (m, n^*, q^*) is approximately equal to

$$\mathcal{F}(m,n^*,q^*) \approx \frac{A(m)}{2(B(m))^{1/2}} N^{1/2}$$

where $A(m)$, $B(m)$ are given by relations (25) - (26).

Proof. We use the Lagrange multipliers method for the optimal point determination. Under constrain $N = nq$ we introduce the Lagrange function

$$\Phi(m,n,q) = \frac{nqA(m)}{B(m)\,q+n-1} + \lambda(nq-N)$$

where $A(m)$, $B(m)$ are given by relations (25) – (26) and λ is the Lagrange multiplier. If we put the first-order derivatives

$$\frac{\partial\Phi}{\partial n} = q[\lambda + A(m)\frac{B(m)\,q-1}{(B(m)\,q+n-1)^2}]$$

$$\frac{\partial\Phi}{\partial q} = n[\lambda + A(m)\frac{n-1}{(B(m)\,q+n-1)^2}]$$

to zeros and use constraint equation $N = nq$ then we have three equations for unknown n^*, q^*, λ^*. The solution is

$$n^* = (NB(m))^{1/2} \qquad q^* = (N/B(m))^{1/2} \qquad \lambda^* = A(m)\frac{1-(NB(m))^{1/2}}{(1-2(NB(m))^{1/2})^2}.$$

Using these results we come to the extremal value of the speed up function

$$\mathcal{F}(m,n^*,q^*) = \frac{NA(m)}{2(NB(m))^{1/2}-1}$$

which is close to $\frac{A(m)}{2(B(m))^{1/2}}N^{1/2}$ for large N.
Finally, we prove that the extremal point is the point of maximum value of the speed up function (24). The second derivatives are

$$\begin{aligned}
\frac{\partial^2\Phi}{\partial n^2} &= 2qA(m)\,\frac{1-qB(m)}{(qB(m)+n-1)^3}\\
\frac{\partial^2\Phi}{\partial q^2} &= 2nA(m)\,B(m)\,\frac{1-n}{(qB(m)+n-1)^3}\\
\frac{\partial^2\Phi}{\partial n\partial q} &= \lambda + A(m)\,\frac{1-n+q(2n-1)\,B(m)}{(qB(m)+n-1)^3}.
\end{aligned}$$

According to the general theory of the conditional extremum we compute the following values at the extremal point (n^*, Q^*, λ^*):

$$\begin{aligned}
a_{11} &= \frac{\partial^2\Phi}{\partial n^2} = 2A(m)(\frac{N}{B(m)})^{1/2}\frac{1-(NB(m))^{1/2}}{(2(NB(m))^{1/2}-1)^3}\\
a_{22} &= \frac{\partial^2\Phi}{\partial q^2} = 2A(m)\,B(m)(NB(m))^{1/2}\frac{1-(NB(m))^{1/2}}{(2(NB(m))^{1/2}-1)^3}\\
a_{12} &= \frac{\partial^2\Phi}{\partial q\partial n} = A(m)\frac{(NB(m))^{1/2}}{(2(NB(m))^{1/2}-1)^3}
\end{aligned}$$

and observe that $a_{11} < 0$ for all considered N, $A(m)$ and $B(m)$. Moreover, the expression

$$a_{11}a_{22} - a_{12}^2 = \frac{NB(m)A^2(m)}{(2(NB(m))^{1/2} - 1)^6}[4(1 - (NB(m))^{1/2})^2 - 1]$$

is positive for the same $N, A(m), B(m)$. This ends the proof. □

5. CONCLUDING REMARKS

The introduced speed up function (24) lead us to the following conclusions.

To achieve the speed up for a given LODE, one needs at least

$$1 + \mathcal{E}\left(\frac{q(2 + 4r_f + 4r(m) + m(22 + 4r(m)) + 20m^2) - m(m+1)}{q[1 + 4(5m + r(m) + r_f)] - m(m+1)}\right)$$

processors for the realization of the considered parallel algorithm. We denote here $\mathcal{E}(x)$ the integer part of x and

$$r(m) = \sum_{k=0}^{m-1} r_k.$$

I' one has a parallel computer system with n processors, then the considered method provides acceleration only if the order m of the LODE is greater than the largest root of the equation

$$4mr(m) + 4r(m)(1-n) + 20m^2 + m\left[22 - 20n + \frac{(n-1)}{q}\right]$$
$$+ (1 + 4r_f)(1-n) + 1 - \frac{(n-1)\,m^2}{q} = 0.$$

(Received November 3, 1995.)

REFERENCES

[1] W. L. Miranker and W. Liniger: Parallel methods for the numerical integration of ordinary differential equations. Math. Com. *21* (1967), 99, 303–320.

[2] J. Nievergelt: Parallel methods for integrating ordinary differential equations. Comm. ACM *7* (1964), 12, 731–733.

[3] I. Podlubny: Parallel algorithms for initial and boundary value problems for linear ordinary differential equations and their systems. Kybernetika *32* (1996), 3, 251–260.

[4] I. Podlubny, M. Pavluš and P. Purcz: Parallel solution of initial value problems for linear ordinary differential equations. In: Proceedings of International Workshop Parallel Numerics 95, Sorrento 1995, pp. 171–182

[5] P. Purcz: The speed up of a parallel algorithm for a Cauchy problem and for various known numerical methods. In: Proceedings of the International Workshop Parallel Numerics'96, Gozd Matuljek 1996, pp. 137–179.

[6] Cs. Török: On the parallel algorithms of initial value problems for systems of linear ordinary differential equations. In: Proceedings of the 113th Panonian Applied Mathematical Meeting, Bardejovske Kupele 1995, pp. 231–237.

[7] P. B. Worland: Parallel methods for the numerical solution of ordinary differential equations. IEEE Trans. Comput. *25* (1976), 1045–1048.

RNDr. Miron Pavluš, CSc., Department of Mathematics, Faculty of Civil Engineering, Technical University, Vysokoškolská 4, 042 00 Košice. Slovakia.

Doc. RNDr. Igor Podlubny, CSc., Department of Management and Control Engineering, B.E.R.G. Faculty, Technical University, B. Němcovej 3, 042 00 Košice. Slovakia.