

# A general linear method approach to the design and optimization of efficient, accurate, and easily implemented time-stepping methods in CFD

Victor DeCaria<sup>a,1,\*</sup>, Sigal Gottlieb<sup>b</sup>, Zachary J. Grant<sup>a,1</sup>, William J. Layton<sup>c</sup>

<sup>a</sup>*Oak Ridge National Laboratory, Computational and Applied Mathematics Group*

<sup>b</sup>*Mathematics Department, University of Massachusetts Dartmouth*

<sup>c</sup>*Department of Mathematics, University of Pittsburgh*

---

## Abstract

In simulations of fluid motion time accuracy has proven to be elusive. We seek highly accurate methods with strong enough stability properties to deal with the richness of scales of many flows. These methods must also be easy to implement within current complex, possibly legacy codes. Herein we develop, analyze and test new time stepping methods addressing these two issues with the goal of *accelerating the development of time accurate methods addressing the needs of applications*. The new methods are created by introducing inexpensive pre-filtering and post-filtering steps to popular methods which have been implemented and tested within existing codes. We show that pre-filtering and post-filtering a multistep or multi-stage method results in new methods which have both multiple steps and stages: these are general linear methods (GLMs). We utilize the well studied properties of

---

\*Corresponding author

*Email addresses:* `decariavp@ornl.gov` (Victor DeCaria), `sgottlieb@umassd.edu` (Sigal Gottlieb), `grantzj@ornl.gov` (Zachary J. Grant), `wjl@pitt.edu` (William J. Layton)

<sup>1</sup>This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

GLMs to understand the accuracy and stability of filtered method, and to design optimal new filters for popular time-stepping methods. We present several new embedded families of high accuracy methods with low cognitive complexity and excellent stability properties. Numerical tests of the methods are presented, including ones finding failure points of some methods. Among the new methods presented is a novel pair of alternating filters for the Implicit Euler method which induces a third order, A-stable, error inhibiting scheme which is shown to be particularly effective.

*Keywords:* Navier-Stokes, general linear methods, time discretization, time filter

---

## 1. Introduction

There is significant cognitive complexity required for understanding, implementing and validating new methods in complex, possibly legacy, codes. This results in a need for improved methods that can be easily implemented through small modifications of simpler and well-tested codes. Herein, we develop high order timestepping methods with favorable stability properties that can be implemented by adding a minimal code modification of a few,  $O(2)$ , extra lines to simple methods often used in legacy codes. The newly developed methods often do not require additional function evaluations or extra storage, as the variables are simply over-written. Some of these methods provide an embedded error estimator, have natural extensions to variable timesteps and arise from a process, Section 3.3, that is amenable to optimization with respect to applications driven design criteria.

To motivate the rest of the paper and provide useful methods, we now give three (constant  $\Delta t$ ) examples of the new methods as time discretizations of the incompressible Navier–Stokes equations (NSE) (the application used to test the methods in Section 5). The NSE are

$$u_t + u \cdot \nabla u - \nu \Delta u + \nabla p = f \text{ and } \nabla \cdot u = 0. \quad (1)$$

Here,  $u$  is the velocity,  $p$  is the pressure,  $\nu$  is the kinematic viscosity, and  $f = f(x, t)$  is an external non-autonomous body force. The two equations describe the conservation of momentum and mass, respectively. We set  $f = 0$  for simplicity in these examples.

**Example 1:** Beginning with the usual fully implicit Euler (IE) method, let  $\Delta t$  denote the timestep and superscript the timestep number. Suppressing

the spatial discretization, the IE method for the Navier-Stokes equations (NSE) is:

$$\frac{u^{n+1} - u^n}{\Delta t} + u^{n+1} \cdot \nabla u^{n+1} - \nu \Delta u^{n+1} + \nabla p^{n+1} = 0 \text{ and } \nabla \cdot u^{n+1} = 0. \quad (2a)$$

Adding two extra lines of code yields the following new method:

- 1) Pre-filter  $\tilde{u}^n = u^n - \frac{1}{2} (u^n - 2u^{n-1} + u^{n-2}),$
- 2) IE Solve:  $\begin{cases} \frac{1}{\Delta t} u^{n+1} + u^{n+1} \cdot \nabla u^{n+1} - \nu \Delta u^{n+1} + \nabla p^{n+1} = \frac{1}{\Delta t} \tilde{u}^n \\ \nabla \cdot u^{n+1} = 0 \end{cases}$
- 3) Post-filter  $u_{3rd}^{n+1} = u^{n+1} - \frac{5}{11} (u^{n+1} - 3u^n + 3u^{n-1} - u^{n-2}),$

The method arising by stopping after Step 2 is second order accurate and L-stable, Section 4.1.2, and is referred to herein as IE-Pre-2. The method after Step 3 is third order accurate and  $A(\alpha)$  stable with  $\alpha \simeq 71^\circ$ , Section 4.1.3, and is referred to as IE-Pre-Post-3. Thus the difference between the Steps 2 and 3 approximation is an estimator of the local truncation error<sup>2</sup>. We stress that the pre and post-filters are themselves comprehensible (not exotic) operations and reduce the discrete fluctuation of the numerical solution.

Steps 2 and 3 are time filters designed to have no effect on (some realization of) smooth solution scales and damp (some realization of) fluctuating scales. For example, for the prefilter  $\tilde{u}^n = u^n - \frac{1}{2} (u^n - 2u^{n-1} + u^{n-2})$ , if  $u^n = a + bt^n$  then the filter does not alter  $u^n$ . Let  $\kappa = u^n - 2u^{n-1} + u^{n-2}$  denote the discrete curvature, then the filter also has the effect of halving the discrete curvature:  $\kappa_{post} = \frac{1}{2} \kappa_{pre}$ . The post-filter is a higher order realization of this process. The coefficient values, 1/2 and 5/11, are derived by applying the order conditions in the general linear method induced by the pre and post filtered method, Section 3.

**Example 2:** One equivalent realization<sup>3</sup> of the usual implicit midpoint

---

<sup>2</sup>The pair of approximations allows implementation as a variable order method, not developed herein.

<sup>3</sup>The next steps would be reorganized for a different implementation of the method.

(1-1 Padé, Crank-Nicolson, one-leg trapezoidal) method is: given  $u^n$

$$\left\{ \begin{array}{l} \frac{1}{\frac{1}{2}\Delta t} u^{n+\frac{1}{2}} + u^{n+\frac{1}{2}} \cdot \nabla u^{n+\frac{1}{2}} - \nu \Delta u^{n+\frac{1}{2}} + \nabla p^{n+\frac{1}{2}} = \frac{1}{\frac{1}{2}\Delta t} u^n, \\ \nabla \cdot u^{n+\frac{1}{2}} = 0, \end{array} \right. \quad (3a)$$

$$u^{n+1} = 2u^{n+\frac{1}{2}} - u^n. \quad (3b)$$

We abbreviate this as MP. The pre- and post-filtered methods in Section 4.2 require introducing temporary variables  $\tilde{u}^n, \tilde{u}^{n+1}$  (that can be overwritten each time step) and produces an embedded triplet of approximations of 2nd, 3rd and 4th order  $u_{2\text{nd}}^{n+1}, u_{3\text{rd}}^{n+1}, u_{4\text{th}}^{n+1}$ . Naturally one must be selected to be  $u^{n+1}$ . The method is

$$\begin{array}{l} 1) \text{ Pre-filter} \quad \tilde{u}^n = u^n + \frac{5}{6} \left( u^n - \frac{3}{2} u^{n-1} + \frac{3}{5} u^{n-2} - \frac{1}{10} u^{n-3} \right) \\ 2) \text{ IM Solve} \quad \left\{ \begin{array}{l} \frac{1}{\frac{1}{2}\Delta t} u^{n+\frac{1}{2}} + u^{n+\frac{1}{2}} \cdot \nabla u^{n+\frac{1}{2}} - \nu \Delta u^{n+\frac{1}{2}} + \nabla p^{n+\frac{1}{2}} = \frac{1}{\frac{1}{2}\Delta t} \tilde{u}^n, \\ \nabla \cdot u^{n+\frac{1}{2}} = 0, \\ u^{n+1} = 2u^{n+\frac{1}{2}} - \tilde{u}^n \end{array} \right. \end{array}$$

$$\begin{array}{l} 3) \text{ Post-filters} \quad u_{3\text{rd}}^{n+1} = \frac{1}{2} (u^{n+1} + \tilde{u}^n) \\ u_{2\text{nd}}^{n+1} = \frac{12}{11} u_{3\text{rd}}^{n+1} - \frac{7}{22} u^n + \frac{9}{22} u^{n-1} - \frac{5}{22} u^{n-2} + \frac{1}{22} u^{n-3}, \\ u_{4\text{th}}^{n+1} = \frac{24}{25} u_{3\text{rd}}^{n+1} + \frac{4}{25} u^n - \frac{6}{25} u^{n-1} + \frac{4}{25} u^{n-2} - \frac{1}{25} u^{n-3}. \end{array}$$

The 2nd order approximation, MP-Pre-Post-2, is  $A$ -stable, and has the same linear stability region as MP. The 3rd order approximation, MP-Pre-Post-3, is  $A(\alpha)$  with  $\alpha = 79.4^\circ$  while the fourth order approximation, MP-Pre-Post-4, is  $A(\alpha)$  with  $\alpha = 70.64^\circ$ , Section 4.2. Their differences provide an estimator, which makes MP-Pre-Post-2/3/4 potentially useful as the basis for a variable stepsize variable order (VSVO) method. Other examples are provided building on BDF2 (Section 4.3) and Runge-Kutta methods (Section 4.4). The above is the natural implementation without requiring additional function evaluations.

**Example 3:** Looking at the pre- and post-filtering process as a general linear method (GLM) also opens the door to creating methods that have the error inhibiting properties described in [EIS20]. An example of this is the

EIS method IE-EIS-3 which we will describe in 4.1.4:

$$\begin{aligned}
1) \text{ Pre-filter} & \quad \tilde{u}^n = \frac{23}{5}u^{n-\frac{1}{3}} - 3u^n - \frac{9}{5}\tilde{u}^{n-1} + \frac{6}{5}\hat{u}^{n-1} \\
2) \text{ IE Solve:} & \quad \begin{cases} \frac{1}{\Delta t}u^{n+\frac{2}{3}} + u^{n+\frac{2}{3}} \cdot \nabla u^{n+\frac{2}{3}} - \nu \Delta u^{n+\frac{2}{3}} + \nabla p^{n+\frac{2}{3}} = \frac{1}{\Delta t}\tilde{u}^n \\ \nabla \cdot u^{n+\frac{2}{3}} = 0 \end{cases} \\
3) \text{ Filter} & \quad \hat{u}^n = \frac{5}{12}u^n - \frac{1}{12}u^{n+\frac{2}{3}} - \frac{5}{12}\hat{u}^{n-1} + \frac{13}{12}\tilde{u}^n \\
4) \text{ IE Solve:} & \quad \begin{cases} \frac{1}{\Delta t}u^{n+1} + u^{n+1} \cdot \nabla u^{n+1} - \nu \Delta u^{n+1} + \nabla p^{n+1} = \frac{1}{\Delta t}\hat{u}^n \\ \nabla \cdot u^{n+1} = 0 \end{cases}
\end{aligned}$$

This method is third order and A-stable, and has the cost of two implicit Euler evaluations.

**Method design and analysis.** In the methods' derivation and implementation, extra variables are introduced. In their analysis and design the extra variables are condensed to obtain an equivalent single method. With pre and post filtering, this equivalent method can be both multi-step and multi-stage. Thus its optimization and analysis must be through the theory of general linear methods, Section 2. Section 3 shows how this theory can be used to design filters so that the induced method satisfies desired optimality conditions. In Section 3 we will show how to rewrite time-filtering methods as general linear methods, and give some examples of such time-filtered methods as GLMs. In Section 3.3 we will show how this formulation can be used to develop an optimization code that is a powerful tool that allows us to find pre- and post- processed methods with advantageous stability properties. We will demonstrate the utility of this approach showing some methods that resulted from this optimization code (Section 4). Section 4 applies Section 2 and 3 to derive the methods in the section above and several more based on design criteria of stability, accuracy and an embedded algorithmic structure. These new methods are embedded families of high order methods based on combinations of pre- and post-filtering for the most commonly used methods for time discretization of incompressible flows, including the fully implicit Euler method, the midpoint rule, and the BDF2 method. (We stress however that the theory can be applied to a wide variety of other applications driven design criteria.) The tools developed in Sections 2 and 3 thus show *how to accelerate the development of time accurate methods addressing the needs of applications*. Finally, we study the performance of some of these methods in Section 5. The higher Reynolds flow problems in Section 5 were selected because they are nonlinearity dominated and have solutions rich in

scales, so accuracy is difficult and stability is essential.

### 1.1. Related work

Time filters are widely used to stabilize leapfrog time discretizations of atmosphere models, e.g., [R69], [A72], [W09]. Our study of this important work, in particular its stress on balancing computational, space and cognitive complexities, was critical for the development path herein. In [GL18] it was shown a well calibrated post-filter can increase accuracy in the fully implicit method to second order, preserve A-stability, anti-diffuse the approximation and yield an error estimator useful for time adaptivity. The new methods and analysis in [GL18] for  $y' = f(t, y)$  were extended to the Navier-Stokes equations in [DLZ18]. In [DGLL18] post-filters were studied starting with BDF3 (yielding an embedded family of orders 2,3,4). It was also proven that post-filtering alone has an accuracy barrier: improvement by at most one power of  $\Delta t$  is possible from any sequence of linear post-filters. The idea of adding a prefilter step herein is simple in principle (though technically intricate in analysis and design, Sections 2,3) and overcomes this accuracy barrier.

