The Search for Time Accuracy: A Variable Time-stepping Algorithm For Computational Fluid Dynamics

by

Wenlong Pei

Bachelor of Economics in Finance, Shanghai University of International Business and Economics, 2011

M.S. in Applied Financial Mathematics, University of Connecticut, 2014

Submitted to the Graduate Faculty of

the Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Wenlong Pei

It was defended on

July 13th 2022

and approved by

William Layton, Department of Mathematics, University of Pittsburgh

Catalin Trenchea, Department of Mathematics, University of Pittsburgh

Ivan Yotov, Department of Mathematics, University of Pittsburgh

Martina Bukač, Department of Applied and Computational Mathematics and Statistics, University of Notre Dame

Dissertation Director: William Layton, Department of Mathematics, University of Pittsburgh

Dissertation Co-advisor: Catalin Trenchea, Department of Mathematics, University of Pittsburgh

The Search for Time Accuracy: A Variable Time-stepping Algorithm For Computational Fluid Dynamics

Wenlong Pei, PhD

University of Pittsburgh, 2022

Dahlquist, Liniger, and Nevanlinna proposed a two-step time-stepping scheme for systems of ordinary differential equations (ODEs) in 1983. The little-explored variable time-stepping scheme has advantages in numerical simulations for its fine properties such as unconditional G-stability and second-order accuracy. However, this numerical scheme is always avoided for time discretization due to its complex form. To solve this issue, we simplify its implementation through time filters (pre-filter and post-filter) on a certain first-order implicit method. By adding only a few lines of code, accuracy will be improved while stability is not sacrificed.

G-stability of the scheme for systems of ODEs corresponds to unconditional, long-time energy stability when applied to flow models. The combination of G-stability and consistency provides the preliminaries for error analysis. We analyze the method of Dahlquist, Liniger, and Nevanlinna (DLN) as a variable step, time discretization of the unsteady Stokes/Darcy model, and the Navier-Stokes equations. We prove that the kinetic energy is bounded for variable time-steps, show that the method is second-order accurate, characterize its numerical dissipation and prove error estimates.

Moreover, the adaptivity algorithm for this variable time-stepping scheme, highly reducing computation cost as well as keeping time accuracy, has been applied to systems of ODEs and flow models. The local truncation error criterion for adapting time steps, with corresponding error estimators, is known in ODE problems. Many methods of error estimation are possible; herein we focus on ones that involve minimal extra storage and computations. First, we extend a classic and highly the efficient idea of Gear from the trapezoid rule to the DLN method. Second, we consider a recent refactorization of the DLN method which eases the implementation of DLN in legacy codes. We show that this refactorization provides methods for effective error estimation, at no extra cost.

For fluid models, the minimum numerical dissipation criterion is used for adjusting the

time steps, as the estimator of the local truncation error would be more complicated. The 2D offset circles problem by this algorithm is used to confirm the stability of the approximate solutions.

Keywords: computational fluid dynamics, variable time-stepping, G-stability, second-order convergence, time adaptivity.

Table of Contents

List of Tables

List of Figures

Preface

I want to express my sincere gratitude to all the people who gave me help and encouragement in the pursuit of the Ph.D. degree.

I would like to thank my advisor, Dr. William Layton for his fantastic ideas in computational fluid dynamics. In addition, he taught me how to do research efficiently, and become a qualified researcher.

I am grateful to my co-advisor, Dr. Catalin Trenchea for helping me check the details of the proofs, computing, and programming in my research.

Moreover, I appreciate Dr. Ivan Yotov and Dr. Martina Bukač for serving on my dissertation committee and providing suggestions on my research work.

Last but not least, I would like to thank my parents for their constant and essential support in such a long and exciting journey.

1.0 Introduction

The accurate numerical simulation of flows of an incompressible, viscous fluid, with the accompanying complexities occurring in practical settings, is a problem where speed, memory and accuracy never seem sufficient. For time discretization (considered herein), many longer time simulations use constant step, low-order methods, and the remainder use the constant time step implicit midpoint or the trapezoidal schemes, e.g., [7,11,78,114] (often combined with fractional steps, or with ad hoc fixes to correct for oscillations due to lack of L-stability, [10,101,124]) or the BDF2 method [3,43,57,81,86,128]. Time *accuracy* requires time step *adaptivity* within the computational, space and cognitive complexity limitations of computational fluid dynamics (CFD). Beyond accuracy, adaptivity has the secondary benefit (depending on implementation) of reducing memory requirements and decreasing the number of floating-point operations.

The richness of scales of higher Reynolds number flows and the cost per step of their solution suggest a preference for A-stable (or even L-stable) multi-step methods called *Smart Integrators* in Gresho, Sani and Engelman [52, Section 3.16.4]. For *constant* time steps, a complete analysis of the general (2 parameter family) 2-step, A-stable *linear multi-step* method is performed in the 1979 book Girault and Raviart [48] but there is no analogous stability or convergence analysis for the important case of *variable* time steps. As an example of the challenges involved in variable steps, BDF2 (a popular member of that A-stable family) loses A-stability for increasing time steps, allowing non-physical energy growth. This BDF2 variable step instability is weak since 0-stability is preserved for smoothly varying time steps [8, 117]. Similarly, the trapezoidal method is unstable [33], [119, pp.181-182] when used with variable steps. Specifically, Nevanlinna and Liniger [98,99] gave a simple example where the trapezoidal rule (multi-step method) is unstable, but the Crank-Nicolson rule (its one-leg 'twin') is stable, for all problems of the form $y'(t) = \lambda(t)y(t), \mathrm{Re}\lambda(t) \leq 0$ and any step size sequence.

Numerical methods for evolution equations are designed based on accuracy and stability. The theory of both is highly developed for constant time steps and linear problems. Less is known for variable time steps and nonlinear problems. Dahlquist, Liniger and Nevanlinna in [33] proposed a one parameter δ-family of variable-step, one-leg, two-step methods (DLN), which are second-order accurate, and variable-step, nonlinearly, long-time stable. Its detailed specification (given in Section 2.1), is sufficiently Gordian to deter its use in complex applications, in which a method with DLN's excellent properties should be valued. Our preliminary work on adaptive time-stepping for flow problems [83, 107] shows that (DLN) has promise, motivating the work herein.

In Chapter 2, we show how (DLN) can be refactorized to be easily implemented in an intricate, possibly legacy/black-box code, without modifying the 'assemble and solve' portion. *Refactorization* generally means a restructuring of an existing algorithm without changing its behavior. The goal of refactorization is to reduce complexity by creating a simple and clean logical structure, improving implementation, code readability, source maintainability, and extensibility. While our refactorization can work for other base methods, to fix ideas for $y' = f(t, y)$, we consider a method based on the fully implicit Euler method

$$\frac{y^{\text{new}} - y^{\text{old}}}{t^{\text{new}} - t^{\text{old}}} = f\left(t^{\text{new}}, y^{\text{new}}\right). \tag{BE}$$

Figure 1 illustrates the implementation of the (DLN) method in Algorithm 1, by adding a pre-filter step to the data ahead of the nonlinear solver (BE), and a poster-filter step after the solver (BE). This algorithmic idea is our main contribution of Chapter 2 and our work builds on and is related to work in [29, 30, 32, 61, 76, 77].

To our knowledge, the variable time-stepping DLN method is the **only** two-step scheme which is non-linear stable and second-order accurate so that it would be an ideal candidacy for adaptivity. We address this herein in two ways. First, we adopt an ingenious idea of Gear [18, 47, 50, 54, 55] to the DLN method in Section 3.2. Second, we show that the refactorized DLN algorithm itself gives an efficient option for error estimation in Section 3.3. These are developed and compared with existing methods in numerical experiments in Section 3.4.

The dissertation is organized as follows. In Chapter 2, we propose the refactorization algorithm of the DLN method in Section 2.1 and explore some properties of the scheme in Section 2.2. We develop two ways of adapting the DLN algorithm and provide various numerical tests to show the efficiency and accuracy of the algorithms in Chapter 3. In Chapter 4, we apply the variable time-stepping DLN method to the unsteady Stokes/Darcy model in Section 4.3 and provide the stability and error analysis of the approximate solutions in Section 4.4, 4.5. Similarly, we prove the stability and the error convergence of the DLN solutions for the Navier Stokes equations(NSEs) in Section 5.4, 5.5 of Chapter 5. Our conclusions are supported by some numerical tests for flow problems. The constant time-stepping DLN algorithm is used to verify the second-order convergence of the solutions in Section 4.6.2, 5.6.1. The variable time-stepping scheme is used to show the unconditional, long-term energy stability in Section 4.6.1, 5.6.2. Moreover, we have tested the adaptive DLN algorithm for flow problems under the minimal numerical dissipation criterion in Section 5.6.3.

2.0 Refactorization of a variable step, unconditionally stable method of Dahlquist, Liniger and Nevanlinna

2.1 The DLN method and its refactorization

We consider a numerical approximation of the initial value problem

$$y'(t) = f\left(t, y(t)\right), \qquad y(0) = y_0. \tag{2.1}$$

at times $\{t_n\}_{n\geq 0}$, with the time step $k_n = t_{n+1} - t_n$. To present the method of [33], let $\varepsilon_n = (k_n - k_{n-1})/(k_n + k_{n-1})$ denote the step size variability and $\delta \in [0, 1]$ be an arbitrary parameter. The (DLN) method is a 2-step method with coefficients:

$$\begin{bmatrix} \alpha_2 \\ \\ \alpha_1 \\ \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\delta+1) \\ \\ -\delta \\ \\ \frac{1}{2}(\delta-1) \end{bmatrix}, \qquad \begin{bmatrix} \beta_2^{(n)} \\ \\ \beta_1^{(n)} \\ \\ \beta_0^{(n)} \end{bmatrix} = \begin{bmatrix} \frac{1}{4}\left(1 + \frac{1-\delta^2}{(1+\varepsilon_n\delta)^2} + \varepsilon_n^2 \frac{\delta(1-\delta^2)}{(1+\varepsilon_n\delta)^2} + \delta\right) \\ \\ \frac{1}{2}\left(1 - \frac{1-\delta^2}{(1+\varepsilon_n\delta)^2}\right) \\ \\ \frac{1}{4}\left(1 + \frac{1-\delta^2}{(1+\varepsilon_n\delta)^2} - \varepsilon_n^2 \frac{\delta(1-\delta^2)}{(1+\varepsilon_n\delta)^2} - \delta\right) \end{bmatrix}. \tag{2.2}$$

Note that $\{\alpha_\ell\}_{\ell=0}^2$ are step size independent, while $\{\beta_\ell^{(n)}\}_{\ell=0}^2$ are step size dependent. Define the average step size $\widehat{k}_n$ as follows:

$$\widehat{k}_n = \alpha_2 k_n - \alpha_0 k_{n-1} = \delta \frac{k_n - k_{n-1}}{2} + \frac{k_n + k_{n-1}}{2}. \tag{2.3}$$

The variable step DLN method of [33] as a one-leg [1] method is

$$\frac{\alpha_2 y_{n+1} + \alpha_1 y_n + \alpha_0 y_{n-1}}{\widehat{k}_n} = f\Big(\beta_2^{(n)} t_{n+1} + \beta_1^{(n)} t_n + \beta_0^{(n)} t_{n-1}, \beta_2^{(n)} y_{n+1} + \beta_1^{(n)} y_n + \beta_0^{(n)} y_{n-1}\Big). \tag{DLN}$$

[1] The 'one-leg' term was coined by Dahlquist in 1975 [28] to name the multistep methods which involve only one value of f in each step. In particular, the leapfrog and BDF methods are one-leg multistep methods.

