GRANT
NGR 22-009-727

A Numerical Scheme to Solve

Unstable Boundary Value Problems

Eugenia Kálnay-Rivas
Department of Meteorology
Massachusetts Institute of Technology
Cambridge, Mass. 02139

iii + 22 pages
2 figures
no tables

Unstable Boundary Value Problems

Eugenia Kálnay-Rivas
Department of Meterology
Massachusetts Institute of Technology
Cambridge, Mass. 02139

Abstract

A new iterative scheme for solving boundary value problems is presented. It consists of the introduction of an artificial time dependence into a modified version of the system of equations. Then explicit forward integrations in time are followed by explicit integrations backwards in time. The method converges under much more general conditions than schemes based in forward time integrations (false transient schemes). In particular it can attain a steady state solution of an elliptical system of equations even if the solution is unstable, in which case other iterative schemes fail to converge. The simplicity of its use makes it attractive for solving large systems of nonlinear equations.

1. Introduction

Elliptic equations are often solved numerically by iterative methods. It is generally possible to associate an iterative scheme with some time marching scheme corresponding to a parabolic equation, i.e., the solution of the elliptic equation is obtained as the steady state solution of a corresponding parabolic equation [1].

If the steady state solution of a parabolic system of equations is unstable, developing instabilities will prevent such solution from being reached starting from any set of initial conditions. In a similar way instabilities will prevent "time marching" iterative schemes from converging to an unstable solution.

We have developed a new iterative scheme with which it is possible to find the steady state solution of a parabolic system of equations even if it is unstable. The new scheme is very simple to use and of rather general application. It can be used to solve boundary value problems, and, more generally, large systems of linear or nonlinear equations not easily solved by other iterative techniques.

For simplicity the scheme is introduced first for the solution of a single complex equation (section 2), and acceleration procedures are discussed in section 3. In section 4 the method is extended to systems of equations. In section 5 we present examples of application to the solution of systems of equations, and section 6 contains a description of three cases in which the method was used to obtain physically unstable steady state solutions.

2. Iterative solution of a single complex equation

Consider the transcendental (complex in general) equation

$$f(u) = 0. \tag{2.1}$$

We introduce an artificial time dependence and try to find the steady state solution of

$$\frac{du}{dt} = f(u). \tag{2.2}$$

If $u = u_o$ is a solution of (2.1) we write $u = u_o + v$, and assuming v sufficiently small, expand (2.2) in a Taylor series

$$\frac{dv}{dt} = \gamma v + 0(v^2). \tag{2.3}$$

Here $\gamma = \frac{\partial f}{\partial u}\Big|_{u_o}$ is a complex number if $f(u)$ is complex. From (2.3) it is clear that for any time marching scheme, there may be values of γ for which $|v|$ will be amplified, so that iterative schemes based on (2.2) will not converge. On the other hand the scheme that we will now present will converge to the solution for any value of γ.

Let us replace (2.2) by a modified equation

$$\frac{du}{dt} = f^*(u), \tag{2.4}$$

where the asterisk represents the complex conjugate. The scheme consists of an explicit Euler "time" integration, followed by an explicit Euler integration going backwards in "time". In both steps of the iteration the modified equation (2.4) is used:

$$\tilde{u} = u^\nu + \Delta t\ f^*(u^\nu) \tag{2.5a}$$

$$u^{\nu+1} = \tilde{u} - \Delta t\ f^*(\tilde{u}). \tag{2.5b}$$

We use the superindex ν to denote iteration number, not time, since we always return to the same "time level".

When we linearize with respect to the departure from the solution, as in (2.3), we obtain

$$\tilde{v} = v^{\nu} + \Delta t \, \gamma^{*} \, v^{\nu^{*}} \tag{2.6a}$$

$$v^{\nu+1} = \tilde{v} - \Delta t \, \gamma^{*} \, \tilde{v}^{*} \tag{2.6b}$$

and eliminating $\tilde{v}$,

$$v^{\nu+1} = (1 - \Delta t^{2} \, \gamma\gamma^{*}) v^{\nu} \; . \tag{2.7}$$

This result indicates that if the stability criterion

$$\Delta t^{2} \, \gamma\gamma^{*} < 2 \tag{2.8}$$

is satisfied the time dependent component v will be damped out. This also shows that the method is of first order convergence, i.e., the error is reduced by an approximately constant factor at each iteration so that it decreases as $v^{\nu} \sim (1 - \gamma\gamma^{*}\Delta t^{2})^{\nu}$.

