Finite Integration Methods for Isospectral Flows

Li Kan[1] Liu Zhongyun[1] Zhang Yulin[2]
(1. School of Mathematics and Computer Science, Changsha University of Science and Technology, Changsha 410004, China)
(2. Centro de Matemática, Universidade do Minho, Braga 4710 - 057, Portugal)

Abstract In this paper we consider the approximate computation of isospectral flows based on finite integration methods (FIM) with radial basis functions (RBF) interpolation, a new algorithm is developed. Our method ensures the symmetry of the solutions. Numerical experiments demonstrate that the solutions have higher accuracy by our algorithm than by the second order Runge - Kutta(RK2) method.

Key words Isospectral flows Hermitian polynomial interpolation Finite integration method Radial basis function Runge - Kutta method

等谱流的有限积分法

李 侃[1] 刘仲云[1] 张育林[2]
(1.长沙理工大学数学与计算科学学院,长沙,410004)
(2. Centro de Matemática, Universidade do Minho, Braga 4710 - 057, Portugal)

摘 要 本文我们考虑基于径向基函数插值的有限积分法来近似求解等谱流问题,并提出了一种新的算法.我们的方法很好地保证了该问题解的对称性.此外,数值实验也证明了我们的新算法比二阶龙格库塔更精确.

关键词 等谱流 Hermite 多项式插值 有限积分法 径向基函数 Runge - Kutta 法

1 Introduction

Isospectral flows can be characterized in terms of the matrix differential equation

$$X' = [B,X], X(0) = X_0, \tag{1}$$

where $X, B \in R^{n\times n}$ and X_0 is a given $n \times n$ initial matrix and $X_0 = X_0^T$. $B \equiv B(t,X)$ is a matrix function, which is allowed to depend on X and on the time t. The square brackets denote the commutator or *Lie bracket* on matrices, namely $[B,X] = BX - XB$, see for example [1].

Supported by the National Natural Science Foundation of China under Grant No. 11371075
收稿日期:2015年1月20日

Numerical methods for solving (1) have intrigued the researchers for decades, see for example [1,2,3,6] and references therein.

We are particularly interested in the case when B is skew-symmetric which frequently appears in the typical isospectral flows, and will assume, unless explicitly mentioned, that $B^T = -B$ throughout the rest of the present paper.

In this paper we focus on the direct solution to (1) by exploiting the finite integration methods. After briefly reviewing the finite integration method with radial basis function in next section, see for example [7], we develop a new algorithms based on FIM for solving (1) in section 3. Then some numerical examples are presented in section 4, in which the numerical results demonstrate that our algorithm yield a much higher degree of accuracy than RK2 method.

2 Finite integration method

In this section we review the FIM with radial basis functions (RBF), which can be outlined as follows. The field variable $f(t)$ in the interval $[a,b]$ can be interpolated over a number of randomly distributed nodes $\{t_k\}_{k=0}^{N}$ and nodal values $\{f_k\}_{k=0}^{N}$ with $t_0 = a$ and $t_N = b$, as

$$\hat{f}(t) = \sum_{i=0}^{N} R_i(t,t_i)\alpha_i + \sum_{q=0}^{Q} P_q(t)\beta_q, \quad with \quad \hat{f}(t_k) = f_k, k = 0,1,\cdots,N, \tag{2}$$

where $\{R_i(t,t_i)\}_{i=0}^{N}$ is a set of radial basis functions centred at t, $\{P_q(t)\}_{q=0}^{Q}$ is a set of the polynomial basis, $\{\alpha_i\}_{i=0}^{N}$ and $\{\beta_q\}_{q=0}^{Q}$ are the coefficients of $R(t)$ and $mP(t)$ respectively. The polynomial term has to satisfy an extra requirement that guarantees unique approximation of a function as follows

