COMPUTATIONAL METHODS IN APPLIED MATHEMATICS, Vol. 8 (2008), No. 2, pp. 116–129

ANALYTIC ITERATIVE PROCESSES AND NUMERICAL ALGORITHMS FOR STIFF PROBLEMS

B. V. FALEICHIK[1]

Abstract — The goal of the research is to construct practicable numerical algorithms for stiff systems of ordinary differential equations which let you increase the accuracy of the approximate solution without decreasing the length of the time interval. To achieve this goal, we have constructed a family of new iterative analytic processes generalising the Picard process. For a basic representative of this family, we demonstrate its better convergence properties on a scalar linear problem in comparison with the classical Picard process. For the general form of such iterative processes, we discuss their connection with existing methods for operator equations and propose a method for choosing their parameters. The efficiency of this parameter determination method is justified with a numerical experiment. In conclusion we propose a general approach to the construction of numerical algorithms which is based on the discretisation of the constructed iterative analytic processes.

2000 Mathematics Subject Classification: 65L05, 65L07.

Keywords: stiff problems, stabilization principle, the Picard process, Runge — Kutta methods, stability function, operator equations.

Introduction

At the present time the most popular choice for a numerical solution of stiff nonlinear systems of ordinary differential equations (ODEs) is implicit Runge — Kutta methods. These methods combine high order and prominent stability properties, but as most implicit methods, they suffer from computational complexity caused by the necessity to solve big systems of nonlinear equations at every step. Furthermore, almost with all existing stepwise numerical methods in order to increase the accuracy we have to decrease the timestep. It does not cause much trouble when the whole interval of numerical investigation is relatively small, but if this interval is very large or its length is not fixed beforehand, then small stepsizes are unsuitable. When the required accuracy is high, the chosen stepsize can be inconsistent with the length of the whole time interval.

The aim of our work is to construct numerical methods for stiff nonlinear ODEs which allow to increase the accuracy of the approximate solution without decreasing the length of the time interval[2] and which are more easy to implement than implicit methods.

The construction of such numerical algorithms is based on the discretization of special analytic iterative processes with improved convergence properties on stiff problems.

[1] *Belarusian State Unversity*, Nezavisimosti ave. 4, 220030 Minsk, Belarus. E-mail: faleichik@bsu.by

[2] It is necessary to understand that we are not going to refuse stepsize control. We want to use *natural* stepsizes and to correct the obtained approximate solution without rejecting it.

1. Analytic iterative processes

Consider an initial value problem for the system of ODEs

$$u'(x) = \varphi(x, u(x)), \quad u(x_0) = u_0, \quad x \in [x_0, x_0 + h], \tag{1.1}$$

where $u : [x_0, x_0 + h] \to \mathbb{R}^n$, $u_0 \in \mathbb{R}^n$, $\varphi : [x_0, x_0 + h] \times \mathbb{R}^n \to \mathbb{R}^n$. It is assumed that the function φ is continuous and satisfies the Lipschitz condition

$$\|\varphi(x, u_1) - \varphi(x, u_2)\| \leqslant L_0\|u_1 - u_2\| \quad \forall\, x \in [x_0, x_0 + h],\ u_1, u_2 \in \mathbb{R}^n.$$

Here $\|\cdot\|$ is a norm in $\mathbb{R}^n$. To simplify the notation, represent (1.1) in the form of the Volterra integral equation on the interval $[0, 1]$

$$v(x) = \int_0^x f(z, v(z))dz, \quad x \in [0, 1], \tag{1.2}$$

$$v = \mathcal{S}F(v) = \mathcal{T}(v), \quad v \in V^n, \tag{1.2'}$$

where $f(z, v) = h\varphi(x_0 + zh, u_0 + v)$, V^n is Banach space $C([0, 1], \mathbb{R}^n)$ or $L_2([0, 1], \mathbb{R}^n)$; $\mathcal{S}$ is the integral operator, $\mathcal{S}v(x) = \int_0^x v(z)dz$; $F : V^n \to V^n$, $(F(v))(z) = f(z, v(z))$.

In $C([0, 1], \mathbb{R}^n)$, we use the norm

$$\|v\|_\infty = \max_{x\in[0,1]} \|v(x)\|,$$

where $\|\cdot\|$ is an arbitrary norm in $\mathbb{R}^n$. In $L_2(C[0, 1], \mathbb{R}^n)$, consider

$$\|v\|_2 = \left(\int_0^1 \|v(x)\|^2 dx\right)^{1/2},$$

where $\|\cdot\|$ is the Euclidean norm. The exact solution of (1.2′) is denoted by v^*. Note that the function f satisfies the Lipschitz condition with a constant $L = L_0h$. This property guarantees the existence and uniqueness of v^* and is used without further mentioning.

1.1. The Picard process. Now consider the classic Picard process for (1.2′):

$$v_{k+1} = \mathcal{T}(v_k), \quad k = 0, 1, 2, \ldots \tag{1.3}$$

As is known, this process converges to v^* whenever L is finite. That's why its discretization seems to be suitable for our purposes (recall that we want to raise the accuracy of the numerical approximation without decreasing the length of the time interval). Unfortunately the Picard process has a significant drawback. It appears that on stiff problems the norm of this process error

$$\varepsilon_k = v^* - v_k$$

does not decrease monotonously even if the initial approximation is close to the exact solution. To illustrate this, consider the following analog of the test equation (1.1.1) from [1, p. 16]:

$$v(x) = \int_0^x \big(\lambda v(z) + g'(z) - \lambda g(z)\big)dx, \quad v \in C[0, 1]. \tag{1.4}$$

