Mathematical Modelling and Analysis, vol. 3, pp. 045–056
R.Čiegis (Ed)

DEGENERATE MATRIX METHOD FOR SOLVING NONLINEAR SYSTEMS OF DIFFERENTIAL EQUATIONS

T. CĪRULIS, O. LIETUVIETIS

University of Latvia
Raiņa bulv. 19, Rīga, LV–1586, Latvia
E-mail: cirulis@lanet.lv

ABSTRACT

Degenerate matrix method for numerical solving nonlinear systems of ordinary differential equations is considered. The method is based on an application of special degenerate matrix and usual iteration procedure. The method, which is connected with an implicit Runge-Kutta method, can be simply realized on computers. An estimation for the error of the method is given.

1. INTRODUCTION

Many problems of physics and mathematics reduce to solving nonlinear systems of differential equations. Many classical methods for the numerical solution of such systems are developed: different versions of Runge-Kutta method, multistep Adam's methods and so forth. If the solution of a system is very smooth, for example, analytical, then high precision spectral or pseudospectral methods can be used. All these methods have been programmed for computers. In this paper we propose a method which we call the DM-method (degenerate matrix method). It is based on the global nonsaturated interpolations and on the application of special degenerate matrices in multistep procedures by means of iterative loops. According to the character of approximation, the method proposed here is similar to the collocation version of the spectral methods [1]. However, these methods differ essentially from each other by numerical realization. Spectral methods use the discrete transforms, while the proposed method applies the special pseudoinverse matrices and the usal iteration process. DM-method can be presented also as implicit Runga-Kutta method or a modification of the collocation scheme. These methods are given in [3,4]. Mathematical algorithm for a solution has been represented in matrix form. Therefore, the method can be used for nonlinear systems of arbitrary degree and simply programmed for computers. The rapid convergence of the iteration process and the possibility to calculate the elements of matrices by elementary representations with a very high precision gives us the numerical solutions of differential equations with the precision near to the working precision on computers. For example, calculations by means of the software package "Mathematica-3" give the possibility to find solutions of nonlinear systems of differential equations with a precision of 10^{-50} or higher.

2. MATHEMATICAL BASIS OF THE METHOD

To obtain a numerical solution of the system

$$\frac{dy_k}{dt} = f_k(t, y_1, y_2, \ldots, y_m),\ y_k(a) = \alpha_k,\ k = 1, 2, \ldots, m \tag{2.1}$$

in the interval $t \in [a, b]$ we recommend the DM-method (degenerate matrix method) based on an application of special degenerate matrices together with iteration process. We start from Lagrange's interpolation with nodes $a = t_0 < t_1 < \ldots < t_{N+1} = b$. Let us assume that $q_{N+2}(t)$ is a polynomial, which has zeroes t_i, $i = 0, 1, \ldots, N + 1$, and the function $y(t)$ is approximated by Lagrange's interpolation polynomial

$$y(t) \approx \mathbf{L}_{N+1}(y, t) \equiv q_{N+2}(t) \sum_{k=0}^{N+1} \frac{y(t_k)}{(t - t_k) q'_{N+2}(t_k)}. \tag{2.2}$$

Then the approximation

$$\mathbf{y}'_{N+2} \approx \mathbf{\Delta}_{N+2} \mathbf{y}_{N+2} \tag{2.3}$$

holds for the vector of derivatives

$$\mathbf{y}'_{N+2} = \{y'(t_0), y'(t_1), \ldots, y'(t_{N+1})\}^T \tag{2.4}$$

and for the vector of the functions

$$\mathbf{y}_{N+2} = \{y(t_0), y(t_1), \ldots, y(t_{N+1})\}^T \tag{2.5}$$

contracted on nodes t_i. Here $\mathbf{\Delta}_{N+2}$ is the $(N + 2) \times (N + 2)$ interpolation matrix for derivatives which has elements [2]

$$\delta_{ik} = \frac{q'_{N+2}(t_i)}{(t_i - t_k) q'_{N+2}(t_k)} \text{ if } i \neq k,\ \delta_{kk} = \frac{q''_{N+2}(t_k)}{2 q'_{N+2}(t_k)},\ i, k = 0, 1, \ldots, N + 1. \tag{2.6}$$

The interpolation matrix for derivatives $\mathbf{\Delta}_{N+2}$ is always degenerate and rank$\mathbf{\Delta}_{N+2} = N+1$. As the matrix $\mathbf{\Delta}_{N+2}$ has no inverse, we define the following pseudoinverse matrix.