Remark 1. *The* (DLN) *methods are indexed by the free parameter* $\delta \in [0,1]$. *When* $\delta = 1$, *the* (DLN) *method becomes the (implicit) midpoint rule [11, 15]*

$$\frac{y_{n+1} - y_n}{k_n} = f\Big(\frac{1}{2}(t_{n+1} + t_n), \frac{1}{2}(y_{n+1} + y_n)\Big), \tag{one-step midpoint}$$

while for $\delta = 0$, *the* (DLN) *method is the (implicit) midpoint rule with double time step*

$$\frac{y_{n+1} - y_{n-1}}{k_n + k_{n-1}} = f\Big(\frac{1}{2}(t_{n+1} + t_{n-1}), \frac{1}{2}(y_{n+1} + y_{n-1})\Big). \tag{two-step midpoint}$$

To reduce the complexity of implementing (DLN), we consider its implementation through pre- and post-processes of an implicit (backward) Euler method, described in Algorithm 1, and illustrated in Figure 1.

Theorem 2. *Algorithm 1 is equivalent to the* (DLN) *method.*

Proof. First using the notations (2.4), the post-processing step writes

$$y^{\text{new}} = \frac{1}{c_2^{(n)}} y_{n+1} - \frac{c_1^{(n)}}{c_2^{(n)}} y_n - \frac{c_0^{(n)}}{c_2^{(n)}} y_{n-1} = \beta_2^{(n)} y_{n+1} + \beta_1^{(n)} y_n + \beta_0^{(n)} y_{n-1}.$$

Using also the pre-processing relations, the backward Euler step in Algorithm 1

$$\frac{y^{\text{new}} - y^{\text{old}}}{(\Delta t)_n^{\text{BE}}} = f\left(t^{\text{new}}, y^{\text{new}}\right) \tag{DLN2BE}$$

translates to

$$\begin{aligned}
&\frac{1}{\widehat{k}_n}\left(\frac{1}{b^{(n)} c_2^{(n)}} y_{n+1} - \frac{1}{b^{(n)}}\Big(\frac{c_1^{(n)}}{c_2^{(n)}} + a_1^{(n)}\Big) y_n - \frac{1}{b^{(n)}}\Big(\frac{c_0^{(n)}}{c_2^{(n)}} + a_0^{(n)}\Big) y_{n-1}\right) \\
&= f\left(\beta_2^{(n)} t_{n+1} + \beta_1^{(n)} t_n + \beta_0^{(n)} t_{n-1}, \beta_2^{(n)} y_{n+1} + \beta_1^{(n)} y_n + \beta_0^{(n)} y_{n-1}\right).
\end{aligned}$$

Finally, by (2.4), this shows that the backward Euler-based Algorithm 1 yields the solution of the (DLN) method. $\square$

Algorithm 1: Refactorization of the (DLN) method

Input: y_n, y_{n-1} and t_{n-1}, t_n, t_{n+1} ;

`// Pre-process : interpolation`

`// Evaluate quantities in` (2.2) `and` (2.3)

$$\begin{aligned}
&\alpha_2 = \tfrac{1}{2}(\delta+1), \quad \alpha_1 = -\delta, \quad \alpha_0 = \tfrac{1}{2}(\delta-1),\\
&\beta_2^{(n)} = \tfrac{1}{4}\left(1 + \tfrac{1-\delta^2}{(1+\varepsilon_n\delta)^2} + \varepsilon_n^2\tfrac{\delta(1-\delta^2)}{(1+\varepsilon_n\delta)^2} + \delta\right),\\
&\beta_1^{(n)} = \tfrac{1}{2}\left(1 - \tfrac{1-\delta^2}{(1+\varepsilon_n\delta)^2}\right), \quad \beta_0^{(n)} = 1 - \beta_2^{(n)} - \beta_1^{(n)}, \quad \widehat{k}_n = \alpha_2 k_n - \alpha_0 k_{n-1}.
\end{aligned}$$

`// Define the refactorization coefficients`

$$\begin{cases} a_1^{(n)} = \beta_1^{(n)} - \dfrac{\alpha_1\beta_2^{(n)}}{\alpha_2}, \quad a_0^{(n)} = 1 - a_1^{(n)}, \quad b^{(n)} = \dfrac{\beta_2^{(n)}}{\alpha_2},\\ c_2^{(n)} = \frac{1}{\beta_2^{(n)}}, \quad c_1^{(n)} = -\frac{\beta_1^{(n)}}{\beta_2^{(n)}}, \quad c_0^{(n)} = -\frac{\beta_0^{(n)}}{\beta_2^{(n)}}. \end{cases} \tag{2.4}$$

`// Evaluate the time-step for BE`

$(\Delta t)_n^{\mathrm{BE}} \Leftarrow b^{(n)}\widehat{k}_n$

`// Set the BE time interval:` $[t^{\mathtt{new}} - (\Delta t)_n^{\mathtt{BE}}, t^{\mathtt{new}}]$`, and` $y_n^{\mathtt{old}}$

$t^{\mathrm{new}} \Leftarrow \beta_2^{(n)}t_{n+1} + \beta_1^{(n)}t_n + \beta_0^{(n)}t_{n-1}$; $\quad y^{\mathrm{old}} \Leftarrow a_1^{(n)}y_n + a_0^{(n)}y_{n-1}$;

`// backward Euler`

Solve for y^{new}: $\qquad \dfrac{y^{\mathrm{new}} - y^{\mathrm{old}}}{(\Delta t)_n^{\mathrm{BE}}} = f\left(t^{\mathrm{new}}, y^{\mathrm{new}}\right)$

`// Post-process : extrapolation`

$y_{n+1} \Leftarrow c_2^{(n)}y^{\mathrm{new}} + c_1^{(n)}y_n + c_0^{(n)}y_{n-1}$; `// the DLN solution`

Output: y_{n+1}, If desired: Estimate Error and adapt k_n

Since $\alpha_0 + \alpha_1 + \alpha_2 = 0, \beta_0^{(n)} + \beta_1^{(n)} + \beta_2^{(n)} = 1$, the coefficients $a_i^{(n)}, b^{(n)}, c_i^{(n)}$ satisfy

$$a_0^{(n)} + a_1^{(n)} = 1, \qquad c_2^{(n)} + c_1^{(n)} + c_0^{(n)} = 1.$$

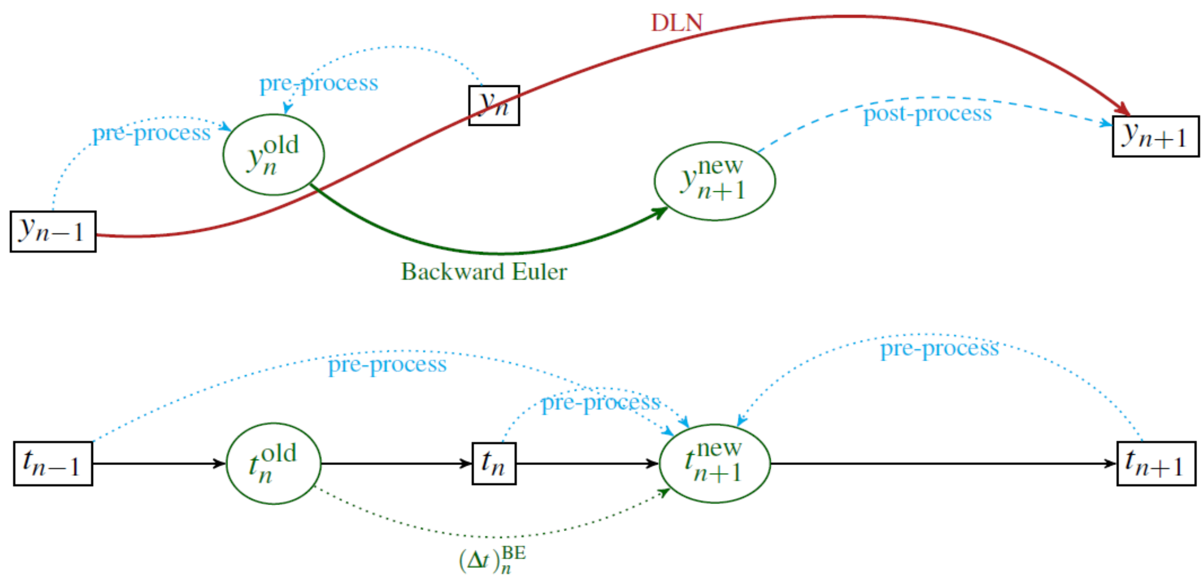


Figure 1: Refactorization of the (DLN) method as a pre- and post-processed (BE) method

2.1.1 Related Work

The (DLN) method is variable-step G-stable outgrowth of a method of Liniger [93], which is non-autonomous A-stable (i.e. for $y' = \lambda(t)y$). The pre- and post-process steps in the Algorithm 1 are akin to time filters, highly developed as numerical methods in atmospheric science [4, 60, 91, 109, 127]. Recently it was noticed in Guzel and Layton [58] that this technique for adding stability can also increase accuracy. The idea of prefilter $\rightarrow$ simple method $\rightarrow$ postfilter was developed in a different direction for constant time steps in [35]. The refactorization of an algorithm to reduce its cognitive complexity has been used in [15] to rearrange a family of one-leg one-step methods into a backward Euler code followed by post-processing, and further applied for partitioning multi-physics problems [11, 13, 14, 125]. In [126], the authors describe the implementation of the (DLN) formulas in a Nordsieck formulation [100, 115] essentially identical to that of the backward differentiation formulas, facilitating to adapt Nordsieck formulation codes like DIFSUB [46,47] to the (DLN) formulas.

2.2 Convergence analysis of one-leg DLN method

While stability and consistency were already addressed in [33], we present complementary details on both, which are useful for developing an adaptive (DLN) method.

