Invariant Discretization Schemes
Using Evolution–Projection Techniques[⋆]

Alexander BIHLO [†‡] *and Jean-Christophe NAVE* [‡]

[†] *Centre de recherches mathématiques, Université de Montréal,*
   C.P. 6128, succ. Centre-ville, Montréal (QC) H3C 3J7, Canada
   E-mail: *bihlo@crm.umontreal.ca*

[‡] *Department of Mathematics and Statistics, McGill University,*
   805 Sherbrooke W., Montréal (QC) H3A 2K6, Canada
   E-mail: *jcnave@math.mcgill.ca*

Abstract. Finite difference discretization schemes preserving a subgroup of the maximal Lie invariance group of the one-dimensional linear heat equation are determined. These invariant schemes are constructed using the invariantization procedure for non-invariant schemes of the heat equation in computational coordinates. We propose a new methodology for handling moving discretization grids which are generally indispensable for invariant numerical schemes. The idea is to use the invariant grid equation, which determines the locations of the grid point at the next time level only for a single integration step and then to project the obtained solution to the regular grid using invariant interpolation schemes. This guarantees that the scheme is invariant and allows one to work on the simpler stationary grids. The discretization errors of the invariant schemes are established and their convergence rates are estimated. Numerical tests are carried out to shed some light on the numerical properties of invariant discretization schemes using the proposed evolution–projection strategy.

Key words: invariant numerical schemes; moving frame; evolution–projection method; heat equation

2010 Mathematics Subject Classification: 65M06; 58J70; 35K05

1   Introduction

Discretization schemes for differential equations that are not solely constructed for the sake of reducing the local discretization error as much as possible, but rather to preserve some of the intrinsic properties of these differential equations have become increasingly popular over the last decades. While preserving one of these properties, namely conservation laws, led to the design of conservative discretization schemes which are quite popular in the scientific community [4, 19, 24] (and in particular in the geosciences, e.g. [18, 31]), there are other geometric features of differential equations that can be attempted to be preserved as well that have received less attention from the side of numerical analysis so far. One of these features are symmetries of differential equations. While there have been theoretical advancements on the methodologies of finding numerical schemes that preserve the maximal Lie invariance groups of systems of differential equations over the past 20 years or so [13, 22, 30, 35], little is known about the numerical properties of these invariant schemes. A part of the problem is that while conservation laws are always properties of the solutions of a differential equation, symmetries are by definition properties of differential equations. Therefore, it is a standing question whether a discretization

---

scheme that preserves numerically a property of a differential equation also improves the quality of the numerical solution of that discretized differential equation.

The present paper is devoted to an investigation of this question and related problems exemplified with invariant discretization schemes for the linear heat equation. The heat equation is particularly suited for this investigation as it is a canonical example in the group analysis of differential and difference equations. Moreover, there are already several studies devoted to invariant numerical schemes for this equation [1, 14, 35]. At the same time, in none of these existing works a deeper background analysis of the numerical properties (e.g. order of approximation or stability) of the developed schemes was investigated. A first account on numerical properties of invariant numerical schemes for the heat equation was given in [12], in which a numerical comparison of invariant and non-invariant schemes for the heat equation regarding accuracy was presented.

There are several reasons why less attention has been paid so far on the numerical analysis of invariant schemes (with the exception of the works [10, 12]). One of the reasons is that the field of invariant discretization schemes is still in its early stages, with new conceptual algorithms being developed only recently [2, 5, 6, 13, 21, 22, 25, 30]. Another reason is that invariant finite difference schemes generally require the use of *adaptive moving meshes*, i.e. it is necessary to include a non-trivial mesh equation in the discretization problem. Moving meshes lead to non-uniform grid point distributions and, in the multi-dimensional case, to non-orthogonal grids. The analysis of schemes on such meshes is considerably more difficult than that for related difference schemes on fixed, uniform and orthogonal meshes. Due to this second reason, most invariant numerical schemes so far have been constructed only for (1+1)-dimensional single evolution equations, as in that case moving meshes can be handled still with limited effort. Although we will be concerned with a (1+1)-dimensional equation in the present paper too, the methods used subsequently can be employed in the multi-dimensional case without substantial modification.

The new approach we propose here is to use the invariant grid equation only for a single time step and then to interpolate the solution to the regular grid. The important observation is that this interpolation can be done in an invariant way, i.e. projecting the solution does not break the invariance of the scheme itself. At the same time, the possibility to project the solution of an invariant scheme to a regular grid is highly desirable as in the multi-dimensional case a freely evolving grid can cause severe numerical problems. Moreover, for realistic numerical models, as e.g. employed in weather and climate predictions, it is in general hard to use adaptive meshes as the discretization of the governing equations is only one part of such model. Other parts are related to the numerical data assimilation, i.e. the preparation of the initial conditions for the numerical model and this step usually involves the forecasting model itself. As the assimilation of the initial conditions cannot be done on an evolving mesh (because the data are given at fixed locations only) this at once renders invariant schemes on moving meshes impractical. Equally important, any realistic numerical model for a nonlinear system of partial differential equations has to contain subgrid-scale models, which mimic the effects of processes taking place at those scales that the numerical model is not capable of resolving explicitly [33, 34]. The construction of subgrid-scale models for non-resolved physical processes involves in general ad-hoc arguments and these arguments rely on the particular scale on which the unresolved processes take place. As a moving mesh locally changes the resolution and thus impacts what processes are explicitly resolved by a numerical model, subgrid-scale schemes have to be designed that can operate on grids with varying resolution. For realistic processes (which are usually not self-similar), this might be difficult to achieve in practice.

All what was said above objects against invariant numerical schemes for multi-dimensional systems of differential equations using freely evolving meshes. Thus, whether mathematically feasible or not, such schemes would be of less practical interest. This is why other approaches

should be sought that on the one hand allow one to retain the invariance group of a system of differential equations in the course of discretization and on the other hand yield schemes that are practical to avoid the above mentioned and related problems. The proposed invariant *evolution–projection strategy* we are going to introduce below may be considered as one such approach.

The further organization of this article is as follows. The subsequent Section 2 features a summary and some extensions on the various methods to construct invariant discretization schemes. In Section 3 the heat equation along with its maximal Lie invariance group G is presented. It is discussed which subgroup G^1 of G we aim to present when constructing invariant numerical discretization schemes. The selection of G^1 is based on preserving the class of periodic boundary value problems we are focussing on. Section 4 contains the construction of an equivariant moving frame for G^1 along with a presentation of some lower order differential invariants of G^1. In Section 5 invariant discretization schemes for the heat equation in computational coordinates are found. The local discretization errors of these schemes are established in Section 6. In Section 7 we introduce the new idea of invariant interpolation schemes that can be used to project the numerical solution obtained from an invariant scheme on a moving mesh to the regular grid. The numerical analysis as well as some numerical tests for the schemes proposed in this paper are found in Section 8. The summary of this article is presented in the final Section 9.

2   Construction of invariant discretization schemes

The construction of invariant discretization schemes for differential equations can be seen as a part of the ongoing effort to turn group analysis into an efficient tool for the analysis of difference equations, see e.g. the review article [25]. As of now, there are three main methods that are used to construct invariant discretization schemes.

2.1   Difference invariant method

The first method was developed by V. Dorodnitsyn, see [1, 13, 14, 25, 35]. It uses the infinitesimal generators of one-parameter symmetry groups of the system of differential equations under consideration that span the maximal Lie invariance algebra $\mathfrak{g}$ of this system. These generators are of the form

$$\mathrm{v} = \zeta^j(x, u)\partial_{x^j} + \eta^\alpha(x, u)\partial_{u^\alpha} = \zeta(x, u)\partial_x + \eta(x, u)\partial_u,$$

where $x = (x^1, \ldots, x^p)$ and $u = (u^1, \ldots, u^q)$ are the tuples of independent and dependent variables, respectively. Here and in the following, the summation convention over repeated indices is used. Rather than prolonging v to higher order derivatives of u with respect to x, which is standard in the symmetry analysis of differential equations [3, 26, 29], in this method the vector fields are prolonged to all the points of the *discretization stencil*, i.e. the collection of grid points which are necessary to approximate a given system of differential equation up to a desired order. This prolongation is of the form

$$\mathrm{pr}\,\mathrm{v} = \sum_{i=1}^m \zeta(x_i, u_i)\partial_{x_i} + \eta(x_i, u_i)\partial_{u_i},$$

where $x_i = (x_i^1, \ldots, x_i^p)$ and $u_i = (u_i^1, \ldots, u_i^q)$, i.e. it is done by evaluating the vector field v at all m stencil points $z_i = (x_i, u_i)$ and summing up the result. An example for such a prolongation is given in Remark 1 in Section 5.

As a next step, the invariants of the group action are found by invoking the infinitesimal invariance criterion [13, 26], which in the present case is $\mathrm{pr}\,\mathrm{v}(I) = 0$. The functions I that fulfill this condition for all $\mathrm{v} \in \mathfrak{g}$ are termed *difference invariants*.

Once the difference invariants on the stencil space are found, one then has to assemble these invariants together to a finite difference approximation of the given system of differential equations. By construction, this procedure guarantees that the resulting numerical scheme is invariant under the symmetry group of the original system of differential equations.

The main drawback of this method is that it might be hard to find a combination of difference invariants that approximates a system of differential equations in the multi-dimensional case. The problem is, as discussed in the introduction, that invariant schemes generally require the use of moving and/or non-orthogonal grids. Formulating consistent discretization schemes using difference invariants as building blocks on moving meshes is rather challenging in higher dimensions and thus limited the application of this method to the case of single $(1+1)$-dimensional evolution equations. We stress though that this problem only enters at the stage of combining difference invariants to a discretization scheme. Computing difference invariants in the multi-dimensional case can be done as effectively as computing differential invariants for multi-dimensional problems using infinitesimal techniques.

2.2 Invariant moving mesh method

Retaining the invariance of finite difference schemes under the maximal Lie invariance groups of physically relevant time-dependent differential equations often requires the use of moving meshes. This is true both for the finite difference method discussed in the previous Section 2.1 and the moving frame method to be discussed in the next Section 2.3. This kind of mesh adaptation in which the number of grid points remains constant throughout the integration is referred to as r-adaptivity in the field of adaptive numerical schemes [7, 20].

The standard strategy to handle r-adaptive meshes is to regard the grid adaptation as a time-dependent mapping from a fixed reference space of *computational coordinates* to the physical space of the independent variables of the differential equation, i.e. $x = x(\xi)$ for $\xi = (\xi^1, \dots, \xi^p)$ being the computational variables. Without loss of generality, we assume that $\xi^1 = \tau = t$ is the time variable. The dependent variables u are expressed in the computational space by setting $\bar{u}(\xi) = u(x(\xi))$. For the sake of simplicity we will omit the bars henceforth.

The significance of the computational coordinates is to provide a reference frame that remains stationary and orthogonal even in the presence of grid adaptation in the physical space of coordinates. In the course of discretization the variable ξ labels the position of the grid points in the mesh and this labeling stays unchanged during the mesh adaptation. Thus, the computational variables can be interpreted physically as Lagrangian coordinates and their invariance under the motion of the grid is equivalent to the identity of fluid particles in ideal hydrodynamics.

