COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. **5**, No. 2-4, pp. 779-792

Commun. Comput. Phys.
February 2009

Accuracy Enhancement Using Spectral Postprocessing for Differential Equations and Integral Equations

Tao Tang[1,*] and Xiang Xu[2]

[1] *Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.*
[2] *School of Mathematics, Fudan University, Shanghai 200433, China.*

Received 4 October 2007; Accepted (in revised version) 28 May 2008

Available online 1 August 2008

Abstract. It is demonstrated that spectral methods can be used to improve the accuracy of numerical solutions obtained by some lower order methods. More precisely, we can use spectral methods to postprocess numerical solutions of initial value differential equations. After a few number of iterations (say 3 to 4), the errors can decrease to a few orders of magnitude less. The iteration uses the Gauss-Seidel type strategy, which gives an explicit way of postprocessing. Numerical examples for ODEs, Hamiltonian system and integral equations are provided. They all indicate that the spectral processing technique can be a very useful way in improving the accuracy of the numerical solutions. In particular, for a Hamiltonian system accuracy is only one of the issues; some other conservative properties are even more important for large time simulations. The spectral postprocessing with the coarse-mesh symplectic initial guess can not only produce high accurate approximations but can also save a significant amount of computational time over the standard symplectic schemes.

AMS subject classifications: 35Q99, 35R35, 65M12, 65M70

Key words: Postprocessing, spectral methods, rate of convergence, Hamiltonian system, integral equations.

1 Introduction

We begin by considering a simple ordinary differential equation with given initial value:

$$y'(x) = g(y;x), \quad 0 < x \leq T, \tag{1.1}$$

$$y(0) = y_0. \tag{1.2}$$

*Corresponding author. *Email addresses:* ttang@math.hkbu.edu.hk (T. Tang), xuxiang@fudan.edu.cn (X. Xu)

There have been many numerical methods for solving (1.1)-(1.2), see, e.g., [9, 10]. However, most of the existing methods have an algebraic rate of convergence, i.e., $\mathcal{O}(h^\alpha)$, with $\alpha=1$ for the Euler method, and $\alpha=4$ for the RK4 method.

A natural question is can we obtain exponential (spectral) rate of convergence for solving problem (1.1)-(1.2)? For boundary value problems, the answer is positive and well known, see, e.g., [2, 4, 14]. However, for the initial value problem (1.1)-(1.2), spectral methods are not attractive due to the following reasons: The problem (1.1)-(1.2) is a local problem, so a global method (such as spectral method) will require larger storage (need to store all data in a fixed interval) and computational time (need to solve a linear system or a *nonlinear* system in case that F in (1.1) is nonlinear). These disadvantages makes the use of the spectral approach for problem (1.1)-(1.2) less attractive.

The motivation of this article is to propose a *spectral postprocessing technique* which uses the numerical solutions of a lower order method to serve as starting value of the spectral methods. Then we take a few Gauss-Seidel type iterations for a well designed spectral method. This postprocessing procedure will help us to recover the exponential rate of convergence with little extra computational resource. In particular, there is no need of solving a linear system or a nonlinear system in case that F in (1.1) is nonlinear. Moreover, the method is found extremely stable for the initial value problem (1.1)-(1.2).

2 Spectral postprocessing for initial value ODEs

2.1 Spectral postprocessing for an ODE equation

Assume the size of $[t_0,T]$ is not too big; otherwise a trick in Section 2.2 will be used. In this case, we introduce the linear coordinate transformation

$$x=\frac{T-t_0}{2}s+\frac{T+t_0}{2},\quad -1\le s\le 1, \tag{2.1}$$

and the transformations

$$Y(s)=y\left(\frac{T-t_0}{2}s+\frac{T+t_0}{2}\right),\quad G(Y;s)=g\left(Y;\frac{T-t_0}{2}s+\frac{T+t_0}{2}\right). \tag{2.2}$$

Then problem (1.1)-(1.2) becomes

$$Y'(s)=G(Y;s),\quad -1<s\le 1; \tag{2.3}$$

$$Y(-1)=y_0. \tag{2.4}$$

2.1.1 Chebyshev collocation approach

Let $\{s_j\}_{j=0}^N$ be the Chebyshev-Gauss-Lobatto points: $s_j=\cos(\pi j/N), 0\le j\le N$. We project G to the polynomial space $\mathcal{P}_N$:

$$G(Y;s)=\sum_{j=0}^{N} G(Y_j;s_j)F_j(s), \tag{2.5}$$

where F_j is the j-th Lagrange interpolation polynomial associated with the Chebyshev-Gauss-Lobato points. Since $F_j \in \mathcal{P}_N$, it can be expanded by the Chebyshev basis functions:

$$F_j(s) = \sum_{m=0}^{N} \alpha_{mj} T_m(s). \tag{2.6}$$

Assume (2.6) is satisfied in the collocation points $\{s_i\}_{i=0}^N$, i.e.,

$$F_j(s_i) = \sum_{m=0}^{N} \alpha_{mj} T_m(s_i), \quad 0 \le i \le N, \tag{2.7}$$

which gives (see (1.3.17) of [14])

$$\alpha_{mj} = \frac{2}{N\tilde{c}_m} \sum_{i=0}^{N} \frac{1}{\tilde{c}_i} F_j(s_i) \cos\left(\frac{im\pi}{N}\right) = \frac{2}{N\tilde{c}_m\tilde{c}_j} \cos\left(\frac{jm\pi}{N}\right),$$

where $\tilde{c}_0 = \tilde{c}_N = 2$ and $\tilde{c}_m = 1$ for $1 \le m \le N-1$. Using the above result and (2.3)-(2.6) gives

$$Y(s_i) = y_0 + \frac{2}{N} \sum_{j=0}^{N} \sum_{m=0}^{N} G(Y_j; s_j) \frac{1}{\tilde{c}_m\tilde{c}_j} \cos\left(\frac{jm\pi}{N}\right) \int_{-1}^{s_i} T_m(s) ds. \tag{2.8}$$