2.2.1 Consistency error

In [76,77], the variable time-step (DLN) method was implemented in an adaptive manner, using a local and global error estimator. Similar to [33], the authors of [28, 76, 77, 126] use the classical definition of the local truncation error (LTE)

$$\mathcal{L}_1\big(y(t), t_{n+1}, k_n\big) = \frac{1}{\widehat{k}_n} \sum_{\ell=0}^{2} \alpha_\ell y(t_{n-1+\ell}) - f\left(t_{n,\beta}, \beta_2^{(n)} y(t_{n+1}) + \beta_1^{(n)} y(t_n) + \beta_0^{(n)} y(t_{n-1})\right),$$

where $t_{n,\beta} = \beta_2^{(n)} t_{n+1} + \beta_1^{(n)} t_n + \beta_0^{(n)} t_{n-1}$. The above definition follows the approach taken in the analysis of linear multi-step methods (see e.g. [79, page 27]), and involves both the differentiation defect $\mathcal{L}_d = \frac{1}{\widehat{k}_n} \sum_{\ell=0}^{2} \alpha_\ell y(t_{n-1+\ell}) - f\big(t_{n,\beta}, y(t_{n,\beta})\big)$, and the interpolation defect $\mathcal{L}_i = f\big(t_{n,\beta}, y(t_{n,\beta})\big) - f\big(t_{n,\beta}, \sum_{\ell=0}^{2} \beta_\ell^{(n)} y(t_{n-1+\ell})\big)$. Dahlquist raised in [31] the question of the appropriateness of this viewpoint: "We accept this definition, but *we do not accept* $\mathcal{L}_1$ *as the adequate local truncation error!*" Using the refactorized form (DLN2BE) and Theorem 2, we now prove that the local truncation error of the one-leg (DLN) method can be evaluated only by the differentiation defect (LTE), similarly to the midpoint rule [15] and the Runge-Kutta methods. The new expression (LTE) simplifies greatly the error estimation.

Proposition 1. *The local truncation error of* (DLN) *is the differentiation error and*

$$\mathcal{L}_d\big(y(t), t_{n+1}, k_n\big) \approx \frac{y'''(t_n)}{2} \Big[\frac{1}{3\widehat{k}_n} \big(k_n^3 - \frac{\alpha_0}{\alpha_2} k_{n-1}^3\big) - \frac{1}{\alpha_2} \big(\beta_2^{(n)} k_n - \beta_0^{(n)} k_{n-1}\big)^2 \Big]. \tag{LTE}$$

Proof. The consistency order and the coefficient of the leading term in (LTE) follow by Taylor expansions. On one hand, since (DLN) can be refactorized as the one-step method (DLN2BE), we further write the (DLN) method as follows

$$\frac{\alpha_2 y_{n+1} + \alpha_1 y_n + \alpha_0 y_{n-1}}{\widehat{k}_n} = f\left(t^{\text{new}}, y^{\text{new}}\right). \tag{2.5}$$

On the other hand, when we integrate (2.1) on $[t_{n-1}, t_n]$ and on $[t_n, t_{n+1}]$, multiply the results by $\frac{1-\delta}{2}$ and $\frac{1+\delta}{2}$, respectively, and add, we obtain

$$\tfrac{1+\delta}{2}y(t_{n+1}) - \delta y(t_n) - \tfrac{1-\delta}{2}y(t_{n-1}) = \tfrac{1-\delta}{2}\int_{t_{n-1}}^{t_n} f(t, y(t))\,dt + \tfrac{1+\delta}{2}\int_{t_n}^{t_{n+1}} f(t, y(t))\,dt.$$

Finally, approximating both integrals on the right-hand-side with the chord quadrature rule, with t^{new} as the point of evaluation on both intervals, gives

$$\tfrac{1+\delta}{2}y(t_{n+1}) - \delta y(t_n) - \tfrac{1-\delta}{2}y(t_{n-1}) \approx \Big(\tfrac{1+\delta}{2}(t_{n+1} - t_n) - \tfrac{\delta-1}{2}(t_n - t_{n-1})\Big) f(t^{\text{new}}, y(t^{\text{new}})).$$

which by (2.2)-(2.3) yields the (2.5) method. □

Remark 3. *In particular, for $\delta = 1$ and $\delta = 0$, from* (LTE) *we have that*

$$\mathcal{L}^{(\text{one-step midpoint})} \approx \tfrac{1}{24}k_n^3 y'''(t_n), \qquad \mathcal{L}^{(\text{two-step midpoint})} \approx \tfrac{1}{24}\left(k_n + k_{n-1}\right)^3 y'''(t_n).$$

2.2.2 G-stability

Let $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the inner product and ℓ^2-norm in Euclidean space $\mathbb{C}^d$. For any pair of solutions $u(t), v(t)$ of (2.1), a necessary and sufficient condition [30, page 384] for $\|u(t) - v(t)\|$ to be a non-increasing function of t is the *contractivity* (one-sided Lipschitz) condition on f:

$$\text{Re}\big\langle f(t,u) - f(t,v), u - v\big\rangle \le 0, \qquad \forall t \ge 0, \quad \forall u, v \in \mathbb{C}^d. \qquad \text{(contractivity)}$$

The system (2.1) for which f satisfies the (contractivity) condition is called *dissipative*, see e.g. the Definition in [80, page 268]. We recall that a Runge-Kutta method is B-stable, if the (contractivity) condition implies $\|y_{n+1} - z_{n+1}\| \le \|y_n - z_n\|$ for any $\{y_n\}, \{z_n\}$ numerical solutions, see e.g. [17, page 359], or Definition 12.2 in [62]. Similarly, a 2-step linear multistep method is called G-stable [29, 30, 32, 62] if there exists a real positive definite matrix G such that its one-leg version is contractive, namely $\|Y_{n+1} - Z_{n+1}\|_G \le \|Y_n - Z_n\|_G$, where

$Y_n = [y_n^{tr}, y_{n-1}^{tr}]^{tr}$. In the case of the (DLN) method, there exists such a positive definite matrix (independent of the step size)

$$G(\delta) := \begin{bmatrix} \frac{1}{4}(1+\delta)\mathbb{I}_d & 0 \\ 0 & \frac{1}{4}(1-\delta)\mathbb{I}_d \end{bmatrix}, \qquad \forall \delta \in [0,1]. \tag{2.6}$$

As pointed out by Dahlquist in [27], both B-stability and G-stability imply A-stability, and A-stability implies G-stability for constant time steps.

Proposition 2. *The* (DLN) *method is unconditionally G-stable, and*

$$\Big\langle \sum_{\ell=0}^{2} \alpha_\ell y_{n-1+\ell}, \sum_{\ell=0}^{2} \beta_\ell^{(n)} y_{n-1+\ell} \Big\rangle_{\mathbb{R}^d} = \left\| \begin{matrix} y_{n+1} \\ y_n \end{matrix} \right\|_{G(\delta)}^2 - \left\| \begin{matrix} y_n \\ y_{n-1} \end{matrix} \right\|_{G(\delta)}^2 + \Big\| \sum_{\ell=0}^{2} \gamma_\ell^{(n)} y_{n-1+\ell} \Big\|^2, \tag{2.7}$$

where the γ-coefficients are $\gamma_1^{(n)} = -\frac{\sqrt{\delta(1-\delta^2)}}{\sqrt{2}(1+\varepsilon_n\delta)}$, $\gamma_2^{(n)} = -\frac{1-\varepsilon_n}{2}\gamma_1^{(n)}$, $\gamma_0^{(n)} = -\frac{1+\varepsilon_n}{2}\gamma_1^{(n)}$, and the corresponding G-norm

$$\left\| \begin{matrix} u \\ v \end{matrix} \right\|_{G(\delta)}^2 := \begin{bmatrix} u^{tr} & v^{tr} \end{bmatrix} G(\delta) \begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{4}(1+\delta)\|u\|_{\mathbb{R}^d}^2 + \frac{1}{4}(1-\delta)\|v\|_{\mathbb{R}^d}^2, \tag{2.8}$$

[2] *for any $u, v \in \mathbb{R}^d$.*

The 'energy' identity (2.7), implicit in [33], follows from algebraic manipulations, see e.g. [83]. The G-stability of (DLN), i.e. $\|Y_{n+1} - Z_{n+1}\|_{G(\delta)} \leq \|Y_n - Z_n\|_{G(\delta)}$, follows from (2.7) and the (contractivity) assumption.

Remark 4. *The only* (DLN) *methods which yield the ℓ^2 invariance of the solution are the symplectic* (one-step midpoint) *and* (two-step midpoint) *rules: the numerical dissipation $\Big\| \sum_{\ell=0}^{2} \gamma_\ell^{(n)} y_{n-1+\ell} \Big\|$ vanishes if and only if $\delta \in \{0, 1\}$.*

[2] The symbol 'tr' means the transpose of the vectors or matrices.

3.0 Time step adaptivity in the method of Dahlquist, Liniger and Nevanlinna

3.1 Introduction

Many numerical methods have been developed for approximate solutions of systems of time-dependent differential equations, e.g., [3, 56, 61, 62]. In complex applications it is however still common to use simple methods such as constant time step backward Euler, the midpoint rule, the trapezoid rule, or increasingly, the BDF2 method [48, 65, 69, 70, 92]. For variable steps, the trapezoid rule loses stability for some specific preset steps [33] and BDF2 is unstable if the step ratio is relatively large. In 1983, Dahlquist, Liniger, and Nevanlinna designed a one-parameter family of one-leg, two-step schemes (henceforce the DLN method) in [33], which is G-stable (non-linearly stable) [29, 30, 32] and second-order accurate for variable steps, with arbitrary step-size ratios. Its ability to increase step sizes rapidly, when local solution behavior warrants, is an attractive feature for large (storage limited) systems. Our simple tests of the variable step DLN method presented in Chapter 4, 5, and in the related paper [83,107] have confirmed its potential. This work is complemented by this thesis' further analysis in Chapter 4, 5. Also see [83, 107] and the refactorization in Chapter 2, to reduce the cognitive complexity of the (DLN) implementation in [84]. One main remaining question addressed herein is how to estimate local errors, and adapt the time step, within the limitations on computational, space, and cognitive complexity in large applications. We propose two options to deal with this issue. In Section 3.2, we extend the ingenious idea of Gear [18, 47, 50, 54, 55] to the DLN method. In Section 3.3, we show that the refactorized DLN algorithm itself gives an efficient option for error estimation. We test these algorithms and compare them with other existing methods in numerical experiments presented in Section 3.4.