Because by construction the grid remains orthogonal in the ξ-coordinates, the usual finite difference approximations for derivatives can be used in the space of computational variables. This simplifies both the practical implementation of the discretization method and the numerical analysis of the resulting schemes.

The expression of the initial physical system of differential equations in terms of computational variables leads to a system of equations that explicitly includes the mesh velocity x_τ, which is yet to be determined in order to close the resulting numerical scheme. A prominent strategy for determining the location of the grid points at the subsequent time level in the one-dimensional case uses the *equidistribution principle*, which in its differential form is $(\rho x_\xi)_\xi = 0$, where ρ is a monitor function that determines the areas of grid convergence and divergence. For higher-dimensional problems, equidistribution has to be combined with heuristic arguments, see [20] for more details.

The invariance of the initial differential equations is brought into the scheme by adequately specifying the monitor function ρ. In [6] it was pointed out that using monitor functions that preserving the scale-invariance of a differential equation is particularly relevant in cases where

the equation is capable of developing a blow-up solution in finite time, see also [7, 8, 20]. This finding is generalized upon requiring that the monitor function is chosen in a manner such that the equidistribution principle is invariant under the same symmetry group as the original differential equation. For a number of symmetry groups this appears to be possible, see [2] for an example.

The invariant moving mesh method was recently extended in [2]. The idea of this extension is to transform the initial system of differential equations to the space of computational coordinates and to determine the form of the symmetry transformations in the computational space. The equations in the computational space are then discretized such that the resulting scheme mimics the transformation behavior of the continuous case. The main advantage of this approach is that it allows one to retain an initial conserved form of the system of differential equations and thus to numerically preserve certain conservation laws in the invariant scheme. This is relevant as preserving conservation laws in the course of invariant numerical modeling is yet a pristine problem. An exception to this is the discretization of equations that follow from variational principles, which, if done in a proper way, can lead to the simultaneous preservation of both symmetries and associated conservation laws, owing to the discrete Noether theorem. See, e.g. [5] for an example of such an invariant Lagrangian discretization.

Another advantage of the extension proposed in [2] is that it allows one to find invariant numerical schemes without the detour of difference invariants. This is essential as it can happen that the single equations in a system of differential equations cannot be approximated directly in terms on differential invariants but only in combination with other equations of that system. If this is the case it is not natural to attempt to discretize the system using difference invariants as this would lead to rather cumbersome discretization schemes.

2.3   Moving frame method

The third method is the most recent one [10, 11, 21, 22, 23, 30]. It relies on the notion of *equivariant moving frames* and their important property to provide a mapping that allows one to associate an invariant function to any given function. As we will mostly work with this method in the present paper, we describe it in greater detail here. We collect some important notions on moving frames below, a more comprehensive exposition can be found in the original references [9, 15, 16, 27, 28, 30].

Definition 1. Let G be a finite-dimensional Lie group acting on a manifold M. A (*right*) *moving frame* ρ is a smooth map $\rho \colon M \to G$ satisfying the equivariance property

$$\rho(g \cdot z) = \rho(z)g^{-1}, \tag{1}$$

for all $z \in M$ and $g \in G$.

Theorem 1. *A moving frame exists in the neighborhood of a point $z \in M$ if and only if the group G acts freely and regularly near z.*

Local freeness of a group action means that $\tilde{z} = g \cdot z = z$ for all z from a sufficiently small neighborhood of each point on M only holds for g being the identity transformation, which implies that all the group orbits have the same dimension. Here and throughout the paper, a tilde over a variable denotes the corresponding transformed form of that variable. *Regularity* of a group action requires that there exists a neighborhood for each point $z \in M$, which is intersected by the orbits of G into a pathwise connected subset.

When a group G does not act freely on M, its action can be made free upon extending it to a suitably high-order *jet space* $J^n = J^n(M, p)$ of M, $0 \leq n \leq \infty$. Locally, the nth order jet space of a p-dimensional submanifold of M has coordinates $z^{(n)} = (x, u^{(n)})$, where as in the previous

subsections $x = (x^1, \ldots, x^p)$ are considered as the independent variables, $u = (u^1, \ldots, u^q)$, $q = \dim M - p$, are the dependent variables and $u^{(n)}$ collects all the derivatives of u with respect to x of order not greater than n including u as the zeroth order derivatives. In practice, the prolongation of the group action of G on J^n is implemented using the chain rule.

Moving frames are determined using a *normalization* procedure. The steps to find a moving frame for a group action G are the following: (i) Define a *cross-section* to the group orbits. A cross-section C is any submanifold $C \subset M$ of complementary dimension to the dimension r of the group orbits, i.e. $\dim C = \dim M - r$, that intersects each group orbit once and transversally. Usually coordinate cross-sections are chosen in which some of the coordinates of M (or of J^n if the group action is not free on M) are set to constants, i.e. $z^1 = c^1, \ldots, z^r = c^r$. (ii) The algebraic system $\tilde{z}^1 = (g \cdot z)^1 = c^1, \ldots, \tilde{z}^r = (g \cdot z)^r = c^r$ is solved for the group parameters $g = (g_1, \ldots, g_r)$. The resulting expression $g = \rho(z)$ is the moving frame.

Moving frames can be used to map any given function to an invariant function by a procedure called *invariantization*.

Definition 2. The *invariantization* of a real-valued function $f \colon M \to \mathbb{R}$ using the (right) moving frame ρ is the function $\iota(f)$, which is defined as $\iota(f)(z) = f(g \cdot z)|_{g=\rho(z)} = f(\rho(z) \cdot z)$.

That the function $\iota(f)$ constructed in this way is indeed invariant follows from the equivariance property (1) of the moving frame ρ,

$$\iota(f)(g \cdot z) = f(\rho(g \cdot z)g \cdot z) = f(\rho(z)g^{-1}g \cdot z) = f(\rho(z) \cdot z) = \iota(f)(z),$$

which boils down to the definition of an invariant function I, i.e. $I(g \cdot z) = I(z)$. In practice, a function $f(z)$ is invariantized by first transforming its argument using the transformations from G and then substituting the moving frame for the group parameters. By definition, an invariant that is defined on the jet space J^n is a *differential invariant*.

Moving frames can also be constructed on a discrete space. In a finite difference approximation, derivatives of functions are approximated using a finite set of values of these functions, and all the points needed to approximate the derivatives arising in a system of differential equations are the points of the stencil introduced in Section 2.1. Because most of the interesting symmetries of differential equations that are broken in standard numerical schemes require the use of non-orthogonal discretization meshes, it is beneficial to both regard x and u as the dependent variables and the computational variables ξ as the independent variables as was discussed in the previous Section 2.2.

Sampling the tuples from the *extended computational space* $M_\xi = \{(\xi, z)\}$ at discrete points, i.e. at $(\xi_i, z(\xi_i)) = (\xi_i, z_i)$, one can introduce the space $M_\xi^{\diamond n} = \{(w_1, \ldots, w_n) \,|\, \xi_i \neq \xi_j \text{ for all } i \neq j\}$, where $w_i = (\xi_i, z_i)$, which can be identified as the joint product space of stencil variables. Because the identifier ξ_i of the point w_i is required to be unique, each element of $M_\xi^{\diamond n}$ only includes distinct grid points in the physical space of equation variables. The dimension of the space $M_\xi^{\diamond n}$ depends on the number of independent and dependent variables in the system of differential equations and the desired order of accuracy of the approximated derivatives.

It is possible to carry out the construction of the moving frame on $M_\xi^{\diamond n}$, i.e. to define the moving frame by an equivariant mapping $\rho_\xi^{\diamond n} \colon M_\xi^{\diamond n} \to G$, where G acts on $M_\xi^{\diamond n}$ by the product action, $g \cdot (w_1, \ldots, w_n) = (g \cdot w_1, \ldots, g \cdot w_n)$. Note that the extension of the group action to the computational variables ξ is trivial, i.e. they remain unaffected by G, $\tilde{\xi} = g \cdot \xi = \xi$, see [2]. The compatibility between the moving frame $\rho_\xi^{\diamond n}$ and the moving frame ρ on the space M (or an appropriate jet space J^n), i.e. that $\rho_\xi^{\diamond n} \to \rho$ in the continuous limit is assured provided that the cross-section defining the moving frame $\rho_\xi^{\diamond n}$ in the continuous limit converges to the cross-section defining the moving frame ρ. Once the moving frame is constructed on the discrete space $M_\xi^{\diamond n}$ of stencil variables, it can be used to invariantize any numerical scheme expressed

in computational coordinates. This will be explicitly shown in Sections 5 and 6 where we will construct invariant schemes for the heat equation.

It is essential that the construction of the moving frame on the grid point space is carried out in terms of computational coordinates rather than physical coordinates. This can be illustrated by the following simple example.

Example 1. The Laplace equation $u_{xx} + u_{yy} = 0$ is, inter alia, invariant under the one-parameter group of rotations SO(2), $\tilde{x} = x \cos \varepsilon - y \sin \varepsilon$, $\tilde{y} = x \sin \varepsilon + y \cos \varepsilon$. Let us obtain the moving frame ρ for this group action from the normalization condition $u_x = 0$, i.e. we determine the moving frame on the first jet space $J^1(M, 2)$, $\rho = \rho(x, y, u, u_x, u_y)$. Prolonging the transformations from SO(2) to the derivative u_x leads to $\tilde{u}_{\tilde{x}} = u_x \cos \varepsilon - u_y \sin \varepsilon$ and thus the moving frame is $\varepsilon = \arctan(u_x/u_y)$.

Let us now find the product frame from the discrete normalization condition $u_x^{\mathrm{d}} = 0$. Computing u_x^{d} in the naïve way, $u_x^{\mathrm{d}} = (u_{i+1} - u_{i-1})/(x_{i+1} - x_{i-1})$, we fail as

$$\tilde{u}_{\tilde{x}}^{\mathrm{d}} = \frac{u_{i+1} - u_{i-1}}{(x_{i+1} - x_{i-1})\cos \varepsilon - (y_{i+1} - y_{i-1})\sin \varepsilon} = 0,$$

cannot be solved for the group parameter ε. On the other hand, setting $u = u(x(\xi^1, \xi^2), y(\xi^1, \xi^2))$ and expressing u_x^{d} in terms of the computational variables ξ^1, ξ^2, the normalization $u_x^{\mathrm{d}} = 0$ reads

$$\tilde{u}_{\tilde{x}}^{\mathrm{d}} = \frac{\tilde{u}_{\xi^1}^{\mathrm{d}} \tilde{y}_{\xi^2}^{\mathrm{d}} - \tilde{u}_{\xi^2}^{\mathrm{d}} \tilde{y}_{\xi^1}^{\mathrm{d}}}{\tilde{x}_{\xi^1}^{\mathrm{d}} \tilde{y}_{\xi^2}^{\mathrm{d}} - \tilde{x}_{\xi^2}^{\mathrm{d}} \tilde{y}_{\xi^1}^{\mathrm{d}}} = \frac{u_{\xi^1}^{\mathrm{d}}\left(x_{\xi^2}^{\mathrm{d}} \sin \varepsilon + y_{\xi^2}^{\mathrm{d}} \cos \varepsilon\right) - u_{\xi^2}^{\mathrm{d}}\left(x_{\xi^1}^{\mathrm{d}} \sin \varepsilon + y_{\xi^1}^{\mathrm{d}} \cos \varepsilon\right)}{x_{\xi^1}^{\mathrm{d}} y_{\xi^2}^{\mathrm{d}} - x_{\xi^2}^{\mathrm{d}} y_{\xi^1}^{\mathrm{d}}} = 0.$$

This expression can be solved for ε and it gives

$$\varepsilon = \arctan\left(\frac{u_{\xi^1}^{\mathrm{d}} y_{\xi^2}^{\mathrm{d}} - u_{\xi^2}^{\mathrm{d}} x_{\xi^1}^{\mathrm{d}}}{u_{\xi^2}^{\mathrm{d}} y_{\xi^1}^{\mathrm{d}} - u_{\xi^1}^{\mathrm{d}} x_{\xi^2}^{\mathrm{d}}}\right) = \arctan\left(\frac{u_x^{\mathrm{d}}}{u_y^{\mathrm{d}}}\right),$$

which in the continuous limit goes to $\varepsilon = \arctan(u_x/u_y)$ as required.

