

# Filtered Leapfrog Time Integration with Enhanced Stability Properties

Ari Aluthge, Scott A. Sarra, Roger Estep

Department of Mathematics, Marshall University, Huntington, WV, USA  
Email: sarra@marshall.edu

Received 27 May 2016; accepted 25 July 2016; published 28 July 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

The Leapfrog method for the solution of Ordinary Differential Equation initial value problems has been historically popular for several reasons. The method has second order accuracy, requires only one function evaluation per time step, and is non-dissipative. Despite the mentioned attractive properties, the method has some unfavorable stability properties. The absolute stability region of the method is only an interval located on the imaginary axis, rather than a region in the complex plane. The method is only weakly stable and thus exhibits computational instability in long time integrations over intervals of finite length. In this work, the use of filters is examined for the purposes of both controlling the weak instability and also enlarging the size of the absolute stability region of the method.

## Keywords

Leapfrog Method, Computational Mode, Weak Stability, Method of Lines, Numerical Weather Prediction

---

## 1. Introduction

In this work the stability properties of a well known linear multistep method (LMM) are examined. The method is based on a three point polynomial finite difference approximation of the first derivative. The Leapfrog method (also known as the explicit midpoint rule or Nyström's method) for the first order Ordinary Differential Equation (ODE) initial value problem (IVP)

$$y' = F[t, y(t)], \quad y(0) = y^0 \quad (1)$$

is a simple three-level scheme with second order accuracy. The leapfrog scheme has a long history in applications. The applications include general oceanic circulation models and atmospheric models that go back at least

as far as the pioneering work of Richardson [1] (p. 150) in 1922 where it was called the *step-over* method. In this context, the time integration method is used to advance in time a semi-discrete system in which the space derivatives of a partial differential equation have been discretized by an appropriate method. Such an application is called the *method of lines*. The leapfrog method remains relevant today as it is a component of many numerical weather forecasting systems and most current geophysical fluid dynamics (GFD) codes use a form of leapfrog time differencing. Reference [2] lists 25 major GFD codes currently using this approach. The recent textbook [3] (p. 90) describes the leapfrog method as an important method for numerical weather forecasting.

The Leapfrog method is efficient as it only requires one function evaluation per time step, with a function evaluation referring to the function  $F$  in Equation (1). The leapfrog scheme has the desirable property of being non-dissipative, but it suffers from phase errors. The most serious drawback of the scheme is its well documented [4] weak stability property which results in computational instability when it is used in long time integrations. Methods that have been used to control the weak instability of the leapfrog method involve a process called filtering. A *Filtering* process replaces a time level with an average of that time level and neighboring time levels. The use of filtering in numerical ODE methods originated in [5] with a two-level averaging. Filtering methods that have been used to control the weak instability of the leapfrog method include the following. The Asselin time filter [6] adds a small artificial viscosity to the leapfrog method and is applied each time step. Gragg [7] used a three-point symmetric filter to replace a time level of the leapfrog solution after every  $N$  steps. Bulirsch and Stoer [8] extended Gragg's method to build an extrapolation scheme. In [9] filtering was shown not to be necessary in extrapolation schemes involving the leapfrog method since the Euler starting step and the extrapolation process alone have been able to control the weak instability. Five point symmetric and asymmetric filters were considered by Iri [10] who recommended the use of an asymmetric filter. Stabilizing the Leapfrog method continues to be an active research area. Examples of recent work in the area can be found in [11] and [12].

In the next section LMMs and their stability properties are reviewed. Then three and five point filters are developed and it is shown that a five-point symmetric filter can be effectively used to control the weak instability that is inherent to the leapfrog method. After the five-point symmetric is shown to be effective, ways to incorporate the filter into a leapfrog time differencing schemes are examined. The application of the filter leads to new LMMs as well as various filter and restart algorithms. The ideal application of the filter will control the weak instability as well as retain the favorable features of the method: efficiency, second order accuracy, an absolute stability region with imaginary axis converge, and will be non-dissipative. Finally, the conclusions are illustrated with numerical examples.

## 2. Linear Multistep Methods

A general  $s$ -step linear multistep method for the numerical solution of the ODE IVP (1) is of the form

$$\sum_{m=0}^s \alpha_m y^{n+m} = k \sum_{m=0}^s \beta_m F[t^{n+m}, y^{n+m}], \quad n = 0, 1, \dots \quad (2)$$

where  $\alpha_m$  and  $\beta_m$  are given constants and  $k$  is the size of the time step. It is conventional to normalize (2) by setting  $\alpha_s = 1$ . When  $\beta_s = 0$  the method is explicit. Otherwise it is implicit. In order to start the method, the first  $s-1$  time levels need to be calculated by a one-step method such as a Runge-Kutta method.

The stability of ODE methods is a relatively new concept in numerical methods and was rarely if ever used before the 1950's [13]. The first description of numerical instability of numerical ODE methods may have been in 1950 in [14]. It was the advent of high speed electronic computers that brought the concept of stability to the forefront. Varying definitions of and types of stability for ODE methods have been proposed over the last 60 years. Indeed, when surveying the literature and textbooks that have been published over that time period it is not straightforward to grasp what is meant by the stability of a LMM due to the multitude of definitions and the same concepts often being given more than one name.

The stability properties of LMMs are examined by applying the method to the ODE

$$y' = \lambda y, \quad y(0) = y^0 \quad (3)$$

where  $\lambda$  is a complex number with  $\Re(\lambda) \leq 0$ . Equation (3) is commonly known as the Dahlquist test equation. The first concepts of stability result from considering the simple case of  $\lambda = 0$  in (3) for which the solution is the constant  $y^0$ . In this case the application of the LMM (2) to the test equation results in a difference equation

with a general solution of the form

$$y^n = c_1 \zeta_1^n + \dots + c_s \zeta_s^n \tag{4}$$

where the  $\zeta_i$  are the zeros of the first characteristic polynomial

$$\rho(\zeta) = \sum_{m=0}^s \alpha_m \zeta^m \tag{5}$$

of the LMM (2). For a *consistent* method one of the terms, say  $\zeta_i^n$ , approaches the solution of (3) as  $n \rightarrow \infty$  while the other roots correspond to extraneous solutions. The following stability definitions quantify to what extent the computed solutions of (4) remain bounded:

- A method is stable (also called zero stable) if  $\rho$  has a simple root equal to one and if all the other roots of  $\rho$  are less than one in absolute value. The criteria that the roots must satisfy for the method to be stable are called a root condition.
- A weakly stable method has all roots of  $\rho$  satisfying  $|\zeta| \leq 1$  and has at least one root of absolute value one other than the simple root  $\zeta = 1$ . The solutions may remain bounded for a fixed interval of time  $[0, T]$  but will eventually become unbounded as  $t \rightarrow \infty$ .
- A method for which a root of  $\rho$  is greater in magnitude than one is unstable. In this case the solutions will rapidly become unbounded.

A stronger definition of stability can be made by considering non-zero  $\lambda$  in Equation (3). In this case the exact solution of (3) is  $y(t) = y^0 e^{\lambda t}$  which approaches zero as  $t \rightarrow \infty$  since the real part of  $\lambda$  is negative. A method that produces numerical solutions with the same asymptotic behavior, that is  $y^n \rightarrow 0$  as  $t^n \rightarrow \infty$ , for a fixed value of  $k$  is said to be absolutely stable. This type of stability has also been called asymptotic stability, or eigenvalue stability. Application of the LMM (2) to the test equation again results in a difference equation with a general solution of the form (4), but this time  $\zeta_i$  is the zeros of the stability polynomial

$$\rho(\zeta) - z\sigma(\zeta) = 0 \tag{6}$$

where  $z = k\lambda$ ,  $\rho$  is the first characteristic polynomial (5), and  $\sigma$  is the second characteristic polynomial

$$\sigma(\zeta) = \sum_{m=0}^s \beta_m \zeta^m \tag{7}$$

of the LMM (2). The method is absolutely stable for a particular value of  $z$  if the zeros of (6) satisfy the root condition. The set of all numbers  $z$  in the complex plane for which the zeros of (6) satisfy the root condition is called the absolute stability region,  $\mathcal{S}$ , of the method. The boundary of the absolute stability region is found by plotting the root locus curve

$$C = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})} \quad 0 \leq \theta \leq 2\pi \tag{8}$$

that is the ratio of the characteristic polynomials of the method. For  $z$  in the stability region, the numerical solutions exhibit the same asymptotic behavior as does the exact solution of (3). For a linear system of ODEs, a necessary condition for absolute stability is that all of the scaled (by  $k$ ) eigenvalues of the coefficient matrix of the system must lie in the stability region. For nonlinear systems, the system must be linearized and the scaled eigenvalues of the Jacobian matrix of the system must be analyzed at each  $t^n$ . Standard references on numerical ODE methods [4] [15]-[19] can be consulted for more details.

