SYMMETRIC COLLOCATION METHODS FOR LINEAR PARTIAL DIFFERENTIAL EQUATIONS

PILWON KIM

Abstract. This paper introduces the symmetric collocation methods that take a new approach to apply Lie symmetry theory to numerical analysis of differential equations. The basic idea of the method is to approximate unknown solutions by collocation of other solutions generated by symmetry transformations. Different from the existing numerical methods, the symmetric collocation method does not use any discretization even for time-dependant differential equations and therefor does not suffer from any geometric restriction on the domain or the distribution of the data locations. Numerical realization of some solutions of a linear heat equation is developed.

1. Introduction

Symmetries have long been recognized to be intrinsic and fundamental features of differential equations. Once one has determined the (point) symmetry group of differential equations, it can be used for a number of applications. Some of main applications are,

(a) A group action transforms a given solution to new ones; so the complete set of solutions is transformed to itself.

(b) Using differential invariants obtainable in a systemical way, a given differential equation reduces to its simpler invariant representation; the group-invariant solutions can be found by solving the reduced system.

(c) The invariance of partial differential equations (PDEs) is a necessary condition to obtain conservation laws in variational problem by the application of Noether's theorem.

In the last decade there has been growing recognition on applications of Lie theory to numerical analysis of differential equations. Most of efforts have been devoted to develop the discrete version of symmetry theory, i.e. symmetric finite difference methods. One of main stream is to determine symmetries for given difference equations [13, 16]. In order to obtain symmetries that coincide with Lie symmetries in the continuous limit, one needs to significantly modify the Lie techniques used in the continuous case. Another idea is to discretize differential equations in a way that preserves some of the symmetries [5, 6, 7]. Given a differential equation one constructs a difference equation and a mesh in such a manner as to be compatible with the original symmetry group. Recently alternative approach was made in [18, 19]. Using the Cartan's moving frame theory that is reformulated in modern way in [10, 11], it improves the existing finite difference scheme greatly for some differential equations.

In this paper we introduce the symmetric collocation method, the novel application of Lie symmetry theory to numerical analysis that is not based on finite difference schemes. The underlying idea of the method is actually simple; we generate sufficiently many solutions by using the property (a) of the symmetry group of the equation, and then collocate them to approximate the target solutions.

For a linear differential equation $Lu = f$, let φ be a particular solution and $\{\phi_i\}_{i\in I}$ a set of solutions of the homogenous equation $Lu = 0$. Now the question is: can we find a finite subset $\{\phi_j\}_{j\in J}$, $J \subset I$ so that

$$U = \varphi + \sum_{j\in J} \phi_j$$

meets the initial and the boundary conditions closely enough?

The answer surely depends on then the number and the variety of the solutions ϕ_i that we have. For the standard linear PDEs like the linear heat equations $u_t = u_{xx}$, there exist infinite number of the polynomial solutions available. Moreover, those PDEs usually have abundant symmetry groups that enable us to generate new solutions from trivial ones. With help of symmetry analysis on a given differential equation, we can derive *all* of this kind of transformations systemically. For example, the most general solution for the linear heat equation that can be obtained is

$$u = \frac{\varepsilon_3}{\sqrt{1+4\varepsilon_6 t}} \exp\Big(-\frac{\varepsilon_5 x + \varepsilon_6 x^2 - {\varepsilon_5}^2 t}{1+4\varepsilon_6 t}\Big) \times f\Big(\frac{e^{-\varepsilon_4}(x-2\varepsilon_5 t)}{1+4\varepsilon_6 t} - \varepsilon_1, \frac{e^{-2\varepsilon_4} t}{1+4\varepsilon_6 t} - \varepsilon_2\Big) + g(x,t)$$

for any $\varepsilon_1, \ldots, \varepsilon_6$, where f, g are any exact solutions. We refer the readers to [17] for more detail.

The symmetric collocation method is basically data-fitting over the domain by the symmetry transformation. Starting from the usual trivial solution, we search the general solution spaces until we obtain one close enough to given data. The symmetry transformations often provide surprisingly various solutions even out of trivial solutions and relatively small number of collocations is needed to yield a good result.

