README
for the MATLAB EXPINT package

Håvard Berland   Bård Skaflestad   Will Wright

November 23, 2005

1 Introduction

This is some sort of appendix associated to the paper and MATLAB package "EXPINT — A MATLAB package for exponential integrators", [2], listing the problems included in the matlab package and also listing all the coefficient schemes. Parts of this files contents of this file may also be found in the Technical Report [3].

We assume that the reader has read the accompanying paper [2].

2 Matlab requirements

The package requires MATLAB release 13 (version 6.5). The following features, introduced in MATLAB 6.5, are needed

- dynamic structure field-names
- short-circuiting logical operators
- regular expression support
- the `mat2cell` function which, although present in the neural networks toolbox prior to MATLAB 6.5, was not introduced into the core language until version 6.5.
- multiple argument version of functions `warning` and `error`

These can certainly be circumvented so that the package would work on versions prior to 6.5 if necessary.

3 Version 1.1 release notes

This section describes user visible changes from version 1.0 to 1.1 of the expint package.

- The file `integrator.m` has changed name to `expglm.m`. This is for easier integration of this package into other existing matlab frameworks, where the name "integrator" could mean something more general than our specialized integrator for exponential general linear methods.
- More exponential integrators have been added. Most notably the eglm* and pec* schemes, from [19, 22]

4 Included problems

4.1 The 1D nonlinear Schrödinger equation — `nls.m`

This problem is the equation

$$\mathrm{i}y_t = -y_{xx} + (V(x) + \lambda|y|^2)y, \qquad x \in [-\pi, \pi].$$

with some initial condition and with periodic boundary conditions. The problem file includes the (spectral) semi-discretization of the problem. Upon initialising of the problem in MATLAB, different choices of the potential, the initial condition and λ may be chosen.

4.2 The 1D KdV–equation — `kdv.m`

The Korteweg–de Vries equation with periodic boundary conditions in 1D is

$$y_t = -y_{xxx} - yy_x, \qquad x \in [-\pi, \pi].$$

This is semi-discretized spectrally and integrated from $t = 0$ to $t = 2\pi/625$ by default, the linear part is the diagonal matrix with entries $L_{kk} = \mathrm{i}k^3$, $k = -\mathtt{ND}/2 + 1, \ldots, \mathtt{ND}/2$ and the nonlinear function is

$$N(y(k), t) = -\frac{\mathrm{i}}{2}k\mathcal{F}(\mathcal{F}^{-1}(y)^2).$$

The default value of `ND` is 128. The eigenvalues of L are complex and therefore, the problem exhibits rapid oscillations for high wave number modes. Various initial conditions are supported, a well known choice, taken from [14] is

$$y(x, 0) = 3\lambda\,\mathrm{sech}^2\left(\sqrt{\lambda}x/2\right).$$

4.3 The Kuramoto–Sivashinsky equation — `kursiv.m`

The Kuramoto–Sivashinsky equation has been used to study many reaction-diffusion systems, in 1D it is written as

$$y_t = -y_{xx} - y_{xxxx} - yy_x, \qquad x \in [0, 32\pi].$$

We use a spectral discretization with periodic boundary conditions and `ND` equal to 128. The problem is integrated from $t = 0$ to $t = 65$ by default. The linear term is diagonal with elements $L_{kk} = k^2 - k^4$, $k = -\texttt{ND}/2+1, \dots, \texttt{ND}/2$, which results in rapid decay for high wave numbers. The nonlinear function is

$$N(y(k), t) = -\frac{\mathrm{i}}{2} k \mathcal{F}(\mathcal{F}^{-1}(y)^2).$$

Various choices of initial condition are supported, the choice of smooth initial condition is taken from [13]

$$y(x, 0) = \cos(x/16)\left(1 + \sin(x/16)\right).$$

4.4 Burgers' equation — `burgers.m`

This equation, dating back to 1915, has been used in the study of turbulence and shock formation, it reads

$$y_t = \lambda y_{xx} - \frac{1}{2}(y^2)_x, \qquad x \in [-\pi, \pi].$$

We use a spectral discretisation with `ND` equal to 128 as default spatial resolution and various choices of initial condition are supported. The most commonly used for this equation, see [13], is

$$y(x, 0) = \exp(-10 \sin^2(x/2)).$$

The rapid oscillations apparent in this problem come from the λy_{xx} term, where $\lambda = 0.03$ is the default value.

4.5 Heat equation with source term — `hochost.m`

This particular equation appeared in [10] and is a good indicator of stiff order of exponential integrators.

$$y_t = y_{xx} + \frac{1}{1+y^2} + \Phi, \qquad x \in [0, 1],$$

where Φ is chosen so that the exact solution is $y(x,t) = x(1-x)\mathrm{e}^t$. Boundary conditions are homogeneous Dirichlet. The problem is discretized in space using a standard finite difference scheme, with `ND` by default set to 200. The resulting ODE is integrated from $t = 0$ to $t = 1$, with various initial conditions supported. This problem results in a reduction in order for almost all schemes.

4.6 The Allen–Cahn equation — `allencahn.m`

The Allen–Cahn equation is a parabolic problem, which reads

$$\begin{aligned} y_t &= \lambda y_{xx} + y - y^3, \qquad x \in [-1, 1], \\ y(x, 0) &= 0.53x + 0.47 \sin(-1.5\pi x), \end{aligned}$$

we choose to implement this equation with the Dirichlet boundary conditions $y(-1, t) = -1$ and $y(1, t) = 1$. The linear part λy_{xx} is discretized using a Chebyshev differentiation matrix, resulting in a full matrix L. Further details may be found in [13]. In order to deal with the boundary conditions, one defines $y = w + x$ and works with the variable w which has homogeneous boundary values. The default parameter values are $\texttt{ND} = 64$ and $\lambda = 0.001$.

