

On Implementation of Vector Gauss Method for Solving Large-Scale Systems of Index 1 Differential-Algebraic Equations

Gennady Yu. Kulikov¹ and Galina Ya. Benderskaya¹

Ulyanovsk State University, L. Tolstoy Str. 42, 432700 Ulyanovsk, Russia

Abstract. In the paper we further develop the idea of parallel factorization of nonzero blocks of sparse coefficient matrices of the linear systems arising from discretization of large-scale index 1 differential-algebraic problems by Runge-Kutta methods and their following solving by Newton-type iterations. We formulate a number of theorems that give estimates for the local fill-in of such matrices on some stages of Gaussian elimination. As the result, we derive that only the suggested modification of Gauss method appeared to be effective and economical one from the standpoint of CPU time and RAM.

1 Introduction

One of the main trends of modern computational mathematics is the development of effective numerical methods and software packages for the solution of index 1 differential-algebraic systems of the form (see, for example, [1], [2], [4]):

$$x'(t) = g(x(t), y(t)), \quad (1a)$$

$$y(t) = f(x(t), y(t)), \quad (1b)$$

$$x(0) = x^0, \quad y(0) = y^0, \quad (1c)$$

where $t \in [0, T]$, $x(t) \in \mathbf{R}^m$, $y(t) \in \mathbf{R}^n$, $g : D \subset \mathbf{R}^{m+n} \rightarrow \mathbf{R}^m$, $f : D \subset \mathbf{R}^{m+n} \rightarrow \mathbf{R}^n$, and the initial conditions (1c) are consistent; i. e., $y^0 = f(x^0, y^0)$. Note that we consider only autonomous systems because any nonautonomous system may be converted to an autonomous one by introducing a new independent variable.

Let problem (1) satisfy the smoothness, nonsingularity and inclusion conditions introduced in [6]. Then, it has a unique solution $z(t) \stackrel{\text{def}}{=} (x(t)^T, y(t)^T)^T \in \mathbf{R}^{m+n}$ (see [6]). Besides, according to the Gear's definition in [3] problem (1) is of index 1.

To solve problem (1) numerically, we apply an l -stage implicit Runge-Kutta (RK) method given by the Butcher's table

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array},$$

* This work was supported in part by the Russian Foundation of the Basic Research (grants No. 01-01-00066 and No. 00-01-00197).

where A is a real matrix of dimension $l \times l$, b and c are real vectors of dimension l , to problem (1) and obtain the following discrete analogue:

$$x_{ki} = x_k + \tau \sum_{j=1}^l a_{ij} g(x_{kj}, y_{kj}), \quad (2a)$$

$$y_{ki} = f(x_{ki}, y_{ki}), \quad i = 1, 2, \dots, l, \quad (2b)$$

$$x_{k+1} = x_k + \tau \sum_{i=1}^l b_i g(x_{ki}, y_{ki}), \quad (2c)$$

$$y_{k+1} = f(x_{k+1}, y_{k+1}), \quad k = 0, 1, \dots, K - 1. \quad (2d)$$

Here $x_0 = x^0$, $y_0 = y^0$, and τ is a stepsize which may be variable. Algebraic system (2) is solved then by an iterative method at each time point t_k . Usually, the iterative process is taken in the form of simple or Newton-type iterations with trivial (or nontrivial) predictor [1], [2], [4], [7], [8], [10], [14].

In [11] it has been substantiated why Newton (or modified Newton) iteration is more preferable for solving differential-algebraic equations (1) than simple one. By this reason, we further consider only Newton-type iterations in the paper.

In order to solve problem (2) it is not necessary to apply an iterative process to all the equations of the discrete problem simultaneously. It is evident that equations (2a,b) do not depend on equations (2c,d). From this standpoint, it makes sense first to solve problem (2a,b) that contains $(m+n)l$ equations. Then, we have to find the solution of problem (2c) which will be a known function of the solution of (2a,b). Finally, we apply an iterations to system (2d) of dimension n . The advantage of this approach for the full and modified Newton methods was investigated in [7]. Moreover, when solving the larger problem (2a,b) we can limit ourselves by less quantity of iterations at the grid points on the segment $[0, T]$ (for details, see [12]).