Take $\lambda = -50$, $g(z) = 0.5 \sin 2\pi z$ and make 12 Picard iterations starting from $v_0(z) \equiv 0$. The plots of $v^*(x) = g(x)$ (gray) and $v_{12}(x)$ (black) and the maximum norm of the error ε_{12} are shown in Fig. 1.1. We see that $\|\varepsilon_{12}\|_\infty \approx 10^{11}$ while $\|\varepsilon_0\|_\infty = 0.5$, which is unacceptable. Such behavior of the error can be explained by the fact that the mapping $\mathcal{T}$ is a contraction not in the norm $\|\cdot\|_\infty$ but in the equivalent norm $\|\cdot\|_\infty^L$,

$$\|v\|_\infty^L = \max_{x\in[0,1]} \|e^{-Lx} v(x)\|,$$

(see [2, p. 26], [3, p. 12]). So we have $\|\varepsilon_{k+1}\|_\infty^L < \|\varepsilon_k\|_\infty^L$ for all $k \geqslant 1$, which obviously does not mean that $\|\varepsilon_{k+1}\|_\infty < \|\varepsilon_k\|_\infty$. Now we proceed with a construction of more robust analytic iterative processes.

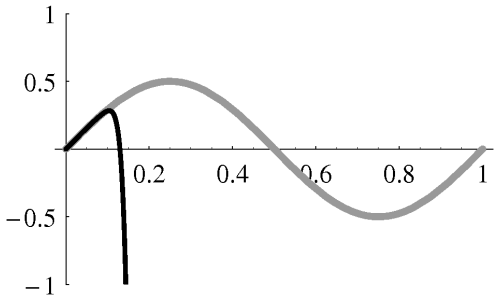

Fig. 1.1. 12th Picard approximation for problem (1.4), error= 1.03×10^{11}

1.2. Stabilization iterative processes (SIPs). The idea is to introduce an auxiliary variable — fictitious time $t \in [0, +\infty)$ — and describe a continuous process of approximation for the solution of (1.2′). More precisely, we define a mapping

$$Y : [0, +\infty) \to V^n$$

such that $Y(0) = v_0$ and

$$\|Y(t) - v^*\|_{V^n} \xrightarrow[t\to\infty]{} 0. \tag{1.5}$$

Then by approximation of $Y(t_k)$, $t_k \xrightarrow[k\to\infty]{} \infty$, we get an iterative process for the solution of the original problem. This approach is well known in numerical analysis and its name can be translated from Russian as "stabilization principle" [4; 5, p. 258].

We define the mapping Y as a solution of the following equation (see (1.2)):

$$\frac{\partial}{\partial t} y(x,t) = -y(x,t) + \int_0^x f(z, y(z,t))dz, \tag{1.6}$$

which can be represented in the operator form

$$Y'(t) = -Y(t) + \mathcal{S}F(Y(t)) = \mathcal{F}(Y(t)). \tag{1.6′}$$

We call here $y : [0,1]\times[0,+\infty) \to \mathbb{R}^n$, $y(x,t) = (Y(t))(x)$. This equation the stabilization equation. It is proved that

i) for any initial condition $Y(0) = v_0 \in C([0,1], \mathbb{R}^n)$ there exists a unique solution of (1.6′) belonging to $C([0,1] \times [0,+\infty), \mathbb{R}^n)$;

ii) if Y is a solution of (1.6′), then (1.5) holds, where v^* is the exact solution of (1.2′), $\|\cdot\|_{V^n}$ is $\|\cdot\|_\infty$ or $\|\cdot\|_2$ (to prove this for the norm $\|\cdot\|_\infty$ the differentiability of f is required).

Now consider an initial value problem for the stabilization equation (1.6′):

$$Y'(t) = \mathcal{F}(Y(t)), \quad Y(0) = v_0. \tag{1.6″}$$

The approximate solution of (1.6″) by means of an explicit Runge — Kutta (RK) method with a constant timestep τ can be represented in the form

$$v_{k+1} = \mathcal{M}(v_k), \tag{1.7a}$$

where the operator $\mathcal{M}$ is defined as

$$\mathcal{M}(v) = v + \tau \sum_{i=1}^{s} b_i \mathcal{K}_i(v), \tag{1.7b}$$

$$\mathcal{K}_i(v) = \mathcal{F}\left(v + \tau \sum_{j=1}^{i-1} a_{ij} \mathcal{K}_j(v)\right), \quad i = \overline{1, s}. \tag{1.7c}$$

Here $v_k \approx Y(t_k)$, $t_{k+1} = t_k + \tau$, s is the number of stages of base Runge — Kutta method, b_i and a_{ij} are the coefficients of this method. Ae shall call an iterative process of the form (1.7) the stabilization iterative process (SIP).

Generalized Picard process. The simplest iterative process of the form (1.7a) is obtained when we apply the implicit Euler method to the stabilization equation:

$$v_{k+1} = v_k + \tau \mathcal{F}(v_k) = (1 - \tau)v_k + \tau\, \mathcal{S}F(v_k). \tag{1.8}$$

It's clear that for $\tau = 1$ the process (1.8) turns into a classic Picard process.

Theorem 1.1. *The sequence* $\{v_k\}_{k=0}^{\infty}$ *of the generalized Picard approximations* (1.8) *converges to* v^* *in the norms* $\|\cdot\|_\infty$ *and* $\|\cdot\|_2$ *for all* τ *in* $(0, 1]$.

It can be noticed that the process (1.8) is a representative of the two-layer iterative scheme from [5, p. 500, (2)] with $B = \mathcal{I}$, $A = \mathcal{I} - \mathcal{S}F$ and $f = 0$. In contrast to the known result on the convergence of such schemes [5, p. 502, Th. 1] Theorem 1.1 does not require that the additional conditions [5, p. 501, (4), (5)] hold. Moreover, the above-mentioned theorem from [5] deals with Hilbert space and is not applicable in the case of the C-norm $\|\cdot\|_\infty$.