4.7 The 2D complex Ginzburg–Landau equations — `cginzland2d.m`

This equation describes reaction-diffusion equations close to a Hopf bifurcation and generates spiral wave fronts

$$y_t = y + (1 + \mathrm{i}\alpha)\nabla^2 y - (1 + \mathrm{i}\beta)y|y|^2, \qquad x, y \in [0, 200].$$

Both smooth (a series of Gaussian pulses) and random initial conditions are supported, see [12] or the source code for more details. We implement this equation using a Fourier spectral discretisation in 2D, this is

$$\widehat{y}_t = (1 - (1 + \mathrm{i}\alpha)(k^2 + l^2))\widehat{y} - \mathcal{F}((1 + \mathrm{i}\beta)y|y|^2).$$

By default we use `ND` equal to 128 and integrate from $t = 0$ to $t = 150$, with bifurcation parameters $\alpha = 0$ and $\beta = 1.3$.

4.8 The Gray–Scott equations in 2D — `grayscott2d.m`

The Gray–Scott equation is a reaction-diffusion equation which exhibits a wide variety of interesting patterns. In non-dimensional form the system is

$$\begin{aligned} u_t &= D_u \nabla^2 u - uv^2 + \alpha(1 - u), \\ v_t &= D_v \nabla^2 v + uv^2 - (\alpha + \beta)v, \end{aligned}$$

where the positive diffusion parameters D_u, D_v are generally chosen so that the ratio $D_u/D_v = 2$. The default choice is $D_u = 2 \cdot 10^{-5}$ and $D_v = 1 \cdot 10^{-5}$. The constants α and β can be viewed as bifurcation parameters. As a default choice we set $\alpha = 0.065$ and $\beta = 0.035$. Defining $y = [u, v]^T$, represents the equations in the appropriate form. Note that the transformed equations in Fourier space are very similar to the original equations, they read

$$\begin{aligned} \widehat{u}_t &= -D_u(k^2 + l^2)\widehat{u}(l, k) - \mathcal{F}(uv^2 - \alpha(1 - u)), \\ \widehat{v}_t &= -D_v(k^2 + l^2)\widehat{v}(l, k) + \mathcal{F}(uv^2 - (\alpha + \beta)v). \end{aligned}$$

The initial conditions we choose can be found in the source code, the smooth initial condition we have implemented are scaled Gaussian pulses. By default `ND` is chosen to be 128 on the grid $x, y \in [0, 2.5]$. The package also includes discretizations of this problem in one and three space dimensions. We refer the reader to the source code in `grayscott.m` and `grayscott3d.m` respectively for more details. In the directory `tests` you will find files `testgrayscott2d.m` and `testgrayscott3d.m`, which can be viewed as typical test scripts.

4.9 The sine-Gordon equation

This is a nonlinear wave equation, here in 1D,

$$
\begin{aligned}
u_{tt} &= u_{xx} - \sin(u) \\
u(x,0) &= u_0(x) \\
u_t(x,0) &= v_0(x)
\end{aligned}
$$

This package requires differential equations to be written in the format $y_t = Ly + N(y,t)$, for this we introduce the extended phase space variable $\dot{y} = [u, u_t]^T$. We use spectral discretization in space.

Two initial conditions are included, one stationary soliton solution (which requires the length parameter to be at least 40 to avoid boundary conditions interfering. The other initial condition is in the unstable regime of the equation and requires the length parameter to be $2\sqrt{2}\pi$, see [11]. The corresponding Hamiltonian function

$$
H(u,v) = \int_0^L \frac{1}{2}v^2 + \frac{1}{2}u_x^2 + 1 - \cos(u)\,\mathrm{d}x \tag{1}
$$

is implemented in the function `sinegordon_hamilt.m`

5 Coefficient functions for included schemes

In this section we list all schemes implemented in this package, with reference to their origin. For space and æstethic reasons we adopt the notation

$$
\varphi_{i,j} = \varphi_i(c_j z) \qquad i = 0, 1, \ldots, \text{ and } j = 1, \ldots, s \tag{2}
$$

where $\varphi_0(z) = \mathrm{e}^z$. Also we use 1 to represent the identity matrix. All schemes with relevant figures describing their performance are listed in Table 1.

5.1 Lawson schemes

Lawson schemes are constructed by applying the Lawson transformation [15] to the semi-linear problem, then solving the transformed equation by a standard numerical scheme then back transforming. This whole process can be written in the original variables see [8], and results in the coefficients of the method involving exponentials. Below we include some Adam–Bashforth–Lawson and Runge–Kutta–Lawson schemes of low order.

5.1.1 Lawson–Euler — `lawsoneuler.m`

This is the simplest example of a Lawson scheme, which we choose to call Lawson–Euler. It has also occasionally been called the “exponential Euler scheme”, but this is confusing given that the Nørsett–Euler scheme also goes by this name. This scheme has stiff order one.

$$
\begin{array}{c|c|c}
0 & & 1 \\
\hline
 & \varphi_0 & \varphi_0
\end{array}
$$

Name	Nonstiff p	Stiff p	Stages s	Output r	$\#\varphi$	matvecs
Lawson–Euler	1	1	1	1	1	1
ABLawson4	4	1	1	4	4	4
Lawson4	4	1	4	1	2	6
Nørsett–Euler	1	1	1	1	2	2
ABNørsett4	4	4	1	4	5	5
ETD4RK	4	2	4	1	6	10
Krogstad	4	3	4	1	7	11
Hochbruck–Ostermann	4	4	5	1	8	13
Cfree4	4	2	4	1	4	9
RKMK4t	4	2	4	1	4	9
GenLawson43	4	4	4	3	8	16
ModGenLawson43	4	4	4	3	9	17
PEC423	4	4	2	3	5	8
PECEC433	4	4	3	3	5	10

Table 1: Selected integrators included in the package, along with relevant figures describing their properties. $\#\varphi$ is the number of distinct φ functions needed to be evaluated for each scheme. Note that counting the number of φ functions and matrix-vector products does not give a complete description of the efficiency of the scheme.