There is a significant body of detailed and technically intricate stability and convergence analysis for CFD problems of, mostly simpler (e.g., IE, IM, BDF2) and mostly constant timestep methods, including [G79], [CR78], [BDK82], [JMRT17], [E04a], [E04b], [J15]. Important work on adaptive timestepping for similar problems occurs in [KGG10], [VV13], [HEPG15], [JR10]. The embedded structure of the new method families herein suggest their further development into adaptive methods.

## 2. Examples: Time Filters induce General Linear Methods (GLMs)

Writing the method induced by adding pre- and post-filters in the form of a GLM allows us to apply the order conditions and stability theory of GLMs to optimize the method. To illustrate how a GLM is induced, for

$$u_t = F(u)$$

we consider the pre- and post-filtered implicit Euler method (Example 1 in Section 1) which can be written in the form

$$y^{(1)} = u^n - \frac{1}{2} (u^n - 2u^{n-1} + u^{n-2}) \quad (4a)$$

$$y^{(2)} = y^{(1)} + \Delta t F(y^{(2)}) \quad (4b)$$

$$u^{n+1} = y^{(2)} - \frac{5}{11} (y^{(2)} - 3u^n + 3u^{n-1} - u^{n-2}) \quad (4c)$$

GLMs represent any combination of multistep and multistage methods. A GLM with  $s$  stages and  $k$  steps is

$$y^{(i)} = \sum_{\ell=1}^k d_{i\ell} u^{n-k+\ell} + \Delta t \sum_{\ell=1}^{k-1} \hat{a}_{i\ell} F(u^{n-k+\ell}) + \Delta t \sum_{j=1}^s a_{ij} F(y^{(j)}) \quad 1 \leq i \leq s \quad (5a)$$

$$u^{n+1} = \sum_{\ell=1}^k \theta_{\ell} u^{n-k+\ell} + \Delta t \sum_{\ell=1}^{k-1} \hat{b}_{\ell} F(u^{n-k+\ell}) + \Delta t \sum_{j=1}^s b_j F(y^{(j)}), \quad (5b)$$

where the  $u^{n-k+\ell}$  denote the steps, while the  $y_j^n$  are intermediate stages used to compute the next solution value  $u^{n+1}$ . We will refer to these coefficients more compactly by the matrices  $D, A, \hat{A}$  given by

$$D_{i\ell} = d_{i\ell}, \quad \hat{A}_{i\ell} = \hat{a}_{i\ell}, \quad \text{and} \quad A_{ij} = a_{ij},$$

and the vectors  $\Theta, \mathbf{b}, \hat{\mathbf{b}}$  that are given by

$$\Theta_{\ell} = \theta_{\ell}, \quad \hat{\mathbf{b}}_{\ell} = \hat{b}_{\ell}, \quad \text{and} \quad \mathbf{b}_j = b_j.$$

The method (4) is in the form (5) with

$$D = \left( -\frac{1}{2}, 1, \frac{1}{2} \right), \quad \theta = \left( \frac{2}{11}, -\frac{9}{11}, \frac{18}{11} \right), \quad \hat{A} = (0, 0, 0), \quad \hat{b} = (0, 0, 0),$$

and  $A = 1, b_1 = \frac{6}{11}$ . Note that we can write this method as a pair of **embedded second and third order methods**. Only pre-filtering the implicit Euler method yields the  $\mathcal{O}(\Delta t^2)$  accurate<sup>4</sup> method

$$\begin{aligned} y^{(1)} &= -\frac{1}{2} u^{n-2} + u^{n-1} + \frac{1}{2} u^n \\ u_{2nd}^{n+1} &= y^{(1)} + \Delta t F(u_{2nd}^{n+1}). \end{aligned}$$

---

<sup>4</sup>All accuracy statements are proven by checking the order conditions for GLMs presented in Section 3.

Adding a final postprocessing line

$$u_{3rd}^{n+1} = \frac{5}{11}u^{n-2} - \frac{15}{11}u^{n-1} + \frac{15}{11}u^n + \frac{6}{11}u_{2nd}^{n+1} \quad (6)$$

makes the approximation  $\mathcal{O}(\Delta t^3)$ . This embedded structure means that the difference between the velocities  $u_{3rd}^{n+1}$  and  $u_{2nd}^{n+1}$  can be used as an error estimator.

**A 1-parameter family of pre- and post-filtered methods.** Adding a parameter to the pre- and post- filters allows optimization with respect to accuracy, stability and error constant criteria in Section 3. For the implicit Euler method the parametric family is

$$y^{(1)} = u^n - d(u^n - u^{n-1}) \quad (7a)$$

$$y^{(2)} = y^{(1)} + \Delta t F(y^{(2)}) \quad (7b)$$

$$u^{n+1} = y^{(2)} - \frac{1}{3-2d} \{(1-2d)y^{(2)} - 2(1-d)u^n + u^{n-1}\}. \quad (7c)$$

This method can be written as a GLM by

$$\begin{aligned} y^{(1)} &= du^{n-1} + (1-d)u^n \\ y^{(2)} &= y^{(1)} + \Delta t F(y^{(2)}) \\ u^{n+1} &= \frac{2d-1}{3-2d}u^{n-1} + \frac{4(1-d)}{3-2d}u^n + \frac{2}{3-2d}\Delta t F(y^{(2)}). \end{aligned}$$

which is of GLM form with the choices  $A = 1$ ,  $b_1 = \frac{2}{3-2d}$  and

$$D = \begin{pmatrix} d & 1-d \\ d & 1-d \end{pmatrix}, \quad \theta = \left( \frac{2d-1}{3-2d}, \frac{4(1-d)}{3-2d} \right), \quad \hat{A} = (0, 0), \quad \hat{b} = (0, 0).$$

Clearly, it is possible to develop easily implemented methods with many free parameters by pre- and post-filtering widely used methods. Next, in Section 3, we show that the induced GLM form can simplify, automate and accelerate filter, and thus method design.

### 3. Optimizing Time Filters using the GLM framework

Section 2 illustrates that adding two lines of time filter code to commonly used linear multistep methods can increase accuracy without significant additional computational work. To design the filter required, it is necessary

to write the filtered methods in the GLM form (5). This section develops the GLM form of filtered general GLMs, shows how to use this in a method design process and generalizes the filtering process to also include, previously computed, trend values (i.e. values of  $F(y^{(j)})$ ).

Subsection 3.1 shows that pre- and post-filtering *any* GLM, whether a multistep, Runge–Kutta, or a combination of these, induces another GLM whose properties depend on the filter parameters in a precise way. Theorem 1 gives a complete characterization of the coefficients of the filtered GLM that results from pre- and/or post-filtering a core GLM method in terms of the filter parameters and the coefficients of the core method. Subsection 3.2 shows how to determine the accuracy and stability properties of the method so induced. The results in the first three subsections open the possibility, developed in Subsection 3.3, of *optimizing accuracy and stability properties of the filtered GLM over the choices of the filter parameters*. The optimization algorithm in Subsection 3.3 is then used to design methods with favorable stability and accuracy properties. We will present the new methods found using this optimization in Section 4.

### 3.1. Pre and post filtering a GLM

In the section above, we showed how time-filtering a linear multistep method can be written as a GLM. Filtering is also useful for multi-stage (Runge–Kutta) methods and multistep multi-stage methods, GLMs. If the core method is a linear multistep method as in (11) the section above then we have one stage only and  $k$  steps. If the core method is a Runge–Kutta method then we have  $k = 1$  steps and  $s$  stages. In more generality, we consider here a core method with  $s$  stages and  $k$  steps, defined by the coefficients  $\check{d}_{i\ell}, \hat{a}_{i\ell}, a_{ij}$  for  $i = 2, \dots, s, j = 2, \dots, s$  and  $\ell = 1, \dots, k$ :

$$\begin{aligned}
 y^{(1)} &= u^n \\
 y^{(i)} &= \sum_{\ell=1}^k \check{d}_{i\ell} u^{n-k+\ell} + \Delta t \sum_{\ell=1}^{k-1} \hat{a}_{i\ell} F(u^{n-k+\ell}) + \Delta t \sum_{j=1}^s a_{ij} F(y^{(j)}) \quad (8) \\
 &\quad \text{for } 2 \leq i \leq s \\
 u^{n+1} &= y^{(s)}.
 \end{aligned}$$

We can write this in the form (5) where the final row coefficients are the same as the prior row coefficients. In the last theorem we limited the form of the

post-filter to be (12c) and not include any function evaluations. However, the post filter does not have to be limited to the form (12c), but in fact can have  $\theta_i, \hat{b}_i, b_j$  as free parameters, chosen only to satisfy stability and accuracy considerations. This does not impose much additional cost to the method, as any function evaluations have already been computed by the final stage.

The following theorem provides the relationship between the core method, the filter parameters and the filtered method.

**Theorem 1.** *If we filter a GLM of the form (8) we obtain a GLM of the form (5) where the first stage is a pre-filter is given by*

$$y^{(1)} = \sum_{\ell=1}^k d_{1\ell} u^{n-k+\ell} \quad (9a)$$

and the final stage is a post filter given by

$$u^{n+1} = \sum_{\ell=1}^k \theta_{\ell} u^{n-k+\ell} + \Delta t \sum_{\ell=1}^{k-1} \hat{b}_{\ell} F(u^{n-k+\ell}) + \Delta t \sum_{j=1}^s b_j F(y^{(j)}). \quad (9b)$$

The coefficients  $d_{1\ell}$  of (9a) are the pre-filter coefficients, and the coefficients  $\theta_{\ell}, \hat{b}_{\ell}, b_j$  in (9b) are the post-filter coefficients. These do not depend on the coefficients of the core method (8) and can be chosen freely, subject only to order and stability constraints.

The middle stages  $2 \leq i \leq s$  of the filtered method have the form

$$y^{(i)} = \sum_{\ell=1}^k d_{i\ell} u^{n-k+\ell} + \Delta t \sum_{\ell=1}^{k-1} \hat{a}_{i\ell} F(u^{n-k+\ell}) + \Delta t \sum_{j=1}^s a_{ij} F(y^{(j)}),$$

where coefficients  $a_{ij}$  and  $\hat{a}_{i\ell}$  are not impacted by the filtering, and the coefficients  $d_{i\ell}$  of the filtered methods are related to the coefficients  $\check{d}_{i\ell}$  of the core method (8) by

$$d_{i\ell} = \check{d}_{i1} d_{1\ell} + \check{d}_{i\ell} \quad \text{for } \ell = 1, k-1 \quad (10a)$$

$$d_{ik} = \check{d}_{i1} d_{1k}. \quad (10b)$$

**Proof.** Clearly, the pre-filter coefficients  $d_{1\ell}$  in (9a) can be freely chosen, subject only to accuracy and stability considerations. We then use  $y^{(1)}$  instead

of  $u^n$  in all the middle stages of (8):

$$\begin{aligned}
y^{(i)} &= \tilde{d}_{i1}y^{(1)} + \sum_{\ell=1}^{k-1} \tilde{d}_{i\ell}u^{n-k+\ell} + \Delta t \sum_{\ell=1}^{k-1} \hat{a}_{i\ell}F(u^{n-k+\ell}) + \Delta t \sum_{j=1}^s a_{ij}F(y^{(j)}) \\
&= \tilde{d}_{i1} \sum_{\ell=1}^k d_{1\ell}u^{n-k+\ell} + \sum_{\ell=1}^{k-1} \tilde{d}_{i\ell}u^{n-k+\ell} + \\
&\quad + \Delta t \sum_{\ell=1}^{k-1} \hat{a}_{i\ell}F(u^{n-k+\ell}) + \Delta t \sum_{j=1}^s a_{ij}F(y^{(j)}) \\
&= \sum_{\ell=1}^k d_{i\ell}u^{n-k+\ell} + \Delta t \sum_{\ell=1}^{k-1} \hat{a}_{i\ell}F(u^{n-k+\ell}) + \Delta t \sum_{j=1}^s a_{ij}F(y^{(j)}),
\end{aligned}$$

where the coefficients of the pre-processed scheme are related to the original coefficients by:

$$d_{i\ell} = \tilde{d}_{i1}d_{1\ell} + \tilde{d}_{i\ell} \quad \text{for } \ell = 1, k-1, \quad \text{and} \quad d_{ik} = \tilde{d}_{i1}d_{1k}$$

To post-process the method, we simply modify the final line (9b) by allowing the postprocessing coefficients  $\theta_\ell, \hat{b}_\ell, b_j$  to be chosen freely, subject only to order conditions and stability considerations. ■

**Remark 2.** *By placing additional constraints on the post-filter coefficients, methods may be derived where the function evaluations in (9b) are replaced by a linear combination of previously computed  $u^{n-k+\ell}$  and stages  $y^{(j)}$ . This may be desirable depending on the problem, computer architecture, and availability of function evaluations from a possibly black box solver.*

### 3.1.1. Filtering a linear multistep method

Consider a  $k$ -step linear multistep method:

$$u^{n+1} = \sum_{\ell=1}^k \alpha_\ell u^{n-k+\ell} + \Delta t \sum_{\ell=1}^{k+1} \beta_\ell F(u^{n-k+\ell}). \quad (11)$$