### 3. The Leapfrog Method

The leapfrog method for the system (1) is

$$y^{n+1} = y^{n-1} + 2kF(t^n, y^n). \tag{9}$$

The method is constructed by replacing the time derivative in (1) by the second order accurate central difference approximation

$$y'(t^n) \approx \frac{y^{n+1} - y^{n-1}}{2k} \tag{10}$$

of the first derivative. The method needs two time levels to start,  $y^0$  which is provided by the initial condition and  $y^1$ . Typically, the time level  $y^1$  is obtained by an explicit Euler step

$$y^1 = y^0 + kF(t^0, y^0). \tag{11}$$

The characteristic polynomials of the leapfrog method are

$$\rho(\zeta) = z^2 - 1 \quad \text{and} \quad \sigma(\zeta) = 2z. \tag{12}$$

Obviously the roots of  $\rho$  are  $\zeta = \pm 1$  and the method is only weakly stable. The boundary of the absolute stability region is

$$C = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})} = \frac{e^{i\theta} - e^{-i\theta}}{2} = i \sin \theta$$

and the absolute stability region only consists of the interval  $\mathcal{S} = [-i, i]$  on the imaginary axis.

### 4. Filter Construction

It is shown in [20] that the solution of the IVP (1) obtained by the leapfrog method using starting values only assumed to have an asymptotic expansion of the very general form

$$\eta_n(k) = y^n + \sum_{m=2}^3 \eta_{mn} k^m + \mathcal{O}(k^4), \quad n = 0, 1 \tag{13}$$

has an asymptotic expansion of the form

$$y^n = \text{physical mode } (y_p) + \text{computational } (y_c) + \mathcal{O}(k^4). \tag{14}$$

The physical mode  $y_p$

$$y_p = y^0 e^{\lambda t} + \left[ -\frac{\lambda^3}{6} y^0 t + \frac{1}{2} (\eta_{20} + \eta_{21}) e^{\lambda t} \right] k^2 + \left\{ \frac{1}{2} \left[ \frac{\lambda^3}{6} y^0 - \lambda \eta_{20} + \eta_{30} + \eta_{31} \right] e^{\lambda t} \right\} k^3 \tag{15}$$

consists of the exact solution  $y^0 e^{\lambda t}$  and a  $\mathcal{O}(k^2)$  truncation error that goes to zero as  $k \rightarrow 0$ . The computational mode

$$y_c = \left\{ \frac{(-1)^n}{2} [\eta_{20} - \eta_{21}] e^{-\lambda t} \right\} k^2 + \left\{ \frac{(-1)^n}{2} \left[ \frac{-\lambda^3}{6} y^0 + \lambda \eta_{20} + \eta_{30} + \eta_{31} \right] e^{-\lambda t} \right\} k^3 \tag{16}$$

contributes a non-physical oscillatory factor to the solution which grows in size with increasing  $t$ . References [7] and [20] can be consulted for the details of the derivation of the expressions for  $y_p$  and  $y_c$ .

In order to damp the computational mode while retaining the second order accurate approximation of the physical mode, a filter of the form

$$P(E) = \sum_{j=-r_1}^{r_2} a_j E^j \tag{17}$$

is considered where  $a_j \in \mathbb{R}$ ,  $r_1, r_2 \in \mathbb{Z}_+$ , and  $E$  is the forward shift operator defined by  $Ey(t) = y(t+k)$  in continuous notation and  $Ey^n = y^{n+1}$  in discrete notation. Let  $s = \lambda k$  and then note that

$Ee^{\lambda t} = e^{\lambda(t+k)} = e^{\lambda t} e^{\lambda k} = e^{\lambda t} e^s$ . Similarly,  $E^2 e^{\lambda t} = E(Ee^{\lambda t}) = E(e^{\lambda t} e^s) = e^{\lambda t} (e^s)^2$  and it follows that

$E^j e^{\lambda t} = e^{\lambda t} (e^s)^j$ . Now

$$P(E)e^{\lambda t} = \sum_{j=-r_1}^{r_2} a_j E^j e^{\lambda t} = \sum_{j=-r_1}^{r_2} a_j e^{\lambda t} (e^s)^j = e^{\lambda t} \sum_{j=-r_1}^{r_2} a_j (e^s)^j = e^{\lambda t} P(e^s)$$

and

$$\begin{aligned}
 P(E)te^{\lambda t} &= \sum_{j=-\eta_1}^{\eta_2} a_j E^j te^{\lambda t} = \sum_{j=-\eta_1}^{\eta_2} a_j [(t + jk)e^{\lambda(t+jk)}] \\
 &= e^{\lambda t} \sum_{j=-\eta_1}^{\eta_2} [a_j te^{\lambda t} (e^s)^j + a_j jke^{\lambda t} (e^s)^j] \\
 &= e^{\lambda t} \left[ tP(e^s) + k \frac{d}{ds} P(e^s) \right].
 \end{aligned}$$

Thus, applying the filter (17) to the physical mode (15) results in

$$\begin{aligned}
 \hat{y}_p &= y^0 e^{\lambda t} P(e^s) + \frac{-\lambda^3}{6} y^0 e^{\lambda t} \left[ tP(e^s) + k \frac{\partial}{\partial s} P(e^s) \right] + \frac{1}{2} [\eta_{20} + \eta_{21}] e^{\lambda t} P(e^s) \\
 &\quad + \frac{1}{2} \left[ \frac{\lambda^3}{6} y^0 - \lambda \eta_{20} + \eta_{30} + \eta_{31} \right] e^{\lambda t} P(e^s).
 \end{aligned} \tag{18}$$

and applying the filter to the computational mode gives

$$\begin{aligned}
 \hat{y}_c &= \left\{ \frac{(-1)^n}{2} [\eta_{20} + \eta_{21}] e^{-\lambda t} P(-e^{-s}) \right\} k^2 \\
 &\quad + \left\{ \frac{-1^n}{2} \left[ \frac{-\lambda^3}{6} y^0 + \lambda \eta_{20} + \eta_{30} - \eta_{31} \right] e^{-\lambda t} P(-e^{-s}) \right\} k^3.
 \end{aligned} \tag{19}$$

Equation (18) suggests that in order for the physical mode to be kept unchanged up to the order  $k^N$  terms that  $P$  must satisfy

$$P(e^s) = 1 + \mathcal{O}(s^{N+1}). \tag{20}$$

In a similar manner Equation (19) reveals that in order for the computational mode to be reduced by a factor of  $k^M$  that  $P$  must satisfy

$$P(-e^{-s}) = \mathcal{O}(s^M). \tag{21}$$

### 4.1. Three Point Filter

First, filters of length three are considered. A three point symmetric filter is of the form

$$P_3^{(0)}(E) = a_{-1}E^{-1} + a_0 + a_1E^1. \tag{22}$$

The filter coefficients can be found by equating the coefficients of (22) with the coefficients from Equation (20) and Equation (21). This can be done by setting  $N = 1$  and  $M = 1$  for which the first terms in the expansions of (20) and (21) respectively become

$$P(e^s) = 1 + \mathcal{O}(s^2) \tag{23}$$

and

$$P(-e^{-s}) = \mathcal{O}(s). \tag{24}$$

Additionally

$$\begin{aligned}
 P(e^s) &= a_{-1}(e^s)^{-1} + a_0 + a_1(e^s)^1 \\
 &= a_{-1}(1-s) + a_0 + a_1(1+s) + \mathcal{O}(s^2)
 \end{aligned} \tag{25}$$

where on the second line  $e^s$  has been replaced by its linear Taylor series expansion. Equating the constant terms (23) and (25) results in the equation

$$a_{-1} + a_0 + a_1 = 1 \tag{26}$$

and equating the coefficients of  $s$  gives

$$-a_{-1} + a_1 = 0. \tag{27}$$

Similarly

$$\begin{aligned} P(-e^{-s}) &= a_{-1}(-e^{-s})^{-1} + a_0 + a_1(-e^{-s})^1 \\ &= -a_{-1} + a_0 - a_1 + \mathcal{O}(s) \end{aligned} \tag{28}$$

where  $-e^{-s}$  has been replaced by its Taylor series truncated after the constant term. Equating the constant terms (24) and (28) results in the equation

$$-a_{-1} + a_0 - a_1 = 0. \tag{29}$$

The linear system consisting of Equations (26), (27), and (29) has the solution  $a_{-1} = \frac{1}{4}$ ,  $a_0 = \frac{1}{2}$ , and  $a_1 = \frac{1}{4}$ . Thus, the three-point symmetric filter is

$$P_3^{(0)} = \frac{1}{4}(E^{-1} + 2 + E)$$

or equivalently

$$\hat{y}^n = \frac{1}{4}(y^{n-1} + 2y^n + y^{n+1}). \tag{30}$$

Three point filters alter the accurate physical mode by a factor of  $k$  and also damp the computational mode by a factor of  $k$ .