The DLN Method. To begin, consider the initial value problem

$$y'(t) = f(t, y(t)), \quad y(0) = y_0, \tag{3.1}$$

where $y : [0, \infty) \to \mathbb{R}^d$ and $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ are two certain mappings and y_0 is the given vector in $\mathbb{R}^d$. Let $\{t_n\}_{n=0}^N$ be the grid on the time interval $[0, T]$ and $k_n = t_n - t_{n-1}$ be the local time step. The variable step, one parameter family (with the parameter $\delta \in [0, 1]$) of the DLN method applying to (3.1) reads

$$\frac{\alpha_2 y_{n+1} + \alpha_1 y_n + \alpha_0 y_{n-1}}{\widehat{k}_n} = f\Big(\beta_2^{(n)} t_{n+1} + \beta_1^{(n)} t_n + \beta_0^{(n)} t_{n-1}, \beta_2^{(n)} y_{n+1} + \beta_1^{(n)} y_n + \beta_0^{(n)} y_{n-1}\Big),$$

where the coefficients α_ℓ and β_ℓ are

$$\begin{bmatrix} \alpha_2 \\ \\ \alpha_1 \\ \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\delta + 1) \\ \\ -\delta \\ \\ \frac{1}{2}(\delta - 1) \end{bmatrix}, \qquad \begin{bmatrix} \beta_2^{(n)} \\ \\ \beta_1^{(n)} \\ \\ \beta_0^{(n)} \end{bmatrix} = \begin{bmatrix} \frac{1}{4}\Big(1 + \frac{1-\delta^2}{(1+\varepsilon_n\delta)^2} + \varepsilon_n^2 \frac{\delta(1-\delta^2)}{(1+\varepsilon_n\delta)^2} + \delta\Big) \\ \\ \frac{1}{2}\Big(1 - \frac{1-\delta^2}{(1+\varepsilon_n\delta)^2}\Big) \\ \\ \frac{1}{4}\Big(1 + \frac{1-\delta^2}{(1+\varepsilon_n\delta)^2} - \varepsilon_n^2 \frac{\delta(1-\delta^2)}{(1+\varepsilon_n\delta)^2} - \delta\Big) \end{bmatrix}. \tag{3.2}$$

The step size variability ε_n and average step $\widehat{k}_n$ are

$$\varepsilon_n = (k_n - k_{n-1})/(k_n + k_{n-1}) \in (-1, 1), \qquad \widehat{k}_n = \alpha_2 k_n - \alpha_0 k_{n-1}. \tag{3.3}$$

From (2.2) and (3.3), the left coefficients α_ℓ are functions of δ. The right coefficients $\beta_\ell^{(n)}$ are functions of δ and ε_n and the average step size $\widehat{k}_n$ constructed to ensure the second order accuracy. The DLN method is reduced to the one-step, and two-step midpoint rule, respectively, when $\delta = 1$ and $\delta = 0$.

Time step adaptivity (adjusting time step size according to the required accuracy) is an essential algorithm in the numerical simulation since it improves time accuracy and keeps the computational cost relatively low. To our knowledge, the variable time-stepping DLN method is the **only** two-step scheme which is non-linear stable, and second-order accurate so it would be an ideal candidate for adaptivity. The adaptivity process involves two issues: estimator for the error and time step controller. Milne [96] was the first to estimate the LTE with two solution approximations, by different methods of the same order but different error constants. The method was also used by Gear [47], Shampine and Gordon [111]. Gresho cited Gear, Shampine, and Gordon and estimated local truncation error (LTE) of trapezoid rule via AB2 method (two-step explicit Adams method) in the numerical simulation of NSEs

[50, 54, 55]. This chapter extends Gresho's idea and utilizes some commonly used explicit schemes to derive estimators for local truncation error (LTE) of the DLN method. This method provides an asymptotically accurate estimate of the local error at nearly minimal extra work and storage.

The second method we consider requires no extra work or storage but gives a pessimistic estimator of local errors. The report [58] tested an adaptive algorithm in which the local truncation error is estimated by the difference between the BE solution and the time filter solution. This method requires no extra storage or calculations but is pessimistic since it uses the difference between first and second-order approximations to adapt the time step for the second-order method. Motivated by this idea, we exploit the first-order approximation embedded in the refactorized DLN method in [15, 84] to adopt. For the time step controller, we adopt the controller proposed in [62], which removes step size restriction and enhances algorithmic robustness.

The chapter is organized as follows: In Section 3.2, we present the adaptivity of the DLN method with the estimator of LTE by certain explicit schemes. Self-adaptivity algorithm by refactorization process is introduced in Section 3.3. Numerical tests, especially tests for adaptivity of variable step DLN method in Section 3.4.2 imply the efficiency of the adaptive algorithms in this chapter.

3.1.1 Related Work

Gresho, Griffiths Silvester [53] applied adaptivity for trapezoidal method and AB2 pairs to scalar advection-diffusion model. Based on this work, Kay, Gresho, Griffiths, and Silvester [75] combined this adaptivity algorithm with robust Krylov subspace solvers for the NSEs. Bukač and Trenchea [12] designed a time adaptive, strongly-coupled partitioned method for fluid and structure interation [6,11,16,34,95,103,110]. Recently Capuano, Sanderse, Angelis, and Coppola proposed a new adaptive algorithm with the numerical dissipation criterion for incompressible viscous flows [20]. Adaptivity based on their minimal dissipation criterion has been tested for the DLN method in [83]. The idea of embedding methods using time filters in [58] has been developed in various directions, e.g., [1, 26, 36, 89, 105, 129], including into VSVO methods in [37].

3.2 Adaptivity by Explicit Schemes

Adaptivity requires a reliable estimator such as the difference of the solutions by DLN and a higher-order method. This typically introduces more time levels and the stability of the higher-order explicit method must be considered.

The first estimator. To address these, we adapt Milnes device [96] for the estimator of the LTE, using a variable two-step Adams-Bashforth method (AB2) [18, 47, 50, 56]:

$$y_{n+1} = y_n + \frac{k_n}{2}\Big[\big(2 + \frac{k_n}{k_{n-1}}\big)f(t_n, y_n) - \frac{k_n}{k_{n-1}}f(t_{n-1}, y_{n-1})\Big],$$

and its corresponding LTE

$$y_{n+1}^{\mathrm{AB2}} - y(t_{n+1}) = -\big(\frac{1}{6} + \frac{1}{4\tau_n}\big)y'''(t_n)k_n^3 + \cdots , \tag{3.4}$$

where y_{n+1}^{AB2} and $y(t_{n+1})$ are the AB2 approximate solution, and the exact solution, at time t_{n+1}, and $\tau_n = k_n/k_{n-1}$ is the ratio of steps. From [84, Proposition 3.1 on page 5], the LTE for the variable step DLN method is

$$y_{n+1}^{\mathrm{DLN}} - y(t_{n+1}) = G^{(n)}y^{'''}(t_n)k_n^3 + \cdots , \tag{3.5}$$

where y_{n+1}^{DLN} is the approximate solution by DLN method and the coefficient $G^{(n)}$ for the LTE is

$$G^{(n)} = \Big(\frac{1}{2} - \frac{\alpha_0}{2\alpha_2}\frac{1-\varepsilon_n}{1+\varepsilon_n}\Big)\Big(\beta_2^{(n)} - \beta_0^{(n)}\frac{1-\varepsilon_n}{1+\varepsilon_n}\Big)^2 + \frac{\alpha_0}{6\alpha_2}\Big(\frac{1-\varepsilon_n}{1+\varepsilon_n}\Big)^3 - \frac{1}{6}.$$

Subtracting (3.4) from (3.5)

$$y_{n+1}^{\mathrm{DLN}} - y_{n+1}^{\mathrm{AB2}} = \Big[G^{(n)} + \Big(\frac{1}{6} + \frac{1}{4\tau_n}\Big)\Big]y'''(t_n)k_n^3 + \cdots \quad \Leftrightarrow \quad y'''(t_n)k_n^3 = \frac{y_{n+1}^{\mathrm{DLN}} - y_{n+1}^{\mathrm{AB2}}}{G^{(n)} + \big(\frac{1}{6} + \frac{1}{4\tau_n}\big)} + \cdots . \tag{3.6}$$

Combining (3.5) and (3.6), we have the estimator for LTE of DLN method by AB2 method (denoted by $\widehat{T}_{n+1}$)

$$\textbf{Estimator 1:} \quad \widehat{T}_{n+1} = \left\| \frac{G^{(n)}}{G^{(n)} + (\frac{1}{6} + \frac{1}{4\tau_n})} (y_{n+1}^{\mathrm{DLN}} - y_{n+1}^{\mathrm{AB2}}) \right\|, \tag{3.7}$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^d$.

The Second Estimator. The critical issue for the above technique is that we need a second-order, explicit time-stepping scheme for which the LTE takes the form LTE $= Ck_n^3 + \cdots$. According to this principle, we have many other choices. We develop this next for an explicit, two-step scheme. Recall the variable step BDF2 method

$$\frac{1+2\tau_n}{1+\tau_n} y_{n+1} - (1+\tau_n) y_n + \frac{\tau_n^2}{1+\tau_n} y_{n-1} = k_n f(t_{n+1}, y_{n+1}). \tag{3.8}$$

To derive a second-order explicit scheme, We approximate the right part of (3.8) by second-order linear extrapolation, i.e.

$$f(t_{n+1}, y_{n+1}) \approx (1+\tau_n) f(t_n, y_n) - \tau_n f(t_{n-1}, y_{n-1}),$$

and we obtain the explicit scheme

$$\frac{1+2\tau_n}{1+\tau_n} y_{n+1} - (1+\tau_n) y_n + \frac{\tau_n^2}{1+\tau_n} y_{n-1} = k_n \Big[(1+\tau_n) f(t_n, y_n) - \tau_n f(t_{n-1}, y_{n-1}) \Big]. \tag{3.9}$$

The LTE of the scheme in (3.9) and the estimator of LTE by this scheme are

$$y_{n+1}^{\mathrm{ExBDF2}} - y(t_{n+1}) = - \frac{(1+\tau_n)^2}{3\tau_n(1+2\tau_n)} y'''(t_n) k_n^3 + \cdots,$$

$$\textbf{Estimator 2:} \quad \widehat{T}_{n+1} = \left\| \frac{G(\varepsilon_n)}{G(\varepsilon_n) + \frac{(1+\tau_n)^2}{3\tau_n(1+2\tau_n)}} (y_{n+1}^{\mathrm{DLN}} - y_{n+1}^{\mathrm{ExBDF2}}) \right\|. \tag{3.10}$$

Time-step Controllers. The main principle for adaptivity is to adjust the step size such that the estimator of LTE by DLN is less than or equal to the tolerance (Tol) [56]. The basic step-size controller for next step size k_{n+1} is

$$\textbf{Basic Controller:} \quad k_{n+1} = \kappa k_n \Big(\mathrm{Tol} / \|\widehat{T}_{n+1}\| \Big)^{1/3}, \tag{3.11}$$

where the safety factor $\kappa \in (0,1]$ is selected to minimize the number of step rejections. At each time-step computing, if $\left\|\widehat{T}_{n+1}\right\| > \mathrm{Tol}$, then the solution at current time is rejected and the current step k_n is adjusted by (3.11) for recalculation. For the robustness of computation, we may employ the floor for step size k_n, especially for stiff problems. To remove the limitation on the step size to a large extent, the following improved time step controller, based on the basic step controller in (3.11), was proposed by Hairer, Nørsett and Wanner in [62]

$$\textbf{Improved Basic Controller:} \quad k_{n+1} = k_n \cdot \min\Big\{1.5, \max\Big\{0.2, \kappa\big(\mathrm{Tol}/\|\widehat{T}_{n+1}\|\big)^{1/3}\Big\}\Big\}. \tag{3.12}$$

Another way of avoiding the restriction, proposed by Söderlind and Wang in [116, 118], replaces the term $(\mathrm{Tol}/\|\widehat{T}_{n+1}\|)$ in (3.11) by taking geometric average of $(\mathrm{Tol}/\|\widehat{T}_{n+1}\|)$, $(\mathrm{Tol}/\|\widehat{T}_{n}\|)$ and $(\mathrm{Tol}/\|\widehat{T}_{n-1}\|)$, i.e.

$$\textbf{Generalized Controller:} \quad k_{n+1} = k_n\big(\mathrm{Tol}/\|\widehat{T}_{n+1}\|\big)^{\lambda_1}\big(\mathrm{Tol}/\|\widehat{T}_{n}\|\big)^{\lambda_2}\big(\mathrm{Tol}/\|\widehat{T}_{n-1}\|\big)^{\lambda_3}\tau_n^{-\eta_2}\tau_{n-1}^{-\eta_3}.$$

The values of λ and η are decided by the order of dynamics of the closed-loop system [116]. To end this section, we summarize the adaptivity algorithm of the DLN method with **Estimator 1** (3.7) or **Estimator 2** (3.10) of LTE and the step controller (3.12) in Algorithm 2.

