Краткие сообщения ОИЯИ №5[62]-93 *JINR Rapid Communications No.5[62]-93*
УДК 519.613

THE NEWTONIAN ITERATIVE SCHEME WITH SIMULTANEOUS CALCULATING THE INVERSE OPERATOR FOR THE DERIVATIVE OF NONLINEAR FUNCTION

I.V.Puzynin, I.V.Amirkhanov, T.P.Puzynina, E.V.Zemlyanaya

The modified iterative Newtonian procedure for solving nonlinear functional equation is proposed. The inversion for the derivative of nonlinear function on each iterative step is replaced by multiplication of some linear operators. On the basis of the proposed procedure an algorithm and a program for solving the eigenvalue problem for integral equation are developed. The calculations demonstrating a convergence of the algorithm and its efficiency for vector computer are performed.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Ньютоновская итерационная схема с одновременным вычислением оператора, обратного к производной нелинейной функции

И.В.Пузынин и др.

Предложен модифицированный ньютоновский итерационный процесс для решения нелинейного функционального уравнения, в котором обращение оператора производной нелинейной функции заменяется на каждом шаге перемножением вспомогательных линейных операторов. На основе предлагаемой модификации разработаны алгоритм и программа для решения задачи на собственные значения для интегрального уравнения. Проведены численные расчеты, иллюстрирующие сходимость описываемого процесса и его эффективность для векторных вычислительных систем.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Introduction

In the context of the development of vector-parallel computers the problem of elaborating special algorithms and programs for the effective using possibilities of such computers is very actual. An algorithm using some possibilities of these computers is presented. This algorithm is developed on the basis of modified continuous analogue of the Newton's method [1]. The

inversion of the derivative operator for nonlinear function on each iterative step is replaced by three multiplications of some linear operators.

The idea of such an approach for the finding of inverse matrixes was proposed in Refs. [2,3] in application to the method of the parameter variation. Its advantage is in the absence of the division operations during all calculations. Therefore, the division on a small number is excluded and the accuracy and stability of the calculations increase.

Below the iterative procedure is described for the nonlinear equation in B-space, the developed on this basis scheme for solving the eigenvalue problem for the integral equation is presented.

The efficiency of the algorithm proposed for system CONVEX C120 has been confirmed by testing calculations for the Schroedinger integral equation with the Coulomb potential in the impulse space /4/.

1. The Modified Algorithm

In accordance with the generalized continuous analogue of the Newton method for a nonlinear functional equation in B-space

$$\phi(z) = 0 \tag{1}$$

we construct the evolutionary equation with respect to additional parameter t

$$\frac{d}{dt}\phi(t, z(t)) = -\phi(t, z(t)), \quad 0 \le t < \infty \tag{2}$$

with the initial condition $z(0) = z_0$.

A parametrization $\phi = \phi(t, z(t))$ is performed so that for $t = 0$ we have a simple equation

$$\phi(0, z(0)) \equiv \phi_0(z_0) = 0,$$

which can be solved easily and $\lim_{t\to\infty} \phi(t, z(t)) = \phi(z)$.

One of variants of the parametrization can be performed by introducing the scalar function $g(t)$ [5], the so-called «function for including a perturbation»: $g(0) = 0$, $\lim_{t\to\infty} g(t) = 1$, and representing the function $\phi(t, z(t))$ as a sum

$$\phi(t, z(t)) = \phi_0(z(t)) + g(t)\,[\phi(z(t)) - \phi_0(z(t))].$$

From Eq.(2) designating $A(t) = \phi_z'(t, z(t))$ we obtain

$$\frac{dz}{dt} = -A^{-1}(t)\,[\phi(t, z(t)) + \phi_t'(t, z(t))]. \tag{3}$$

Since the integral of Eq.(2) is $\phi(t, z(t)) = e^{-t}\phi(0, z_0)$, then for $t \to \infty$, $\| \phi(t, z(t)) \| \to 0$, and $z(t)$ converges to desired solution z^*.

The discrete approximation of Eq.(3) with respect to the continuous parameter t: $(t_0, t_1, \ldots t_k)$; $t_0 = 0$, $t_{k+1} - t_k = \tau_k$ is performed in the framework of the Euler method

$$z(t_{k+1}) = z(t_k) + \tau_k V_k, \tag{4}$$

where

$$V_k = - B(t_k) [\phi(t_k, z(t_k)) + \phi_t'(t_k, z(t_k))],$$

Calculating an iterative correction V_k and step τ_k for each t_k, we receive a new approximation $z(t_{k+1})$ to solution z^*.

