REPORTS
IN
INFORMATICS

ISSN 0333-3590

Application of numerical Lie group integrators to parabolic PDEs

Andrzej Susłowicz

REPORT NO 219 **December 2001**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL
http://www.ii.uib.no/publikasjoner/texrap/ps/2001-219.ps

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Application of numerical Lie group integrators to parabolic PDEs

Andrzej Susłowicz*

Department of Informatics
University of Bergen
N-5020 Bergen
Norway

17th December 2001

Abstract

This paper presents a way of solving a parabolic PDE with Runge-Kutta Methods on manifolds as described in papers [6], [1]. These methods are designed to solve ODEs on manifolds using Lie-group theory. An essential step of Runge-Kutta Methods on manifolds is a Lie algebra action which in this case corresponds to solving a simple parabolic PDE. Both Spectral Methods and Finite Difference Methods were tested to obtain approximations to Lie algebra actions. The goal was to develop methods which are more stable than classical Runge-Kutta Methods applied to a parabolic PDE.

AMS Subject Classification: 58J35.

Keywords: Geometric integration, differential equations on manifolds, CFL condition, parabolic PDE.

1 Introduction

In recent years much research has been devoted to finding numerical methods which preserve the same quantities as the analytic solution of ODEs, [1]. A parabolic PDE can be used to govern the variation of temperature in a bar. Finite Difference Methods for a parabolic PDE are discussed in [8] while Spectral Methods for a parabolic PDE are presented in [2]. Munthe-Kaas presented in [6], [1] Runge-Kutta Methods on manifolds which may be employed to solve many ODEs. The main feature of these methods is that the numerical solution evolves on the same manifold as the analytical solution. In [5] Lodden presents how to apply Runge-Kutta Methods on manifolds to (1) with periodic boundary conditions. Explicit schemes for a parabolic PDE are limited by the CFL condition. It is interesting to investigate if the Runge-Kutta Methods on manifolds applied to a parabolic PDE are more stable than the classical Runge-Kutta Methods. In this paper it is shown how the Runge-Kutta Methods on manifolds may be generalized to a parabolic PDE with variable diffusivity

$$\frac{\partial u}{\partial t} = \nabla \cdot (\mu(x)\nabla u) \tag{1}$$

*Email: Andrzej.Suslowicz@ii.uib.no, WWW: http://www.ii.uib.no/~mikolaj

together with the initial condition

$$u(x,0) = u_0(x)$$

and various boundary conditions. In each case the boundary conditions are compatible with the initial condition. The stability of the methods developed in this paper will be discussed.

2 The algorithm

The solution of a differential equation at any time is considered to be a point on a manifold, i.e.

$$u(x,t) \in \mathcal{M},\ t \geq 0.$$

In this case the manifold is $C_0^\infty[-1,1]$.

Definition 2.1 *Lie algebra is a vector space* $\mathfrak{g}$ *equipped with a bilinear skew-symmetric bracket*
$[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ *(the Lie bracket on* $\mathfrak{g}$*) satisfying the Jacobi identity:*

$$[a,[b,c]] + [b,[c,a]] + [c,[a,b]] = 0.$$

The solution is transformed from the manifold to a Lie algebra $\mathfrak{g}$ by

$$u(x,t) = exp(v)u_0,\ v \in \mathfrak{g}. \tag{2}$$

where exp is the solution corresponding to v with the initial condition u_0. The element v of Lie algebra corresponds to a simpler vector field. There is also an assumption for the equation. There must exist a Lie algebra $\mathfrak{g}$ with a Lie bracket [·,·], a left Lie algebra action $\lambda : \mathfrak{g} \times \mathcal{M} \to \mathcal{M}$ and a function $f : \mathbb{R} \times \mathcal{M} \to \mathfrak{g}$ such that the differential equation for $u(x,t) \in \mathcal{M}$ is

$$u_t = (\lambda_* f(t,u))(u), \qquad u(x,0) = u_0(x) \tag{3}$$

where

$$\lambda_* : \mathfrak{g} \to \mathfrak{X}(\mathcal{M})$$

is a homomorphism. The left algebra action

$$\lambda : \mathfrak{g} \times \mathcal{M} \to \mathcal{M},$$

assigns to the arguments $v \in \mathfrak{g}$ and $u_0(x) \in \mathcal{M}$ the solution of

$$u_t = (\lambda_* v)(u)$$

at time $t = 1$ with the initial condition $u_0(x)$ and some boundary conditions. There is freedom to choose the function f in (3). Combining the assumption (2) for the solution with the assumption (3) for the equation yields the corresponding differential equation in the Lie algebra

$$\frac{dv}{dt} = dexpinv(v, f(t, \lambda(v, y_0))),\ v(0) = 0.$$

Lie algebra is chosen as functions corresponding to simpler vector fields on $\mathcal{M}$ which are easy to exponentiate. The correspondence between the Lie algebra and the vector fields on $\mathcal{M}$ will be one to one.

Lemma 2.1 *Given two vector fields $G, \bar{G} \in \mathfrak{X}(\mathcal{M})$, the commutator is defined as*

$$[G, \bar{G}](y) = \frac{\partial}{\partial s}[G(y + s\bar{G}(y)) - \bar{G}(y + sG(y))]\bigg|_{s=0} \tag{4}$$

according to [1].

The bracket of Lie algebra is deduced from the commutator between two vector fields. The Lie algebra is a linear vector space where standard Runge-Kutta Methods can be employed. The details on the derivation of the algorithm may be found in [6]. The Runge-Kutta Methods on manifolds are as follows (p corresponds to $u_0(x)$)

$$\begin{aligned}
& y_0 = p \\
& \text{for } i = 1, 2, ..., s \\
& \quad u_i = \Delta t \sum_{j=1}^{s} a_{ij}\tilde{k_j}, \\
& \quad k_i = f(\Delta t c_i, \lambda(u_i, y_0)), \\
& \quad \tilde{k_i} = dexpinv(u_i, k_i, q),\ q = \text{the order of RK method}, \\
& \text{end} \\
& v = \Delta t \sum_{j=1}^{s} b_j \tilde{k_j}, \\
& y_1 = \lambda(v, y_0).
\end{aligned} \tag{5}$$

The numbers a_{ij}, b_i, c_i are the coefficients in a classic RK Method. A truncated approximation of the mapping $dexpinv : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ is applied in the algorithm. It is evaluated in terms of bracket,

$$\begin{aligned}
& dexpinv(u, v, q) = \sum_{k=0}^{q-1} \frac{B_k}{k!} ad_u^k(v), \\
& ad_u^0(v) = v, \\
& ad_u^1(v) = [u, v], \\
& ad_u^k(v) = ad_u(ad_u^{k-1}(v)).
\end{aligned} \tag{6}$$

The number q denotes the order of the underlying Runge Kutta Method and B_k is the kth Bernoulli number. The first few coefficients are

$$\frac{B_k}{k!} = \begin{cases} 0 & \text{for } k \text{ odd, and } k \neq 1, \\ 1, -\frac{1}{2}, \frac{1}{12}, -\frac{1}{720}, \frac{1}{30240}, -\frac{1}{1209600}, & \text{for } k = 0, 1, 2, 4, 6, 8. \end{cases}$$

Since RK Methods on manifolds are explicit, the first sum in the algorithm can end at $j = i - 1$ where i denotes the stage number. For the first stage, i.e. $i = 1$

$$u_1 = (0, 0),$$

yields

$$\begin{aligned} k_1 &= f(\Delta t c_1, \lambda(u_1, y_0)) = f(0, y_0), \\ \tilde{k_1} &= dexpinv(u_1, k_1, q) = dexpinv((0,0), k_1, q) = k_1. \end{aligned}$$

This means one computation of the left algebra action λ and $dexpinv$ may be avoided.