Definition 2.1. *Matrix $\mathbf{B}_{N+2}$ is called a pseudoinverse matrix for $\mathbf{\Delta}_{N+2}$ if the equality*

$$\mathbf{B}_{N+2}\mathbf{\Delta}_{N+2} = \mathbf{E}_{N+2} - \mathbf{I}_{N+2} \tag{2.7}$$

holds, where $\mathbf{E}_{N+2}$ is the identity matrix and all elements of $\mathbf{I}_{N+2}$ are zeroes except each element of first column is the digit 1.

Contracting the system (2.1) on nodes t_i and replacing derivatives by (2.3) we reduce this system into the following matrix form

$$\mathbf{\Delta}_{N+2}\mathbf{Y}[N+2,m] = \mathbf{F}[N+2,m], \tag{2.8}$$

where $\mathbf{Y}[N+2,m]$ and $\mathbf{F}[N+2,m]$ are $(N+2)\times m$ matrices with the elements

$$y_{ik} \approx y_k(t_i) \text{ and } f_{ik} = f_k\left(t_i, y_1(t_i), y_2(t_i), \ldots, y_m(t_i)\right), \tag{2.9}$$

respectively. Multiplying the equation (2.8) from the left side by a pseudoinverse matrix $\mathbf{B}_{N+2} = h\mathbf{G}_{N+2}$, we obtain

$$\mathbf{Y}[N+2,m] = h\mathbf{G}_{N+2}\mathbf{F}[N+2,m] + \mathbf{Y}_0[N+2,m]. \tag{2.10}$$

Here $\mathbf{Y}_0$ is matrix $(N+2)\times m$; its elements are the inital values of unknown functions $y_k(a) = \alpha_k$ given by (2.1), i.e.,

$$y_{ik}^{(0)} = \alpha_k,\ i = 0, 1, \ldots, N+1; k = 1, 2, \ldots, m. \tag{2.11}$$

The system (2.10) can be solved iteratively if a norm of the matrix $\mathbf{B}_{N+2}$ is small enough. The properties of the solution of (2.10) depend on the choise of the nodes t_i and the matrix $\mathbf{B}_{N+2}$. The most important properties of matrices $\mathbf{\Delta}_{N+2}$ and $\mathbf{B}_{N+1}$ are the following.

Lemma 2.2. *Substituting $t = \alpha x + \beta$ and $t_i = \alpha x_i + \beta$ leads to multiplying the matrices $\mathbf{\Delta}_{N+2}$ and $\mathbf{B}_{N+2}$ by factors α and $1/\alpha$, respectively.*

Lemma 2.3. *Elements $d_{ik}(m)$ of the m-degree matrix $\mathbf{\Delta}_{N+2}$, i.e., of the matrix $\mathbf{\Delta}_{N+2}^m$ are exact representatoins*

$$d_{ik} = \frac{m!}{q'_{N+2}(t_k)} \sum_{j=0}^{m-1} \frac{(-1)^j q_{N+2}^{(m-j)}(t_i)}{(m-j)!(t_i - t_k)^{j+1}} \text{ if } i \neq k,$$

$$d_{kk}(m)=\frac{q_{N+2}^{(m+1)}(t_k)}{(m+1)q'_{N+2}(t_k)},\ i,k=0,1,\dots,N+1. \tag{2.12}$$

Proof of this Lemma is given in [2].

Lemma 2.4. *Matrix $\mathbf{\Delta}_{N+2}$ has the representation*

$$\mathbf{\Delta}_{N+2}=(\mathbf{DV}_{N+2})\mathbf{V}_{N+2}^{-1}, \tag{2.13}$$

where $\mathbf{V}_{N+2}$ and $\mathbf{DV}_{N+2}$ are the Vandermonde matrix and its derivative respectively to nodes $t_0,t_1,\dots,t_{N+1}$.

Proof. For any polynomial $P_{N+2}(t)$ the formula $P'_{N+2}=\mathbf{\Delta}_{N+2}P_{N+2}$ is exact. Taking P_{N+2} equal to $1,t,t^2,\dots,t^{N+1}$, we obtain the equalities, which may be joined in one matrix equality

$$\mathbf{\Delta}_{N+2}\mathbf{V}_{N+2}=\mathbf{DV}_{N+2}. \tag{2.14}$$

Multiplying (2.14) from the right side by the matrix $\mathbf{V}_{N+2}^{-1}$, which exist because $t_0,t_1,\dots,t_{N+1}$ are different numbers, leads to (2.13). □

Lemma 2.5. *Matrix $\mathbf{\Delta}_{N+2}$ is degenerate and the rank$\mathbf{\Delta}_{N+2}=N+1$.*

Proof is gained from the Lemma 2.4.