The coefficients of non-symmetric filters  $P_{(3)}^{-1} = \sum_{j=-2}^0 a_j E^j$  and  $P_3^{(1)} = \sum_{j=0}^2 a_j E^j$  can be found in a similar manner. Their coefficients are listed in [Table 1](#).

### 4.2. Five-Point Filter

The consideration of a five point filter presents two more degrees of freedom that can be used to develop a filter that alters the physical mode by one less power of  $k$  and that damps the computational mode by one more power of  $k$  in comparison to a three point filter. That is, in the asymptotic expansion of the physical mode  $N = 2$  and in the asymptotic expansion of the computational mode  $M = 2$ .

A five point symmetric filter is of the form

$$P_5^{(0)}(E) = a_{-2}E^{-2} + a_{-1}E^{-1} + a_0 + a_1E^1 + a_2E^2. \tag{31}$$

In this case the first few terms in the expansions of (20) and (21) respectively are

$$P(e^s) = 1 + \mathcal{O}(s^3) \tag{32}$$

and

$$P(-e^{-s}) = \mathcal{O}(s^2). \tag{33}$$

Additionally

**Table 1.** Three point filter coefficients.

	$4a_{-2}$	$4a_{-1}$	$4a_0$	$4a_1$	$4a_2$
$P_3^{(-1)}$	-1	2	3	-	-
$P_3^{(0)}$	-	1	2	1	-
$P_3^{(1)}$	-	-	3	2	-1

$$\begin{aligned}
 P(e^s) &= a_{-2}(e^s)^{-2} + a_{-1}(e^s)^{-1} + a_0 + a_1(e^s)^1 + a_2(e^s)^2 \\
 &= a_{-2}\left(1 - 2s + \frac{(-2s)^2}{2}\right) + a_{-1}\left(1 - s + \frac{(-s)^2}{2}\right) + a_0
 \end{aligned} \tag{34}$$

$$+ a_1\left(1 + s + \frac{s^2}{2}\right) + a_2\left(1 + 2s + \frac{(2s)^2}{2}\right) + \mathcal{O}(s^3) \tag{35}$$

where this time  $e^s$  has been replaced by its Taylor series expansion.

Equating the constant terms (32) and (34) results in the equation

$$a_{-2} + a_{-1} + a_0 + a_1 + a_2 = 1 \tag{36}$$

and then equating the coefficients of  $s$  gives

$$-2a_{-2} - a_{-1} + a_1 + 2a_2 = 0 \tag{37}$$

and finally equating the coefficients of  $s^2$  gives

$$2a_{-2} \frac{1}{2} a_{-1} + \frac{1}{2} a_1 + 2a_2 = 0. \tag{38}$$

Similarly

$$\begin{aligned}
 P(-e^{-s}) &= a_{-2}(-e^{-s})^{-2} + a_{-1}(-e^{-s})^{-1} + a_0 + a_1(-e^{-s})^1 + a_2(-e^{-s})^2 \\
 &= a_{-2}(1 + 2s) + a_{-1}(1 + s) + a_0 + a_1(1 - s) + a_2(1 - 2s) + \mathcal{O}(s^2)
 \end{aligned} \tag{39}$$

where  $-e^{-s}$  has been replaced by its Taylor series expansion. Equating the constant and linear terms in (33) and (39) results in the equations

$$a_{-2} - a_{-1} + a_0 - a_1 + a_2 = 0 \tag{40}$$

and

$$2a_{-2} - a_{-1} + a_1 - 2a_2 = 0. \tag{41}$$

The linear system consisting of Equations (36)-(38), (40), and (41) has the solution  $a_{-2} = \frac{-1}{16}$ ,  $a_{-1} = \frac{1}{4}$ ,  $a_0 = \frac{10}{16}$ ,  $a_1 = \frac{1}{4}$ , and  $a_2 = \frac{-1}{16}$ . Thus, the five-point symmetric filter is

$$P_5^{(0)} = \frac{1}{16}(-E^{-2} + 4E^{-1} + 10 + 4E - E^2)$$

or

$$\hat{y}^n = \frac{1}{16}(-y^{n-2} + 4y^{n-1} + 10y^n + 4y^{n+1} - y^{n+2}). \tag{42}$$

The coefficients of non-symmetric filters  $P_5^{(-2)} = \sum_{j=4}^0 a_j E^j$ ,  $P_5^{(-1)} = \sum_{j=3}^1 a_j E^j$ ,  $P_5^{(1)} = \sum_{j=-1}^3 a_j E^j$ , and

$P_5^{(2)} = \sum_{j=0}^4 a_j E^j$  can be found in a similar manner. Their coefficients are listed in **Table 2**.

The advantages of the five point filter when compared to the three point filter are now clear. The five point filter retains the second order accuracy of the leapfrog method and damps the computational mode by a factor of  $k^2$ . In contrast, the three point filter reduces the accuracy of the method to first order and only damps the computational mode by a factor of  $k$ . It remains to find the best way to apply the filter. In the following sections the filter is applied to derive new LMMs and various filter and restart algorithms are analyzed.

**Table 2.** Five point filter coefficients.

	$16a_{-4}$	$16a_{-3}$	$16a_{-2}$	$16a_{-1}$	$16a_0$	$16a_1$	$16a_2$	$16a_3$	$16a_4$
$P_5^{(-2)}$	3	-4	-6	12	11	-	-	-	-
$P_5^{(-1)}$	-	-1	0	6	8	3	-	-	-
$P_5^{(0)}$	-	-	-1	4	10	4	-1	-	-
$P_5^{(1)}$	-	-	-	3	8	6	0	-1	-
$P_5^{(2)}$	-	-	-	-	11	12	-6	-4	3

## 5. New LMMs

Incorporating the filter in each leapfrog step by averaging  $y^{n-1}$  with  $P_5^{(0)}$  results in the new LMM

$$\begin{aligned}
 y^{n+1} &= y^{n-1} + 2kF(t^n, y^n) \\
 &= \frac{1}{16}(-y^{n-3} + 4y^{n-2} + 10y^{n-1} + 4y^n - y^{n+1}) + 2kF(t^n, y^n) \\
 &= \frac{1}{16}(-y^{n-3} + 4y^{n-2} + 10y^{n-1} + 4y^n - [y^{n-1} + 2kF(t^n, y^n)]) + 2kF(t^n, y^n)
 \end{aligned}$$

or

$$y^{n+1} = \frac{1}{4}y^n + \frac{9}{16}y^{n-1} + \frac{1}{4}y^{n-2} - \frac{1}{16}y^{n-3} + \frac{15}{8}kF(t^n, y^n) \quad (43)$$

This approach has the advantage that the result may be analyzed within the LMM framework. The characteristic polynomials of the method are  $\rho(z) = z^4 - \frac{1}{4}z^3 - \frac{9}{16}z^2 - \frac{1}{4}z + \frac{1}{16}$  and  $\sigma(z) = \frac{15}{8}z^3$ . The first characteristic polynomial has  $\zeta = 1$  as a root and three other roots less than one in magnitude and thus the method is zero-stable. The absolute stability region of the method is shown in **Figure 1**. The maximum stability ordinate along the imaginary axis is approximately 0.87 and along the real axis  $-0.53$ . The new method is second order accurate.