3 The Lie algebra action arising from Spectral Methods

3.1 The periodic boundary conditions

The solution of the simple PDE

$$\frac{\partial u}{\partial t} = \mu_c \frac{\partial^2 u}{\partial x^2} + g(x),\ \mu_c = \frac{1}{N} \sum_{j=0}^{N-1} \mu(x_j), \tag{7}$$

approximates locally the solution of (1) and it is chosen as the left algebra action λ. The approximate solution of (7) has the representation

$$u^N(x,t) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} a_k(t) e^{ikx},$$

and it fulfills (7) at the nodes

$$x_j = \frac{2\pi j}{N},\ j = 0, 1, 2, ..., N,$$

i.e.

$$\begin{aligned} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{da_k(t)}{dt} e^{\frac{i2\pi jk}{N}} = \mu_c \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} a_k(t)(-k^2) e^{\frac{i2\pi jk}{N}} + \\ + \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} c_n e^{\frac{i2\pi jk}{N}},\ j = 0, 1, ..., N-1. \end{aligned}$$

Due to periodicity of u^N, the same equation for a_k at x_0 and x_N is obtained. In the frequency space the system of ODEs is

$$\frac{da_k(t)}{dt} = -\mu_c k^2 a_k + c_k,\ k = -\frac{N}{2}, -\frac{N}{2}+1, ..., \frac{N}{2} - 1.$$

The solution of this system is exact

$$a_k(t) = \begin{cases} c_0 t + a_0(0) & \text{if } k = 0, \\ (a_k(0) - \frac{c_k}{\mu_c k^2})e^{-\mu_c k^2 t} + \frac{c_k}{\mu_c k^2} & \text{if } k \neq 0. \end{cases}$$

$a_k(0)$ are determined from

$$u^N(x_j, 0) = u_0(x_j) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} a_k(0) e^{\frac{i2\pi jk}{N}}.$$

Transformation to and from the the physical space is done with fast Fourier transform. It is explained in [2]. The vector field in (7) has the form

$$G(y) = \mu_c \frac{\partial^2 y}{\partial x^2} + g(x).$$

The Lie algebra is chosen as

$$\mathfrak{g} = \{(\mu_c, g(x))\}$$

and thus

$$\lambda_*(\mu_c, g(x))(u) = \mu_c \frac{\partial^2 u}{\partial x^2} + g(x).$$

Given two vector fields $G, \bar{G} \in \mathfrak{X}(\mathcal{M})$, i.e.

$$G(y) = \mu_c \frac{\partial^2 y}{\partial x^2} + g(x),\ \bar{G}(y) = \bar{\mu}_c \frac{\partial^2 y}{\partial x^2} + \bar{g}(x),$$

their commutator derived according to (4) is

$$[G, \bar{G}](y) \quad = \quad \frac{\partial^2(\mu_c \bar{g}(x) - \bar{\mu}_c g(x))}{\partial x^2}.$$

The bracket in the corresponding Lie algebra is

$$[(\mu_c, g(x)), (\bar{\mu}_c, \bar{g}(x))] = (0, \frac{\partial^2(\mu_c \bar{g}(x) - \bar{\mu}_c g(x))}{\partial x^2}).$$

It is easy to see from the way (7) is solved with discrete Fourier transform that the bracket may easily be computed in similar way

$$\left.\frac{\partial^2(\mu_c \bar{g}^N(x) - \bar{\mu}_c g^N(x))}{\partial x^2}\right|_{x=x_k} = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} (\mu_c \bar{c}_k - \bar{\mu}_c c_k)(-k^2) e^{\frac{i2\pi jk}{N}}.$$

The function f in (3) is chosen as

$$f(t, u) = (\mu_c, \nabla \cdot (\mu(x) \nabla u) - \mu_c \frac{\partial^2 u}{\partial x^2}).$$

It is obvious that

$$(\lambda_* f(t, u))(u) = \nabla \cdot (\mu(x) \nabla u).$$

3.2 The Dirichlet boundary conditions

The solution of the PDE

$$\frac{\partial u}{\partial t} = \mu_c \sqrt{1-x^2}\frac{\partial}{\partial x}(\sqrt{1-x^2}\frac{\partial u}{\partial x}) + g(x),\ \mu_c = \frac{1}{N+1}\sum_{j=0}^{N} \mu(x_j) \tag{8}$$

approximates locally the solution of (1) and it is chosen as the left algebra action λ. The vector field in (8) was chosen to yield a system of ODEs in the frequency space which can be integrated exactly. It will reduce the complexity of the algorithm. The approximate solution of (8) has the representation

$$u^N(x,t) = \sum_{k=0}^{N} a_k(t)T_k(x),\ T_k(x) = cos(k\, cos^{-1}(x)), \tag{9}$$

but first $x = cos(\theta)$ is introduced in (8) which yields

$$\frac{\partial u}{\partial t} = \mu_c \frac{\partial^2 u(cos(\theta), t)}{\partial \theta^2} + g(cos(\theta)). \tag{10}$$

In a collocation method u^N given by (9) fulfills (10) at the collocation points

$$x_j = cos(\frac{\pi j}{N}),\ j = 0, 1, ..., N,$$

which correspond to

$$\theta_j = \frac{\pi j}{N},\ j = 0, 1, ..., N,$$

i.e.

$$\sum_{k=0}^{N} \frac{da_k(t)}{dt} cos(\frac{\pi kj}{N}) = \mu_c \sum_{k=0}^{N} a_k(t)(-k^2) cos(\frac{\pi kj}{N}) + \\ + \sum_{k=0}^{N} c_k cos(\frac{\pi kj}{N}),\ j = 0, 1, ..., N.$$

The introduction of matrix notation yields

$$\mathbf{C}^{-1}\frac{d\underline{a}}{dt} = \mu_c \mathbf{C}^{-1} \cdot \mathbf{D} \cdot \underline{a} + \mathbf{C}^{-1} \cdot \underline{c},$$

where

$$\underline{a} = (a_0, a_1, ..., a_N)^t,\ \underline{c} = (c_0, c_1, ..., c_N)^t,\ [\mathbf{C}^{-1}]_{jk} = cos(\frac{\pi jk}{N}), \\ \mathbf{D} = diag(0, -1^2, -2^2, ..., -N^2).$$

According to [2]

$$[\mathbf{C}]_{kj} = \frac{2}{N\bar{c}_j\bar{c}_k} cos(\frac{\pi jk}{N}),$$

where

$$\bar{c}_j = \begin{cases} 2 & \text{if } j = 0 \text{ or } N, \\ 1 & \text{if } 1 \leq j \leq N-1. \end{cases}$$

In the frequency space the system of ODEs is

$$\frac{da_k(t)}{dt} = -\mu_c k^2 a_k + c_k, \; k = 0, 1, ..., N.$$

The solution of this system is exact

$$a_k(t) = \begin{cases} c_0 t + a_0(0) & \text{if } k = 0, \\ (a_k(0) - \frac{c_k}{\mu_c k^2})e^{-\mu_c k^2 t} + \frac{c_k}{\mu_c k^2} & \text{if } k = 1, 2, ..., N. \end{cases}$$

The initial condition for a_k is determined from

$$u^N(x_j, 0) = u_0(x_j) = \sum_{k=0}^{N} a_k(0) T_k(x_j) = \sum_{k=0}^{N} a_k(0) cos(\frac{\pi k j}{N}).$$

After transformation to the physical space the boundary values $u(-1, t) = 0$ and $u(1, t) = 0$ are assigned. Transformation to and from the physical space requires evaluation of a cosine transform. This is usually carried out by employing FFT, [2], [4]. For a small number of collocation points, cosine transform is faster computed as a matrix vector product. The vector field in (8) has the form

$$G(y) = \mu_c \sqrt{1-x^2} \frac{\partial}{\partial x} (\sqrt{1-x^2} \frac{\partial y}{\partial x}) + g(x).$$

The Lie algebra is chosen as

$$\mathfrak{g} = \{(\mu_c, g(x))\}$$

and thus

$$\lambda_*(\mu_c, g(x))(u) = \mu_c \sqrt{1-x^2} \frac{\partial}{\partial x} (\sqrt{1-x^2} \frac{\partial u}{\partial x}) + g(x).$$

Given two vector fields $G, \bar{G} \in \mathfrak{X}(\mathcal{M})$, i.e.

$$\begin{aligned} G(y) &= \mu_c \sqrt{1-x^2} \frac{\partial}{\partial x} (\sqrt{1-x^2} \frac{\partial y}{\partial x}) + g(x), \\ \bar{G}(y) &= \bar{\mu}_c \sqrt{1-x^2} \frac{\partial}{\partial x} (\sqrt{1-x^2} \frac{\partial y}{\partial x}) + \bar{g}(x), \end{aligned}$$

their commutator derived according to (4) is

$$[G, \bar{G}](y) \quad = \quad \sqrt{1-x^2} \frac{\partial}{\partial x} (\sqrt{1-x^2} \frac{\partial}{\partial x} (\mu_c \bar{g}(x) - \bar{\mu}_c g(x))).$$