3 Lie symmetries of the heat equation

The one-dimensional linear heat transport equation is

$$u_t - u_{xx} = 0, \tag{2}$$

where we scaled the thermal diffusivity ν to 1, i.e. equation (2) is in non-dimensional form.

The heat equation (2) admits the following infinitesimal generators of one-parameter groups, which generate the maximal Lie invariance algebra $\mathfrak{g}$ of equation (2):

$$\begin{aligned}
&\partial_t, \quad \partial_x, \quad u\partial_u, \quad 2t\partial_t + x\partial_x, \quad 2t\partial_x - xu\partial_u, \\
&4t^2\partial_t + 4tx\partial_x - (x^2 + 2t)u\partial_u, \quad \alpha(t, x)\partial_u,
\end{aligned} \tag{3}$$

where α runs through the set of solutions of equation (2), see e.g. [26]. These vector fields generate (i) time-translations, (ii) space translations, (iii) scalings in u, (iv) simultaneous scalings in t and x, (v) Galilean boosts, (vi) inversions and (vii) the superposition principle symmetry.

In this paper, we will construct invariant numerical schemes for a class of initial value problems of the heat equation using periodic boundary conditions. This class of initial-boundary value problems only admits a subgroup of the symmetry group of the heat equation as inversions are no longer admitted; inversions do not send an initial value problem from the considered class to another initial value problem. The symmetries associated with the first five vector fields in (3)

are compatible with the class of initial-boundary value problems we are interested in, i.e. they map the class of initial-boundary value problems for the heat equation under consideration to itself. This re-interpretation of symmetries of differential equations without initial and boundary conditions as equivalence transformations for a class of initial-boundary value problems was recently pointed out in [2].

In what follows we will thus focus our attention on constructing numerical schemes that preserve the symmetries generated by the first five operators of (3). The associated subalgebra of $\mathfrak{g}$ will be denoted by $\mathfrak{g}^1$. We do not require to preserve the linearity operator here by construction. At the same time, as will be shown in Section 8 the numerical schemes we propose in this paper preserve the linearity property up to the discretization error expected.

4 Moving frame and differential invariants for the heat equation

We determine the moving frame for the subgroup G^1 of transformations associated with the subalgebra $\mathfrak{g}^1$. Transformations of G^1 are of the form

$$\tilde{t} = e^{2\varepsilon_4}(t + \varepsilon_1), \qquad \tilde{x} = e^{\varepsilon_4}(x + \varepsilon_2 + 2\varepsilon_5 t), \qquad \tilde{u} = e^{\varepsilon_3 - \varepsilon_5 x - \varepsilon_5^2 t}u. \tag{4}$$

Because the group action of G^1 is not free on $M = \{(t, x, u)\}$ we construct the moving frame on a suitably high-order jet space. In the present case, the group action of G^1 becomes free on $J^1 = J^1(M, 2)$. Thus, it is necessary to extend the transformations (4) to derivatives of u with respect to t and x.

Using the chain rule we can compute the transformed operators of total differentiation, which read as

$$\mathrm{D}_{\tilde{t}} = e^{-2\varepsilon_4}(\mathrm{D}_t - 2\varepsilon_5\mathrm{D}_x), \qquad \mathrm{D}_{\tilde{x}} = e^{-\varepsilon_4}\mathrm{D}_x,$$

where $\mathrm{D}_x = \partial_x + u_x\partial_u + u_{tx}\partial_{u_t} + u_{xx}\partial_{u_x} + \cdots$ and $\mathrm{D}_t = \partial_t + u_t\partial_u + u_{tt}\partial_{u_t} + u_{tx}\partial_{u_x} + \cdots$ denote the usual operators of total differentiation. With the transformed total differentiation operators at hand it is possible to compute the transformed partial derivatives of u with respect to t and x. The transformation rules for the lowest order derivatives are

$$\tilde{u}_{\tilde{t}} = e^{-2\varepsilon_4 + \varepsilon_3 - \varepsilon_5 x - \varepsilon_5^2 t}\big(u_t - 2\varepsilon_5 u_x + \varepsilon_5^2 u\big), \qquad \tilde{u}_{\tilde{x}} = e^{-\varepsilon_4 + \varepsilon_3 - \varepsilon_5 x - \varepsilon_5^2 t}\big(u_x - \varepsilon_5 u\big),$$
$$\tilde{u}_{\tilde{x}\tilde{x}} = e^{-2\varepsilon_4 + \varepsilon_3 - \varepsilon_5 x - \varepsilon_5^2 t}\big(u_{xx} - 2\varepsilon_5 u_x + \varepsilon_5^2 u\big).$$

In fact, for the construction of the moving frame already the knowledge of the first order derivatives is sufficient.

We compute the moving frame for the five-parameter group of transformations of the form (4) using the following normalization conditions which determine a valid cross-section to the group orbits of the prolonged action of G^1 on J^1,

$$t = 0, \qquad x = 0, \qquad u = 1, \qquad u_t = 1, \qquad u_x = 0. \tag{5}$$

The moving frame is computed by taking the transformed form of the normalization conditions, $\tilde{t} = 0$, $\tilde{x} = 0$, $\tilde{u} = 1$, $\tilde{u}_{\tilde{t}} = 1$ and $\tilde{u}_{\tilde{x}} = 0$ and by solving the resulting algebraic system for the group parameters $\varepsilon_1, \ldots, \varepsilon_5$. The result of this computation is the following moving frame $g = \rho(z^{(1)})$,

$$\varepsilon_1 = -t, \qquad \varepsilon_2 = -\left(x + 2t\frac{u_x}{u}\right), \qquad \varepsilon_3 = -\left(\ln u - x\frac{u_x}{u} - t\frac{u_x^2}{u^2}\right),$$
$$\varepsilon_4 = \ln\sqrt{\frac{u_t}{u} - \frac{u_x^2}{u^2}}, \qquad \varepsilon_5 = \frac{u_x}{u}. \tag{6}$$

With the moving frame at hand, we can invariantize any of the partial derivatives of u with respect to t and x and thus obtain a complete set of differential invariants for the subgroup G^1 of the maximal Lie invariance group of the heat equation. As an example, invariantizing the derivative u_{xx}, i.e. computing $\iota(u_{xx})$ as $(g \cdot u_{xx})|_{g=\rho(z^{(1)})}$ we produce the differential invariant

$$\iota(u_{xx}) = \frac{uu_{xx}^2 - u_x^2}{uu_t - u_x^2}.$$

Invariantizing the heat equation, i.e. computing $\iota(u_t - u_{xx}) = 0$ and recalling that $\iota(u_t) = 1$, we obtain

$$\frac{u(u_t - u_{xx})}{uu_t - u_x^2} = 0,$$

which yields the original heat equation expressed in terms of differential invariants. This re-expression of a differential equation using the differential invariants of its symmetry group is known as the *replacement theorem* [9].

5 Invariant discretization of the heat equation

The invariant discretization of equation (2) cannot be done on a fixed, uniform grid. To see this, let us check the transformation behavior of the grid equation $x_i^{n+1} - x_i^n = 0$, which is the definition of a stationary grid, under the transformations (4). This yields

$$\tilde{x}_i^{n+1} - \tilde{x}_i^n = e^{\varepsilon_4}\big(x_i^{n+1} - x_i^n + 2\varepsilon_5\big(t^{n+1} - t^n\big)\big),$$

which is only zero in the case when $\varepsilon_5 = 0$. Stated in another way, a discretization on a fixed grid can at most preserve the symmetry subgroup of G, which is generated by the first four elements of the maximal Lie invariance algebra $\mathfrak{g}$ of the heat equation (3).

Thus, the discretization of (2) preserving G^1 will require the use of moving grids. For this reason it is convenient to express (2) in terms of computational coordinates initially, i.e. we set $u(\tau, \xi) = u(\tau, x(\tau, \xi))$, where ξ is the single spatial computational variable and $\tau = t$. The heat equation in this set of coordinates reads

$$u_\tau - x_\tau \frac{u_\xi}{x_\xi} - \frac{1}{x_\xi^2}\left(u_{\xi\xi} - \frac{x_{\xi\xi}}{x_\xi}u_\xi\right) = 0. \tag{7}$$

So as to find the invariant discretization of the heat equation in the form (7), we determine the moving frame in the space of stencil variables $M_\xi^{\diamond 4}$ using the discrete analogs of the normalization conditions (5) expressed in terms of computational coordinates.

For the sake of convenience we introduce the notation $h^+ = x_{i+1}^n - x_i^n$, $h^- = x_i^n - x_{i-1}^n$, $\Delta\tau = \tau^{n+1} - \tau^n$. The discretization stencil we use is depicted in Fig. 1.