Thus, from the above approach it follows necessity to solve discrete systems (2a,b) of dimension $(m+n)l$ many times during the integration. The latter allows us to conclude that Newton-type iterations put severe requirements on RAM and CPU time caused by the increasing dimension of discrete problem (2) in l times when an l -stage implicit RK method has been used. Therefore the basic problem is how to simplify and speed up the numerical solving of the linear systems with sparse coefficient matrices of the special form arising from the application of Newton method (full or modified) to problem (2a,b).

In [11], [13] we gave rise to the idea of modification of Gaussian elimination for parallel factorization of nonzero blocks of the matrices. In the present paper we first introduce a new term to denote this modification and it will be referred to further as *vector Gauss method*. This modification allows RAM and CPU time to be significantly reduced in the numerical integration of problem (1) by an implicit RK method. Now we discuss full details associated with pivoting in the course of implementation of the vector Gauss method. In addition, we give a simple example which illustrate theoretical results of the paper.

2 Preliminary Notes for Vector Gauss Method

As we have derived above, the basic part of Newton iteration applied to problem (2) consists of solving linear systems of the form

$$\partial \bar{F}_k^\tau(Z_{k+1}^{i-1})(Z_{k+1}^{i-1} - Z_{k+1}^i) = \bar{F}_k^\tau Z_{k+1}^{i-1} \tag{3}$$

where lower indices mean time points, and upper ones denote iterations. Here $Z_{k+1} \stackrel{\text{def}}{=} ((z_{k1})^T, \dots, (z_{k,l-1})^T, (z_{kl})^T)^T \in \mathbf{R}^{(m+n)l}$, where the vector $z_{kj} \stackrel{\text{def}}{=} ((x_{kj})^T, (y_{kj})^T)^T \in \mathbf{R}^{m+n}$, $j = 1, 2, \dots, l$, unites components of the j -th stage value of l -stage RK formula (2). The mapping \bar{F}_k^τ is the nontrivial part of discrete problem (2) for computing the stage values Z_{k+1} , and $\partial \bar{F}_k^\tau(Z_{k+1}^{i-1})$ denotes the Jacobian of the mapping \bar{F}_k^τ evaluated at the point Z_{k+1}^{i-1} .

From system (2a,b) it follows that the matrix $\partial \bar{F}_k^\tau(Z_{k+1}^{i-1})$ has the following block structure:

$$\partial \bar{F}_k^\tau(Z) \stackrel{\text{def}}{=} \begin{pmatrix} \partial \bar{F}_k^\tau(Z)_1 \\ \partial \bar{F}_k^\tau(Z)_2 \\ \vdots \\ \partial \bar{F}_k^\tau(Z)_l \end{pmatrix} \tag{4}$$

where each block $\partial \bar{F}_k^\tau(Z)_j, j = 1, 2, \dots, l$, is an $(m+n) \times (m+n)l$ -matrix of the form

$$\left(\begin{array}{cccccccccccc} O(\tau) & \cdots & O(\tau) & 1+O(\tau) & \cdots & O(\tau) & O(\tau) & \cdots & O(\tau) & O(\tau) & \cdots & O(\tau) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ O(\tau) & \cdots & O(\tau) & O(\tau) & \cdots & 1+O(\tau) & O(\tau) & \cdots & O(\tau) & O(\tau) & \cdots & O(\tau) \\ 0 & \cdots & 0 & z & \cdots & z & z & \cdots & z & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & z & \cdots & z & z & \cdots & z & 0 & \cdots & 0 \end{array} \right) \left. \begin{array}{l} \right\} m \\ \left. \right\} n \end{array} \right.$$

$\underbrace{\hspace{10em}}_{(m+n)(j-1)} \quad \underbrace{\hspace{10em}}_m \quad \underbrace{\hspace{10em}}_n \quad \underbrace{\hspace{10em}}_{(m+n)(l-j)}$

Here z means in general a nontrivial element.

Having used the structure of matrix (4), Kulikov and Thomsen suggested in [7] to exclude the zero blocks situated below the main diagonal from LU-factorization and forward substitution. That reduces the number of arithmetical operations and, hence, CPU time significantly when linear system (3) is solved. This approach was called *Modification I* in [9]. Moreover, the advantage of the new version of Gaussian elimination becomes greater if we have interchanged the x -components of the vector Z_{k+1} with the corresponding y -components. It means that we have interchanged the first m rows with the last n ones in each submatrix $\partial \bar{F}_k^\tau(Z)_j$. In this case we can exclude all the zero blocks of matrix (4) from the LU-factorization, forward and backward substitutions and solve problem (3) very effectively (see *Modification II* of Gauss method in [9]).