The bracket in the corresponding Lie algebra is

$$[(\mu_c, g(x)), (\bar{\mu}_c, \bar{g}(x))] = (0, \sqrt{1-x^2} \frac{\partial}{\partial x} (\sqrt{1-x^2} \frac{\partial}{\partial x} (\mu_c \bar{g}(x) - \bar{\mu}_c g(x)))). \quad (11)$$

It is easy to see from the way (8) is solved with Chebyshev Collocation Method in the beginning of Section 3 that the bracket may easily be computed by

$$\sqrt{1-x^2}\frac{\partial}{\partial x}(\sqrt{1-x^2}\frac{\partial}{\partial x}(\mu_c\bar{g}^N(x)-\bar{\mu}_c g^N(x)))\Big|_{x=x_j} =$$
$$= \sum_{k=0}^{N}(-k^2)cos(k\theta)(\mu_c\bar{c}_k-\bar{\mu}_c c_k)\Big|_{\theta=\theta_j}.$$

The bracket is equal to zero at the boundaries and this value is assigned after transformation to the physical space. The function f in (3) is chosen as

$$f(t,u)=(\mu_c,\nabla\cdot(\mu(x)\nabla u)-\mu_c\sqrt{1-x^2}\frac{\partial}{\partial x}(\sqrt{1-x^2}\frac{\partial u}{\partial x})).$$

At the boundaries f is assigned the value zero. It is obvious that

$$(\lambda_* f(t,u))(u)=\nabla\cdot(\mu(x)\nabla u). \tag{12}$$

4 The Lie algebra action arising from Finite Difference discretization

4.1 The periodic boundary conditions

In this section the periodic boundary conditions will be considered, i.e.

$$u(-1,t)=u(1,t).$$

The original (1) may be discretized in space with Finite Difference. The nodes are chosen as

$$x_j=-1+\frac{(1-(-1))}{N}\cdot j=-1+\frac{2\cdot j}{N},\ j=0,1,...,N.$$

Additional nodes are introduced

$$x_{j+\frac{1}{2}}=\frac{1}{2}(x_j+x_{j+1})=-1+\frac{2\cdot j+1}{N},\ j=0,1,...,N-1.$$

The derivatives are approximated by central differences

$$\frac{\partial u}{\partial x}\Big|_{x=x_{i+\frac{1}{2}}}=\frac{u(x_{i+1},t)-u(x_i,t)}{\Delta x}.$$

This yields

$$\frac{\partial}{\partial x}(\mu(x)\frac{\partial u}{\partial x})\Big|_{x=x_i}=\frac{(\mu(x)\frac{\partial u}{\partial x})\big|_{x_{i+\frac{1}{2}}}-(\mu(x)\frac{\partial u}{\partial x})\big|_{x_{i-\frac{1}{2}}}}{\Delta x}=$$
$$=\frac{\mu(x_{i-\frac{1}{2}})u(x_{i-1},t)-(\mu(x_{i-\frac{1}{2}})+\mu(x_{i+\frac{1}{2}}))u(x_i,t)+\mu(x_{i+\frac{1}{2}})u(x_{i+1},t)}{(\Delta x)^2}. \tag{13}$$

Thinking of μ as constant suggests the solution of

$$\frac{\partial u}{\partial t} = \mu_c \frac{\partial^2 u}{\partial x^2} + g(x),\ \mu_c = \frac{1}{N}\sum_{j=0}^{N-1} \mu(x_{j+\frac{1}{2}}) \tag{14}$$

to solve (1). The solution of (14) defines the left algebra action λ. Employing central differences twice yields

$$\left.\frac{\partial u}{\partial t}\right|_{x=x_i} = \frac{\mu_c(u(x_{i-1},t) - 2u(x_i,t) + u(x_{i+1},t))}{(\Delta x)^2} + g(x_i). \tag{15}$$

For some points the periodicity of the function u must be employed to the formula (15)

$$\left.\frac{\partial u}{\partial t}\right|_{x=x_0} = \frac{\mu_c(u(x_{N-1},t) - 2u(x_0,t) + u(x_1,t))}{(\Delta x)^2} + g(x_0), \tag{16}$$

$$\left.\frac{\partial u}{\partial t}\right|_{x=x_{N-1}} = \frac{\mu_c(u(x_{N-2},t) - 2u(x_{N-1},t) + u(x_0,t))}{(\Delta x)^2} + g(x_{N-1}). \tag{17}$$

Introducing matrix notation

$$\underline{u} = \begin{pmatrix} u(x_0,t) \\ u(x_1,t) \\ \vdots \\ u(x_{N-1},t) \end{pmatrix},\ \mathbf{M} = \begin{pmatrix} -2 & 1 & & \dots & 1 \\ 1 & -2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & 1 \\ 1 & \dots & & 1 & -2 \end{pmatrix},$$

$$\underline{b} = \begin{pmatrix} g(x_0) \\ g(x_1) \\ \vdots \\ g(x_{N-1}) \end{pmatrix},$$

yields

$$\frac{d\underline{u}}{dt} = \frac{\mu_c}{(\Delta x)^2}\mathbf{M}\underline{u} + \underline{b},\ \underline{u}(0) = (u_0(x_0), u_0(x_1), ..., u_0(x_{N-1}))^t. \tag{18}$$

It is shown in [7] that the eigenvalues of $\mathbf{M}$ are

$$\lambda_r(\mathbf{M}) = -4sin^2(\frac{\pi r}{2n}),\ r = 1,2,...,n-1,\ \text{where } n = \frac{N}{2},$$

corresponding to the eigenvectors

$$\underline{y}_r = (1, cos(\frac{\pi r}{n}), cos(\frac{2\pi r}{n}), ..., cos(\frac{(2n-1)\pi r}{n}))^t,$$
$$\underline{z}_r = (0, sin(\frac{\pi r}{n}), sin(\frac{2\pi r}{n}), ..., sin(\frac{(2n-1)\pi r}{n}))^t.$$

and there are two eigenvalues of unit multiplicity given by

$$\lambda_0(\mathbf{M}) = 0,$$
$$\lambda_n(\mathbf{M}) = -4,$$

which correspond to the the eigenvectors

$$\begin{aligned}
\underline{y}_0 &= (1, 1, 1, ..., 1)^t, \\
\underline{y}_n &= (1, -1, 1, ..., -1)^t.
\end{aligned}$$

The vectors $\underline{y}_r, \underline{z}_r$ form a complete set of mutually orthogonal vectors. A matrix $\mathbf{P}$, which diagonalizes $\mathbf{M}$, is given by

$$\mathbf{P} = (\underline{y}_0 | \underline{y}_1 | ... | \underline{y}_n | \underline{z}_1 | \underline{z}_2 | ... | \underline{z}_{n-1}).$$

where

$$\mathbf{P}^{-1} = diag(\frac{1}{2n}, \frac{1}{n}, ..., \frac{1}{n}, \frac{1}{2n}, \frac{1}{n}, ..., \frac{1}{n})\mathbf{P}^t.$$

Introducing a vector $\underline{y}$ given by

$$\underline{u} = \mathbf{P}\underline{y},$$

in (18) yields an uncoupled system of ODEs

$$\frac{d\underline{y}}{dt} = \frac{\mu_c}{(\Delta x)^2} diag(\lambda_0, \lambda_1, ..., \lambda_n, \lambda_1, \lambda_2, ..., \lambda_{n-1})\underline{y} + \underline{c}, \ \underline{c} = \mathbf{P}^{-1}\underline{b},$$

with the initial condition

$$\underline{y}(0) = \mathbf{P}^{-1}\underline{u}(0).$$

It has the solution

$$\begin{aligned}
y_0(t) &= c_0 t + y_0(0), \\
y_i(t) &= (y_i(0) + \frac{(\Delta x)^2 c_i}{\mu_c \lambda_i}) e^{\frac{\mu_c \lambda_i}{(\Delta x)^2} t} - \frac{(\Delta x)^2 c_i}{\mu_c \lambda_i}, \ i = 1, 2, ..., n, \\
y_i(t) &= (y_i(0) + \frac{(\Delta x)^2 c_i}{\mu_c \lambda_{i-n}}) e^{\frac{\mu_c \lambda_{i-n}}{(\Delta x)^2} t} - \frac{(\Delta x)^2 c_i}{\mu_c \lambda_{i-n}}, \ i = n+1, n+2, ..., 2n-1.
\end{aligned}$$