Lemma 2.6. *A pseudoinverse matrix $\mathbf{B}_{N+2}$ is not unique for the given matrix $\mathbf{\Delta}_{N+2}$. The general form for the elements b_{ik} of $\mathbf{B}_{N+2}$ is*

$$b_{ik}=b_{ik}^{(0)}+\frac{c_i}{q'_{N+2}(t_k)}\ ,i,k=0,1,2,\dots,N+1, \tag{2.15}$$

$$\text{where}\quad b_{ik}^{(0)}=\frac{1}{q'_{N+2}(t_k)}\int_{t_0}^{t_i}\frac{q_{N+2}(\tau)}{\tau-t_k}d\tau, \tag{2.16}$$

and c_i are arbitrary constants not depending on k.

Proof. At first we prove that the matrix $\mathbf{B}_{N+2}^{(0)}$ with elements (2.13) satisfies the matrix equation (2.7). Integrating the identity

$$P'_{N+2}(t)=q_{N+2}(t)\sum_{k=0}^{N+1}\frac{P'_{N+2}(t_k)}{(t-t_k)q'_{N+2}(t_k)} \tag{2.17}$$

on the interval $[t_0,t]$ leads to

$$P_{N+2}(t)-P_{N+2}(t_0)=\sum_{k=0}^{N+1}\frac{P'_{N+2}(t_k)}{q'_{N+2}(t_k)}\int_{t_0}^{t}\frac{q_{N+2}(\tau)}{\tau-t_k}d\tau. \tag{2.18}$$

Contracting this equality at nodes t_i and writing ones in the matrix form, we can obtain

$$\mathbf{P}_{N+2} = \mathbf{B}^{(0)}_{N+2}\mathbf{P}'_{N+2} + P_{N+2}(t_0)\mathbf{1}_{N+2}, \tag{2.19}$$

where $\mathbf{1}_{N+1} = (1,1,\ldots,1)^T$. Substituting $\mathbf{P}'_{N+2} = \mathbf{\Delta}_{N+2}\mathbf{P}_{N+2}$ gives the matrix equation

$$(\mathbf{B}^{(0)}_{N+2}\mathbf{\Delta}_{N+2} - \mathbf{E}_{N+2} + \mathbf{I}_{N+2})\mathbf{P}_{N+2} = 0 \tag{2.20}$$

for each polynomial $P_{N+2}(t)$. It follows that the matrix $\mathbf{B}^{(0)}_{N+2}$ is a pseudoinverse matrix for $\mathbf{\Delta}_{N+2}$. Let us the vector $\mathbf{w}_{N+2}$ is a nontrivial solution of the equation $\mathbf{w}_{N+2}\mathbf{\Delta}_{N+2} = 0$. By means of the equality

$$\sum_{k\neq i}\frac{1}{t_k - t_i} = -\frac{q''_{N+2}(t_i)}{q'_{N+2}(t_i)}, \tag{2.21}$$

it is possible to prove directly that one of the nontrivial solutions is

$$\mathbf{w}_{N+2} = (\frac{1}{q'_{N+2}(t_0)}, \frac{1}{q'_{N+2}(t_1)}, \ldots, \frac{1}{q'_{N+2}(t_{N+1})}). \tag{2.22}$$

□

Corollary 2.7. *The general form for the pseudoinverse matrix* $\mathbf{B}_{N+2}$ *corresponding to* $\mathbf{\Delta}_{N+2}$ *is*

$$\mathbf{B}_{N+2} = \mathbf{B}^{(0)}_{N+2} + \mathrm{diag}(c_0, c_1, \ldots, c_{N+2})\mathbf{W}_{N+2}, \tag{2.23}$$

where elements of $\mathbf{B}^{(0)}_{N+2}$ *are given by (2.16), elements of* $\mathbf{W}_{N+2}$ *are* $w_{ik} = 1/q'_{N+2}(t_k)$, $i,k = 0,1,\ldots,N+1$, *and* c_i *are arbitrary constants. The choise of* c_i *allows to use various pseudoinverse matrices for the given matrix* $\mathbf{\Delta}_{N+2}$.