Averaging  $y^{n-1}$  with  $P_3^{(0)}$  in each leapfrog step results in the new LMM

$$y^{n+1} = \frac{1}{4}y^n + \frac{1}{2}y^{n-1} + \frac{1}{4}y^{n-2} + 2hF(t^n, y^n). \quad (44)$$

The roots of the first characteristic polynomial of the method are  $\zeta = 1$  and two other roots with magnitude one-half. The method is zero-stable but only first order accurate. The maximum stability ordinate along the imaginary axis is approximately  $3/4$  and along the real axis  $-1/2$ .

A strong point of the leapfrog method, especially in the method of lines integration of wave propagation type PDEs, is that it is non-dissipative. The phase and amplitude errors of the two new LMMs are compared with the phase and amplitude errors of the leapfrog method in **Figure 2**. The  $P_5^{(0)}$  LMM has phase errors are similar to those of the leapfrog method and is by far the least dissipative of the two filtered methods. The  $P_3^{(0)}$  LMM suffers from larger phase errors than does the leapfrog method and is very dissipative. The effects of the dispersion and dissipation errors are illustrated in example 7.3.

## 6. $P_5^{(0)}$ Filter Application

In this section, four filter and restart algorithms are explored. For each option the effects on accuracy and stability properties are examined. The options include:

- Method 1 (M1). Filter after every step.
- Method 2 (M2). Filter followed by Euler restart every  $N$  steps.
- Method 3 (M3). Alternative starting/restarting-algorithm 1.
- Method 4 (M4). Alternative starting/restarting-algorithm 2.



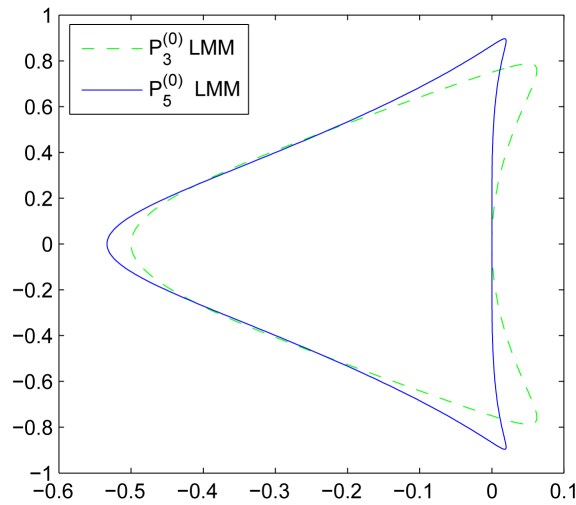


Figure 1. Stability regions for the (solid line)  $P_5^{(0)}$  LLM (43) and (dashed line)  $P_3^{(0)}$  LLM (44).

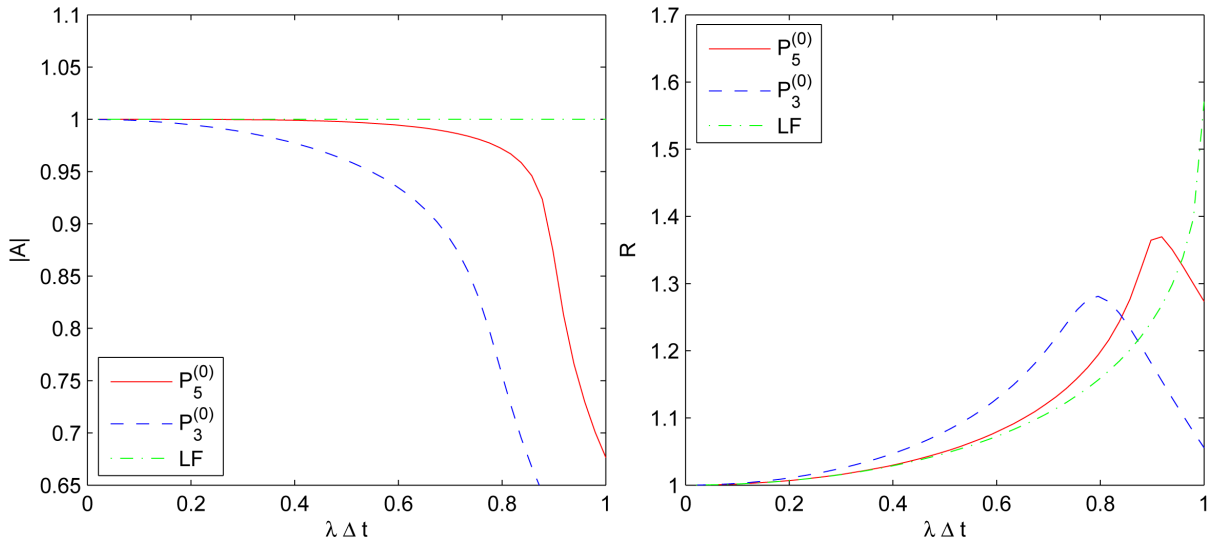


Figure 2. Amplitude and phase errors of the leapfrog method (9), the  $P_5^{(0)}$  LLM (43), and the  $P_3^{(0)}$  LLM (44).

To allow for the absolute stability region to be examined, all four algorithms have a start up procedure following by multiple steps of leapfrog and/or filtering. Then after a fixed number,  $N$ , of steps the algorithm is restarted with the start up procedure. In each of the four algorithms  $N = 20$  or  $N = 21$  has been used. In most cases, increasing  $N$  results in the method having an absolute stability region with less coverage of the left half plane and with more imaginary axis coverage. The desired filtering algorithm will maintain second order accuracy, retain as much of the stability interval on the imaginary axis as possible, and extend the absolute stability region into the left half plane.

### 6.1. Filter after Every Step (M1)

The first approach uses an explicit Euler step (11) to calculate the first time level. Each subsequent step takes one leapfrog step to the time level  $n$  plus two additional leapfrog steps to calculate time levels  $n + 1$  and  $n + 2$  that are needed by the filter. Then the  $P_5^{(0)}$  filter is applied and time level  $n$  is replaced with the average. Time levels  $n + 1$  and  $n + 2$  are disregarded and then the process is repeated until time level  $N$  is reached. At that time the method is restarted using time level  $N$  as the initial condition. The M1 approach retains second order accuracy but requires three function evaluations per time step.

The absolute stability region of the M1 approach with  $N = 20$  is shown in **Figure 3**. In the left image of the figure the stability region is seen to cover significant area in the left half plane and appears to retain coverage of the portion of the imaginary axis over the interval  $[-i, i]$ . However, the right image of the figure zooms in on the imaginary axis and reveals that a large portion of the interval is not included in the stability region.

The absolute stability regions of methods M1, M2, M2, and M4 are found in the following manner. For example, the stability region of M1 with  $N = 3$  is found by applying the method to the test problem (3). Then the amplification factor  $R(z)$  of the method is found by looking at

$$\begin{aligned} 2y^1 &= y^0 + k\lambda y^0 = (1+z)y^0 \\ y^2 &= y^0 + 2k\lambda y^1 = (1+2z+2z^2)y^0 \\ y^3 &= y^1 + 2k\lambda y^2 = (1+3z+4z^2+4z^3)y^0 \\ y^4 &= y^2 + 2k\lambda y^3 = (1+4z+8z^2+8z^3+8z^4)y^0 \\ y^5 &= y^3 + 2k\lambda y^4 = (1+5z+10z^2+20z^3+16z^4+15z^5)y^0 \end{aligned}$$

and

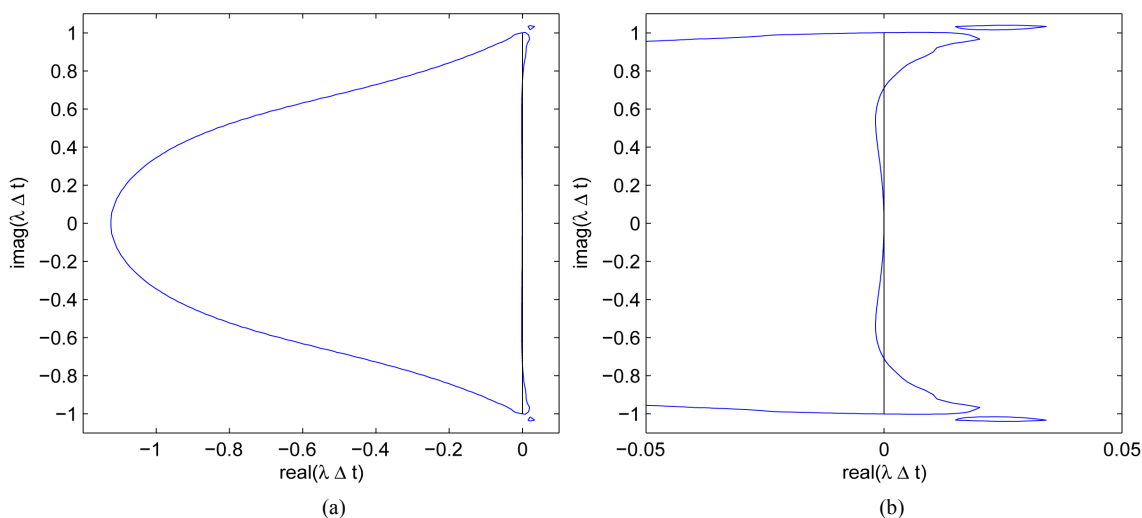
$$\begin{aligned} y^3 &= \frac{1}{16}(-y^1 + 4y^2 + 10y^3 + 4y^4 - y^5)y^0 \\ &= \left(1 + 3z + \frac{17}{4}z^2 + \frac{13}{4}z^3 + z^4 - z^5\right)y^0 \\ &= R(z)y^0. \end{aligned}$$