3.3 Adaptivity by the DLN Refactorization Process

In this section, we will present another adaptive algorithm, via refactorization of the DLN method with parameter $\delta \in (0,1)$ and midpoint rule. Recall the refactorization of the DLN method in [84]: given previous two step solutions y_{n-1}, y_n and time grid t_{n-1}, t_n, t_{n+1},

Step 1 (Pre-process):

$$(\Delta t)_n^{\mathrm{BE}} = b^{(n)}\widehat{k}_n, \quad t_{n+1}^{\mathrm{new}} = \beta_2^{(n)} t_{n+1} + \beta_1^{(n)} t_n + \beta_0^{(n)} t_{n-1}, \quad y_n^{\mathrm{old}} = a_1^{(n)} y_n + a_0^{(n)} y_{n-1},$$

Step 2 (BE solver):

$$\frac{y_{n+1}^{\mathrm{new}} - y_n^{\mathrm{old}}}{(\Delta t)_n^{\mathrm{BE}}} = f\big(t_{n+1}^{\mathrm{new}}, y_{n+1}^{\mathrm{new}}\big), \qquad \text{(BE solver on interval } [t_{n+1}^{\mathrm{new}} - (\Delta t)_n^{\mathrm{BE}}, t_{n+1}^{\mathrm{new}}])$$

Algorithm 2: Adaptivity with **Estimator 1** or **Estimator 2** of LTE and step size controller in (3.12)

Input: tolerance Tol, initial value y_1, initial stepsize k_1, safety factor κ, time interval $[T_1, T_2]$;

$n \Leftarrow 1; \quad t_n \Leftarrow T_1$;

$t_{n+1} \Leftarrow t_n + k_n$; `// update the current time`

compute y_{n+1} by one-step method (e.g. backward Euler);

$n \Leftarrow n+1$, $k_n \Leftarrow k_{n-1}$;

while $t_n + k_n < T_2$ **do**

  $t_{n+1}^{\text{temp}} = t_n + k_n; \quad \tau_n = k_n / k_{n-1}$; `// update current time and step size ratio`

  compute y_{n+1}^{DLN} ; `// find the DLN solution`

  compute y_{n+1}^{AB2} if **Estimator 1** used or compute y_{n+1}^{ExBDF2} if **Estimator 2** used ;

  $\widehat{T}_{n+1} \Leftarrow \left\| \frac{G^{(n)}}{G^{(n)} + (\frac{1}{6} + \frac{1}{4\tau_n})} (y_{n+1}^{\text{DLN}} - y_{n+1}^{\text{AB2}}) \right\|$ or $\widehat{T}_{n+1} \Leftarrow \left\| \frac{G(\varepsilon_n)}{G(\varepsilon_n) + \frac{(1+\tau_n)^2}{3\tau_n(1+2\tau_n)}} (y_{n+1}^{\text{DLN}} - y_{n+1}^{\text{ExBDF2}}) \right\|$;

  if $\widehat{T}_{n+1} < \text{Tol}$ **then**

    $t_{n+1} \Leftarrow t_{n+1}^{\text{temp}}$; `// accept current estimator for LTE`

    $k_{n+1} \Leftarrow k_n \cdot \min\left\{1.5, \max\left\{0.2, \kappa\left(\text{Tol}/\|\widehat{T}_{n+1}\|\right)^{1/3}\right\}\right\}$; `// adjust step by (3.12)`

    $y_{n+1} \Leftarrow y_{n+1}^{\text{DLN}}$; `// accept result`

    $n \Leftarrow n+1$; `// come to next time step`

  else

    $k_n \Leftarrow k_n \cdot \min\left\{1.5, \max\left\{0.2, \kappa\left(\text{Tol}/\|\widehat{T}_{n+1}\|\right)^{1/3}\right\}\right\}$; `// adjust step for recomputing`

Step 3 (Post-process):

$$y_{n+1}^{\text{DLN}} = c_2^{(n)} y_{n+1}^{\text{new}} + c_1^{(n)} y_n + c_0^{(n)} y_{n-1}, \quad (\text{from } t_{n+1}^{\text{new}} \text{ to } t_{n+1} \text{ to obtain the DLN solution})$$

where the coefficients in the DLN refactorization process are

$$\begin{gathered}
a_1^{(n)} = \beta_1^{(n)} - \frac{\alpha_1 \beta_2^{(n)}}{\alpha_2}, \quad a_0^{(n)} = \beta_0^{(n)} - \frac{\alpha_0 \beta_2^{(n)}}{\alpha_2}, \quad b^{(n)} = \frac{\beta_2^{(n)}}{\alpha_2}, \\
c_2^{(n)} = \frac{1}{\beta_2^{(n)}}, \quad c_1^{(n)} = -\frac{\beta_1^{(n)}}{\beta_2^{(n)}}, \quad c_0^{(n)} = -\frac{\beta_0^{(n)}}{\beta_2^{(n)}}.
\end{gathered}$$

It's easy to check

$$t_{n+1} - t_{n+1}^{\text{new}} = (\Delta t)_n^{\text{BE}},$$

which induces approximate solutions at t_{n+1} by the midpoint rule. Recall the refactorization of the midpoint rule on interval $[t_n, t_{n+1}]$ (see [15])

$$\frac{y_{n+1}^{\text{mid}} - y_n}{k_n} = f\Big(t_{n+1/2}, \frac{1}{2} y_{n+1} + \frac{1}{2} y_n\Big), \quad \Leftrightarrow \quad \begin{cases} \dfrac{y_{n+1/2} - y_n}{t_{n+1/2} - t_n} = f(t_{n+1/2}, y_{n+1/2}), \\ y_{n+1/2} = \dfrac{1}{2} y_n + \dfrac{1}{2} y_{n+1}^{\text{mid}}, \end{cases} \tag{3.13}$$

where $t_{n+1/2}$ is the midpoint between t_n and t_{n+1}, $y_{n+1/2}$ the BE solution at time $t_{n+1/2}$ and y_{n+1}^{mid} the midpoint solution at time t_{n+1}. After we finish step 2 in the DLN refactorization process, we obtain another approximate solution $\widetilde{y}_{n+1}$ by the midpoint algorithm in (3.13), i.e.

$$\widetilde{y}_{n+1} = 2 y_{n+1}^{\text{new}} - y_n^{\text{old}}, \quad \Leftrightarrow \quad y_{n+1}^{\text{new}} = \frac{1}{2} y_n^{\text{old}} + \frac{1}{2} \widetilde{y}_{n+1}. \tag{3.14}$$

For the post-process in (3.14), we replace k_n by $t_{n+1} - 2(\Delta t)_n^{\text{BE}}$, y_n by y_n^{old} and $y_{n+1/2}$ by y_{n+1}^{new} in the algorithm (3.13) and obtain $\widetilde{y}_{n+1}$. We will show that $\widetilde{y}_{n+1}$ is exact first order approximation to $y(t_{n+1})$ in Theorem 5 and thus the difference between y_{n+1}^{DLN} (second order approximation) and $\widetilde{y}_{n+1}$ works as one estimator of LTE. Estimator of LTE is then

$$\textbf{Estimator 3:} \quad \widehat{T}_{n+1} = \| y_{n+1}^{\text{DLN}} - \widetilde{y}_{n+1} \|. \tag{3.15}$$

We summarize the refactorization process of two solutions y_{n+1}^{DLN} (the DLN solution) and $\widetilde{y}_{n+1}$ in Algorithm 3 and Figure 2

Algorithm 3: The estimator of LTE by the Refactorization Algorithm

---

Input: y_n, y_{n-1} and t_{n-1}, t_n, t_{n+1} ;

`// Pre-process`

$(\Delta t)_n^{\text{BE}} \Leftarrow b^{(n)} \widehat{k}_n$; `// time-step for BE`

$t_{n+1}^{\text{new}} \Leftarrow \beta_2^{(n)} t_{n+1} + \beta_1^{(n)} t_n + \beta_0^{(n)} t_{n-1}$; `//` $[t_{n+1}^{\text{new}} - (\Delta t)_n^{\text{BE}}, t_{n+1}^{\text{new}}]$ `BE interval`

$y_n^{\text{old}} \Leftarrow a_1^{(n)} y_n + a_0^{(n)} y_{n-1}$;

`// backward Euler`

Solve for y_{n+1}^{new}: $\qquad \dfrac{y_{n+1}^{\text{new}} - y_n^{\text{old}}}{(\Delta t)_n^{\text{BE}}} = f\big(t_{n+1}^{\text{new}}, y_{n+1}^{\text{new}}\big)$

`// Post-process : extrapolation`

$y_{n+1}^{\text{DLN}} \Leftarrow c_2^{(n)} y_{n+1}^{\text{new}} + c_1^{(n)} y_n + c_0^{(n)} y_{n-1}$; `// the DLN solution`

$\widetilde{y}_{n+1} \Leftarrow 2 y_{n+1}^{\text{new}} - y_n^{\text{old}}$; `// first order solution for adaptivity`

$\widehat{T}_{n+1} \Leftarrow \| y_{n+1}^{\text{DLN}} - \widetilde{y}_{n+1} \|$; `// Estimator of LTE`

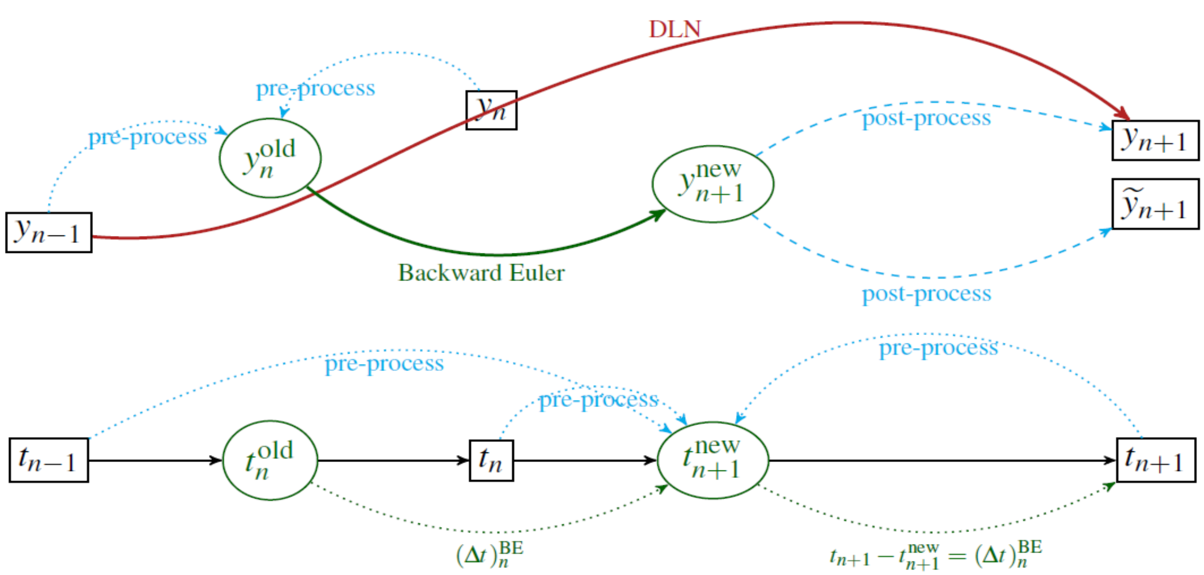


Figure 2: The estimator of LTE by the Refactorization Algorithm

We change (3.12) to obtain the step controller for Algorithm 3

$$k_{n+1} = k_n \cdot \min\Big\{1.5, \max\Big\{0.2, \kappa\big(\mathrm{Tol}/\|\widehat{T}_{n+1}\|\big)^{1/2}\Big\}\Big\}. \tag{3.16}$$