As the collocated functions are all globally defined, the methods basically works the same for *any* type of boundary or initial condition no matter what geometric restriction is imposed. In addition, in case the given equation contains some unknown parameter, the method can manage them just as extra symmetry transformation. Therefore the ill-posed problems caused by bad geometry, unknown parameters, backward process in time, or any combination of these can be managed as well as the usual conditions in the symmetric collocation method.

One of the benefits of the method is that we can always check exactly what the error is and how trustworthy the numerical solution is. Since the numerical solution is always one of the general solutions, the only error comes from discrepancy between the given data and the obtained value, which we can easily confirm.

2. Differential Error and Data Error

We are concerned with the numerical linear PDE problem of the form

$$Lu = f(x), \quad x \in \Omega \qquad \text{“differential condition”}$$
$$u = g(x), \quad x \in D \qquad \text{“data condition”}$$

where Ω is a subset of $\mathbf{R}^n$ and D is a finite subset of $\overline{\Omega}$. [1] The data condition may include a partial derivatives. In the examples through the rest of the paper we fix the differential conditions to the 1-D heat equation $u_{x_2} - u_{x_1x_1} = 0$ or more conventionally, $u_t - u_{xx} = 0$.

In order to verify validity of a numerical scheme, we usually apply it to the simple problem that has the known solution u, and compare the numerical solution $\tilde{u}$ to u pointwise. However when we try to solve the practical problem without knowing the solution, it is generally hard to find out how close the numerical solution is to the real solution. The situation becomes worse considering that the practical problems given with a finite data condition does not have unique solution, i.e., it is intrinsically ill-posed.

To verify a specific numerical solution, we introduce two error estimators,

$$E_{\text{diff}}(u) = \frac{1}{\sqrt{A}} \| Lu - f \|_{2,\Omega} \qquad \text{“differential error”}$$

$$E_{\text{data}}(u) = \frac{1}{\sqrt{N}} | u - g |_{2,D} \qquad \text{“data error”}$$

where A, N are the area of Ω and the size of D respectively. Here $\| \cdot \|_{2,\Omega}$, $| \cdot |_{2,D}$ denote L_2-norm and l_2-norm defined on each domain respectively. For most of the existing numerical schemes including finite difference methods and finite element methods, the data errors of their numerical solutions are zero or close to zero. Therefore their actual errors are the differential errors. Unfortunately it is generally challenging to compute the differential errors. In the case of finite difference method, we even have to make proper local interpolations to compute integration.

On the contrary, since the collocations of solutions make another solutions, the differential error of (1.1) is zero. Therefore the solutions by the symmetric collocation method have only the data error, of which computation is very easy and straightforward.

3. Simple Collocation of Solutions

For a linear homogeneous differential condition, suppose we are given a data condition of size n, i.e., the values $d_1, \ldots, d_n$ of the unknown function u measured at n different locations $p_1, \ldots, p_n$. With the same number of known solutions $\phi_1, \ldots, \phi_n$ of the differential condition, one natural idea of approximation of u is to set

$$u = \sum_{i=1}^{n} c_i \phi_i.$$

Then it gives the n by n matrix equation on the points $p_1, \ldots, p_n$,

$$\begin{bmatrix} \phi_1(p_1) & \phi_2(p_1) & \ldots & \phi_n(p_1) \\ \phi_1(p_2) & \phi_2(p_2) & \ldots & \phi_n(p_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(p_n) & \phi_2(p_n) & \ldots & \phi_n(p_n) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

[1]We can convert a given nonhomogeneous problem into homogeneous one by constructing any particular solution numerically. This particular solution does not need to satisfy the data condition, so it can be easily constructed with small cost up to any accuracy level we want.