Now we are going to demonstrate the advantages that process (1.8) may have in comparison with the classic Picard process. Consider the following generalization of model equation (1.4):

$$v = \lambda \mathcal{S}(v + a), \quad \lambda \in \mathbb{R}, \quad a \in V. \tag{1.9}$$

The processes to compare are the Picard process

$$v_{k+1} = \lambda \mathcal{S}(v_k + a) \tag{1.10}$$

and the generalized Picard process

$$v_{k+1} = (1 - \tau)v_k + \tau\lambda \mathcal{S}(v_k + a). \tag{1.11}$$

The convergence properties of these processes are determined by the norms of the corresponding linear operators

$$\mathcal{R}_\lambda = \lambda\mathcal{S} \quad \text{for (1.10)}, \tag{1.12}$$

$$\mathcal{R}_{\lambda,\tau} = (1-\tau)\mathcal{I} + \lambda\tau\mathcal{S} \quad \text{for (1.11)}. \tag{1.13}$$

Here $\mathcal{I} : V^n \to V^n$ is the identity mapping. In general, the computation of $\|\cdot\|_2$ is easier than of $\|\cdot\|_\infty$, so we shall use the linear operator norm induced by $\|\cdot\|_2$. The following result for the generalized Picard process is proved.

Theorem 1.2. *For any $\lambda < 0$ there exists $\tau_\lambda > 0$, such that for all τ from $(0, \tau_\lambda)$*

$$\|\mathcal{R}_{\lambda,\tau}\| < 1,$$

holds, where $\mathcal{R}_{\lambda,\tau}$ is defined in (1.13), *$\|\cdot\|$ is a linear operator norm induced by $\|\cdot\|_2$.*

This result means that for all negative values of λ with a proper choice of τ for the process (1.11) we have $\|\varepsilon_{k+1}\|_2 < \|\varepsilon_k\|_2 \; \forall\, k \geqslant 1$. On the other hand, the reader will easily prove that for $\lambda > 2$ we have

$$\|\mathcal{R}_{\lambda,\tau}\| > 1 \quad \forall\tau > 0.$$

It is not a big problem though, since we are targeted to stiff problems that are characterized by $\lambda \ll 0$.

As for the classic Picard process (1.10), it is trivial to prove that if $|\lambda| > \sqrt{3}$, then $\|\mathcal{R}_\lambda\| > 1$. Therefore the generalized Picard process (1.8) is more preferable than the classic one at least on the model equation (1.9).

1.3. How to choose the parameters? Now we proceed to the problem of choosing the values of the parameters τ, b_i and a_{ij} for the SIP (1.7). The behavior of the iterative process strongly depends on problem (1.2′), so first consider the model equation (1.9).

1.3.1. Linear problem. Applying a SIP to (1.9) we see (similarly to (1.11)) that the Lipshitz constant of the corresponding operator $\mathcal{M}$ is equal to the norm of the linear operator

$$\mathcal{R}^{[s]}_{\lambda,\tau} = R_s(\tau(\lambda\mathcal{S} - \mathcal{I})), \tag{1.14}$$

where

$$R_s(z) = \sum_{i=0}^{s} \alpha_i z^i = 1 + z\sum_j b_j + z^2 \sum_{j,k} b_j a_{jk} + z^3 \sum_{j,k,l} b_j a_{jk} a_{kl} + \dots \tag{1.15}$$

is a classic stability function of the base RK method [1, p. 52, 86]. Our goal now is to select the values of the free parameters depending on λ. More precisely, we first determine the coefficients α_i of the stability function and then construct a corresponding RK method.

The convergence rate of the process depends on the norm of operator (1.14). Since it is difficult to calculate this norm for $s \geqslant 2$, we have to use an heuristic and consider the following function instead:

$$K_0^{[s]}(\lambda,\tau) = \|\mathcal{R}^{[s]}_{\lambda,\tau} v_0\|_2, \tag{1.16}$$

where $v_0 \in L_2[0,1]$ is a function with unity norm. It's clear that the condition $K_0^{[s]}(\lambda,\tau) < 1$ is necessary for $\|\mathcal{R}^{[s]}_{\lambda,\tau}\| < 1$. Therefore we shall minimize the values of $K_0^{[s]}(\lambda,\tau)$. Two kinds of such optimization were considered. The first one is simple:

$$\left(K_0^{[s]}(\lambda,\tau)\right)^2 \to \min$$

and the second is

$$-\frac{1}{\mu}\int\limits_{\mu}^{0}\left(K_0^{[s]}(\lambda,\tau)\right)^2 d\lambda \to \min \tag{1.17}$$

for some $\mu < 0$. Further we will consider the latter one since it appeared more effective. Besides (1.17) we require the condition of first order for base RK method to hold

$$\alpha_1 = \sum_{i=1}^{s} b_i = 1. \tag{1.18}$$

Combining (1.16), (1.15) and (1.14) we obtain

$$-\frac{1}{\mu}\int\limits_{\mu}^{0}\left(K_0^{[s]}(\lambda,\tau)\right)^2 d\lambda = \sum_{i,j=0}^{s}\alpha_i\alpha_j\tau^{i+j}p_{ij}(\mu,\tau), \tag{1.19}$$

where

$$p_{ij}(\mu) = -\int\limits_{\mu}^{0}\langle\mathcal{A}^i v_0, \mathcal{A}^j v_0\rangle d\lambda, \tag{1.20a}$$

$$\mathcal{A} = \lambda\mathcal{S} - \mathcal{I}, \quad \langle u, v\rangle = \int\limits_{0}^{1} u(x)v(x)dx. \tag{1.20b}$$