Using the relation

$$2T_n(x) = \frac{1}{n+1} T'_{n+1}(x) - \frac{1}{n-1} T'_{n-1}(x),$$

we finally obtain the following numerical scheme

$$Y_i = y_0 + \sum_{j=0}^{N} \omega_{ij} G(Y_j; s_j), \tag{2.9}$$

where

$$\omega_{ij} = \frac{1}{N\tilde{c}_j} \sum_{m=0}^{N} \frac{1}{\tilde{c}_m} \cos\left(\frac{jm\pi}{N}\right) \left[\frac{1}{m+1} T_{m+1}(s_i) - \frac{1}{m-1} T_{m-1}(s_i) + \frac{2(-1)^m}{m^2-1}\right]. \tag{2.10}$$

It is noticed that $T_{m\pm1}(s_i) = \cos((i(m\pm1)\pi/N)$.

2.1.2 Legendre collocation approach

Let $\{x_j\}_{j=0}^N$ be the Legendre-Gauss-Lobatto points, i.e., $x_0 = -1, x_N = 1$ and x_j $(1 \le j \le N-1)$ be the roots of $L'_N(x)$. We expand $F_j(s)$ in (2.5) by

$$F_j(s) = \sum_{m=0}^{N} \beta_{mj} L_m(s). \tag{2.11}$$

Assuming the above equation holds at the collocation points $\{x_j\}_{j=0}^N$, we have (see, e.g., (1.3.30) of [14]) that

$$\beta_{mj}=\frac{1}{N+1}\sum_{i=0}^{N}F_j(x_i)\frac{L_m(x_i)}{L_N(x_i)}=\frac{1}{N+1}\frac{L_m(x_j)}{L_N(x_j)}. \tag{2.12}$$

Using the relation

$$L_n(x)=\frac{1}{2n+1}\left(L'_{n+1}(x)-L'_{n-1}(x)\right), \tag{2.13}$$

and (2.11)-(2.12), we obtain the following numerical scheme

$$Y_i=y_0+\sum_{j=0}^{N}w_{ji}G(Y_j,x_j), \tag{2.14}$$

where

$$w_{ji}=\frac{1}{N+1}\sum_{m=0}^{N}\frac{L_m(x_j)}{L_N(x_j)}\frac{1}{2m+1}\Big[L_{m+1}(x_i)-L_{m-1}(x_i)\Big]. \tag{2.15}$$

It is known that the size of the domain $[t_0,T]$ of (1.1) has an impact on the convergence of the method. In particular, it involves computations of too many points in formulas (2.9) and (2.14), even for the points very close to t_0. To fully utilize the advantage of the initial value problems, it is reasonable to partition the integration interval into sub-intervals with smaller size, as will be demonstrated in Section 2.2.

It is pointed out that the methods in Section 2.1.2 are the *Labatto IIIA* methods, see, e.g., [5].

2.1.3 Numerical examples

Our first example is a linear problem with exact solution.

Example 2.1. Consider a simple example

$$\begin{aligned} &y'=y+\cos(x+1)e^{x+1}, \quad x\in(-1,1],\\ &y(-1)=1. \end{aligned} \tag{2.16}$$

The exact solution of the above problem is $y=(1+\sin(x+1))\exp(x+1)$.

We use the first order explicit Euler method to solve the problem first (with a fixed mesh size $h=0.1$). Then we use the spectral postprocessing formula (2.14) to update the solutions using the Gauss-Seidel type iterations. The maximum error after using the Euler method is about 0.6663; however, after 6 iterations, the error is reduced to 10^{-10} with about 14 collocation points. The result can be seen from Fig. 1 (a).

We then use the 2nd-order Runge-Kutta method to obtain the initial value. The maximum error after using RK2 is about 0.0157. After 6 iterations, the error is reduced to 10^{-12} with about 14 collocation points. The result can be seen from Fig. 1 (b).

Finally, we use the 4th-order Runge-Kutta method to obtain the initial value. The maximum error after using RK4 is about $1.45*10^{-6}$. After 6 iterations, the error is reduced to 10^{-14} with about 14 collocation points. The result can be seen from Fig. 1 (c).