$$\sum_{i=1}^{N} P_q(t_i)\alpha_i = 0, q = 0,1,2,\cdots,Q.$$

So we can get the coefficient

$$a = R_0^{-1}[I - P(P^T R_0^{-1} P)^{-1} P^T R_0^{-1}] f, b = (P^T R_0^{-1} P)^{-1} P^T R_0^{-1} f, \tag{3}$$

where $a = [\alpha_0,\alpha_1,\cdots,\alpha_N]^T$, $b = [\beta_0,\beta_1,\cdots,\beta_N]^T$, $f = [\hat{f}_0,\hat{f}_1,\cdots,\hat{f}_N]^T$, I denotes the identity matrix, $R_0 = (R_i(t_k,t_i))$, $P = (P_q(t_i))$ ($q = 1,2,\cdots,Q$ and $k,i = 1,2,\cdots,N$). Substituting a, b in (3) into (2) yields

$$\hat{f}(t) = R(t) R_0^{-1}[I - P(P^T R_0^{-1} P)^{-1} P^T R_0^{-1}] f + P(t)(P^T R_0^{-1} P)^{-1} P^T R_0^{-1} f = \sum_{i=1}^{N} \phi_i(t)\hat{f}_i, \tag{4}$$

where $\phi_i(t)$, $i = 0,1,\cdots,N$, are called shape functions which have the *Kronecker-delta* property, Consequently we obtain its first order integration easily as

$$F(t) = \int_a^t f(t)\,dt = \sum_{i=1}^{N} \hat{\phi}_i(t) f_i, \tag{5}$$

where

$$\hat{\phi}_i(t) = \int_a^t \phi_i(t)\,dt. \tag{6}$$

Let $F_k = \int_a^k f(t)\,dt$, $F = [F_0, F_1, \cdots, F_N]^T$, and $A = (\hat{\phi}_{ki})$, with $\hat{\phi}_{ik} = \hat{\phi}_i(t_k)$, then the discrete finite integration of the first order can be rewritten as

$$F = Af. \tag{7}$$

3 Finite integration method for isospectral flows

In this section, we consider the computation of isospectral flows (1) with FIM.

Assume that X_m is obtained. We first introduce a uniform division to $[t_m, t_{m+1}]$ for ν subintervals, that is to say that $t_m^{(k)} = t_0 + \delta k$, $\delta = (t_{m+1} - t_m)/\nu$, $k = 0, 1, 2, \cdots, \nu$ are collocation points in the region of $t \in [t_m, t_{m+1}]$, then a direct integration shows that

$$\begin{cases} X_{m+1}^{(1)} - X_m = \int_{t_m}^{t_m+\delta} (B_m X_m - X_m B_m)\,dt, \\ X_{m+1}^{(2)} - X_m = \int_{t_m}^{t_m+2\delta} (B_{m+1}^{(1)} X_{m+1}^{(1)} - X_{m+1}^{(1)} B_{m+1}^{(1)})\,dt, \\ \vdots \\ X_{m+1}^{(\nu)} - X_m = \int_{t_m}^{t_{m+1}} (B_{m+1}^{(\nu-1)} X_{m+1}^{(\nu-1)} - X_{m+1}^{(\nu-1)} B_{m+1}^{(\nu-1)})\,dt. \end{cases} \tag{8}$$

Thus we obtain ν nodal approximative values $\{X_{m+1}^{(k)}\}_{k=1}^{\nu}$ and their corresponding derivatives $\{X'^{(k)}_{m+1}\}_{k=1}^{\nu}$ which are defined by

$$X'^{(k)}_{m+1} = B_{m+1}^{(k)} X_{m+1}^{(k)} - X_{m+1}^{(k)} B_{m+1}^{(k)}. \tag{9}$$

Then let

$$\hat{X}_{m+1}(t) = Hermite\,(X_{m+1}^{(1)}, X_{m+1}^{(2)}, \cdots, X_{m+1}^{(\nu)}) \tag{10}$$

be the piecewise hermitian interpolation polynomial of $X_{m+1}(t)$ of degree $2\nu - 1$ at (t_m, t_{m+1}). Thus we just get an approximate function of integral function $f(X, t) = BX - XB$.