The appropriate normalization conditions for a compatible moving frame $\rho_\xi^{\diamond 4}$ are

$$\tau^n = 0, \qquad x_i^n = 0, \qquad u_i^n = 1, \qquad u_t^{\mathrm{d}} = 1, \qquad u_x^{\mathrm{d}} = 0, \tag{8}$$

where

$$u_t^{\mathrm{d}} = \frac{u_i^{n+1} - u_i^n}{\Delta\tau} - x_\tau^{\mathrm{d}}\frac{u_{i+1}^n - u_{i-1}^n}{h^+ + h^-}, \qquad u_x^{\mathrm{d}} = \frac{u_{i+1}^n - u_{i-1}^n}{h^+ + h^-}$$

are the discretizations of the first time and space derivatives expressed in computational coordinates and $x_\tau^{\mathrm{d}} = (x_i^{n+1} - x_i^n)/\Delta\tau$ is the discrete grid velocity. Replacing the single equations
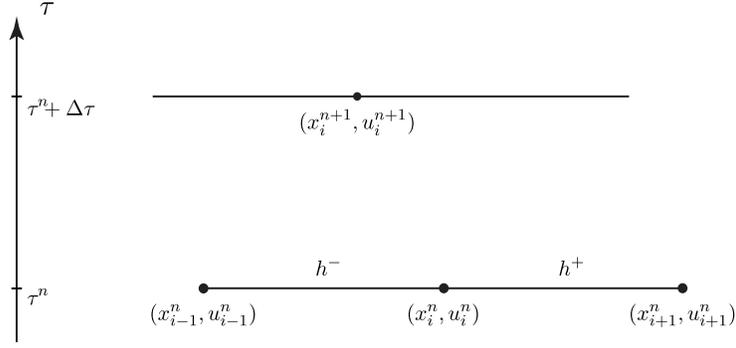
Figure 1. Stencil for an invariant discretization scheme of the heat equation.

in the above normalization conditions by their respective transformed expressions and solving the resulting algebraic system for the group parameters we obtain the following moving frame on the space of stencil variables $M_\xi^{\diamond 4}$,

$$\varepsilon_1 = -\tau^n, \qquad \varepsilon_2 = -\left(x_i^n + 2\tau^n(\ln u)_x^{\mathrm{d}}\right), \qquad \varepsilon_3 = -\left(\ln u_i^n - x_i^n(\ln u)_x^{\mathrm{d}} - \tau^n\left((\ln u)_x^{\mathrm{d}}\right)^2\right),$$

$$\varepsilon_4 = \frac{1}{2}\ln\left(\frac{\exp\left(-\Delta\tau\left(x_\tau^{\mathrm{d}}(\ln u)_x^{\mathrm{d}} + \left((\ln u)_x^{\mathrm{d}}\right)^2\right)\right)u_i^{n+1} - u_i^n}{u_i^n\Delta\tau}\right), \qquad \varepsilon_5 = (\ln u)_x^{\mathrm{d}}, \tag{9}$$

where we introduced $(\ln u)_x^{\mathrm{d}} = (\ln u_{i+1}^n - \ln u_{i-1}^n)/(h^+ + h^-)$. This moving frame is compatible with the moving frame (6) in that it converges to (6) in the continuous limit $\Delta\xi \to 0$ and $\Delta\tau \to 0$ upon using

$$h^+ = x_\xi\Delta\xi + x_{\xi\xi}(\Delta\xi)^2/2 + O\left(\Delta\xi^3\right), \qquad h^- = x_\xi\Delta\xi - x_{\xi\xi}(\Delta\xi)^2/2 + O\left(\Delta\xi^3\right),$$
$$x_i^{n+1} = x_i^n + x_\tau\Delta\tau + O\left(\Delta\tau^2\right), \qquad u_{i+1}^n = u_i^n + u_\xi\Delta\xi + u_{\xi\xi}(\Delta\xi)^2/2 + O\left(\Delta\xi^3\right), \tag{10}$$
$$u_{i-1}^n = u_i^n - u_\xi\Delta\xi + u_{\xi\xi}(\Delta\xi)^2/2 - O\left(\Delta\xi^3\right), \qquad u_i^{n+1} = u_i^n + u_\tau\Delta\tau + O\left(\Delta\tau^2\right).$$

The moving frame (9) can now be used to invariantize any non-invariant finite difference discretization of (7) on $M_\xi^{\diamond 4}$. To illustrate this, we invariantize the standard FTCS (forward in time centered in space) scheme

$$u_t^{\mathrm{d}} - \frac{4}{(h^+ + h^-)^2}\left(u_{i+1}^n + u_{i-1}^n - 2u_i^n - (h^+ - h^-)u_x^{\mathrm{d}}\right) = 0.$$

This is done by first replacing all terms by their respective transformed expressions and substituting the moving frame (9) for the arising group parameters. The result of this procedure is the invariant scheme

$$S = \frac{\exp\left(-\Delta\tau\left(x_\tau^{\mathrm{d}}(\ln u)_x^{\mathrm{d}} + ((\ln u)_x^{\mathrm{d}})^2\right)\right)u_i^{n+1} - u_i^n}{\Delta\tau}$$
$$- \frac{4\left(u_{i+1}^n\left(\frac{u_{i+1}^n}{u_{i-1}^n}\right)^{-h^+/(h^++h^-)} + u_{i-1}^n\left(\frac{u_{i+1}^n}{u_{i-1}^n}\right)^{h^-/(h^++h^-)} - 2u_i^n\right)}{(h^+ + h^-)^2} = 0. \tag{11}$$

Again, it can be checked that the above scheme (11) indeed converges to (7) in the limit of $\Delta\xi \to 0$ and $\Delta\tau \to 0$. This will be shown explicitly in Section 6, where we establish the order of approximation of (11).

So as to complete the scheme (11) it is necessary to determine x_i^{n+1}, which is the ingredient missing in (11). There are different ways to determine a grid equation, such as using the equidistribution principle as outlined in Section 2.2. The problem with this strategy in the present case is that while it might be beneficial from the numerical point of view, it might not be easy to obtain an invariant discretization of this principle which does not lead to a fully coupled equation–grid system. In other words, it can happen that the grid equation includes values of u at both t^n and t^{n+1}. While this coupling is not a problem in the one-dimensional case, it can lead to a severe restriction of the applicability for multi-dimensional equations as solving the coupled equation–grid system might then be too expensive.

A G^1-invariant grid equation that circumvents the aforementioned coupling problem can be derived from the invariantization of x_i^{n+1}. This invariantization yields

$$\iota\big(x_i^{n+1}\big) = e^{\varepsilon_4}\left(x_i^{n+1} - x_i^n + \frac{2\Delta\tau}{h^+ + h^-}\big(\ln u_{i-1}^n - \ln u_{i+1}^n\big)\right),$$

where we did not explicitly substitute the frame value for ε_4. An appropriate grid is then given through $\iota(x_i^{n+1}) = 0$, or

$$M = x_i^{n+1} - x_i^n + \frac{2\Delta\tau}{h^+ + h^-}\big(\ln u_{i-1}^n - \ln u_{i+1}^n\big) = 0. \tag{12}$$

This grid equation is quite similar to the grid equation

$$x_i^{n+1} - x_i^n + \frac{2\Delta\tau}{h^+ + h^-}\left(\frac{h^+}{h^-}\ln\left(\frac{u_{i-1}^n}{u_i^n}\right) - \frac{h^-}{h^+}\ln\left(\frac{u_{i+1}^n}{u_i^n}\right)\right) = 0, \tag{13}$$

which was found in [1, 14] using the method of difference invariants. This last grid (13) is not only invariant under the subgroup G^1 but under the whole maximal Lie invariance group G of the heat equation. In the continuous limit, both equation (12) and (13) converge to

$$x_\tau = -\frac{2}{x_\xi}(\ln u)_\xi.$$

We have tested all our numerical schemes with both (12) and (13) and found that the resulting schemes give asymptotically the same numerical results. In fact, as in the evolution–projection strategy that will be introduced in Section 7 we have $h^+ = h^- = h$, equation (12) and (13) coincide.

Remark 1. While the invariantization algorithm guarantees that the scheme (11) is indeed invariant under the subgroup G^1 of the maximal Lie invariance group of the heat equation, the invariance can be checked in a straightforward fashion using the infinitesimal invariance criterion as invoked in the Dorodnitsyn method discussed in Section 2.1. Let us recall that this criterion states that an invariant I of a group action is annihilated by the associated infinitesimal generators, i.e. $v(I) = 0$ for all $v \in \mathfrak{g}$. Because in the present case, the invariants are defined on the stencil space with coordinates τ^n, $\Delta\tau$, x_i^n, x_{i+1}^n, x_{i-1}^n, x_i^{n+1}, u_i^n, u_{i+1}^n, u_{i-1}^n and u_i^{n+1}, we have to prolong the operators of $\mathfrak{g}$ accordingly. The prolongations of the first five operators of (3) to the variables of the stencil are

$$\partial_{\tau^n}, \qquad \partial_{x_i^n} + \partial_{x_{i+1}^n} + \partial_{x_{i-1}^n} + \partial_{x_i^{n+1}}, \qquad u_i^n\partial_{u_i^n} + u_{i+1}^n\partial_{u_{i+1}^n} + u_{i-1}^n\partial_{u_{i-1}^n} + u_i^{n+1}\partial_{u_i^{n+1}},$$

$$2\tau^n\partial_{\tau^n} + 2\Delta\tau\partial_{\Delta\tau} + x_i^n\partial_{x_i^n} + x_{i+1}^n\partial_{x_{i+1}^n} + x_{i-1}^n\partial_{x_{i-1}^n} + x_i^{n+1}\partial_{x_i^{n+1}},$$

$$2\tau^n(\partial_{x_i^n} + \partial_{x_{i+1}^n} + \partial_{x_{i-1}^n}) + 2(\tau^n + \Delta\tau)\partial_{x_i^{n+1}} - x_i^n u_i^n\partial_{u_i^n} - x_{i+1}^n u_{i+1}^n\partial_{u_{i+1}^n}$$
$$- x_{i-1}^n u_{i-1}^n\partial_{u_{i-1}^n} - x_i^{n+1}u_i^{n+1}\partial_{u_i^{n+1}},$$

see [13, 25] for more details. It can be checked that $\mathrm{pr}\,\mathrm{v}(S) = 0$ and $\mathrm{pr}\,\mathrm{v}(M) = 0$ hold on $S = 0$ and $M = 0$ for all the prolonged infinitesimal generators and thus $S = 0$ is a proper invariant numerical scheme and $M = 0$ an invariant grid equation.

Remark 2. The heat equation is a linear partial differential equation in two independent variables. One might thus consider to set up a grid equation not only for spatial but also for temporal adaptation. The reason why we refrain from spatial-temporal adaptation here is that the symmetry group G of the heat equation is compatible with flat time layers, i.e. $t_{i+1}^n - t_i^n = 0$ is a G-invariant equation. Improving the invariant numerical scheme constructed above using temporal adaptation would thus not allow one a fair comparison against the original non-invariant FTCS scheme for the heat equation. Moreover, flat time layers are well-agreed with the physics of the heat transfer problem, which affects all points of the domain simultaneously.

6 Numerical properties of the invariant scheme

In this section we investigate the numerical properties of the scheme (11) and related schemes. We start our consideration with the estimation of the local truncation error of the scheme. The study of this question is relevant because so far little is known about the relation between the order of a non-invariant scheme and its invariantized counterpart.

The discretization of the heat equation in computational coordinates (7) can be formally represented as

$$u_\tau^{\mathrm{d}} - x_\tau^{\mathrm{d}} \frac{u_\xi^{\mathrm{d}}}{x_\xi^{\mathrm{d}}} - \frac{1}{(x_\xi^{\mathrm{d}})^2} \left(u_{\xi\xi}^{\mathrm{d}} - \frac{x_{\xi\xi}^{\mathrm{d}}}{x_\xi^{\mathrm{d}}} u_\xi^{\mathrm{d}} \right) = 0, \tag{14}$$

where in the present case we assume that derivatives are approximated with the aid of a standard FTCS scheme. More general schemes will be considered after the order of the invariantized FTCS scheme is established.