However, the discussed modifications may appear useless when we solve large-scale semi-explicit index 1 differential-algebraic systems. As an example, we may take the model of overall regulation of body fluids [5]. This model is a problem

of the form (1) containing about two hundred variables. Having applied any implicit 3- or 4-stage RK method to the model we encounter the situation when the dimension of discrete problem (2) is too high to solve it by Newton-type iterations (see the numerical example in [11]). On the other hand, the Jacobian of a large-scale differential-algebraic system is often a sparse matrix. For instance, the model mentioned above is the case. Thus, the main problem must be how to implement Modification II of Gaussian elimination for matrix (4) effectively.

As in [11], we rearrange the variables and define the vectors:

$$\begin{aligned} X_{k+1} &\stackrel{\text{def}}{=} ((x_{k1})^T, \dots, (x_{k,l-1})^T, (x_{kl})^T)^T \in \mathbf{R}^{ml}, \\ Y_{k+1} &\stackrel{\text{def}}{=} ((y_{k1})^T, \dots, (y_{k,l-1})^T, (y_{kl})^T)^T \in \mathbf{R}^{nl}. \end{aligned}$$

Now $Z_{k+1} \stackrel{\text{def}}{=} ((Y_{k+1})^T, (X_{k+1})^T)^T$ and matrix (4) has the form

$$\partial \bar{F}_k^\tau(Z) \stackrel{\text{def}}{=} \begin{pmatrix} \partial \bar{F}_k^\tau(Z)^Y \\ \partial \bar{F}_k^\tau(Z)^X \end{pmatrix}. \quad (5)$$

Here each submatrix has also the block structure:

$$\partial \bar{F}_k^\tau(Z)^Y \stackrel{\text{def}}{=} \begin{pmatrix} \partial \bar{F}_k^\tau(Z)^Y_1 \\ \partial \bar{F}_k^\tau(Z)^Y_2 \\ \vdots \\ \partial \bar{F}_k^\tau(Z)^Y_l \end{pmatrix} \quad \text{and} \quad \partial \bar{F}_k^\tau(Z)^X \stackrel{\text{def}}{=} \begin{pmatrix} \partial \bar{F}_k^\tau(Z)^X_1 \\ \partial \bar{F}_k^\tau(Z)^X_2 \\ \vdots \\ \partial \bar{F}_k^\tau(Z)^X_l \end{pmatrix}$$

where

$$\begin{aligned} \partial \bar{F}_k^\tau(Z)^Y_i &\stackrel{\text{def}}{=} \left(\begin{array}{cccccc} 0 & \cdots & 0 & z & \cdots & z & 0 & \cdots & 0 & z & \cdots & z & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & z & \cdots & z & 0 & \cdots & 0 & z & \cdots & z & 0 & \cdots & 0 \end{array} \right) \Bigg\} n \\ &\quad \underbrace{\hspace{1.5cm}}_{n(i-1)} \quad \underbrace{\hspace{1.5cm}}_n \quad \underbrace{\hspace{1.5cm}}_{n(l-i)+m(i-1)} \quad \underbrace{\hspace{1.5cm}}_m \quad \underbrace{\hspace{1.5cm}}_{m(l-i)} \\ \partial \bar{F}_k^\tau(Z)^X_i &\stackrel{\text{def}}{=} \left(\begin{array}{cccccc} O(\tau) & \cdots & O(\tau) & 1+O(\tau) & \cdots & O(\tau) & O(\tau) & \cdots & O(\tau) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ O(\tau) & \cdots & O(\tau) & O(\tau) & \cdots & 1+O(\tau) & O(\tau) & \cdots & O(\tau) \end{array} \right) \Bigg\} m \\ &\quad \underbrace{\hspace{1.5cm}}_{nl+m(i-1)} \quad \underbrace{\hspace{1.5cm}}_m \quad \underbrace{\hspace{1.5cm}}_{m(l-i)} \end{aligned}$$

Now we note that, when solving linear system (3) with matrix (5) by Modification II of Gaussian elimination, LU-factorization of any submatrix $\partial \bar{F}_k^\tau(Z)^Y_i$ does not influence the submatrices $\partial \bar{F}_k^\tau(Z)^Y_j$ for $j \neq i$. This means that the factorization of the matrix $\partial \bar{F}_k^\tau(Z)^Y$ falls into l independent LU-factorizations of the submatrices $\partial \bar{F}_k^\tau(Z)^Y_i$, $i = 1, 2, \dots, l$. Moreover, structures of the similar nonzero blocks of all the submatrices $\partial \bar{F}_k^\tau(Z)^Y_i$ (i.e., the number and the places of nonzero elements) coincide, if the stepsize τ is sufficiently small.