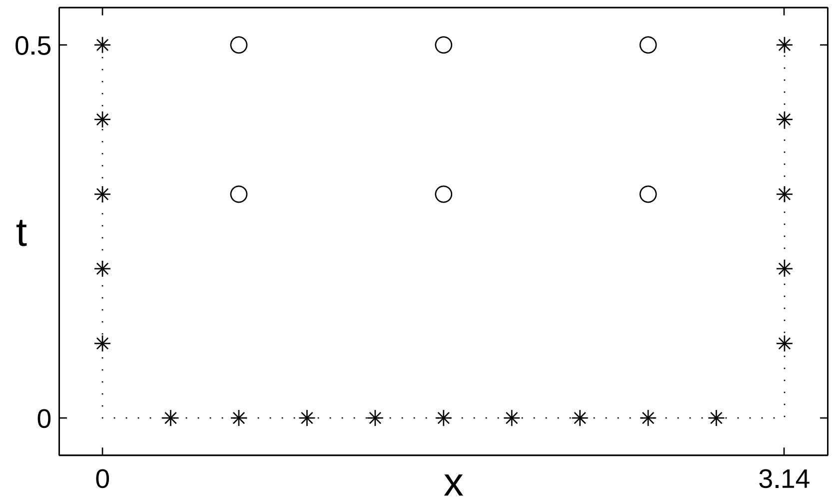

FIGURE 1. : The distribution of the 20 data locations (O) for the collocation of solution and Crank-Nicolson methods. The points (□) at which the error is evaluated.

which determines the coefficient $c_1, \ldots, c_n$. In fact this is the simple analogue of manipulation of general solutions in linear ordinary differential equations. If this equation can be exactly solved, those coefficients make one of the *exact* numerical solutions with both zero for the differential and the data error.

Unfortunately this is not the usual case, since the matrices often become ill-conditioned as n increases. This means that it is hard to keep the solutions different enough from each other with respect to the data locations for the large data set. This situation can be avoided to some extent if we apply the symmetry transformations to simple solutions.

Example 3.1. Simple Collocation of Solutions The differential condition is again the 1-D heat equation in the domain $0 \le x \le \pi, 0 \le t \le 0.5$. The data condition consists of the values at the 16 points which are generated by $u = \exp(-t)\sin(x)$ on the boundary since we want to compare the result of the method with that of finite difference methods. See Figure 1. There are infinite number of polynomial solutions for the heat equation,

$$\phi_0 = 1 \qquad \phi_1 = x$$

$$\phi_2 = t + \frac{1}{2}x^2 \qquad \phi_3 = tx + \frac{1}{6}x^3$$

$$\phi_4 = t^2 + tx^2 + \frac{1}{12}x^4 \qquad \phi_5 = t^2x + \frac{1}{3}tx^3 + \frac{1}{60}x^5$$

$$\phi_6 = t^3 + \frac{3}{2}t^2x^2 + \frac{1}{24}tx^4 + \frac{1}{120}x^6 \qquad \phi_7 = \ldots\ldots$$

These polynomials are similar with respect to the data locations, especially where $t = 0$. Since this may cause the corresponding matrix to be ill-conditioned, we apply the symmetry transformation to them as in (1.2) to obtain a new set of solutions. Numerical experiment shows that even a small transformations can greatly improve the situation.

Some of the transformed solutions we use in this example are,

$$
\begin{aligned}
\psi_0 &= \phi_0(x,t) \\
&= 1 \\
\psi_1 &= \exp(-x+t)\ \phi_0(x-2t,t) \\
&= \exp(-x+t) \\
\psi_2 &= \frac{1}{\sqrt{1+4t}}\ \exp\Big(-\frac{x^2}{1+4t}\Big)\ \phi_0(\frac{x}{\sqrt{1+4t}},\frac{t}{\sqrt{1+4t}}) \\
&= \frac{1}{\sqrt{1+4t}}\ \exp\Big(-\frac{x^2}{1+4t}\Big) \\
\psi_3 &= \phi_1(x,t) \\
&= x \\
\psi_4 &= \exp(-x+t)\ \phi_1(x-2t,t) \\
&= (x-2t)\exp(-x+t) \\
\psi_5 &= \frac{1}{\sqrt{1+4t}}\ \exp\Big(-\frac{x^2}{1+4t}\Big)\ \phi_1(\frac{x}{\sqrt{1+4t}},\frac{t}{\sqrt{1+4t}}) \\
&= \frac{x}{1+4t}\ \exp\Big(-\frac{x^2}{1+4t}\Big) \\
\psi_6 &= \dots\dots
\end{aligned}
$$

Note that the first six functions are generated out of $\phi_0 = 1, \phi_1 = x$ only.