The absolute stability region contains all the  $z \in \mathcal{C}$  for which  $|R(z)| \leq 1$ .

## 6.2. Filter, Euler Restart Every $N$ Steps (M2)

Method M2 approximates the first time level with Euler's method (11) and then advances in time with  $N + 2$  leapfrog steps. Then the  $P_5^{(0)}$  filter is applied and the value of  $y^N$  is assigned to be the filtered average. Then the method is restarted with the initial condition  $y^N$ . Thus the method is

$$\begin{aligned} y^1 &= y^0 + kF[t^0, y^0] \\ y^n &= u^{n-2} + 2kF[t^{n-1}, y^{n-1}], \quad n = 2, 3, \dots, N+2 \\ y^N &= P_5^{(0)}(y^{N-2}, y^{N-1}, y^N, y^{N+1}, y^{N+2}). \end{aligned} \tag{45}$$



**Figure 3.** (a): Absolute stability region of method M1. (b): close up of the stability region along the imaginary axis.

The M2 approach remains second order accurate and requires  $(N + 2)/N$  function evaluations per time step. In the given example an average of 1.1 function evaluation per time step are needed. The main drawback of approach M2 is that the stability region (Figure 4) lacks any coverage along the imaginary axis. Method M2 is unstable for all values of  $z > 0$  for which leapfrog method is absolutely stable.

### 6.3. Alternative Start/Restart-Algorithm 1 (M3)

The absolute stability region of Euler’s method is a circle of radius one that is centered at  $z = -1$  in the complex plane. The region does not include any purely complex numbers. Calculating the first step of method M2 with Euler’s method is causing the method’s stability region to lack imaginary axis coverage.

The following starting procedure

$$\begin{aligned} \bar{y}^1 &= y^0 + \frac{k}{M} F[t^0, y^0] \\ \bar{y}^n &= \bar{y}^{n-2} + 2\frac{k}{M} F[t^0 + k/M, \bar{y}^{n-1}], \quad n = 2, \dots, M \\ y^1 &= \bar{y}^M \end{aligned} \tag{46}$$

is suggested for minimizing this effect. To advance from the initial condition to time level one,  $M$  smaller substeps are taken consisting of one Euler step followed by  $M - 1$  leapfrog steps. After the first step, algorithm M3 is identical to algorithm M2. The absolute stability region of method M3 with  $N = 20$  and  $M = 4$  is shown in Figure 5. The proposed starting procedure has improved the imaginary axis coverage of the stability region to approximately the interval  $[-0.5i, 0.5i]$ , or about half the size of the leapfrog methods stability interval. Method M3 requires  $(N + M + 1)/N$  function evaluations per time step, which in the given example is equal to 1.25. Method M4 in the next section makes one more modification to further increase the coverage of the stability region along the imaginary axis.

### 6.4. Alternative Start/Restart Algorithm 2 (M4)

Algorithm M4 uses the same starting procedure (46) as does M3 but it does not terminate and restart after filtering at time step  $N$ . Instead time levels  $N - 1$  and  $N$  are filtered and then  $N$  more leapfrog steps are taken. The filtering and continuing process is done  $C$  times before restarting. At the end of each continuation step the final time level is filtered. The region of absolute stability for the M4 algorithm with  $M = 4$ ,  $C = 2$ , and  $N = 7$  (a total of 21 time steps in order to approximate the 20 time steps used in the M1, M2, and M3 examples) is in Figure 6. The imaginary axis coverage of the M4 stability region is nearly equal to that of the leapfrog method,

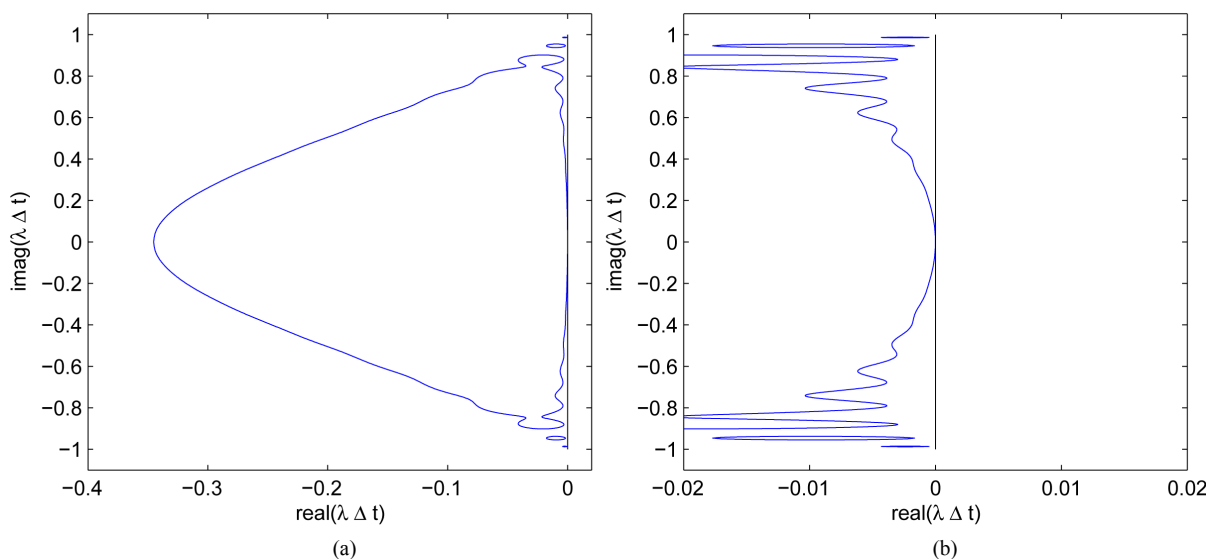


Figure 4. (a): Absolute stability region of method M2. (b): close up of the stability region along the imaginary axis.

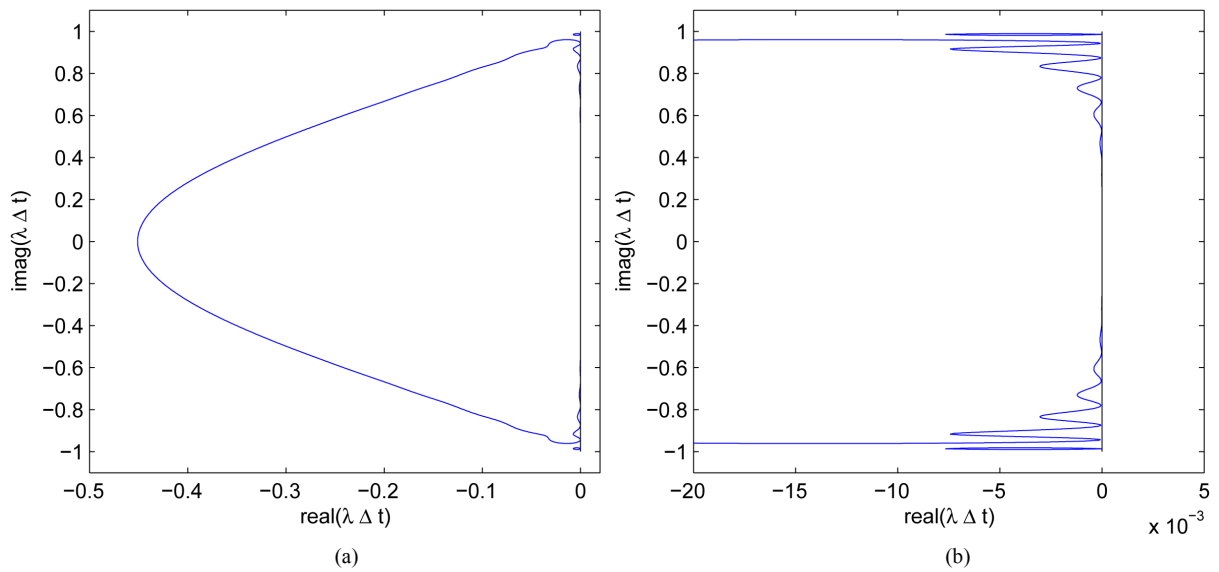


Figure 5. (a): Absolute stability region of method M3. (b): close up of the stability region along the imaginary axis.