It is evident that packing suggested in [11], [13] allows the parallel factorization of the matrix $\partial \bar{F}_k^\tau(Z)^Y$ to be implemented in practice. To store the matrix $\partial \bar{F}_k^\tau(Z)^X$, we can use any packing appropriate for sparse matrices in a general case because the matrix A of coefficients of the RK method may have zero elements and, hence, $\partial \bar{F}_k^\tau(Z)^X$ also contains zero blocks.

3 Implementation of Vector Gauss Method

As it is well-known, each step of Gauss method is split up into two independent stages. The first stage is pivoting in the active submatrix, and at the second one we have to eliminate the next variable from the remaining equations of linear system. We start with the discussion of different ways of pivoting for linear system (3) with the sparse coefficient matrix (5).

When solving a linear system with sparse coefficient matrix by Gauss method we have to determine a set of *admissible elements* of the active submatrix; i.e., the elements which are greater or equal to some $\epsilon > 0$. As the pivot, we then take the admissible element giving the minimum (or small) *local fill-in* (i.e., quantity of zero elements of the coefficient matrix which become nonzero ones) for the next step of Gaussian elimination. There are several ways to implement this idea in practice. In the paper we consider two of them and show how this process should be optimized for linear system (3) with the packed coefficient matrix (5).

The first strategy of pivoting is based on theorem 2.5.5 in [15]. According to this theorem, if we take the $(\mu + p, \mu + q)$ -th element of matrix $\partial \bar{F}_k^T(Z)^{(\mu)}$ derived for μ steps of the Gauss method as a pivot ($p, q = 1, 2, \dots, (m + n)l - \mu$) then the local fill-in at the $\mu + 1$ -th step is given by the (p, q) -th element of matrix $G^{(\mu)}$, where

$$G^{(\mu)} \stackrel{\text{def}}{=} B^{(\mu)} \left(\bar{B}^{(\mu)} \right)^T B^{(\mu)} \quad (6)$$

and the square matrix $B^{(\mu)}$ of dimension $(m + n)l - \mu$ is built from the elements of active submatrix of $\partial \bar{F}_k^T(Z)^{(\mu)}$ by replacing nonzero elements with the unity ones, and $(\bar{B}^{(\mu)})^T$ is a transposed matrix to $\bar{B}^{(\mu)} \stackrel{\text{def}}{=} M_{(m+n)l-\mu} - B^{(\mu)}$ ($M_{(m+n)l-\mu}$ is a square matrix of dimension $(m+n)l - \mu$ with the unity elements). Then, as the pivot for the $\mu + 1$ -th step we choose that admissible element which gives the minimum local fill-in. In practice, implementation of the first strategy means calculation of the matrix $G^{(\mu)}$ before each step of Gauss method; i.e., for $\mu = 0, 1, \dots, (m + n)l - 1$. As the result, the cost of this strategy (in terms of multiplication operations) can be determined by the following formula:

$$2 \sum_{\mu=0}^{(m+n)l-1} ((m+n)l - \mu)^3 = \frac{(m+n)^2 l^2 ((m+n)l + 1)^2}{2}. \quad (7)$$

According to the second strategy, if we take the $(\mu + p, \mu + q)$ -th element of matrix $\partial \bar{F}_k^T(Z)^{(\mu)}$ as a pivot then the maximum possible local fill-in at the $\mu + 1$ -th step of Gauss method is given by the (p, q) -th element of matrix $\hat{G}^{(\mu)}$, where

$$\hat{G}^{(\mu)} \stackrel{\text{def}}{=} (B^{(\mu)} - I_{(m+n)l-\mu}) M_{(m+n)l-\mu} (B^{(\mu)} - I_{(m+n)l-\mu}) \quad (8)$$

with the matrices $B^{(\mu)}$ and $M_{(m+n)l-\mu}$ having the above sense (see theorem 2.5.14 in [15]). Then, as the pivot for the $\mu + 1$ -th step of Gauss method we choose that admissible element which gives the minimum from the maximum possible local

fill-ins. Thus, the computational cost of this strategy (in terms of multiplication operations) can be expressed as follows:

$$\sum_{\mu=0}^{(m+n)l-1} ((m+n)l - \mu)^2 = \frac{(m+n)l((m+n)l+1)(2(m+n)l+1)}{6}. \quad (9)$$

Obviously, the second strategy is not as exact as the first one since it does not allow the real local fill-in to be computed, but just gives some upper estimate. However, its practical implementation is much cheaper than that of the first strategy (compare (7) and (9)).