The iterative procedure should be continued until the next relation is fulfilled:

$$\| \phi(t_k, z(t_k)) \| \leq \varepsilon, \tag{5}$$

where $\varepsilon > 0$ is a given small number.

The convergence of this iterative procedure is justified in Refs. [1,5].

Let us turn to the description of the proposed modified iterative procedure. Let us consider the following system of functional-operator equations

$$\begin{cases} \phi(z) = 0, \\ BA - I = 0, \end{cases} \tag{6}$$

where $A = \phi_z'$, $B = A^{-1}$, I is the unit operator.

Introducing the continuous parameter $t (0 \leq t < \infty)$ and coming to the evolutionary equations system, we obtain:

$$\begin{cases} \dfrac{d}{dt} \phi(t, z(t)) = - \phi(t, z(t)), \\ \dfrac{d}{dt} [B(t)A(t) - I] = I - B(t)A(t). \end{cases} \tag{7}$$

After simple transforms we have finally:

$$\begin{cases} \dfrac{d}{dt} z(t) = - B(t) [\phi(t, z(t)) + \phi_t'(t, z(t))], \\ \dfrac{d}{dt} B(t) = [I - B(t)(A(t) + A'(t))] B(t). \end{cases} \tag{8}$$

As a result of discrete approximating of Eqs.(8) by the Euler scheme, $z_{k+1} = z(t_{k+1})$ and $B_{k+1} = B(t_{k+1})$ can be calculated on each iterative step if z_k and B_k are known:

$$\begin{cases} z_{k+1} = z_k + \tau_k V_k, \\ B_{k+1} = B_k + \tau_k W_k, \end{cases} \tag{9}$$

where

$$V_k = - B_k [\phi(t_k, z_k) + \phi_t'(t_k, z_k)],$$
$$W_k = [I - B_k(A_k + A_{kt}')] B_k. \tag{10}$$

Thus, having initial approximations z_0, B_0, all next approximations z_k, B_k can be found sequentially.

The iterative procedure is continued until the relation (15) and the following relation will be satisfied

$$|| B_k A_k - I|| < \varepsilon. \tag{11}$$

The practical calculations show that $B_0 = A^{-1}(z_0)$ is the best initial approximation for B.

2. Solution of the Eigenvalue Problem for Integral Equation

We consider a following equation:

$$\phi(z) \equiv y(x)\,(Q(x) - \lambda) + \alpha \int_0^R K(x, x')\, y(x')dx' = 0, \tag{12}$$

where $z = (y(x), \lambda)$ and the normalization condition

$$\int_0^R y^2(x)dx - N = 0. \tag{13}$$

Let us introduce parameter t, $0 \le t < \infty$ and the function $g(t) = 1 - e^{-t}$. Then $\phi(t, z)$ we represent as follows

$$\phi(t, z) = \phi_0(z) + g(t)\,(\phi(z) - \phi_0(z)). \tag{14}$$

We suppose that $\phi_0(z) = 0$ is some simple function equation with known solution z_0 being used as on initial approximation.

Taking into account $z(t) = (y(x, t), \lambda(t))$ and turning our attention to evolutionary equation, we receive for Eq.(14):

$$[(\phi_0(z(t)))_y' + g(t)\,[(\phi(z(t)))_y' - (\phi_0(z(t)))_y']]\, y_t' =$$
$$= [\phi_0(z(t)) + (g(t) + g_t'(t))\,(\phi(z(t)) - \phi_0(z(t)))] -$$
$$- [(\phi_0(z(t)))_\lambda' + g(t)\,((\phi(z(t)))_\lambda' - (\phi_0(z(t)))_\lambda')]\,\lambda_t'(t). \tag{15}$$

Using the method described in Sec.1 and designating $y_k(x) = y(x, t_k)$, $g_k = g(t_k)$, $g_k' = g'(t_k)$, $v_k = y_t'(x, t_k)$, $\mu_k = \lambda_t'(t_k)$, $B_k = A_k^{-1}$, taking into account Eqs. (12), (14), we obtain for each t_k

$$v_k = -B_k[G_k + \mu_k F_k],$$
$$W_k = [I - B_k(A_k + A_{k_t}')]B_k, \tag{16}$$

where

$$A_k(v_k(x)) = \phi_{0_y}'(z_k)v_k(x) + g_k[(Q(x) - \lambda_k)v_k(x) + \alpha \int_0^R K(x, x')v_k(x')dx' - \phi_{0_y}'(z_k)v_k(x)], \tag{17}$$