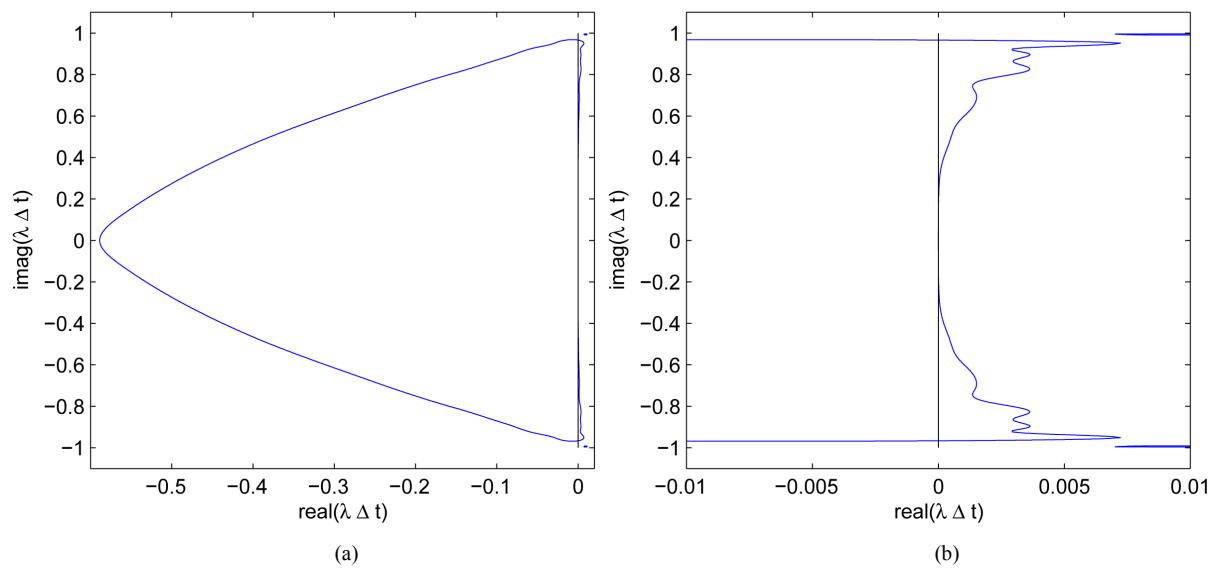


Figure 6. (a): Absolute stability region of method M4. (b): close up of the stability region along the imaginary axis.

covering approximately the interval  $[-0.95i, 0.95i]$ . The M4 algorithm requires an average of  $(N + M + 1 + C \times N) / ((C + 1)N)$  function evaluations per time step which is approximately 1.24 in the example given. Matlab source code that implements method M4 is available on the website of the second author at <http://www.scottsarra.org/math/math.html>.

## 7. Numerical Results

### 7.1. Example 1

The purpose of the first example is to demonstrate the ability of the proposed methods to control the computational mode that is introduced by the leapfrog weak instability. The example considers the nonlinear IVP

$$y' = 1 - y^2, \quad y(0) = 0, \tag{47}$$

which has the exact solution  $y(t) = \tanh(t)$ . A time step size of  $k = 0.1$  is used with all methods. Figure 7

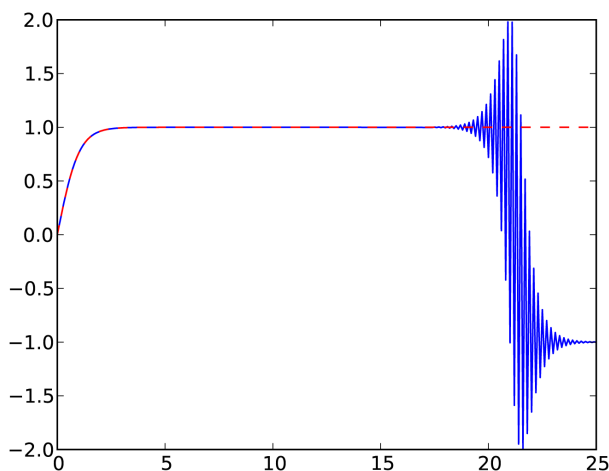
illustrated behavior typical of a weakly stable method. Over the finite time interval  $[0,18]$  the leapfrog method gives a good approximation to the true solution. **Table 3** reports the solution having five digits of accuracy at  $t = 5$ . However, after approximately  $t = 18$ , non-physical oscillations appear in the solution and eventually the solution becomes unbounded.

**Table 3** records the results of applying the two new LMMs from section 2 and as well as using the four filter and restart algorithms. Additionally, the five point asymmetric filter  $P_5^{(-2)}$  that was recommend in [10] is used. By  $t = 25$  and later, all seven of the filtered method have either zero or near zero error. The numerical solutions nearly exactly approximate the exact solution which becomes the constant  $y = 1$  as  $t \rightarrow \infty$ . In this example, all seven methods have effectively damped the computational mode and prevented the numerical solution from becoming infinite. In five of the methods, graphically non-physical oscillations are not observed. However, in methods  $P_5^{(-2)}$  and M2 slight oscillations (as in **Figure 8**) appear in the numerical solution prior to the computational mode being damped sufficiently.

### 7.2. Example 2

This example numerically calculates the order of convergence of the leapfrog method and the seven filtered methods. The test Equation (3) is used with  $\lambda = -1$  and with the initial condition  $y(0) = 1$ . Each method is repeated on the time interval  $[0,1]$  using the sequence of number of time steps

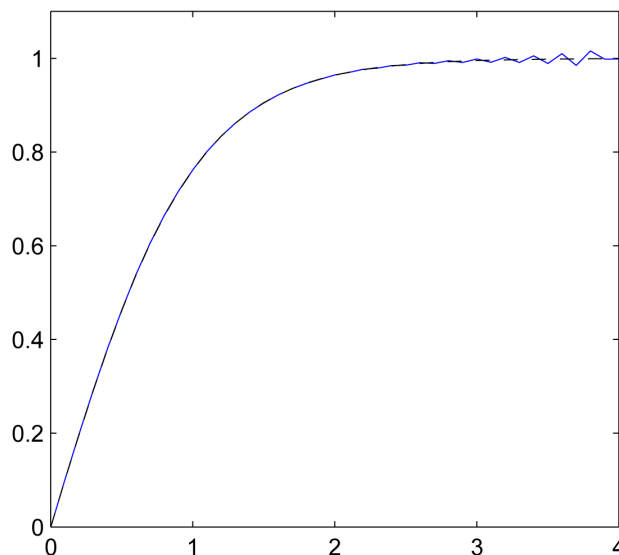
$N = 20, 40, 80, 160, 320, 640, 1280, 2560, 5120$ . The resulting convergence plots are shown in the left and right image of **Figure 9**. On a log-log plot of the absolute value of the error versus the time step size the plot will be a straight line with a slope that is the order of convergence of the method. The LMM3, Asym  $P_5^{(-2)}$ -M2, M1, and



