An improved element-free Galerkin method for solving the generalized fifth-order Korteweg–de Vries equation*

Feng Zhao(冯 昭), Wang Xiao-Dong(王晓东), and Ouyang Jie(欧阳洁)†

Department of Applied Mathematics, Northwestern Polytechnical University, Xi'an 710129, *China*

(Received 11 October 2012; revised manuscript received 8 January 2013)

In this paper, an improved element-free Galerkin (IEFG) method is proposed to solve the generalized fifth-order Korteweg–de Vries (gfKdV) equation. When the traditional element-free Galerkin (EFG) method is used to solve such an equation, unstable or even wrong numerical solutions may be obtained due to the violation of the consistency conditions of the moving least-squares (MLS) shape functions. To solve this problem, the EFG method is improved by employing the improved moving least-squares (IMLS) approximation based on the shifted polynomial basis functions. The effectiveness of the IEFG method for the gfKdV equation is investigated by using some numerical examples. Meanwhile, the motion of single solitary wave and the interaction of two solitons are simulated using the IEFG method.

Keywords: element-free Galerkin method, shifted polynomial basis, generalized fifth-order Korteweg–de Vries equation, solitary wave

PACS: 47.35.Fg, 04.30.Nk, 02.70.Dh, 47.10.ab **DOI:** 10.1088/1674-1056/22/7/074704

1. Introduction

Most of the physical phenomena in nature can be described by nonlinear evolutionary equations. The generalized fifth-order Korteweg–de Vries (gfKdV) equation is a class of important fifth-order nonlinear partial differential equation with wide applications in fluid mechanics, quantum mechanics, plasma physics, and nonlinear optics.[1,2] The wide applicability of the gfKdV equation is the main reason why it has attracted so much attention from many researchers.

In the past decades, various methods have been developed to find the analytical solution of the gfKdV equation. Those methods mainly include Hirota's method,[3,4] the Adomian decomposition method,[5] the direct algebraic method,[6] the extended tanh method,[7] and the Bäcklund transformation method.[8] However, the analytical solution only exists under some special conditions. In most cases, the gfKdV equation needs to be solved numerically to predict the behavior of the system. Recently, Mohyud-Din *et al*.[9] used radial basis functions combined with the method of line (MOL-RBF) to obtain a numerical solution of the gfKdV equation. The numerical results demonstrated the good accuracy of that method. The MOL-RBF method is a meshless collocation method based on the strong form. Instead of meshing the domain of the problem as in the conventional mesh-based methods, it requires a set of scattered nodes with no connectivity information.

The element-free Galerkin (EFG) method[10–12] has also been developed recently. The EFG method is also a kind of meshless method. However, different from the MOL-RBF method, the EFG method is a meshless method based on the weak form. Compared with the MOL-RBF method, the EFG method can reduce the requirement for continuity on the approximate function via integrals by parts to reduce the order of the differential operators. In addition, the EFG method can also improve the stability and computational accuracy. In the EFG method, the shape functions are constructed with the moving least-squares (MLS) approximation, which uses the non-shifted polynomial basis functions as the basis functions. The MLS approximation has two major features[13] that make it popular: (i) the approximation field function is continuous and smooth in the entire problem domain; (ii) it is capable of producing an approximation with the desired order of consistency. So the EFG method is very suited for solving the partial differential equations with high-order derivatives. Unfortunately, so far the EFG method has only been applied to solve the third-order nonlinear Korteweg–de Vries (KdV) equation,[14] the Korteweg–de Vries–Burgers (KdVB) equation,[15] and the regularized long-wave (RLW) equation.[16] To the authors' knowledge, there is no previous work on the gfKdV equation with the EFG method. The major difficulty is that the violation of the consistency conditions of the shape functions may lead to unstable or even wrong numerical solutions due to the use of a high-order polynomial basis.[17] The same problem exists in the finite cloud method (FCM), which is also a kind of meshless method. The FCM uses the kernel approximation and non-shifted polynomial basis functions to construct the interpolation functions. To overcome the problem of consistency conditions in the FCM, an improved finite cloud method (IFCM), which uses the kernel approximation and shifted polynomial basis functions for constructing the interpolation functions, was presented by Jin

*Project supported by the National Basic Research Program of China (Grant No. 2012CB025903).

†Corresponding author. E-mail: jieouyang@nwpu.edu.cn

© 2013 Chinese Physical Society and IOP Publishing Ltd

et al.[17]. They also testified that the shifted and the non-shifted polynomial basis functions are equivalent mathematically. The difference between the two kinds of basis functions is that for large-node distributions or high-order polynomial basis, the numerical errors with the FCM based on non-shifted basis functions increase quickly compared to the IFCM based on shifted basis functions, resulting in the violation of the consistency condition.

In the EFG method, the MLS shape functions violate the consistency conditions when higher-order polynomial basis functions are adopted to compute the higher-order derivatives in the equation. To overcome this problem, an improved moving least-squares (IMLS) approximation is developed to obtain the approximation functions in this paper. In the IMLS approximation, the shifted polynomial basis functions are used as the basis functions. By coupling the EFG method with the IMLS approximation, an improved element-free Galerkin (IEFG) method is presented for the gfKdV equation. A variation of the method is used to obtain the discrete equation, and the essential boundary conditions are enforced by using the Lagrange multiplier method.[18]

The rest of the paper is organized as follows. In Section 2, we introduce the IEFG method. Section 3 presents the numerical schemes of the gfKdV equation. In Section 4, we use some numerical examples to test the effectiveness of the IEFG method. A brief conclusion is given in Section 5.

2. The IEFG method

The MLS approximation in the EFG method is used to approximate the unknown functions. This method only uses a set of scattered nodes and requires no predefined nodal connectivity. It has many advantages over the mesh-based methods.[19]

2.1. The MLS approximation

In the MLS approximation, the approximation of an unknown field variable $u(\boldsymbol{x})$ at $\boldsymbol{x}$ is defined as

$$u^{\mathrm{h}}(\boldsymbol{x}) = \sum_{j=1}^{m} p_j(\boldsymbol{x}) a_j(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}), \tag{1}$$

where m is the number of monomials in the polynomial basis functions, $\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})$ is the basis vector of the spatial coordinate $\boldsymbol{x}$, and $\boldsymbol{a}(\boldsymbol{x})$ is a vector of coefficients.

