

Dynamic Iterations for Nonlinear Systems Applied in Population Dynamics



Barbara Zubik-Kowal

Abstract We investigate the application of dynamic iterations to nonlinear systems of differential equations. Such an application allows to use implicit time integration methods without solving nonlinear algebraic equations at each time step. Another advantage of the application of dynamic iterations is that the resulting numerical schemes can be solved in parallel computing environments. We conclude that the sequence of how the dynamic iterations are applied is significant and influences their rate of convergence to the solution of the given system of nonlinear differential equations. This conclusion is illustrated by numerical experiments involving Volterra equations for predator-prey interactions. We also conclude that the proposed numerical scheme is faster than the variable order method.

1 Introduction

Dynamic iterations have been broadly investigated as numerical methods applied to solve differential systems on parallel computers and are often called waveform relaxation techniques. These techniques have been introduced by Lelarsmee et al. [4] and investigated by many authors for different kinds of differential equations, see, for example, [3] and [5] for systems of ordinary differential equations, [1] and [2] for systems of delay differential equations and [6] and [7] for general functional differential equations. However, these techniques have been mainly investigated in the context of parallel computations and not much attention has been given to answer the question of whether or not permutations of the equations in a given system influence the convergence of the applied dynamic iterations. Recent investigations [8] in this direction for linear systems of differential equations show that appropriately chosen permutations, in light of the values of the model parameters, present a way to speed up the convergence of dynamic iterations.

B. Zubik-Kowal (✉)

Department of Mathematics, Boise State University, Boise, ID, USA

e-mail: bzubik@boisestate.edu

The goal of the current paper is to address this question for nonlinear differential equations.

In this paper, we investigate dynamic iterations for systems written in the form

$$\begin{cases} \frac{d}{dt}x = ax + xf_1(y) + g_1(t) \\ \frac{d}{dt}y = \tilde{a}y + yf_2(x) + g_2(t) \end{cases} \quad (1)$$

where a, \tilde{a} are real parameters and $f_i, g_i, i = 1, 2$, are given real functions. The system (1) is supplemented by the initial conditions

$$x(0) = \xi_0, \quad y(0) = \eta_0.$$

For an arbitrary continuous function $y^{(0)}(t)$, we consider the sequences $\{x^{(k)}(t)\}_{k=1}^{\infty}, \{y^{(k)}(t)\}_{k=0}^{\infty}$ defined by

$$\begin{cases} \frac{d}{dt}x^{(k+1)} = ax^{(k+1)} + x^{(k+1)}f_1(y^{(k)}) + g_1(t), \\ \frac{d}{dt}y^{(k+1)} = \tilde{a}y^{(k+1)} + y^{(k+1)}f_2(x^{(k+1)}) + g_2(t), \end{cases} \quad (2)$$

where $k = 0, 1, 2, \dots$ and

$$x^{(k+1)}(0) = \xi_0, \quad y^{(k+1)}(0) = \eta_0.$$

The numerical scheme (2) is called Gauss-Seidel waveform relaxation. The advantage of (2) over (1) is that the application of implicit methods, for example BDF methods, for integration of (2) in time t does not require solving nonlinear algebraic equations at each time step. Note that the application of implicit time integration methods to the nonlinear differential system (1) leads to a system of nonlinear algebraic equations that require an additional process to solve them at each time step (more time steps mean that more nonlinear algebraic systems need to be solved). Such an additional process would not be needed if system (2) would be applied.

The paper is organized as follows. In Sect. 2, we analyze the convergence of the sequence $\{(x^{(k)}(t), y^{(k)}(t))\}_{k=0}^{\infty}$ to the exact solution $(x(t), y(t))$ as $k \rightarrow \infty$. Then, in Sect. 3, we present results of numerical experiments involving nonlinear systems applied in population dynamics. Finally, we finish with concluding remarks in Sect. 4.