Minimizing the quadratic functional (1.19) with respect to α_i, τ and taking into account condition (1.18) after some transformations we get the following expressions for the parameters:

$$\tau(\mu) = \frac{q_1(\mu)}{r(\mu)}, \tag{1.21a}$$

$$\alpha_i(\mu) = \frac{q_i(\mu)r(\mu)^{i-1}}{q_1(\mu)^i}, \quad i = \overline{2,s}. \tag{1.21b}$$

Here

$$r(\mu) = \det P(\mu), \quad q_i(\mu) = \det P_i(\mu), \tag{1.22}$$

where $P(\mu) = \{p_{ij}(\mu)\}_{i,j=1}^{s}$, $P_i(\lambda)$ is the matrix $P(\lambda)$ with the i-th column replaced by the vector $-(p_{10}(\mu),\ldots,p_{s0}(\mu))^\top$.

Having found α_i via (1.21b), we express the coefficients of the base RK method using (1.15). This procedure is ambiguous since the number of unknowns b_i and a_{ij} for $s > 1$ is greater than s, therefore we have to fix some parameters. For example, by putting $b_s = 1$ and $b_i = 0\ \forall\, i < s$ we can save several calculation operations.

It is necessary to emphasize that the described approach is an heuristic one and generally does not guarantee the fastest convergence. To prove its efficiency, we are going to perform a computational experiment. Recall the test equation (1.4). We apply one- two- and three-stage SIPs to this problem taking $\mu = \lambda = -50$ in expressions (1.21) to determine the free parameters.

Before we proceed to the results of the experiment it is reasonable to describe the scheme of parameters determination more thoroughly. Let us do it for the case of $s = 3$. We have seven unknowns:

$$\tau, a_{21}, a_{31}, a_{32}, b_1, b_2, b_3.$$

We first compute the matrix $P(\mu)$ from (1.22) using (1.20), then find the expressions for $r(\mu)$ and $q_i(\mu)$ (we used *Mathematica* for this purpose). These expressions are rather cumbersome, so we do not give them here. Taking $\mu = -50$ in (1.21), we get $\tau = 0.221261838364$, $\alpha_2 = 0.438692861462$, $\alpha_3 = 0.075717822473$ (recall that $\alpha_1 = 1$ due to (1.18)). Then from (1.15) we get

$$1 = b_1 + b_2 + b_3, \quad \alpha_2 = b_2 a_{21} + b_3(a_{31} + a_{32}), \quad \alpha_3 = b_3 a_{32} a_{21}.$$

Taking $b_1 = b_2 = 0$, $b_3 = 1$, we obtain

$$a_{31} = \alpha_2 - \alpha_3/a_{21}, \quad a_{32} = \alpha_3/a_{21}. \tag{1.23}$$

Finally we choose $a_{21} = 1/3$ and from (1.23) get $a_{31} = 0.21153939404$, $a_{32} = 0.2271534674$. All unknown parameters for the three-stage SIP are determined now.

The results of the calculations are shown in Fig. 1.2. Note that the computational complexity in all cases is equal to 12 evaluations of $\mathcal{F}$, same as for the Picard process in Fig. 1.1. We see that our processes give much more adequate approximations and their accuracy increases with increasing number of stages.

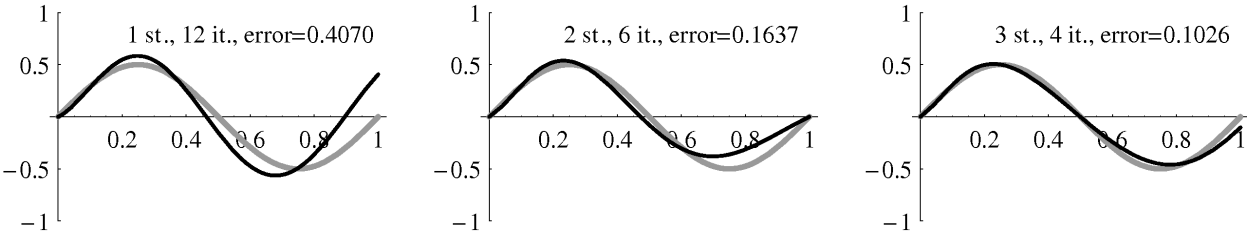

Fig. 1.2. One-, two- and three-stage SIPs for (1.4) with parameters determined by (1.21)

1.3.2. General nonlinear case. In the case of a nonlinear system of ODEs, we suggest to use the above approach by establishing a correspondence between the nonlinear problem and the scalar model problem (1.9). This can be done by using the eigenvalues of Jacobi matrix for the right-hand side of the initial equation[1]. The following example should make this idea clear.

Consider an initial value problem for the Van-der-Pol equation

$$\begin{cases} u_1'(x) = u_2(x), \\ u_2'(x) = 20\big((1 - u_1^2(x))u_2(x) - u_1(x)\big), \end{cases} \quad u_1(0) = 2, \quad u_2(0) = 0, \quad x \in [0, 0.2]. \tag{1.24}$$

[1]We realize that for a large system of ODEs the exact computation of the leading eigenvalue of the Jacobi matrix is very expensive. In this case, one should use more cheap estimates. One of the possible approaches is described in [7, p. 34] where the problem of stiffness detection is discussed. We can also try to use the Rayleigh quotient for the Jacobi matrix to estimate the leading eigenvalue, or fractions like $\|F(v_k) - F(v_{k-1})\|/\|v_k - v_{k-1}\|$ to estimate the Lipschitz constant. Anyway, the question of robust practical determination of SIP's parameters requires further theoretical investigation and numerical testing.