For a one-dimensional (1D) problem, in general, the following basis functions are used:

$$\begin{aligned}
&\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) = (1, x), \quad \text{linear basis}, \quad (m=2),\\
&\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) = (1, x, x^2), \quad \text{quadratic basis}, \quad (m=3).
\end{aligned}$$

These kinds of basis functions are called as non-shifted polynomial basis functions in this paper. The coefficient vector $\boldsymbol{a}(\boldsymbol{x})$ can be obtained by minimizing the following weighted discrete L_2 norm:

$$J = \sum_{i=1}^{N} w(\boldsymbol{x}-\boldsymbol{x}_i)\left[\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_i)\boldsymbol{a}(\boldsymbol{x}) - u_i\right]^2, \tag{2}$$

where N is the number of nodes in a compact support domain of point $\boldsymbol{x}$, $w(\boldsymbol{x}-\boldsymbol{x}_i)$ is a weight function, and $u_i = u(\boldsymbol{x}_i)$ is the nodal value of the approximated function at node i. In order to ensure that the fourth-order derivatives of the weight function are continuous within the support domain, the sextic spline function[11] is used in this paper. The sextic spline function is given by

$$\begin{aligned}
& w(\boldsymbol{x}-\boldsymbol{x}_i)\\
=\ & w(r)\\
=\ & \begin{cases} 1-15r^2+40r^3-45r^4+24r^5-5r^6, & r\le 1,\\ 0, & \text{other}, \end{cases}
\end{aligned} \tag{3}$$

where $r = d_i/r_i$, $d_i = \|\boldsymbol{x}-\boldsymbol{x}_i\|$, and r_i is the size of the support domain for the weight function.

To solve the unknown coefficient vector $\boldsymbol{a}(\boldsymbol{x})$, minimizing J in Eq. (2), that is,

$$\partial J/\partial \boldsymbol{a} = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}) - \boldsymbol{B}(\boldsymbol{x})\boldsymbol{U} = 0, \tag{4}$$

leads to the following set of linear relations:

$$\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{A}^{-1}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\boldsymbol{U}, \tag{5}$$

where

$$\begin{aligned}
\boldsymbol{A}(\boldsymbol{x}) &= \sum_{i=1}^{N} w(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{p}(\boldsymbol{x}_i)\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_i),\\
\boldsymbol{B}(\boldsymbol{x}) &= \big(w(\boldsymbol{x}-\boldsymbol{x}_1)\boldsymbol{p}(\boldsymbol{x}_1), w(\boldsymbol{x}-\boldsymbol{x}_2)\boldsymbol{p}(\boldsymbol{x}_2), \ldots,\\
&\quad\ w(\boldsymbol{x}-\boldsymbol{x}_N)\boldsymbol{p}(\boldsymbol{x}_N)\big),\\
\boldsymbol{U} &= (u_1, u_2, \ldots, u_N)^{\mathrm{T}}.
\end{aligned}$$

To describe $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ more clearly, for example, for a 1D problem and using the quadratic basis $(m=3)$, we can explicitly write $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ as

$$\begin{aligned}
\boldsymbol{A}(\boldsymbol{x}) &= \sum_{i=1}^{N} w(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{p}(\boldsymbol{x}_i)\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_i)\\
&= w(\boldsymbol{x}-\boldsymbol{x}_1)\begin{bmatrix} 1 & x_1 & x_1^2\\ x_1 & x_1^2 & x_1^3\\ x_1^2 & x_1^3 & x_1^4 \end{bmatrix}\\
&\quad + w(\boldsymbol{x}-\boldsymbol{x}_2)\begin{bmatrix} 1 & x_2 & x_2^2\\ x_2 & x_2^2 & x_2^3\\ x_2^2 & x_2^3 & x_2^4 \end{bmatrix}\\
&\quad + \cdots + w(\boldsymbol{x}-\boldsymbol{x}_N)\begin{bmatrix} 1 & x_N & x_N^2\\ x_N & x_N^2 & x_N^3\\ x_N^2 & x_N^3 & x_N^4 \end{bmatrix},
\end{aligned} \tag{6}$$

$$\begin{aligned}\boldsymbol{B}(\boldsymbol{x}) &= (w(\boldsymbol{x}-\boldsymbol{x}_1)\boldsymbol{p}(\boldsymbol{x}_1), w(\boldsymbol{x}-\boldsymbol{x}_2)\boldsymbol{p}(\boldsymbol{x}_2), \ldots, w(\boldsymbol{x}-\boldsymbol{x}_N)\boldsymbol{p}(\boldsymbol{x}_N)) \\ &= \left[w(\boldsymbol{x}-\boldsymbol{x}_1)\begin{bmatrix}1\\ x_1\\ x_1^2\end{bmatrix}, \; w(\boldsymbol{x}-\boldsymbol{x}_2)\begin{bmatrix}1\\ x_2\\ x_2^2\end{bmatrix}, \ldots, \; w(\boldsymbol{x}-\boldsymbol{x}_N)\begin{bmatrix}1\\ x_N\\ x_N^2\end{bmatrix}\right]. \end{aligned} \tag{7}$$

By substituting Eq. (5) into Eq. (1), we get

$$u^{\mathrm{h}}(\boldsymbol{x}) = \sum_{i=1}^{N} \phi_i(\boldsymbol{x}) u_i = \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{U}, \tag{8}$$

where $\boldsymbol{\Phi}(\boldsymbol{x})$ presents the vector formed by the shape functions $\phi_i(\boldsymbol{x})$, $(i = 1, 2, \ldots, N)$ and is defined by

$$\begin{aligned}\boldsymbol{\Phi}(\boldsymbol{x}) &= (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_N(\boldsymbol{x})) \\ &= \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{A}^{-1}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x}). \end{aligned} \tag{9}$$

In general, the shape functions in the MLS approximation do not satisfy the Kronecker δ conditions, that is,

$$\phi_i(\boldsymbol{x}_j) \neq \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \tag{10}$$

then the essential boundary conditions cannot be applied directly. So the essential boundary conditions in this paper are enforced by using the Lagrange multiplier method.[18]

2.2. The IMLS approximation

To solve the problem that the shape functions in the MLS approximation violate the consistency conditions, we use the shifted polynomial basis functions to construct the shape functions in the IMLS approximation. The shifted polynomial basis functions are defined in a local coordinate system which moves following the movement of the least-squares approximations.