Theorem 2. *The order of the invariant scheme* (11) *is the same as the order of the scheme* (14), *namely first order in time and second order in space, provided that an Euler forward step and second order centered differences are used to approximate the time and space derivatives arising in both the differential equation* (14) *and the normalization conditions* (8).

Proof. Invariantizing the scheme (14) using the normalization conditions (8) leads to

$$1 - \frac{4}{\iota\left(\left(x_\xi^{\mathrm{d}}\right)^2\right)} \iota\left(u_{\xi\xi}^{\mathrm{d}}\right) = 0, \tag{15}$$

where $\iota(f)(z)$ denotes the invariantization of the function $f(z)$. By definition, invariantization of a function $f(z)$ means to transform the argument z and plug in the moving frame for the group parameters. In the present case, the transformed form of (15) can be written as

$$1 - \frac{4e^{\varepsilon_3 - \varepsilon_5 x - \varepsilon_5^2 t - 2\varepsilon_4}}{\left(x_\xi^{\mathrm{d}}\right)^2} \left(e^{-\varepsilon_5 h^+} u_{i+1}^n + e^{\varepsilon_5 h^-} u_{i-1}^n - 2u_i^n \right) = 0.$$

Using the normalization condition $u_i^n = 1$ we obtain that

$$\tilde{u}_i^n = 1 = e^{\varepsilon_3 - \varepsilon_5 x - \varepsilon_5^2 t} u_i^n$$

and thus the last expression can be recast as

$$e^{2\varepsilon_4} u_i^n - \frac{4}{(h^+ + h^-)^2} \left(e^{-\varepsilon_5 h^+} u_{i+1}^n + e^{\varepsilon_5 h^-} u_{i-1}^n - 2u_i^n \right) = 0. \tag{16}$$

Let us now determine the local discretization error in the parameter ε_5. The respective moving frame component is

$$\varepsilon_5 = \frac{\ln u_{i+1}^n - \ln u_{i-1}^n}{h^+ + h^-},$$

which upon using (10) expands to

$$\varepsilon_5 = \frac{1}{x_\xi} \frac{u_\xi}{u_i^n} + O(\Delta\xi^2). \tag{17}$$

Substituting ε_5 into the second term of equation (16) and expanding the exponential functions in the same term into Taylor series, we obtain after some rearranging

$$e^{2\varepsilon_4} u_i^n - \frac{1}{x_\xi^2 \Delta\xi^2 + O(\Delta\xi^4)} \left(u_{i+1}^n + u_{i-1}^n - 2u_i^n - \varepsilon_5 \left(x_\xi \Delta\xi (u_{i+1}^n - u_{i-1}^n) \right. \right.$$
$$\left. \left. + \frac{1}{2} x_{\xi\xi} \Delta\xi^2 (u_{i+1}^n + u_{i-1}^n) \right) + \frac{1}{2} \varepsilon_2^2 x_\xi^2 \Delta\xi^2 (u_{i+1}^n + u_{i-1}^n) + O(\Delta\xi^4) \right) = 0.$$

This can be further simplified to

$$e^{2\varepsilon_4} u_i^n - \frac{1}{x_\xi^2} \left(u_{\xi\xi} - \frac{u_\xi^2}{u_i^n} - \frac{x_{\xi\xi}}{x_\xi} u_\xi \right) + O(\Delta\xi^2) = 0. \tag{18}$$

It now remains to expand the first term in equation (18). The moving frame component for ε_4 in (9) can be recast as

$$e^{2\varepsilon_4} u_i^n = \frac{\exp\left(-\Delta\tau \left(\frac{x_i^{n+1} - x_i^n}{\Delta\tau} \frac{\ln u_{i+1}^n - \ln u_{i-1}^n}{h^+ - h^-} + \left(\frac{\ln u_{i+1}^n - \ln u_{i-1}^n}{h^+ - h^-} \right)^2 \right) \right) u_i^{n+1} - u_i^n}{\Delta\tau}.$$

Using $x_i^{n+1} = x_i^n + x_\tau \Delta\tau + O(\Delta\tau^2)$ and $u_i^{n+1} = u_i^n + u_\tau \Delta\tau + O(\Delta\tau^2)$ and again expanding the exponential function into a Taylor series, we derive

$$e^{2\varepsilon_4} u_i^n = u_\tau - x_\tau \frac{u_\xi}{x_\xi} - \frac{1}{x_\xi^2} \frac{u_\xi^2}{u_i^n} + O(\Delta\tau, \Delta\xi^2).$$

Plugging this into equation (18) we arrive at

$$u_\tau - x_\tau \frac{u_\xi}{x_\xi} - \frac{1}{x_\xi^2} \left(u_{\xi\xi} - \frac{x_{\xi\xi}}{x_\xi} u_\xi \right) + O(\Delta\tau, \Delta\xi^2) = 0,$$

which completes the proof of the theorem. ∎

A more general statement is the following one:

Theorem 3. *The order of spatial discretization of an invariant finite difference scheme for the heat equation in computational variables equals the order $p \in \mathbb{N}$ of the spatial discretization of the associated non-invariant finite difference scheme provided that centered differences of order p are used to approximate both the derivatives in the heat equation and in the normalization conditions (8).*

Proof. In view of the general form (15) of the invariantization of scheme (14), we study the invariantization of the terms x_ξ and $u_{\xi\xi}$.

The invariantization of $(x_\xi^{\mathrm{d}})^2 = (x_\xi + O(\Delta\xi^p))^2$ is $\iota((x_\xi^{\mathrm{d}})^2) = e^{2\varepsilon_4}(x_\xi^2 + O(\Delta\xi^p))$ and is of the same order p if, as required, the moving frame component ε_4 stems from the approximation of $u_\tau^{\mathrm{d}} = 1$ using pth order accuracy and thus only includes approximations of derivatives with that accuracy.

Let us now investigate the invariantization of discretizations of $u_{\xi\xi}$. The general form of a centered difference approximation of $u_{\xi\xi}$ of even order p is

$$u_{\xi\xi} = \frac{1}{\Delta\xi^2} \sum_{j=-p/2}^{p/2} c_{p,j}^2 u_j^n + O(\Delta\xi^p),$$

where $c_{p,j}^2 = 2c_{p,j}^1/j$, $j \in A = \{-p/2, \ldots, -1, 1, \ldots, p/2\}$, $c_{p,0}^2 = -2\sum_{i=1}^{p/2} 1/i^2$ and

$$c_{p,j}^1 = \frac{(-1)^{j+1}(p/2)!^2}{j(p/2+j)!(p/2-j)!}, \qquad j \in A$$

and $c_{p,0}^1 = 0$ are the coefficients from the pth order approximation of u_ξ, i.e.

$$u_\xi = \frac{1}{\Delta\xi} \sum_{j=-p/2}^{p/2} c_{p,j}^1 u_j^n + O(\Delta\xi^p).$$

See [17] for a discussion of the algorithm for finding the weights $c_{p,j}^k$ in higher-order finite difference approximations of the kth derivative of u. The invariantization of $u_{\xi\xi}$ is

$$\iota(u_{\xi\xi}) = \frac{1}{\Delta\xi^2} \sum_{j=-p/2}^{p/2} c_{p,j}^2 \exp\left(\varepsilon_3 - \varepsilon_5 x_j^n - \varepsilon_5^2 \tau^n\right) u_j^n,$$

or, upon using the normalization condition $u_i^n = 1$,

$$\iota(u_{\xi\xi}) = \frac{1}{\Delta\xi^2} \frac{1}{u_i^n} \sum_{j=-p/2}^{p/2} c_{p,j}^2 \exp(-\varepsilon_5 \Delta x_j) u_j^n, \tag{19}$$

where we expand

$$\Delta x_j = x_j^n - x_i^n = \sum_{k=1}^{\infty} \frac{(j\Delta\xi)^k}{k!} \frac{\partial^k x}{\partial \xi^k}, \qquad u_i^n = \sum_{l=0}^{\infty} \frac{(j\Delta\xi)^l}{l!} \frac{\partial^l u}{\partial \xi^l}.$$

Using the expressions for Δx_j and u_i^n, the expression (19) can be expanded and rearranged in powers of $j\Delta\xi$ in the form

$$\iota(u_{\xi\xi}) = \frac{1}{\Delta\xi^2} \frac{1}{u_i^n} \sum_{j=-p/2}^{p/2} \sum_{k=0}^{\infty} (-1)^k c_{p,j}^2 A_k (j\Delta\xi)^k,$$

where

$$A_2 = \frac{1}{2}\left(u_{\xi\xi} - \varepsilon_5\left(2x_\xi u_\xi + x_{\xi\xi} u_i^n\right) + \varepsilon_5^2 x_\xi^2 u_i^n\right).$$

The expressions for A_k, $k \neq 2$, are not required subsequently. The proof is completed upon substituting for ε_5 the corresponding moving frame component (which is of order p if the normalization $u_x^{\mathrm{d}} = 0$ is approximated with pth order accuracy) and by noting that

$$\sum_{j=-p/2}^{p/2} c_{p,j}^2 j^k = \begin{cases} 0 & \text{for } k \in \{0,1,3,\ldots p+2, 2n\}, \quad n \in \mathbb{N}, \\ 2 & \text{for } k = 2, \\ c_k \neq 0 & \text{else,} \end{cases}.$$

where the precise values of the constants c_k follow from evaluating the respective sums.   ∎

The scheme (11) is only of first order in time $\tau = t$. To construct a scheme that is second order in time, we can start with a non-invariant scheme (14) and discretize the time derivative u_τ^{d} with second order accuracy, i.e. we set $u_\tau^{\mathrm{d}} = (u_i^{n+1} - u_i^{n-1})/(2\Delta\tau)$, where u_i^{n-1} is the value of u at the previous time step $\tau^{n-1} = \tau^n - \Delta\tau$. It is now necessary to check whether invariantizing this leapfrog discretization leads to an invariant scheme that is also second order in time.