In the case of a double root, i.e., when $\gamma = \frac{\partial f}{\partial u} = 0$ at $u = u_o$, the method can still be applied. Then $f(u) = \left(\frac{\partial^{2} f}{\partial u^{2}}\right)_{u=u_o} \frac{v^{2}}{2} + 0(v^{3})$, and when (2.5) is applied we obtain

$$v^{\nu+1} = \left(1 - \frac{\Delta t^{2}}{2} \, \eta\eta^{*} \, v^{\nu} v^{\nu^{*}}\right) v^{\nu} \tag{2.9}$$

where $\eta = \frac{\partial^{2} f}{\partial u^{2}}$ at $u = u_o$. Therefore the convergence rate is much slower, and $v^{\nu} \sim \nu^{-\frac{1}{2}}$.

The "time step" Δt is a computational parameter, which may vary with ν. For a linear equation it is obvious that the optimum value is $\Delta t_{opt} = \frac{1}{|\gamma|}$. For a nonlinear equation γ can be determined from the equation itself or from the solution after the first iteration (see section 3), but for rapidly varying functions it is safe to take $\Delta t < \Delta t_{opt}$ and go "curve crawling" [2]. In practice, however, Δt can be determined as the largest value that produces stable results.

Figure 1 shows graphically how the method works for the case of a real γ. Values of $\frac{du}{dt}$ for different values of u (assuming γ constant) are plotted. Since the $\frac{du}{dt}$ is proportional to $u-u_o$, the derivatives are steeper for values of u far from the solution u_o. If $\Delta t = \Delta t_{opt} = \frac{1}{|\gamma|}$, the method converges in one iteration (for linear equations). It should be noted that if, as in Figure 1, $Re(\gamma) \geq 0$, a simple time marching scheme is unstable.

3. Acceleration of the scheme applied to a single equation

3.1 Extrapolation

When the error v is sufficiently small, γ may be considered approximately constant, and from the results of each iteration we can extrapolate and obtain second order convergence. From (2.6) we get

$$\begin{aligned} u^{\nu} - \tilde{u} &\approx -\gamma^{*}\, \Delta t\, v^{\nu^{*}} \\ u^{\nu} - u^{\nu+1} &\approx \gamma\gamma^{*}\, \Delta t^{2} v^{\nu} \end{aligned} \tag{3.1}$$

and therefore the best estimate of v^{ν} is

$$v^{\nu} \approx \frac{(u^{\nu}-\tilde{u})(u^{\nu}-\tilde{u})^{*}}{(u^{\nu}-u^{\nu+1})^{*}} \tag{3.2}$$

and

$$\gamma^{*}\Delta t \approx \frac{u^{\nu} - u^{\nu+1}}{u^{\nu} - \tilde{u}} \quad . \tag{3.3}$$

Then for the next iteration optimum estimates of $u^{\nu+1}$ and Δt are obtained as follows:

$$u^{\nu+1}_{opt} = u^{\nu} - \frac{(u^{\nu}-\tilde{u})(u^{\nu}-\tilde{u})^{*}}{(u^{\nu}-u^{\nu+1})^{*}} = u_{o} + O(v^{2}) \tag{3.4}$$

$$\Delta t^{\nu+1}_{opt} = \frac{\Delta t^{\nu}}{|\gamma^{*\nu}\Delta t^{\nu}|} \quad . \tag{3.5}$$

3.2 Real equation

If f(u) is real, it is necessary to apply only one of the forward and backward parts of scheme (2.5). In other words, (2.5) reduces to

$$u^{\nu+1} = u^{\nu} + \Delta t' f(u^{\nu}) \tag{3.6}$$

where Δt is determined from a single complete iteration (2.5):

$$\Delta t' = \text{sign}\ (\frac{u^1-u^o}{\tilde{u}-u^o})\Delta t \qquad (3.7)$$

Fig. 2 indicates how the method works if this option is used.