For a 1D problem, the concrete form of the shifted polynomial basis functions is given as follows:

$$\begin{aligned}&\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) = (1, x-\bar{x}), \quad \text{linear basis}, \quad (m=2), \\ &\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) = (1, x-\bar{x}, (x-\bar{x})^2), \quad \text{quadratic basis}, \quad (m=3), \end{aligned}$$

where $\bar{x}$ presents the coordinate of the computed point.

After adopting the shifted polynomial basis functions in the IMLS approximation, for a 1D problem and using the quadratic shifted polynomial basis $(m = 3)$, we then can explicitly write $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ as

$$\begin{aligned}\boldsymbol{A}(\boldsymbol{x}) &= \sum_{i=1}^{N} w(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{p}(\boldsymbol{x}_i)\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_i) \\ &= w(\boldsymbol{x}-\boldsymbol{x}_1)\begin{bmatrix}1 & x_1-\bar{x} & (x_1-\bar{x})^2\\ x_1-\bar{x} & (x_1-\bar{x})^2 & (x_1-\bar{x})^3\\ (x_1-\bar{x})^2 & (x_1-\bar{x})^3 & (x_1-\bar{x})^4\end{bmatrix} \\ &\quad + w(\boldsymbol{x}-\boldsymbol{x}_2)\begin{bmatrix}1 & x_2-\bar{x} & (x_2-\bar{x})^2\\ x_2-\bar{x} & (x_2-\bar{x})^2 & (x_2-\bar{x})^3\\ (x_2-\bar{x})^2 & (x_2-\bar{x})^3 & (x_2-\bar{x})^4\end{bmatrix} + \cdots \\ &\quad + w(\boldsymbol{x}-\boldsymbol{x}_N)\begin{bmatrix}1 & x_N-\bar{x} & (x_N-\bar{x})^2\\ x_N-\bar{x} & (x_N-\bar{x})^2 & (x_N-\bar{x})^3\\ (x_N-\bar{x})^2 & (x_N-\bar{x})^3 & (x_N-\bar{x})^4\end{bmatrix}, \end{aligned} \tag{11}$$

$$\begin{aligned}\boldsymbol{B}(\boldsymbol{x}) &= (w(\boldsymbol{x}-\boldsymbol{x}_1)\boldsymbol{p}(\boldsymbol{x}_1), w(\boldsymbol{x}-\boldsymbol{x}_2)\boldsymbol{p}(\boldsymbol{x}_2), \ldots, w(\boldsymbol{x}-\boldsymbol{x}_N)\boldsymbol{p}(\boldsymbol{x}_N)) \\ &= \left[w(\boldsymbol{x}-\boldsymbol{x}_1)\begin{bmatrix}1\\ x_1-\bar{x}\\ (x_1-\bar{x})^2\end{bmatrix}, \; w(\boldsymbol{x}-\boldsymbol{x}_2)\begin{bmatrix}1\\ x_2-\bar{x}\\ (x_2-\bar{x})^2\end{bmatrix}, \ldots, \; w(\boldsymbol{x}-\boldsymbol{x}_N)\begin{bmatrix}1\\ x_N-\bar{x}\\ (x_N-\bar{x})^2\end{bmatrix}\right]. \end{aligned} \tag{12}$$

It can be noted that the difference between the IMLS and the MLS approximations is the use of the shifted polynomial basis functions in the IMLS approximation. The other procedures in the IMLS approximation are the same as those in the MLS approximation. By combining the EFG method with the IMLS approximation, the IEFG method is established in this paper.

3. The governing equation and numerical scheme

Consider the following generalized fifth-order Korteweg–de Vries equation:[20]

$$\begin{aligned}&u_t + \alpha u^2 \frac{\partial u}{\partial x} + \beta \frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + \gamma u \frac{\partial^3 u}{\partial x^3} + \sigma \frac{\partial^5 u}{\partial x^5} = 0, \\ &x \in [a, b], \end{aligned} \tag{13a}$$

with the following initial and boundary conditions:

$$\begin{cases} u(x,0) = f(x), \\ u(a,t) = u_1(t), \;\; u(b,t) = u_2(t), \end{cases} \tag{13b}$$

where α, β, γ, and σ are real constants.

The Galerkin weak form of Eq. (13a) is given by

$$\begin{aligned}&\int_{\Omega} \left(u_t + \alpha u^2 \frac{\partial u}{\partial x} + \beta \frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + \gamma u \frac{\partial^3 u}{\partial x^3} + \sigma \frac{\partial^5 u}{\partial x^5} \right) \delta u \, \mathrm{d}\Omega \\ &= 0, \end{aligned} \tag{14}$$

where Ω denotes the domain of x, and δu is the variational function of unknown function u.