Table 1 : Simple Collocation vs. Crank-Nicolson

Method	Error at (x,t)					
	(0.2π,0.3)	(0.5π,0.3)	(0.8π,0.3)	(0.2π,0.5)	(0.5π,0.5)	(0.8π,0.5)
Sim.Colloc.	2.3e-7	8.1e-8	2.5e-9	3.7e-7	8.8e-8	1.8e-8
Cr-Ni	1.0e-3	1.6e-3	1.0e-3	1.3e-3	2.2e-3	1.3e-3

Table 1 compares the numerical result of the collocation using these solutions with that of Crank-Nicolson method that is a well-known finite difference method. It is definite that the collocation of solutions beats the Crank-Nicolson method all over the domain. It is also noteworthy that the method does not need any discretization even in time since the basis functions are globally defined both in time and space, i.e., the method is a true collocation method.

The differential error of the collocation method is zero as mentioned before and the data error is 1.54×10^{-13} which is much lower than the actual errors in Table 1. This discrepancy can be explained by ill-posedness of the given problem, i.e., the data at 19 points is too small to decide one unique solution. As we see in the examples in the section 4 later, for the problems with large data conditions, this difference disappears and the data error actually indicates the real error of solutions. This means that one can easily check the validity of the numerical solutions produced by the symmetric collocation methods even without the real solutions. On the contrary, the error of the Crank-Nicolson method is hard to figure out if the real solution is not given.

However it becomes hard to find a appropriate set of solutions that yields a well-conditioned matrix as the data set grows, even if we adopt the symmetry transformations. For example, the condition number of the matrix in Example 3.1 is 4.42×10^{11} and it is the cause of the nonzero data error in practical computation. This difficulty makes us turn our attention to finding a better scheme to adjust transformation parameters well without increasing the number of solutions collocated.

4. Symmetric Collocation by Data-Fitting

To deal with transformations in clear way, we rewrite them in the view of group actions. Then the symmetry transformation (1.2) is now,

$$\begin{aligned}\varepsilon \cdot f(x,t) &= (\varepsilon_1, \dots, \varepsilon_6) \cdot f(x,t) \\ &= \frac{1}{\sqrt{1+4\varepsilon_6 t}} \ \exp\left(\varepsilon_3 - \frac{\varepsilon_5 x + \varepsilon_6 x^2 - {\varepsilon_5}^2 t}{1+4\varepsilon_6 t}\right) \\ &\quad \times f\Big(\frac{e^{-\varepsilon_4}(x - 2\varepsilon_5 t)}{1+4\varepsilon_6 t} - \varepsilon_1, \frac{e^{-2\varepsilon_4} t}{1+4\varepsilon_6 t} - \varepsilon_2\Big)\end{aligned}$$

where $\varepsilon = (\varepsilon_1, \dots, \varepsilon_6)$ is an element of a local Lie group G.

Now we claim that if a given differential equation have a sufficient number of symmetry transformations, any solution can be approximated well by some transformations of relatively small number of known solutions. In other words, for known solution $\phi_1, \dots, \phi_n$ and a given solution u, there exist elements $\varepsilon^1, \dots, \varepsilon^n \in G$ such that $\tilde{u} = \varepsilon^1 \cdot \phi_1 + \dots + \varepsilon^n \cdot \phi_n$ is close enough to u. Note that we have total mn number of transformation parameters. Using the optimization algorithm the symmetric collocation method adjust the parameters until they reach the ones that minimize the data error $E_{data}(\tilde{u})$. This implies that we turn a given numerical PDE to a large-scale nonlinear optimization problem.

In the following two examples we collocate only seven polynomial solutions $\phi_0, \dots, \phi_6$ as in Example 3.1 for the data collected at over 100 points. The data is generated by the function,

$$u = \frac{1}{4}\exp(x+t) - \exp(-\pi^2 t)\sin(\pi x) + 3\exp(-4t)\cos(2x)$$

which is purposely made out of special functions as a target solution. For the both examples we use the general optimization tool `fminsearch` in Matlab and the costs of computation are not considered as the accuracy of the result is the concern at this point.

Example 4.1. Standard Data Condition

The values of a unknown function are collected at some points on the boundaries. Starting with only seven polynomial solutions, we find the proper approximation which agrees with the target function as closely as at 111 data locations. We can keep minimizing the data error as far as it reduces to the level we want.