Lemma 2.8. *For elements* b_{ik} *of each pseudoinverse matrix* $\mathbf{B}_{N+2}$, *in the case of* $N \geq 2$ *the following equalities hold*

$$\sum_{k=0}^{N+1} b_{ik} = t_i - t_0,\ i = 0,1,\ldots,N+1. \tag{2.24}$$

Proof. We have for $b^{(0)}_{ik}$ by (2.16) and (2.1) with $y(t) \equiv 1$

$$\sum_{k=0}^{N+1} b^{(0)}_{ik} = \int_{t_0}^{t_i}\sum_{k=0}^{N+1}\frac{q_{N+2}(\tau)}{(\tau - t_k)q'_{N+2}(t_k)}d\tau = \int_{t_0}^{t_i} d\tau = t_i - t_0.$$

For the b_{ik} in general equalities hold by the (2.15) and by the identity

$$\sum_{k=0}^{N+1} \frac{(x-t_k)^m}{q'_{N+2}(t_k)} \equiv 0, m=0,1,\ldots,N, \tag{2.25}$$

and arbitrary x. The identity (2.20) can be proven using the classical Cauchy residue theorem for the contour integral in the following way:

$$\sum_{k=0}^{N+1} \frac{(x-t_k)^m}{q'_{N+2}(t_k)} = \sum_{k=0}^{N+1} \mathrm{res}[\frac{(x-z)^m}{q_{N+2}(z)}, t_k] =$$

$$\frac{1}{2\pi i} \oint_L \frac{(x-z)^m dz}{q_{N+2}(z)} = -\mathrm{res}[\frac{(x-z)^m}{q_{N+2}(z)}, \infty] = 0,$$

where L is a closed contour in the complex z-plane which includes strictly in itself the points t_k. □

LEMMA 2.9. *For the norm of each pseudoinverse matrix $\mathbf{B}_{N+2}$ with elements (2.15)–(2.16), we have the inequality*

$$\|\mathbf{B}_{N+2}\| := \max_{0\le i\le N+1} \sum_{k=0}^{N+1} |b_{ik}| \ge t_{N+1} - t_0.$$

Proof. $\|\mathbf{B}_{N+2}\| \ge \max|\sum_{k=0}^{N+1} b_{ik}| = \max|t_i - t_0| = t_{N+1} - t_0$. □

Futher we choose the nodes t_i and the corresponding pseudoinverse matrix $\mathbf{B}_{N+2}$ to ensure the following two properties:

1) The approximation of functions is nonsaturated, that is, the rate of the convergence increases together with the smoothness of functions under the approximation.

2) $\|\mathbf{B}_{N+2}\| = t_{N+1} - t_0$, that is, the norm is the minimal possible.

The first property guarantees that an error for the smooth solution under approximation is small, but the second one allows to eliminate round-off errors by the increasing N. These two properties hold true if we choose, for example, $c_i = 0, i = 0, 1, \ldots, N+1$ (the method is a collocation method [3]) and

$$t_i = a + 0.5(b-a)(1+x_i),\ x_i = \cos\frac{i\pi}{N+1}. \tag{2.26}$$

Elements b_{ik} for the matrix $\mathbf{B}_{N+2}$ are present in the form $b_{ik} = (b-a)g_{ik}$, where

$$g_{ik} = \frac{1}{2p'_{N+2}(x_k)} \int_{-1}^{x_i} \frac{p_{N+2}(x)}{x-x_k} dx, \quad p_{N+2}(x) = (1-x^2)U_N(x). \tag{2.27}$$

$U_N(x)$ is the Chebyshev's polynomial of the second kind. The elements g_{ik} do not depend on the interval $[a,b]$ and

$$\|\mathbf{G}_{N+2}\| = \max_{0\le i\le N+1} \sum_{k=0}^{N+1} |g_{ik}| = 1. \tag{2.28}$$