Using the integration by parts for Eq. (14), we can get

$$\int_\Omega \delta u\left(u_t+\alpha u^2\frac{\partial u}{\partial x}+\beta\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+\gamma u\frac{\partial^3 u}{\partial x^3}\right) -\sigma\frac{\partial(\delta u)}{\partial x}\frac{\partial^4 u}{\partial x^4}\,\mathrm{d}\Omega=0. \tag{15}$$

With Eq. (8), the approximation functions of u, u_t, u_x, u_{xx}, u_{xxx}, and u_{xxxx} can be obtained as follows:

$$u^{\mathrm{h}}(x,t) = \sum_{i=1}^{N}\phi_i(x)u_i(t)=\boldsymbol{\Phi}(x)\boldsymbol{U}(t), \tag{16}$$

$$\frac{\partial u^{\mathrm{h}}(x,t)}{\partial t} = \frac{\partial}{\partial t}\sum_{i=1}^{N}\phi_i(x)u_i(t)=\sum_{i=1}^{N}\phi_i(x)\frac{\partial u_i(t)}{\partial t} =\boldsymbol{\Phi}(x)\dot{\boldsymbol{U}}(t), \tag{17}$$

$$\frac{\partial u^{\mathrm{h}}(x,t)}{\partial x} = \frac{\partial}{\partial x}\sum_{i=1}^{N}\phi_i(x)u_i(t)=\sum_{i=1}^{N}\phi_{i,x}(x)u_i(t) =\boldsymbol{\Phi}_x(x)\boldsymbol{U}(t), \tag{18}$$

$$\frac{\partial^2 u^{\mathrm{h}}(x,t)}{\partial x^2} = \frac{\partial^2}{\partial x^2}\sum_{i=1}^{N}\phi_i(x)u_i(t)=\sum_{i=1}^{N}\phi_{i,xx}(x)u_i(t) =\boldsymbol{\Phi}_{xx}(x)\boldsymbol{U}(t), \tag{19}$$

$$\frac{\partial^3 u^{\mathrm{h}}(x,t)}{\partial x^3} = \frac{\partial^3}{\partial x^3}\sum_{i=1}^{N}\phi_i(x)u_i(t)=\sum_{i=1}^{N}\phi_{i,xxx}(x)u_i(t) =\boldsymbol{\Phi}_{xxx}(x)\boldsymbol{U}(t), \tag{20}$$

$$\frac{\partial^4 u^{\mathrm{h}}(x,t)}{\partial x^4} = \frac{\partial^4}{\partial x^4}\sum_{i=1}^{N}\phi_i(x)u_i(t)=\sum_{i=1}^{N}\phi_{i,xxxx}(x)u_i(t) =\boldsymbol{\Phi}_{xxxx}(x)\boldsymbol{U}(t), \tag{21}$$

where

$$\begin{aligned}
&\boldsymbol{U}(t)=(u_1(t),u_2(t),\ldots,u_N(t))^{\mathrm{T}},\\
&\dot{\boldsymbol{U}}(t)=\left(\frac{\partial u_1(t)}{\partial t},\frac{\partial u_2(t)}{\partial t},\ldots,\frac{\partial u_N(t)}{\partial t}\right)^{\mathrm{T}},\\
&\boldsymbol{\Phi}(x)=(\phi_1(x),\phi_2(x),\ldots,\phi_N(x)),\\
&\boldsymbol{\Phi}_x(x)=\left(\frac{\partial\phi_1(x)}{\partial x},\frac{\partial\phi_2(x)}{\partial x},\ldots,\frac{\partial\phi_N(x)}{\partial x}\right),\\
&\boldsymbol{\Phi}_{xx}(x)=\left(\frac{\partial^2\phi_1(x)}{\partial x^2},\frac{\partial^2\phi_2(x)}{\partial x^2},\ldots,\frac{\partial^2\phi_N(x)}{\partial x^2}\right),\\
&\boldsymbol{\Phi}_{xxx}(x)=\left(\frac{\partial^3\phi_1(x)}{\partial x^3},\frac{\partial^3\phi_2(x)}{\partial x^3},\ldots,\frac{\partial^3\phi_N(x)}{\partial x^3}\right),\\
&\boldsymbol{\Phi}_{xxxx}(x)=\left(\frac{\partial^4\phi_1(x)}{\partial x^4},\frac{\partial^4\phi_2(x)}{\partial x^4},\ldots,\frac{\partial^4\phi_N(x)}{\partial x^4}\right).
\end{aligned}$$

Substituting Eqs. (16)–(21) into Eq. (15) yields

$$\delta\boldsymbol{U}(t)^{\mathrm{T}}\boldsymbol{K}\dot{\boldsymbol{U}}(t)-\delta\boldsymbol{U}(t)^{\mathrm{T}}\boldsymbol{G}\boldsymbol{U}(t)+\delta\boldsymbol{U}(t)^{\mathrm{T}} \times\int_\Omega\boldsymbol{\Phi}^{\mathrm{T}}(x)\left(\alpha u^2\frac{\partial u}{\partial x}+\beta\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+\gamma u\frac{\partial^3 u}{\partial x^3}\right)\mathrm{d}\Omega=0, \tag{22}$$

where $\boldsymbol{K}$ and $\boldsymbol{G}$ are $N\times N$ matrices, and the elements of the i-th row and j-th column respectively are

$$K_{ij}=\int_\Omega\phi_i\phi_j\,\mathrm{d}\Omega,\quad G_{ij}=\sigma\int_\Omega\phi_{i,x}\phi_{j,xxxx}\,\mathrm{d}\Omega.$$

Because of the arbitrariness of $\delta\boldsymbol{U}(t)$ in Eq. (22), the following equation is obtained:

$$\boldsymbol{K}\dot{\boldsymbol{U}}(t)-\boldsymbol{G}\boldsymbol{U}(t) +\int_\Omega\boldsymbol{\Phi}^{\mathrm{T}}(x)\left(\alpha u^2\frac{\partial u}{\partial x}+\beta\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+\gamma u\frac{\partial^3 u}{\partial x^3}\right)\mathrm{d}\Omega=0. \tag{23}$$

Applying the Crank–Nicolson scheme to Eq. (23), we obtain

$$\begin{aligned}
&\boldsymbol{K}\frac{\boldsymbol{U}^{n+1}-\boldsymbol{U}^n}{\Delta t}-\boldsymbol{G}\frac{\boldsymbol{U}^{n+1}+\boldsymbol{U}^n}{2}\\
&+\int_\Omega\boldsymbol{\Phi}^{\mathrm{T}}(x)\frac{\left(\alpha u^2\frac{\partial u}{\partial x}+\beta\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+\gamma u\frac{\partial^3 u}{\partial x^3}\right)^{n+1}}{2}\mathrm{d}\Omega\\
&+\int_\Omega\boldsymbol{\Phi}^{\mathrm{T}}(x)\frac{\left(\alpha u^2\frac{\partial u}{\partial x}+\beta\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+\gamma u\frac{\partial^3 u}{\partial x^3}\right)^{n}}{2}\mathrm{d}\Omega=0,
\end{aligned} \tag{24}$$

where $\boldsymbol{U}^n=\boldsymbol{U}(n\Delta t)=(u_1(n\Delta t),u_2(n\Delta t),\ldots,u_N(n\Delta t))^{\mathrm{T}}$, and Δt is the time step size.