Let us now refer to the special structure of matrix $\partial \bar{F}_k^\tau(Z)$. It allows the factorization process to be split up into two stages. At the first stage we will eliminate y -components of the vector Z_{k+1} , and the x -components will be eliminated at the second one. Consider pivoting for the first stage of LU-factorization.

As we remember, the LU-factorization of matrix $\partial \bar{F}_k^\tau(Z)^Y$ falls into l independent LU-factorizations of the submatrices $\partial \bar{F}_k^\tau(Z)_i^Y$, $i = 1, 2, \dots, l$, if when factorizing each submatrix $\partial \bar{F}_k^\tau(Z)_i^Y$ at the $\mu+1$ -th step of Gauss method we take the pivot from the active submatrix which is the right-hand side minor located on intersection of the last $n - \mu$ rows and columns of the block $\partial_{y_{k,i}} \bar{F}_k^\tau(z_{k,i})_i^{Y(\mu)}$. By virtue of the nonsingularity condition (see [6]) we can always find at least one admissible element among the elements of the above minor if $\partial_{y_{k,i}} \bar{F}_k^\tau(z_{k,i})_i^Y$ is not an improperly stipulated matrix. So, if we choose the pivot by this way then, taking into account the structure of matrix $\partial \bar{F}_k^\tau(Z)$, the local fill-in may influence only the matrix of dimension $(n - \mu + ml) \times (n - \mu + m)$

$$\partial_{z_{k,i}} \bar{F}_k^\tau(z_{k,i})^{(\mu)} \stackrel{\text{def}}{=} \begin{pmatrix} \partial_{z_{k,i}} \bar{F}_k^\tau(z_{k,i})_i^{Y(\mu)} \\ \partial_{z_{k,i}} \bar{F}_k^\tau(z_{k,i})_1^{X(\mu)} \\ \partial_{z_{k,i}} \bar{F}_k^\tau(z_{k,i})_2^{X(\mu)} \\ \vdots \\ \partial_{z_{k,i}} \bar{F}_k^\tau(z_{k,i})_l^{X(\mu)} \end{pmatrix}. \quad (10)$$

To speed up the computation process, we suggested the simultaneous factorization of all the submatrices $\partial \bar{F}_k^\tau(Z)_i^Y$, $i = 1, 2, \dots, l$. Thus, every step of the Gauss method now implies elimination not one but l variables; i. e., all the arithmetical operations are implemented in vector form with vectors of dimension l . Moreover, in this case we have to choose *the vector of pivots* (or shortly *the pivot vector*). That is why we call this approach *the vector Gauss method*. The packing suggested in [11], [13] makes it possible to realize the vector Gauss method in practice if we additionally require all the components of pivot vectors to be on the same places in active submatrices of the blocks $\partial_{y_{k,i}} \bar{F}_k^\tau(z_{k,i})_i^{Y(\mu)}$, $i = 1, 2, \dots, l$, $\mu = 0, 1, \dots, n - 1$.

As we have discussed above, in the process of numerical solving of linear systems with sparse coefficient matrices by Gauss method it is necessary first to determine some set of admissible elements and then to choose the pivot. In the context of vector Gauss method, this approach requires determination of the set

of *admissible vectors*; i.e., the vectors with all their components being admissible elements. In this case we have to guarantee that this set will not prove to be empty for the nonsingular matrix $\partial \bar{F}_k^\tau(Z)$ (i.e., the sets of admissible elements of active submatrices of the blocks $\partial_{y_{ki}} \bar{F}_k^\tau(z_{ki})_i^{Y^{(\mu)}}$ will produce nonempty intersection for any $\mu = 0, 1, \dots, n - 1$). By virtue of the smoothness, nonsingularity and inclusion conditions the above requirement can be easily met in the process of solving system (3) with the coefficient matrix (5), if the stepsize τ is sufficiently small. The detailed explanation of this fact will appear in [13].