We pre-filter (Eqn. (12a)) and post-filter (Eqn. (12c) ) as in Section 2:

$$\hat{u}^n = u^n - \frac{\alpha}{2} \sum_{\ell=1}^k \hat{d}_\ell u^{n-k+\ell} \quad (12a)$$

$$\begin{aligned} \hat{u}^{n+1} &= \alpha_k \hat{u}^n + \sum_{\ell=1}^{k-1} \alpha_\ell u^{n-k+\ell} + \Delta t \sum_{\ell=1}^{k-1} \beta_\ell F(u^{n-m+\ell}) \\ &+ \Delta t \beta_k F(\hat{u}^n) + \Delta t \beta_{k+1} F(\hat{u}^{n+1}) \end{aligned} \quad (12b)$$

$$u^{n+1} = \hat{u}^{n+1} - \frac{\omega}{2} \sum_{\ell=1}^k \hat{q}_\ell u^{n-k+\ell}. \quad (12c)$$

Following Theorem 1, the coefficients of the filtered methods satisfy:

$$d_{1k} = 1 - \frac{\alpha}{2} \hat{d}_k, \quad d_{2k} = \alpha_k d_{1k} \quad \text{and} \quad d_{1\ell} = -\frac{\alpha}{2} \hat{d}_\ell, \quad d_{2\ell} = \alpha_k d_{1\ell} + \alpha_\ell \quad \text{for } \ell < k.$$

$$\hat{a}_{2\ell} = \beta_\ell \quad \text{for } \ell \leq k-1, \quad \text{and} \quad a_{21} = \beta_k, \quad a_{22} = \beta_{k+1}.$$

Generally, the post-filtering coefficients  $\theta_\ell$  and  $\hat{b}_\ell$  for  $\ell \leq k$ , and  $b_j$  for  $j = 1, 2$  can be chosen freely subject only to order and stability considerations. However, if we wish to limit our post-filtering to the form (12c), which does not include function evaluations, then we have

$$\theta_\ell = d_{2\ell} - \frac{\omega}{2} \hat{q}_\ell, \quad \hat{b}_\ell = \hat{a}_{2\ell}, \quad \text{and} \quad b_j = a_{2j}. \quad (13)$$

Notice that in the pre-filter (12a), for consistency  $\sum \hat{d}_\ell u^{n-k+\ell}$  should annihilate polynomials up to a certain (non-zero) degree so that this quantity is related to a discrete derivative. In particular, for  $u^n \equiv 1$  this implies  $\sum_{\ell=1}^k \hat{d}_\ell = 0$ . In general, the quantity  $\sum_{\ell=1}^k \hat{d}_\ell u^{n-k+\ell}$  represents a discrete fluctuation. Pre-filtering acts to reduce the discrete fluctuation of the numerical solution.

**Proposition 3.** *If*

$$0 < \alpha \hat{d}_k < 2,$$

*then the filter*

$$u_{pre}^n = u^n - \frac{\alpha}{2} \sum_{\ell=1}^k \hat{d}_\ell u^{n-k+\ell}$$

strictly reduces without changing sign the discrete fluctuation  $\sum \hat{d}_\ell u^{n-k+\ell}$

$$\left( \hat{d}_k u_{pre}^n + \sum_{\ell=1}^{k-1} \hat{d}_\ell u^{n-k+\ell} \right) = \left( 1 - \frac{\alpha \hat{d}_k}{2} \right) \left( \sum_{\ell=1}^k \hat{d}_\ell u^{n-k+\ell} \right)$$

**Proof.** Multiply by  $\hat{d}_k$  to obtain

$$\hat{d}_k u_{pre}^n = \hat{d}_k u^n - \frac{\alpha \hat{d}_k}{2} \sum_{\ell=1}^k \hat{d}_\ell u^{n-k+\ell}.$$

Add  $\sum_{\ell=1}^{k-1} \hat{d}_\ell u^{n-k+\ell}$  to both sides:

$$\begin{aligned} \left( \hat{d}_k u_{pre}^n + \sum_{\ell=1}^{k-1} \hat{d}_\ell u^{n-k+\ell} \right) &= \left( \hat{d}_k u^n + \sum_{\ell=1}^{k-1} \hat{d}_\ell u^{n-k+\ell} \right) - \frac{\alpha \hat{d}_k}{2} \left( \sum_{\ell=1}^k \hat{d}_\ell u^{n-k+\ell} \right) \\ &= \left( 1 - \frac{\alpha \hat{d}_k}{2} \right) \left( \sum_{\ell=1}^k \hat{d}_\ell u^{n-k+\ell} \right) \end{aligned}$$

which establishes the result. ■

### 3.2. Background on General Linear Methods

To analyze the order and stability of (5) we convert to the compact form

$$y^{(i)} = u^{n-k+i} \quad \text{for } i=1 \dots, k-1 \quad (14a)$$

$$y^{(i)} = \sum_{\ell=1}^k d_{i\ell} u^{n-k+\ell} + \Delta t \sum_{j=1}^{k+s} \tilde{A}_{ij} F(y^{(j)}) \quad \text{for } i=k \dots, k+s \quad (14b)$$

$$u^{n+1} = \sum_{\ell=1}^k \theta_\ell u^{n-k+\ell} + \Delta t \sum_{j=1}^{k+s} \tilde{b}_j F(y^{(j)}) \quad (14c)$$

where

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0}_{(k-1) \times (s+k-1)} \\ \hat{A} & A \end{bmatrix}, \quad \tilde{\mathbf{D}} = \begin{bmatrix} I_{(k-1) \times (k-1)} & \mathbf{0}_{(k-1) \times (1)} \\ & D \end{bmatrix}, \quad \tilde{\mathbf{b}} = \begin{bmatrix} \hat{\mathbf{b}} & \mathbf{b} \end{bmatrix}$$

are matrices of dimension  $(s+k-1) \times (s+k-1)$ ,  $(s+k-1) \times k$ , and  $1 \times (s+k-1)$ , respectively. This compact form is obtained by defining the first few stages to be older steps. The consistency and stability properties of GLMs are delineated in [B66, B06, 19], reviewed next.

**Definition 4.** Let  $\mathbf{e}$  denote a column vector of 1's, and define the vectors

$$\ell = -[k-1, k-2, \dots, 1, 0]^T \quad \text{and} \quad \mathbf{c} = \tilde{\mathbf{A}}\mathbf{e}_{s+k-1} + \tilde{\mathbf{D}}\ell.$$

Below, let terms of the form  $\mathbf{c}^p, \ell^p$  mean that each element is raised to the power  $p$ , and terms of the form  $\mathbf{c} \cdot \tilde{\mathbf{A}}\mathbf{c}$  mean element-wise multiplication of the vector  $\mathbf{c}$  with the vector  $\tilde{\mathbf{A}}\mathbf{c}$ .

A  $k$ -step  $s$ -stage method of the form (14) is **consistent of order  $p$**  if the order conditions are satisfied

$$\tau_{q_i} = 0, \quad \forall q_i \text{ such that } q \leq p.$$

The values  $\tau_{q_i}$  are given in Appendix A.

Order conditions for GLMs are similar to those of Runge-Kutta methods, but take into account how previous time levels provide additional information in the form of the elementary differentials.

**Definition 5.** A General Linear Method (14) has an **evolution operator**

$$M(z) = \begin{bmatrix} \mathbf{0} & I^{k-1} \\ \Phi(z) & \end{bmatrix}$$

where  $\mathbf{0}$  is a vector of zeros of length  $k-1$ , and  $I^{k-1}$  is a  $(k-1) \times (k-1)$  identity matrix, and

$$\Phi(z) = \Theta + z\tilde{\mathbf{b}}(I - z\tilde{\mathbf{A}})^{-1}\tilde{\mathbf{D}}.$$

**Definition 6.** A GLM of the form (14) is linearly stable if and only if the roots of  $M$  satisfy the root condition: the eigenvalues  $\lambda_i(M)$  are less than or equal to one in magnitude, and that when a given eigenvalue has magnitude one then it must have algebraic multiplicity equal to one.

### 3.3. Formulating the Optimization Problem

Building on the work of Ketcheson [DK08], this section formulates methods to optimize pre- and post-filters. Optimization requires specifying the objective function to be optimized, the inputs, the free parameters, and the equality and inequality constraints. We begin with a core method with given coefficients, and aim to determine the coefficients of a pre-filter and post-filter so that the resulting order is of a specified value  $p$  and the  $A(\alpha)$  stability region is maximized. The optimization problem is described in the following algorithm:

**Optimization algorithm:**

- The GLM is defined by:
  - *Core method coefficients:* The coefficients of the core method  $\mathbf{A}, \hat{\mathbf{A}}, \tilde{\mathbf{D}}$ .
  - *Free variables:* The **pre- and post-filter coefficients**, which are the coefficients in  $\boldsymbol{\theta}, \mathbf{b}, \hat{\mathbf{b}}$  and the first row of  $\mathbf{D}$ .
  - *Computed coefficients:* The coefficients in all but the first row of  $\mathbf{D}$  are defined by (10a), which depends on the core coefficients and the free variables.

- *Select the free variables to maximize  $R$  subject to conditions:*
  1. *(Inequality constraints:)* The eigenvalues  $\lambda_j$  of  $M(z)$  defined in Definition 5 satisfy

$$\max_j |\lambda_j| < 1$$

for all  $z = |z| \exp(i\theta)$  in the wedge defined by

$$0 \leq |z| \leq \infty, \quad \text{and} \quad \theta \in \left( \frac{\mu^2 + 1}{2\mu^2 + 1} \pi, \pi \right) \quad \text{for} \quad 0 \leq \mu \leq r.$$

(We enforce this condition for  $0 \leq |z| \leq 10^4$  in the optimization, but then verify the results for larger values of  $z$ .)

2. *(Equality constraints:)* The order conditions in Definition 4 are satisfied to order  $p$ .

The  $\alpha$  that corresponds to the value of  $r$  resulting from this optimization algorithm is given by  $\alpha = 180 \left( 1 - \frac{r^2+1}{2r^2+1} \right)$ .

If the needs of an application requires optimization of a different type of stability region (e.g. imaginary axis stability or real axis stability), we replace the definition of  $A(\alpha)$  with a different type of stability region, such as:

1. **Imaginary axis stability:** For imaginary axis stability, we require that the eigenvalues  $\lambda_j$  of  $M(z)$  satisfy  $\max_j |\lambda_j| < 1$  for all  $z = i\nu$  where  $0 \leq \nu \leq r$ .
2. **Negative real axis stability:** For negative real axis stability, we require that the eigenvalues  $\lambda_j$  of  $M(z)$  satisfy  $\max_j |\lambda_j| < 1$  for all  $z = -\nu$  where  $0 \leq \nu \leq r$ .

Of course, it is possible to optimize with respect to other properties as well simply by adding these to the objective function or the constraints.

#### 4. Some new methods induced by time-filters

In the following sections we present some of the methods we discussed above and some new methods that we obtained by optimization.

##### 4.1. Core method: Implicit Euler (IE)

Consider the implicit Euler method as the starting point or core method:

$$u^{n+1} = u^n + \Delta t F(u^{n+1}).$$

##### 4.1.1. Second order method based on implicit Euler with 2-point filters (IE-Filt( $d$ ))

The simplest possible pre-filter to the implicit Euler method is a 2 point filter of the form:

$$y^{(1)} = u^n - d(u^n - u^{n-1}) \text{ for some } 0 < d < 1.$$

If a 3-point post-filter is also added the resulting 1-parameter family of second order methods, given in Eqn. (7), can be written as a GLM of the form:

$$\begin{aligned} y^{(1)} &= du^{n-1} + (1-d)u^n \\ y^{(2)} &= y^{(1)} + \Delta t F(y^{(2)}) \\ u^{n+1} &= \frac{2d-1}{3-2d}u^{n-1} + \frac{4(1-d)}{3-2d}u^n + \frac{2}{3-2d}\Delta t F(y^{(2)}). \end{aligned}$$

The case  $d = 0$  gives a method that is not pre-filtered at all, only post-filtered. However, this does not impact the order of the scheme which is  $O(\Delta t^2)$ . This pre-filter does not enhance the order of the method, but it may serve to improve the error constants. The post-filter is designed to impact both the order enhancement and the stability properties of the method.

When implementing this method, especially in a black-box setting, it is easier to write it as

$$\begin{aligned} y^{(1)} &= du^{n-1} + (1-d)u^n \\ y^{(2)} &= y^{(1)} + \Delta t F(y^{(2)}) \\ u^{n+1} &= \frac{1}{3-2d} (2y^{(2)} + 2(1-d)u^n - u^{n-1}) \end{aligned}$$

All values of  $0 \leq d \leq 1$  give an A-stable second order method. We also show that this method is energy-stable for all values  $0 \leq d \leq 1$  (see proof in Appendix). An interesting value is

$$d = \frac{3 - \sqrt{3}}{3},$$

the resulting method is second order, but for linear problems we will see third order convergence.

#### 4.1.2. Second order L-stable method based on implicit Euler with 3-point pre-filter (IE-Pre-2)

We saw above that a 2-point pre-filter introduces a parameter that can then be used to optimize some property of the scheme but does not increase order of accuracy, while the post-filter allows the enhancement of accuracy. In this section we show that 3-point pre-filter can be used to increase order of accuracy while maintaining favorable stability properties.

Consider the 3-point pre-filter added to the core implicit Euler method:

$$y^{(1)} = -\frac{1}{2}u^{n-2} + u^{n-1} + \frac{1}{2}u^n \quad (15a)$$

$$u^{n+1} = y^{(1)} + \Delta t F(u^{n+1}). \quad (15b)$$

This produces a second order approximation to the solution, and the method is L-stable.