The nonlinear terms in Eq. (24) are linearized by the following term:[21]

$$(fg)^{n+1}=f^{n+1}g^n+f^ng^{n+1}-f^ng^n. \tag{25}$$

So we can obtain

$$\begin{aligned}
\left(u^2\frac{\partial u}{\partial x}\right)^{n+1} &= (u^2)^{n+1}\left(\frac{\partial u}{\partial x}\right)^n+(u^2)^n\left(\frac{\partial u}{\partial x}\right)^{n+1}\\
&\quad-(u^2)^n\left(\frac{\partial u}{\partial x}\right)^n\\
&=\left(u^{n+1}u^n+u^nu^{n+1}-u^nu^n\right)\left(\frac{\partial u}{\partial x}\right)^n\\
&\quad+(u^2)^n\left(\frac{\partial u}{\partial x}\right)^{n+1}-(u^2)^n\left(\frac{\partial u}{\partial x}\right)^n\\
&=2u^{n+1}u^n\left(\frac{\partial u}{\partial x}\right)^n+(u^2)^n\left(\frac{\partial u}{\partial x}\right)^{n+1}\\
&\quad-2(u^2)^nu_x^n,
\end{aligned} \tag{26}$$

$$\begin{aligned}
\left(\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}\right)^{n+1} &= \left(\frac{\partial u}{\partial x}\right)^{n+1}\left(\frac{\partial^2 u}{\partial x^2}\right)^n+\left(\frac{\partial u}{\partial x}\right)^n\left(\frac{\partial^2 u}{\partial x^2}\right)^{n+1}\\
&\quad-\left(\frac{\partial u}{\partial x}\right)^n\left(\frac{\partial^2 u}{\partial x^2}\right)^n,
\end{aligned} \tag{27}$$

$$\begin{aligned}
\left(u\frac{\partial^3 u}{\partial x^3}\right)^{n+1} &= u^{n+1}\left(\frac{\partial^3 u}{\partial x^3}\right)^n+u^n\left(\frac{\partial^3 u}{\partial x^3}\right)^{n+1}\\
&\quad-u^n\left(\frac{\partial^3 u}{\partial x^3}\right)^n.
\end{aligned} \tag{28}$$

Substituting Eqs. (26)–(28) into Eq. (24) yields

$$\begin{aligned}
&\boldsymbol{K}\frac{\boldsymbol{U}^{n+1}-\boldsymbol{U}^n}{\Delta t}-\boldsymbol{G}\frac{\boldsymbol{U}^{n+1}+\boldsymbol{U}^n}{2}\\
&+(\boldsymbol{H}+\boldsymbol{R}+\boldsymbol{Q})\boldsymbol{U}^{n+1}-\boldsymbol{T}=0,
\end{aligned} \tag{29}$$

where $\boldsymbol{H}$, $\boldsymbol{R}$, and $\boldsymbol{Q}$ are $N \times N$ matrices, and the elements of the i-th row and j-th column respectively are

$$H_{ij} = \frac{\alpha}{2}\int_{\Omega} 2\left(\boldsymbol{\Phi U}^n\right)\left(\boldsymbol{\Phi}_x \boldsymbol{U}^n\right)\phi_i\phi_j + \left(\boldsymbol{\Phi U}^n\right)\left(\boldsymbol{\Phi U}^n\right)\phi_i\phi_{j,x}\,\mathrm{d}\Omega,$$
$$R_{ij} = \frac{\beta}{2}\int_{\Omega} \left(\boldsymbol{\Phi}_{xx}\boldsymbol{U}^n\right)\phi_i\phi_{j,x} + \left(\boldsymbol{\Phi}_x \boldsymbol{U}^n\right)\phi_i\phi_{j,xx}\,\mathrm{d}\Omega,$$
$$Q_{ij} = \frac{\gamma}{2}\int_{\Omega} \left(\boldsymbol{\Phi}_{xxx}\boldsymbol{U}^n\right)\phi_i\phi_j + \left(\boldsymbol{\Phi U}^n\right)\phi_i\phi_{j,xxx}\,\mathrm{d}\Omega,$$

$\boldsymbol{T}$ is an $N \times 1$ matrix, and the element of the i-th row is

$$T_i = \frac{\alpha}{2}\int_{\Omega} \left(\boldsymbol{\Phi U}^n\right)\left(\boldsymbol{\Phi U}^n\right)\left(\boldsymbol{\Phi}_x \boldsymbol{U}^n\right)\phi_i\,\mathrm{d}\Omega.$$

Equation (29) is the discrete equation of the gfKdV equation. By solving the iteration equation, the numerical solution of the gfKdV equation can be obtained.

4. Numerical examples

In this section, the effectiveness of the IEFG method for the gfKdV equation is investigated by using some numerical examples. They are the Sawada–Kotra (SK) and the Kaup–Kupershmidt (KK) equations[22] if we respectively use the parameters $\alpha = 45$, $\beta = -15$, $\gamma = -15$, $\sigma = 1$ and $\alpha = 45$, $\beta = -75/2$, $\gamma = -15$, $\sigma = 1$ in Eq. (13a).