Table 2 : Point-wise Error according to the Data Error in Example 4.1

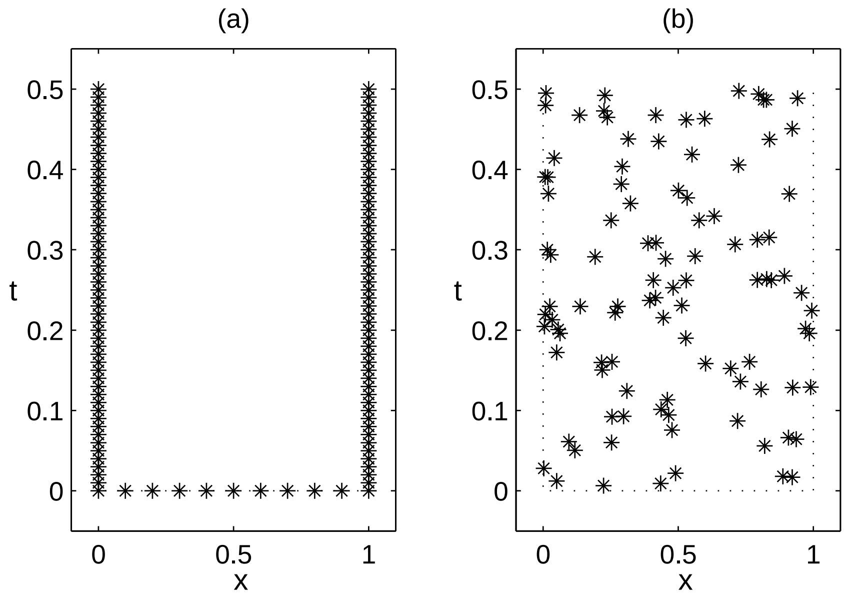

FIGURE 2. : (a) The distribution of the 111 data measured on the rectangular boundary. (b) The distribution of the 100 data measured at totally scattered locations in time and space.

Data Error	Error at (x,t)					
$E_{\text{data}}(u)$	(0.25,0.3)	(0.5,0.3)	(0.75,0.3)	(0.25,0.5)	(0.5,0.5)	(0.75,0.5)
4.7e-3	1.0e-3	2.5e-3	4.4e-3	2.6e-3	4.5e-3	7.1e-3
4.7e-4	6.9e-5	2.0e-5	2.5e-5	2.5e-4	8.4e-5	4.7e-4
5.0e-5	1.7e-6	3.0e-6	2.7e-5	1.4e-6	3.2e-6	1.9e-5

Table 2 shows that the data error reflects error at each point, so it is a good indicator of the real error. Of course the numerical result is characterized in the values of parameters. The following is the list of the 42 parameters at a step when the data error is 5.0×10^{-5}.

$$
\begin{aligned}
\varepsilon^1 &= (0, 0, 0.2518, 0, -1.6165, -0.0283) \\
\varepsilon^2 &= (0.3114, 0.0477, -0.2843, -1.2158, -0.7438, -0.0166) \\
\varepsilon^3 &= (0.0702, 0.0906, 0.0253, -0.1699, -3.5317, 0.4295) \\
\varepsilon^4 &= (-5.4555, 27.2645, 0.0012, -2.4305, 0.9266, 0.5495) \\
\varepsilon^5 &= (-4.0792, 2.4244, 0.0124, -0.9771, -1.0142, 0.4039) \\
\varepsilon^6 &= (0.2032, 0.0093, -0.0104, 0.0890, -4.8000, 0.7450) \\
\varepsilon^7 &= (-1.2717, 3.1953, -0.0997, -0.2840, 0.3282, 0.8198)
\end{aligned}
$$

Thus we have a good approximation on the domain,

$$u \approx \varepsilon^1 \cdot \phi_1 + \ldots + \varepsilon^6 \cdot \phi_6$$

or indeed,

$$\begin{aligned}
&\frac{1}{4}\exp(x+t) - \exp(-\pi^2 t)\sin(\pi x) + 3\exp(-4t)\cos(2x) \\
&\approx \varepsilon^0 \cdot 1 + \varepsilon^1 \cdot x + \varepsilon^2 \cdot (t + \frac{1}{2}x^2) + \varepsilon^3 \cdot (tx + \frac{1}{3}x^3) \\
&+ \varepsilon^4 \cdot (t^2 + tx^2 + \frac{1}{12}x^4) + \varepsilon^5 \cdot (t^2 x + \frac{1}{3}tx^3 + \frac{1}{60}x^5) \\
&+ \varepsilon^6 \cdot (t^3 + \frac{3}{2}t^2x^2 + \frac{1}{24}tx^4 + \frac{1}{120}x^6)
\end{aligned}$$

Note that the two functions on the both sides of the approximated equality are all solutions of the heat equation.