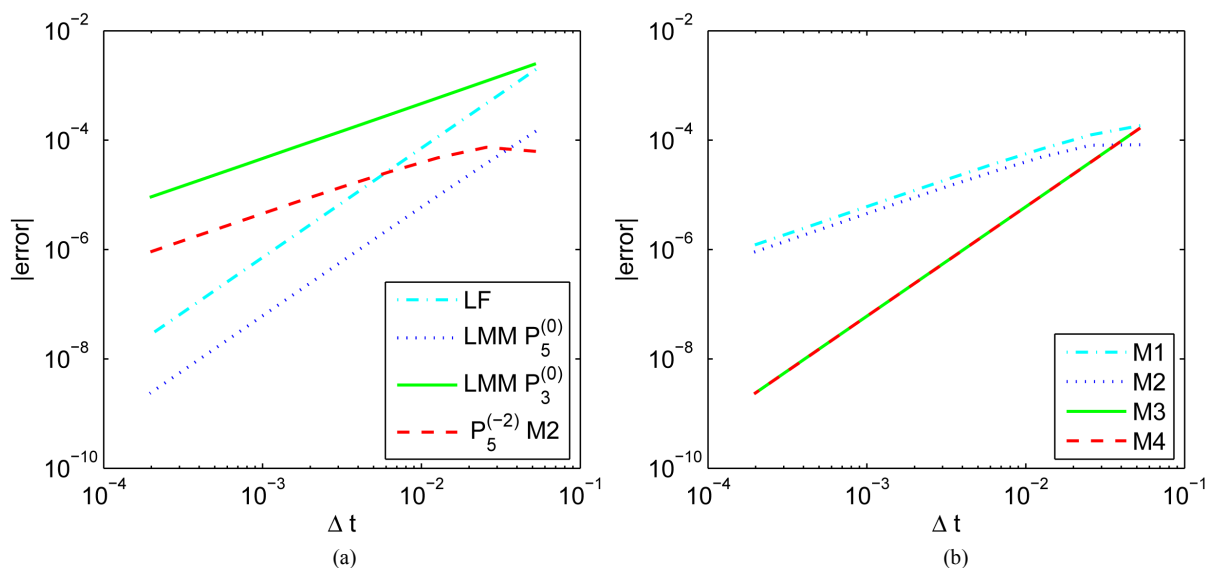
**Figure 7.** Exact (dashed) and leapfrog (solid) solution of problem (47) with  $k = 0.1$ .

**Table 3.** Errors from problem (47).

	$t = 5$	$t = 25$	$t = 100$
leapfrog	4.2e-6	3.0	$\infty$
$P_3^{(0)}$ LMM	2.9e-5	0	0
$P_5^{(0)}$ LMM	4.2e-6	3.3e-16	3.3e-16
M1	1.3e-6	2.2e-16	2.2e-16
M2	7.8e-5	0	0
M3	1.7e-7	0	0
M4	2.6e-6	0	0
$P_5^{(-2)}$ M2	4.5e-5	2.2e-16	2.2e-16



**Figure 8.** Oscillations appear in the asymmetric  $P_5^{(-2)}$  and M2 method solutions prior to the computational mode being damped.



**Figure 9.** Convergence plots. (a): LF, LMM3, LMM5,  $P_5^{(-2)}$ -M2. (b): M1, M2, M3, M4. The M3 and M4 plots are visually nearly identical.

M2 methods are only first order accurate while the LMM5, M3, and M4 methods retain the second order accuracy of the leapfrog method. In this example, the three second order accurate filtered methods are about a decimal place more accurate for a given value of  $k$  than is the leapfrog method.

### 7.3. Advection

The purpose of the third example is to give an illustration of the effects of the dispersion and dissipation properties of the two LMM described in Section 2 and to illustrate the effectiveness of the M4 algorithm as a method of lines integrator of hyperbolic PDEs. A model hyperbolic PDE problem is the advection equation

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = 0. \tag{48}$$

The domain for the problem is the interval  $[0,1]$  and periodic boundary conditions are enforced. The initial condition is taken from the exact solution  $u(x,t) = \sin^{40}(\pi(x+t))$ .

The space derivative of (48) is discretized with the Fourier Pseudospectral method [21] [22]. A relatively large number of points,  $N = 179$ , are used to discretize the space interval so that the space errors will be negligible. After discretizing in space the linear system of ODEs

$$u_t = Du \tag{49}$$

remains to be advanced in time where  $D$  is the  $N \times N$  Fourier differentiation matrix [23]. The eigenvalues of  $D$  are purely imaginary requiring an ODE method with a stability region having imaginary axis coverage. The four methods in this work that are applicable are the leapfrog method, the  $P_5^{(0)}$  LMM, the  $P_3^{(0)}$  LMM, and methods M3 and M4. However, the imaginary axis coverage of the stability region of the M3 is small compared to that of M4 so the M3 method is not applied.

Runs with two different time step sizes are made. The first with  $k = 0.45/\lambda$  and the second with  $k = 0.7/\lambda$ , where  $\lambda$  is the magnitude of the largest eigenvalue of  $D$ . With the first time step the  $P_5^{(0)}$  LMM is practically non-dissipative while the second is near the stability limit for the  $P_3^{(0)}$  LMM. As analyzed in section 6.4, the M4 method is absolutely stable for time step sizes as large as  $k = 0.95/\max(ew)$  for the advection equation. With the maximum stable time step size the error at  $t=100$  is  $3.7e-1$ . The results at  $t=10$  are shown in Figure 10 and additional results are in Table 4 and Table 5.

### 7.4. Advection-Diffusion

Discretized advection-diffusion operators do not have purely imaginary eigenvalues and thus the leapfrog method is not an appropriate method of lines integrator for this type of problem. Advection-diffusion problems in which the advection term is dominant will have differentiation matrices with eigenvalues that have negative real parts that are of small to moderate size. Algorithm M4 can be effectively applied in such a problem.

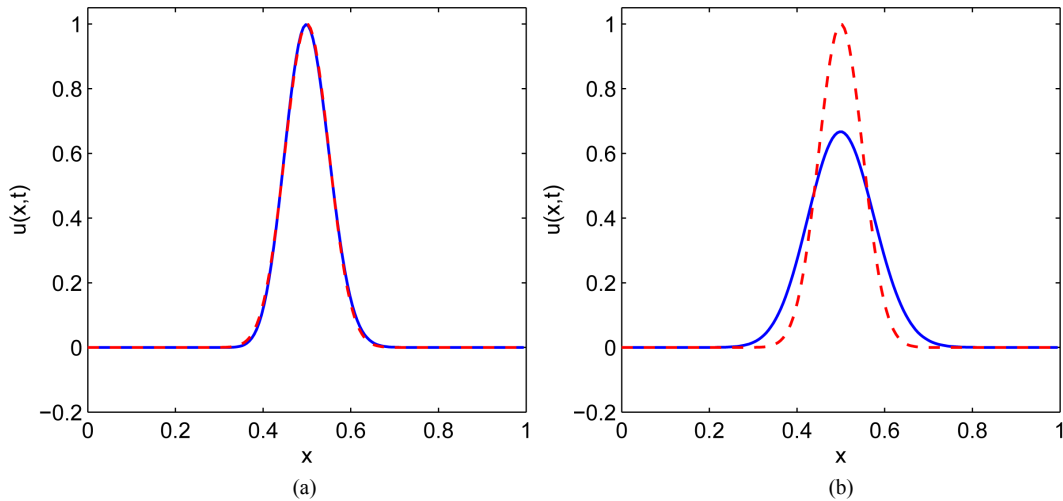


Figure 10. Solutions of problem (48) at  $t=10$  (exact dashed, numerical solid) with  $k = 0.7/\max(ew)$ : (a):  $P_5^{(0)}$  LMM; (b):  $P_3^{(0)}$  LMM.

Table 4. Errors from problem (48),  $k = 0.7/\max(ew)$ .