5.1.2 ABLawson2 — `ablawson2.m`

This scheme is based on the Adams–Bashforth scheme of order two, we represent the scheme in this form so that the incoming approximations are $y^{[n-1]} = [y_{n-1}, hN_{n-2}]$, in accordance with the starting procedure implemented in `expglm.m`. This scheme has stiff order one.

$$
\begin{array}{c|cc|cc}
0 & & & 1 & 0 \\
1 & \frac{3}{2}\varphi_0 & & \varphi_0 & -\frac{1}{2}\varphi_0^2 \\
\hline
 & \frac{3}{2}\varphi_0 & 0 & \varphi_0 & -\frac{1}{2}\varphi_0^2 \\
 & 1 & 0 & 0 & 0
\end{array}
$$

5.1.3 ABLawson3 — `ablawson3.m`

This scheme has stiff order one and is based on the Adams–Bashforth scheme of order three and is represented in this way so that the incoming approximation has the form $y^{[n-1]} = [y_{n-1}, hN_{n-2}, hN_{n-3}]^T$.

$$
\begin{array}{c|cc|ccc}
0 & & & 1 & 0 & 0 \\
1 & \frac{23}{12}\varphi_0 & & \varphi_0 & -\frac{4}{3}\varphi_0^2 & \frac{5}{12}\varphi_0^3 \\
\hline
 & \frac{23}{12}\varphi_0 & 0 & \varphi_0 & -\frac{4}{3}\varphi_0^2 & \frac{5}{12}\varphi_0^3 \\
 & 1 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

5.1.4 ABLawson4 — `ablawson4.m`

This scheme has stiff order one and is based on the Adams–Bashforth scheme of order four and is represented in this way so that the incoming approximation has the form

$$
y^{[n-1]} = [y_{n-1}, hN_{n-2}, hN_{n-3}, hN_{n-4}]^T.
$$

$$
\begin{array}{c|cc|cccc}
0 & & & 1 & 0 & 0 & 0 \\
1 & \frac{55}{12}\varphi_0 & & \varphi_0 & -\frac{59}{24}\varphi_0^2 & \frac{37}{24}\varphi_0^3 & -\frac{3}{8}\varphi_0^4 \\
\hline
 & \frac{55}{12}\varphi_0 & 0 & \varphi_0 & -\frac{59}{24}\varphi_0^2 & \frac{37}{24}\varphi_0^3 & -\frac{3}{8}\varphi_0^4 \\
 & 1 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 1 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

5.1.5 Lawson2a — `lawson2a.m`

Based on the midpoint rule see [5, Eq. (232b)], this scheme has stiff order one.

$$
\begin{array}{c|cc|c}
0 & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{0,2} & & \varphi_{0,2} \\
\hline
 & 0 & \varphi_{0,2} & \varphi_0
\end{array}
$$

5.1.6 Lawson2b — `lawson2b.m`

Based on the trapezoidal rule see [5, Eq. (232a)], this scheme has stiff order one.

$$
\begin{array}{c|cc|c}
0 & & & 1 \\
1 & \varphi_0 & & \varphi_0 \\
\hline
 & \frac{1}{2}\varphi_0 & \frac{1}{2} & \varphi_0
\end{array}
$$

5.1.7 Lawson4 — `lawson4.m`

Based on the classical fourth order scheme of Kutta see [5, Eq. (235i)], this scheme has stiff order one.

$$
\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{0,2} & & & & \varphi_{0,2} \\
\frac{1}{2} & & \frac{1}{2} & & & \varphi_{0,2} \\
1 & & & \varphi_{0,2} & & \varphi_0 \\
\hline
 & \frac{1}{6}\varphi_0 & \frac{1}{3}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{6} & \varphi_0
\end{array}
\tag{3}
$$

5.2 ETD-schemes

ETD schemes are based on algebraic approximations to the nonlinear term in the variation of constants formula. ETD means “Exponential Time Differencing” and the name stems from [7]. This is not a particularly good name for this type of scheme as the name does not indicate anything about this type of schemes distinguishing it from other exponential integrators. Nevertheless, we still adopt the term for the time being.

Nørsett [18] developed a class of schemes which reduced to the Adams–Bashforth methods when the linear part of the problem is zero. The Adam–Bashforth–Nørsett schemes have recently been reinvented by Cox and Matthews [7] and Beylkin, Keiser and Vozovoi [4]. Many ETD schemes based on Runge–Kutta methods have also been developed, below we give some of the well known schemes.

5.2.1 Nørsett–Euler — `norsetteuler.m`

The most well known exponential version of the Euler method was first found by Nørsett in [18]. It is also known as the exponentially fitted Euler, ETD Euler, ETD1RK, filtered Euler scheme, or Lie–Euler as it is the simplest Lie group integrator with the affine Lie group, it has stiff order one.

$$
\begin{array}{c|c|c}
0 & & 1 \\
\hline
 & \varphi_1 & \varphi_0
\end{array}
$$

5.2.2 ABNørsett2 — `abnorsett2.m`

This stiff order two scheme of Nørsett [18], is implemented in this way so that the incoming approximation has the same form as in ABLawson2.

$$
\begin{array}{c|cc|cc}
0 & & & 1 & 0 \\
1 & \varphi_1+\varphi_2 & & \varphi_0 & -\varphi_2 \\
\hline
 & \varphi_1+\varphi_2 & 0 & \varphi_0 & -\varphi_2 \\
 & 1 & 0 & 0 & 0
\end{array}
$$

5.2.3 ABNørsett3 — `abnorsett3.m`

This stiff order three scheme of Nørsett [18], is implemented in this way so that the incoming approximation has the same form as in ABLawson3.

$$
\begin{array}{c|cc|ccc}
0 & & & 1 & 0 & 0 \\
1 & \varphi_1 + \frac{3}{2}\varphi_2 + \varphi_3 & & \varphi_0 & -2\varphi_2 - 2\varphi_3 & \frac{1}{2}\varphi_2 + \varphi_3 \\
\hline
 & \varphi_1 + \frac{3}{2}\varphi_2 + \varphi_3 & 0 & \varphi_0 & -2\varphi_2 - 2\varphi_3 & \frac{1}{2}\varphi_2 + \varphi_3 \\
 & 1 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