1/3 in (3.12) is replaced by 1/2 in (3.16) since $\widetilde{y}_{n+1}$ is first order accurate. Then we have the DLN adaptivity algorithm (Algorithm 4) with the refactorization process in Algorithm 3 and step controller in (3.16)

Algorithm 4: Adaptivity with refactorization (Algorithm 3) and step size controller in (3.16)

Input: tolerance Tol, initial value y_1, initial stepsize k_1, safety factor κ, time interval $[T_1, T_2]$;

$n \Leftarrow 1$; $t_n \Leftarrow T_1$;

$t_{n+1} \Leftarrow t_n + k_n$; `// update the current time`

compute y_{n+1} by one-step method (e.g. backward Euler);

$n \Leftarrow n+1$, $k_n \Leftarrow k_{n-1}$;

while $t_n + k_n < T_2$ **do**

  $t_{n+1} = t_n + k_n$;

  Input y_n, y_{n-1} and t_{n-1}, t_n, t_{n+1} in Algorithm 3 and obtain y^{DLN}_{n+1} and $\widehat{T}_{n+1}$;

  if $\widehat{T}_{n+1} < \mathrm{Tol}$ **then**

    $k_{n+1} \Leftarrow k_n \cdot \min\big\{1.5, \max\big\{0.2, \kappa\big(\mathrm{Tol}/\|\widehat{T}_{n+1}\|\big)^{1/2}\big\}\big\}$; `// adjust step by` (3.12)

    $y_{n+1} \Leftarrow y^{\mathrm{DLN}}_{n+1}$; `// accept result`

    $n \Leftarrow n+1$; `// come to next time step`

  else

    $k_n \Leftarrow k_n \cdot \min\big\{1.5, \max\big\{0.2, \kappa\big(\mathrm{Tol}/\|\widehat{T}_{n+1}\|\big)^{1/2}\big\}\big\}$; `// adjust step for recomputing`

We have the following theorems about consistency and stability of the midpoint rule solution in Algorithm 3

Theorem 5 (Consistency). *The numerical solution $\widetilde{y}_{n+1}$ in Algorithm 3 is exactly first order approximation to $y(t_{n+1})$ and the local truncation error is*

$$\mathcal{L} = \frac{(1+\varepsilon_n)f(\delta,\varepsilon_n)}{2b^{(n)}\big[(\alpha_2-\alpha_0)+\varepsilon_n(\alpha_2+\alpha_0)\big]}y''(t_{n+1}^{\mathrm{new}})k_n + \cdots .$$

where

$$f(\delta,\varepsilon_n) := \Big\{\big(1-2\beta_2^{(n)}\big) + 2\big(\beta_0^{(n)} - a_0^{(n)}\beta_2^{(n)}\big)\Big(\frac{1-\varepsilon_n}{1+\varepsilon_n}\Big) + a_0^{(n)}\big(2\beta_0^{(n)}-1\big)\Big(\frac{1-\varepsilon_n}{1+\varepsilon_n}\Big)^2\Big\}$$

Proof. By definition, the LTE is

$$\begin{aligned}\mathcal{L} =& \frac{1}{t_{n+1}-t_n^{\mathrm{old}}}\Big\{y(t_{n+1}) - \big(a_1^{(n)}y(t_n) + a_0^{(n)}y(t_{n-1})\big) - f\big(t_{n+1}^{\mathrm{new}}, y(t_{n+1}^{\mathrm{new}})\big)\Big\}\\ &\Big(\approx\frac{\widetilde{y}_{n+1}-y_n^{\mathrm{old}}}{t_{n+1}-t_n^{\mathrm{old}}} - f\big(t_{n+1}^{\mathrm{new}}, y_{n+1}^{\mathrm{new}}\big)\Big)\end{aligned}$$

We use Taylor expansion and expand the above expression at time t_{n+1}^{new}, which results in (5). Also we have: for fixed $\delta \in (0,1)$, $f(\delta,\varepsilon_n) > 0$ for $-1 < \varepsilon_n < 1$ since it's easy to check

$$f(\delta,1)=0, \quad \lim_{\varepsilon_n\to-1} f(\delta,\varepsilon_n) = -\infty, \quad \frac{df(\delta,\varepsilon_n)}{d\varepsilon_n} > 0, \quad \frac{1+\varepsilon_n}{(\alpha_2-\alpha_0)+\varepsilon_n(\alpha_2+\alpha_0)} > 0.$$

$\square$

3.4 Numerical Tests

3.4.1 Constant Step Tests

In this section, we implement the constant time-stepping DLN algorithm with parameter $\delta = 2/3, 2/\sqrt{5}, 1$ ($\delta = 1$ corresponds to the midpoint rule). $\delta = 2/3$ was suggested in [33] to minimize the constant in the LTE of the constant DLN method. $\delta = 2/\sqrt{5}$ was proposed in [76], to ensure the best stability at infinity. When $\delta = 1$, DLN reduces to the midpoint rule and provides energy conservation.

3.4.1.1 Quasi-periodic oscillations

Consider the quasi-periodic oscillations,

$$
\begin{aligned}
&y^{(4)} + (\pi^2+1)y'' + \pi^2 y = 0, \quad 0 \le t \le 20, \\
&y(0) = 2,\ y'(0) = 0,\ y''(0) = -(1+\pi^2),\ y'''(0) = 0.
\end{aligned}
$$

The exact solution is $y(t) = \cos(t) + \cos(\pi t)$. We use this problem to verify the second-order convergence. Let $e_n = \|y(t_n) - y_n\|$ be the error at time t_n and k be constant step size for the DLN method. We have the following two discrete norms

$$
\|e\|_{2,\infty} = \max_{1\le n\le N}\{e_n\}, \qquad \|e\|_{2,2} = \Big(\sum_{n=1}^{N} k e_n^2\Big)^{1/2}.
$$

Table 1 and 2 and Figure 3 summarize the errors and rate of convergence for the constant step DLN method with specific parameter.

Table 1: Second-order convergence of the constant step DLN method using $\|\cdot\|_{2,\infty}$-norm

Step size	$\|e\|_{2,\infty}$ ($\delta = \frac{2}{3}$)	Rate	$\|e\|_{2,\infty}$ ($\delta = \frac{2}{\sqrt{5}}$)	Rate	$\|e\|_{2,\infty}$ ($\delta = 1$)	Rate
0.05	0.32233672	-	0.19537687	-	0.12271718	-
0.025	0.08202388	1.9745	0.04926517	1.9876	0.03084194	1.9924
0.0125	0.02056438	1.9959	0.01234158	1.9970	0.00771706	1.9988
0.00625	0.00514472	1.9990	0.00308709	1.9992	0.00192962	1.9997
0.003125	0.00128642	1.9997	0.00077188	1.9998	0.00048244	1.9999

From the two tables and the log-log plot, we confirm the second-order convergence of the constant step DLN method with the three specific values of parameter δ, and the error is reduced as δ is increasing. In section 3.4.2.1, we also test adaptivity algorithms of variable step DLN in Section 3.2 and Section 3.3 for this problem.

Table 2: Second-order convergence of the constant step DLN method using $\|\cdot\|_{2,2}$-norm

Step size	$\|e\|_{2,2}$ ($\delta = \frac{2}{3}$)	Rate	$\|e\|_{2,2}$ ($\delta = \frac{2}{\sqrt{5}}$)	Rate	$\|e\|_{2,2}$ ($\delta = 1$)	Rate
0.05	0.61799316	-	0.37320014	-	0.23460108	-
0.025	0.15634451	1.9829	0.09391299	1.9906	0.05876962	1.9971
0.0125	0.03917128	1.9969	0.02350951	1.9981	0.01469880	1.9994
0.00625	0.00979800	1.9992	0.00587936	1.9995	0.00367508	1.9998
0.003125	0.00244989	1.9998	0.00146999	1.9999	0.00091879	2.0000

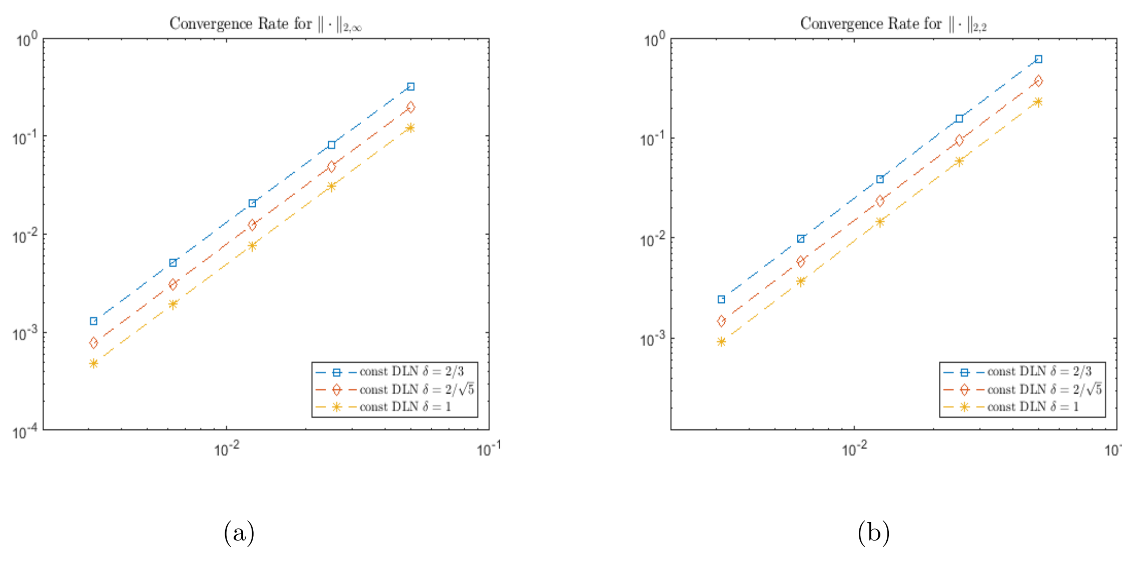


Figure 3: log-log plot of convergence rate for the constant step DLN algorithm

3.4.1.2 Increase of Oscillation

This test aims to compare the constant step DLN method with backward Euler(BE), BE plus filter (with parameter $\nu = 2/3$) [58] and BDF2 algorithms, showing that these commonly used methods are so stable that they fail in some specific problems like (3.17). The test problem is

$$x' = \mu x + \frac{1}{\mu} y, \quad y' = -\frac{1}{\mu} x + \mu y, \quad x(0) = 1, \quad y(0) = 0, \quad \mu = 1.\mathrm{e} - 2. \tag{3.17}$$

We simulate the test on the interval $[0, 20]$. We test the backward Euler with constant step $k = 1.\mathrm{e} - 4$ and other methods with $k = 1.\mathrm{e} - 3$. Figures 4 show that the oscillations of constant step backward Euler, BE plus filter and BDF2 decrease as the time grows, which shows that these methods are too stable to approximate the true solutions well. BE performance (Figure 4(a) and 4(b)) is worst among these methods even under smaller time steps ($k = 1.\mathrm{e} - 4$). From Figures 5, we can see that the oscillations of the DLN solutions increase and these patterns are the same as those of the exact solution. In particular, the constant DLN method simulates the exact solution better as δ increases.