	$t = 1$	$t = 10$	$t = 100$
leapfrog	$1.5e-2$	$1.5e-2$	$2.3e-1$
$P_3^{(0)}$ LMM	$5.7e-2$	$3.3e-1$	$7.3e-1$
$P_5^{(0)}$ LMM	$1.4e-2$	$1.5e-2$	$2.2e-1$
M4	$2.8e-3$	$2.8e-2$	$2.5e-1$

As an example, the advection diffusion equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \tag{50}$$

where  $\nu \ll 1$  is considered. The viscosity coefficient is taken as  $\nu = 0.001$  and periodic boundary conditions are applied on the interval  $[-5, 5]$ . The space derivatives are discretized with the Fourier Pseudospectral method with 199 evenly spaced points. The scaled spectrum of the discretized space derivative operators and the stability region of method M4 is shown in the left image of **Figure 11**. An initial condition of

$$u(x, 0) = \begin{cases} 1 & -1 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

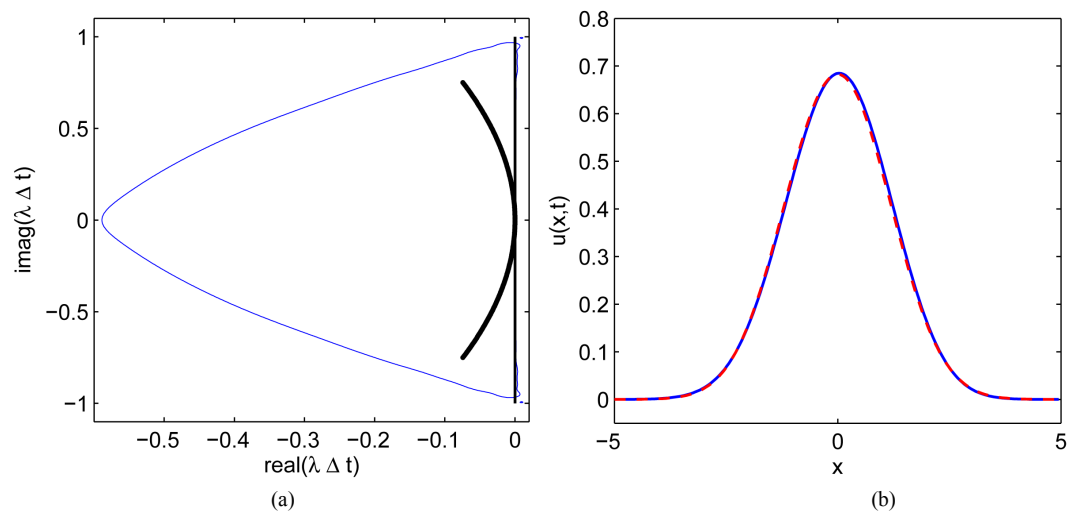
is used and the problem is advanced in time to  $t = 500$  using a time step of  $k = 0.75/\lambda$ , where  $\lambda$  is the maximum magnitude of the imaginary part of the eigenvalues of the discretized operator. In the right image of **Figure 11** the numerical solution at  $t = 500$  is compared with the exact solution. The maximum error is  $1.3e-2$ . The results illustrate the ability of method M4 to accurately advance advection dominated problems over long time intervals.

### 8. Conclusion

Filters of length three and five have been examined for the purpose of controlling the weak instability of the leapfrog method. Filtering the leapfrog method with a three point filter reduces the accuracy of the overall scheme to first order. If a five point filter is used, it is possible for the method to retain the second order accuracy as well as damp the computational mode by a factor of  $k^2$ . Several ways of incorporating the filter into the leapfrog method have been examined. Applying the five-point symmetric filter to the time level  $u^{n-1}$

**Table 5.** Errors from problem (48),  $k = 0.45/\max(e_w)$ .

	$t = 1$	$t = 10$	$t = 100$
leapfrog	$1.2e-2$	$7.9e-3$	$1.1e-1$
$P_3^{(0)}$ LMM	$3.9e-2$	$2.5e-1$	$6.7e-1$
$P_5^{(0)}$ LMM	$1.5e-2$	$1.0e-2$	$8.5e-2$
M4	$1.1e-3$	$1.1e-2$	$1.2e-1$



**Figure 11.** (a): Stability region of method M4 and the scaled spectrum of the discretized problem (50); (b): Numerical solution (solid) vs. exact (dashed) at  $t = 500$ .



of the leapfrog method (9) results in a new second order accurate LMM (43) which is zero stable and that has an absolute stability region that includes both coverage of a portion of the imaginary axis as well as a region in the left half plane. Of the four filter and restart algorithms described, algorithm M4 is the most attractive. The M4 method, which restarts after every 21 time steps, has an absolute stability region that includes nearly as much of the imaginary axis as does the leapfrog method as well as includes a region in the left half plane. This allows the method to be an effective method of lines integrator for pure advection and advection dominated problems as is illustrated in the examples. In addition to the second order accuracy, the method also retains the computational efficiency of the leapfrog method. Method M4 is a potential replacement for existing leapfrog type time differencing schemes that are used in geophysical fluid dynamics codes.

## References

- [1] Richardson, L.F. (1922) *Weather Prediction by Numerical Process*. Cambridge University Press, Cambridge.
- [2] Williams, P. (2009) A Proposed Modification to the Robert-Asselin Time Filter. *Monthly Weather Review*, **137**, 2538-2546. <http://dx.doi.org/10.1175/2009MWR2724.1>
- [3] Coiffier, J. (2011) *Fundamentals of Numerical Weather Prediction*. Cambridge University Press, Cambridge. <http://dx.doi.org/10.1017/CBO9780511734458>
- [4] Butcher, J. (2003) *Numerical Methods for Ordinary Differential Equations*. Wiley. <http://dx.doi.org/10.1002/0470868279>
- [5] Milne, W. and Reynolds, R. (1959) Stability of a Numerical Solutions of Differential Equations. *Journal of the ACM*, **6**, 196-203. <http://dx.doi.org/10.1145/320964.320976>
- [6] Asselin, R. (1972) Frequency Filter for Time Integrations. *Monthly Weather Review*, **100**, 487-490. [http://dx.doi.org/10.1175/1520-0493\(1972\)100<0487:FFFTI>2.3.CO;2](http://dx.doi.org/10.1175/1520-0493(1972)100<0487:FFFTI>2.3.CO;2)
- [7] Gragg, W.B. (1963) *Repeated Extrapolation to the Limit in the Numerical Solution of Ordinary Differential Equations*. PhD Thesis, UCLA.
- [8] Bulirsch, R. and Stoer, J. (1966) Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods. *Numerische Mathematik*, **8**, 1-13. <http://dx.doi.org/10.1007/BF02165234>
- [9] Shampine, L. and Baca, L. (1983) Smoothing the Extrapolated Midpoint Rule. *Numerische Mathematik*, **41**, 165-175.
- [10] Iri, M. (1963) A Stabilizing Device for Unstable Numerical Solutions of Ordinary Differential Equations—Design Principle and Application of a Filter. *Journal of the Information Processing Society of Japan*, **4**, 249-260.
- [11] Li, Y. and Trenchea, C. (2014) A Higher-Order Robert-Asselin Type Time Filter. *Journal of Computational Physics*, **259**, 23-32. <http://dx.doi.org/10.1016/j.jcp.2013.11.022>
- [12] Norton, T. and Hill, A. (2015) An Iterative Starting Method to Control Parasitism for the Leapfrog Method. *Applied Numerical Mathematics*, **87**, 145-156. <http://dx.doi.org/10.1016/j.apnum.2014.09.005>
- [13] Dahlquist, G. (1985) 33 Years of Numerical Instability, Part I. *BIT*, 188-204. <http://dx.doi.org/10.1007/BF01934997>
- [14] Todd, J. (1950) Solution of Differential Equations by Recurrence Relations. *Mathematical Tables and Other Aids to Computation*, **4**, 39-44. <http://dx.doi.org/10.2307/2002701>
- [15] Hairer, E., Norsett, S. and Wanner, G. (2000) *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer.
- [16] Hairer, E. and Wanner, G. (2000) *Solving Ordinary Differential Equations II: Stiff and Differential—Algebraic Problems*. 2nd Edition, Springer.
- [17] Henrici, P. (1962) *Discrete Variable Methods in Ordinary Differential Equations*. Wiley.
- [18] Iserles, A. (1973) *A First Course in the Analysis of Differential Equations*. Wiley.
- [19] Lambert, J. (1973) *Computational Methods in Ordinary Differential Equations*. Wiley.
- [20] Aluthge, A. (1985) *Filtering and Extrapolation Techniques in the Numerical Solution of Ordinary Differential Equations*. Master's Thesis, University of Ottawa, Ottawa.
- [21] Boyd, J.P. (2000) *Chebyshev and Fourier Spectral Methods*. 2nd Edition, Dover Publications, Inc., New York.
- [22] Canuto, C., Hussaini, M., Quarteroni, A. and Zang, T. (1988) *Spectral Methods in Fluid Dynamics*. Springer. <http://dx.doi.org/10.1007/978-3-642-84108-8>
- [23] Weideman, J.A.C. and Reddy, S. (2000) A MATLAB Differentiation Matrix Suite. *ACM Transactions on Mathematical Software*, **26**, 465-519. <http://dx.doi.org/10.1145/365723.365727>