$$A_{k_t}'(v_k(x)) = (\phi_{0_y}'(z_k))_t' v_k(x) + g_k'[(Q(x) - \lambda_k)v_k(x) + \alpha \int_0^R K(x, x')v_k(x')dx' - \phi_{0_y}'(z_k)v_k(x)] - g_k v_k(x)[\mu_k + (\phi_{0_y}'(z_k))_t'], \tag{18}$$

$$G_k = \phi_0(z_k) + (g_k + g_k') \times \times [(Q(x) - \lambda_k)y_k(x) + \alpha \int_0^R K(x, x')y_k(x')dx' - \phi_0(z_k)], \tag{19}$$

$$F_k = \phi_{0_\lambda}'(z_k) - g_k(1 + \phi_{0_\lambda}'(z_k)). \tag{20}$$

Iterative correction v_k is calculated in the following way:

$$v_k = v_k^{(1)} + \mu_k v_k^{(2)}, \tag{21}$$

where

$$v_k^{(1)} = -B_k G_k, \qquad v_k^{(2)} = -B_k F_k. \tag{22}$$

Formula for the calculation of μ_k results from the normalization condition (13):

$$\mu_k = \frac{N - \int_0^R y_k(x)(y_k(x) - 2v_k^{(1)}(x))dx}{\int_0^R 2y_k(x)v_k^{(2)}(x)dx}. \tag{23}$$

After the calculation of iterative corrections μ_k, v_k, W_k one can find the new approximations by the Euler scheme:

$$y_{k+1} = y_k + \tau_k v_k,$$
$$\lambda_{k+1} = \lambda_k + \tau_k \mu_k,$$
$$B_{k+1} = B_k + \tau_k W_k. \tag{24}$$

The numerical approximation of Eq.(12) on the discrete mesh for x: $(0 = x_0 < x_1 < \ldots < x_n = R)$ reduces to the system of n algebraic equations:

$$\phi_i = y_i(Q_i - \lambda) + \alpha \sum_{j=1}^{n} K_{ij} y_j \xi_j , \qquad (25)$$

where $y_i = y(x_i)$, $Q_i = Q(x_i)$, $K_{ij} = K(x_i, x_j)$, ξ_j are coefficients of the quadrature formula of the numerical integrating. Operators A, A_t', F,G in (16)—(20) are approximated by matrices $\{a_{ij}\}$, $\{\tilde{a}_{ij}\}$ and vectors (F_i), (G_i) correspondingly. In particular, the matrix elements a_{ij} and $\tilde{a}_{ij}$ on kth iterative step are determined as follows:

$$a_{ij} = a_{ij}^{(0)} + g_k [(Q_i - \lambda_k)\delta_{ij} + \alpha K_{ij}\xi_j - a_{ij}^{(0)}], \qquad (26)$$

$$\tilde{a}_{ij} = \tilde{a}_{ij}^{(0)} + g_k' [(Q_i - \lambda_k)\delta_{ij} + \alpha [K_{ij}\xi_j - \tilde{a}_{ij}^{(0)}] - g_k [\mu_k + \tilde{a}_{ij}^{(0)}], \qquad (27)$$

where δ_{ij} is the Kronecker symbol, $a_{ij}^{(0)}$ and $\tilde{a}_{ij}^{(0)}$ are elements of matrices approximating the operators $\phi_{0y}'(z_k)$ and $[\phi_{0y}'(z_k)]_t'$ correspondingly on the discrete mesh.

We note that the additional normalization of function $y_k(x)$ for each iteration essentially improves the convergence of the iterative procedure.

3. The Program Realization

In order to solve Eq.(12) numerically two program complexes on the FORTRAN-77 are developed:

a) CANMIM realizes the iterative procedure (3)—(4) with the calculation of matrix $B_k = A_k^{-1}$ by inverse of A_k on each iteration (with the help of a standard subprogram MATIN2 from the CERNLIB [6]).

б) CANMIT realizes the iterative procedure (6)—(10) where B_k is parallel calculated in iterations.

Coefficients $\{\xi_j\}$ for the test presented below are selected in accordance with the Gregory formula [7]. The step τ_k was calculated by algorithms described in ref. [8]. Test calculations and comparison of time operation of the two programs are performed.

When comparing schemes (3)—(4) and (6)—(10), it can be seen that the scheme (3)—(4) contains one inverse of matrix A, and in scheme (6)—(10) it is replaced by three matrix multiplications: BA, BA_t', $[I - BA - BA_t']B$.