For this problem we have the Jacobi matrix

$$\frac{\partial\varphi}{\partial u}(x,u) = \begin{pmatrix} 0 & 1 \\ -20(1+2u_1(x)u_2(x)) & 20(1-u_1^2(x)) \end{pmatrix}.$$

To get the value of μ, we compute $\frac{\partial\varphi}{\partial u}(x_0,u_0)$ and find the eigenvalue with a negative real part of the largest magnitude: $\lambda_0 = -59.664794$. Finally, we should take into account the scaling:

$$\mu = \lambda_0 h \approx -11.933.$$

Using this value in (1.21), we find the values of the free parameters just like in the scalar linear case. Both components of the approximations obtained with corresponding SIPs for problem (1.24) in comparison to the Picard process are shown in Fig. 1.3. We see that the result is similar to the previously discussed experiment in the linear case.

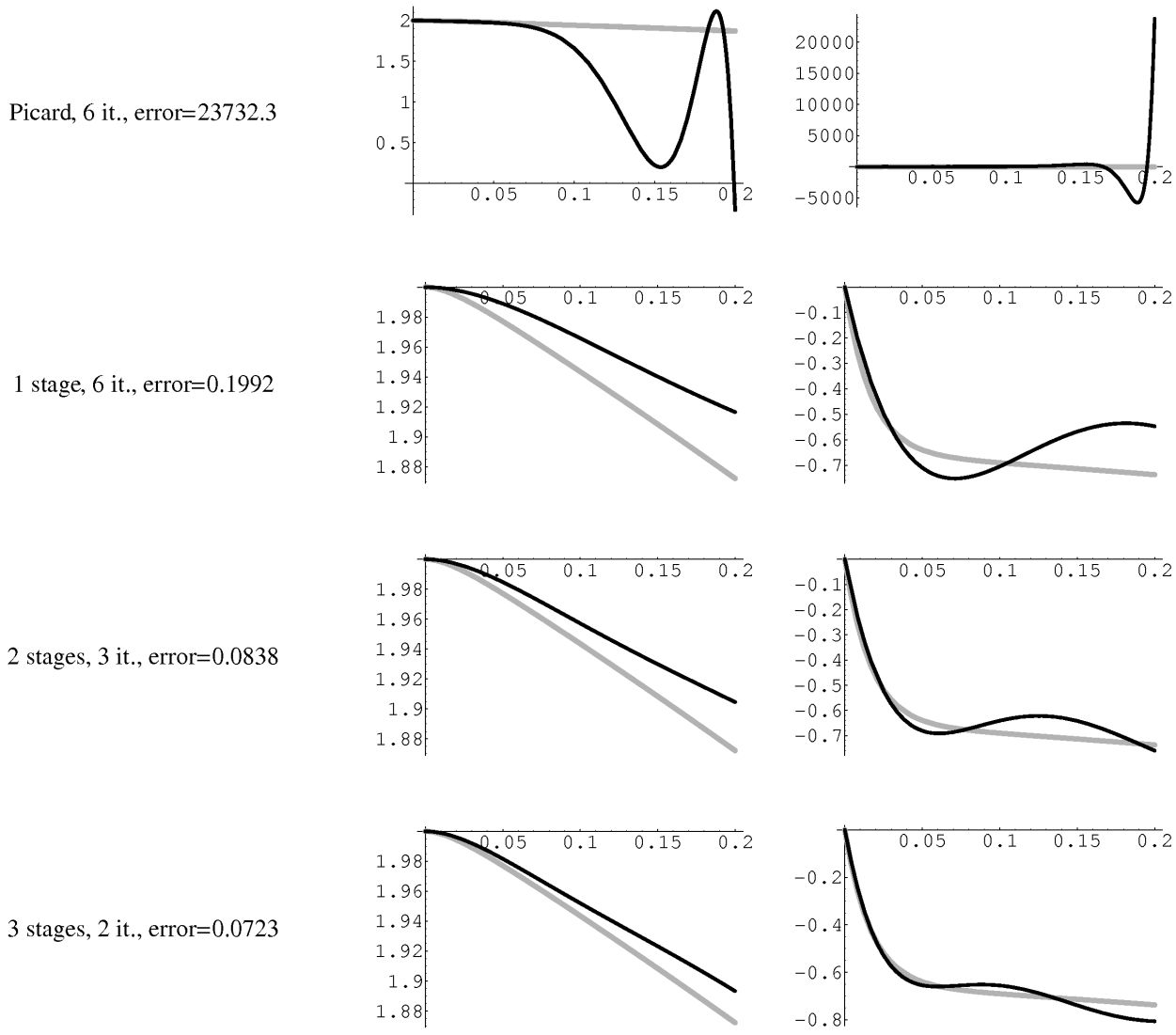

Fig. 1.3. The Picard process, one-, two- and three-stage SIPs for (1.24) with parameters determined by (1.21a), (1.21b)

2. Numerical iterative processes

In order to construct numerical iterative processes, we simply apply the previously constructed analytic processes to the finite-dimensional version of problem (1.2′). To introduce such a discrete problem, consider the subspace

$$U_m^n = \left\{ \sum_{l=1}^{m} \alpha_l \varphi_l^m \;\middle|\; \alpha_l \in \mathbb{R}^n, \quad \varphi_l^m : [0,1] \to \mathbb{R} \right\} \subset V^n. \tag{2.1}$$

The basis functions φ_l^m are assumed to be linearly independent. Let

$$\Pi_m : V^n \to U_m^n \tag{2.2}$$

be the projection onto U_m^n. We assume Π_m to be uniquely determined by the set of linear functionals

$$Q_l^m : V^n \to \mathbb{R}^n, \tag{2.3}$$

$$\Pi_m v = \sum_{l=1}^{m} Q_l^m(v) \varphi_l^m \quad \forall v \in V^n. \tag{2.4}$$