To illustrate the accuracy of the IEFG method, the error norms L_2 and L_∞ are computed, which are defined as follows:

$$L_2 = \left\|u^{\mathrm{e}} - u^{\mathrm{h}}\right\|_2 = \left(\int_{\Omega}\left(u^{\mathrm{e}} - u^{\mathrm{h}}\right)^2\right)^{1/2},$$
$$L_\infty = \left\|u^{\mathrm{e}} - u^{\mathrm{h}}\right\|_\infty = \max_i \left|u_i^{\mathrm{e}} - u_i^{\mathrm{h}}\right|,$$

where u^{e} and u^{h} denote the exact and the approximate solutions, respectively.

We examine our results by calculating the following conservative laws[23] for the SK equation:

$$I_1 = \int_{\Omega} u\,\mathrm{d}x, \quad I_2 = \int_{\Omega}\left(\frac{1}{3}u^3 - u_x^2\right)\mathrm{d}x,$$
$$I_3 = \int_{\Omega} \frac{1}{4}u^4 - \frac{9}{4}uu_x^2 + \frac{3}{4}u_{xx}^2\,\mathrm{d}x,$$

and for the KK equation:

$$I_1 = \int_{\Omega} u\,\mathrm{d}x, \quad I_2 = \int_{\Omega}\left(\frac{1}{3}u^3 - \frac{1}{8}u_x^2\right)\mathrm{d}x,$$
$$I_3 = \int_{\Omega}\left(\frac{1}{4}u^4 - \frac{9}{16}uu_x^2 + \frac{3}{64}u_{xx}^2\right)\mathrm{d}x.$$

The discrete linear equations (29) are assembled by integrating over the domain of the problem using the Gauss quadrature. To evaluate these integrals, a regular arrangement of nodes is used, and the cell structure is set so that the quadrature cells are equal to the intervals between the nodes. In each integration cell, the 5-Gauss quadrature is used. Once the cells and the corresponding quadrature points are defined, the discrete equations are assembled by looping over each quadrature point.

Example 1 (single soliton) If we put parameters $\alpha = 45$, $\beta = -15$, $\gamma = -15$, and $\sigma = 1$ in Eq. (13a), the SK equation can be obtained

$$u_t + 45u^2\frac{\partial u}{\partial x} - 15\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} - 15u\frac{\partial^3 u}{\partial x^3} + \frac{\partial^5 u}{\partial x^5} = 0. \quad (30)$$

The analytical solution is given by[6]

$$u(x,t) = a_0 - 2c\,\mathrm{sech}^2\left(\sqrt{c}\xi\right), \quad (31)$$

where $\xi = x - \left(45a_0^2 - 60a_0c + 16c^2\right)t$, and a_0 and c are an arbitrary constant and a positive parameter, respectively.

We use the analytical solution evaluated at $t = 0$ as the initial condition. The boundary conditions $u_1(t)$ and $u_2(t)$ are also extracted from the analytical solution. The IEFG method is applied to solve the above SK equation with the parameters $a_0 = 0.1$, $c = 0.04$, the space step size $\Delta x = 0.3$, and the time step size $\Delta t = 0.1$. The basis function is chosen to be quintic.

In Fig. 1, the numerical and the exact solutions at different time are plotted. In Fig. 2, we present the space–time graph of the numerical solution at different time. It is clear from Fig. 1 that the numerical results are in good agreement with the exact solutions in this case. In Table 1, the error norms L_2 and L_∞ are given. It can be seen from Table 1 that the error norms L_2 and L_∞ are less than 3.224×10^{-4} and 2.733×10^{-4} respectively at $t = 20$. The invariants I_1, I_2, and I_3 are recorded in Table 2. From Table 2, it is clear that all the three invariants are almost constants in the simulation and change by $2.5 \times 10^{-3}\%$, $1.8 \times 10^{-3}\%$, and $2.8 \times 10^{-2}\%$ of their original values respectively at $t = 20$.

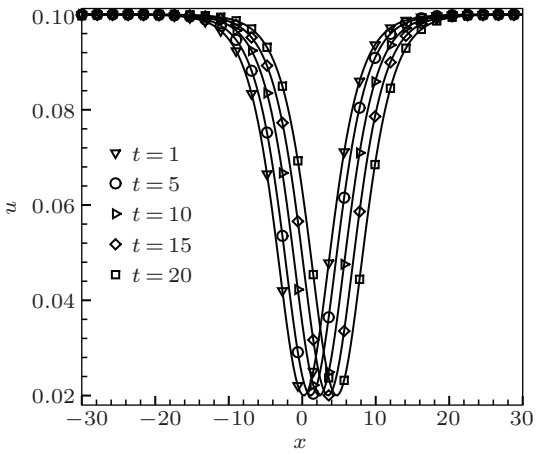


Fig. 1. Numerical (symbol) and exact (line) solutions of the SK equation at different time.

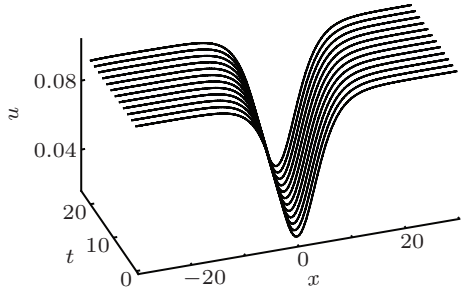


Fig. 2. Space–time graph of numerical solution of the SK equation at different time.

Table 1. Error norms of the SK equation at different time.

t	1	5	10	15	20
$L_\infty/10^{-5}$	0.9027	2.278	3.132	8.779	27.33
$L_2/10^{-5}$	1.803	5.286	7.653	13.56	32.24

Table 2. Invariants for the SK equation at different time.

t	I_1	I_2	I_3
0	5.230009253	0.01409120613	0.001071515763
3	5.230012745	0.01409142147	0.001071416820
6	5.230041211	0.01409193972	0.001071389822
9	5.230049066	0.01409221453	0.001071379136
12	5.230111698	0.01409298810	0.001071448390
15	5.230102945	0.01409308524	0.001071397640
18	5.230006113	0.01409222919	0.001071292123
20	5.229878872	0.01409095065	0.001071216182

Example 2 (single soliton) Using parameters $\alpha = 45$, $\beta = -75/2$, $\gamma = -15$, and $\sigma = 1$ in Eq. (13a), we get the KK equation

$$u_t + 45u^2\frac{\partial u}{\partial x} - \frac{75}{2}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} - 15u\frac{\partial^3 u}{\partial x^3} + \frac{\partial^5 u}{\partial x^5} = 0. \quad (32)$$

Similar to example 1, we still extract the initial condition $f(x)$ and the boundary conditions $u_1(t)$ and $u_2(t)$ from the analytical solution. The analytical solution of the KK equation is given as[22]

$$u(x,t) = -k^2\frac{2\cosh\theta + 1}{(\cosh\theta + 2)^2}, \quad (33)$$

where $\theta = kx - k^5 t + \eta$, and k, η are arbitrary (real) parameters.