Theorem 4. *Invariantization of the scheme* (14) *in which a leapfrog step and second order centered differences are used to approximate the time and space derivatives, leads to an invariant scheme that is both second order in time and space provided that the normalization conditions* (8) *are approximated using discretizations that are of second order.*

Proof. To prove this theorem it is sufficient to establish the order of the first term in equation (18). We proceed in an analog manner as in the proof of Theorem 2, i.e. we discretize the normalization condition $u_\tau^{\mathrm{d}} = 1$ but now with second order accuracy. This yields

$$\tilde{u}_{\tilde{\tau}}^{\mathrm{d}} = \frac{e^{\varepsilon_3 - \varepsilon_5 x_i^n - \varepsilon_5^2 \tau^n}}{2 e^{2\varepsilon_4}\Delta\tau}\left(e^{-\varepsilon_5 x_\tau \Delta\tau - \varepsilon_5^2 \Delta\tau} u_i^{n+1} - e^{\varepsilon_5 x_\tau \Delta\tau + \varepsilon_5^2 \Delta\tau} u_i^{n-1}\right) = 1.$$

Using the normalization condition $u_i^n = 1$ as before and expanding the exponential functions we derive

$$e^{2\varepsilon_4} u_i^n = \frac{1}{2\Delta\tau}\left(u_i^{n+1} - u_i^{n-1} - \Delta\tau\left(\varepsilon_5 x_\tau + \varepsilon_5^2\right)\left(u_i^{n+1} + u_i^{n-1}\right)\right) + O\left(\Delta\tau^2\right)$$

and upon noting that $u_i^{n+1} + u_i^{n-1} = 2u_i^n + O(\Delta\tau^2)$ we obtain

$$e^{2\varepsilon_4} u_i^n = u_\tau - x_\tau \frac{u_\xi}{x_\xi} - \frac{1}{x_\xi^2}\frac{u_\xi^2}{u_i^n} + O\left(\Delta\tau^2, \Delta\xi^2\right),$$

where we have substituted the expression (17) for ε_5. Plugging this result into equation (18) completes the proof of the theorem.   ∎

The actual form of the resulting invariant leapfrog scheme is

$$\frac{\exp\left(-\Delta\tau\left(\hat{x}_\tau^{\mathrm{d}}(\ln u)_x^{\mathrm{d}} + ((\ln u)_x^{\mathrm{d}})^2\right)\right) u_i^{n+1} - \exp\left(\Delta\tau\left(\check{x}_\tau^{\mathrm{d}}(\ln u)_x^{\mathrm{d}} + ((\ln u)_x^{\mathrm{d}})^2\right)\right) u_i^{n-1}}{2\Delta\tau}$$
$$- \frac{4\left(u_{i+1}^n\left(\frac{u_{i+1}^n}{u_{i-1}^n}\right)^{-h^+/(h^+ + h^-)} + u_{i-1}^n\left(\frac{u_{i+1}^n}{u_{i-1}^n}\right)^{h^-/(h^+ + h^-)} - 2u_i^n\right)}{(h^+ + h^-)^2} = 0,$$

where $\hat{x}_\tau^{\mathrm{d}} = (x_i^{n+1} - x_i^n)/\Delta\tau$ and $\check{x}_\tau^{\mathrm{d}} = (x_i^n - x_i^{n-1})/\Delta\tau$.

Higher order in time schemes can be constructed upon invariantizing multi-stage schemes. Combining this result with the result established in Theorem 3 we have found the following:

Corollary 1. *Invariantizing a non-invariant finite difference scheme for the heat equation in computational coordinates preserves the spatial and temporal order of the initial non-invariant finite difference scheme provided that centered differences are used and the normalization conditions for the moving frame are discretized with the same order as the respective derivatives in the non-invariant finite difference scheme.*

7 Invariant interpolation schemes

A common property of invariant numerical schemes for evolution equations possessing a non-trivial maximal Lie invariance group is that it is not possible to use a fixed, orthogonal discretization mesh. The continuous evolution of the mesh, if not handled properly, can lead to several undesirable properties, such as an overly strong concentration of grid points in certain regions and therefore too poor a resolution in other parts of the integration domain. The problem gets worse in the multi-dimensional case where mesh tangling or strongly skewed meshes can occur. But even if the mesh movement can be managed in an optimal way there are various physical problems for which continuously adapting grids pose a severe challenge. An example for this are practically all models that are in operational use in weather and climate prediction. These models employ sophisticated data assimilation strategies and are coupled to subgrid-scale parameterizations that aim to mimic the effects of unresolved processes on the grid scale variables. Attempting to make use of data assimilation or parameterization schemes on moving meshes is not only a technical problem that would cause a significant computational overhead compared to standard schemes but also a conceptual challenge for it is unclear on how to design parameterization schemes that can operate on grids with varying resolution. In order to promote the ideas of invariant numerical discretization schemes beyond their application to simple evolution equations it is thus instructive to study possible ways of overcoming the limitations imposed by the requirement of using moving meshes.

One straightforward idea is to use invariant schemes on fixed (i.e. non-invariant grids). As was shown in [30] this can lead to improved numerical solutions compared to non-invariant integrators, while still being excelled by the results that can be obtained using completely invariant schemes. On the other hand, if moving (invariant) meshes are not tractable for a particular class of problems, preserving the invariance of a system of differential equations at least for the discretization of the system itself might be a possible trade-off to take.

Another idea is to use an *evolution–projection strategy*, which will be proposed in the following. This concept relies on using the invariant scheme for the system of differential equations together with the invariant mesh equations for a single time step followed by the projection of the numerical solution back to the regular mesh. A similar strategy has proven successful in semi-Lagrangian time integration schemes [32].

In the present case, the projection step can be practically realized by using *interpolation*. Obviously, any standard interpolation scheme can be used to map the numerical solution u_i^{n+1} defined at x_i^{n+1} to the uniformly spaced ξ-grid. This, however, can break the invariance of the numerical scheme as a whole and so the question arises whether it is possible to accomplish the interpolation step in a symmetry-preserving fashion.

In the following we discuss two possible ways of formulating *invariant interpolation schemes*, both of which can be used for finding interpolations that allow the re-mapping of the numerical solution on a moving mesh to a fixed, Cartesian, equally-spaced grid. These ways are the *invariantization of non-invariant interpolation schemes* with moving frames and the construction of *interpolations using difference invariants*.

Invariantization of interpolation schemes. The moving frame constructed in the course of invariantizing a finite difference scheme can also be used to invariantize a certain interpolation

method. We exemplify this idea by invariantizing the formula for linear interpolation,

$$u_i^{n+1}(y) = u_i^{n+1} + \left(y - x_i^{n+1}\right)\frac{u_{i+1}^{n+1} - u_i^{n+1}}{x_{i+1}^{n+1} - x_i^{n+1}},$$

where $y \in [x_i^{n+1}, x_{i+1}^{n+1}]$. The invariantization of this expression using a moving frame associated with G^1 yields the invariant interpolation formula

$$u_i^{n+1}(y) = U_i^{n+1} + \left(y - x_i^{n+1}\right)\frac{U_{i+1}^{n+1} - U_i^{n+1}}{x_{i+1}^{n+1} - x_i^{n+1}},$$
$$U_i^{n+1} = \exp\left((\ln \hat{u}_{\hat{x}})^{\mathrm{d}}\left(y - x_i^{n+1}\right)\right)u_i^{n+1}. \tag{20}$$

Note that we have used a slightly different moving frame for the invarianization as we have used for invariantizing the finite difference scheme for the heat equation. Specifically, this moving frame is constructed by replacing the normalization condition $u_x^{\mathrm{d}} = 0$ with $\hat{u}_{\hat{x}}^{\mathrm{d}} = 0$. The reason for this is that the moving frame used earlier yielded $\varepsilon_5 = (\ln u_x)^{\mathrm{d}}$, i.e. it involves the solutions of u_i at the time step τ^n rather than at $\tau^n + \Delta\tau$. Irrespectively of what normalization is used, both interpolations are invariant. Setting $y = \xi_i$ in the above interpolation formula yields u_i^{n+1} on the regular computational grid. Note that the interpolation (20) is consistent in that $u_i^{n+1}(x_i^{n+1}) = u_i^{n+1}$ and $u_i^{n+1}(x_{i+1}^{n+1}) = u_{i+1}^{n+1}$.

In a similar manner more sophisticated interpolation schemes can be invariantized. In the following, we will use the invariantization of quadratic interpolation. Usual quadratic interpolation is based on the expression

$$u^{n+1}(y) = u_{i-1}^{n+1}L_{i-1}(y) + u_i^{n+1}L_i(y) + u_{i+1}^{n+1}L_{i+1}(y), \qquad L_j(y) = \prod_{\substack{k=i-1 \\ k \neq j}}^{i+1} \frac{y - x_k^{n+1}}{x_j^{n+1} - x_k^{n+1}}, \tag{21}$$

where $y \in [x_{i-1}^{n+1}, x_{i+1}^{n+1}]$ and $L_j(y)$ are the Lagrangian interpolation polynomials. Invariantizing this formula using the same moving frame as above we get

$$u^{n+1}(y) = U_{i-1}^{n+1}L_{i-1}(y) + U_i^{n+1}L_i(y) + U_{i+1}^{n+1}L_{i+1}(y), \tag{22}$$

where $U_i = \exp((\ln \hat{u}_{\hat{x}})^{\mathrm{d}}(\hat{x} - x_i^{n+1}))u_i^{n+1}$, as in the case of the invariant linear interpolation (20). Numerical examples using the invariant quadratic interpolation will be given in Section 8.

Interpolation using difference invariants. The product frame on the grid point space allows invariantizing the elementary variables x_i^n and u_i^n, which yields the system of joint invariants. In the continuous limit these invariants take the normalization values chosen for x and u to construct the usual moving frame ρ [27]. On the other hand, on the discrete space $M_\xi^{\diamond n}$ we only normalize one x_i^n (i, n fixed) among all the grid points x_l^k and the analog statement is true for the associated values u_l^k. This means that the joint invariants $\iota(x_l^k)$ and $\iota(u_l^k)$, $l \neq i, k \neq n$, are nontrivial and can be used to assemble invariant interpolation schemes.

In the present case, while we have normalized $u_i^n = 1$ in the course of constructing the moving frame $\rho^{\diamond 4}$, we are free to use the moving frame to invariantized any u_l^k where $l \neq i, k \neq n$ and this will yield a proper (nontrivial) invariant on the discrete space $M_\xi^{\diamond 4}$. As above, we again recompute the moving frame for G^1 by replacing the normalization conditions $u_i^n = 1$ and $u_x^{\mathrm{d}} = 0$ with $u_i^{n+1} = 0$ and $\hat{u}_{\hat{x}}^{\mathrm{d}} = 0$, respectively, which yields new expressions for the moving frame components of ε_3 and ε_5 given by

$$\varepsilon_3 = -\left(\left(\ln u_i^{n+1} - x_i^{n+1}(\ln \hat{u}_{\hat{x}})^{\mathrm{d}} - \tau^{n+1}\left((\ln u_{\hat{x}})^{\mathrm{d}}\right)\right)^2\right), \qquad \varepsilon_5 = (\ln \hat{u}_{\hat{x}})^{\mathrm{d}}.$$

Using this modified moving frame, we then invarianize the variable $u^{n+1}(\xi_i)$, which is the sought value of u at the point (τ^{n+1}, ξ_i) of the computational domain. This invariantization yields

$$\iota\big(u^{n+1}(\xi_i)\big) = \frac{u^{n+1}(\xi_i)}{u_i^{n+1}} \exp\big((\ln \hat{u}_{\hat{x}})^{\mathrm{d}}\big(x_i^{n+1} - \xi_i\big)\big).$$

Because in the continuous limit the invariantization of $u^{n+1}(\xi_i)$ must reproduce the normalization condition $u = 1$, we restrict the difference invariant to the manifold $\iota(u^{n+1}(\xi_i)) = 1$. The invariant interpolation is thus

$$u^{n+1}(\xi_i) = \exp\big((\ln \hat{u}_{\hat{x}})^{\mathrm{d}}\big(\xi_i - x_i^{n+1}\big)\big)u_i^{n+1}$$

and it is again consistent as $u^{n+1}(x_i^{n+1}) = u_i^{n+1}$. More accurate interpolations could be constructed by combining the invariants $\iota(x_l^k)$ and $\iota(u_l^k)$ in a suitable way.