Finally $\underline{u}$ is determined from $\underline{y}$. The vector field in (14) has the form

$$G(u) = \mu_c \frac{\partial^2 u}{\partial x^2} + g(x).$$

The Lie algebra is chosen as

$$\mathfrak{g} = \{(\mu_c, g(x))\}$$

and thus

$$\lambda_*(\mu_c, g(x))(u) = \mu_c \frac{\partial^2 u}{\partial x^2} + g(x).$$

Given two vector fields $G, \bar{G} \in \mathfrak{X}(\mathcal{M})$, i.e.

$$\begin{aligned} G(u) &= \mu_c \frac{\partial^2 u}{\partial x^2} + g(x), \\ \bar{G}(u) &= \bar{\mu}_c \frac{\partial^2 u}{\partial x^2} + \bar{g}(x). \end{aligned}$$

their commutator derived according to (4) is

$$[G, \bar{G}](u) = \frac{\partial^2 (\mu_c \bar{g}(x) - \bar{\mu}_c g(x))}{\partial x^2}.$$

The bracket in the corresponding Lie algebra is

$$[(\mu_c, g(x)), (\bar{\mu}_c, \bar{g}(x))] = (0, \frac{\partial^2 (\mu_c \bar{g}(x) - \bar{\mu}_c g(x))}{\partial x^2}).$$

The bracket may be computed employing (15), (16) and (17), this corresponds to multiplication with the matrix $\mathbf{M}$. The function f in (3) is chosen as

$$f(t, u) = (\mu_c, \nabla \cdot (\mu(x) \nabla u) - \mu_c \frac{\partial^2 u}{\partial x^2}).$$

The function f in (3) may be computed employing (13), (15), (16) and (17). For some points the periodicity of the function u must be employed to the formula (13). This yields at $x = x_0$

$$\frac{\mu(x_{N-\frac{1}{2}}) u(x_{N-1}, t) - (\mu(x_{N-\frac{1}{2}}) + \mu(x_{\frac{1}{2}})) u(x_0, t) + \mu(x_{\frac{1}{2}}) u(x_1, t)}{(\Delta x)^2},$$

and at $x = x_{N-1}$

$$\frac{\mu(x_{N-\frac{3}{2}}) \left. u \right|_{x=x_{N-2}} - (\mu(x_{N-\frac{3}{2}}) + \mu(x_{N-\frac{1}{2}})) \left. u \right|_{x=x_{N-1}} + \mu(x_{N-\frac{1}{2}}) \left. u \right|_{x=x_0}}{(\Delta x)^2}.$$

4.2 The Dirichlet boundary conditions

In this subsection the Dirichlet boundary conditions will be considered, i.e.

$$\begin{aligned} u(-1, t) &= 0, \\ u(1, t) &= 0. \end{aligned}$$

Introducing matrix notation

$$\underline{u} = \begin{pmatrix} u(x_1,t) \\ u(x_2,t) \\ \vdots \\ u(x_{N-1},t) \end{pmatrix}, \ \mathbf{M} = \begin{pmatrix} -2 & 1 & & \dots & 0 \\ 1 & -2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & 1 \\ \mathbf{0} & \dots & & 1 & -2 \end{pmatrix},$$

$$\underline{b} = \begin{pmatrix} g(x_1) \\ g(x_2) \\ \vdots \\ g(x_{N-1}) \end{pmatrix},$$

for (15) yields

$$\frac{d\underline{u}}{dt} = \frac{\mu_c}{(\Delta x)^2}\mathbf{M}\underline{u} + \underline{b}, \ \underline{u}(0) = (u_0(x_1), u_0(x_2), ..., u_0(x_{N-1}))^t.$$

It is shown in [7] that the eigenvalues of $\mathbf{M}$ are

$$\lambda_r(\mathbf{M}) = -4sin^2(\frac{\pi r}{2N}), \ r = 1,2,...,N-1,$$

and the corresponding eigenvectors

$$\underline{v}_r = (sin(\frac{\pi r}{N}), sin(\frac{2\pi r}{N}), ..., sin(\frac{(N-1)\pi r}{2N}))^t, \ r = 1,2,...,N-1.$$

The matrix $\mathbf{M}$ is symmetric and there exists an orthogonal matrix $\mathbf{P}$ which diagonalizes the matrix $\mathbf{M}$, i.e.

$$\mathbf{P}^t\mathbf{M}\mathbf{P} = \bar{\mathbf{D}} = diag(\lambda_1(\mathbf{M}), \lambda_2(\mathbf{M}), ..., \lambda_{N-1}(\mathbf{M})).$$

Matrix $\mathbf{P}$ is given by

$$[\mathbf{P}]_{ij} = \sqrt{\frac{2}{N}}sin(\frac{\pi ij}{N}).$$

It is easy to verify

$$[\mathbf{P}^{-1}]_{ij} = [\mathbf{P}^t]_{ij} = [\mathbf{P}]_{ij}.$$

(18) has the solution

$$\begin{aligned} \underline{u}(t) &= \mathbf{P} \cdot diag(e^{\frac{\mu_c\lambda_1}{(\Delta x)^2}t}, e^{\frac{\mu_c\lambda_2}{(\Delta x)^2}t}, ..., e^{\frac{\mu_c\lambda_{N-1}}{(\Delta x)^2}t}) \cdot \mathbf{P} \cdot (\underline{u}(0) - \underline{u}_p) + \underline{u}_p, \\ \underline{u}_p &= -\frac{(\Delta x)^2}{\mu_c} \cdot \mathbf{P} \cdot \bar{\mathbf{D}}^{-1} \cdot \mathbf{P} \cdot \underline{b}. \end{aligned}$$

Multiplication by $\mathbf{P}$ can be carried out by employing FFT. At the boundaries the values $u(-1,t) = 0$ and $u(1,t) = 0$ are known. The bracket is defined as zero at the boundaries. (15) shows how the bracket may be computed, which corresponds to multiplication with the matrix $\mathbf{M}$. (13) and (15) show how the second part of f may be computed.

4.3 The Neumann boundary conditions

In this subsection the Neumann boundary conditions will be considered, i.e.

$$\begin{aligned}\frac{\partial u}{\partial x}(-1,t) &= 0,\\ \frac{\partial u}{\partial x}(1,t) &= 0.\end{aligned}$$

The formula (15) is not well defined at the boundaries. The Neumann boundary conditions approximated with central differences yield

$$\begin{aligned}0 &= \frac{\partial u}{\partial x}(x_0,t) = \frac{u(x_1,t)-u(x_{-1},t)}{2\Delta x},\\ 0 &= \frac{\partial u}{\partial x}(x_N,t) = \frac{u(x_{N+1},t)-u(x_{N-1},t)}{2\Delta x}.\end{aligned}$$

Combining these formulas with (15) yields

$$\left.\frac{\partial u}{\partial t}\right|_{x=x_0} = \frac{\mu_c(-2u(x_0,t)+2u(x_1,t))}{(\Delta x)^2} + g(x_0), \tag{19}$$

$$\left.\frac{\partial u}{\partial t}\right|_{x=x_N} = \frac{\mu_c(2u(x_{N-1},t)-2u(x_N,t))}{(\Delta x)^2} + g(x_N). \tag{20}$$

Introducing matrix notation

$$\underline{u} = \begin{pmatrix} u(x_0,t)\\ u(x_1,t)\\ \vdots\\ u(x_N,t)\end{pmatrix},\ \bar{\mathbf{M}} = \begin{pmatrix} -2 & 2 & & \dots & 0\\ 1 & -2 & 1 & & \vdots\\ & \ddots & \ddots & \ddots & \\ \vdots & & 1 & -2 & 1\\ 0 & \dots & & 2 & -2\end{pmatrix},$$

$$\underline{b} = \begin{pmatrix} g(x_0)\\ g(x_1)\\ \vdots\\ g(x_N)\end{pmatrix},$$

for (15), (19) and (20) yields

$$\frac{d\underline{u}}{dt} = \frac{\mu_c}{(\Delta x)^2}\bar{\mathbf{M}}\underline{u} + \underline{b},\ \underline{u}(0) = (u_0(x_0), u_0(x_1), ..., u_0(x_N))^t. \tag{21}$$