In order to obtain more accurate approximation X_{m+1} of $X(t_{m+1})$ than $\hat{X}_{m+1}$, we next equally partition the interval $[t_m, t_{m+1}]$ into N subintervals. Let $t_m^{(k)} = t_m + \Delta k$, $\Delta = (t_{m+1} - t_m)/N$, $i = 0, 1, 2, \cdots, N$, are collocation points in the region of $t \in [t_m, t_{m+1}]$. After we have $N+1$ nodal values $\hat{X}_{m+1}^{(0)}, \hat{X}_{m+1}^{(1)}, \cdots, \hat{X}_{m+1}^{(N)}$, then by (5), we obtain a sequence of matrices $\{X_{m+1}^{(k)}\}_{k=0}^{N}$.

Finally, we denote $X_{m+1}^{(N)} = X_{m+1}$. Thus we have got a sequence of approximate matrices of $X(t_{m+1})$, for $m = 0, 1, \cdots$, which are the approximate solutions of (1) at $t = 1, \cdots$.

We conclude the above process in the following algorithm.

ALGORITHM [FIM with RBF]

1. Input X_0, $t_0 < t_1 < \cdots < t_m < t_{m+1} < \cdots$, for $m = 0, 1, \cdots$,
2. partition $[t_m, t_{m+1}]$ into ν_m subintervals,
3. for $k = 1, 2, \cdots, \nu_m$, compute $\{X_{m+1}^{(k)}\}_{k=1}^{\nu_m}$ by (8) and $X'^{(k)}_{m+1}$ by (9),
4. construct hermitian interpolation polynomial of $\hat{X}_{m+1}(t)$ of degree $2\nu_m - 1$ at (t_m, t_{m+1}) by (10),
5. randomly partition $[t_m, t_{m+1}]$ into N_m subintervals: $[t_m, t_m^{(1)}], \cdots, [t_m^{(N_m-1)}, t_{m+1}]$,
6. input radial basis functions $\{R_i(t, t_i)\}_{i=0}^{N_m}$ and polynomial basis $\{P_q(t)\}_{q=0}^{Q_m}$,
7. for $k = 0, 1, \cdots, N_m$, compute $f_{m+1}^{(k)} = \hat{X}_{m+1}(t_m^{(k)})$, where $t_m^{(0)} = t_m$ and $t_m^{(N_m)} = t_{m+1}$, and form $f_{m+1} = [f_{m+1}^{(0)}, \cdots, f_{m+1}^{(k)}, \cdots, f_{m+1}^{(N_m)}]^T$,
8. compute $a = [\alpha_0, \alpha_1, \cdots, \alpha_N]^T$, $b = [\beta_0, \beta_1, \cdots, \beta_N]^T$ by (3),
9. form $\hat{f}^{(m+1)}(t)$ in (2) and shape functions $\phi_i^{(m+1)}$ in (4),
10. compute $\hat{\phi}_{ik}^{(m+1)} = \hat{\phi}_i^{(m+1)}(t_m^{(k)})$ defined in (6) and $A_{m+1} = (\hat{\phi}_{ik}^{(m+1)})_{i,k=0}^{N_m}$,
11. compute $F_{m+1} = [F_{m+1}^{(0)}, \cdots, F_{m+1}^{(k)}, \cdots, F_{m+1}^{(N_m)}]^T$ by (7),
12. set $F_{m+1}^{(N_m)} = X_{m+1}$ (This is the approximation of the solution of (1) at $t = t_{m+1}$).