2 Error Analysis

In this section, we analyze the errors

$$e_x^{(k)}(t) = x^{(k)}(t) - x(t), \quad e_y^{(k)}(t) = y^{(k)}(t) - y(t),$$

as $k \rightarrow \infty$. We assume that the unknown solutions x, y are bounded and the given functions f_1, f_2 are Lipschitz continuous. In what follows, we use the following notation. Let L_1, L_2 be Lipschitz constants for f_1 and f_2 , respectively; that is,

$$\begin{aligned} |f_1(y) - f_1(\tilde{y})| &\leq L_1|y - \tilde{y}|, & \text{for all } y, \tilde{y} \in \mathbb{R}, \\ |f_2(x) - f_2(\tilde{x})| &\leq L_2|x - \tilde{x}|, & \text{for all } x, \tilde{x} \in \mathbb{R} \end{aligned}$$

and X, Y, R, E_0 be positive constants such that

$$\begin{aligned} |x(t)| &\leq X, & |y(t)| &\leq Y, & \text{for all } 0 \leq t, \\ f_1(y) + a &\leq R, & f_2(x) + \tilde{a} &\leq R, & \text{on bounded sets,} \\ |e_y^{(0)}(t)| &\leq E_0, & \text{for all } 0 \leq t. \end{aligned}$$

The following theorem provides error bounds for scheme (2).

Theorem 1 *Let $k = 0, 1, 2, \dots$ and $t \geq 0$. Then,*

$$|e_x^{(k+1)}(t)| \leq E_0 X L_1 (X Y L_1 L_2)^k \frac{e^{Rt}}{R^{2k+1}} \sum_{j=2k+1}^{\infty} \frac{(-1)^{j+1} (Rt)^j}{j!}, \tag{3}$$

$$|e_y^{(k+1)}(t)| \leq E_0 (X Y L_1 L_2)^{k+1} \frac{e^{Rt}}{R^{2k+2}} \sum_{j=2k+2}^{\infty} \frac{(-1)^j (Rt)^j}{j!}. \tag{4}$$

Proof Note that $e_x^{(k)}(0) = 0$ and $e_y^{(k)}(0) = 0$, for all $k = 0, 1, 2, \dots$. Then, from (1) and (2), we get

$$|e_x^{(k+1)}(t)| \leq X L_1 \int_0^t |e_y^{(k)}(\tau)| e^{R(t-\tau)} d\tau, \tag{5}$$

$$|e_y^{(k+1)}(t)| \leq Y L_2 \int_0^t |e_x^{(k+1)}(\tau)| e^{R(t-\tau)} d\tau, \tag{6}$$

for $k = 0, 1, 2, \dots$ and $t \geq 0$. From (5), we get

$$\begin{aligned} |e_x^{(1)}(t)| &\leq X L_1 E_0 \int_0^t e^{R(t-\tau)} d\tau = \frac{X L_1 E_0}{R} (e^{Rt} - 1) = \frac{X L_1 E_0}{R} e^{Rt} \left(1 - \sum_{j=0}^{\infty} \frac{(-Rt)^j}{j!}\right) \\ &= \frac{X L_1 E_0}{R} e^{Rt} \sum_{j=1}^{\infty} \frac{(-1)^{j+1} (Rt)^j}{j!}, \end{aligned}$$

which shows (3) for $k = 0$. We now use (6) and get

$$\begin{aligned}
|e_y^{(1)}(t)| &\leq \frac{XY L_1 L_2 E_0}{R} e^{Rt} \int_0^t \sum_{j=1}^{\infty} \frac{(-1)^{j+1} (R\tau)^j}{j!} d\tau \\
&= \frac{XY L_1 L_2 E_0}{R} e^{Rt} \sum_{j=1}^{\infty} \frac{(-1)^{j+1} R^j t^{j+1}}{(j+1)!} = \frac{XY L_1 L_2 E_0}{R^2} e^{Rt} \sum_{j=2}^{\infty} \frac{(-Rt)^j}{j!},
\end{aligned}$$

which shows (4) for $k = 0$. We now assume (3) and (4) for a certain k . Then, from (5) and (4), we get

$$\begin{aligned}
|e_x^{(k+2)}(t)| &\leq X L_1 \int_0^t (XY L_1 L_2)^{k+1} E_0 \frac{e^{R\tau}}{R^{2k+2}} \sum_{j=2k+2}^{\infty} \frac{(-R\tau)^j}{j!} e^{R(t-\tau)} d\tau \\
&= X L_1 (XY L_1 L_2)^{k+1} E_0 \frac{e^{Rt}}{R^{2k+2}} \sum_{j=2k+2}^{\infty} \frac{(-R)^j t^{j+1}}{(j+1)!} \\
&= X L_1 (XY L_1 L_2)^{k+1} E_0 \frac{e^{Rt}}{R^{2k+3}} \sum_{j=2k+3}^{\infty} \frac{(-1)^{j+1} (Rt)^j}{j!},
\end{aligned}$$