It is shown in [7] that the eigenvalues of $\bar{\mathbf{M}}$ are

$$\lambda_r(\bar{\mathbf{M}}) = -4sin^2(\frac{\pi r}{2N}),\ r = 0,1,...,N,$$

and the corresponding eigenvectors

$$\underline{v}_r = (1, cos(\frac{\pi r}{N}), cos(\frac{2\pi r}{N}), ..., cos(\frac{(N-1)\pi r}{N}), (-1)^r)^t,\ r = 0,1,...,N,$$

are $\mathbf{D}$-conjugate with respect to the diagonal matrix $\mathbf{D} = diag(\frac{1}{2}, 1, ..., 1, \frac{1}{2})$. A matrix $\mathbf{P}$ which diagonalizes $\bar{\mathbf{M}}$ is given by

$$\mathbf{P} = (\underline{v}_0|\underline{v}_1|...|\underline{v}_N).$$

It is easy to verify

$$\mathbf{P}^{-1} = diag(\frac{1}{N}, \frac{2}{N}, ..., \frac{2}{N}, \frac{1}{N})\mathbf{P}^t\mathbf{D} = diag(\frac{1}{N}, \frac{2}{N}, ..., \frac{2}{N}, \frac{1}{N})\mathbf{P}\mathbf{D}.$$

since $\mathbf{P}$ is symmetric. Introducing a vector $\underline{y}$ given by

$$\underline{u} = \mathbf{P}\underline{y},$$

in (21) yields an uncoupled system of ODEs

$$\frac{d\underline{y}}{dt} = \frac{\mu_c}{(\Delta x)^2} diag(\lambda_0, \lambda_1, ..., \lambda_N)\underline{y} + \underline{c}, \ \underline{c} = \mathbf{P}^{-1}\underline{b},$$

with the initial condition

$$\underline{y}(0) = \mathbf{P}^{-1}\underline{u}(0).$$

This has the solution

$$\begin{aligned} y_0(t) &= c_0 t + y_0(0), \\ y_i(t) &= (y_i(0) + \frac{(\Delta x)^2 c_i}{\mu_c \lambda_i})e^{\frac{\mu_c \lambda_i}{(\Delta x)^2}t} - \frac{(\Delta x)^2 c_i}{\mu_c \lambda_i}, \ i = 1, 2, ..., N. \end{aligned}$$

Finally $\underline{u}$ is determined from $\underline{y}$. The bracket may be computed employing (15), (19) and (20), which corresponds to multiplication with the matrix $\bar{\mathbf{M}}$. The function f in (3) may be computed employing (13), (15), (19) and (20). At the boundaries (13) breaks down, but employing the central differences to approximate the Neumann boundary conditions (13) yields

$$\begin{aligned} \frac{\partial}{\partial x}(\mu(x)\frac{\partial u}{\partial x})\Big|_{x=x_0} &= \frac{-2\mu(x_{\frac{1}{2}})u(x_0,t) + 2\mu(x_{\frac{1}{2}})u(x_1,t)}{(\Delta x)^2}, \\ \frac{\partial}{\partial x}(\mu(x)\frac{\partial u}{\partial x})\Big|_{x=x_N} &= \frac{2\mu(x_{N-\frac{1}{2}})u(x_{N-1},t) - 2\mu(x_{N-\frac{1}{2}})u(x_N,t)}{(\Delta x)^2}. \end{aligned} \tag{22}$$

4.4 The Dirichlet-Neumann boundary conditions

In this subsection the Dirichlet-Neumann boundary conditions will be considered, i.e.

$$u(-1,t) = 0,$$
$$\frac{\partial u}{\partial x}(1,t) = 0.$$

Introducing matrix notation

$$\underline{u} = \begin{pmatrix} u(x_1,t) \\ u(x_2,t) \\ \vdots \\ u(x_N,t) \end{pmatrix},\ \mathbf{M} = \begin{pmatrix} -2 & 1 & & \dots & 0 \\ 1 & -2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & 1 \\ \mathbf{0} & \dots & & 2 & -2 \end{pmatrix},$$

$$\underline{b} = \begin{pmatrix} g(x_1) \\ g(x_2) \\ \vdots \\ g(x_N) \end{pmatrix},$$

for (15) and (20) yields

$$\frac{d\underline{u}}{dt} = \frac{\mu_c}{(\Delta x)^2}\mathbf{M}\underline{u} + \underline{b},\ \underline{u}(0) = (u_0(x_1), u_0(x_2), ..., u_0(x_N))^t.$$

It is shown in [7] that the eigenvalues of $\mathbf{M}$ are

$$\lambda_r(\mathbf{M}) = -4sin^2(\frac{(2r-1)\pi}{4N}),\ r = 1,2,...,N,$$

and the corresponding eigenvectors

$$\underline{v}_r =$$
$$= (sin(\frac{(2r-1)\pi}{2N}), sin(\frac{2(2r-1)\pi}{2N}), ..., sin(\frac{(N-1)(2r-1)\pi}{2N}), (-1)^{r+1})^t,$$
$$r = 1,2,...,N,$$

are $\mathbf{D}$-conjugate with respect to the diagonal matrix $\mathbf{D} = diag(1,1,...,1,\frac{1}{2})$. A matrix $\mathbf{P}$ which diagonalizes $\bar{\mathbf{M}}$ is given by

$$\mathbf{P} = (\underline{v}_1|\underline{v}_2|...|\underline{v}_N).$$

It is easy to verify

$$\mathbf{P}^{-1} = diag(\frac{2}{N}, \frac{2}{N}, ..., \frac{2}{N})\mathbf{P}^t\mathbf{D} = diag(\frac{2}{N}, \frac{2}{N}, ..., \frac{2}{N})\mathbf{P}^t\mathbf{D}.$$

Introducing a vector $\underline{y}$ given by

$$\underline{u} = \mathbf{P}\underline{y},$$

in (21) yields an uncoupled system of ODEs

$$\frac{d\underline{y}}{dt} = \frac{\mu_c}{(\Delta x)^2} diag(\lambda_0, \lambda_1, ..., \lambda_N)\underline{y} + \underline{c},\ \underline{c} = \mathbf{P}^{-1}\underline{b},$$

with the initial condition

$$\underline{y}(0) = \mathbf{P}^{-1}\underline{u}(0).$$

This has the solution

$$y_i(t) = (y_i(0) + \frac{(\Delta x)^2 c_i}{\mu_c \lambda_i}) e^{\frac{\mu_c \lambda_i}{(\Delta x)^2} t} - \frac{(\Delta x)^2 c_i}{\mu_c \lambda_i},\ i = 1, 2, ..., N.$$

Finally $\underline{u}$ is determined from $\underline{y}$. At $x = -1$ the value $u(-1, t) = 0$ is known. The bracket is defined as zero at $x = -1$. (15) and (20) show how the bracket may be computed, which correspond to multiplication with the matrix $\mathbf{M}$. (13), (15), (20) and (22) show how the second part of f may be computed. At $x = -1$, the second part of f is zero.

5 Stability of Euler Forward Method on manifolds

The Euler Forward Method on manifolds with the Lie algebra from Section 3.2 is in exact arithmetics

$$\begin{aligned}
y_0 &= u_0(x), \\
u_1 &= (0, 0), \\
k_1 &= (\mu_c, \nabla \cdot (\mu(x) \nabla u_0(x)) - \mu_c \sqrt{1 - x^2} \frac{\partial}{\partial x} (\sqrt{1 - x^2} \frac{\partial u_0(x)}{\partial x})), \\
\tilde{k}_1 &= (\mu_c, \nabla \cdot (\mu(x) \nabla u_0(x)) - \mu_c \sqrt{1 - x^2} \frac{\partial}{\partial x} (\sqrt{1 - x^2} \frac{\partial u_0(x)}{\partial x})), \\
v &= (\Delta t \mu_c, \Delta t \nabla \cdot (\mu(x) \nabla u_0(x)) - \Delta t \mu_c \sqrt{1 - x^2} \frac{\partial}{\partial x} (\sqrt{1 - x^2} \frac{\partial u_0(x)}{\partial x})), \\
y_1 &= \lambda(v, y_0).
\end{aligned}$$

The last expression means that one step of the Euler Forward Method on manifolds corresponds to solving the simpler equation

$$\frac{\partial u}{\partial t} = \bar{\mu}_c \sqrt{1 - x^2} \frac{\partial}{\partial x} (\sqrt{1 - x^2} \frac{\partial u(x)}{\partial x}) + g(x) \tag{23}$$

where

$$\bar{\mu}_c = \Delta t \mu_c$$

and

$$g(x) = \Delta t \nabla \cdot (\mu(x) \nabla u_0(x)) - \Delta t \mu_c \sqrt{1 - x^2} \frac{\partial}{\partial x} (\sqrt{1 - x^2} \frac{\partial u_0(x)}{\partial x})$$

at $t = 1$ with Dirichlet boundary conditions. It is explained in Section 3.2 how to solve (23) with the Chebyshev Collocation Method. In [2] it is shown that for the Chebyshev Collocation Method applied to a diffusion problem the limit on the time step Δt decreases like CN^{-4}. There is no guarantee that (23) will be solved unconditionally even if the corresponding system of ODEs in the frequency space is solved exactly.