5.2.4 ABNørsett4 — `abnorsett4.m`

This stiff order four scheme of Nørsett [18], is implemented in this way so that the incoming approximation has the same form as in ABLawson4.

$$
\begin{array}{c|cc|cccc}
0 & & & 1 & 0 & 0 & 0 \\
1 & \varphi_1 + \frac{11}{6}\varphi_2 + 2\varphi_3 + \varphi_4 & & \varphi_0 & -3\varphi_2 - 5\varphi_3 - 3\varphi_4 & \frac{3}{2}\varphi_2 + 4\varphi_3 + 3\varphi_4 & \Phi \\
\hline
1 & \varphi_1 + \frac{11}{6}\varphi_2 + 2\varphi_3 + \varphi_4 & 0 & \varphi_0 & -3\varphi_2 - 5\varphi_3 - 3\varphi_4 & \frac{3}{2}\varphi_2 + 4\varphi_3 + 3\varphi_4 & \Phi \\
 & 1 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 1 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

where

$$\Phi = -\frac{1}{3}\varphi_2 - \varphi_3 - \varphi_4.$$

5.2.5 PEC322 — `pec322.m`

The stiff order three scheme is based on an ABNørsett2 predictor and AMNørsett3 corrector. This method was constructed in [19]. Methods in this class can be derived directly from the order conditions by choosing the free parameter $c_2 = 1$, which minimises the number of φ function evaluations required.

$$
\begin{array}{c|cc|cc}
0 & & & 1 & \\
1 & \varphi_1 + \varphi_2 & & \varphi_0 & -\varphi_2 \\
\hline
 & \varphi_1 - 2\varphi_3 & \frac{1}{2}\varphi_2 + \varphi_3 & \varphi_0 & -\frac{1}{2}\varphi_2 + \varphi_3 \\
 & 1 & 0 & 0 & 0
\end{array}
$$

5.2.6 PEC423 — `pec423.m`

The stiff order three scheme is based on an ABNørsett3 predictor and AMNørsett4 corrector. This method was constructed in [19].

$$
\begin{array}{c|cc|ccc}
0 & & & 1 & & \\
1 & \varphi_1 + \frac{3}{2}\varphi_2 + \varphi_3 & & \varphi_0 & -2\varphi_2 - 2\varphi_3 & \frac{1}{2}\varphi_2 + \varphi_3 \\
\hline
 & \varphi_1 + \frac{1}{2}\varphi_2 - 2\varphi_3 - 3\varphi_4 & \frac{1}{3}\varphi_2 + \varphi_3 + \varphi_4 & \varphi_0 & -\varphi_2 + \varphi_3 + 3\varphi_4 & \frac{1}{6}\varphi_2 - \varphi_4 \\
 & 1 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

Also included in the package are the schemes `pec524.m`, `pec625.m` and `pec726.m` with orders 5, 6 and 7 respectively. Note that `pec221.m` is not included as this is the same as the scheme `etd2rk.m`.

5.2.7 PECEC332 — `pecec332.m`

The stiff order three scheme is based on an ABNørsett2 predictor and AMNørsett3 corrector applied twice. Methods in this class can be derived directly from the order conditions by choosing the free parameter $c_2 = 1$ and $c_3 = 1$, which minimises the number of φ function evaluations required.

$$
\begin{array}{c|ccc|cc}
0 & & & & 1 & \\
1 & \varphi_1 + \varphi_2 & & & \varphi_0 & -\varphi_2 \\
1 & \varphi_1 - 2\varphi_3 & \frac{1}{2}\varphi_2 + \varphi_3 & & \varphi_0 & -\frac{1}{2}\varphi_2 + \varphi_3 \\
\hline
 & \varphi_1 - 2\varphi_3 & 0 & \frac{1}{2}\varphi_2 + \varphi_3 & \varphi_0 & -\frac{1}{2}\varphi_2 + \varphi_3 \\
 & 1 & 0 & 0 & 0 & 0
\end{array}
$$

5.2.8 PECEC433 — `pecec433.m`

The stiff order three scheme is based on an ABNørsett2 predictor and AMNørsett3 corrector applied twice. Methods in this class can be derived directly from the order conditions by choosing the free parameter $c_2 = 1$ and $c_3 = 1$.

$$
\begin{array}{c|ccc|ccc}
0 & & & & 1 & & \\
1 & \varphi_1 + \frac{3}{2}\varphi_2 + \varphi_3 & & & \varphi_0 & -2\varphi_2 - 2\varphi_3 & \frac{1}{2}\varphi_2 + \varphi_3 \\
1 & \varphi_1 + \frac{1}{2}\varphi_2 - 2\varphi_3 - 3\varphi_4 & \frac{1}{3}\varphi_2 + \varphi_3 + \varphi_4 & & \varphi_0 & -\varphi_2 + \varphi_3 + 3\varphi_4 & \frac{1}{6}\varphi_2 - \varphi_4 \\
\hline
 & \varphi_1 + \frac{1}{2}\varphi_2 - 2\varphi_3 - 3\varphi_4 & 0 & \Phi & \varphi_0 & -\varphi_2 + \varphi_3 + 3\varphi_4 & \frac{1}{6}\varphi_2 - \varphi_4 \\
 & 1 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

where

$$
\Phi = \frac{1}{3}\varphi_2 + \varphi_3 + \varphi_4.
$$

Also included in the package are the schemes `pec534.m`, `pec635.m` and `pec736.m` with orders 5, 6 and 7 respectively.

5.2.9 EGLM332 — `EGLM332.m`

The stiff order three scheme is derived directly from the order conditions given the absiccae values $c_2 = 1/2$ and $c_3 = 1$.

$$
\begin{array}{c|ccc|cc}
0 & & & & 1 & \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} + \frac{1}{4}\varphi_{2,2} & & & \varphi_0 & -\frac{1}{4}\varphi_{2,2} \\
1 & \varphi_1 - \varphi_2 - 4\varphi_3 & \frac{4}{3}\varphi_2 + \frac{8}{3}\varphi_3 & & \varphi_0 & -\frac{1}{3}\varphi_2 + \frac{4}{3}\varphi_3 \\
\hline
 & \varphi_1 - 2\varphi_2 - 2\varphi_3 + 12\varphi_4 & \frac{8}{3}\varphi_2 - 16\varphi_4 & -\frac{1}{2}\varphi_2 + \varphi_3 + 6\varphi_4 & \varphi_0 & -\frac{1}{6}\varphi_2 + \varphi_3 - 2\varphi_4 \\
 & 1 & 0 & 0 & 0 & 0
\end{array}
$$

Also included in the package are the schemes `eglm433.m` of order 4.