The next step in the procedure of pivoting for the vector Gauss method is determination of the admissible vector which gives the minimum (or sufficiently small) local fill-in. In general, this problem tends to be very hard to solve. However, the situation is simplified significantly if RK methods with *dense coefficient matrix* A (i.e., $a_{ij} \neq 0, i, j = 1, 2, \dots, l$) have been applied to problem (1). Then the blocks $\partial_{z_{ki}} \bar{F}_k^\tau(z_{ki})_i^{Y^{(\mu)}}$ and $\partial_{z_{ki}} \bar{F}_k^\tau(z_{ki})_j^{X^{(\mu)}}$, $j = 1, 2, \dots, l$, are of the same structure. Therefore matrices $\partial_{z_{ki}} \bar{F}_k^\tau(z_{ki})_i^{(\mu)}$, $i = 1, 2, \dots, l$, also have the same structure if we put the block $\partial_{z_{ki}} \bar{F}_k^\tau(z_{ki})_i^{X^{(\mu)}}$ on the last position in all the matrices (see (10)). Thus, all the components of any admissible vector lead to one and the same fill-in and, hence, we actually can choose the pivot vector with all its components giving the minimal estimate for the local fill-in. In such a way, to determine the pivot vector it is enough just to find any of its components. Moreover, from the two theorems below we conclude that to compute such a component we may use not the whole matrix $\partial_{z_{ki}} \bar{F}_k^\tau(z_{ki})_i^{(\mu)}$ but only the matrix

$$\begin{pmatrix} \partial_{z_{ki}} \bar{F}_k^\tau(z_{ki})_i^{Y^{(\mu)}} \\ \partial_{z_{ki}} g(z_{ki})^{(\mu)} \end{pmatrix}$$

of dimension $n - \mu + m \times n - \mu + m$, the last m rows of which are the corresponding block of the matrix

$$\partial g^l(Z) = (\partial_{y_{k1}} g(z_{k1}) \quad \dots \quad \partial_{y_{kl}} g(z_{kl}) \quad \partial_{x_{k1}} g(z_{k1}) \quad \dots \quad \partial_{x_{kl}} g(z_{kl})).$$

This reduces execution time essentially.

The proofs of both theorems will appear in [13].

Theorem 1 *Let an l -stage RK formula with dense coefficient matrix A be used for constructing matrix (5). Suppose that the $\mu+1$ -th step of vector Gauss method is being implemented and the $(\mu+p, \mu+q)$ -th element of the block $\partial_{y_{ki}} \bar{F}_k^\tau(z_{ki})_i^{Y^{(\mu)}}$, $\mu = 0, 1, \dots, n - 1, i = 1, 2, \dots, l, p, q = 1, 2, \dots, n - \mu$, is chosen as the i -th component of pivot vector. Then the local fill-in of the matrix $\partial_{z_{ki}} \bar{F}_k^\tau(z_{ki})_i^{(\mu)}$ from (10) is given by the (p, q) -th element of the matrix*

$$\begin{aligned} G^{(\mu)} = & \left(B_{yi}^{Y^{(\mu)}} \left(\bar{B}_{yi}^{Y^{(\mu)}} \right)^T + B_{xi}^{Y^{(\mu)}} \left(\bar{B}_{xi}^{(\mu)} \right)^T \right) B_{yi}^{Y^{(\mu)}} \\ & + l \cdot B_{yi}^{Y^{(\mu)}} \left(\bar{B}_y^{(\mu)} \right)^T B_{yi}^{(\mu)} + B_{xi}^{Y^{(\mu)}} \left(\bar{B}_x^{l(\mu)} \right)^T B_y^{(\mu)}, \end{aligned} \tag{11}$$

where the matrices $B_{y_i}^{Y(\mu)}$, $B_{x_i}^{Y(\mu)}$, $B_y^{(\mu)}$, $B_x^{(\mu)}$ are obtained from the active submatrices of $\partial_{y_{k_i}} \bar{F}_k^\tau(z_{k_i})^{Y(\mu)}$, $\partial_{x_{k_i}} \bar{F}_k^\tau(z_{k_i})^{Y(\mu)}$, $\partial_{y_{k_i}} g(z_{k_i})^{(\mu)}$, $\partial_{x_{k_i}} g(z_{k_i})^{(\mu)}$ by replacing nonzero elements with the unity ones, $\bar{B}_{y_i}^{Y(\mu)} = M_{(n-\mu) \times (n-\mu)} - B_{y_i}^{Y(\mu)}$, $\bar{B}_{x_i}^{Y(\mu)} = M_{(n-\mu) \times m} - B_{x_i}^{Y(\mu)}$, $\bar{B}_y^{(\mu)} = M_{m \times (n-\mu)} - B_y^{(\mu)}$, $\bar{B}_x^{(\mu)} = M_{m \times (n-\mu)} - B_x^{(\mu)}$, and the matrix $\bar{B}_x^{l(\mu)}$ is obtained by multiplication of the diagonal elements of matrix $\bar{B}_x^{(\mu)}$ by $l - 1$ and the nondiagonal elements by l .