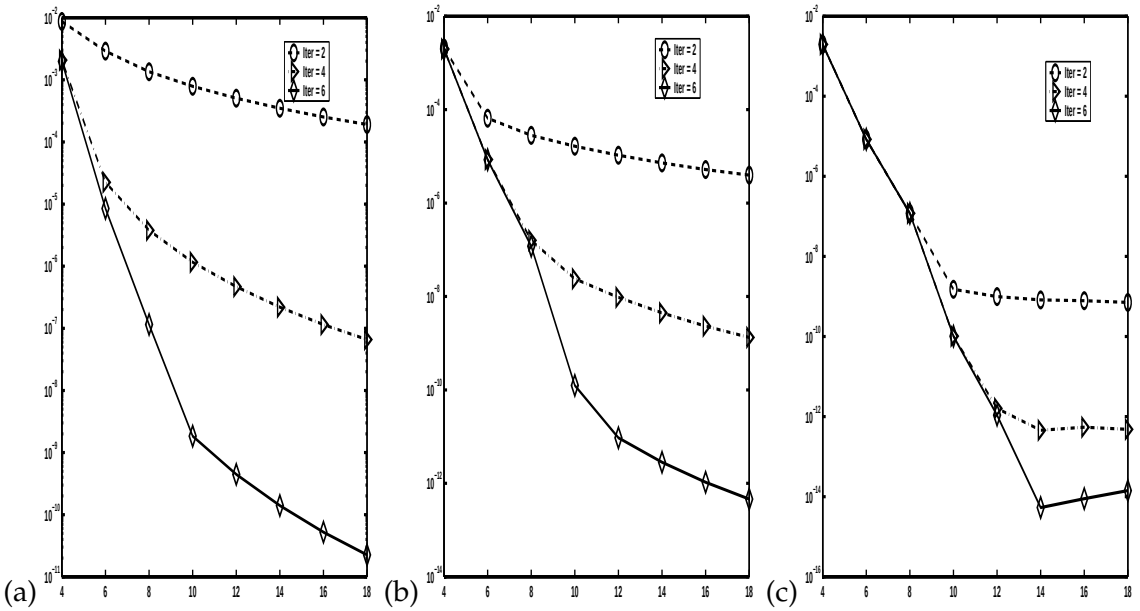


Figure 1: Example 2.1: errors vs N_s for spectral postprocessing method (2.14), with (a): Euler, (b): RK2, and (c): RK4 solutions as the initial data.

2.2 Spectral postprocessing for Hamiltonian systems

As an application, we apply in this section the spectral postprocessing technique for the Hamiltonian system:

$$\begin{aligned} &\frac{dp}{dt} = \partial_q H(p,q), \\ &\frac{dq}{dt} = -\partial_p H(p,q), \qquad t_0 \le t \le T, \end{aligned} \tag{2.17}$$

with the initial value $p(t_0)=p_0$, $q(t_0)=q_0$. First we review some traditional methods such as Runge-Kutta method and symplectic method for (2.17) (see [8,11,12]). These methods are explicit and easy for implementation. We denote $p^n \approx p(t_n)$ and $q^n \approx q(t_n)$.

- 4th-order explicit Runge-Kutta: method

$$\begin{aligned} &p_1 = \Big(\frac{\partial H}{\partial q}\Big)_{q^n},\ q_1 = -\Big(\frac{\partial H}{\partial p}\Big)_{p^n},\ p_2 = \Big(\frac{\partial H}{\partial q}\Big)_{q^n+\frac{\Delta t}{2}q^1},\ q_2 = -\Big(\frac{\partial H}{\partial p}\Big)_{p^n+\frac{\Delta t}{2}p^1}; \\ &p_3 = \Big(\frac{\partial H}{\partial q}\Big)_{q^n+\frac{\Delta t}{2}q^2},\ q_3 = -\Big(\frac{\partial H}{\partial p}\Big)_{p^n+\frac{\Delta t}{2}p^2},\ p_4 = \Big(\frac{\partial H}{\partial q}\Big)_{q^n+\frac{\Delta t}{2}q^3},\ q_4 = -\Big(\frac{\partial H}{\partial p}\Big)_{p^n+\frac{\Delta t}{2}p^3}; \\ &p^{n+1} = p^n + \frac{\Delta t}{6}(p_1+2p_2+2p_3+p_4),\quad q^{n+1} = q^n + \frac{\Delta t}{6}(q_1+2q_2+2q_3+q_4). \end{aligned} \tag{2.18}$$

- 4th-order explicit symplectic method:

$$p_1=p^n+c_1\tau\Big(\frac{\partial H}{\partial q}\Big)_{q^n},\quad q_1=q^n-d_1\tau\Big(\frac{\partial H}{\partial p}\Big)_{p_1},\quad p_2=p_1+c_2\tau\Big(\frac{\partial H}{\partial q}\Big)_{q_1};$$
$$q_2=q_1-d_2\tau\Big(\frac{\partial H}{\partial p}\Big)_{p_1},\quad p_3=p_2+c_3\tau\Big(\frac{\partial H}{\partial q}\Big)_{q_2},\quad q_3=q_2-d_3\tau\Big(\frac{\partial H}{\partial p}\Big)_{p_2};$$
$$p_{n+1}=p_3+c_4\tau\Big(\frac{\partial H}{\partial q}\Big)_{q_3},\quad q^{n+1}=q_3-d_4\tau\Big(\frac{\partial H}{\partial p}\Big)_{p^{n+1}},\tag{2.19}$$

where $c_1=0, c_2=c_4=\alpha, c_3=\beta, d_1=d_4=\alpha/2, d_2=d_4=(\alpha+\beta)/2$; or $c_1=c_4=\alpha/2$, $c_2=c_3=(\alpha+\beta)/2$, $d_1=d_3=\alpha, d_2=\beta, d_4=0$. Here $\alpha=(2-\sqrt[3]{2})^{-1}$, $\beta=1-2\alpha$.