We can verify that this method is L-stable by analyzing the eigenvalues of the incremental operator,  $M(z)$ , which advances the solutions to the next time level i.e.

$$\begin{pmatrix} u^{n-1} \\ u^n \\ u^{n+1} \end{pmatrix} = M(z) \begin{pmatrix} u^{n-2} \\ u^{n-1} \\ u^n \end{pmatrix} \quad \text{where} \quad M(z) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{-1}{2(1-z)} & \frac{1}{1-z} & \frac{-1}{2(1-z)} \end{bmatrix}$$

When we take  $\lim_{z \rightarrow -\infty} M(z)$ ,  $M$  becomes upper triangular so all its eigenvalues become zero. This shows that the second order pre-filtered method (15) is L-stable. The stability region of this method is shown in Figure 1 on the left.

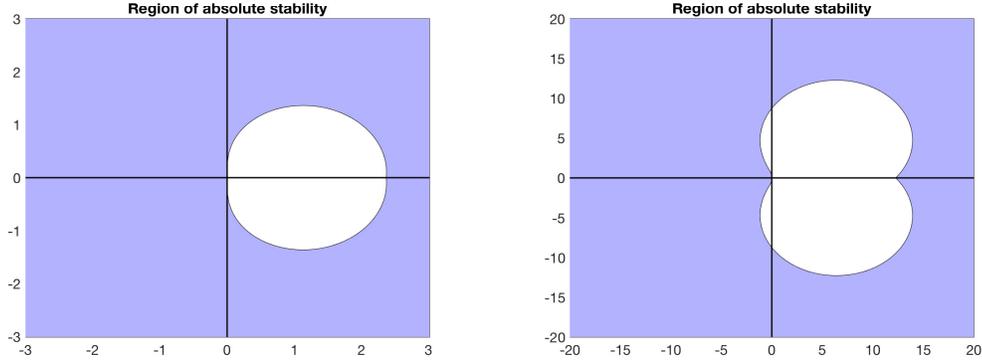


Figure 1: Stability Region of the pre-filtered implicit Euler method IE-Pre-2 (15) (left) which is L-stable, and of the pre- and post-filtered implicit Euler IE-Pre-Post-3 (16) (right) which is not A-stable, but is  $A(\alpha)$  stable with  $\alpha \approx 71.51$ .

#### 4.1.3. Third order method using with 3-point pre- and post-filters (IE-Pre-Post-3)

The method above has only a pre-filter. To raise the order of accuracy of the scheme to third order, we can add a post-filter as well:

$$y^{(1)} = -\frac{1}{2}u^{n-2} + u^{n-1} + \frac{1}{2}u^n \quad (16a)$$

$$y^{(2)} = y^{(1)} + \Delta t F(y^{(2)}) \quad (16b)$$

$$u^{n+1} = \frac{5}{11}u^{n-2} - \frac{15}{11}u^{n-1} + \frac{15}{11}u^n + \frac{6}{11}y^{(2)} \quad (16c)$$

While we have gained order of accuracy, we lost stability properties by adding a post-filter. This third order method is not A-stable, but it has an  $A(\alpha)$  region of stability with  $\alpha \approx 71.51$ . The stability region of this method is shown in Figure 1 on the right.

#### 4.1.4. A-stable third order error inhibiting method (IE-EIS-3)

An A-stable third order method that is based on the implicit Euler method can be obtained by using the error inhibiting approach presented in [EIS20]. In this formulation, we retain previous stages as well as previous

steps, to create a method of the form

$$\begin{aligned}
u^{n+\frac{2}{3}} &= \frac{14}{5}u^{n-\frac{1}{3}} - \frac{9}{5}u^n + \frac{9}{5}\Delta t F(u^{n-\frac{1}{3}}) - \frac{6}{5}\Delta t F(u^n) + \Delta t F(u^{n+\frac{2}{3}}) \\
u^{n+1} &= \frac{14}{5}u^{n-\frac{1}{3}} - \frac{9}{5}u^n + \frac{9}{5}\Delta t F(u^{n-\frac{1}{3}}) \\
&\quad - \frac{47}{60}\Delta t F(u^n) - \frac{1}{12}\Delta t F(u^{n+\frac{2}{3}}) + \Delta t F(u^{n+1}).
\end{aligned}$$

This method takes the time levels  $u^{n-\frac{1}{3}}$  and  $u^n$  and advances them to  $u^{n+\frac{2}{3}}$  and  $u^{n+1}$ . To reveal the dependence on previous stages and previous steps, and the fact that the method is based on the implicit Euler method, we re-write it in the form

$$\begin{aligned}
y_n^{(1)} &= \frac{14}{5}y_{n-1}^{(2)} - \frac{9}{5}u^n + \frac{9}{5}\Delta t F(y_{n-1}^{(2)}) - \frac{6}{5}\Delta t F(u^n) \\
y_n^{(2)} &= y_n^{(1)} + \Delta t F(y_n^{(2)}) \\
y_n^{(3)} &= y_n^{(2)} + \frac{5}{12}\Delta t F(u^n) - \frac{13}{12}\Delta t F(y_n^{(2)}) \\
u^{n+1} &= y_n^{(3)} + \Delta t F(u^{n+1}).
\end{aligned}$$

It is usually preferable to implement this in the form:

$$\begin{aligned}
y_n^{(1)} &= \frac{23}{5}y_{n-1}^{(2)} - 3u^n - \frac{9}{5}y_{n-1}^{(1)} + \frac{6}{5}y_{n-1}^{(3)} \\
y_n^{(2)} &= y_n^{(1)} + \Delta t F(y_n^{(2)})
\end{aligned} \tag{17}$$

$$\begin{aligned}
y_n^{(3)} &= \frac{5}{12}u^n - \frac{1}{12}y_n^{(2)} - \frac{5}{12}y_{n-1}^{(3)} + \frac{13}{12}y_n^{(1)} \\
u^{n+1} &= y_n^{(3)} + \Delta t F(u^{n+1}).
\end{aligned} \tag{18}$$

This method satisfies the order conditions up to second order, but its coefficients also satisfy the error inhibiting property in [EIS20] and so the resulting numerical solution is third order. As mentioned above, it is A-stable, and we notice that while  $y^{(1)}$  and  $y^{(3)}$  are linear combinations of previous steps and stages,  $y^{(2)}$  and  $u^{n+1}$  are simply implicit Euler computations, that can be computed in any legacy code that is based on the implicit Euler method.

It is important to note that, as in Runge–Kutta methods, if the problem is non-autonomous we need to compute the function evaluations at the correct time-levels. In this case the time-levels are  $t_n + \frac{2}{3}\Delta t$  for  $y_n^{(2)}$ , and  $t_n + \Delta t$  for  $u^{n+1}$ .

#### 4.2. Core method: Implicit Midpoint Rule (MP)

Next, we will consider the second order implicit midpoint rule as our core method:

$$u^{n+1} = u^n + \Delta t F \left( \frac{1}{2}u^n + \frac{1}{2}u^{n+1} \right)$$

which, as we saw in Section 2, can be written in the equivalent form

$$\begin{aligned} y^{(1)} &= u^n + \frac{1}{2}\Delta t F(y^{(1)}) \\ u^{n+1} &= 2y^{(1)} - u^n. \end{aligned}$$

Using pre- and post-filters we can raise the order of this method. While the resulting methods are not A-stable, they are  $A(\alpha)$  stable for large values of  $\alpha$ .

##### 4.2.1. Third order filtered implicit midpoint rule (MP-Pre-Post-3)

We can filter the implicit midpoint method to obtain the following third order method:

$$y^{(1)} = -\frac{1}{12}u^{n-3} + \frac{1}{2}u^{n-2} - \frac{5}{4}u^{n-1} + \frac{11}{6}u^n \quad (19a)$$

$$y^{(2)} = y^{(1)} + \frac{1}{2}\Delta t F(y^{(2)}) \quad (19b)$$

$$y^{(3)} = 2y^{(2)} - y^{(1)} \quad (19c)$$

$$u^{n+1} = \frac{1}{2}y^{(1)} + \frac{1}{2}y^{(3)}. \quad (19d)$$

Observe that the final step is simply  $y^{(2)}$ , so we can say

$$\begin{aligned} y^{(1)} &= -\frac{1}{12}u^{n-3} + \frac{1}{2}u^{n-2} - \frac{5}{4}u^{n-1} + \frac{11}{6}u^n \\ y^{(2)} &= y^{(1)} + \frac{1}{2}\Delta t F(y^{(2)}) \\ u^{n+1} &= y^{(2)}. \end{aligned}$$

However, if one is working with a code that treats the implicit midpoint rule as a black box and does not output the intermediate value in the core method, it is more convenient to use (19).

This method is not A-stable, but it is  $A(\alpha)$  stable with  $\alpha = 79.4$ . The advantage of this method is that using the same pre-filter but a different

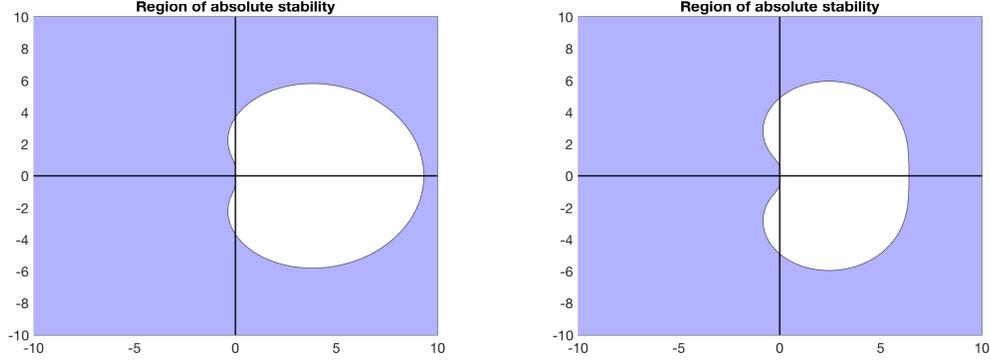


Figure 2: Stability regions of third order MP-Pre-Post-3 (left) and fourth order MP-Pre-Post-4 (right) filtered implicit midpoint methods.

post-filter gives a fourth order method, as we see in the next subsection. The two approaches form an embedded pair which is convenient for error estimation.

#### 4.2.2. Fourth order filtered implicit midpoint rule (MP-Pre-Post-4)

If we use a method similar to (19), but with a different post-filter

$$u^{n+1} = -\frac{1}{25}u^{n-3} + \frac{4}{25}u^{n-2} - \frac{6}{25}u^{n-1} + \frac{4}{25}u^n + \frac{24}{25}y^{(2)}, \quad (20a)$$

we obtain a fourth order method. If the implicit midpoint rule is coded as a black box, we may prefer to write this in the form

$$\begin{aligned} y^{(1)} &= -\frac{1}{12}u^{n-3} + \frac{1}{2}u^{n-2} - \frac{5}{4}u^{n-1} + \frac{11}{6}u^n \\ y^{(2)} &= y^{(1)} + \frac{1}{2}\Delta t F(y^{(2)}) \\ y^{(3)} &= 2y^{(2)} - y^{(1)} \\ u^{n+1} &= -\frac{2}{25}u^{n-3} + \frac{2}{5}u^{n-2} - \frac{21}{25}u^{n-1} + \frac{26}{25}u^n + \frac{12}{25}y^{(3)} \end{aligned}$$

This fourth order method has  $A(\alpha)$  stability region with  $\alpha \approx 70.64$ . By using the post-filter from the third order method (19) and comparing it with the result from this fourth order method, we obtain an error estimator.

### 4.3. Filtered BDF2

In this subsection, we begin with the second order backward differentiation formula (BDF-2) scheme

$$u^{n+1} = -\frac{1}{3}u^{n-1} + \frac{4}{3}u^n + \frac{2}{3}\Delta t F(u^{n+1}).$$

as the core method. We write this method in GLM form as

$$\begin{aligned} y^{(1)} &= u^n \\ y^{(2)} &= -\frac{1}{3}u^{n-1} + \frac{4}{3}u^n + \frac{2}{3}\Delta t F(y^{(2)}). \\ u^{n+1} &= y^{(2)}. \end{aligned}$$

#### 4.3.1. Third order filtered method (BDF2-Post-3)

We can post-filter the BDF2 method to obtain a third order method:

$$\begin{aligned} y^{(1)} &= u^n \\ y^{(2)} &= -\frac{1}{3}u^{n-1} + \frac{4}{3}u^n + \frac{2}{3}\Delta t F(y^{(2)}) \\ u^{n+1} &= y^{(2)} - \frac{2}{11} (y^{(2)} - 3u^n + 3u^{n-1} - u^{n-2}) \\ &= \frac{9}{11}y^{(2)} + \frac{6}{11}u^n - \frac{6}{11}u^{n-1} + \frac{2}{11}u^{n-2}. \end{aligned}$$

This method has stability region  $A(\alpha)$  with  $\alpha = 83.89$ .

#### 4.3.2. A third order method with enhanced stability region (BDF2-Pre-Post-3)

By adding a four-step pre- and post-filter, we can obtain a third order method. The following third order method has four steps, three stages, and stage order  $q = 2$ . The method was created to optimize the value  $\alpha$  in the  $A(\alpha)$  linear stability region. This method has value  $\alpha = 89.59$ .