Moreover, the elements g_{ik} have the following simple exact representations by elementary functions which are very suitable for calculations by computers with any desired precision:

$$g_{i,0} = \frac{2N^2+4N+1+(-1)^i[(N+1)^2 x_i - N(N+2)]}{4N(N+1)^2(N+2)}, \tag{2.29}$$

$$g_{i,N+1} = \frac{(-1)^{N+1}\{1+(-1)^i[(N+1)^2 x_i + N(N+2)]\}}{4N(N+1)^2(N+2)}, \tag{2.30}$$

$$g_{ik} = \frac{1}{(N+1)\sin\theta_k} \sum_{n=0}^{N} s[i,n]\sin{(n+1)\theta_k}, \tag{2.31}$$

where $i = 0,1,2,\ldots,N+1,\ \ k = 1,2,\ldots,N,$

$$s[i,n] = \frac{\cos(n+3)\theta_i}{4(n+3)} + \frac{\cos(n-1)\theta_i}{4(n-1)} - \frac{\cos{(n+1)\theta_i}}{2(n+1)} - \frac{2}{(n^2-1)(n+3)}$$

if $n \ge 2$ and

$$s[i,0] = (1+x_i)^2(2-x_i)/3; \ \ s[i,1] = (1-x_i^2)^2/2,$$

$$\theta_i = \pi i/(N+1), x_i = -\cos\theta_i.$$

Now we obtain instead of (2.10) the equation

$$\mathbf{Y}[N+2,m] = h\mathbf{G}_{N+2}\mathbf{F}[N+2,m] + \mathbf{Y}_0[N+2,m]. \tag{2.32}$$

Due to the equality $\|\mathbf{G}_{N+2}\| = 1$, the sufficient condition for convergence of the iteration process is the inequality $hL < 1$, where L is a Lipschiz's constant for the system (2.1) on $[a,b]$. This condition guarantees that the nonlinear operator $h\mathbf{G}_{N+2}\mathbf{F}[N+2,m]$ in (2.20) is a contracting operator mapping the space of $(N+2)\times m$ matrices into itself. For the large interval $[a,b]$ it is necessary to divide it into small enough parts $a = h_0 < h_1 < \ldots < h_n = b$, and to solve the system (2.1) in each subinterval $[h_l, h_{l+1}]$, $l = 0,1,\ldots,n-1$, separately. In addition, the solution at the points $t = h_{l+1}$ must be choosen as an initial value for the solution in the next interval $[h_{l+1}, h_{l+2}]$. The matrix $\mathbf{G}_{N+2}$ does not change, but the new nodes t_k, $k = 0,1,\ldots,N+1$ may be calculated by (2.26) with $a = h_l, b = h_{l+1}$ and $h = h_{l+1} - h_l$. Thus, the global solution for system (2.1) can be found by using the aforementioned multistep procedure involving iterative loops.

Comentaries. 1. In practice the condition $Lh < 1$ or the choice of step h can usually be determined by means of analysis of the rate of convergence in the iteration procedure. Also, the number of relevant iterations can be chosen in various ways. For example: until the fixed point is reached, until the required precision is attained, and so forth.

2. The step h and the number N of interior nodes on the subinterval are independent of each other. By repeating calculations of the solution for various N and comparing the obtained results, we can also make a conclusion about the precision of the final result.

3. If certain functions of the solution increase (decrease) rapidly, then the Lipschitz constant can be reduced for the next step by a substitution $y_i(t) = \exp(\lambda_i t)v_i(t), i = 1, 2, \ldots, m$, with suitable constants λ_i. Thus, it is possible to choose a larger step for this case.

4. DM-method can be used as a starting method for multistep methods.

3. THE ESTIMATION OF THE ERROR OF THE METHOD

The precise solution $y_k(t_i)$ and rough solution $\tilde{y}_k(t_i)$ of (2.1) contracted on nodes by (2.26) satisfy the matrix equations

$$\mathbf{Y}[N+2, m] = h\mathbf{G}_{N+2}\mathbf{F}[N+2, m] + \mathbf{Y}_0[N+2, m] + \mathbf{R}[N+2, m], \quad (3.1)$$

$$\tilde{\mathbf{Y}}[N+2, m] = h\mathbf{G}_{N+2}\tilde{\mathbf{F}}[N+2, m] + \tilde{\mathbf{Y}}_0[N+2, m], \quad (3.2)$$

respectively, where elements of matrix $\mathbf{R}[N+2, m]$ are

$$r_{ik} = \int_{t_0}^{t_i} R_{N+1}(f_k, \tau)d\tau, (i = 1, \ldots, N+1; k = 1, \ldots, m), \quad (3.3)$$