4. Solution of a system of equations

Consider the system of equations

$$F(u) = 0 \tag{4.1}$$

where F is a nonlinear, complex in general, matrix operator, and u is the vector of the dependent variables. If u_o is a solution we expand in a Taylor series with respect to the departure from the solution, and neglect terms higher than linear:

$$F(u) \approx F(u_o) + Jv = Jv . \tag{4.2}$$

Here $J = \frac{\partial F}{\partial u}\Big|_{u=u_o}$ is the jacobian such that $J_{ij} = \partial F_i/\partial u_j$.

Note that a simple extension of scheme (2.5) to higher dimension

$$\tilde{u} = u^{\nu} + \Delta t\ F^{*}(u^{\nu})$$

$$u^{\nu+1} = \tilde{u} - \Delta t\ F^{*}(\tilde{u}) \tag{4.3}$$

will <u>not</u> converge in general to the solution, because J can have complex eigenvalues even if F is real.

Let $J = S + A$, where $S = (J + J^T)/2$, $A = (J - J^T)/2$ are symmetric and antisymmetric matrices respectively and the superindex T denotes transpose. Then the extension of scheme (2.6) to a system of equations is:

$$\tilde{v} = v^{\nu} + \Delta t\ (S^{*} + A^{*})\ v^{\nu^{*}} \tag{4.4a}$$

$$\tilde{\tilde{v}} = v^{\nu} + \Delta t\ (S^{*} - A^{*})\ v^{\nu^{*}}. \tag{4.4b}$$

$$v^{\nu+1} = \tilde{v} - \Delta t\ (S^{*} + A^{*})\ \tilde{\tilde{v}}^{*} . \tag{4.4c}$$

Here, as before, the asterisk denotes complex conjugate. When $\tilde{v}$ and $\tilde{\tilde{v}}$ are eliminated in (4.4), we obtain

$$v^{\nu+1} = [I - \Delta t^2(S^* + A^*)(S - A)]v^\nu \qquad (4.5)$$

where I is the identity matrix.

Therefore a <u>sufficient</u> condition for convergence is

$$S^*A = A^*S \; . \qquad (4.6)$$

It is easily verified that if J is real, condition (4.6) is equivalent to J being normal, i.e., $JJ^T = J^TJ$.

If condition (4.6) is fulfilled, then from (4.5) and $S^{*T} = S^*$, $A^{*T} = -A^*$ we obtain

$$v^{\nu+1} = [I - \Delta t^2(S^{*T}S + A^{*T}A)]v^\nu. \qquad (4.7)$$

The matrix $S^{*T}S + A^{*T}A$ is the sum of positive definite matrices and therefore it is positive definite, i.e., it has real positive eigenvalues μ_n. If the linear stability criterion

$$\Delta t^2\mu_n < 2 \qquad (4.8)$$

is satisfied for all n, the method will converge.

For a nonlinear problem, scheme (4.4) becomes

$$\tilde{u} = u^\nu + \Delta t(F_S^*(u^\nu) + F_A^*(u^\nu)) \qquad (4.9a)$$

$$u = u^\nu + \Delta t(F_S^*(u^\nu) - F_A^*(u^\nu)) \qquad (4.9b)$$

$$u^{\nu+1} = \tilde{u} - \Delta t(F_S^*(\tilde{\tilde{u}}) + F_A^*(\tilde{\tilde{u}})) \qquad (4.9c)$$

where F_S and F_A are the nonlinear operators corresponding to S and A respectively. It should be noted that, computationally, the most expensive part of the scheme, which is the evaluation of F(u), is executed only twice per iteration, in steps (4.9a) and (4.9c).

It is possible to design an extrapolation procedure as in (3.2) to determine v^ν and obtain second order convergence. It would require, however, an explicit evaluation of the jacobian J and of its inverse, and therefore it would destroy the simplicity of the method.

5. Examples of applications

In the following examples we have not tried to optimize the procedure, but rather to test the ability of the method to solve nonlinear equations. It should be noted that a good initial guess is not necessary.