The advantage of the interpolation methods introduced in this section is that they are invariant under the group G^1, i.e. using these interpolation formulas to map u_i^{n+1} back to the ξ-grid does not break the invariance of the numerical schemes for the heat equation, while still allowing one to use a regular grid. Invariant interpolations thus allow avoiding the complications that moving meshes impose on the applicability of symmetry-preserving finite difference discretizations.

8 Numerical verification

In order to verify the accuracy predicted above for the various schemes proposed, we set-up the following problem. On a periodic domain $x \in [0, 2\pi[$, consider

$$u_t = u_{xx},$$
$$u(x, t = 0) = \sin(x - 1) + 2.$$

On a sequence of grids with $N \in \{2, 4, 8, \ldots, 256\}$, the number of grid points, we compute the error in the maximum norm between the numerical solution and the exact solution at $t = 1$. The time step $\Delta\tau$ is taken as proportional to h^2, $h = h^+ = h^-$, in all simulations.

In each of the following figures, we plot the reference line corresponding to $\mathcal{O}(h^2)$ dash–dotted, and the L_∞ error as black line where,

$$\|E\|_{L_\infty} = \max_{x \in [0, 2\pi]} |u(x, 1) - u_{\mathrm{exact}}t(x, 1)|.$$

Note that for all approaches described below we expect a second order convergence since the numerical scheme being used is of second order, its invariantization was shown to preserve this order and the quadratic interpolation and its invariantization is also of second order.

8.1 Invariant scheme without projection

In this test run we use the scheme (11) without projection. As a result, the solution is evolving along the trajectories of the grid equation (12). Since the spacing between trajectories is not constant, i.e. h^+ and h^- are changing in time, we choose to plot the error versus $1/N$. In Fig. 2, we observe the second order convergence expected.

8.2 Invariant scheme with non-invariant quadratic interpolation

In this scheme we interpolate the solution of the invariant scheme (11) at every step back onto the regular grid using standard Lagrange quadratic interpolation (21). As a result, we have the solution on a regular grid with step-size $h = 1/N$. In Fig. 3, we observe the second order convergence expected.
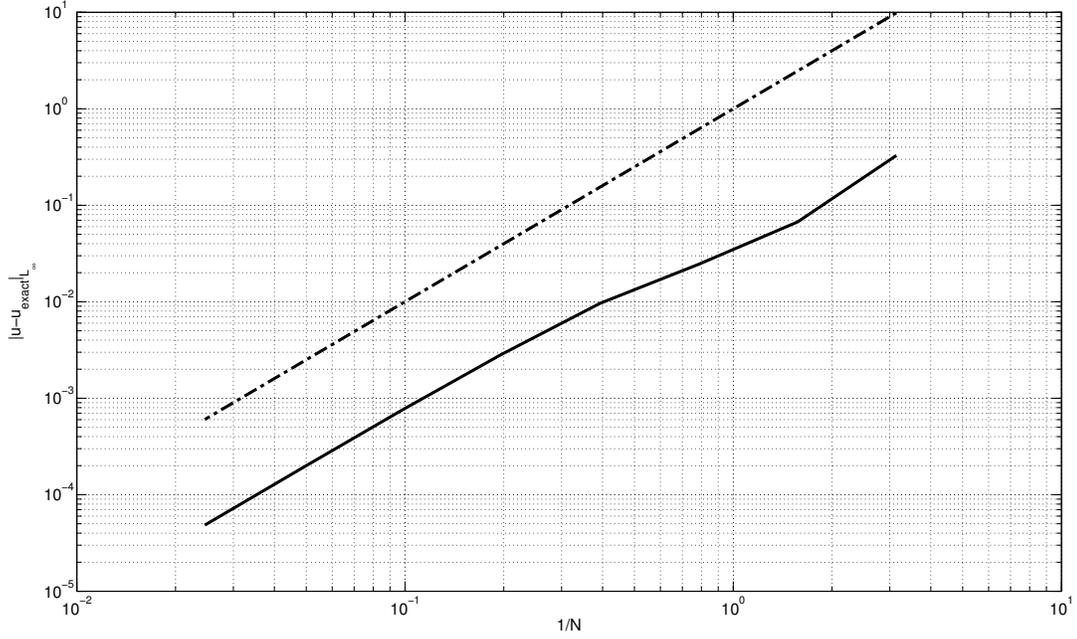
Figure 2. Convergence plot for the invariant scheme (11) with invariant grid equation (12) and without re-mapping.
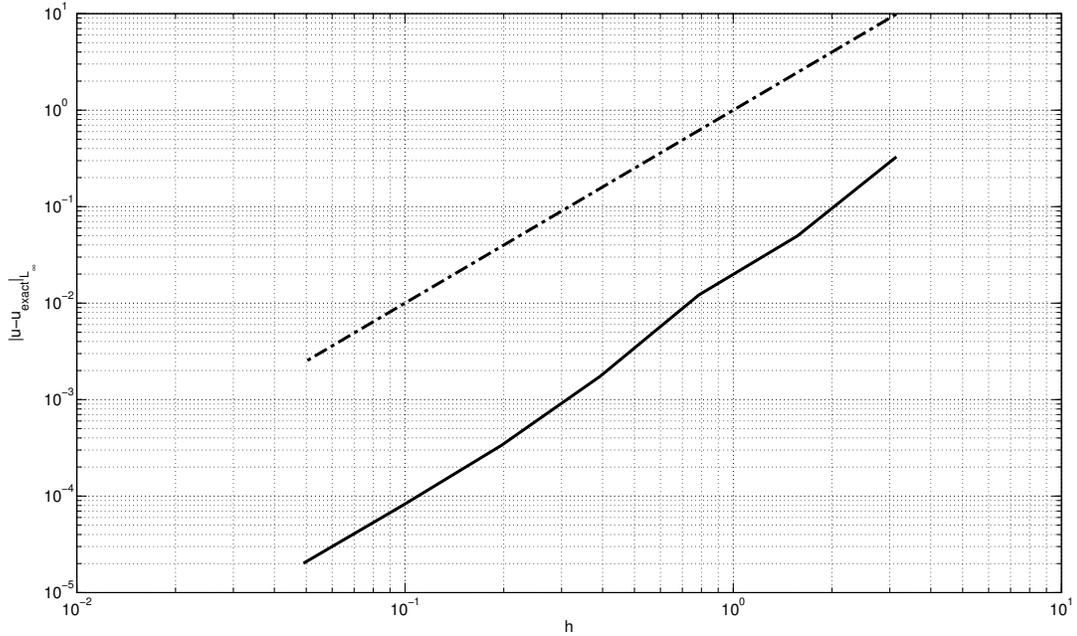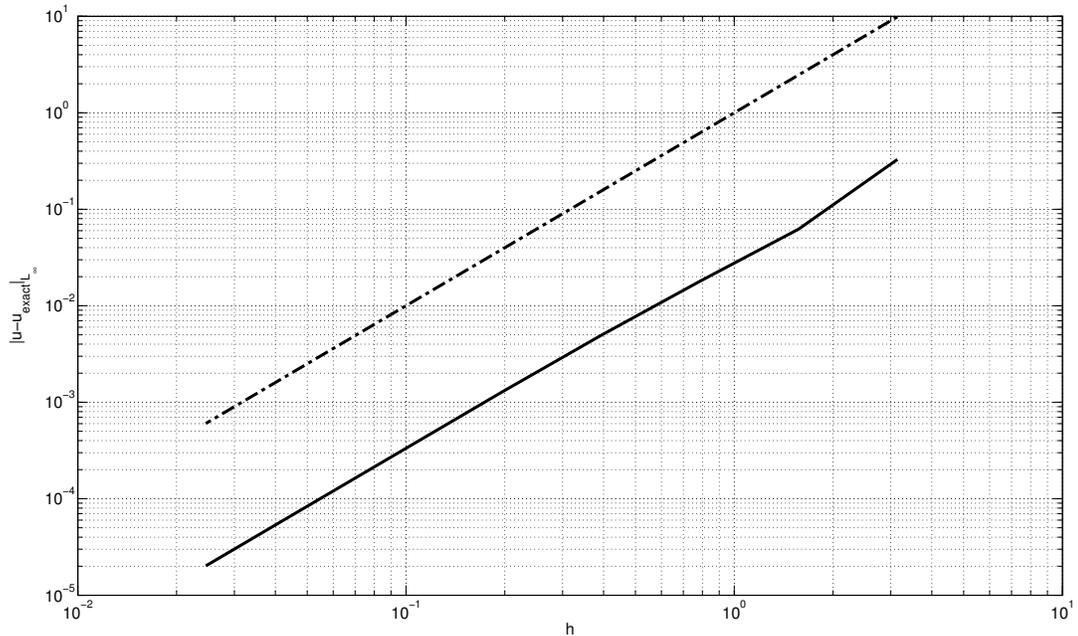

Figure 3. Convergence plot for the invariant scheme (11) with invariant grid equation (12) using non-invariant quadratic interpolation (21) as projection.

8.3   Invariant scheme with invariant quadratic interpolation

In this scheme, we interpolate the solution of the invariant scheme (11) at every step back onto the regular grid using the invariant Lagrange quadratic interpolation (22) described in the previous section. As for the case above, at each time step we have the solution on a regular grid with step-size $h = 1/N$. In Fig. 4, we observe the second order convergence expected.

Figure 4. Convergence plot for the invariant scheme (11) with invariant grid equation (12) using invariant quadratic interpolation (22) as projection.

8.4 Linearity preservation in the invariant numerical scheme

Linearity is not preserved by construction in the schemes proposed. For this reason it is instructive to check numerically whether or not linearity is preserved in the fully invariant scheme.

Consider the initial value problem,

$$u_t = u_{xx}$$
$$u(x, t = 0) = (\sin(x - 1) + 2) + (\cos x + 2),$$

the solution of which we call u_{exact}. We then solve numerically the following two equations

$$u_t^a = u_{xx}^a \text{ with } u^a(x, 0) = \sin(x - 1) + 2,$$
$$u_t^b = u_{xx}^b \text{ with } u^b(x, 0) = \cos x + 2,$$

and define $u_s = u^a + u^b$.

Fig. 5 depicts the L_∞ error between u_{exact} and u_s. We observe a convergence rate of second order. In other words, despite the fully invariant numerical scheme does not explicitly preserve the symmetry associated with the linear superposition principle, we observe that the linearity property is preserved approximately to the order of the method.

9   Conclusions

In this paper we construct invariant discretization schemes using the method of invariantization via equivariant moving frames. The advantage of this technique is that it allows one to start with a given non-invariant scheme and convert this initial scheme into a finite difference approximation of a system of differential equations $\mathcal{L}$ that is invariant under the same maximal Lie invariance group G (or a suitably chosen subgroup of G) as admitted by $\mathcal{L}$.