Should be noted we succeeded in finding the conditions in which the number of iterations for two processes is identical. Taking into account that in the scalar variant of the realization the time of the inversion of one matrix is approximately equal to the time of multiplication of two matrixes, the expected calculation time for the scheme (6)—(10) is thrice more than that for the scheme (3)—(4). However, the matrix multiplication is more preferable than the matrix inverse from the point of view of vector representation of operations. It can be expected that algorithm (6)—(10) will give an advantage of time on the vector computer. Naturally, this advantage will be obtained at the expense of additional of the computer memory required for additional matrixes.

4. Numerical Example

In the present paper the proposed method is illustrated by the solving of the Schroedinger equation in impulse representation for the Coulomb potential [4].

This equation is as follows:

$$\phi(z) \equiv (p^2 - \lambda)\, y(p) - 2\tilde{Z} \int_0^\infty dp' \ln \left| \frac{p + p'}{p - p'} \right| y(p') = 0 \qquad (28)$$

with the normalization condition

$$\int_0^\infty y^2(p)dp = 1,$$

where $z = (y(p), \lambda)$, λ is eigenvalue, $\lambda = 2E$, E is energy level. One of analytical solutions for $\tilde{Z} = 1$ is following

$$\lambda_1^* = -1.0, \qquad y_1^*(p) = \sqrt{\frac{2}{\pi}} \frac{p}{(p^2 + 1)^2}.$$

To eliminate the singular point $p = p'$ let us transform Eq. (28) in the following way

$$(p^2 - \lambda)\, y(p) - 2\tilde{Z} \int_0^\infty dp' \ln \left| \frac{p + p'}{p - p'} \right| (y(p') - y(p)) - 2\tilde{Z} y(p) I_1 = 0,$$

where $I_1 = \int_0^\infty dp' \ln \left|\frac{p+p'}{p-p'}\right|$. In this case the integrand function in singular point has a finite limit equal to 0.

Thus, this equation has the next form:

$$(p^2 - 2\tilde{Z}I_1 - \lambda)y(p) - 2\tilde{Z}\int_0^R dp' \ln\left|\frac{p+p'}{p-p'}\right| (y(p') - y(p)) = 0, \quad R \to \infty. \quad (29)$$

Choosing a finite interval $0 \le p \le R$, $R >> 1$, introducing a discrete mesh on parameter p and approximating the Eq.(29) numerically, we obtain a system (25), where $Q_i = p_i^2 + \alpha I_{1i} + \alpha \sum_{j=1}^{n} K_{ij}\xi_j$, $K_{ij} = \ln \left|\frac{p_i + p_j'}{p_i - p_j}\right|$, $\alpha = -2\tilde{Z}$. The solving of Eq.(29) is performed according to the scheme described in Sec.2 and defined by formulas (15)—(27). As an initial approximation for $y(p)$ the function $y_0(p) = pe^{-p}$, $\phi_0(z) = (c - \lambda)\, y(p)$ (c = const) was used.

The numerical investigation of the convergence of the discrete solutions to the precise ones with respect to mesh parameters h and R was performed.

In Table 1 for $\lambda_1^* = -1$ there is a dependence of the eigenvalues λ_{1h} of the discrete problem on the mesh step h. Results are given for the scheme (6)—(10) on the sequence of three twice condensing meches with steps h, $h/2$, $h/4$. The obtained value of

$$\sigma = |\lambda_h - \lambda_{h/2}| / |\lambda_{h/2} - \lambda_{h/4}| \sim 8.1$$

corresponds to theoretical accuracy of the Gregory formula $O(h^3)$.

Table 1. Interval [0, 30]

n	h	λ_h
151	0.2	− 1.00031044
301	0.1	− 1.00009837
601	0.05	− 1.00007235

The comparison analysis of execution time for programs CANMIM and CANMIT on different computers has particular interest. It should be noted that the programs use only library programs of linear algebra without some special optimizations with respect to operation systems and special features of the computer. In Tables 2a—2c the dependence of the execution time T (sec) for each program on the number of mesh points n for problem under