5.1 Single transcendental equations

a) Real equations

i) We found roots of u = tan u by determining the steady state solutions of $\frac{\partial u}{\partial t} = (u - \tan u)/u$. We divided by u in order to avoid the trivial solution u = 0. The conservative value $\Delta t = .2\, \Delta t_{opt}$ was used. Once two succesive approximations differed by less than 1% the extrapolation procedure was used. 10-40 iterations were needed to get a value close to the solution, depending on the initial guess, and then 1-3 extrapolations were enough to converge with an error smaller than 10^{-7}. When the option described in section 3.2 was used, about one fourth the number of iterations were necessary.

ii) sin u = .5. Using the option described in section 3.2 only 2-4 iterations and 1-2 extrapolations are necessary for convergence. The value $\Delta t = 1$ was determined from the equation itself.

iii) sin u = 1. This is a double root case, and convergence is very slow. It takes about 100 iterations to reduce the error to 1%.

b) Complex equations

We solved sin u = 1.17520 using the standard scheme (2.5). Starting from Re(u) = 2, Im(u) = 0.5, $\Delta t = 1$, 10 iterations and 2 extrapolations yielded the solution $\mathrm{Re}(u_o) = 1.570796$, $\mathrm{Im}(u_o) = 0.583628$. The same result was obtained after 25 standard iterations.

5.2 Solution of ordinary differential equations

a) Steady state solution of $\frac{\partial u}{\partial t} = (u+2)\frac{\partial u}{\partial x} + (u+1)$, with periodic boundary conditions: $u(0) = u(1)$. The only steady state solution is $u_o = -1$, but it is unstable. We found iteratively the steady state solution by applying (4.9) in the following way:

$$\tilde{u} = u^{\nu} + \Delta t \{(u^{\nu} + 2)u_x^{\nu} + (u^{\nu} + 1)\}$$

$$\tilde{\tilde{u}} = u^{\nu} + \Delta t \{-(u^{\nu} + 2)u_x^{\nu} + (u^{\nu} + 1)\}$$

$$u^{\nu+1} = \tilde{u} - \Delta t\{(u^{\nu} + 2)\tilde{\tilde{u}}_x + (\tilde{\tilde{u}} + 1)\}.$$

The x-derivatives on the right hand side were replaced by centered finite differences. We started from the initial guess $u = \sin 2\pi x$ and used a time step $\Delta t = .1$. About 400 iterations were necessary to obtain the solution $u_o = -1$ with an error smaller than 1%.

b) A nonlinear boundary value problem

This method can be used to solve nonlinear boundary value problems in a rather inefficient way, but with minimum human effort. Consider the example given by Acton [2], page 171:

$$y'' - y^2(1 - 0.2 \sin 2x) = 0$$

$$y(0) = 1, \; y(1) = 0 \; .$$

Since there are no first derivatives the problem is solved in the following way, using finite differences to express x-derivatives:

$$\tilde{y} = y^{\nu} + \Delta t\{y_{xx}^{\nu} - (y^{\nu})^2(1 - 0.2 \sin 2x)\}$$

$$y^{\nu+1} = \tilde{y} - \Delta t\{\tilde{y}_{xx} - (\tilde{y})^2(1 - 0.2 \sin 2x)\} \; .$$

About 300 iterations were necessary to converge within 1%, and after

800 iterations the error was smaller than 10^{-6}.

c) A "stiff" equation

The equation $\frac{d^2y}{dx^2} - 101\frac{dy}{dx} + 100y = 0$ has solutions $y_1 = e^x$ and $y_2 = e^{100x}$. Only y_1 satisfies the boundary conditions $y(0) = 1$, $y(1) = e$, but the equation cannot be solved by shooting or by relaxation because the "hidden solution" y_2 is amplified.

We tried to solve this equation by writing

$$\tilde{y} = y^\nu + \Delta t\{y^\nu_{xx} - 101y^\nu_x + 100y^\nu\} \tag{5.1a}$$

$$\tilde{\tilde{y}} = y^\nu + \Delta t\{y^\nu_{xx} + 101y^\nu_x + 100y^\nu\} \tag{5.1b}$$

$$y^{\nu+1} = \tilde{y} - \Delta t\{\tilde{\tilde{y}}_{xx} - 101\tilde{\tilde{y}}_x + 100\tilde{\tilde{y}}\} \tag{5.1c}$$

and replacing the x-derivatives by centered second order finite differences in the interval (0,1). After about 5000 iterations the solution remained bounded but it failed to converge. A closer analysis shows that the scheme satisfies condition (4.5) everywhere except at the interior points next to the boundaries, where the finite differences corresponding to the first and second derivatives do not commute. This is easily solved by eliminating $\tilde{y}$, $\tilde{\tilde{y}}$ in (5.1) and then applying finite differences on

$$y^{\nu+1} = y^\nu - \Delta t^2\{y^\nu_{xxxx} - 10001y^\nu_{xx} + 10000y^\nu\}$$

at the points next to the boundaries. With 20 grid intervals and $\Delta t = 2 \times 10^{-4}$ the method converged with an error smaller than 10^{-4} in about 2000 iterations.