Substituting $\Pi_m F$ for F in (1.2′), we obtain the required finite-dimensional problem

$$v = \mathcal{S}\Pi_m F(v), \quad v \in \mathcal{S}U_m^n = V_m^n, \tag{1.2′$_m$}$$

where V_m^n is a subspace analogous to (2.1) with the basis

$$\{\psi_l^m\} = \{\mathcal{S}\varphi_l^m\}_{l=1}^m.$$

Applying the SIP (1.7a) to the discrete problem (1.2′$_m$), we get an iterative process suitable for numerical implementation. Note that the only difference between the analytic and discrete versions of the SIP is in the mapping $\mathcal{F}$. For discrete processes we have

$$\mathcal{F} = \mathcal{F}^m = \mathcal{S}\Pi_m F - \mathcal{I}.$$

We denote the exact solution of (1.2′$_m$)

$$v^m = \sum_{l=1}^{m} \eta_l^m \psi_l^m. \tag{2.5}$$

To define the numerical algorithm, we should describe the procedure of transition from the k-th approximation

$$v_k^m = \sum_{l=1}^{m} \eta_{k,l} \psi_l^m \tag{2.6}$$

to the $(k+1)$-th one with the new coefficients $\eta_{k+1,l}$. This algorithm is given below.

Input: $\eta_k = \{\eta_{k,l}\}_{l=1}^m \in \mathbb{R}^{mn}$.
Output: $\eta_{k+1} = \{\eta_{k+1,l}\}_{l=1}^m \in \mathbb{R}^{mn}$.
Intermediate variables: $\gamma = \{\gamma_l\}_{l=1}^m \in \mathbb{R}^{mn}$,
$\kappa_i = \{\kappa_{i,l}\}_{l=1}^m \in \mathbb{R}^{mn}, \quad i = \overline{1,s}$.

$$\left.\begin{aligned} &\gamma \leftarrow \eta_k + \tau \sum_{j=1}^{i-1} a_{ij}\kappa_j, \\ &\kappa_{i,l} \leftarrow Q_l^m\left(F\left(\textstyle\sum_{p=1}^m \gamma_p \mathcal{S}\varphi_p^m\right)\right) - \gamma_l, \quad l = \overline{1,m}, \end{aligned}\right] \quad i = \overline{1,s}; \tag{2.7a}$$

$$\eta_{k+1} \leftarrow \eta_k + \tau \sum_{i=1}^{s} b_i \kappa_i. \tag{2.7b}$$

To define the numerical algorithm of type (2.7), we formally need to choose the basis functions φ_l^m and the rule for evaluating functionals (2.3). Before to discuss concrete numerical algorithms we present the basic theoretical result for the convergence of the discreet iterative process for $(1.2'_m)$.

Theorem 2.1. *Let $V^n = C([0,1], \mathbb{R}^n)$ and the following conditions are satisfied:*
— the initial differential equation (1.1) is autonomous,
— the sequence of operators $\{\Pi_m\}$ is uniformly bounded,
— the sequence of operators $\{\mathcal{S}\Pi_m\}$ converges pointwise to $\mathcal{S}$,
— for all $m \geqslant m_0$

$$\|\mathcal{S}\Pi_m\| L < 1. \tag{2.8}$$

holds. Then

i) for all $m \geqslant m_0$ the discreet generalized Picard process

$$v_{k+1}^m = (1-\tau)v_k^m + \tau \mathcal{S}\Pi_m F(v_k^m)$$

converges to the function $v^m = \mathcal{S}\Pi_m F(v^m)$ if $\tau \in (0,1]$;

ii) the sequence $\{v^m\}$ converges uniformly to $v^* = \mathcal{S}F(v^*)$:

$$\|v^m - v^*\|_\infty \xrightarrow[m\to\infty]{} 0.$$

This result should be considered as a preliminary one because of the condition (2.8) which becomes very restrictive for stiff problems with very large values of the Lipschitz constant. Our numerical experiments have shown that this condition is too strong and in practice discrete iterative processes do converge when it is violated (see Table 1 below). Note that as in the "continuous" case of the SIP the convergence of general multistage discrete SIPs have not been proved so far.

2.1. Discretization by means of interpolation. The most straightforward approximation technique of the form (2.4) is the polynomial interpolation in the nodes $\xi_l^m \in [0,1]$, $l = \overline{1,m}$:

$$\varphi_l^m(\xi) = \prod_{p \neq l} \frac{\xi - \xi_p^m}{\xi_l^m - \xi_p^m}, \quad Q_l^m(v) = v(\xi_l^m). \tag{2.9}$$

In this case, the space U_m^n (2.1) is a space of algebraic polynomials of a degree not greater than $m-1$, and $V_m^n = \mathcal{S}U_m^n$ is a space of polynomials P of a degree not greater than m such that $P(0) = 0$. Using (2.9) in the second line of (2.7a), we obtain

$$\kappa_{i,l} \leftarrow f\left(\xi_l^m, \sum_{p=1}^m \gamma_p \psi_p^m(\xi_l^m)\right) - \gamma_l.$$

2.1.1. The connection with implicit RK methods. Since the approximate solution v^m satisfies $(1.2'_m)$, with (2.9) we have

$$\frac{d}{dx}v^m(\xi_l^m) = (\Pi_m F(v^m))(\xi_l^m) = \sum_{p=1}^{m} f(\xi_p^m, v^m(\xi_p^m))\varphi_p^m(\xi_l^m) = f(\xi_l^m, v^m(\xi_l^m)). \tag{2.10}$$