and $R_{N+1}(f, t) = f(t) - \mathbf{L}_{N+1}(f, t)$ is the remainder of Lagrange's interpolation (2.2) with nodes (2.26), where $f_k(t) \equiv f_k(t, y_1(t), \ldots, y_m(t))$. Further, we will suppose that the functions $y_k(t)$ and $f_k(t)$ are elements of some function space $\mathbf{C}^s[t_0, t_{N+1}]$. The elements r_{ik} can be estimated in various way. Let us assume that, for all $i, k, \quad |r_{ik}| \leq h\sigma(N, h)$ holds. Then according to the nonsaturatedness of the interpolation [2] we have

$$\sigma(N, h) \leq \frac{M_s \ln(N+1)}{(N+1)^{s+1}}. \quad (3.4)$$

Here M_s is a constant which does not depend on i, k, N. For small $h = t_{N+1} - t_0$ we have

$$\sigma(N, h) \leq h^{-1} \int_{t_0}^{t_{N+1}} |R_{N+1}(f_k, t)| dt = O(h^{N+2}) \quad (3.5)$$

as an estimation for the remainder of an interpolating quadrature.

We denote a norm of the error-matrix by

$$\epsilon_N \equiv \|\mathbf{Y}[N+2,m] - \tilde{\mathbf{Y}}[N+2,m]\| := \max_{0\le i\le N+1} \sum_{k=1}^{m} |y_k(t_i) - \tilde{y}_k(t_i)|, \quad (3.6)$$

and also suppose that the iteration process goes on until the fixed point is reached. Taking into account the condition $\mathbf{Y}_0 \equiv \tilde{\mathbf{Y}}_0$ the error of the solution can be estimated on the first step in the following way:

$$\epsilon_N \le q\sigma(N,h)/L, \text{ where } q = \frac{mhL}{1-mhL}. \quad (3.7)$$

Now suppose that we realize n equal steps and $nh = b - a$. If $\epsilon_N^{(j)}$, $(j = 1,2,\ldots,n)$ is the norm of error-matrix on the j-th step defined analogously as in (3.6), then the following recurrence inequality can be proved for $\epsilon_N^{(n)}$ after a simple but lengthy conversion:

$$\epsilon_N^{(n)} \le q\sum_{j=1}^{n-1} \epsilon_N^{(j)} + nq\sigma(N,h)/L, \quad \epsilon_N^{(1)} \equiv \epsilon_N. \quad (3.8)$$

Solving these inequalities we obtain

$$\epsilon_N^{(n)} \le nq(1+q)^{n-1}\sigma(N,h)/L, (n = 1,2,\ldots), \quad (3.9)$$

where $\sigma(N,h)$ and q are given by (3.4)–(3.5) and (3.7) respectively. Therefore, by (3.4)–(3.5) and (3.9) we conclude that if $0 \le mhL \le 1$:

1) $\epsilon_N^{(n)} = O(h^{N+2})$ as $h \to 0$, N is fixed;

2) $\epsilon_N^{(n)} = O\left((N+1)^{-s-1}\ln(N+1)\right)$ as $N \to +\infty$.

The O-character of the error can be observed by practical calculations too. For example, we consider numerical results for the Lorenz's system

$$\left\{ \begin{array}{lcl} \dot{x} & = & \sigma(y-x) \\ \dot{y} & = & rx - xz - y \\ \dot{z} & = & xy - bz \end{array} \right. \qquad \sigma = 10, \quad r = 28, \quad b = 8/3 \quad (3.10)$$

with the initial values $x(0) = 96/100$ and $y(0) = z(0) = 0$ in the interval $t \in [0,1]$. The solution at point $t = 1$ obtained by means of the DM-method with precision of 10^{-50} is

$$x(1) = -9.4185265666\ 8328650990\ 6763403446\ 0148597258\ 7325487820$$

$$y(1) = -9.1460603281\ 9364807619\ 4431441281\ 1484693600\ 3068231643$$

Table 1.
Numbers of the correct decimal places of the solution.