Theorem 2 *Let all the conditions of Theorem 1 hold. Then the maximum possible local fill-in of the matrix $\partial_{z_{k_i}} \bar{F}_k^\tau(z_{k_i})^{(\mu)}$ from (10) (not necessarily coinciding with the real one) is given by the (p, q) -th element of the matrix*

$$\hat{G}^{(\mu)} = \left(\left(B_{y_i}^{Y(\mu)} - I_{n-\mu} \right) M_{(n-\mu) \times (n-\mu)} + B_{x_i}^{Y(\mu)} M_{m \times (n-\mu)} \right) \left(B_{y_i}^{Y(\mu)} - I_{n-\mu} \right) + l \left(\left(B_{y_i}^{Y(\mu)} - I_{n-\mu} \right) M_{(n-\mu) \times m} + B_{x_i}^{Y(\mu)} M_{m \times m} \right) B_y^{(\mu)}. \tag{12}$$

Theorems 1 and 2 not only allow the pivot vector to be determined but also make it possible to estimate the cost of the first and the second strategies (in terms of multiplication operations). Thus, taking into account that only n steps of the vector Gauss method are required for the full factorization of the matrix $\partial \bar{F}_k^\tau(Z)^Y$, the whole number of multiplications for the first strategy is

$$\sum_{\mu=0}^{n-1} (2(n-\mu)^3 + (4m+1)(n-\mu)^2 + m^2(n-\mu)) = \frac{n^2(n+1)^2}{2} + (4m+1) \frac{n(n+1)(2n+1)}{6} + m^2 \frac{n(n+1)}{2}. \tag{13}$$

Similarly, from formula (12) it follows that the quantity of multiplication operations for the second strategy is estimated by formula:

$$3 \sum_{\mu=0}^{n-1} (n-\mu)^2 = \frac{n(n+1)(2n+1)}{2}. \tag{14}$$

To illustrate the theoretical result, we consider a simple example. Let $m = n = 2$ and $l = 1, 2, 3, 4$. Now using formulas (7), (9), (13), (14) of the paper and (3.11), (3.12) from [13] we calculate the number of multiplications required to implement the first and the second strategies in the frames of standard Gauss method, Modification II and the vector Gauss method for RK methods with dense coefficient matrices. Table 1 gives the data for the first strategy, and Table 2 shows the result for the second one. Even from this small example, we can see that only in the vector Gauss method computational cost of pivoting does not depend on the number of stages of RK formulas and remains rather minor.

At the end of this section we stress the following aspect. It was noted earlier that the elimination of variables from system (3) is split up into two stages. We eliminated the y -components by using parallel factorization of the matrix

Table 1. The number of multiplication operations in the first strategy of pivoting for the factorization of matrix $\partial\bar{F}_k^\tau(Z)^Y$ ($m = n = 2$)

number of stages	Gauss method	Modification II	vector Gauss method
1	182	60	75
2	2392	184	75
3	11286	372	75
4	34400	624	75

Table 2. The number of multiplication operations in the second strategy of pivoting for the factorization of matrix $\partial\bar{F}_k^\tau(Z)^Y$ ($m = n = 2$)

number of stages	Gauss method	Modification II	vector Gauss method
1	25	10	15
2	174	30	15
3	559	60	15
4	1292	100	15

$\partial\bar{F}_k^\tau(Z)^Y$ and by applying the vector Gauss method. Then we eliminated the x -components. The first stage is more important for optimization because the most part of arithmetical operations falls on it. Therefore in [11], [13] we gave a way to further reduction of the number of operations at this stage. The whole version was called *Modification III*. It incorporates the approach of vector Gauss method with the idea to use the reduced matrix

$$\begin{pmatrix} \partial\bar{F}_k^\tau(Z)^Y \\ \partial g^l(Z) \end{pmatrix}$$

of dimension $(nl + m) \times (m + n)l$ instead of the full matrix (5) while eliminating the y -components. We refer to [11] or [13] for the numerical example.