5.3 A highly nonlinear system of equations

Scarf [3] developed a constructive method to find fixed points of a mapping that transforms a space into itself. He tested the method on a highly nonlinear system of equations

$$\sum_{\ell=1}^{3} \left(\frac{a_{\ell i} \sum_{k=1}^{5} w_{\ell k} \pi_k}{\pi_i^{b_\ell} \sum_{k=1}^{5} a_{\ell k} \pi_k^{1-b_\ell}} - w_{\ell i} \right) = 0 \qquad i = 1,\ldots,5 \qquad (5.2)$$

subject to the normalization condition $\sum_{i=1}^{5} \pi_i = 1$. Here $a_{\ell i}$, $w_{\ell k}$ and b_ℓ are positive data matrices, and π_i is a vector of unknowns. According to Scarf [3] this system of equations cannot be solved by gradient methods.

We solved the problem by multiplying (5.2) by π_i and applying only 4.9a and 4.9c under the assumption $F_A(\pi) \approx 0$. The solution was normalized after each iteration. The method converged in 300 iterations within an error of 10^{-6}. However when we tried a simple forward scheme (4.9a), it also converged and in only 60 iterations. This illustrates the point that whenever a simple forward scheme converges, it should be used, because it is more efficient than the forward-backward scheme (cf. section 3.2).

For comparison we note that Scarf's [3] constructive algorithm converged in 168 iterations, but the averaged solution given by Scarf has errors varying between 2 and 20%.

6. Physically unstable steady state solutions

As discussed in the introduction, this scheme was originally devised as a method of finding steady state solutions of partial differential equations which are unstable, so that small perturbations are amplified by regular iterative techniques.

The method has been tested and found to yield an unstable steady state solution in three different cases.

6.1 Rotating annulus: example of a spectral (Fourier) model

Lorenz [4] developed a simplified spectral model of a rotating annulus heated in its rim and cooled in the center. It consists of a system of only 14 ordinary differential equations for the real and imaginary Fourier components. For certain values of the external parameters, the thermal Rossby number and the Taylor number, the steady state solution becomes unstable and Rossby waves develop.

In a spectral (Fourier) model the linearized jacobian J (see eq. 4.2) is complex, but it has only diagonal elements. Therefore the straightforward scheme (4.3) can be used.

Following Lorenz [4] we made a regular forecast for an unstable situation, until Rossby waves were well developed. Then, using the same time step as Lorenz, we applied (4.3) and, after 300 iterations, the model converged to the unstable steady state solution found analytically by Lorenz.

6.2 An unstable steady state boundary layer

Inez Fung [5] has used scheme (4.9) to obtain the unstable steady state axisymmetric nonlinear boundary layer of a geophysical vortex, where the Coriolis parameter is different from zero. Two cases were considered: a) a V-R vortex, and b) Carrier et al. [6] model hurricane vortex. Convergence occurred in about 1000 iterations in both cases, and the results for case b) compared well with those for the downdraft region in Carrier's model.

6.3 A steady state solution with shear instability: example of a finite differences model

The flow defined by the equation

$$\frac{\partial\zeta}{\partial t} = + J(\psi,\zeta) + \nu\nabla^2\zeta + C \sin y$$

has a steady state solution

$$\zeta_o = \frac{C}{\nu} \sin y$$

which is unstable due to the presence of inflection points in the profile of the mean velocity field. Here ζ is the vorticity, the stream function ψ is defined through the equation

$$\nabla^2\psi = \zeta ,$$

ν is a diffusion coefficient and C is a constant.

We used centered finite differences and periodic boundary conditions to solve the problem. Starting from the analytical steady state solution we made a regular forecast, using Lorenz' [7] N-cycle scheme with N = 8. Immediately strong instabilities developed, until the mean flow was completely distorted. Then we started again from the analytical solution but adding a strong perturbation of the form $\frac{C}{2\nu}$ sin 2x sin 2y, and applied scheme (4.9). After about 800 iterations the analytical solution was recovered.