Since the Hamiltonian system needs *large time* evolution to reach a stable state, we partition the time interval $[t_0,T]$ into some sub-intervals of length 2. This partition makes the spectral collocation and time marching easier. Set $t_k=t_0+2k$ and $I_k=[t_k,t_{k+1}]$. So problem (2.17) becomes

$$\begin{aligned}&\frac{dp}{dt}=\partial_q H(p,q),\quad \frac{dq}{dt}=-\partial_p H(p,q),\quad t\in I_k,\\ &p(t_k)=p_k,\quad q(t_k)=q_k.\end{aligned}\tag{2.20}$$

Integrating (2.20) leads to a system of integral equation

$$p(t)=p_k+\int_{t_k}^{t}\partial_q H(p,q)ds,\quad q(t)=q_k-\int_{t_k}^{t}\partial_p H(p,q)ds.\tag{2.21}$$

Assume that we already have a coarse solution obtained from some difference schemes (say Runge-Kutta method or symplectic methods). Assume (2.21) holds at the Legendre or Chebyshev collocation points:

$$p(t_{kj})=p_k+\int_{t_k}^{t_{kj}}\partial_q H(p,q)ds,\quad q(t_{kj})=q_k-\int_{t_k}^{t_{kj}}\partial_p H(p,q)ds,\tag{2.22}$$

where $t_{kj}=(t_k+1)+\theta_j, 0\le j\le N$. Here $\theta_j\in[-1,1]$ are the Legendre or Chebyshev collocation points. The corresponding spectral scheme for the above system can be obtained by following the approach (2.9)-(4.1) or (2.14)-(2.15).

2.2.1 Numerical Examples

Example 2.2. Consider the Hamiltonian problem (2.17) with $H(p,q)=(p^2+q^2)/2$ and $p(t_0)=\sin(t_0), q(t_0)=\cos(t_0)$. This system has an exact solution $(p,q)=(\sin t,\cos t)$.

We take $T=1000$ in our computations. Since the exact solution of (p,q) is known, the maximum errors can be computed after obtaining the numerical solutions. Table 1(a) presents the maximum error in $t\in[0,1000]$ using both the RK4 method (2.18) and the symplectic method (2.19).

Table 1: Example 2.2. (a): the maximum errors obtained by RK4 and the symplectic method; (b): spectral postprocessing results using the RK4 (2.18) ($\Delta t=0.1$) as the initial data in each sub-interval $[t_k,t_k+2]$; (c): same as (b), except that RK4 is replaced by the symplectic method (2.19). Here N denotes the number of spectral collocation points used.

(a)	RK4 Max. Error	CPU time	Symplectic Max. Error	CPU time
$\Delta t=10^{-1}$	blow up	–	9.20e-03	0.16s
$\Delta t=10^{-2}$	1.81e-0	1.60s	1.32e-06	1.52s
$\Delta t=10^{-3}$	2.43e-2	12.02s	9.16e-11	10.60s

(b)	iter step = 3 Max. Error	CPU time	iter step =6 Max. Error	CPU time
$N=8$	1.74e-2	1.796s	5.49e-07	1.828s
$N=10$	1.74e-2	1.813s	5.49e-07	1.843s
$N=12$	1.74e-2	1.828s	5.49e-07	1.859s

(c)	iter step = 3 Max. Error	CPU time	iter step =6 Max. Error	CPU time
$N=8$	2.23e-5	1.797s	7.07e-10	1.843s
$N=10$	2.17e-5	1.812s	6.83e-10	1.862s
$N=12$	2.14e-5	1.860s	6.81e-10	1.906s

Table 1(b) shows the performance of the postprocessing procedure with initial data in $[t_k,t_k+2]$ generated by using RK4 ($\Delta t=0.1$). It is seen that with $N=8$ this postprocessing improve the numerical errors the order to 10^{-2} (after 3 iterations) and finally to the order to 10^{-7} (after 6 iterations). Without using the spectral iterations, the approximation is blow up with $\Delta t=0.1$.

If we use the symplectic scheme (2.19) with $\Delta t=0.1$ to produce the starting value in each $[t_k,t_k+2]$, then the accuracy of the spectral postprocessing is improved. It is observed from Table 1(c) that with $N=8$ this postprocessing changes the numerical errors from 0.0092 (no interaction) to the order to 10^{-5} (after 3 iterations) and finally to the order to 10^{-10} (after 6 iterations). To reach the same accuracy of about 10^{-10}, the symplectic scheme without postprocessing requires about 5 times more CPU time.

In Fig. 2, we show the numerical errors against the number of spectral collocation points using different number of iterations. Again it is observed that spectral accuracy is recovered after a few postprocessing iterations. It is also seen that more accurate results are obtained if the initial guess is given by the symplectic methods.

In the following example, we consider a problem proposed by Shen and Wang [16] where a set of Fourier-like basis functions is constructed for Legendre-Galerkin method and a new space-time spectral method is proposed.