This yields that the unknown coefficients η_l^m from (2.5) satisfy the following system of equations:

$$\eta_l^m = f\big(\xi_l^m, \sum_{p=1}^{m} \eta_p^m \psi_p^m(\xi_l^m)\big), \quad l = \overline{1,m}. \tag{2.11}$$

In general, (2.11) is a nonlinear system of equations of dimension mn and the iterative process (2.7) can be considered as a means for solving this system. Moreover, the reader can see that the structure of (2.11) resembles the system which is solved in implicit Runge — Kutta methods. To see this connection from a different point of view, notice that the approximate solution of the initial ODE (1.1)

$$u^m(x) = u_0 + v^m\left(\frac{x - x_0}{h}\right) \tag{2.12}$$

which corresponds to v^m, is a polynomial of degree m satisfying

$$u^m(x_0) = u_0, \quad u^m(x_0 + \xi_l^m h) = \varphi(x_0 + \xi_l^m h, u^m(x_0 + \xi_l^m h)).$$

This means that u^m is a collocation polynomial for (1.1) [6, p. 220]. As is known, collocation methods are implicit RK methods, hence, in fact, our algorithm (2.7) in the case of (2.9) presents an alternative realization of some implicit RK method. Nonlinear systems in implicit RK methods are usually solved using the Newton method with a constant Jacobi matrix. Every iteration of this method requires solving a linear system of equations. On the other hand, our algorithm (2.7) is similar to the fixed-point iteration (we have this in the trivial case of the discrete Picard method) and do not require solving intermediate equations. Moreover, we can provide our algorithm with additional useful features which will be discussed in the following paragraph.

2.1.2. Features of numerical implementation. The essence of the modifications we are going to suggest now consists in making the iterative process of solving the discrete problem $(1.2'_m)$ more closely related to the underlying differential equation. The first and most important modification exploits the idea used in multigrid methods for integral and partial differential equations.

Multigrid-like modification. As was claimed at the very beginning, the numerical methods we develop should be capable of increasing accuracy without decreasing the stepsize. That's why it is necessary to be able to raise the level of the approximation during the discrete iterative process. Consider an approximate solution of the form (2.6), $v_k^m \in V_m^n$. To continue the iterative process on a "more precise" subspace $V_{\widehat{m}}^n$, $\widehat{m} > m$, we need to find a set of coefficients $\{\widehat{\eta}_{k,l}\}_{l=1}^{\widehat{m}}$ such that

$$v_k^{\widehat{m}}(x) = \sum_{l=1}^{\widehat{m}} \widehat{\eta}_{k,l}\psi_l^{\widehat{m}}(x) = v_k^m(x) \quad \forall x \in [0,1]. \tag{2.13}$$

Differentiating the left-hand side of (2.13), we have

$$\frac{d}{dx}v_k^{\widehat{m}}(x)\Big|_{x=\xi_p^{\widehat{m}}} = \frac{d}{dx}\left(\sum_{l=1}^{\widehat{m}}\widehat{\eta}_{k,l}\psi_l^{\widehat{m}}(x)\right)\Big|_{x=\xi_p^{\widehat{m}}} = \sum_{l=1}^{\widehat{m}}\widehat{\eta}_{k,l}\varphi_l^{\widehat{m}}(\xi_p^{\widehat{m}}) = \widehat{\eta}_{k,p}.$$

So finally we get

$$\widehat{\eta}_{k,p} = \sum_{l=1}^{m}\eta_{k,l}\varphi_l^m(\xi_p^{\widehat{m}}), \quad p = \overline{1,\widehat{m}}. \tag{2.14}$$

After this transformation we put $m = \widehat{m}$ and continue the iterative process until the required accuracy is reached. If a more accurate approximation is needed, then we can repeat the above procedure of transition to a "more accurate" subspace.

Now let's consider the results of the computational experiment (see Table 1). We have tested two versions of the discrete analog of the analytic three-stage SIP which we used earlier for problem (1.24) (see Fig. 1.3). The first version is multigrid-like and the second uses one fixed grid on every iteration. For each iteration of these processes, we display the current number of collocation nodes m_k, the norm of the error $\|\varepsilon_k\|_\infty$ and the computational complexity measured in the number of evaluations of the right-hand side of the ODE. The collocation nodes ξ_l^m for both processes are the nodes of Radau polynomial $P(\xi) = \frac{d^{m-1}}{dx^{m-1}}\big(\xi^{m-1}(\xi-1)^m\big)$. This makes our algorithms equivalent to implicit the Radau methods.

Computational experiment: comparison of multigrid and fixed-grid discrete iterative processes

	Multigrid			Fixed grid		
k	m_k	$\|\varepsilon_k\|$	N_f	m_k	$\|\varepsilon_k\|$	N_f
1	3	0.4068620	9	10	0.4068384	30
2	4	0.0754135	21	10	0.0723049	60
3	5	0.0321644	36	10	0.0216543	90
4	6	0.0137002	54	10	0.0067173	120
5	7	0.0064819	75	10	0.0018221	150
6	8	0.0024815	99	10	0.0005156	180
7	9	0.0008777	126	10	0.0002904	210
8	10	0.0002960	156	10	0.0001182	240
9	10	0.0001404	186	10	0.0000650	270
10	10	0.0000752	216	10	0.0000565	300
11	10	0.0000589	246	10	0.0000549	330
12	10	0.0000549	276	10	0.0000552	360
13	10	0.0000551	306	10	0.0000553	390
14	11	0.0000218	339	10	0.0000553	420
15	11	0.0000133	372	10	0.0000553	450
16	11	0.0000106	405	10	0.0000553	480
17	11	9.59×10^{-6}	438	10	0.0000553	510

As we see, a multigrid-like modification gives a more accurate and less expensive result than simple fixed-grid iterations. In this example we manually selected the number of iterations on each level of discretization. The general strategy of choosing this number is a subject of further research.