Table 2a. Interval [0, 10], n = 101

SCHEME	PROGRAM	k	λ_k	δ_k	T	T_2/T_1
(3)—(4)	CANMIM	5	−0.997	$1.3E^{-5}$	92	3.01
(6)—(10)	CANMIT	5	−0.997	$1.4E^{-5}$	274	

Table 2b. Interval [0, 20], n = 201

SCHEME	PROGRAM	k	λ_k	δ_k	T	T_2/T_1
(3)—(4)	CANMIM	5	−0.998	$1.1E^{-5}$	623	3.19
(6)—(10)	CANMIT	5	−0.998	$1.1E^{-5}$	1989	

Table 2c. Interval [0, 30], n = 301

SCHEME	PROGRAM	k	λ_k	δ_k	T	T_2/T_1
(3)—(4)	CANMIM	5	−1.0001	$1.06E^{-5}$	1903	2.94
(6)—(10)	CANMIT	5	−1.000098	$1.4E^{-5}$	5597	

solution are presented for VAX 8350. Besides these Tables demonstrate a convergence of eigenvalues of the discrete problem when increasing parameter R. Everywhere $h = 0.1$, $\varepsilon = 10^{-4}$, $\lambda_0 = -0.9$, $\delta_0 = 0.1$.

As is obvious from the Tables, both algorithms for identical parameters R and h and similar values of residual δ_k have equal number k of iterations.

It is also seen that the program CANMIT spends thrice more execution time than the program CANMIM.

In order to check up the possibilities of the vector representation of operations on the level of the operation system CONVEX C120, each program complex was executed twice:

1) without all optimization parameters of translation (TC — time of calculation);

2) with the optimization parameter -O2 (TCO2 — time of calculation with the vector representation of operations).

In Table 3 the execution times TC and TCO2 (sec) and relation TC/TCO2 characterizing a property of «accelerating the calculations» are presented for both schemes. Besides, for comparison the execution times for the system VAX 8350 and for SUN Spark Station 2 are given.

Table 3. Interval [0, 10], n = 101

		VAX	SUN	CONVEX		
SCHEME	PROGRAM			TC	TCO2	TC/TCO2
(3)—(4)	CANMIM	92	15	31	12	≅ 2.6
(6)—(10)	CANMIT	274	48	91.9	7.8	≅ 11.8

From Table 3 it follows that the scheme (6)—(10) is more effective for the vector system using only one of the possibilities of the CONVEX operation system.

In the considered case the vector representation gives a relative acceleration of working this scheme nearly 1.5 times if comparing with the scheme (3)—(4).

Much greater advantage is observed if comparing the execution times of vector variant of the CANMIT-program with the scalar variants of this program and the CANMIM-program.

Conclusion

The aim of the performed numerical experiences is to demonstrate advantages of the proposed iterative procedure for its realization on vector computers. Special methods of vector representation of operations for calculations and special features of solved problems such as, for instance, the symmetry of integral operator kernel are not knowingly used in the calculations. Calculations were performed on comparatively sparse meshes in order to compare the work of the programs on different computers including computers with relatively small memory capacity.

The Newton iterative scheme for solving the nonlinear equations with simultaneous iterations of the inversion operator to derivative of nonlinear function is perspective for using on the vector computing systems with sufficiently large memory capacity or a system of external memory with fast access. The efficiency of the corresponding programs can be significantly increased by using special methods of vector and parallel calculations on multiprocessor computers.

We thank the Commission of the European Communities for partial financial support in the frame of the EC-Russia collaboration Contract №ECRU002.

References

1. Zhanlav T., Puzynin I.V. — Comput. Maths Math. Phys., 1992, 32, No.1, p.1.
2. Davydenko D.F. — Docl. Ac. Nauk SSSR (In Russian), 1960, 131, 3, p.500
3. Holbrow W., Hass R., Kalaba E., Zagustin E. — University of Southern California, Los Angeles, 1972.
4. Bete G., Salpeter E.E. — Quantum Mechanics with One and Two Electrons, M., Fizmatgiz (in Russian), 1960.
5. Zhanlav T., Puzynin I.V. — Comput. Maths Math. Phys., 1992, 32, No.6, p.729.
6. CERN Computer Centre, Program Library, Geneva, 1988.
7. Kalitkin N.N. — Numerical Methods, M., Nauka (in Russian), 1978.
8. Ermakov V.V., Kalitkin N.N. — Zh. Vych. Mat. i Mat. Fiz. (in Russian), 1981, 21, p.49.

Puzynin I.V., Puzynina T.P., Strizh T.A. — JINR Comm., P11-87-332, Dubna, 1987.

Amirkhanov I.V., Zemlyanaya E.V., Puzynina T.P. — JINR Comm., P11-91-139, Dubna, 1991.

Received on December 17, 1993.