N/h	0.25	0.2	0.1	0.05	0.025	0.01	0.005	0.0025
3		0-1 (384)	2-3 (133)	3-5 (84)	5-6 (60)	7-9 (44)	9-10 (37)	11-12 (31)
7	1-2 (231)	2-4 (175)	6 (96)	8-10 (68)	9-10 (52)	16-17 (40)	19-20 (34)	22-23 (29)
11	3-4 (152)	6-7 (126)	9 (80)	12-13 (60)	18-20 (48)	24-25 (38)	28-30 (32)	33-34 (28)
15	5-6 (125)	8-9 (107)	12-14 (73)	17 (55)	23-24 (44)	32-33 (36)	38-39 (31)	43-44 (27)
19	7-8 (110)	11-12 (97	15-16 (68)	21-22 (53)	28-29 (43)	41 (35)	47-48 (31)	54-55 (27)
23	9-10 (103)	12-14 (89)	18-19 (66)	25-26 (51)	33-34 (42)	49-50 (35)	56-57 (31)	
27	10-11 (96)	15-16 (87)	21-22 (63)	29-30 (49)	37-39 (43)	56-58 (35)		
51	19-21 (81)	27-29 (75)	39-40 (60)	54-55 (50)				

$$z(1) = 28.5481201472\ 8984748207\ 2902880080\ 6776859476\ 7334323951.$$

To eliminate the possibility of round-off errors we have calculated elements g_{ik} of matrix $\mathbf{G}_{N+2}$ with a precision $\delta = 10^{-60}$. The computation was done with various steps h and numbers N of nodes on each subinterval. The results are showed in the Table 1, where numbers on the first line show the correct decimal places of the solution at point $t = 1$ for each h and N. Numbers on the second line in parentheses show the maximum number of iterations made on each of n subintervals with $nh < 1$.

4. DM-METHOD WITH NODES AS ZEROES OF THE GEGENBAUER POLYNOMIALS

We will consider a sheme where the nodes are the set of zeroes of the Gegenbauer polynomials $C_N^\lambda(x)$ that can be extended with endpoints ±1. There it is four cases: $-1 < x_1, x_2 < \ldots < x_N < 1$, $\quad -1 = x_0 < x_1 < \ldots < x_N < 1$, $-1 < x_1 < x_2 < \ldots < x_N < x_{N+1} = 1$, $\quad -1 = x_0 < x_1 < \ldots < x_N < x_{N+1} = 1$. Here $x_k, k = 1, 2, \ldots, N$ are zeroes of the polynomials $C_N^\lambda(x)$. Let us choose the case 2). Pseudoinverse matrices can be obtained by formulas of Section 2 if we replace $N+2$ for $N+1$. Thus,

$$\mathbf{B}_{N+1} = \mathbf{B}_{N+1}^{(0)} + \operatorname{diag}(c_0, c_1, \ldots, c_N)\mathbf{W}_{N+1}, \tag{4.1}$$

where elements of the matrices $\mathbf{B}^{(0)}_{N+1}$ and $\mathbf{W}_{N+1}$ are

$$b^{(0)}_{ik} = \frac{1}{p'_{N+1}(x_k)}\int_{-1}^{x_i}\frac{p_{N+1}(\tau)}{\tau - x_k}d\tau, \;\; w_{ik} = \frac{1}{p'_{N+1}(x_k)},$$

$$i,k = 0,1,\ldots,N,\; p_{N+1}(x) = (1+x)C^{\lambda}_n(x). \tag{4.2}$$

Using (2.10) and (4.1) in the case of one equation $y' = f(x,y)$, we obtain:

$$\mathbf{y}_{N+1} = y_0\mathbf{1} = [\mathbf{B}^{(0)}_{N+1} + \mathrm{diag}(c_0,c_1,\ldots,c_N)\mathbf{W}_{N+1}]\mathbf{f}_{N+1}, \tag{4.3}$$

where $\mathbf{y}_{N+1} = (y_0,y_1,\ldots,y_N)^T, \mathbf{f}_{N+1} = (f(x_0,y_0), f(x_1,y_1),\ldots,f(x_N,y_N))^T,$

$$\mathbf{1} = (1,1,\ldots,1)^T,\; y_k = y(x_k), k = 0,1,\ldots,N.$$

In order to create the sheme with a higer-order precision for $y_{N+1} = y(x_{N+1}) \equiv y(1)$ at the endpoint, we use the following idea. Multiplying the equation $y' = f(x,y)$ by $(1-x^2)^{\lambda-1/2}, \lambda > 1/2$, integrating it on the interval $[-1,1]$, applying the integration by parts and the Gausa-Lobatto quadrature formula at the nodes $-1 = x_0, x_1 < \ldots < x_N < x_{N+1} = 1$, lead to

$$m_{N+1}y_{N+1} - m_0y_0 + \sum_{k=1}^{N} m_kx_ky_k = (2\lambda-1)^{-1}\sum_{k=1}^{N} m_k(1-x_k^2)f(x_k,y_k), \tag{4.4}$$