It is easy to verify that one step of the Euler Forward Method on manifolds with each Lie algebra from Section 4 corresponds to solving the equation

$$\frac{\partial u}{\partial t} = \bar{\mu}_c \frac{\partial^2 u(x)}{\partial x^2} + g(x) \tag{24}$$

where

$$\bar{\mu}_c = \Delta t \mu_c \text{ and } g(x) = \Delta t \nabla \cdot (\mu(x) \nabla u_0(x)) - \Delta t \mu_c \frac{\partial^2 u_0(x)}{\partial x^2}$$

at $t = 1$ with some boundary conditions. This approach is expected to be very stable since exact integration in time is applied to semidiscretized (24).

6 The effect of $dexpinv$ on stability of the algorithm

The mapping $dexpinv(u, v, q)$ given by (6) is an identity mapping with respect to the second variable, v, when the order, q, of the underlying Runge Kutta Method is equal to one. For methods of order higher than one $dexpinv$ mapping will influence the stability of the algorithm.

Theorem 6.1 *For RK Methods of order $q > 1$ the mapping $dexpinv(u, v, q)$ limits the time step in terms of the number of collocation points for Spectral Methods in Section 3 in the following way*

$$\Delta t < \frac{2\pi}{\mu_c} N^{-2}.$$

A similar condition

$$\Delta t < \frac{\pi (\Delta x)^2}{2\mu_c}$$

is valid for Finite Difference Methods in Section 4.

$Proof.$
By considering the Lie algebra in Section 4 and the definition of $dexpinv(u, v, q)$ we obtain

$$\begin{aligned}
ad^0_{(\mu_c,g)}(\bar{\mu}_c, \bar{g}) &= (\bar{\mu}_c, \bar{g}), \\
ad^1_{(\mu_c,g)}(\bar{\mu}_c, \bar{g}) = [(\mu_c, g), (\bar{\mu}_c, \bar{g})] &= (0, \frac{\partial^2(\mu_c \bar{g} - \bar{\mu}_c g)}{\partial x^2}), \\
ad^k_{(\mu_c,g)}(\bar{\mu}_c, \bar{g}) &= (0, {\mu_c}^{k-1} \frac{\partial^{2k}(\mu_c \bar{g} - \bar{\mu}_c g)}{\partial x^{2k}}), \text{ k} = 2, 3, ...
\end{aligned}$$

This yields the formal expansion

$$\begin{aligned}
&dexpinv((\mu_c, g), (\bar{\mu}_c, \bar{g}), q) = \\
&= \frac{B_0}{0!}(\bar{\mu}_c, \bar{g}) + \frac{B_1}{1!}(0, \frac{\partial^2(\mu_c \bar{g} - \bar{\mu}_c g)}{\partial x^2}) + ... + \\
&+ \frac{B_{q-1}}{(q-1)!}(0, {\mu_c}^{q-2} \frac{\partial^{2q-2}(\mu_c \bar{g} - \bar{\mu}_c g)}{\partial x^{2q-2}}) = \\
&= (\frac{B_0}{0!}\bar{\mu}_c, \frac{B_0}{0!}\frac{\bar{\mu}_c}{\mu_c} g + p_{q-1}(\mu_c \frac{\partial^2}{\partial x^2})(\bar{g} - \frac{\bar{\mu}_c}{\mu_c} g)), \\
p_{q-1}(x) &= \frac{B_0}{0!} + \frac{B_1}{1!}x + ... + \frac{B_{q-1}}{(q-1)!}x^{q-1}, \ q > 1.
\end{aligned}$$

Finite Difference discretization of the bracket as in Section 4 yields

$$
\begin{aligned}
ad^0_{(\mu_c,\underline{b})}(\bar{\mu}_c,\underline{\bar{b}}) &= (\bar{\mu}_c,\underline{\bar{b}}),\\
ad^1_{(\mu_c,\underline{b})}(\bar{\mu}_c,\underline{\bar{b}}) = [(\mu_c,\underline{b}),(\bar{\mu}_c,\underline{\bar{b}})] &= (0, \frac{1}{(\Delta x)^2}\mathbf{M}(\mu_c\underline{\bar{b}} - \bar{\mu}_c\underline{b})),\\
ad^k_{(\mu_c,\underline{b})}(\bar{\mu}_c,\underline{\bar{b}}) &= (0, \frac{{\mu_c}^{k-1}}{((\Delta x)^2)^k}\mathbf{M}^k(\mu_c\underline{\bar{b}} - \bar{\mu}_c\underline{b})),\ \mathrm{k} = 2,3,...
\end{aligned}
$$

which results in

$$
\begin{aligned}
& dexpinv((\mu_c,\underline{b}),(\bar{\mu}_c,\underline{\bar{b}}),q) = \\
&= \frac{B_0}{0!}(\bar{\mu}_c,\underline{\bar{b}}) + \frac{B_1}{1!}(0, \frac{1}{(\Delta x)^2}\mathbf{M}(\mu_c\underline{\bar{b}} - \bar{\mu}_c\underline{b})) + ... + \qquad (25)\\
&+ \frac{B_{q-1}}{(q-1)!}(0, \frac{{\mu_c}^{q-2}}{((\Delta x)^2)^{q-1}}\mathbf{M}^{q-1}(\mu_c\underline{\bar{b}} - \bar{\mu}_c\underline{b})) = \\
&= (\frac{B_0}{0!}\bar{\mu}_c, \frac{B_0}{0!}\frac{\bar{\mu}_c}{\mu_c}\underline{b} + \mathbf{P}\cdot diag(p_{q-1}(\frac{\mu_c\lambda_1}{(\Delta x)^2}), p_{q-1}(\frac{\mu_c\lambda_2}{(\Delta x)^2}), ..., \qquad (26)\\
& p_{q-1}(\frac{\mu_c\lambda_{N-1}}{(\Delta x)^2}))\cdot \mathbf{P}^t(\underline{\bar{b}} - \frac{\bar{\mu}_c}{\mu_c}\underline{b})),\\
& p_{q-1}(x) = \frac{B_0}{0!} + \frac{B_1}{1!}x + ... + \frac{B_{q-1}}{(q-1)!}x^{q-1}.
\end{aligned}
$$

p_{q-1} converges to the scalar function $\frac{x}{e^x-1}$ for $|x| < 2\pi$. Combining this convergence radius with the observation that $(\mu_c,\underline{b})$ is multiplied by Δt in (5), the following condition is obtained for (26)

$$|\frac{\Delta t\mu_c\lambda_i(\mathbf{M})}{(\Delta x)^2}| < 2\pi. \qquad (27)$$

Since $|\lambda_i(\mathbf{M})| \leq 4$ this condition is equivalent to

$$\Delta t < \frac{\pi(\Delta x)^2}{2\mu_c}.$$

It is easy to derive a condition for time step for $dexpinv$ from Section 3. It is

$$|\Delta t\mu_c(-k^2)| < 2\pi,\ k = 0,1,...,N,$$

i.e.

$$\Delta t < \frac{2\pi}{\mu_c}N^{-2}.$$

The polynomial p_{q-1} can be replaced by the scalar function $\frac{x}{e^x-1}$ in (26) within the convergence radius. The diagonal matrix in (26) will then have elements of the form

$$\frac{x}{e^x-1}\Bigg|_{x=\frac{\Delta t\mu_c\lambda_i(\mathbf{M})}{(\Delta x)^2}} = \frac{\frac{\Delta t\mu_c\lambda_i(\mathbf{M})}{(\Delta x)^2}}{e^{\frac{\Delta t\mu_c\lambda_i(\mathbf{M})}{(\Delta x)^2}} - 1} \approx -\frac{\Delta t\mu_c\lambda_i(\mathbf{M})}{(\Delta x)^2},$$

$$\text{if } \frac{\Delta t\mu_c\lambda_i(\mathbf{M})}{(\Delta x)^2} \ll 0.$$

This expression approaches infinity as Δt increases. If the spatial discretization $(\Delta x)^2$ decreases the time step Δt must decrease. This is again a classical condition on explicit schemes for a parabolic equation. $\square$

The spectral decomposition of $dexpinv$ in (26) should only be used to derive theoretical results. $dexpinv$ can be computed in $O(N)$ flops if the definition of it with ad operators is employed. Computing of an ad operator requires only a matrix vector product with a sparse matrix.