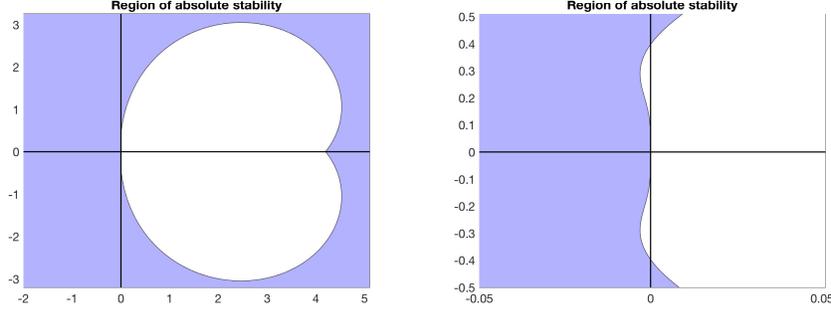


Figure 3: Stability region for the BDF2-Pre-Post-3 (21a), with  $k = 4, s = 3, q = 2$ . On the left we see the stability region, the zoomed image on the right shows that the method is not quite A-stable. In fact, we have  $A(\alpha)$  stability with  $\alpha \approx 89.59$ .

$$y^{(1)} = d_1 u^{n-3} + d_2 u^{n-2} + d_3 u^{n-1} + d_4 u^n, \quad (\text{Pre-filter}) \quad (21a)$$

$$y^{(2)} = -\frac{1}{3}u^{n-1} + \frac{4}{3}y^{(1)} + \frac{2}{3}\Delta t F(y^{(2)}), \quad (21b)$$

$$\begin{aligned} u^{n+1} &= \theta_1 u^{n-3} + \theta_2 u^{n-2} + \theta_3 u^{n-1} + \theta_4 u^n + b\Delta t F(y^{(2)}) \quad (\text{Post-filter}) \quad (21c) \\ &= \theta_1 u^{n-3} + \theta_2 u^{n-2} + \theta_3 u^{n-1} + \theta_4 u^n + \frac{3b}{2}\Delta t \left( y^{(2)} + \frac{1}{3}u^{n-1} - \frac{4}{3}y^{(1)} \right) \\ &= \theta_1 u^{n-3} + \theta_2 u^{n-2} + \left( \theta_3 + \frac{1}{2}b \right) u^{n-1} + \theta_4 u^n + \frac{3b}{2}\Delta t \left( y^{(2)} - \frac{4}{3}y^{(1)} \right), \end{aligned}$$

with coefficients

$$\begin{aligned} d_1 &= 2.670130894410204, & d_2 &= -3.311517498805319, \\ d_3 &= -3.489799303077245, & d_4 &= 5.131185907472361, \\ \theta_1 &= 0.370742163920604, & \theta_2 &= -0.631064728171402, \\ \theta_3 &= -0.729528261935270, & \theta_4 &= 1.989850826186068, \\ b &= 0.120568773483737, \end{aligned}$$

The linear stability region is presented in Figure 3.

#### 4.4. Filtered fully implicit Runge–Kutta (2,2)

We emphasize that this approach works to pre- and post-filter all GLMs, not just linear multistep methods. Consider the L-stable fully implicit Lo-

batto IIIC scheme:

$$\begin{aligned} y^{(1)} &= u^n + \frac{1}{2}\Delta t F(y^{(1)}) - \frac{1}{2}\Delta t F(y^{(2)}) \\ u^{n+1} &= u^n + \frac{1}{2}\Delta t F(y^{(1)}) + \frac{1}{2}\Delta t F(y^{(2)}) \end{aligned}$$

The 2-step time-filtered scheme, which we call RK22-Pre-Post-3, can be written as :

$$\begin{aligned} \hat{u} &= d_1 u^{n-1} + d_2 u^n \\ y^{(1)} &= \hat{u} + \frac{1}{2}\Delta t F(y^{(1)}) - \frac{1}{2}\Delta t F(y^{(2)}) \\ y^{(2)} &= \hat{u} + \frac{1}{2}\Delta t F(y^{(1)}) + \frac{1}{2}\Delta t F(y^{(2)}) \\ u^{n+1} &= q_1 u^{n-1} + q_2 u^n + q_3 y^{(1)} + q_4 y^{(2)} \end{aligned}$$

where

$$\begin{aligned} d_1 &= 0.373461706729200, & d_2 &= 0.626538293270800, \\ q_1 &= -0.075425887737539, & q_2 &= 0.551112405533260, \\ q_3 &= -0.596071637983322, & q_4 &= 1.120385120187601. \end{aligned}$$

This scheme is third order and is A-stable, but not L-stable.

## 5. Numerical tests of the methods

This section presents several numerical test and comparisons of the timestepping methods applied to the Navier-Stokes equations. The spacial terms are discretized by a standard (not upwind) finite element method with inf-sup stable elements. Let  $\mathcal{P}_k^d$  be Lagrange finite elements with  $d$  components with  $d = 2$  or  $3$ , and degree  $k$ . We use Hood-Taylor elements described by  $(\mathcal{P}_k^d, \mathcal{P}_{k-1}^1)$ , which correspond to  $\mathcal{P}_k^d$  vector elements for velocity and  $\mathcal{P}_{k-1}^1$  scalar elements for pressure. We use a sufficiently fine meshes such that we expect the error to be dominated by time discretization.

The fully discrete methods are based on a standard weak formulation for the incompressible NSE. Let  $\Omega$  be and open subset of  $\mathbb{R}^2$ , and let  $(\cdot, \cdot)$  denote the  $L^2(\Omega)$  inner product. In (1), test the momentum equation with a vector valued function  $v$  which vanishes on the boundary and the mass equation

with a scalar function  $q$  with zero mean. After integrating by parts, the weak formulation of (1) is

$$\begin{aligned} (u_t, v) + \nu(\nabla u, \nabla v) + (u \cdot \nabla u + \frac{1}{2}(\nabla \cdot u)u, v) - (p, \nabla \cdot v) &= (f, v), \\ (\nabla \cdot u, q) &= 0. \end{aligned}$$

The nonlinearity has been explicitly skew-symmetrized (when boundary conditions allow) in a standard way by adding  $\frac{1}{2}(\nabla \cdot u)u$ .

While the methods we test are derived for autonomous ODEs, the ensuing tests involve non-autonomous sources and time dependent boundary conditions. There is also a question about the impact of the pre- and post-processors on the fluid pressure since it is an unknown which does not satisfy an evolution equation. These issues are addressed in Appendix C.

The first test in Section 5.1 is a convergence rate verification against a closed form, exact solution. The second test, in Section 5.2, is a benchmark test of flow through a channel past a cylindrical obstacle for which there are published reference values in [J04]. The last test, in Section 5.3, is for a quasi-periodic flow where phase accuracy is important.

### 5.1. Convergence Benchmark Test

For this test we solve the homogeneous NSE (so  $f(x) = 0$ ) with  $d = 2$  under  $2\pi$  periodic boundary conditions with zero mean. Since the solution is analytic, we used higher order Hood-Taylor,  $(\mathcal{P}_4^2, \mathcal{P}_3^1)$  elements and 125 element edges per side of the periodic box, resulting in 640,625 degrees of freedom. The boundary conditions and zero mean condition are imposed strongly (as usual) on the FEM spaces. The Taylor-Green exact solution used is

$$\begin{aligned} u(x, y, t) &= e^{-2\nu t}(\cos x \sin y, -\sin x \cos y) \text{ and} \\ p(x, y, t) &= -\frac{1}{4}e^{-4\nu t}(\cos 2x + \cos 2y) \end{aligned}$$

The solutions are computed to a final time at  $T_f = 1$  for several stepsizes starting from  $\Delta t = 0.2$ , and then halving. Since the solution decays exponentially, absolute errors have little meaning. Thus we compute relative errors at final time  $T_f = 1$

$$\text{relative error} = \sqrt{\frac{\int_{\Omega} |u(T_f) - u_h(T_f)|^2 dx}{\int_{\Omega} |u(T_f)|^2 dx}}.$$

Comparing the methods, for implicit Euler the core method (upper row of Figure 4) *pre- and post-filtered implicit Euler and EIS3 are by far the most accurate*. EIS3 requires 2 implicit Euler solves per step compared to 1 solve/step for pre & post-filtered IE. Both attain their the  $\mathcal{O}(\Delta t^3)$  rate of convergence, as predicted by the theory. The middle row treats the midpoint rule plus filters. The result here is entire consistent: *higher accuracy (in the sense of consistency error) produces a more accurate approximation*. We note that in the right side figure the 4th order approximation hits an error plateau of  $10^{-7}$  where the spacial errors are no longer negligible. In the bottom row the second BDF2 filtered method performed far better, attaining its expected rate of convergence. In all tests, *run times depended on the number of core method solves, independent of the number of filter steps*, as expected.

### 5.2. Benchmark test: Flow past a cylinder

This next test is a commonly used benchmark described in [ST96]. Fluid flows into a channel from the left and flows around slightly off center cylindrical obstacle. The fluid starts at rest and the inflow velocity is ramped up from zero. When the inflow velocity is high enough, vortices shed off the obstacle (see Figure 5). The data monitored are the lift and drag due to the cylinder, and the pressure difference before and after the cylinder. The geometry and flow profile is given by [ST96]; we compare our results to benchmark lift and drag values obtained in a DNS study from [J04].

Time dependent boundary conditions present questions for both multi-step and multi-stage methods. Here it also means that the numerical solution will satisfy filtered BCs rather than their exact values since we applied the filtering steps as written with no special treatment of the inflow. The error committed at the boundaries is still consistent up to the order of the method and no problem was observed.

The flow configuration, [ST96], [J04], is as follows. The kinematic viscosity  $\nu = 10^{-3}$ , the final time is  $T_f = 8$ , and the domain is

$$\Omega = \{(x, y) \mid 0 < x < 2.2, \ 0 < y < 0.41 \text{ and } (x - 0.2)^2 + (y - 0.2)^2 > 0.05^2\}.$$

The external body force  $f$  is set to zero. The inflow and outflow velocities are parabolic:

$$u|_{(0,y)} = u|_{(2.2,y)} = 0.41^{-2}(6y(0.41 - y), 0).$$

We use  $(\mathcal{P}_2^2, \mathcal{P}_1^1)$  elements with a well resolved, static mesh with 479,026 degrees of freedom with 1000 edges on the interior cylinder boundary. This is the same mesh used in [DLZ18] and was generated by adaptive refinement from solving the steady problem.

We measure the maximum drag  $c_{d,\max}$ , the time of maximum drag  $t(c_{d,\max})$ , maximum lift  $c_{l,\max}$ , time of maximum lift  $t(c_{l,\max})$  and the pressure drop at the final time between the front and back of the cylinder,  $\Delta p(8) = p_h(0.15, 0.2) - p_h(0.25, 0.2)$ . We run the tests for the same  $\Delta t$ 's in [J04], which have a largest value of  $\Delta t = 0.04$ , and are successively halved until the smallest value of  $\Delta t = 0.00125$ . The stepsizes are doubled for IE-EIS-3 for a fair comparison since it requires two implicit Euler solves for one timestep. If a simulation failed, the missing values are filled in with dashes in the tables. Failure only happened for IE-Pre-Post-3 and MP-Pre-Post-4 when the energy of the solution grew which was followed by both Newton and fixed-point iterations failing in the nonlinear solve.

The results for the methods based on IE are shown in Table 1, results for methods based on BDF2 are shown in Table 2, and results for methods based on MP are shown in Table 3.

For the IE based methods, every method shows improvement over IE in the prediction of the maximum lift with the exception IE-Pre-Post-3 for the larger stepsizes. IE-Pre-Post-3 was the only IE based method to show instability, and the simulation failed to run to completion for larger stepsizes. IE-EIS-3 showed superior accuracy at the smallest stepsize in the lift coefficient and pressure drop.

For the BDF2 based methods, both BDF2-Post-3 and BDF2-Pre-Post-3 show a dramatic improvement in predicting the final pressure drop over BDF2. Interestingly, BDF2-Pre-Post-3 exhibits better convergence of pressure than was suggested by the test in Section 5.1.

For the MP based methods, MP and MP-Pre-Post-2 yielded essentially identical results. The most noticeable improvement over the base method is MP-Pre-Post-3's pressure drop which matches all four digits of the reference values for the smallest three  $\Delta t$ s. For all the methods, the maximum lift coefficient appears to be converging to a value slightly higher than the reference value. MP-Pre-Post-4 did not finish for  $\Delta t$ s higher than 0.0025.

IE based methods.					
$\Delta t$	$t(c_{d,\max})$	$c_{d,\max}$	$t(c_{l,\max})$	$c_{l,\max}$	$\Delta p(8)$
Reference Values					
—	3.93625	2.950921575	5.693125	0.47795	-0.1116
IE					
0.005	3.93	2.950301672	6.28	0.17604	-0.1005
0.0025	3.9325	2.950371110	6.215	0.30336	-0.1070
0.00125	3.93375	2.950529384	5.7175	0.38229	-0.1114
IE-Pre-2 (one IE solve and one filter per timestep)					
0.005	3.935	2.950802171	5.72	0.45978	-0.1111
0.0025	3.935	2.950872330	5.7	0.47413	-0.1120
0.00125	3.93625	2.950889791	5.695	0.47728	-0.1117
IE-Filt( $\frac{3-\sqrt{3}}{3}$ ) (one IE solve and two filters per timestep)					
0.005	3.93	2.950839424	5.71	0.46722	-0.1127
0.0025	3.9325	2.950880844	5.695	0.47567	-0.1124
0.00125	3.935	2.950891744	5.6925	0.47762	-0.1120
IE-Pre-Post-3 (one IE solve and two filter per timestep)					
0.005	7.825	435.1275230	7.84	205.33324	-5.1966
0.0025	3.935	2.950897874	5.6925	0.47895	-0.1116
0.00125	3.93625	2.950895596	5.6925	0.47833	-0.1116
IE-EIS-3 (two IE solves and two filters per timestep)					
0.01	3.93667	2.950816639	5.69667	0.46183	-0.1119
0.005	3.935	2.950884378	5.69333	0.47608	-0.1117
0.0025	3.93667	2.950893844	5.6925	0.47797	-0.1116

Table 1: For the smallest stepsize, the lift calculated by IE does not have any digits agreement for the smallest  $\Delta t$ , but the methods based on it have at least two digits of accuracy. IE-EIS-3 has the best agreement with the reference lift and pressure drop values for the smallest  $\Delta t$ . IE-Pre-Post-3 is unstable and/or did not finish for several stepsizes.