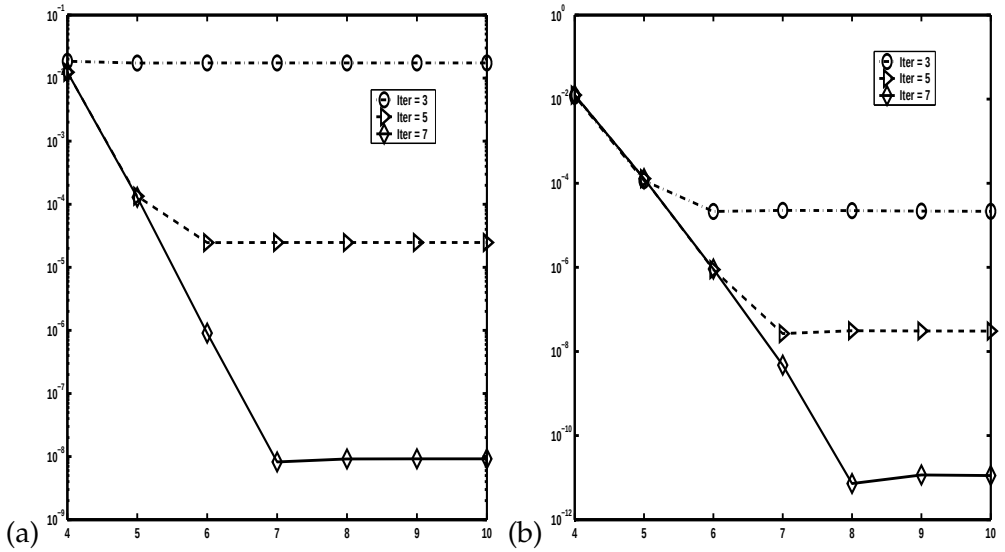


Figure 2: Example 2.2: errors vs N_s and iterative steps with (a): RK4 results and (b): symplectic results as the initial data.

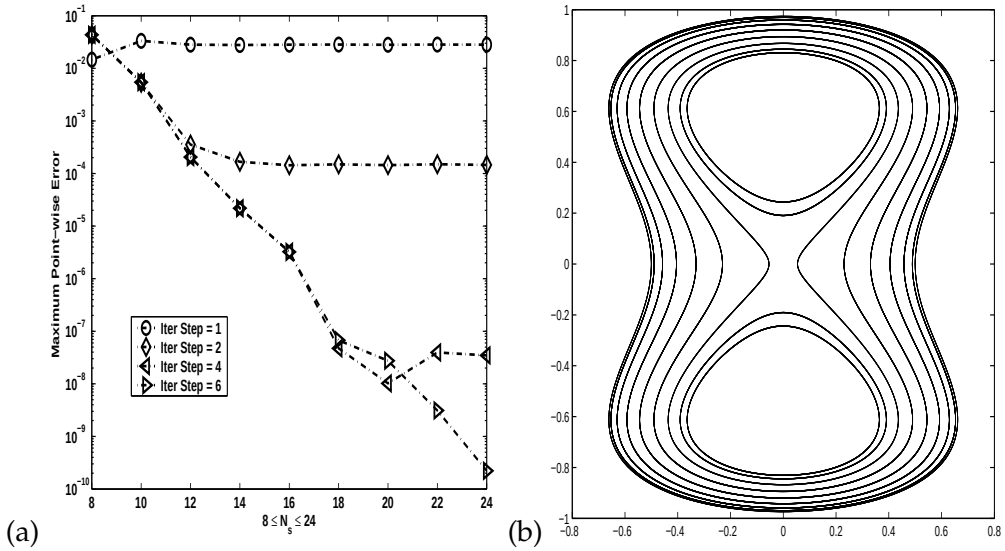


Figure 3: Example 2.3: (a) errors vs N_s and iterative steps and (b) numerical solutions with multiple initial values.

Example 2.3. In the following example, we take

$$H(p,q)=\frac{1}{2}q^2-\frac{2}{3}q^4-\frac{1}{2}p^2.$$

The initial condition is chosen as $p(t_0)=2a\cos(2t_0), q(t_0)=a\sin(2t_0), a=0.25t_0\in[0,\pi)$.

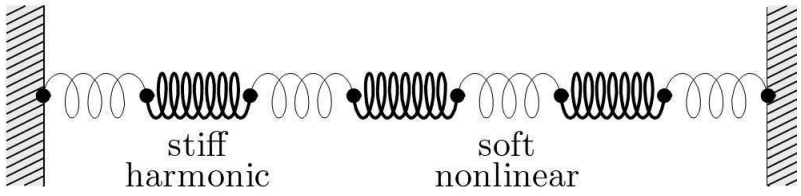


Figure 4: Example 2.4: Model of Fermi, Pasta & Ulam from [9].

This system does not have an exact solution for p and q, but its Hamiltonian is conserved in the sense that

$$H(p(t),q(t)) \equiv H(p_0,q_0). \tag{2.23}$$

Fig. 3 (a) shows the error between the *exact* solution and the numerical solution obtained by the proposed spectral postprocessing method, where the *exact* solution is obtained by using the 4th-order explicit symplectic scheme (2.19) with $\Delta t = 10^{-4}$. Moreover, Fig. 3 (b) gives the numerical solution of this system with multiple initial values, which is comparable with the published results [16].

Example 2.4. The Fermi-Pasta-Ulam Problem is a simple model for simulations in statistical mechanics which revealed highly unexpected dynamical behavior. Here we only consider a modification presented in [9]. It consists of a chain of m mass points, connected with alternating soft nonlinear and stiff linear springs, and fixed at the end points; see Fig. 4. We can form a Hamiltonian system from this model [9]

$$\begin{aligned} H(y,x) = &\frac{1}{2}\sum_{i=0}^{2m} y_i^2 + \frac{\omega^2}{2}\sum_{i=1}^{m} x_{m+i}^2 + \frac{1}{4}\Big((x_1 - x_{m+1})^4 \\ &+ \sum_{i=1}^{m-1}(x_{i+1} - x_{m+i+1} - x_i - x_{m+i})^4 + (x_m + x_{2m})^4\Big), \end{aligned} \tag{2.24}$$

where x_i ($i = 1,\cdots,m$) represents a scaled displacement of i-th stiff spring, x_{m+i} a scaled expansion of the i-th stiff spring, and y_i, y_{m+i} their momenta, and ω is assumed to be large. The initial values are 0 except $x_1(0) = 1, y_1(0) = 1, x_{m+1}(0) = \omega^{-1}, y_{m+1}(0) = 1$.