Example 4.2. Scattered Data Condition

There are many numerical methods that can deal with the data scattered in space with good efficiency, such as radial basis functions[8, 9].

However, in many practical situations, the data streaming could come scattered both in time and space. For example of atmospheric data assimilation, the observation plane in the air measures the data round-the-clock, moving its position. There are still not many methods for this kind of ill-conditioned problems.

One of the benefit of the symmetric collocation method is that the method can work with any data condition no matter what geometric restriction it has. Suppose we have the data totally scattered in time and space as in Figure 2(b). Even though the distribution of the data locations is quite different from that in the previous example, the symmetric collocation method works the same way.

Table 3 : Point-wise Error according to the Data Error in Example 4.2

Data Error	Error at (x,t)					
$E_{\text{data}}(u)$	(0.25,0.3)	(0.5,0.3)	(0.75,0.3)	(0.25,0.5)	(0.5,0.5)	(0.75,0.5)
4.7e-3	6.1e-3	3.9e-3	2.8e-3	7.6e-3	5.2e-3	2.5e-3
4.9e-4	6.8e-4	4.6e-4	4.6e-4	5.0e-4	5.2e-4	4.8e-4
5.0e-5	4.1e-5	2.1e-5	1.8e-5	6.5e-5	4.4e-5	6.6e-6

The following is the list of the parameters at a step when the data error is 5.0×10^{-5}.

$$\begin{aligned}
\varepsilon^1 &= (0, 0, 4.4607, 0, 1.7378, 2.0731) \\
\varepsilon^2 &= (0.5493, -1.6470, -0.1371, 0.5961, -1.2232, -0.1353) \\
\varepsilon^3 &= (0.2455, -0.8448, 0.4464, 0.0218, -0.3851, -0.1911) \\
\varepsilon^4 &= (1.5755, 41.1203, -0.0008, -1.8603, 2.1399, 1.8788) \\
\varepsilon^5 &= (2.3376, 1.9193, -0.7298, -0.4620, -0.2444, 0.1288) \\
\varepsilon^6 &= (-1.3951, 0.3192, 0.5335, 3.4575, -3.6871, 0.3692) \\
\varepsilon^7 &= (-4.8248, 0.4228, -0.0298, -0.4769, 0.0082, 1.3220)
\end{aligned}$$

In the result we see again that the result made out of seven polynomial solutions is surprisingly close to the target solution. Also the data error reflects error at the each point better than in Example 4.1.

While comparing Table 2 and 3 tells that the method performs almost equal results in both cases, it is important to note that two sets of parameters corresponding the same error level are very different. It suggests that the optimization problems produced from the method might have many solutions up to some level and the method is strong enough to search the solution space.

Though the results of other numerical schemes are often highly questionable for ill-posed data problems, the result of the symmetric collocation method is much more trustable as far as the data error is small, which we can always easily confirm.

There are remarks on the symmetric collocation method as an optimization. First we can adopt any optimization algorithm based on gradient method, since a partial derivative of the data error is easily obtainable as,

$$\frac{\partial E_{\text{data}}(u)}{\partial {\varepsilon^i}_j} = \frac{1}{\sqrt{N}} \mid u - g \mid_{2,D}^{-1} \sum_{x \in D} (u(x) - g(x)) \times \frac{\partial(\varepsilon^i \cdot \phi_i)}{\partial {\varepsilon^i}_j}$$

where $u = \sum \varepsilon^i \cdot \phi_i$. Also it is a highly flexible optimization. We can try the procedure repeatedly with an updated data condition, even changing a solution set. This is possible due to linearity of a given PDE.

However, even though the method often comes with nice explicit functions whose gradients are computable, the computational cost is relatively expensive compared to the standard methods. This is why usefulness of the symmetric collocation methods should be found in ability to deal with ill-conditioned problems, sometimes extremely ill-posed far beyond ability of the standard schemes.