where $m_k > 0$ are coefficients of quadrature, which can be represented by the Gegenbauer polynomials. Using (4.4) and taking account that $m_{N+1} = m_0$, we obtain

$$y_{N+1} = y_0 - \sum_{k=1}^{N} b_kx_ky_k + (2\lambda-1)^{-1}\sum_{k=1}^{N} b_k(1-x_k^2)f(x_k,y_k), \tag{4.5}$$

where $b_k = m_k/m_0, k = 1,2,\ldots,N$. The constants c_i in (4.3) are arbitrary. For example, we can choose $c_i = 0$. The choice of the constant c_0 gives a possibility to make the Dalquist function of the stability $R(z)$ given in [4] so that

1) $|R(z)| < 1$, if $\mathrm{Re}z < 0$; 2) $R(\infty) = 0$. This ensure the L-stability of the method regardless the choise of the other constants $c_1, c_2, \ldots, c_N$. It is possible also to choose constants c_i in the other way.

Concluding remarks. Formally, the DM-method can be used at the nodes (2.26) with arbitrary distributed x_i satisfying the inequalities $-1 = x_0 < x_1 < \ldots < x_{N+1} = 1$, if elements of the matrix $\mathbf{G}_{N+2}$ are calculated by (2.27) with $p_{N+2}(x) = (1-x^2)(x-x_1)\ldots(x-x_N)$. However, in the generalal case the equality (2.16) does not hold and the approximation is not nonsaturated. We mention some interesting special cases.

1. The nodes x_i are distributed on $[-1,1]$ uniformly. In this case $\|\mathbf{G}_{N+2}\|$ increases together with N for $N > 8$, and the DM-method becomes unstable. Therefore, the DM-method can be used in the case of uniformly distributed nodes only with small N, for example, with $N \leq 8$.

2. x_i are zeroes of $T_N(x)$ and endpoints ±1, where $T_N(x)$ is Chebyshev's polynomial of the first kind. In this case $\|\mathbf{G}_{N+2}\| = 1 + \alpha_N$, and $\alpha_N \to 0$ as $N \to +\infty$. The corresponding approximation is also nonsaturated. The DM-method can be used with precision near to the case with nodes as zeroes of Chebyshev's polynomials of the second kind.

3. x_i are zeroes of Legendre's polynomial $P_N(x)$ and endpoints ±1. In this case $\|\mathbf{G}_{N+2}\| = 1$, the approximation is also nonsaturated and the precision of the DM-method is near to the precision in the case with Chebyshev's polynomials.

4. x_i are zeroes of Jacobi's polynomial $P_N^{(1,1)}(x)$ and endpooints ±1. In this case $\|\mathbf{G}_{N+2}\| = 1$, the approximation is nonsaturated, but the precision of the DM-method is higher than for Chebyshev or Legendre polynomials. This follows from the other estimation for $\sigma(N,h)$ in (3.5). In this case $\sigma(N,h) = O(h^{2N+1})$ as an estimation for the remainder of the Gauss-Lobatto quadrature.

5. The choise $c_i = 0, i = 1, 2, \ldots, N+1$, and x_i the same as in **4**, leads to the Lobatto 3A sheme [4]. The method is A-stable, but not L-stable.

Choosing x_i as above and

$$c_i = \int_{x_i}^{1} (1-x) P_N^{(1,1)}(x) dx, i = 0, 1, \ldots, N+1, \tag{4.6}$$

lead to the Lobatto 3C method. The stage of this method is $N+2$. Similar method with $N = 2$ is given in [4] and it is proven that this method is algebraically stable. It is possible to obtain other schemes by choise of the c_i.

REFERENCES

[1] C. Canuto, M. Hussaini, A. Quateroni, T. Zang. *Spectral Methods in Fluid Dinamics.* Springer Verlag, Berlin, 1988, pp. 557.

[2] T. Cirulis. *Proc. Latv. Acad. Sci.* **B**, vol.52, No. 5(598), 1998, 234–244.

[3] E.Haier, P.Norsett, G. Wagner. *Solwing Ordinary Differential Equations, Vol.1: Nonstiff Problems.* Springer Verlag, Berlin, 1993, pp. 528.

[4] E. Haier, G. Wanner. *Solwing Ordinary Differential Equations, Vol.2: Stiff and Differential-Algebraic Problems.* Springer Verlag, Berlin, 1996, pp. 614.