We should mention that we have described only the most simple multigrid-like modification of our algorithm. The analytic form of approximate solutions that we use allows us to compute the residuals

$$\frac{d}{dx}v_k^m(\xi) - f(\xi, v_k^m(\xi))$$

at any point $\xi \in [0,1]$. This property with some additional details makes it possible to construct more sophisticated algorithms analogous to multigrid methods for the integral equations [2].

The next two features are based on considering the approximate solution (2.12) not just as a function defined locally on $X = [x_0, x_0 + h]$, but as a globally-defined function which can be close enough to the exact solution outside the interval X.

Floating stepsize. This may be an alternative to a very important component of all ODE solvers — the stepsize control technique. The idea is to change the stepsize h within the iterative process without loss of the currently achieved approximation (2.6). To perform this operation, we need to recalculate the coefficients $\eta_{k,l}$ and substitute the new value of stepsize $\hat{h}$ for h in the expression for the function f (see (1.2), (1.2′)). To be definite, assume

$$\hat{h} = \delta h,$$

then the "scaled" approximation $\hat{v}_k^m$ should satisfy

$$\hat{v}_k^m(x) = \sum_{l=1}^{m} \widehat{\eta}_{k,l}\psi_l^m(x) = \sum_{l=1}^{m} \eta_{k,l}\psi_l^m(\delta x) = v_k^m(\delta x) \quad \forall x \in [0,1].$$

Now differentiate the last expression just like we did before for (2.13) and obtain

$$\widehat{\eta}_{k,p} = \delta \sum_{l=1}^{m} \eta_{k,l}\varphi_l^m(\delta\xi_p^m), \quad p = \overline{1,m}. \tag{2.15}$$

This procedure can be used in two different cases. When the convergence rate is too small (this fact can easily be monitored), we take some $\delta < 1$, perform transformation (2.15), put $h = \hat{h}$ and continue the process. In the converse case, if the process converges very fast, it makes sence to increase the timestep by taking $\delta > 1$.

Choice of the initial approximation. The approach to the choice of the initial approximation we are going to use is analogous to that described in [7, p. 141] for collocation RK methods. Since (2.6) can be evaluated outside $[0,1]$, we can use the "extrapolated" approximate solution from the previous step as the initial approximation at the new step. More precisely, let

$$u_0^m(x) = u_0 + \tilde{v}^m\left(\frac{x - x_0}{h_0}\right), \tag{2.16}$$

where $\tilde{v}^m = \sum_{l=1}^{m} \tilde{\eta}_l\psi_l^m$, be the approximate solution on the interval $[x_0, x_1]$, $x_1 - x_0 = h_0$. To start the iterative process on the next interval $[x_1, x_1 + h_1]$ with the new stepsize $h_1 = \delta h_0$, we choose such coefficients $\eta_{0,l}$ that the corresponding intial approximation

$$\tilde{u}_1^m(x) = u_0^m(x_1) + \sum_{l=1}^{m} \eta_{0,l}\psi_l^m\left(\frac{x - x_1}{h_1}\right) \tag{2.17}$$

coincides with (2.16) everywhere on $[x_1, x_1 + h_1]$. Differentiating (2.16), (2.17) and substituting $x_p = x_1 + \xi_p^m h_1$ for x we obtain

$$\eta_{0,p} = \delta \sum_{l=1}^{m} \tilde{\eta}_l\varphi_l^m(1 + \delta\xi_p^m), \quad p = \overline{1,m}. \tag{2.18}$$

Conclusions

Of course this paper does not thoroughly cover the subject of research. There are two main directions of the further investigation: the theoretical and the practical one. Among the theoretical problems we can mention the following.

1. Proof of the convergence of the stabilization iterative processes (1.7) in the general multistage case (recall that we have only proved the convergence of the generalized Picard process, see Theorem 1.1). The scheme of this proof seems to be clear: one should combine the property (1.5) of the stabilization equation (1.6′) with the known results on the convergence of Runge–Kutta methods.

2. Derivation of less restrictive convergence conditions for the discrete Picard process and proof of the convergence of the general multistage discrete SIPs (see Theorem 2.1 and comments below).

The practical part of the further research is the creation of a robust program based on the features we have briefly described above: the multigrid approach, the stepsize control and the initial approximation selection techniques. Without such a program we think it incorrect to compare the proposed numerical algorithms to the existing stiff ODE solvers. That is why in this article we have compared only *analytic* stabilization iterative processes with a Picard process in order to show that the proposed approach is worthy of attention.

Acknowledgements. The author is grateful to professor V. V. Bobkov for his support and patience.

References

1. K. Dekker and J. G. Verwer, *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations.* M.: Mir, 1988 (in Russian, transl. from English).

2. Wolfgang Hackbusch, *Integral Equations: Theory and Numerical Treatment.* Basel; Boston; Berlin: Birkhäuser, 1995 (International series of numerical mathematics; vol. 120).

3. M. A. Krasnoselskii [et al.], *Approximate Solution of Operator Equations.* Moscow, 1969 (in Russian).

4. N. S. Bakhvalov, N. P. Zhidkov, and G. M. Kobelkov, *Numerical Methods* M.: BINOM, 2004, pp. 345–353 (in Russian).

5. A. A. Samarskii and E. S. Nikolaev, *Solution Methods of Grid Equations.* Moscow: Nauka, 1978.

6. E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems.* M.: Mir, 1990 (in Russian, transl. from English).

7. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff Problems.* M.: Mir, 1999 (in Russian, transl. from English).