5.2.10 ETD2RK — `etd2rk.m`

This scheme first derived by Strehmel and Weiner [20, Eq. (3.6)], has roots in chemistry [16, Section 3] and was recently derived by Cox and Matthews [7, Eq. (22)], it has stiff order two.

$$
\begin{array}{c|cc|c}
0 & & & 1 \\
1 & \varphi_1 & & \varphi_0 \\
\hline
 & \varphi_1 - \varphi_2 & \varphi_2 & \varphi_0
\end{array}
$$

5.2.11 ETD3RK — `etd3rk.m`

This scheme was first derived by Friedli in [9, Section 4], and more recently appeared in [7, Eq. (23)–(25)], it has stiff order three.

$$
\begin{array}{c|ccc|c}
0 & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & \varphi_{0,2} \\
1 & -\varphi_1 & 2\varphi_1 & & \varphi_0 \\
\hline
 & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 4\varphi_2 - 8\varphi_3 & -\varphi_2 + 4\varphi_3 & \varphi_0
\end{array}
$$

5.2.12 Ehle–Lawson — `ehlelawson.m`

This is a scheme of Ehle and Lawson [8] made to remedy problems with the Lawson schemes. It has four stages but only stiff order two.

$$
\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \varphi_{0,2} \\
\frac{1}{2} & & \frac{1}{2}\varphi_{1,2} & & & \varphi_{0,2} \\
1 & & & \varphi_1 & & \varphi_0 \\
\hline
 & \varphi_1 - 3\varphi_2 + \varphi_3 & 2\varphi_2 - \varphi_3 & 2\varphi_2 - \varphi_3 & -\varphi_2 + \varphi_3 & \varphi_0
\end{array}
$$

5.2.13 ETD4RK — `etd4rk.m`

This scheme due to Cox and Matthews in [7, Eq. (26)–(29)], was one of the schemes that kick started the recent focus on exponential integrators, unfortunately it has only stiff order two.

$$
\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \varphi_{0,2} \\
\frac{1}{2} & & \frac{1}{2}\varphi_{1,2} & & & \varphi_{0,2} \\
1 & \frac{1}{2}\varphi_{1,2}(\varphi_{0,2} - 1) & & \varphi_{1,2} & & \varphi_0 \\
\hline
 & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & -\varphi_2 + 4\varphi_3 & \varphi_0
\end{array}
$$

5.2.14 Krogstad — `krogstad.m`

This scheme appeared in [14, Eq. (51)] as a variant of ETD4RK by adding the φ_2-function in the internal stages, it does not require stage splittings and has stiff order three.

$$
\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \varphi_{0,2} \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} - \varphi_{2,2} & \varphi_{2,2} & & & \varphi_{0,2} \\
1 & \varphi_1 - 2\varphi_2 & & 2\varphi_2 & & \varphi_0 \\
\hline
 & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & -\varphi_2 + 4\varphi_3 & \varphi_0
\end{array}
$$

5.2.15 Strehmel–Weiner — `strehmelweiner.m`

This scheme first appeared in [21, Example 4.5.5] one of the earliest exponential Runge–Kutta methods, it has stiff order three.

$$
\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \varphi_{0,2} \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} - \frac{1}{2}\varphi_{2,2} & \frac{1}{2}\varphi_{2,2} & & & \varphi_{0,2} \\
1 & \varphi_1 - 2\varphi_2 & -2\varphi_2 & 4\varphi_2 & & \varphi_0 \\
\hline
 & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 0 & 4\varphi_2 - 8\varphi_3 & -\varphi_2 + 4\varphi_3 & \varphi_0
\end{array}
$$

5.2.16 Friedli — `friedli.m`

This scheme appeared in [9, Section 5], it is also one of the earliest exponential Runge–Kutta methods, it has stiff order three.

$$
\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \varphi_{0,2} \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} - \frac{1}{2}\varphi_{2,2} & \frac{1}{2}\varphi_{2,2} & & & \varphi_{0,2} \\
1 & \varphi_1 - 2\varphi_2 & -\frac{26}{25}\varphi_1 + \frac{2}{25}\varphi_2 & \frac{26}{25}\varphi_1 + \frac{48}{25}\varphi_2 & & \varphi_0 \\
\hline
 & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 0 & 4\varphi_2 - 8\varphi_3 & -\varphi_2 + 4\varphi_3 & \varphi_0
\end{array}
$$

5.2.17 Hochbruck–Ostermann — `hochost4.m`

This scheme was developed by Hochbruck and Ostermann [10, Section 5], with five-stages is the only known exponential Runge–Kutta method with stiff order four.

$$
\begin{array}{c|ccccc|c}
0 & & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & & \varphi_{0,2} \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} - \varphi_{2,2} & \varphi_{2,2} & & & & \varphi_{0,2} \\
1 & \varphi_1 - 2\varphi_2 & \varphi_2 & \varphi_2 & & & \varphi_0 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} - 2a_{5,2} - a_{5,4} & a_{5,2} & a_{5,2} & a_{5,4} & & \varphi_{0,2} \\
\hline
 & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 0 & 0 & -\varphi_2 + 4\varphi_3 & 4\varphi_2 - 8\varphi_3 & \varphi_0
\end{array}
$$

where

$$
a_{5,2} = \frac{1}{2}\varphi_{2,2} - \varphi_3 + \frac{1}{4}\varphi_2 - \frac{1}{2}\varphi_{3,2}, \qquad a_{5,4} = \frac{1}{4}\varphi_{2,2} - a_{5,2}
$$