BDF2 based methods.					
$\Delta t$	$t(c_{d,\max})$	$c_{d,\max}$	$t(c_{l,\max})$	$c_{l,\max}$	$\Delta p(8)$
Reference Values					
—	3.93625	2.950921575	5.693125	0.47795	-0.1116
BDF2					
0.02	3.94	2.950423752	5.86	0.34749	-0.1063
0.005	3.935	2.950858401	5.705	0.47141	-0.1120
0.00125	3.93625	2.950893074	5.69375	0.47787	-0.1117
BDF2-Post-3 (one BDF2 solve and one filter per timestep)					

0.02	3.94	2.951551390	5.7	0.56819	-0.1119
0.005	3.935	2.950905566	5.695	0.48028	-0.1116
0.00125	3.93625	2.950895363	5.6925	0.47828	-0.1116
BDF2-Pre-Post-3 (one BDF2 solve and two filters per timestep)					
0.02	3.94	2.950855922	5.86	0.44850	-0.1028
0.005	3.935	2.950932030	5.695	0.48538	-0.1116
0.00125	3.93625	2.950896341	5.6925	0.47845	-0.1116

Table 2: BDF2-Post-3 and BDF2-Pre-Post-3 have better agreement with the reference pressure drop than BDF2. BDF2-Post-3 tends to overestimate the lift coefficient for larger stepsizes.

Methods based on the implicit midpoint rule					
$\Delta t$	$t(c_{d,\max})$	$c_{d,\max}$	$t(c_{l,\max})$	$c_{l,\max}$	$\Delta p(8)$
Reference Values					
—	3.93625	2.950921575	5.693125	0.47795	-0.1116
MP					
0.005	3.935	2.950893884	5.695	0.47780	-0.1118
0.0025	3.9375	2.950894329	5.6925	0.47809	-0.1117
0.00125	3.93625	2.950895120	5.6925	0.47821	-0.1116
MP-Pre-Post-2 (one MP solve and two filters per timestep)					
0.005	3.935	2.950886785	5.695	0.47683	-0.1118
0.0025	3.935	2.950892638	5.6925	0.47785	-0.1117
0.00125	3.93625	2.950894681	5.6925	0.47815	-0.1116
MP-Pre-Post-3 (one MP solve and two filters per timestep)					
0.005	3.935	2.950896750	5.695	0.47840	-0.1116
0.0025	3.935	2.950894895	5.6925	0.47830	-0.1116
0.00125	3.93625	2.950895215	5.6925	0.47825	-0.1116
MP-Pre-Post-4 (one MP solve and two filters per timestep)					
0.005	—	—	—	—	—
0.0025	3.935	2.950894654	5.6925	0.47824	-0.1116
0.00125	3.93625	2.950895183	5.6925	0.47824	-0.1116

Table 3: MP-Pre-Post-3 has exact agreement with the reference pressure drop for the smallest three stepsizes. MP and MP-Pre-Post-2 give nearly identical values. MP-Pre-Post-4 was unstable for  $\Delta t$  larger than 0.0025.

### 5.3. Offset cylinder test

We now consider a body forced internal flow between two offset cylinders inspired by e.g. [EP00]. This flow is transitional between periodic and tur-

bulent. Similar tests have been used in, e.g., [J15]. The domain is a unit cylinder centered at the origin, minus a smaller cylinder of radius 0.1 centered at  $(0, 0.5)$ . The flow is forced by

$$f(t, x, y) = 4 \min(1, t)(1 - x^2 - y^2) \langle -y, x \rangle.$$

The kinematic viscosity is set to  $\nu = \frac{1}{150}$ . The flow is started at rest with an initial condition is  $u \equiv 0$ . The simulation is run from  $t \in [0, 30]$ . We first obtain accurate reference data using the implicit midpoint rule for several mesh refinements and step sizes, with a fine mesh refinement of 255,870 degrees of freedom using  $(\mathcal{P}_2^2, \mathcal{P}_1^1)$  elements, and a smallest stepsize of  $\Delta t = 0.000625$ . We capture the evolution of the kinetic energy, several snapshots, and probe each component of velocity at several points.

For timesteps where the base methods fail to capture the correct dynamics the filtered methods capture the reference solution better. Solution snapshots for IE based methods are shown in Figure 10, and the snapshots for BDF2 and MP based methods are shown in Figure 11.

For a quantitative comparison, we measure the error of the  $x$ -component of the velocity vector at the point  $(0.5, 0.2)$ , which is in the wake of the obstacle. Since the solution is quasi-periodic, we normalize the error by the difference between the maximum and minimum values taken on by the reference solution. To calculate the error, let  $u = u(t, x, y)$  be the reference solution and  $u_h(t, x, y)$  is the discrete solution, which has been linearly interpolated in time. The absolute and relative error at time  $t$  is defined as follows:

$$\begin{aligned} \text{absolute error}(t) &= \sup_{s \in [0, t]} |u(s, 0.5, 0.2) - u_h(s, 0.5, 0.2)|, \\ \text{relative error}(t) &= \frac{\text{absolute error}(t)}{\sup_{s \in [0, T]} u(s, 0.5, 0.2) - \inf_{s \in [0, T]} u(s, 0.5, 0.2)}. \end{aligned}$$

The reference  $x$ -component of the velocity is shown in Figure 6. The errors of the IE, BDF2, and MP based methods are given in Figures 7, 8, and 9.

IE-Pre-Post-3, MP-Pre-Post-3 and MP-Pre-Post-4 were unstable for some or all of the stepsizes shown. The solution exhibits the richness of scales typical for higher Reynolds number flows. One consequence may be increased importance of A-stability and instabilities seen in methods that are only  $A(\alpha)$  stable. *We have derived the methods herein by optimization of accuracy. Deriving methods by optimizing stability instead can be done using the tools herein.* Doing so is an important open question at higher Reynolds numbers.

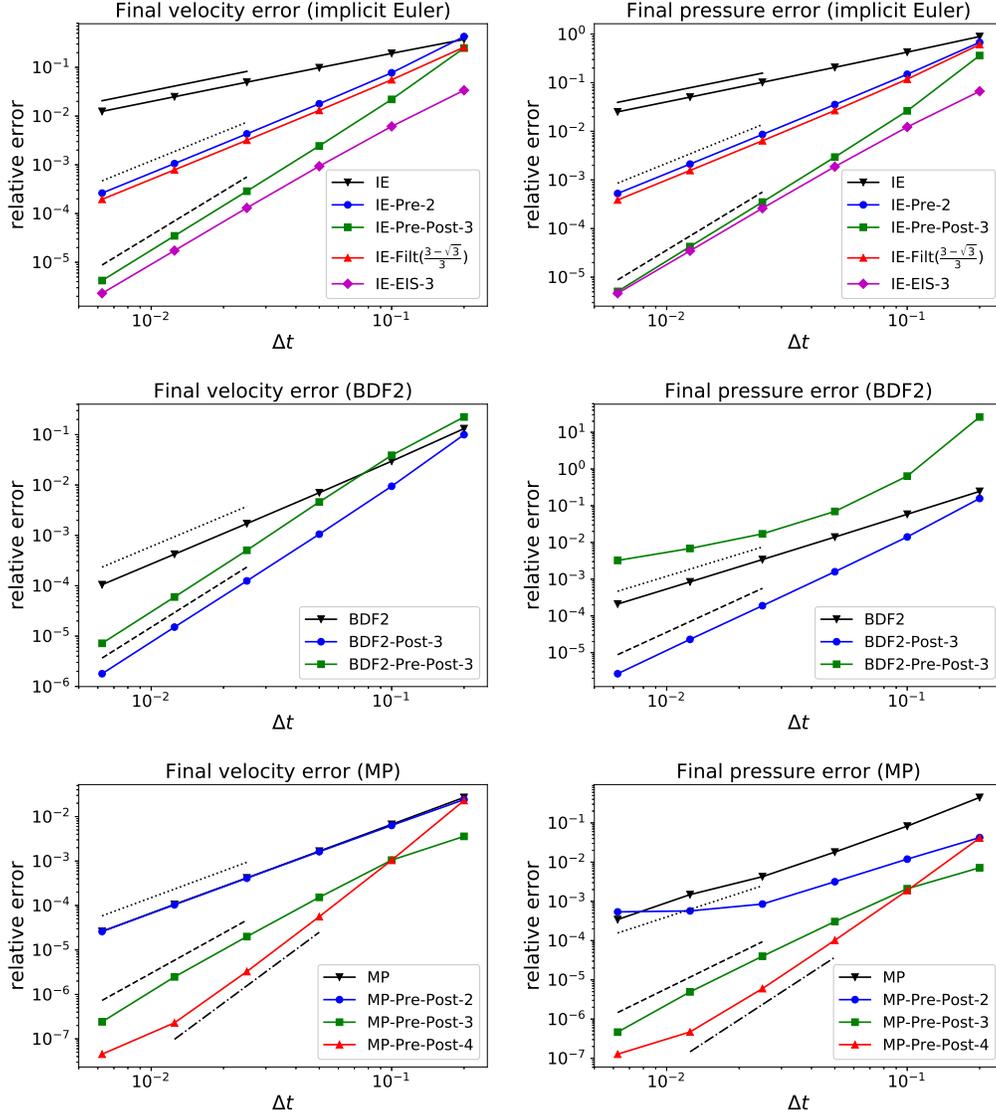


Figure 4: Convergence rates of velocity and pressure for different base methods. The reference lines for the orders of the method are solid (first), dotted (second), dashed (third) and dashed-dotted (fourth). The velocities of the methods converge at the expected rates except. The errors in pressure for BDF2-Pre-Post-3 and MP-Pre-Post-2 either plateau or converge sub-optimally. The reason is unknown and is the subject of future research.

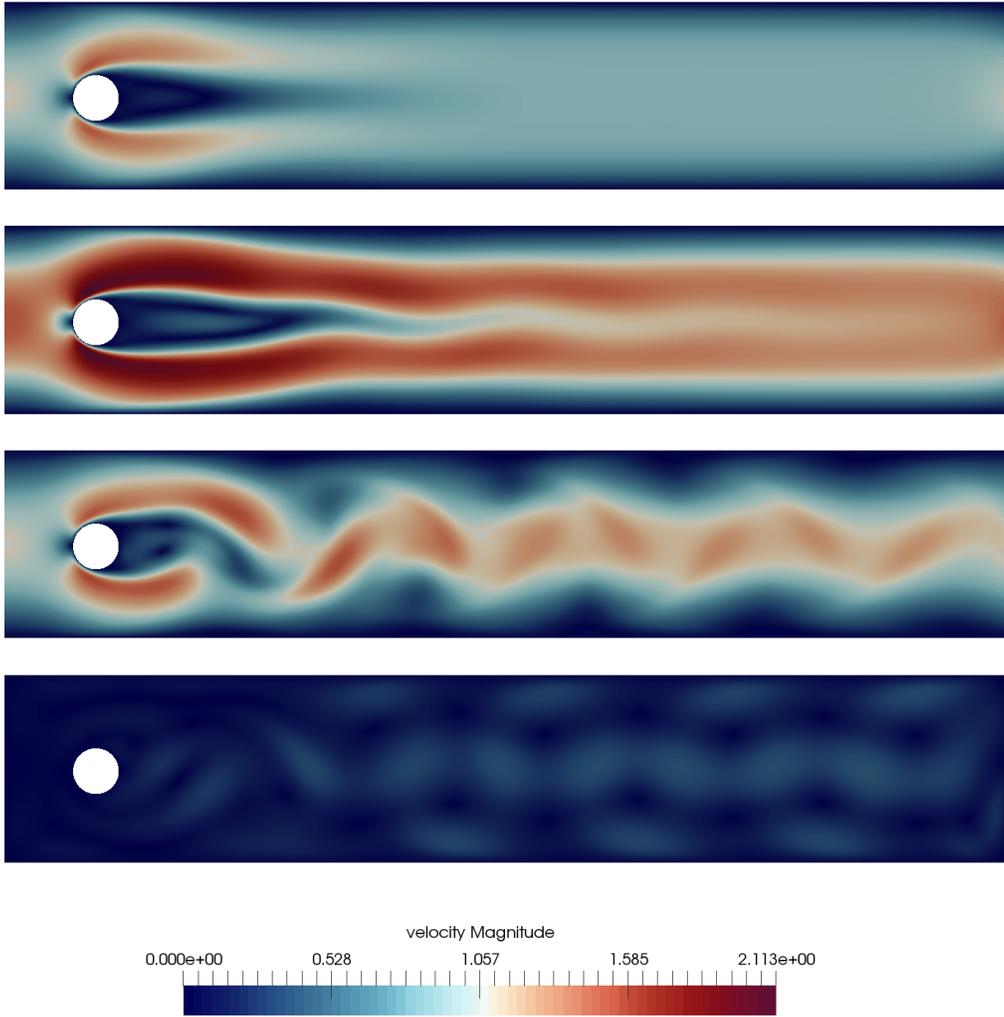


Figure 5: Snapshots of the flow past a cylinder solution, described in Section 5.2, at times (from top to bottom)  $t = 2, 4, 6,$  and  $8.$

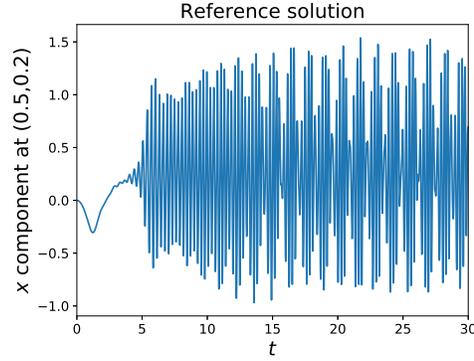


Figure 6: The  $x$  component of the reference solution shows quasi-periodic behavior.