which, by mathematical induction, shows (3). We now use (6) and (3) and obtain the following result,

$$\begin{aligned}
|e_y^{(k+2)}(t)| &\leq Y L_2 \int_0^t X L_1 (XY L_1 L_2)^{k+1} E_0 \frac{e^{R\tau}}{R^{2k+3}} \sum_{j=2k+3}^{\infty} \frac{(-1)^{j+1} (R\tau)^j}{j!} e^{R(t-\tau)} d\tau \\
&= (XY L_1 L_2)^{k+2} E_0 \frac{e^{Rt}}{R^{2k+4}} \sum_{j=2k+4}^{\infty} \frac{(-Rt)^j}{j!},
\end{aligned}$$

which shows (4) and finishes the proof. \square

3 Numerical Experiments and Methods Comparison

In this section, we present results of numerical experiments involving dynamic iterations applied to Volterra equations for predator-prey interactions.

The system of interest is of the form

$$\begin{cases} \frac{d}{dt}x = ax - bxy + g_1(t) \\ \frac{d}{dt}y = -cy + dxy + g_2(t) \end{cases} \quad (7)$$

where $a = 2/3, b = 20, c = 50, d = 0.01, 0 \leq t \leq 10$. We apply (2) and obtain the following scheme

$$\begin{cases} \frac{d}{dt}x^{(k+1)} = ax^{(k+1)} - bx^{(k+1)}y^{(k)} + g_1(t) \\ \frac{d}{dt}y^{(k+1)} = -cy^{(k+1)} + dx^{(k+1)}y^{(k+1)} + g_2(t). \end{cases} \tag{8}$$

If we write the equations in system (7) in the opposite order and then apply (2), we obtain a different scheme of the form

$$\begin{cases} \frac{d}{dt}y^{(k+1)} = -cy^{(k+1)} + dx^{(k)}y^{(k+1)} + g_2(t) \\ \frac{d}{dt}x^{(k+1)} = ax^{(k+1)} - bx^{(k+1)}y^{(k+1)} + g_1(t). \end{cases} \tag{9}$$

Numerical solutions $x^{(k)}(t_n)$ and $y^{(k)}(t_n)$ computed by (8) and (9) are presented in Fig. 1 as functions of t_n for $k = 6$ in the case of (8) and for $k = 4$ in the case of (9).

Although both schemes (8) and (9) originate from the same application of (2), their errors are different and demonstrate different convergence rates. The errors of both schemes are presented in Fig. 2. The upper subplot demonstrates the errors resulting from the application of (8) and the lower subplot demonstrates the errors resulting from the application of (9). Each scheme is integrated by BDF3 with $h = 10^{-4}$.

We now compare the accuracy and CPU time using the solver `ode15s` and scheme (9) integrated by BDF6 with $h = 10^{-2}$. The maximum error

$$\max \left\{ \max_n |x_1(t_n) - x_{1,n}^{(k)}|, \max_n |x_2(t_n) - x_{2,n}^{(k)}| \right\}$$

is $7.03 \cdot 10^{-13}$ and $1.91 \cdot 10^{-14}$ using the solver `ode15s` and (9), respectively. The errors resulting from the application of both methods are comparable. The CPU