3.4.1.3 The Lorenz system

Given the Lorenz system [94]:

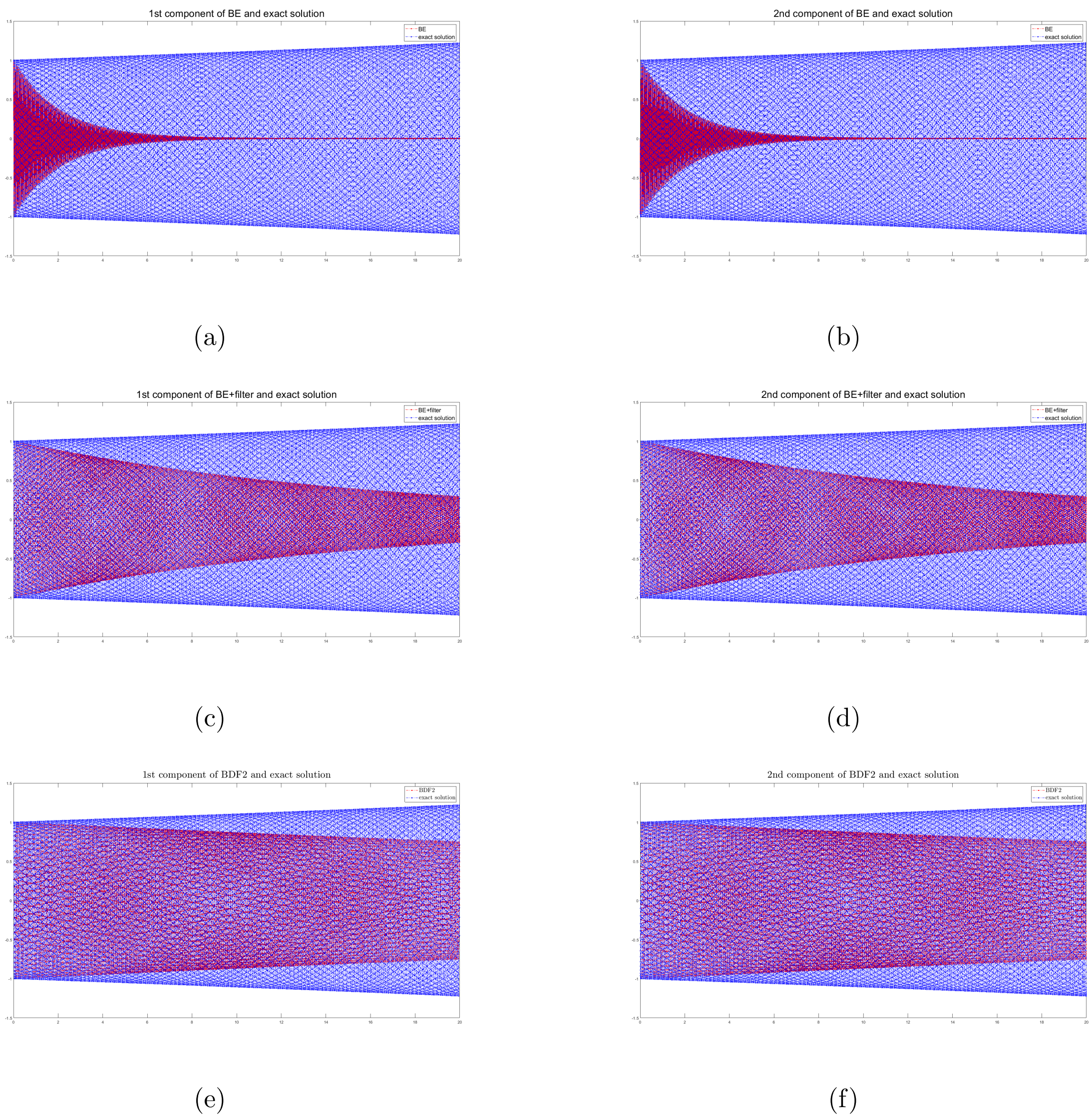
$$x' = \sigma \left(y - x \right), \quad y' = -xz + \lambda x - y, \quad z' = xy - \eta z. \tag{3.18}$$

We choose two arrays of parameters and initial values from [42, 94]:

$$\begin{pmatrix} \sigma & \lambda & \eta \\ x_0 & y_0 & z_0 \end{pmatrix} = \begin{pmatrix} 12 & 12 & 6 \\ -10 & -10 & 25 \end{pmatrix}, \tag{LorenzData1}$$

$$\begin{pmatrix} \sigma & \lambda & \eta \\ x_0 & y_0 & z_0 \end{pmatrix} = \begin{pmatrix} 10 & 28 & 8/3 \\ 0 & 1 & 0 \end{pmatrix}. \tag{LorenzData2}$$

The system of equations (3.18) is simulated over time interval $[0, 5]$. We choose constant step size k to be 0.02 for both arrays (LorenzData1) and (LorenzData2). We don't know the

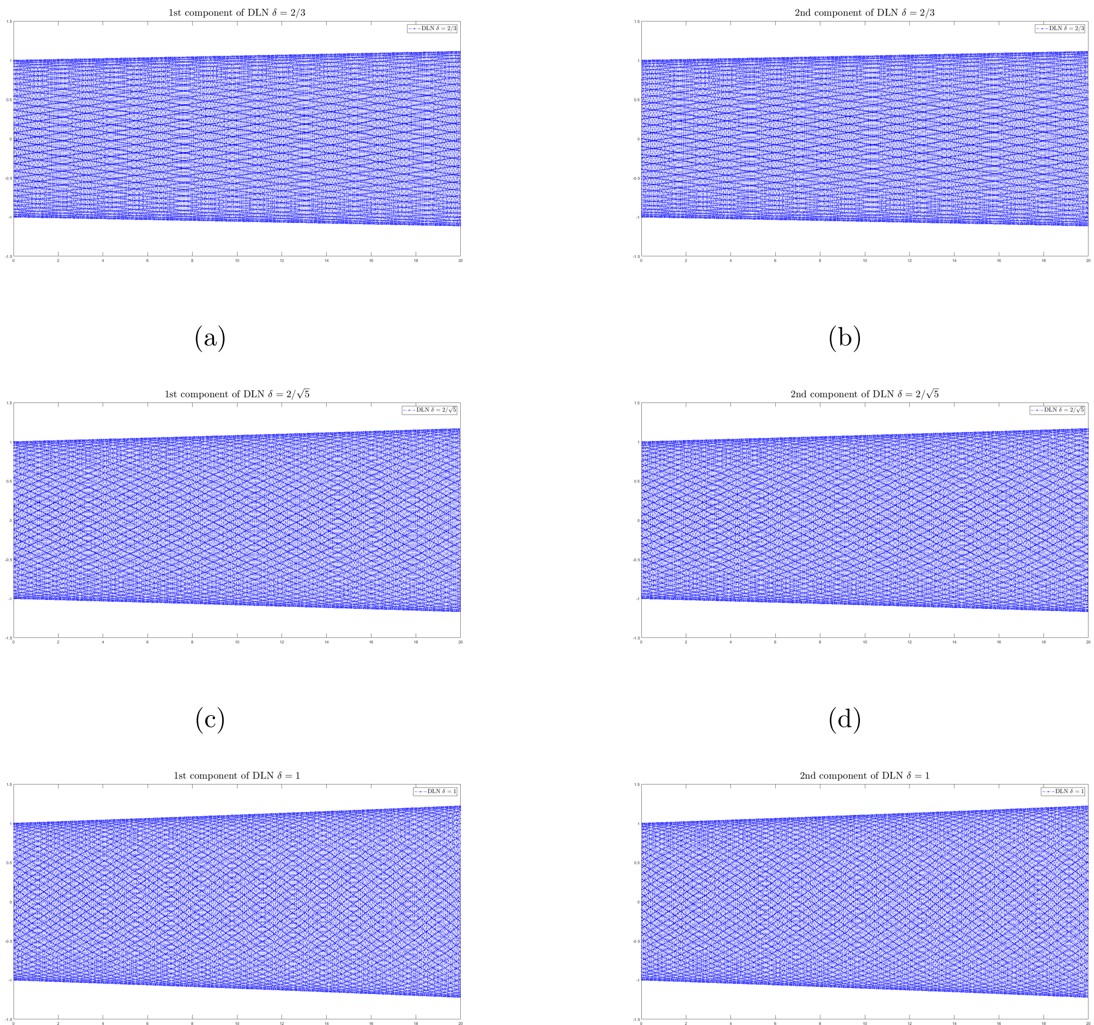


(a) (b)

(c) (d)

(e) (f)

Figure 4: Oscillations of BE, BE Plus Filter, and BDF2 solutions decrease as time grows and in contrast the oscillations of the exact solution increase.



(a) (b)

(c) (d)

(e) (f)

Figure 5: Oscillations of Constant Step DLN ($\delta = 2/3, 2/\sqrt{5}, 1$) solutions increase, which shows that the simulations approximate exact solutions well.