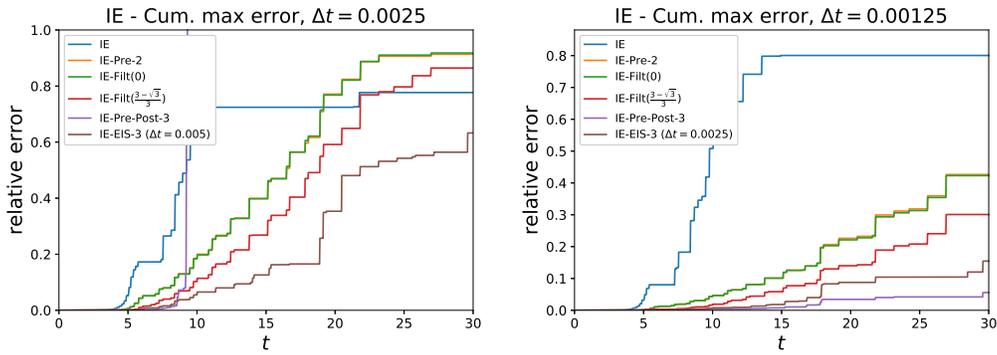


Figure 7: For  $\Delta t = 0.0025$ , IE-Pre-Post-3 is unstable while IE-EIS-3 performs the best. For  $\Delta t = 0.00125$ , IE-Pre-Post-3 becomes stable and is the most accurate solution.

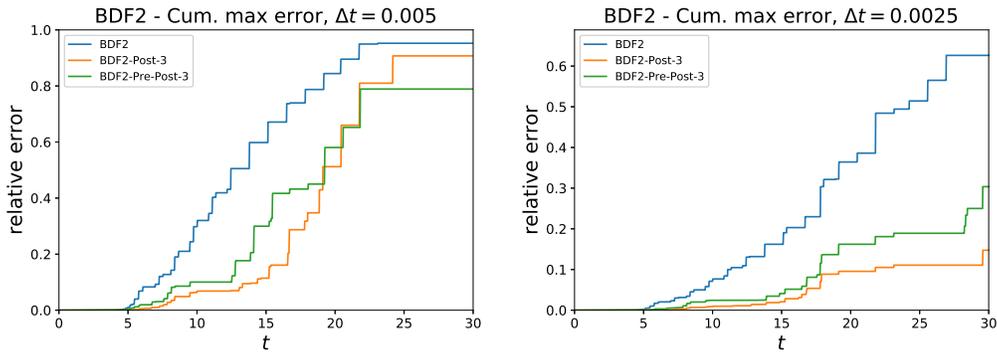


Figure 8: The third order methods are more accurate than their base method, BDF2.

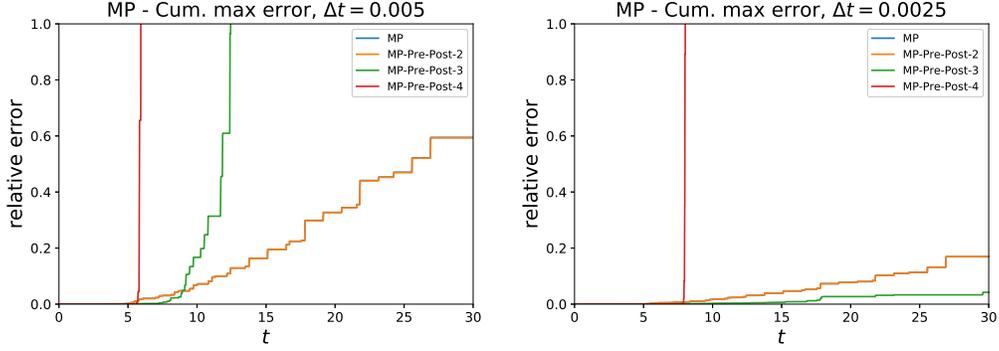


Figure 9: For both  $\Delta t$ s, MP and MP-Pre-Post-2 give essentially the same answer and MP-Pre-Post-4 is unstable. For  $\Delta t = 0.0025$ , MP-Pre-Post-3 is the most accurate, but is unstable when  $\Delta t$  is doubled. IE-Filt( $\frac{3-\sqrt{3}}{3}$ ) performs the best of the second order methods. IE-Pre-2 and IE-Filt(0) are essentially the same.

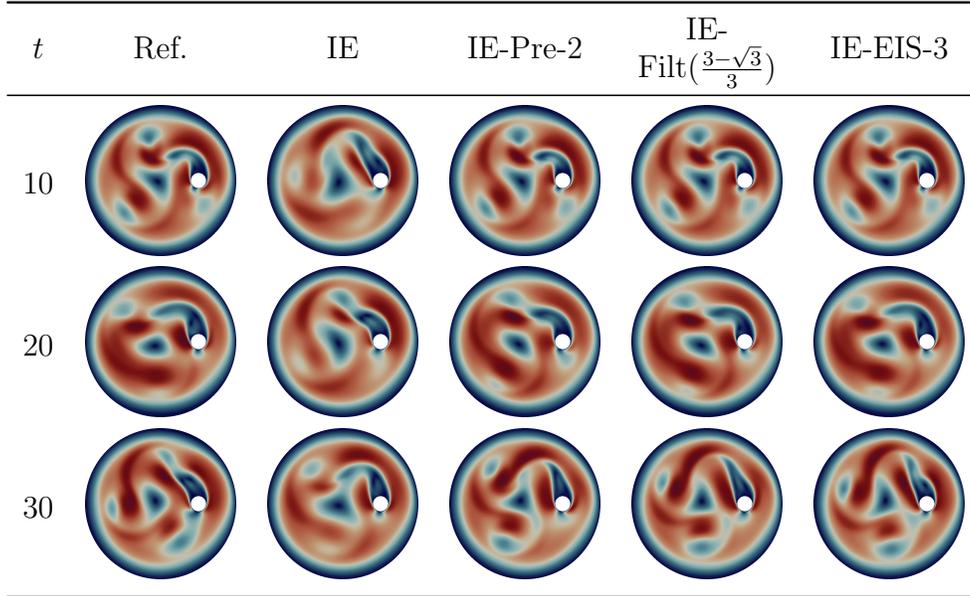


Figure 10: Snapshots of the IE based methods with  $\Delta t = 0.0025$  (and  $\Delta t = 0.005$  for IE-EIS-3). All methods showed improvement over IE. IE-EIS-3 gave the closest results to the reference solution, as observed, by smaller phase error at  $t = 30$  (for example, compare the detached vortex shown by IE-EIS-3 versus the attached vortex in the second order methods.). The IE-Pre-Post-3 simulation failed before the first snapshot could be generated.

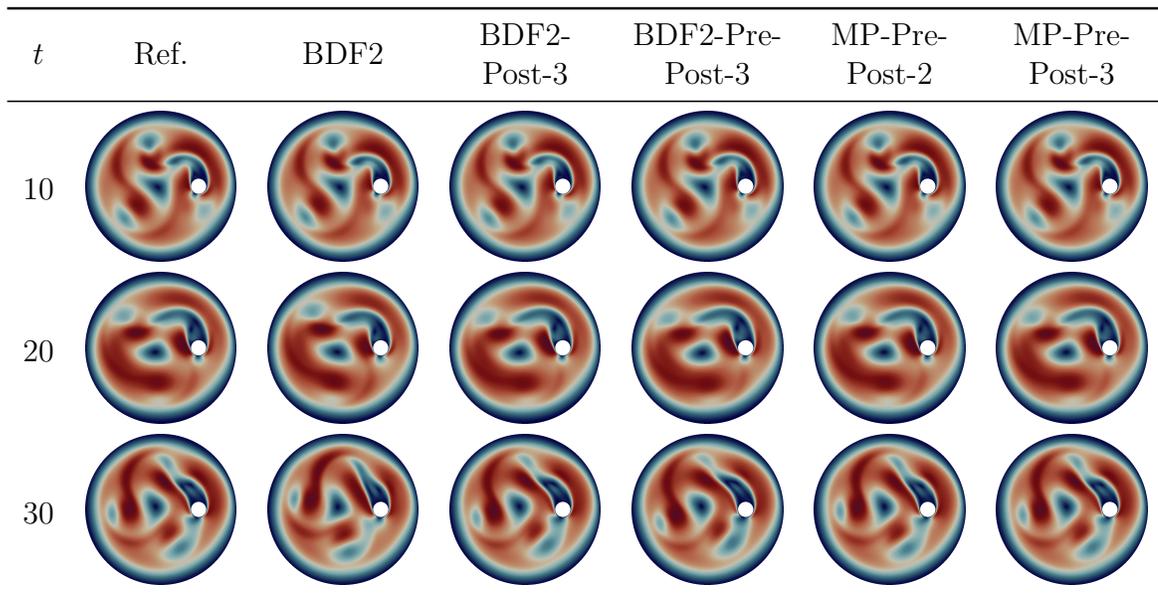


Figure 11: Snapshots of the BDF2 and MP based methods with an effective  $\Delta t = 0.0025$ . Both of the third order methods show better agreement with the reference solution than BDF2. For the MP based methods, the MP snapshots (not shown) were nearly identical to the MP-Pre-Post-2 snapshots. MP-Pre-Post-3 appears to most accurately capture the reference solution out of the methods in this table. The MP-Pre-Post-4 simulation failed before the first snapshot could be taken.

## 6. Conclusions and future problems

This work presented new time discretization methods with attractive properties and a new GLM framework for developing and analyzing time-filtered methods. Using the tools of GLM analysis a number of interesting new methods have been derived and tested. These methods are easy to implement within existing codes, and can be used to improve the time accuracy of legacy codes. The novel time-filtering methods have low cognitive complexity, and are in some cases optimized for other application-specific criteria. Among these new methods, the error inhibiting method shows special promise in tests even adjusting results for its added complexity.

In this work, we presented (mostly) a linear stability analysis and tested the methods for a nonlinearity dominated application. The development of a general energy stability theory of any new method herein would be a significant further advance. The new methods presented have an embedded structure, so that step and order adaptivity are natural next steps. Our long term goal is to develop self-adaptive, variable step, variable order methods that are of low cognitive complexity and easily implemented in legacy codes based on these embedded families.

**Acknowledgements:** The authors acknowledge support from AFOSR Grant No. FA9550-18-1-0383 (S.G.) and NSF Grant No. DMS1817542 (W.L.).

### Appendix A. Order Conditions

The order conditions are typically given in the form:

1. Consistency conditions ( $q = 0$ ):

$$\tau_{0_1} = \Theta e_k - 1 \quad \text{and} \quad \tau_{0_2} = \tilde{\mathbf{D}} e_k - e_{s+k-1}.$$

2. First order condition ( $q = 1$ ):

$$\tau_{1_1} = \tilde{\mathbf{b}} e_{s+k-1} + \Theta \ell - 1.$$

3. Second order condition ( $q = 2$ ):

$$\tau_{2_1} = \tilde{\mathbf{b}} \mathbf{c} + \frac{1}{2} \Theta \ell^2 - \frac{1}{2}.$$

4. Third order conditions ( $q = 3$ ):

$$\tau_{3_1} = \tilde{\mathbf{b}}\mathbf{c}^2 + \frac{1}{3}\Theta\ell^3 - \frac{1}{3} \quad \text{and} \quad \tau_{3_2} = \tilde{\mathbf{b}}^t\tilde{\mathbf{A}}\mathbf{c} + \frac{1}{2}\tilde{\mathbf{b}}\tilde{\mathbf{D}}\ell^2 + \frac{1}{6}\Theta\ell^3 - \frac{1}{6}.$$

5. Fourth order conditions ( $q = 4$ ):

$$\begin{aligned} \tau_{4_1} &= \tilde{\mathbf{b}}\mathbf{c}^3 + \frac{1}{4}\Theta\ell^4 - \frac{1}{4}, \\ \tau_{4_2} &= \tilde{\mathbf{b}}\tilde{\mathbf{A}}\mathbf{c}^2 + \frac{1}{3}\tilde{\mathbf{b}}\tilde{\mathbf{D}}\ell^3 + \frac{1}{12}\Theta\ell^4 - \frac{1}{12}, \\ \tau_{4_3} &= \tilde{\mathbf{b}}\tilde{\mathbf{A}}\tilde{\mathbf{A}}\mathbf{c} + \frac{1}{2}\tilde{\mathbf{b}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}\ell^2 + \frac{1}{6}\tilde{\mathbf{b}}\tilde{\mathbf{D}}\ell^3 + \frac{1}{24}\Theta\ell^4 - \frac{1}{24} \\ \tau_{4_4} &= \tilde{\mathbf{b}}(\mathbf{c} \cdot \tilde{\mathbf{A}}\mathbf{c}) + \frac{1}{2}\tilde{\mathbf{b}}(\mathbf{c} \cdot \tilde{\mathbf{D}}\ell^2) + \frac{1}{8}\Theta\ell^4 - \frac{1}{8}. \end{aligned}$$