4 Examples and numerical results

In our tests, the matrix B has the form

$$B = X(t)^{-} - X(t)^{+}.$$

Moreover, we consider the case of uniformly distributed interpolation collocation points and $\nu_m = 2$, as well as equally distributed approximative nodes and $N_m = 10$. And we take into account that $R_i(t, t_i) = \sqrt{c^2 + (t - t_i)^2}$, $P_q(t) = t^q$ and the free parameter $c = 1/N_m$, $Q = 7$. Additionally, the typical finite difference method (RK2) is compared to demonstrate the accuracy of the finite integration method. We compare the error in relative similarity for three strategies, presenting

$$e_i = \frac{\|v_{i+1} - v_i\|_\infty}{\|v_i\|_\infty},$$

where v_i is a vector that is composed of eigenvalues sorted accordingly to increasing size of the approximation X_i, for $i = 0, 1, \cdots$.

Example 4.1 We have used the same 10×10 randomly generated symmetric matrix X_0 for stepsize $\Delta = 1/10$. The *Table* 4.1 denotes the numerical results obtained for RK2 method and FIM (RBF) respectively.

Table 1 Stepsize $\Delta = 1/5$, relative similarity errors (Er) for different methods

e_i	*RK2*	*RBF*
e_1	1.0951e-001	4.0797e-002
e_2	1.6081e-001	7.4533e-002
e_3	4.0796e-002	1.9994e-003
e_4	1.6289e-002	2.3372e-003
e_5	5.4959e-003	4.7456e-004
e_6	1.7184e-003	2.6977e-004
e_7	5.1724e-004	1.4034e-004
e_8	2.4323e-004	6.0153e-005
e_9	1.4415e-004	4.4094e-005
e_{10}	8.7037e-005	3.3145e-005

Example 4.2 We exploit the same way to obtain X_0 different from that in example 4.1 for stepsize $\Delta = 1/5$. The numerical results as follows:

Table 1 Stepsize $\Delta = 1/5$, relative similarity errors (Er) for different methods

e_i	*RK2*	*RBF*
e_1	1.8844e-001	4.2309e-002
e_2	6.7262e-002	7.3751e-002
e_3	4.0796e-002	1.6579e-003
e_4	4.6225e-002	2.4193e-003
e_5	3.2873e-002	4.6971e-004
e_6	2.4049e-002	6.8787e-005
e_7	1.8045e-002	3.1867e-005
e_8	1.4104e-002	1.6507e-005
e_9	1.1694e-002	9.6746e-006
e_{10}	1.0492e-002	5.8707e-006

It is apparent that we can get more ideal results with FIM than FDM. And in our tests, there are some problems deserve to be mentioned. When Δ is smaller(for instance $\Delta = 0.01$) , the solution was presented with NAN using RBF and the result with RK2 method is better.

Reference

[1] M. E Calvo, A. Iserles, and A. Zanna, Runge – Kutta methods for orthogonal and isospectral flows, Applied Numerical Mathematics,1996,22: 153 – 163.

[2] M. E. Calvo, A. Iserles and A. Zanna, Numerical solution of isospectral flows, Mathematics of Computation, 1997,Vol. 66: 1461 – 1486.

[3] M. E Calvo, A. Iserles and A. Zanna, Semi – explicit methods for isospectral flows, Advances in Computational Mathematics,2001,14: 1 – 24.

[4] G. H. Golub, C. F. Van Loan, Matrix Computations, third ed. , Johns Hopkins University Press, Baltimore, London, 1996.

[5] M. Li, Y. C. Hon, T. Korakianitis, and P. H. Wen, Finite integration method for nonlocal elastic bar under static and dynamic loads, Engineering Analysis with Boundary Elements,2013,37: 842 – 849.

[6] D. V. Watkins, Isospectral flows SIAM Review, 1984,Vol. 26: 379 – 391.

[7] P. H. Wen, Y. C. Hon M. Li, and T. Korakianitis, Finite integration method for partial differential equations, Applied Mathematical Modelling,2013,37: 10092 – 10106.