For the numerical computation, we choose the parameters $k = 1.0$, $\eta = 0.0$, the space step size $\Delta x = 0.08$, and the time step size $\Delta t = 0.1$ .The basis function is chosen to be quintic.

In Fig. 3, we display the exact and the numerical solutions, whereas Fig. 4 presents the space–time graph of the numerical solution of the KK equation at different time. As can be seen in Fig. 3, the numerical and the exact solutions show an excellent agreement. It is also clearly shown in Fig. 3 that the solution moves towards the right with a constant speed with increasing time. The error norms L_2 and L_∞ are computed and given in Table 3. As can be seen in Table 3, the error norms L_2 and L_∞ are less than 2.493×10^{-3} and 9.667×10^{-4} respectively at $t = 10$. The invariants I_1, I_2, and I_3 are recorded in Table 4. From Table 4, we can see that all the three invariants are kept nearly constant during the simulation and change by $2.2 \times 10^{-2}\%$, $1.9 \times 10^{-2}\%$, and $6.1 \times 10^{-2}\%$ of their original values respectively at $t = 10$.

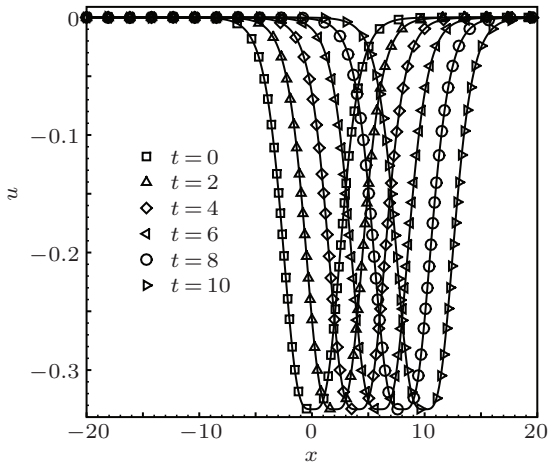


Fig. 3. Numerical (symbol) and exact (line) solutions of the KK equation at different time.

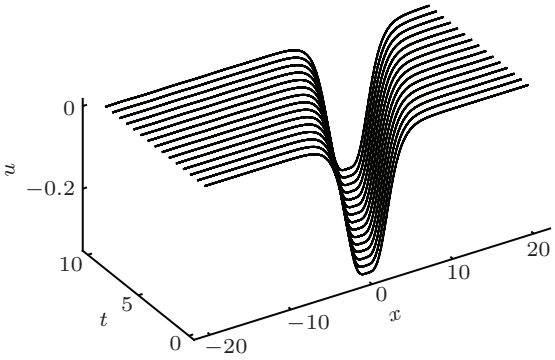


Fig. 4. Space–time graph of numerical solution of the KK equation at different time.

Table 3. Error norms of the KK equation at different time.

t	2	4	6	8	10
$L_\infty/10^{-4}$	3.651	4.082	7.927	72.41	9.667
$L_2/10^{-3}$	0.8197	1.061	1.571	1.857	2.493

Table 4. Invariants for the KK equation at different time.

t	I_1	I_2	I_3
0	−1.999999984	−0.05545944345	0.01764197496
2	−2.000123413	−0.05547488542	0.01765378675
4	−1.999733774	−0.05545218388	0.01762825276
6	−1.999643763	−0.05547974489	0.01765997337
8	−2.000750761	−0.05547059818	0.01765285049
10	−1.999564082	−0.05547022248	0.01765266082

Example 3 (two-soliton interaction) Consider the KK equation to study the interaction of two solitary waves with the following initial and boundary conditions:

$$\begin{cases} u(x,0) = -\sum_{i=1}^{2} A_i^2 \dfrac{2\cosh(A_i - x_i) + 1}{(\cosh(A_i - x_i) + 2)^2}, \\ u(a,t) = 0, \quad u(b,t) = 0. \end{cases} \tag{34}$$

We solve this example by taking the computational area $[a,b] = [-30,30]$ with the space step size $\Delta x = 0.075$ and the time step size $\Delta t = 0.1$. The calculation is carried out by choosing the parameters in Eq. (34) as $A_1 = 1.2$, $x_1 = -20$, $A_2 = 0.8$, $x_2 = 0$. The basis function is chosen to be quintic.

The numerical solution is plotted in Figs. 5 and 6. From Fig. 5, we can see that the two solitary waves propagate towards right as the time increases. At the beginning, the solitary wave with a higher amplitude stands on the left of that with a smaller amplitude. The solitary wave with a higher amplitude moves faster than that with a smaller amplitude, so two solitary waves gradually get closer, deform, and finally merge into a single wave at $t = 7$. This wave packet splits into two solitary waves at $t = 10$. At $t = 15$, the shapes of the two solitary waves are fully recovered, and the solitary wave with a higher amplitude stands on the right. It is also clear from Figs. 5 and 6 that the two solitary waves separate from each other and maintain their original shapes after the collision. The peaks and positions of two solitary waves are listed in Table 5. The results show that the peak of the big amplitude wave decreases by $2.7 \times 10^{-1}\%$ after the collision, while the peak of the small amplitude wave increases by $9.4 \times 10^{-2}\%$.