## Appendix B. Energy stability Proof for the IE-Filt(d) methods

The d-filtered family of methods for Implicit Euler methods IE-Filt(d) are given by

$$\begin{aligned} y^{(1)} &= du^{n-1} + (1-d)u^n \\ y^{(2)} &= y^{(1)} + \Delta t F(y^{(2)}) \\ u^{n+1} &= \frac{1}{3-2d}(2y^{(2)} + 2(1-d)u^n - u^{n-1}) \end{aligned}$$

We consider the energy at the second stage and rewrite  $y^{(2)}$  in terms of  $u^{n-1}, u^n$  and  $u^{n+1}$ .

$$\langle y^{(2)}, F(y^{(2)}) \rangle = \langle y^{(2)}, y^{(2)} - du^{n-1} + (1-d)u^n \rangle$$

where

$$y^{(2)} = \frac{3-2d}{2}u^{n+1} + (d-1)u^n + \frac{1}{2}u^{n-1}.$$

If the operator F is dissipative then the inner product is negative,

$$\langle x, F(x) \rangle \leq 0$$

and so we have

$$\left\langle \frac{3-2d}{2}u^{n+1} + (d-1)u^n + \frac{1}{2}u^{n-1}, \frac{3-2d}{2}u^{n+1} + 2(d-1)u^n + \frac{1-2d}{2}u^{n-1} \right\rangle \leq 0.$$

Expanding and re-arranging, we have

$$\begin{aligned}
& \frac{2d^2 - 7d + 6}{4} \|u^{n+1}\|^2 - \frac{(2d-3)(d-1)}{2} \langle u^{n+1}, u^n \rangle + \frac{2d^2 - 3d + 2}{4} \|u^n\|^2 \\
& \leq \frac{2d^2 - 7d + 6}{4} \|u^n\|^2 - \frac{(2d-3)(d-1)}{2} \langle u^n, u^{n-1} \rangle \\
& + \frac{2d^2 - 3d + 2}{4} \|u^{n-1}\|^2 - \frac{(2d-3)(d-1)}{4} \|u^{n+1} - 2u^n + u^{n-1}\|^2. \quad (\text{B.1})
\end{aligned}$$

$$\text{Let } X^n = \begin{pmatrix} u^n \\ u^{n-1} \end{pmatrix}, \text{ and so } X^{n+1} = \begin{pmatrix} u^{n+1} \\ u^n \end{pmatrix},$$

and define the G-norm, for any SPD matrix G:

$$\|X\|_G^2 = X^t G X.$$

In particular, we are interested in the matrix

$$G = \frac{1}{4} \begin{bmatrix} 2d^2 - 7d + 6 & -(2d-3)(d-1) \\ -(2d-3)(d-1) & 2d^2 - 3d + 2 \end{bmatrix}$$

which is SPD for  $d \leq \frac{3}{2}$ . Observe that eqn (B.1) becomes

$$\|X^{n+1}\|_G^2 \leq \|X^n\|_G^2 - \frac{(2d-3)(d-1)}{4} \|u^{n+1} - 2u^n + u^{n-1}\|^2$$

Clearly, then, the d-filtered Implicit Euler method is energy stable for  $0 \leq d \leq 1$ .

### Appendix C. Considerations for non-autonomous forces, time-dependent boundary conditions, and pressure recovery

In this section, we list the abscissas and pressure recovery formulas for the methods developed herein. While the methods were derived using autonomous theory for simplicity, the extension to non-autonomous ODEs is straightforward. The pre-processing steps can shift the abscissas of the data, so non-autonomous sources (such as  $f(t)$  in (1) or externally applied time dependent boundary conditions) must be adjusted accordingly.

For the incompressible Navier-Stokes equations, pressure is not solved via an evolution equation, but the gradient of the pressure is a force in the evolution equation for velocity. Thus, the resulting pressure of the fully discrete

methods are collocated at the same time as the non-autonomous forces. Some quantities of interest, such as lift and drag, require velocity and pressure approximations at the same time level, so pressure must be interpolated or extrapolated as necessary. All the methods presented herein produce velocity approximations at time  $t = t^{n+1}$  after post-processing. Thus, we explain how to derive a pressure approximation such that  $p^{n+1} \approx p(t^{n+1})$ .

Let  $\tilde{p}^{n+1}$  denote the intermediate pressure from solving the coupled velocity-pressure system for  $(u^{n+1}, \tilde{p}^{n+1})$ . Note that  $\tilde{p}^{n+1}$  is not necessarily an approximation at time level  $t^{n+1}$ . Denote the abscissa for non-autonomous forces by  $\tilde{t}$ .

The formulas for IE, IE-Pre-2, IE-Pre-Post-3, IE-Filt(0), BDF2, BDF2-Post-3, and MP-Pre-Post-2/3/4 are

$$\tilde{t} = t^n + \Delta t, \quad p^{n+1} = \tilde{p}^{n+1},$$

the formulas for MP are

$$\tilde{t} = t^n + \frac{1}{2}\Delta t, \quad p^{n+1} = \frac{3}{2}\tilde{p}^{n+1} - \frac{1}{2}\tilde{p}^n.$$

the formulas for IE-Filt( $d$ ) are

$$\tilde{t} = t^n + (1 - d)\Delta t, \quad p^{n+1} = (1 + d)\tilde{p}^{n+1} - d\tilde{p}^n,$$

and the formulas for BDF2-Pre-Post-3 are

$$\tilde{t} = t^n + c \Delta t, \quad p^{n+1} = a_3\tilde{p}^{n+1} + a_2\tilde{p}^n + a_1\tilde{p}^{n-1},$$

$$c = 3.930023404911324, \quad \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 2.827506874208412 \\ -2.7249903435055 \\ 0.8974834692970881 \end{bmatrix}.$$

IE-EIS-3 contains two implicit Euler solves per timestep and yields pressure approximations at both  $t^{n+2/3}$  and  $t^{n+1}$ . There is a pair  $(\tilde{t}^{(2)}, \tilde{p}^{(2)})$  associated with the intermediate solve in (17), and a pair  $(\tilde{t}, \tilde{p}^{n+1})$  associated with the final solve in (18). The formulas are simple;

$$\tilde{t}^{(2)} = t^n + \frac{2}{3}\Delta t, \quad p^{n+2/3} = \tilde{p}^{(2)}, \quad \tilde{t} = t^{n+1}, \quad p^{n+1} = \tilde{p}^{n+1}.$$

## References

- [A72] R.A. ASSELIN, *Frequency filter for time integration*, Mon. Weather Review 100(1972), 487-490.
- [A115] M. ALNÆS, J. BLECHTA, J. HAKE, A. JOHANSSON, B. KEHLET, A. LOGG, C. RICHARDSON, J. RING, M.E. ROGNES, G.N. WELLS, *The FEniCS project version 1.5*, Archive of Numerical Software 3 (2015), 9–23.
- [B76] G.A. BAKER, *Galerkin approximations for the Navier-Stokes equations*, Technical Report, 1976.
- [BDK82] G.A. BAKER, V.A. DOUGALIS, O.A. KARAKASHIAN, *On a higher order accurate fully discrete Galerkin approximation to the Navier-Stokes equations*, Math of Comp. 39 (1982), 339-375.
- [B66] J.C. BUTCHER *A multistep generalization of Runge-Kutta methods with four or five stages*, J. Assoc. Comput. Mach. 14 (1967), 84-89.
- [B06] J.C. BUTCHER *General linear methods*, Acta Numerica 15 (2006), 157–256
- [CR78] M. CROUZEIX AND P.A. RAVIART, 1978. *Approximation d'équations d'évolution linéaires par des méthodes multipas*. Étude numérique des grands systèmes. Proc. Sympos., Novosibirsk, 1976, Méthodes Math. Inform, 7, pp.133-150.
- [D78] G. DAHLQUIST, *Positive functions and some applications to stability questions for numerical methods*, 1-29 in: Recent advances in numerical analysis, (editors: C. de Boor and G. Golub) Academic Press, 1978.
- [D79] G. DAHLQUIST, *Some properties of linear multistep and one-leg methods for ordinary differential equations*, Conference Proceeding, 1979 SIGNUM Meeting on Numerical ODE's, Champaign, Ill., available at: <http://cds.cern.ch/record/1069163/files/CM-P00069449.pdf> .

- [DGLL18] V. DECARIA, A. GUZEL W. LAYTON AND YI LI, *A new embedded variable stepsize, variable order family of low computational complexity*, <https://arxiv.org/abs/1810.06670>, 2018.
- [DLZ18] V. DECARIA, W. LAYTON AND HAIYUN ZHAO, *Analysis of a low complexity, time-accurate discretization of the Navier-Stokes equations*, IJNAM, 17 (2020), 254-280.
- [EIS20] A Ditkowski, S Gottlieb, Z. J. Grant *Explicit and implicit error inhibiting schemes with post-processing* Computers & Fluids, **208**(2020).
- [D13] D.R. DURRAN, *Numerical methods for wave equations in geophysical fluid dynamics*, Vol. 32. Springer Science & Business Media, 2013.
- [EP00] C. EGBERS AND G. PFISTER, *Physics of rotating fluids*, Springer LN in Physics 549 (2018).
- [E04a] E. EMMRICH, *Error of the two-step BDF for the incompressible Navier-Stokes*, ESAIM: M2AN, 38 (2004), 757-764.
- [E04b] E. EMMRICH, *Stability and convergence of the two-step BDF for the incompressible Navier-Stokes Problem*, International Journal of Nonlinear Sciences and Numerical Simulation, 5 (2004), 199-209.
- [G79] V. GIRAULT AND P.A. RAVIART, *Finite element approximation of the Navier-Stokes equations*, Berlin, Springer-Verlag, Lecture Notes in Mathematics, vol. 749, 1979.
- [G83] R.D. GRIGORIEFF, *Stability of multi-step methods on variable grids*, Numer. Math. 42(1983) 359-377.
- [19] E HAIRER, SP NORSETT, AND G WANNER, *Solving Ordinary Differential Equations I*, Berlin, Springer Series in Computational Mathematics, 1987.
- [GL18] A. GUZEL AND W. LAYTON, *Time filters increase accuracy of the fully implicit method*, BIT Numerical Mathematics, 58 (2018), 301-315.

- [HEPG15] A. HAY, S. ETIENNE, D. PELLETIER AND A. GARON, *hp-Adaptive time integration based on the BDF for viscous flows*. JCP, 291 (2015), 151-176.
- [J15] N. JIANG, *Higher Order Ensemble Simulation Algorithm for Fluid Flows*, Journal of Scientific Computing 64 (2015), 264-288.
- [J04] V. JOHN, *Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder*. International Journal for Numerical Methods in Fluids, 44(2004), 777–788.
- [JR10] V. JOHN, J. RANG, *Adaptive time step control for the incompressible Navier-Stokes equations*. Computer Methods in Applied Mechanics and Engineering, 199, 514–524, 2010.
- [JMRT17] N. JIANG, M. MOHEBUJAMAN, L. G. REBHOLZ, AND C. TRENCHIA, *An optimally accurate discrete regularization for second order timestepping methods for Navier–Stokes equations*, Computer Methods in Applied Mechanics and Engineering 310 (2016): 388-405.
- [KGG10] D.A. KAY, P.M. GRESHO, P.M., GRIFFITHS AND D.J. SILVESTER, *Adaptive time-stepping for incompressible flow Part II: Navier–Stokes equations*. SIAM Journal on Scientific Computing, 32(2010), 111-128.
- [K03] E. KALNAY, *Atmospheric Modeling, data assimilation and predictability*, Cambridge Univ. Press, Cambridge, 2003.
- [DK08] D.I. KETCHESON, *Highly efficient strong stability-preserving Runge–Kutta methods with low-storage implementations*, SIAM Journal on Scientific Computing 30 (4), 2113-2136.
- [LLT16] W. LAYTON, Y. LI, AND C. TRENCHIA, *Recent developments in IMEX methods with time filters for systems of evolution equations*, J. Comp. Applied Math. 299 (2016), 50–67.
- [LT15] Y. LI AND C. TRENCHIA, *Analysis of time filters used with the leapfrog scheme*, technical report, 2015, available at: <http://www.mathematics.pitt.edu/research/technical-reports>.

- [N17] L. NAJMAN, *Modern approaches to discrete curvature*, Lecture Notes in Mathematics, Springer, Berlin, 2017.
- [N84] O. NEVANLINNA, *Some remarks on variable step integration*, Z. Angew. Math. Mech. 64(1984)315-316.
- [R69] A. ROBERT, *The integration of a spectral model of the atmosphere by the implicit method*, Proc. WMO/IUGG Symposium on NWP, Japan Meteorological Soc. , Tokyo, Japan, pp. 19-24, 1969.
- [S10] M. SUSSMAN, *A stability example*, technical report, 2010, available at:  
*<http://www.mathematics.pitt.edu/sites/default/files/research-pdfs/stability.pdf>*.
- [ST96] M. SCHAFER AND S. TUREK *Benchmark computations of laminar flow around a cylinder*, Flow Simulation with High-Performance Computers II. Notes on Numerical Fluid Mechanics, 48 (1996), 547–566.
- [VV13] A. VENEZIANI AND U. VILLA, *ALADINS: An algebraic splitting time-adaptive solver for the incompressible Navier-Stokes equations*, JCP 238(2013) 359-375.
- [W11] P.D. WILLIAMS, *The RAW Filter: An Improvement to the Robert–Asselin Filter in Semi-Implicit Integrations*, Mon. Weather Rev., 139 (2011) 1996–2007.
- [W13] P.D. WILLIAMS, *Achieving seventh-order amplitude accuracy in leapfrog integration*, Monthly Weather Review 141.9 (2013), 3037-3051.
- [W09] P.D. WILLIAMS, *A proposed modification to the Robert-Asselin time filter*, Monthly Weather Review, 137(2009), 2538-2546.