5.2.18 ETD5RKF — `etd5rkf.m`

This is a non-stiff fifth order scheme developed in [1]. It usually performs worse than other order four schemes presented here due to bad error constant. It is based on the six stage fifth order scheme of Fehlberg.

$$
c = \begin{bmatrix} 0 & \frac{2}{9} & \frac{1}{3} & \frac{3}{4} & 1 & \frac{5}{6} \end{bmatrix}^T \qquad u_{i1}(z) = \varphi_{0,i}(z) \qquad v_{11}(z) = \varphi_0(z)
$$

$$
A(z) =
$$

$$
\begin{bmatrix}
 & & & & & \\
-\frac{2}{3}\varphi_2 + \frac{10}{9}\hat{\varphi}_2 & & & & & \\
\frac{569}{11544}\varphi_2 + \frac{1355}{11544}\hat{\varphi}_2 & -\frac{831}{3848}\varphi_2 + \frac{2755}{3848}\hat{\varphi}_2 & & & & \\
-\frac{77157}{61568}\varphi_2 + \frac{143535}{61568}\hat{\varphi}_2 & \frac{587979}{61568}\varphi_2 - \frac{821745}{61568}\hat{\varphi}_2 & -\frac{405}{64}\varphi_2 + \frac{675}{64}\hat{\varphi}_2 & & & \\
\frac{655263}{7696}\varphi_2 - \frac{2031205}{23088}\hat{\varphi}_2 & -\frac{1148769}{7696}\varphi_2 + \frac{1252665}{7696}\hat{\varphi}_2 & \frac{1593}{40}\varphi_2 - \frac{405}{8}\hat{\varphi}_2 & \frac{144}{5}\varphi_2 - \frac{80}{3}\hat{\varphi}_2 & & \\
-\frac{2212835}{277056}\varphi_2 + \frac{6888625}{831168}\hat{\varphi}_2 & \frac{477285}{30784}\varphi_2 - \frac{496525}{30784}\hat{\varphi}_2 & -\frac{39}{16}\varphi_2 + \frac{65}{16}\hat{\varphi}_2 & -\frac{4}{9}\varphi_2 + \frac{20}{27}\hat{\varphi}_2 & -\frac{185}{96}\varphi_2 + \frac{575}{288}\hat{\varphi}_2 &
\end{bmatrix}
$$

$$
\begin{aligned}
b_{11}(z) &= \frac{47}{150}\varphi_1 - \frac{188}{75}\varphi_2 + \frac{94}{15}\varphi_3 & b_{12}(z) &= 0 \\
b_{13}(z) &= -\frac{43}{25}\varphi_1 + \frac{132}{5}\varphi_2 - 66\varphi_3 & b_{14}(z) &= \frac{4124}{75}\varphi_1 - \frac{6152}{15}\varphi_2 + \frac{2704}{3}\varphi_3 \\
b_{15}(z) &= \frac{189}{10}\varphi_1 - \frac{662}{5}\varphi_2 + 284\varphi_3 & b_{16}(z) &= -\frac{1787}{25}\varphi_1 + \frac{12966}{25}\varphi_2 - \frac{5628}{5}\varphi_3
\end{aligned}
$$

where $\hat{\varphi}_2(z) = \varphi_2(\frac{3}{5}z)$. Note that the coefficient in front of φ_3 is different from [1] due to differing definitions of φ_i.

5.3 Affine Lie group schemes

The construction of Lie group integrators for the solution of semi-discretized PDEs started with the paper of Munthe–Kaas [17], where the affine Lie group was used. The RKMK methods require the computation of the dexp^{-1} operator which involves iterated commutators. To overcome the need for commutators which often result in stepsize restrictions Celledoni, Martinsen and Owren [6], constructed the commutator-free methods.

5.3.1 RKMK2e — `rkmk2e.m`

This scheme is a generalization of the trapezoidal rule, can be derived from [17, Ex. 4], but is also a standard scheme found in the chemistry literature, known as the Pseudo-Steady-State-Approximation (PASSA) scheme, see [23, Section 2], it has stiff order one.

$$
\begin{array}{c|cc|c}
0 & & & 1 \\
1 & \varphi_1 & & \varphi_0 \\
\hline
 & \frac{1}{2}\varphi_1 & \frac{1}{2}\varphi_1 & \varphi_0
\end{array}
$$

5.3.2 ETD2CF3 — `etd2cf3.m`

This is a stiff order three ETD version [10] of a commutator-free scheme in [6].

$$
\begin{array}{c|ccc|c}
0 & & & & 1 \\
\frac{1}{3} & \frac{1}{3}\varphi_{1,2} & & & \varphi_{0,2} \\
\frac{2}{3} & \frac{2}{3}\varphi_{1,3} - \frac{4}{3}\varphi_{2,3} & \frac{4}{3}\varphi_{2,3} & & \varphi_{0,3} \\
\hline
 & \varphi_1 - \frac{9}{2}\varphi_2 + 9\varphi_3 & 6\varphi_2 - 18\varphi_3 & -\frac{3}{2}\varphi_2 + 9\varphi_3 & \varphi_0
\end{array}
$$

5.3.3 Cfree4 — `cfree4.m`

This scheme given in [6, Eq. (7)] assuming affine Lie group action is used, is of stiff order two. Note that the internal stages are equivalent to those of ETD4RK.

$$
\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \varphi_{0,2} \\
\frac{1}{2} & & \frac{1}{2}\varphi_{1,2} & & & \varphi_{0,2} \\
1 & \frac{1}{2}\varphi_{1,2}(\varphi_{0,2} - 1) & & \varphi_{1,2} & & \varphi_0 \\
\hline
 & \frac{1}{2}\varphi_1 - \frac{1}{3}\varphi_{1,2} & \frac{1}{3}\varphi_1 & \frac{1}{3}\varphi_1 & -\frac{1}{6}\varphi_1 + \frac{1}{3}\varphi_{1,2} & \varphi_0
\end{array}
$$