The equations of motion are Hamiltonian system, so the total energy is exactly conserved, i.e., $H(y,x) = H(y(0),x(0))$. Moreover, there is another interesting feature which is the energy exchanged between stiff springs. Assume that the energy of the j-th stiff spring is defined by

$$I_j(x_{m+j},y_{m+j}) = \frac{1}{2}(y_{m+j}^2 + \omega^2 x_{m+j}^2). \tag{2.25}$$

It is known that the total oscillatory energy $I = \sum_{j=1}^{m} I_j$ remains close to a constant

$$I(y,x) = I(y(0),x(0)) + o(\frac{1}{\omega}),$$

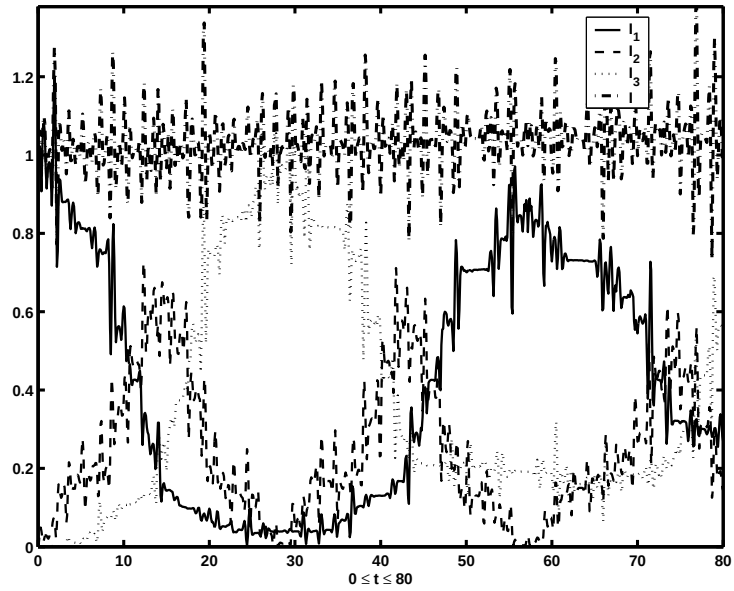


Figure 5: Example 2.4: the oscillatory energies I_1, I_2, I_3 and I, with $m=3$, $\omega=10$.

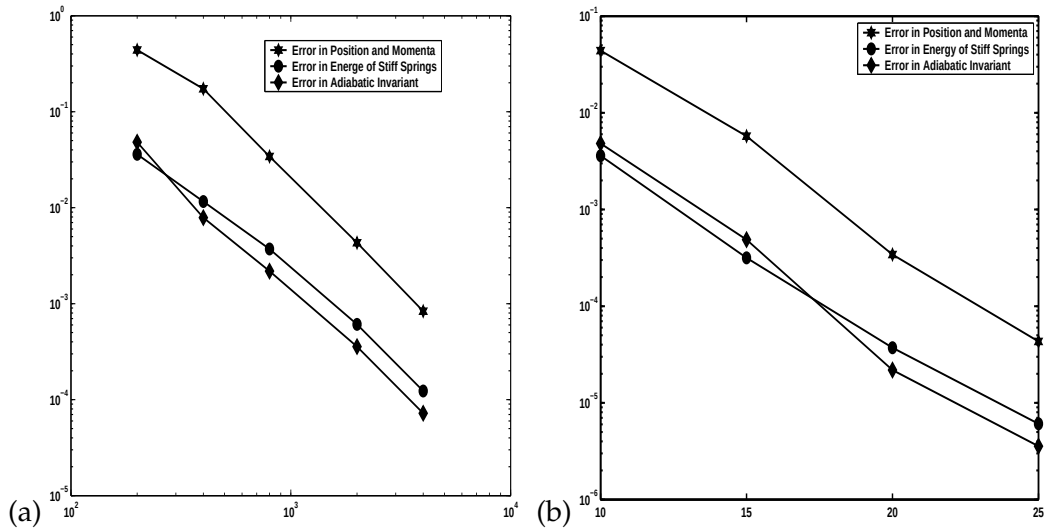


Figure 6: Example 2.4: Maximum error by using (a) implicit midpoint scheme and (b) 4 postprocessing iterations, with the initial guess given by the implicit midpoint with $\Delta t=10^{-2}$.

which can be seen from Fig. 5.

In the following numerical experiment we choose $m=3$, $\omega=10$, and $t\in[0,80]$. From Fig. 6 we can see that the traditional implicit mid points method can only give us an algebraic rate of convergence (about Δt^2), but base on this method we can improve the errors from 10^{-2} (implicit mid points) to 10^{-5} (after 4 iterations).

3 Spectral postprocessing for Volterra integral equations

In [15], Legendre spectral method is proposed and analyzed for Volterra type integral equations:

$$u(x)+\int_a^x k(x,s,u(s))ds=g(x), \qquad x\in[a,b] \tag{3.1}$$

where the kernel k and the source term g are given. We will show that the spectral postprocessing technique introduced in this work can be used to speed up the convergence of a standard method (such as the Trapezoidal method or a collocation method [3, 13, 17]).