7 Numerical experiments

The purpose of experiments is to find out which stability properties our methods have. The numerical solution is examined after 100 time steps with a varying number of collocation points. Two Runge-Kutta Methods were tested, the Euler Forward Method which is of order one and the RK4 Method which is of order four. The coefficients of the Euler Forward Method are specified through Table 1

$$\begin{array}{c|c} c_1 & a_{11} \\ \hline & b_1 \end{array} = \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

Table 1: The coefficients of the Euler Forward Method.

and the coefficients of the RK4 Method are specified through Table 2

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & a_{13} & a_{14} \\ c_2 & a_{21} & a_{22} & a_{23} & a_{24} \\ c_3 & a_{31} & a_{32} & a_{33} & a_{34} \\ c_4 & a_{41} & a_{42} & a_{43} & a_{44} \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array} = \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Table 2: The coefficients of the RK4 Method.

The results for the Fourier Method are shown in Table 3 and the results for the Chebyshev Collocation Method are gathered in Table 4. It was not a goal to determine exactly the largest time step which guarantees the stability of the algorithm for a given number of nodes. The time step was divided by ten until the algorithm was stable for a given number of nodes. It is easy to judge from the tables given below whether the stability limit decreases as N^{-2} or N^{-4}.

	Euler Forward		RK4 Method	
	On manifolds	Classical	On manifolds	Classical
N	Δt	stable Δt	stable Δt	stable Δt
8	unconditional	0.1	1.0	0.1
16	unconditional	0.01	0.1	0.01
32	unconditional	0.001	0.01	0.01
64	unconditional	0.001	0.01	0.001
128	unconditional	0.0001	0.001	0.0001
256	unconditional	0.0001	0.0001	0.0001

Table 3: The results for parabolic PDE with periodic boundary conditions. The Lie algebra action is based on Fourier Collocation Method. Number of nodes $N+1$ and stable time step for Euler Forward and a RK4 Method. The initial condition is $u_0(x) = exp(cos(x))$ and the diffusion is $\mu(x) = 1 - 0.1(\frac{x}{\pi} - 1)^2$.

	Euler Forward		RK4 Method	
	On manifolds	Classical	On manifolds	Classical
$N+1$	stable Δt	stable Δt	stable Δt	stable Δt
8	0.01	0.01	0.01	0.01
16	0.0001	0.0001	0.001	0.001
32	0.00001	0.00001	0.00001	0.00001
64	0.000001	0.000001	0.000001	0.000001
128	0.0000001	0.0000001	0.0000001	0.0000001
256	0.00000001	0.00000001	0.00000001	0.00000001

Table 4: The results for parabolic PDE with Dirichlet boundary conditions. The Lie algebra action is based on Chebyshev Collocation Method. Number of collocation points $N+1$ and stable time step for Euler Forward and a RK4 Method. The initial condition is $u_0(x) = e^{-10x^2} - e^{-10}$ and the diffusion is $\mu(x) = 1 - 0.1x^2$.

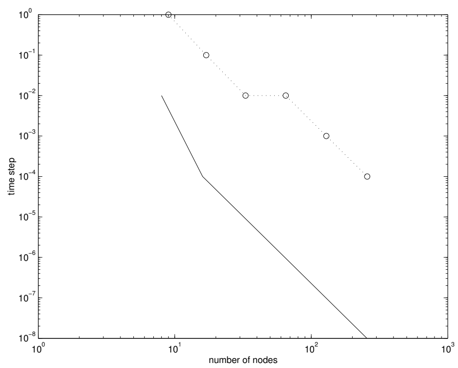

Figure 1: (o) Stable time step for the RK4 Method on manifolds based on Table 3. (-) Stable time step for Euler Forward on manifolds based on Table 4.

It is clear from Table 3 and Figure 1 that the time step is limited by the spatial discretization

$$\Delta t \le \frac{C_1}{N^2} \tag{28}$$

for the RK4 Method based on the Fourier Collocation Method. It is clear from Table 4 and

Figure 1 that the time step is limited by the spatial discretization

$$\Delta t \leq \frac{C_2}{N^4} \tag{29}$$

for the method based on the Chebyshev Collocation Method. In Section 5 it was shown that one step of the Euler Forward on manifolds corresponds to solving a simpler equation. The experiment confirms that a time step for the Chebyshev Collocation Method applied to (8) is limited according to (29). This affects the stability of the RK4 Method on manifolds. For this method (8) is also solved with Chebyshev Collocation Method. The condition on the time step Δt due to $dexpinv$ derived in Section 6 is weaker than condition (29). The instability is not caused by $dexpinv$. The Euler Forward Method yields global order 1, while the RK4 Method on manifolds yields global order only 1. One way of explaining the loss of order is that the PDE in Lie algebra is stiffer than the original equation. The Euler Forward Method on manifolds employing Finite Difference Method seems to be unconditionally stable with any boundary conditions. The method is based on determining the solution of a simpler problem and that is a local assumption. Even if the method seems to be unconditionally stable, it may not yield good approximation of the solution for arbitrary large time steps. The results for periodic boundary conditions are gathered in Table 5. The results for Dirichlet boundary conditions are gathered in Table 6. The results for Neumann boundary conditions are gathered in Table 7. The results for Dirichlet-Neumann boundary conditions are gathered in Table 8.

	Euler Forward		RK4 Method	
	On manifolds	Classical	On manifolds	Classical
N	Δt	stable Δt	stable Δt	stable Δt
8	unconditional	0.01	0.1	0.01
16	unconditional	0.001	0.01	0.01
32	unconditional	0.001	0.01	0.001
64	unconditional	0.0001	0.001	0.0001
128	unconditional	0.0001	0.0001	0.0001
256	unconditional	0.00001	0.0001	0.00001

Table 5: The results for parabolic PDE with periodic boundary conditions. The Lie algebra action is based on a Finite Difference Method. Number of nodes $N+1$ and stable time step for Euler Forward and a RK4 Method. The initial condition is $u_0(x) = e^{-10x^2} - e^{-10}$ and the diffusion is $\mu(x) = 1 - 0.1x^2$.

	Euler Forward		RK4 Method	
	On manifolds	Classical	On manifolds	Classical
$N-1$	Δt	stable Δt	stable Δt	stable Δt
8	unconditional	0.01	0.1	0.01
16	unconditional	0.001	0.01	0.001
32	unconditional	0.001	0.01	0.001
64	unconditional	0.0001	0.001	0.0001
128	unconditional	0.0001	0.0001	0.0001
256	unconditional	0.00001	0.0001	0.00001

Table 6: The results for parabolic PDE with Dirichlet boundary conditions. The Lie algebra action is based on a Finite Difference Method. Number of nodes $N+1$ and stable time step for Euler Forward and a RK4 Method. The initial condition is $u_0(x) = e^{-10x^2} - e^{-10}$ and the diffusion is $\mu(x) = 1 - 0.1x^2$.

	Euler Forward		RK4 Method	
	On manifolds	Classical	On manifolds	Classical
$N+1$	Δt	stable Δt	stable Δt	stable Δt
8	unconditional	0.01	0.1	0.01
16	unconditional	0.001	0.01	0.01
32	unconditional	0.001	0.01	0.001
64	unconditional	0.0001	0.001	0.0001
128	unconditional	0.0001	0.0001	0.0001
256	unconditional	0.00001	0.0001	0.00001

Table 7: The results for parabolic PDE with Neumann boundary conditions. The Lie algebra action is based on a Finite Difference Method. Number of nodes $N+1$ and stable time step for Euler Forward and a RK4 Method. The initial condition is $u_0(x) = \frac{1}{3}x^3 - x$ and the diffusion is $\mu(x) = 1 - 0.1x^2$.