The possibility of converting non-invariant numerical schemes into invariant discretizations may lead to the overly optimistic speculation that the schemes constructed by invariantization
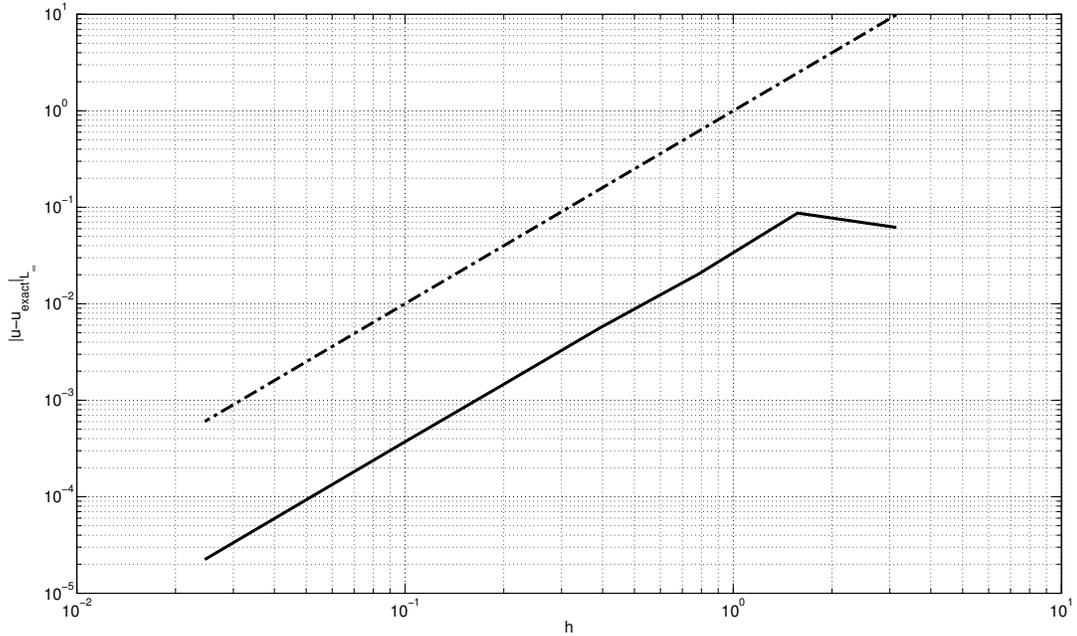
Figure 5. Convergence plot for the linearity test using the invariant scheme (11) with invariant grid equation (12) using invariant quadratic interpolation (22) as projection.

could be easily included in existing numerical models using the original scheme. The hurdle preventing this in practice is that preserving symmetry groups of systems of evolution equations more complicated than scalings or translations requires the use of moving grids. Converting numerical models that use standard discretization schemes based on fixed lattices to (invariant) discretization schemes on moving meshes is not an easy task. At the same time, rewriting numerical models from scratch for the simulation of involved physical processes using symmetry-preserving schemes might be a time-consuming and costly task too and it not certain that this is feasible at all. Moreover, it is as of now unclear whether preserving symmetries in numerical schemes for multi-dimensional systems of partial differential equations gives enough added value compared to standard schemes that one might justify such an undertaking in practice.

This is why one relies on finding methods allowing one to efficiently include invariant discretization schemes into existing numerical models without the need to rewrite new models from scratch that incorporate the invariance methodology. The method proposed in this article solves this problem by breaking the integration procedure into two steps, the time-stepping using the invariant numerical scheme with an invariant numerical grid equation and the projection (interpolation) of the results obtained at intermediate grid points to the regular mesh. This interpolation can be done in an invariant way by applying the moving frame map used to invariantize the initial discretization scheme also to a particular interpolation method. An alternative is to assemble the invariant interpolation method using joint invariants. Either way, it is worthwhile pointing out that interpolations requiring only data given at a single time level are already invariant under most symmetry groups as admitted by physical systems of differential equations. Thus, invariantization of interpolation formulas will often only lead to minor modifications of the initial interpolation method chosen and the influence on the numerical solution might be rather small. In the numerical tests carried out above for the heat equation, the difference in the convergence properties we found when using invariant or non-invariant interpolation methods is indeed small although using the invariant interpolation gave slightly better numerical results. This is encouraging and the reason why we plan to further investigate invariant numerical schemes using the projection procedure.

We illustrate the evolution–projection strategy by integrating the one-dimensional linear heat equation with an invariant numerical scheme. The heat equation has been studied quite extensively in light of its invariance properties and in particular it is a standard model for the construction of invariant numerical schemes [1, 14, 35]. At the same time, a comprehensive numerical analysis of such schemes was not given before and thus seems relevant to be reported. This is another aim of the present paper. Again, the analysis of numerical properties of discretization schemes is considerably easier if one can use non-evolving meshes.

Further work we intend to do is to employ the evolution–projection strategy to multi-dimensional systems of differential equations using both higher-order discretization and interpolation schemes.

Acknowledgements

References

[1] Bakirova M.I., Dorodnitsyn V.A., Kozlov R.V., Symmetry-preserving difference schemes for some heat transfer equations, *J. Phys. A: Math. Gen.* **30** (1997), 8139–8155, math.NA/0402367.

[2] Bihlo A., Popovych R.O., Invariant discretization schemes for the shallow-water equations, *SIAM J. Sci. Comput.* **34** (2012), B810–B839, arXiv:1201.0498.

[3] Bluman G.W., Kumei S., Symmetries and differential equations, *Applied Mathematical Sciences*, Vol. 81, Springer-Verlag, New York, 1989.

[4] Bridges T.J., Reich S., Numerical methods for Hamiltonian PDEs, *J. Phys. A: Math. Gen.* **39** (2006), 5287–5320.

[5] Budd C., Dorodnitsyn V., Symmetry-adapted moving mesh schemes for the nonlinear Schrödinger equation, *J. Phys. A: Math. Gen.* **34** (2001), 10387–10400.

[6] Budd C.J., Huang W., Russell R.D., Moving mesh methods for problems with blow-up, *SIAM J. Sci. Comput.* **17** (1996), 305–327.

[7] Budd C.J., Huang W., Russell R.D., Adaptivity with moving grids, *Acta Numer.* **18** (2009), 111–241.

[8] Budd C.J., Iserles A., Geometric integration: numerical solution of differential equations on manifolds, *R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci.* **357** (1999), 945–956.

[9] Cheh J., Olver P.J., Pohjanpelto J., Algorithms for differential invariants of symmetry groups of differential equations, *Found. Comput. Math.* **8** (2008), 501–532.

[10] Chhay M., Hamdouni A., A new construction for invariant numerical schemes using moving frames, *C. R. Mécanique* **338** (2010), 97–101.

[11] Chhay M., Hoarau E., Hamdouni A., Sagaut P., Comparison of some Lie-symmetry-based integrators, *J. Comput. Phys.* **230** (2011), 2174–2188.

[12] Dawes A.S., Invariant numerical methods, *Internat. J. Numer. Methods Fluids* **56** (2008), 1185–1191.

[13] Dorodnitsyn V., Applications of Lie groups to difference equations, *Differential and Integral Equations and Their Applications*, Vol. 8, CRC Press, Boca Raton, FL, 2011.

[14] Dorodnitsyn V., Kozlov R., A heat transfer with a source: the complete set of invariant difference schemes, *J. Nonlinear Math. Phys.* **10** (2003), 16–50, math.AP/0309139.

[15] Fels M., Olver P.J., Moving coframes. I. A practical algorithm, *Acta Appl. Math.* **51** (1998), 161–213.

[16] Fels M., Olver P.J., Moving coframes. II. Regularization and theoretical foundations, *Acta Appl. Math.* **55** (1999), 127–208.

[17] Fornberg B., A practical guide to pseudospectral methods, *Cambridge Monographs on Applied and Computational Mathematics*, Vol. 1, Cambridge University Press, Cambridge, 1996.

[18] Frank J., Gottwald G., Reich S., A Hamiltonian particle-mesh method for the rotating shallow-water equations, in Meshfree Methods for Partial Differential Equations (Bonn, 2001), *Lect. Notes Comput. Sci. Eng.*, Vol. 26, Editors M. Griebel, M.A. Schweitzer, T.J. Barth, M. Griebel, D.E. Keyes, R.M. Nieminen, D. Roose, T. Schlick, Springer, Berlin, 2003, 131–142.

[19] Hairer E., Lubich C., Wanner G., Geometric numerical integration: structure-preserving algorithms for ordinary differential equations, *Springer Series in Computational Mathematics*, Vol. 31, 2nd ed., Springer-Verlag, Berlin, 2006.

[20] Huang W., Russell R.D., Adaptive moving mesh methods, *Applied Mathematical Sciences*, Vol. 174, Springer, New York, 2011.

[21] Kim P., Invariantization of numerical schemes using moving frames, *BIT* **47** (2007), 525–546.

[22] Kim P., Invariantization of the Crank–Nicolson method for Burgers' equation, *Phys. D* **237** (2008), 243–254.

[23] Kim P., Olver P.J., Geometric integration via multi-space, *Regul. Chaotic Dyn.* **9** (2004), 213–226.

[24] Leimkuhler B., Reich S., Simulating Hamiltonian dynamics, *Cambridge Monographs on Applied and Computational Mathematics*, Vol. 14, Cambridge University Press, Cambridge, 2004.

[25] Levi D., Winternitz P., Continuous symmetries of difference equations, *J. Phys. A: Math. Gen.* **39** (2006), R1–R63, nlin.SI/0502004.

[26] Olver P.J., Applications of Lie groups to differential equations, *Graduate Texts in Mathematics*, Vol. 107, 2nd ed., Springer-Verlag, New York, 1993.

[27] Olver P.J., Geometric foundations of numerical algorithms and symmetry, *Appl. Algebra Engrg. Comm. Comput.* **11** (2001), 417–436.

[28] Olver P.J., Generating differential invariants, *J. Math. Anal. Appl.* **333** (2007), 450–471.

[29] Ovsiannikov L.V., Group analysis of differential equations, Academic Press Inc., New York, 1982.

[30] Rebelo R., Valiquette F., Symmetry preserving numerical schemes for partial differential equations and their numerical tests, *J. Difference Equ. Appl.* **19** (2013), 738–757, arXiv:1110.5921.

[31] Sommer M., Névir P., A conservative scheme for the shallow-water system on a staggered geodesic grid based on a Nambu representation, *Q. J. R. Meteorol. Soc.* **135** (2009), 485–494.

[32] Staniforth A., Côté J., Semi-Lagrangian integration schemes for atmospheric models – a review, *Mon. Weather Rev.* **119** (1991), 2206–2223.

[33] Stensrud D.J., Parameterization schemes: keys to understanding numerical weather prediction models, Cambridge University Press, Cambridge, 2007.

[34] Stull R.B., An introduction to boundary layer meteorology, *Atmospheric Sciences Library*, Vol. 13, Kluwer Academic Publishers, Dortrecht, 1988.

[35] Valiquette F., Winternitz P., Discretization of partial differential equations preserving their physical symmetries, *J. Phys. A: Math. Gen.* **38** (2005), 9765–9783, math-ph/0507061.