Let $\{\theta_i\}_{i=0}^{N_s}$ be the zeros of Legendre polynomials of degree N_s+1, i.e., $L_{N_s+1}(x)$. Then the spectral collocation points are $x_i^s=(b-a)\theta_i/2+(b+a)/2$. We collocate (3.1) at the above points:

$$u(x_i^s)=g(x_i^s)-\int_a^{x_i^s} k(x_i^s,s,u(s))ds, \qquad 0\le i\le N_s. \tag{3.2}$$

Using the linear transform

$$s(\theta)=\frac{x-a}{2}\theta+\frac{x+a}{2}, \qquad s_i(\theta)=\frac{x_i^s-a}{2}\theta+\frac{x_i^s+a}{2}, \qquad -1\le\theta\le 1 \tag{3.3}$$

and the spectral approach of [15], we have

$$u(x_i^s)=g(x_i^s)-\frac{x_i^s-a}{2}\sum_k^{N_s} k\Big(x_i^s,s_i(\theta_k),u(s_i(\theta_k))\Big)w_k, \tag{3.4}$$

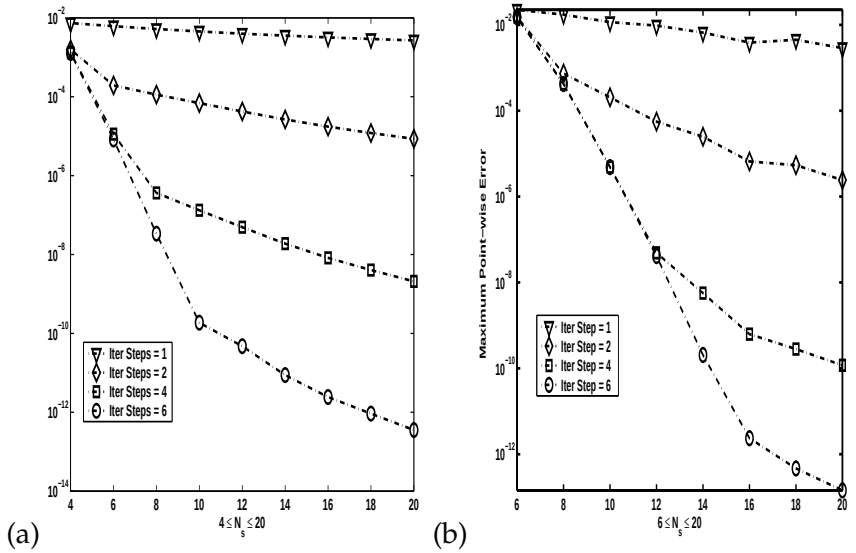
where the weights $\{w_j\}$ and the detailed implementation of Eq. (3.4) can be found in [15].

We use the Trapezoidal method to obtain numerical solution for (3.1) on the even distributed grid $x_i^d=a+ih/N_d$, where $h=(b-a)/N_d$. This gives a set of approximation $\{u(x_i^d)\}_{i=0}^{N_d}$. Here in order to calculate $u(s_i(\theta_k))$, we do not use interpolation with $\{u(x_i^d)\}_{i=0}^{N_d}$ directly. Instead, we first use these $\{u(x_i^d)\}_{i=0}^{N_d}$ to interpolate $\{u(x_i^s)\}_{i=0}^{N_s}$, then we use $\{u(x_i^s)\}_{i=0}^{N_s}$ to interpolate $u(s_i(\theta_k))$. This method takes the advantage that the newly updated $u(x_i^s)$ can be used immediately; i.e., we can use the Gauss-Seidel type iterations.

Table 2: Example 3.1: the maximum point-wise error vs the number of spectral collocation points after 4 iterations.

N	6	8	10	12
error	1.13e-5	3.63e-7	1.32e-7	4.91e-8
N	14	16	18	20
error	1.88e-8	8.22e-9	4.03e-9	2.08e-9

Example 3.1. Consider Eq. (3.1) with $k(x,s,u)=2\tan(u)/(1+x^2+s^2)$, $a=-1$, $b=1$, $g(x)=\arctan(x)+\ln(1+2x^2)-\ln(2+x^2)$. The exact solution of the above problem is $u(x)=\arctan(x)$.



(a) (b)

Figure 7: Errors vs N_s and iterative steps: (a) for Example 3.1; and (b): for Example 3.2.

With $N_d=200$ in the Trapezoidal method (which means $h=0.01$), the maximum point-wise error is $3.37*10^{-3}$. We then use (3.4) to postprocess the data obtained by the Trapezoidal method. Table 2 shows the maximum point-wise error between the exact solution and the approximated solutions after using 4 iterations. The errors against the number of iteration are plotted in Fig. 7(a).

Example 3.2. Consider a two-dimensional second-kind Volterra equation

$$u(x,y)+\int_a^x\int_a^y k(x,y,s,t,u(s,t))dtds=g(x,y),\qquad (x,y)\in[a,b]^2, \tag{3.5}$$

with

$$g(x,y)=\sin\pi(x+y)+\frac{(x+1)(y+1)}{1+x^2+y^2}-\frac{\cos4\pi(x+y)+\cos4\pi x+\cos4\pi y-1}{16\pi^2(1+x^2+y^2)},$$
$$k(x,y,s,t,u)=\frac{8\cos\pi(s+t)u^2}{1+x^2+y^2},\ a=-1,b=1.$$

This problem has a unique solution $u(x,y)=\sin\pi(x+y)$. With $N_d=50$ (i.e., $h=0.04$), the Trapezoidal method gives an error of $1.31*10^{-3}$. With a spectral postprocessing (the formula can be found in [15]), this error can be improved to the order of 10^{-10}. Table 3 presents the results with various spectral collocation points after 4 spectral postprocessing iterations. Fig. 7(b) shows the convergence rate versus the number of spectral collocation points with four different iterations. Clearly, the accuracy has been improved significantly after 4 or 6 iterations.