5. Conclusion and Future Work

While the most studies so far on applications of symmetry methods to the numerical schemes have focused on finite difference schemes, the introduced new method adopts collocation scheme. Through optimization process, the method constructs out of known solutions an analytic solution that is close to the target one.

One of the most attractive features of the symmetric collocation methods is that it does not suffer from ill-located data conditions. The method is purely based on collocations, so it does not require any discretiztion even for time dependant equations. The idea of method is simple, and its application is so flexible that one can combine it with other numerical methods.

Further research should provide clearer relation between the type of symmetries and the efficiency of the method. Dependance of computational cost on initial and boundary conditions should be investigated to improve its data-fitting process as well.

References

1. E. A. Galperin and Q. Zheng, *Solution and Control of PDE via Global Optimization Methods*, Computers Math. Applic. Vol. 25, No. 10/11 (1993), pp. 103-118.

2. E. A. Galperin, Q. Zheng and Z. Pan, *Application of Global Optimization to Implicit Solution of Partial Differential Equations*, Computers Math. Applic. Vol. 25, No. 10/11 (1993), pp. 119-124.
3. C. J. Budd and C. B. Collins, *Symmetry based numerical methods for partial differential equations*, in D. F. Griffiths, D. J. Higham and G. A. Watson (eds), Numerical analysis, Pitman Res. Notes Math., vol. 380, Longman, Harlow, 1998, pp. 16-36.
4. C. J. Budd and A. Iserles, *Geometric integration: numerical solution of differential equations on manifolds*, Phil. Trans. Roy. Soc. London A, 357 (1999), pp. 945-956.
5. C. J. Budd and V. A. Dorodnitsyn, *Symmetry adapted moving mesh schemes for the nonlinear Schrodinger equation*, J. Phys. A, Math. Gen., 34 (48) (2001), pp. 10387-10400.
6. V. A. Dorodnitsyn, *Finite difference models entirely inheriting continuous symmetry of original differential equations*, Int. J. Mod. Phys. C, 5 (1994), pp. 723-724.
7. V. A. Dorodnitsyn, R. Kozlov and P. Winternitz, *Lie group classification of second order difference equations*, J. Math. Phys., 41(1) (2000), pp. 480-504.
8. C. Franke and R. Schaback, *Solving partial differential equation by collocation using radial basis function*, Appl. Math. Comput. 93 (1998), pp. 73-82.
9. E. Fornberg and E. Larsson,*A numerical study of some radial basis function based solution methods for elliptic PDEs*, Comp. Math. App., 46 (2003), pp. 891-902.
10. M. Fels and P. J. Olver, *Moving coframes. I. a practical algorithm*, Acta Appl. Math., 51 (1998), pp. 161-213.
11. M. Fels and P. J. Olver, *Moving Coframes. II. regularization and theoretical foundations*, Acta Appl. Math., 55 (1999), pp. 127-208.
12. E. Hairer, Ch. Lubich, and G. Wanner, Geometric Numerical Integration Structure Preserving Algorithms for Ordinary Differential Equations, Springer Series in Computational Mathematics 31, Springer-Verlag, 2002.
13. P.E. Hydon, *Discrete point symmetries of ordinary differential equations*, Proc. Roy. Soc. Lond. A, 454 (1998), pp. 1961-1972.
14. P. E. Hydon, *Symmetries and first integrals of ordinary difference equations*, Proc. Roy. Soc. Lond. A, 456 (2000), pp. 2835-2855.
15. N. H. Ibragimov (eds), CRC handbook of Lie group to differential equations. V.1. Symmetries, exact solutions and conservation laws, CRC Press, 1994.
16. D. Levi, L. Vinet, and P. Winternitz, *Lie group formalism for difference equations*, J. Phys. A, Math. Gen., 30 (1997), pp. 633-649.
17. P. J. Olver, Applications of Lie Groups to Differential Equations, Second Edition, Graduate Texts in Mathematics, vol. 107, Springer-Verlag, New York, 1993.
18. P. J. Olver, *Geometric foundations of numerical algorithms and symmetry*, Appl. Alg. Engin. Comp. Commun., 11 (2001), pp. 417-436.
19. P. Kim, *Invariantization of Numerical Schemes using Moving Frames*, submitted.

The School of Mathematics, University of Minnesota
E-mail address: pwkim@math.umn.edu