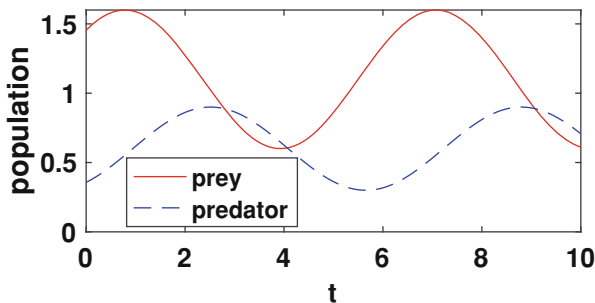


Fig. 1 Numerical solutions of (7): $x(t)$ solid and $y(t)$ dashed

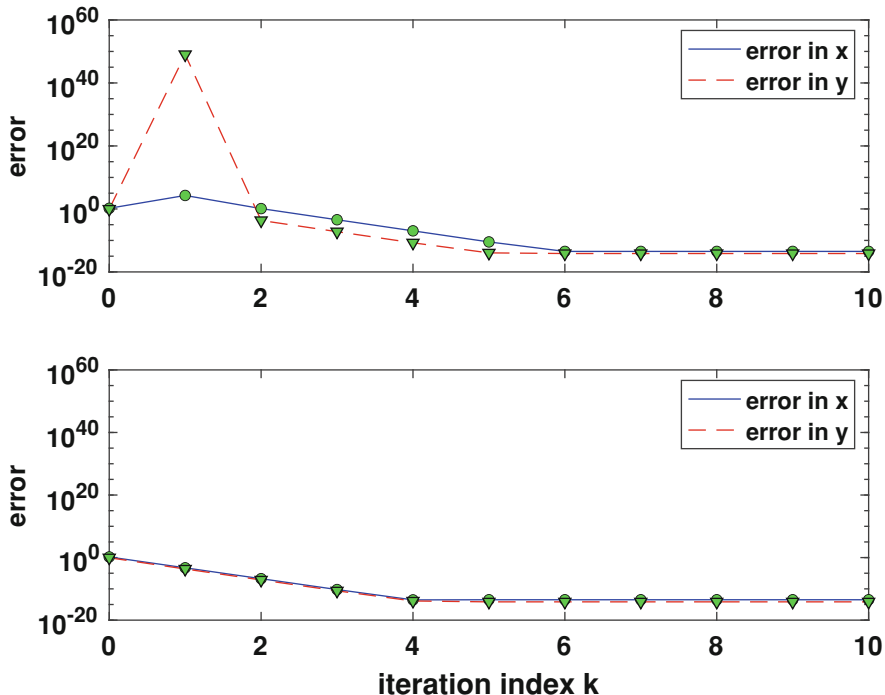


Fig. 2 Methods comparison: errors by (8) (upper subplot) and errors by (9) (lower subplot)

time is 0.21 s when the solver `ode15s` is applied and it is 0.03 s when the numerical scheme (9) is applied, demonstrating that (9) is faster than `ode15s`.

4 Conclusions

In this work, we investigate dynamic iterations for nonlinear systems of differential equations applied in population dynamics. The advantage of dynamic iterations is that they allow to apply implicit time integration methods without the cost of solving nonlinear algebraic equations at each time step. We conclude that the convergence of dynamic iterations is different if we swap the order of the nonlinear differential equations in the given system even though the iterations are applied to the same system. That is, only by swapping the order of the equations, we can increase the rate of the convergence of the iterations. We also conclude that after choosing the optimal permutation of the equations, the proposed numerical scheme based on dynamical iterations is faster than the variable order method.

References

1. Bjorhus, M.: On dynamic iteration for delay differential equations. *BIT* **43**, 325–336 (1994)
2. Bjorhus, M.: A note on the convergence of discretized dynamic iteration. *BIT* **35**, 291–296 (1995)
3. Burrage, K.: *Parallel and Sequential Methods for Ordinary Differential Equations*. Oxford University Press, Oxford (1995)
4. Lelarasme, E., Ruehli, A., and Sangiovanni-Vincentelli, A.: The waveform relaxation method for time-domain analysis of large scale integrated circuits, *IEEE Trans. CAD*, **1**, 131–145 (1982)
5. Miekkala, U., and Nevanlinna, O.: Iterative solution of systems of linear differential equations. *Acta Numerica*, 259–307 (1996)
6. Zubik-Kowal, B., and Vandewalle, S.: Waveform relaxation for functional-differential equations. *SIAM J. Sci. Comput.* **21**, 207–226 (1999)
7. Zubik-Kowal, B.: Error bounds for spatial discretization and waveform relaxation applied to parabolic functional differential equations. *J. Math. Anal. Appl.* **293**, 496–510 (2004)
8. Zubik-Kowal, B.: Error analysis and the role of permutation in dynamic iteration schemes. *Computational and Analytic Methods in Science and Engineering*, 239–256, Birkhäuser/Springer, Cham, (2020)