5.3.4 RKMK4t — `rkmk4t.m`

Using a suitable truncation of the dexp^{-1} operator leads to the method of Munthe–Kaas [17, Ex. 4], which again is of stiff order two but suffers from instabilities, especially when non-periodic boundary conditions are used.

$$
\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \varphi_{0,2} \\
\frac{1}{2} & \frac{z}{8}\varphi_{1,2} & \frac{1}{2}(1 - \frac{z}{4})\varphi_{1,2} & & & \varphi_{0,2} \\
1 & & & \varphi_1 & & \varphi_0 \\
\hline
 & \frac{1}{6}\varphi_1(1 + \frac{z}{2}) & \frac{1}{3}\varphi_1 & \frac{1}{3}\varphi_1 & \frac{1}{6}\varphi_1(1 - \frac{z}{2}) & \varphi_0
\end{array}
$$

5.4 Generalized Lawson schemes

Krogstad [14], constructed these schemes as a means of overcoming some of the undesirable properties of the Lawson schemes. The methods boil down to using a more sophisticated transformation which better approximates the dynamics of the original differential equation see [19] for more details. The transformation involves using approximations of the nonlinear term from previous steps, resulting in an exponential general linear method.

Two schemes are included in the package but not listed here due to space reasons, that is the `genlawson44.m` and `genlawson45.m`.

5.4.1 GenLawson41 — `genlawson41.m`

This exponential Runge–Kutta scheme closely related to Lawson4 has stiff order two.

$$
\begin{array}{c|cccc|c}
0 & & & & & 1 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} & & & & \varphi_{0,2} \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} - \frac{1}{2} & \frac{1}{2} & & & \varphi_{0,2} \\
1 & \varphi_1 - \varphi_{0,2} & & \varphi_{0,2} & & \varphi_0 \\
\hline
 & \varphi_1 - \frac{2}{3}\varphi_{0,2} - \frac{1}{6} & \frac{1}{3}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{6} & \varphi_0
\end{array}
$$

5.4.2 GenLawson42 — `genlawson42.m`

This scheme requires y_{n-1} and N_{n-2} as initial data. At least one step of an alternative method is needed to start the integration. The overall stiff order is three.

$$
\begin{array}{c|cccc|cc}
0 & & & & & 1 & 0 \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} + \frac{1}{4}\varphi_{2,2} & & & & \varphi_{0,2} & -\frac{1}{4}\varphi_{2,2} \\
\frac{1}{2} & \frac{1}{2}\varphi_{1,2} + \frac{1}{4}\varphi_{2,2} - \frac{3}{4} & \frac{1}{2} & & & \varphi_{0,2} & -\frac{1}{4}\varphi_{2,2} + \frac{1}{4} \\
1 & \varphi_1 + \varphi_2 - \frac{3}{2}\varphi_{0,2} & & \varphi_{0,2} & & \varphi_0 & -\varphi_2 + \frac{1}{2}\varphi_{0,2} \\
\hline
 & \varphi_1 + \varphi_2 - \varphi_{0,2} - \frac{1}{3} & \frac{1}{3}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{6} & \varphi_0 & -\varphi_2 + \frac{1}{3}\varphi_{0,2} + \frac{1}{6} \\
 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

5.4.3 GenLawson43 — `genlawson43.m`

This scheme requires y_{n-1}, N_{n-2} and N_{n-3} as initial data. At least two steps of an alternative method are needed to start the integration. The overall stiff order is four and for æsthetic reasons the method is broken into the individual matrices.

$$c = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}^T$$

$$A(z) = \begin{bmatrix} & & & \\ \frac{1}{2}\varphi_{1,2} + \frac{3}{8}\varphi_{2,2} + \frac{1}{8}\varphi_{3,2} & & & \\ \frac{1}{2}\varphi_{1,2} + \frac{3}{8}\varphi_{2,2} + \frac{1}{8}\varphi_{3,2} - \frac{15}{16} & \frac{1}{2} & & \\ \varphi_1 + \frac{3}{2}\varphi_2 + \varphi_3 - \frac{15}{8}\varphi_{0,2} & 0 & \varphi_{0,2} & \end{bmatrix}$$

$$U(z) = \begin{bmatrix} 1 & 0 & 0 \\ \varphi_{0,2} & -\frac{1}{4}\varphi_{2,2} - \frac{1}{4}\varphi_{3,2} & \frac{1}{8}\varphi_{2,2} + \frac{1}{8}\varphi_{3,2} \\ \varphi_{0,2} & -\frac{1}{2}\varphi_{2,2} - \frac{1}{4}\varphi_{3,2} - \frac{3}{16} & \frac{1}{8}\varphi_{2,2} + \frac{1}{8}\varphi_{3,2} - \frac{3}{16} \\ \varphi_0 & -\varphi_2 - 2\varphi_3 + \frac{5}{4}\varphi_{0,2} & \frac{1}{2}\varphi_2 + \varphi_3 - \frac{3}{8}\varphi_{0,2} \end{bmatrix}$$

$$B(z) = \begin{bmatrix} \varphi_1 + \frac{3}{2}\varphi_2 + \varphi_3 - \frac{5}{4}\varphi_{0,2} - \frac{1}{2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{6} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$V(z) = \begin{bmatrix} \varphi_0 & -\varphi_2 - 2\varphi_3 + \frac{5}{6}\varphi_{0,2} + \frac{1}{2} & \frac{1}{2}\varphi_2 + \varphi_3 - \frac{1}{4}\varphi_{0,2} - \frac{1}{6} \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

5.5 Modified generalized Lawson schemes

To overcome a loss of stability which some times occurs in the generalized Lawson schemes a modification is suggested in [19]. This modification only affects the solution approximation but significantly improves the accuracy of the schemes. Therefore, the following listing only include the $B(z)$ and $V(z)$ matrices.

Two schemes are included in the package but not listed here due to space reasons, they are `modgenlawson44.m` and `modgenlawson45.m`.