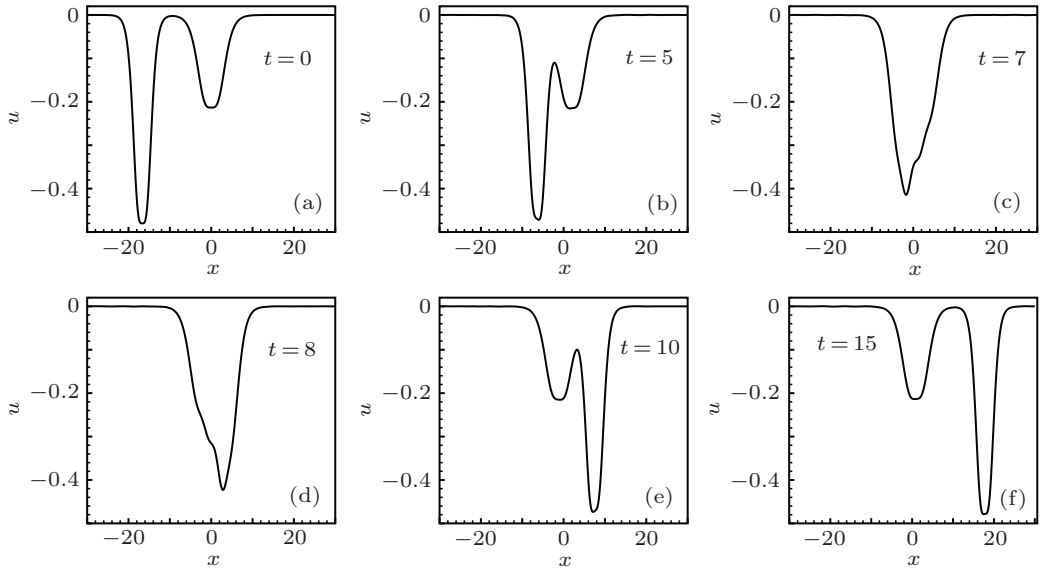


Fig. 5. Numerical solution of the KK equation at different times: (a) $t = 0$, (b) $t = 5$, (c) $t = 7$, (d) $t = 8$, (e) $t = 10$, (f) $t = 15$.

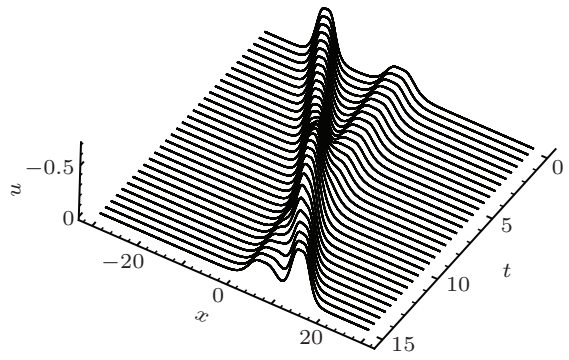


Fig. 6. Interaction of two solitary waves with different amplitudes.

Table 5. Peaks and positions of the two solitary waves at different time.

t	Big amplitude wave		Small amplitude wave	
	Peak	Position	Peak	Position
0	−0.4800	−16.73	−0.2133	−0.075
5	−0.4712	−5.775	−0.2157	1.5000
10	−0.4742	7.3500	−0.2148	−0.675
15	−0.4787	17.475	−0.2135	0.4500

5. Conclusion

In this paper, the IEFG method is proposed to solve the gfKdV equation. The motion of single solitary wave and the interaction of two solitons are simulated using the IEFG method. The numerical results show that the IEFG method is accurate and efficient. This means that the IEFG method is superior to the traditional EFG method in solving the partial differential equations with high-order derivatives. So the IEFG method can be extended to solve some other nonlinear partial differential equations with high-order derivatives in physics and other fields.

References

[1] Kawahara T 1972 *J. Phys. Soc. Jpn.* **33** 260
[2] Benjamin T B, Bona J L and Mahony J J 1972 *Phil. Trans. R. Soc.* **272** 47
[3] Hereman W and Nuseir A 1997 *Math. Comput. Simul.* **43** 13
[4] Tam H W and Hu X B 2000 *Phys. Lett. A* **272** 174

[5] Lnc M 2006 *Appl. Math. Comput.* **172** 72
[6] Fan E 2003 *Chaos Soliton. Fract.* **16** 819
[7] Wazwaz A M 2007 *Appl. Math. Comput.* **184** 1002
[8] Xin Y, Gao Y T, Sun Z Y and Liu Y 2010 *Phys. Scr.* **81** 045402
[9] Mohyud-Din S T, Negahdary E and Usman M 2012 *Int. J. Numer. Method H* **22** 641
[10] Belytschko T, Lu Y Y and Gu L 1994 *Int. J. Numer. Methods Eng.* **37** 229
[11] Liu G R and Gu Y T 2005 *An Introduction to Meshfree Methods and Their Programming* (Netherlands: Springer) pp. 102–106
[12] Zhang L, Ouyang J and Zhang X H 2008 *Phys. Lett. A* **372** 5625
[13] Zhang X H, Ouyang J and Zhang L 2009 *Eng. Anal. Bound. Elem.* **33** 356
[14] Wang J F, Sun F X and Cheng R J 2010 *Chin. Phys. B* **19** 060201
[15] Cheng R J and Cheng Y M 2011 *Chin. Phys. B* **20** 070206
[16] Cheng R J and Ge H X 2012 *Chin. Phys. B* **21** 040203
[17] Jin X Z, Li G and Aluru N R 2001 *Comput. Model. Eng. Sci.* **2** 447
[18] Fernánez-Méndez S and Huerta A 2004 *Comput. Methods Appl. Mech. Eng.* **193** 1257
[19] Belytschko T, Krongauz Y and Organ D 1996 *Comput. Methods Appl. Mech. Eng.* **139** 3
[20] Yang L, Zhang F J and Wang Y H 2002 *Chaos Soliton. Fract.* **13** 337
[21] Uddin M, Haq S and Islam S U 2009 *Appl. Math. Comput.* **212** 458
[22] Parker A 2000 *Physica D* **137** 25
[23] Goktas U and Hereman W 1997 *J. Symb. Comput.* **24** 591