4 Conclusion

In the paper we further developed the idea suggested in [11], [13] for parallel factorization of the submatrices $\partial\bar{F}_k^\tau(Z)_i^Y$, $i = 1, 2, \dots, l$. We showed that the most effective algorithm will be obtained only if we use the vector Gauss method. The main advantage of the new method is that we can significantly reduce the number of arithmetical operations required to solve problem (1). Moreover, the computation cost of pivoting in the vector Gauss method does not depend on the number of stages of dense RK formulas applied to (1). Finally, we note that vector computers are the most useful to implement the vector Gauss method.

References

1. Ascher, U.M., Petzold, L.P.: Computer methods for ordinary differential equations and differential-algebraic equations. SIAM, Philadelphia, 1998
2. Brenan K.E., Campbell S.L., Petzold L.R.: Numerical solution of initial-value problems in differential-algebraic equations. North-Holland, N.Y., Amsterdam, L., 1989
3. Gear, C.W.: Differential-algebraic equations index transformations. SIAM J. Sci. Stat. Comput., **9** (1988) 39–47
4. Hairer, E., Wanner, G.: Solving ordinary differential equations II: Stiff and differential-algebraic problems. Springer-Verlag, Berlin, 1996
5. Ikeda, N., Marumo, F., Shiratare, M., Sato, T.: A model of overall regulation of body fluids. Ann. Biomed. Eng. **7** (1979) 135–166
6. Kulikov, G.Yu.: The numerical solution of the autonomous Cauchy problem with an algebraic relation between the phase variables (non-degenerate case). (*in Russian*) Vestnik Moskov. Univ. Ser. 1 Mat. Mekh. (1993) No. 3, 6–10; *translation in* Moscow Univ. Math. Bull. **48** (1993) No. 3, 8–12
7. Kulikov, G.Yu., Thomsen, P.G.: Convergence and implementation of implicit Runge-Kutta methods for DAEs. Technical report 7/1996, IMM, Technical University of Denmark, Lyngby, 1996
8. Kulikov, G.Yu.: Convergence theorems for iterative Runge-Kutta methods with a constant integration step. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. **36** (1996) No. 8, 73–89; *translation in* Comp. Maths Math. Phys. **36** (1996) No. 8, 1041–1054
9. Kulikov, G.Yu., Korneva, A.A.: On effective implementation of iterative Runge-Kutta methods for differential-algebraic equations of index 1. (*in Russian*) In: Melnikov, B.F. (ed.): Basic problems of mathematics and mechanics. **3** (1997). Ulyanovsk State University, Ulyanovsk, 103–112
10. Kulikov, G.Yu.: Numerical solution of the Cauchy problem for a system of differential-algebraic equations with the use of implicit Runge-Kutta methods with nontrivial predictor. (*in Russian*) Zh. Vychisl. Mat. Mat. Fiz. **38** (1998) No. 1, 68–84; *translation in* Comp. Maths Math. Phys. **38** (1998) No. 1, 64–80
11. Kulikov, G.Yu., Korneva, A.A.: On efficient application of implicit Runge-Kutta methods to large-scale systems of index 1 differential-algebraic equations. In: Alexandrov, V.N. et al (eds.): Computational Science — ICCS 2001. International Conference, San Francisco, CA, USA, May 28–30, 2001. Proceedings, Part I. Lecture Notes in Computer Science, **2073** (2001), 832–841
12. Kulikov, G.Yu.: On effective implementation of implicit Runge-Kutta methods for systems of differential and differential-algebraic equations of index 1. (*in Russian*) In: Andreev, A.C. (ed.): Basic problems of mathematics and mechanics. Ulyanovsk State University, Ulyanovsk. (to appear)
13. Kulikov, G.Yu., Korneva, A.A., Benderskaya G.Ya.: On numerical solution of large-scale systems of index 1 differential-algebraic equations. (*in Russian*) Fundam. Prikl. Mat. (to appear)
14. Kværnø, A.: The order of Runge-Kutta methods applied to semi-explicit DAEs of index 1, using Newton-type iterations to compute the internal stage values. Technical report 2/1992, Mathematical Sciences Div., Norwegian Institute of Technology, Trondheim, 1992
15. Tewarson R.P.: Sparse matrices. Academic Press, New York and London, 1973