5.5.1 ModGenLawson41 — `modgenlawson41.m`

This scheme of stiff order two but performs significantly better than its counterpart GenLawson1.

$$B(z) = \begin{bmatrix} \varphi_1 - \varphi_2 - \frac{1}{3}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \varphi_2 - \frac{1}{2}\varphi_{0,2} \end{bmatrix}$$

$$V(z) = \begin{bmatrix} \varphi_0 \end{bmatrix}$$

5.5.2 ModGenLawson42 — `modgenlawson42.m`

This scheme of stiff order three but performs significantly better than its counterpart GenLawson2. It requires y_{n-1} and N_{n-2} to be passed from step to step.

$$B(z) = \begin{bmatrix} \varphi_1 - 2\varphi_3 - \frac{1}{2}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{2}\varphi_2 + \varphi_3 - \frac{1}{4}\varphi_{0,2} \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$V(z) = \begin{bmatrix} \varphi_0 & \frac{1}{2}\varphi_2 + \varphi_3 - \frac{1}{4}\varphi_{0,2} \\ 0 & 0 \end{bmatrix}$$

5.5.3 ModGenLawson43 — `modgenlawson43.m`

This scheme of stiff order four and does not have stability problems for large values of the timestep. It requires y_{n-1}, N_{n-2} and N_{n-3} to be passed from step to step.

$$B(z) = \begin{bmatrix} \varphi_1 + \frac{1}{2}\varphi_2 - 2\varphi_3 - 3\varphi_4 - \frac{5}{8}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{3}\varphi_{0,2} & \frac{1}{3}\varphi_2 + \varphi_3 + \varphi_4 - \frac{5}{24}\varphi_{0,2} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$V(z) = \begin{bmatrix} \varphi_0 & -\varphi_2 + \varphi_3 + 3\varphi_4 + \frac{5}{24}\varphi_{0,2} & \frac{1}{6}\varphi_2 - \varphi_4 - \frac{1}{24}\varphi_{0,2} \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

References

[1] H. Berland, B. Owren, and B. Skaflestad. B-series and order conditions for exponential integrators. *SIAM J. Numer. Anal.*, 2005. To appear.

[2] H. Berland, B. Skaflestad, and W. Wright. Expint — A Matlab package for exponential integrators. 2005. Submitted to ACM Transactions on Mathematical Software.

[3] H. Berland, B. Skaflestad, and W. Wright. Expint — A Matlab package for exponential integrators. Technical Report 4/05, Department of Mathematical Sciences, NTNU, Norway, 2005. http://www.math.ntnu.no/preprint/.

[4] G. Beylkin, J. M. Keiser, and L. Vozovoi. A new class of time discretization schemes for the solution of nonlinear PDEs. *J. of Comp. Phys.*, 147:362–387, 1998.

[5] J. C. Butcher. *Numerical methods for ordinary differential equations.* John Wiley & Sons Ltd., Chichester, UK, 2003.

[6] E. Celledoni, A. Marthinsen, and B. Owren. Commutator-free Lie group methods. *FGCS*, 19(3):341–352, 2003.

[7] S. M. Cox and P. C. Matthews. Exponential time differencing for stiff systems. *J. Comput. Phys.*, 176(2):430–455, 2002.

[8] B. L. Ehle and J. D. Lawson. Generalized Runge–Kutta processes for stiff initial-value problems. *J. Inst. Maths. Applics.*, 16:11–21, 1975.

[9] A. Friedli. Verallgemeinerte Runge–Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme. In *Numerical treatment of differential equations*, pages 35–50. Lecture Notes in Math., Vol. 631. Springer, Berlin, 1978.

[10] M. Hochbruck and A. Ostermann. Explicit exponential Runge–Kutta methods for semilinear parabolic problems. *SIAM J. Numer. Anal.*, 43(3):1069–1090, 2005.

[11] A.L Islas and C.M Schober. On the preservation of phase space structure under multisymplectic discretization. *J. of Comp. Phys.*, 197:585–609, 2004.

[12] A.-K. Kassam. *High Order Timestepping for Stiff Semilinear Partial Differential Equations.* PhD thesis, University of Oxford, 2004.

[13] A.-K. Kassam and L. N. Trefethen. Fourth-order time-stepping for stiff PDEs. *SIAM J. Sci. Comput.*, 26(4):1214–1233 (electronic), 2005.

[14] S. Krogstad. Generalized integrating factor methods for stiff PDEs. *J. of Comp. Phys.*, 203(1):72–88, 2005.

[15] J. D. Lawson. Generalized Runge–Kutta processes for stable systems with large Lipschitz constants. *SIAM J. Numer. Anal.*, 4:372–380, 1967.

[16] D. R. Mott, E. S. Oran, and B. van Leer. A quasi-steady-state solver for stiff ordinary differential equations of reaction kinetics. *J. of Comp. Phys.*, 164:407–428, 2000.

[17] H. Munthe-Kaas. High order Runge–Kutta methods on manifolds. In *Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations (Golden, CO, 1997)*, volume 29,1, pages 115–127, 1999.

[18] S. P. Nørsett. An A-stable modification of the Adams–Bashforth methods. In *Conf. on Numerical Solution of Differential Equations (Dundee, Scotland, 1969)*, pages 214–219. Springer, Berlin, 1969.

[19] A. Ostermann, M. Thalhammer, and W. M. Wright. A class of explicit exponential general linear methods. In preparation, 2005.

[20] K. Strehmel and R. Weiner. B-convergence results for linearly implicit one step methods. *BIT*, 27:264–281, 1987.

[21] K. Strehmel and R. Weiner. *Linear-implizite Runge–Kutta-Methoden und ihre Anwendung*, volume 127 of *Teubner-Texte zur Mathematik.* B. G. Teubner Verlagsgesellschaft mbH, Stuttgart, 1992.

[22] M. Thalhammer and W. M. Wright. Generalised lawson methods. Submitted to IMA J. Numer. Anal., 2005.

[23] J. G. Verwer and M. van Loon. An evaluation of explicit pseudo-steady-state approximation schemes for stiff ODE systems from chemical kinetics. *J. of Comp. Phys.*, 113:347–352, 1994.