7. Summary and conclusions

We have presented a new numerical scheme to solve nonlinear boundary value problems. It consists of the introduction of an artificial time dependence into a modified version of the equations. Then explicit forward integrations in "time" are followed by explicit integrations backwards in "time". For a real system of equations, if only the forward part of the step is executed, the method reduces to one of the class of "false transient" methods that has been widely used to solve elliptical systems of equations. (See discussion by Varga, [1] pp. 270-278, and Mallinson and de Vahl Davis, [8]) The forward-backward method is less efficient than the forward methods, since it requires twice as many computations per iteration and the convergence rate is usually slower. On the other hand, the forward-backward scheme converges under much more general conditions than the forward schemes. In particular, it converges to the steady state solution of an elliptical system of equations even if the solution is unstable, in which case the forward schemes fail.

The method has been tested for several cases, and the numerical examples indicate that when the size of the time step is chosen close to the maximum value compatible with numerical stability, convergence occurs on the order of serveral hundred iterations.

The number of iterations necessary for convergence is highly dependent on the size of the time step, and on the difference between largest and smallest eigenvalues μ_n of the matrix $S^{*T}S + A^{*T}A$ (section 4). Therefore a thoughtful variation of the time step size with the iteration number, as done for example in the Okamura-Rivas scheme (Grant and Rivas, [9] may increase significantly the efficiency of the method. It may also be possible to normalize the system of equations in order to decrease the range of the eigenvalues μ_n. This is equivalent to choosing different time scales for different equations.

The numerical method presented here, though strictly iterative, shares all the simplicity of initial value problems solved with explicit time differencing schemes. It is hoped that it will prove useful for solving systems of large numbers of nonlinear equations not easily solved with other iterative techniques.

Acknowledgements

I am very grateful to Prof. J.G. Charney for suggesting the problem of devising a scheme to find unstable solutions of boundary value problems. Several useful discussions with Dr. Mark Cane, Prof. E.N. Lorenz and Ms. Diana L. Spiegel are also gratefully acknowledged. This research has been supported by the National Science Foundation under grant no. NSF OCD 71-00333. The computations were performed at the National Aeronautics and Space Administration (NASA) Goddard Institute for Space Studies in New York (NASA Grant NGR 22-009-727).

References

1. R.S. Varga, Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs, N.J. (1962).

2. F.S. Acton, Numerical Methods that Work. Harper and Row, New York (1970).

3. H. Scarf, The approximation of fixed points of a continuous mapping, SIAM J. Appl. Math. 15 (1967), 1328-1343.

4. E.N. Lorenz, The mechanics of vacillation, J. Atmos. Sci. 20 (1963), 448-464.

5. I. Fung, The organization of spiral rainbands in a hurricane, Ph.D. Thesis MIT Dept. of Meteorology (in preparation).

6. G.F. Carrier, A.L. Hammond and O.D. George, A model of the mature hurricane, J. Fl. Mech. 47, part 1 (1971), 145-170.

7. E.N. Lorenz, An N-cycle time-differencing scheme for stepwise numerical integration, Mon. Wea. Rev. 99 (1971), 644-648.

8. G.D. Mallinson, and G. de Vahl Davis, The method of false transient for the solution of coupled elliptic equations, J. Comp. Phys. 12 (1973), 435-461.

9. W.K-F. Grant, and E. Kálnay-Rivas, Initialization procedures for primitive equations models, submitted to Mon. Wea. Rev.

Figure Captions

Fig. 1: Graphic interpretation of the iterative scheme (2.5) for a real equation $f(u) = 0$. γ is assumed constant and the dashed lines indicate the time derivative for different values of u.

Fig. 2: Same as Fig. 1, except for the scheme (3.6).

List of Symbols

$u, \tilde{u}, u^{\nu}, \tilde{\tilde{u}}, u_o$

$v, w,$

f, F, Fa, Fs

f^*, F^*

S, S^*, S^*T

A, A^*, A^*T

I

J

Greek letters

γ	gamma
Δ	delta
ν	nu (superindex)
∂	(partial derivative)
η	eta
π	pi
ψ	Psi
ζ	Zeta
μ	mu