	Euler Forward		RK4 Method	
	On manifolds	Classical	On manifolds	Classical
N	Δt	stable Δt	stable Δt	stable Δt
8	unconditional	0.01	0.1	0.01
16	unconditional	0.001	0.01	0.01
32	unconditional	0.001	0.01	0.001
64	unconditional	0.0001	0.001	0.0001
128	unconditional	0.0001	0.0001	0.0001
256	unconditional	0.00001	0.0001	0.00001

Table 8: The results for parabolic PDE with Dirichlet-Neumann boundary conditions. The Lie algebra action is based on a Finite Difference Method. Number of nodes $N+1$ and stable time step for Euler Forward and a RK4 Method. The initial condition is $u_0(x) = -\frac{1}{3}x^2 + \frac{2}{3}x + 1$ and the diffusion is $\mu(x) = 1 - 0.1x^2$.

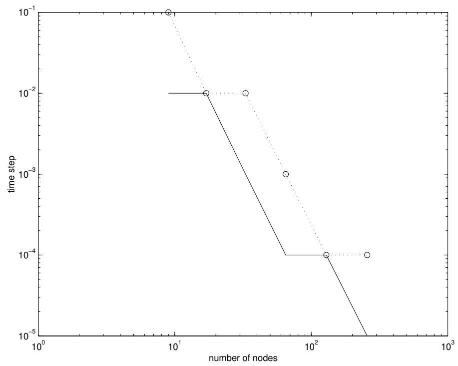

Figure 2: (o) Stable time step for the RK4 Method on manifolds based on Table 5. (-) Stable time step for the classical RK4 Method based on Table 5.

The results in Tables 5 - 8 are similar even though they come from different boundary conditions. From Tables 5 - 8 and Figure 2 it is clear the time step for the RK4 Method on manifolds is limited by

$$\Delta t \le C_3 (\Delta x)^2,$$

and the Runge Kutta Methods on manifolds allow larger time steps than classical Runge Kutta methods.

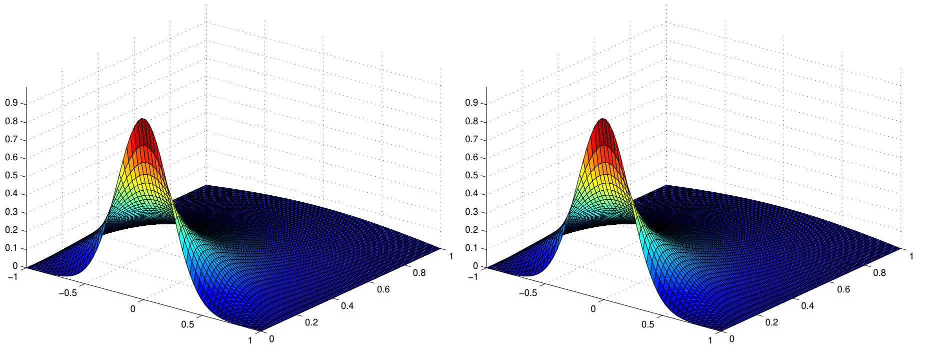

Figure 3: Solution of (1) with Dirichlet boundary conditions employing Euler Forward Method on manifolds and a RK4 Method on Manifolds.

It can be seen from Figure 3 that the Euler Forward Method and the RK4 Method yield similar results.

8 Higher order methods obtained by extrapolation

The Lie algebra action arising from Finite Difference discretization yields the Euler Forward Method on manifolds which seems unconditionally stable. $u(\frac{\Delta t}{n})$ is defined as the solution of (1) at time $t = \Delta t$ after n time steps $\frac{\Delta t}{n}$ with the Euler Forward Method. Assuming the error in $u(\Delta t)$ compared with the exact value $I = u(x, \Delta t)$ is of order $O(\Delta t)$, i.e.

$$\begin{aligned} u(\Delta t) &= I + c_1 \Delta t, \\ u(\frac{\Delta t}{2}) &= I + c_1 \frac{\Delta t}{2}, \end{aligned}$$

yields

$$I = -u(\Delta t) + 2u(\frac{\Delta t}{2}). \tag{30}$$

It is easy to compute $u(\Delta t)$ and $u(\frac{\Delta t}{2})$, finally I given by (30) is the final approximation of $u(x, \Delta t)$. Terms of higher order were neglected in this extrapolation. By using terms of order $O((\Delta t)^2)$

$$\begin{aligned} u(\Delta t) &= I + c_1 \Delta t + c_2 (\Delta t)^2, \\ u(\frac{\Delta t}{2}) &= I + c_1 \frac{\Delta t}{2} + c_2 (\frac{\Delta t}{2})^2, \\ u(\frac{\Delta t}{3}) &= I + c_1 \frac{\Delta t}{3} + c_2 (\frac{\Delta t}{3})^2, \end{aligned}$$

the following formula is obtained

$$I = \frac{u(\Delta t) - 8u(\frac{\Delta t}{2}) + 9u(\frac{\Delta t}{4})}{2}.$$

This is a method of order 3. Another way to obtain a higher order method is to define $u(\frac{\Delta t}{2^n})$ as the solution of (1) at time $t = \Delta t$ after 2^n time steps $\frac{\Delta t}{2^n}$ with the Euler Forward Method. Assuming the error in $u(\Delta t)$ compared with the exact value $I = u(x, \Delta t)$ is of order $O(\Delta t)$, i.e.

$$
\begin{aligned}
u(\Delta t) &= I + c_1 \Delta t, \\
u(\frac{\Delta t}{2}) &= I + c_1 \frac{\Delta t}{2},
\end{aligned}
$$

yields again (30). Terms of higher order were neglected in this extrapolation. By using terms of order $O((\Delta t)^2)$

$$
\begin{aligned}
u(\Delta t) &= I + c_1 \Delta t + c_2 (\Delta t)^2, \\
u(\frac{\Delta t}{2}) &= I + c_1 \frac{\Delta t}{2} + c_2 (\frac{\Delta t}{2})^2, \\
u(\frac{\Delta t}{4}) &= I + c_1 \frac{\Delta t}{4} + c_2 (\frac{\Delta t}{4})^2,
\end{aligned}
$$

the following formula is obtained

$$
I = \frac{u(\Delta t) - 6u(\frac{\Delta t}{2}) + 8u(\frac{\Delta t}{4})}{3}.
$$

This is a method of order 3. Numerical experiments on our test problem confirm the expected order of the methods and the unconditional stability. A drawback of the extrapolation is the rapidly growing complexity to obtain a higher order.

9 Conclusions

Implicit Finite Difference Schemes for a parabolic equation lead to systems of equations which must be solved by some algorithm with complexity $O(N)$, while explicit Finite Difference Schemes for a parabolic equation are limited by CFL Condition. The complexity of our schemes is $O(Nlog(N))$ and they seem to be unconditionally stable in some cases. Runge Kutta Methods on manifolds based on Finite Difference Methods allows larger time steps than with classical, explicit Runge Kutta Methods.

The methods based on Finite Difference Methods in this paper may easily be generalized to PDEs with several spatial dimensions. Finite Difference discretization of a simpler parabolic PDE with constant diffusivity on a equidistant grid yields a differentiation matrix. The eigenvalues and the eigenvectors of that matrix may be determined using tensor theory. Lie algebra action may be again defined as a solution of the semidiscretized simpler parabolic PDE.

An interesting question is whether other PDEs may be solved efficiently with Runge Kutta Methods on manifolds. In [3] the authors have applied the Runge Kutta Methods on manifolds to solve the KdV equation and also report improved stability conditions.

Acknowledgment

I would like to thank Hans Munthe-Kaas for suggesting the investigation on this topic which was started by him and his former student Espen Lodden.

References

[1] S. P. Nørsett A. Zanna A. Iserles, H. Z. Munthe-Kaas. Lie-group methods. *Acta Numerica*, pages 1–148, 1999.

[2] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, 1988.

[3] Jing Bo Chen and Meng Zhao Qin. Square-conservative schemes of any order for the kdv equation. 2001.

[4] Bengt Fornberg. *A Practical Guide to Pseudospectral Methods*. CAMBRIDGE UNIVERSITY PRESS, 1998.

[5] Espen Lodden. Geometric integration of the heat equation. Master's thesis, Department of Informatics, University of Bergen, March 2000.

[6] Hans Munthe-Kaas. High order Runge-Kutta methods on manifolds. *Applied Numerical Mathematics*, 29:115–127, 1999.

[7] Morgan Pickering. *An Introduction to Fast Fourier Transform Methods for Partial Differential Equations with Applications*. Research Studies Press and John Wiley and Sons, 1986.

[8] John C. Strikwerda. *Finite difference schemes and partial differential equations*. Chapman & Hall, 1989.