Table 3: Example 3.2: the maximum point-wise error vs the number of spectral collocation points after 4 iterations.

N	6	8	10	12
error	1.50e-02	4.21e-04	4.87e-06	4.88e-08
N	14	16	18	20
error	5.65e-09	6.24e-10	2.81e-10	1.18e-10

4 Conclusion

In this work, we proposed a method for enhancing the accuracy of the numerical solution to the initial value problems using the spectral postprocessing technique. The method uses the Gauss-Seidel iteration idea, so the resulting method is explicit which is easy to implement. The numerical experiments show that the postprocessing method is stable and efficient. The future research along this direction includes theoretical analysis of the proposed method and applications to more practical problems such as wave interaction problems and Lotka-Volterra population system [1, 6].

One of the main points of this work is to demonstrate that with the spectral postprocessing approach a significant amount of computational time can be saved while a high accuracy can be achieved. For example, Table 1 shows that even for Hamiltanian systems a non-symplectic method (neither the method (2.9) nor the method (2.14) are symplectic) together with a coarse-mesh symplectic initial guess can reach an accuracy of $\mathcal{O}(10^{-10})$ with less than 2 CPU seconds; while a straightforward symplectic method requires about 10 seconds.

In using the spectral postprocessing technique, the Gauss-Seidel procedure of the type

$$Y_i^{(k+1)} = y_0 + \sum_{j\le i} \omega_{ij} G(Y_j^{(k+1)}; s_j) + \sum_{j>i} \omega_{ij} G(Y_j^{(k)}; s_j), \tag{4.1}$$

for the ODE problems (i.e., problems in Section 2); and

$$Y_i^{(k+1)} = y_0 + \sum_{j<i} \omega_{ij} G(Y_j^{(k+1)}; s_j) + \sum_{j\ge i} \omega_{ij} G(Y_j^{(k)}; s_j), \tag{4.2}$$

for the integral equations (i.e., problems in Section 3). The method (4.2) is easy to implement; but (4.1) needs some nonlinear solver if G is nonlinear. For the Volterra equations, the very explicit iteration formula (4.1) can be used directly, which is found simple and stable. The resulting formula requires a simple Newton iteration to deal with the term $Y_i^{(k+1)} - \omega_{ii} G(Y_i^{(k+1)}; s_i)$. Since the initial guess of the iteration is already quite accurate, the Newton iteration turns out to be very efficient; only one or two iterations will be enough to guarantee the convergence.

Acknowledgments

The research of the first author was supported by Hong Kong Baptist University, the Research Grants Council of Hong Kong. He is also supported in part by the Chinese Academy of Sciences while visiting its Institute of Computational Mathematics. Part of the research was carried out while the second author was visiting Hong Kong Baptist University.

References

[1] W. Bao and C. Zheng, A time-splitting spectral method for three-wave interactions in media with competing quadratic and cubic nonlinearities, Commun. Comput. Phys., 2 (2007), 123-140.
[2] J. P. Boyd, Chebyshev and Fourier Spectral Methods, Springer-Verlag, 2001.
[3] H. Brunner, Collocation Methods for Volterra Integral and Related Functional Equations Methods, Cambridge University Press, 2004.
[4] C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A. Zang, Spectral Methods, Fundamentals in Single Domains, Springer-Verlag, 2006.
[5] K. Dekker and J. G. Verwer, Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations, North-Holland, 1984.
[6] M. R. Dudek, Lotka-Volterra population model of genetic evolution, Commun. Comput. Phys., 2 (2007), 1174-1183.
[7] D. Gottlieb and S. A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, SIAM, Philadelphia, 1977.
[8] K. Feng, Difference schemes for Hamiltonian formalism an symplectic geometry, J. Comput. Math., 4 (1986), 279-289.
[9] E. Hairer, C. Lubich and G. Wanner, Geometric Numerical Integration, Springer, 2002.
[10] E. Hairer, S. P. Norsett and G. Wanner, Solving Ordinary Differential Equations I. Nonstiff Problems, Springer Series in Comput. Mathematics, Vol. 8, 1993 (2nd ed.), Springer-Verlag.
[11] J. Hong, S. Jiang, C. Li and H. Liu, Explicit multi-symplectic methods for Hamiltonian wave equations, Commun. Comput. Phys., 2 (2007), 662-683.
[12] X. S. Liu, Y. Y. Qi, J. F. He and P. Z. Ding, Recent progress in symplectic algorithms for use in quantum system, Commun. Comput. Phys., 2 (2007), 1-53.
[13] S. Mckee, T. Tang and T. Diogo, An Euler-type method for two-dimensional Volterra integral equations of the first kind, IMA J. Numer. Anal., 20 (2000), 423-440.
[14] J. Shen and T. Tang, Spectral and High-Order Methods with Applications, Science Press, Beijing, 2007.
[15] T. Tang, X. Xu and J. Cheng, On spectral methods for Volterra type integral equations and the convergence analysis, J. Comput. Math., 26 (2008), 825-837.
[16] J. Shen and L.-L. Wang, Fourierization of the Legendre-Galerkin method and a new space-time spectral method, Appl. Numer. Math., 57 (2007), 710-720.
[17] T. Tang, Superconvergence of numerical solutions to weakly singular Volterra integro-differential equations, Numer. Math., 61 (1992), 373-382.