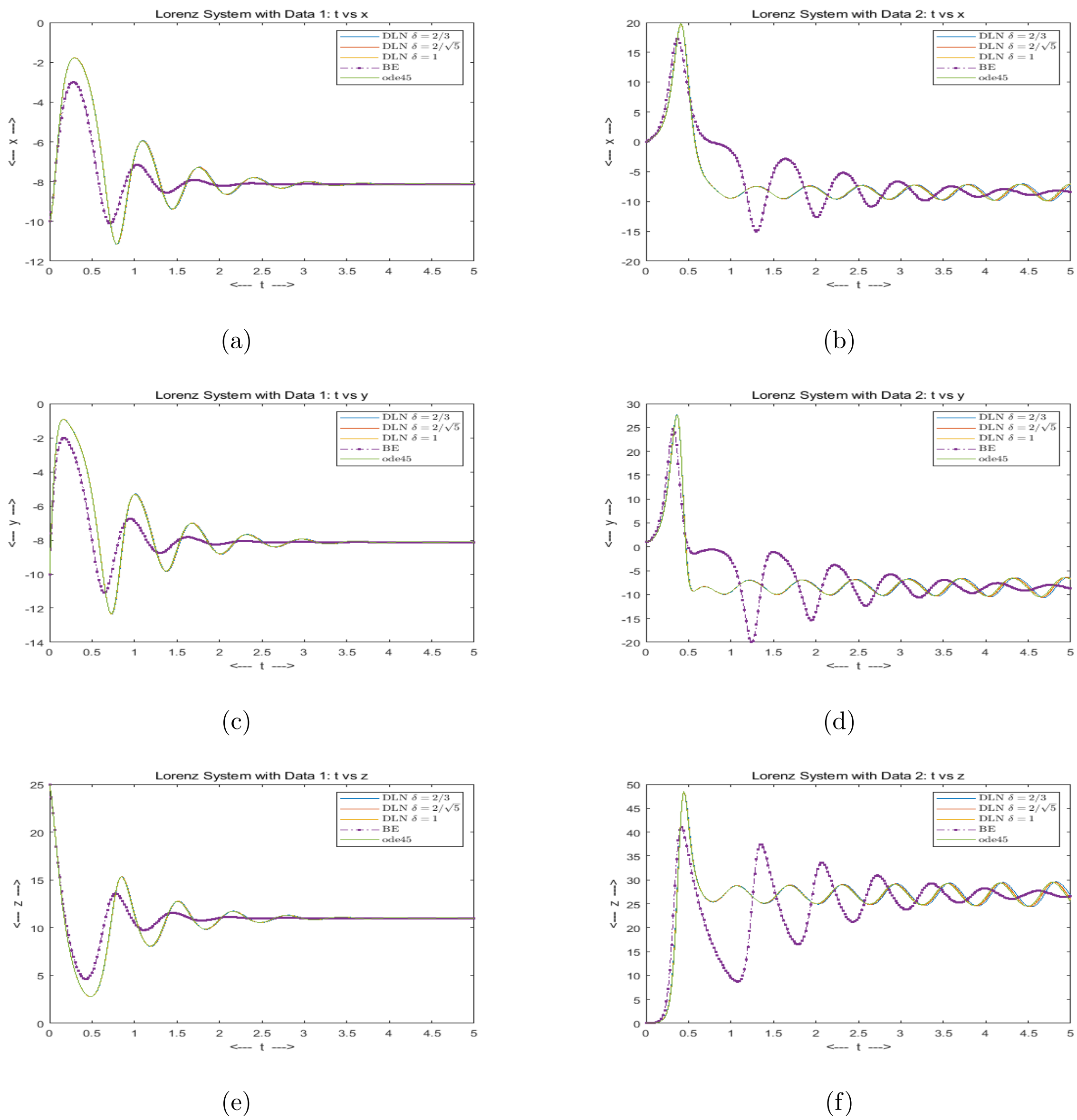


Figure 6: The constant step DLN solutions oscillate correctly to steady state while the constant step BE solutions over damp to equilibrium.

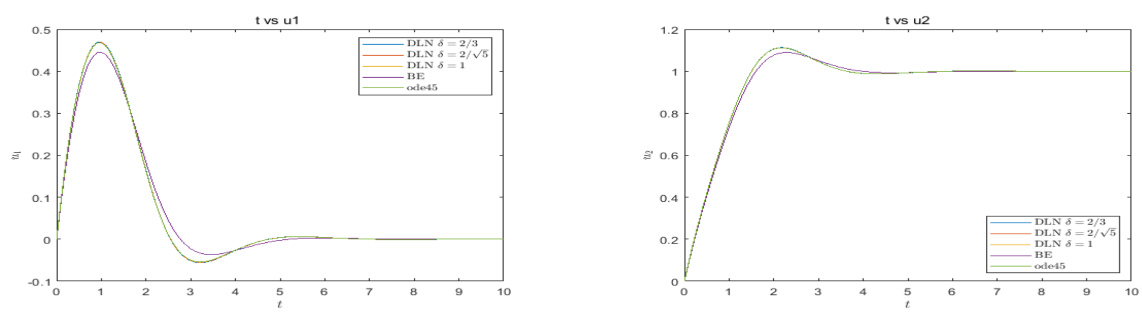
exact solution to (3.18) thus we use the MATLAB ode45 function for reference. Results are given in Figures 6. From the graphs, we can see that the results of the constant step DLN methods and the adaptive MATLAB ode45 function are almost the same. The constant step DLN solutions oscillate correctly to steady state while the constant step BE solutions over damp to equilibrium.

3.4.1.4 The example of Sussman

In this part, we test the non-linear autonomous system proposed by Sussman [120]:

$$
\begin{aligned}
&u_1'(t) + u_2 u_2 + u_1 = 1, \\
&u_2'(t) - u_2 u_1 + u_2 = 1, \qquad u_1(0) = 0, \quad u_2(0) = 0.
\end{aligned}
$$

Sussman found that non-linearly implicit methods converges to equilibrium as $t \to \infty$ (correctly) but semi-implicit (linearly implicit) methods do not. We simulate this system over time interval $[0, 10]$ and set the step size k to be 0.1. From Figure 7, we can see that the approximate solutions by constant step DLN method approach to steady state correctly.



(a) (b)

Figure 7: Constant step DLN solutions approach to steady state correctly.

3.4.2 Variable Step Tests

In this subsection, we implement the variable time-stepping DLN method to several test problems. For all test problems in this subsection, we use Algorithm 2 (with estimator of LTE by AB2) and Algorithm 4 **without** setting the minimum step size. We still set parameter $\delta = 2/3, 2/\sqrt{5}, 1$.

3.4.2.1 Quasi-periodic oscillations

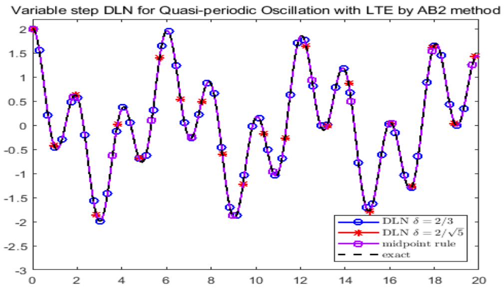
We still consider the Quasi-periodic oscillations system in (3.4.1.1). To test Algorithm 2 and Algorithm 4, we set the initial time step to be 0.01 and the tolerance to be 1.e−4 **without** the restriction of step size. Figure 8(a) and 8(b) show that the DLN method with Algorithm 2 works well and the number of steps are 2948, 2118 and 1678 for $\delta = 2/3, 2/\sqrt{5}$ and 1 respectively. Figure 8(c) and 8(d) display the results of the adaptivity of DLN by Algorithm 4. The only difference is that the results by Algorithm 4 take more steps (Number of steps are 24880 and 25649 for $\delta = 2/3$ and $2/\sqrt{5}$ respectively). We also compare the errors of the adaptive DLN algorithm and the constant DLN method (with the same number of steps for each δ) in Table 3 and Table 4. For both $\| \cdot \|_{2,2}$ and $\| \cdot \|_{2,\infty}$ of the error, the constant step DLN method works better than the adaptive DLN algorithm under the same number of step in that the exact solution of quasi-periodic oscillations problem is smooth. However, in some extreme stiff test problems (Van der Pol's equation 3.4.2.4), the adaptive DLN algorithm has the advantage over the constant step DLN algorithm.

3.4.2.2 Lotka-Volterra Equations

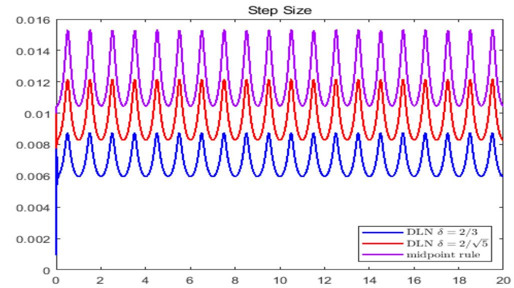
We consider the following Lotka-Volterra equations [123]

$$\begin{aligned} x' = 2x - xy, \quad y' = -y + xy, \\ x(0) = 4, \quad y(0) = 2. \end{aligned} \tag{3.19}$$

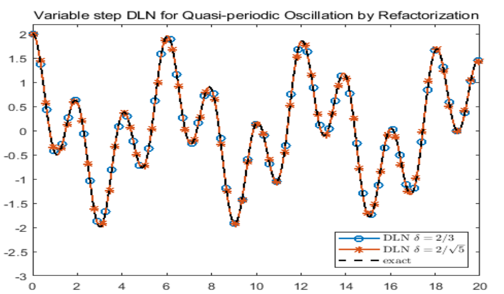
We still apply the variable time-stepping DLN method to Lotka-Volterra system (3.19) with Algorithm 2 and Algorithm 4. We set the total time interval to be $[0, 500]$, the initial time



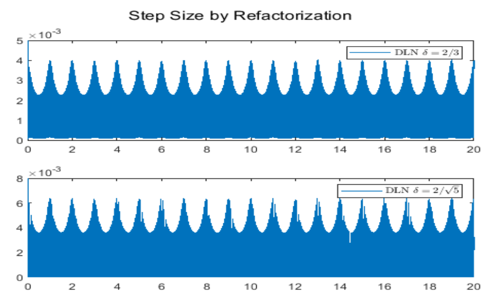
(a)



(b)



(c)



(d)

Figure 8: Periodic oscillations

Table 3: Comparison of Algorithm 2 and constant DLN algorithm with the same number of time steps

	$\delta = \frac{2}{3}$	$\delta = \frac{2}{\sqrt{5}}$	$\delta = 1$
$\|e\|_{2,\infty}$	0.00638129	0.00740505	0.00737554
$\|e\|_{2,2}$	0.01215392	0.01410517	0.01404667
	Const ($\delta = \frac{2}{3}$)	Const ($\delta = \frac{2}{\sqrt{5}}$)	Const ($\delta = 1$)
$\|e\|_{2,\infty}$	0.00606172	0.00704536	0.00701630
$\|e\|_{2,2}$	0.01154431	0.01341880	0.01336421

Table 4: Comparison of Algorithm 4 and constant DLN algorithm with the same number of time steps

	$\delta = \frac{2}{3}$	$\delta = \frac{2}{\sqrt{5}}$	Const ($\delta = \frac{2}{3}$)	Const ($\delta = \frac{2}{\sqrt{5}}$)
$\|e\|_{2,\infty}$	0.00038190	0.00061367	0.00008513	0.00004806
$\|e\|_{2,2}$	0.00072598	0.00116607	0.00016212	0.00009153

step k_0 to be $1.e-4$ **without** restrictions on time steps. We set the tolerance to be $1.e-6$ and $1.e-8$ for Algorithm 2 and the tolerance to be $1.e-6$ for Algorithm 4. We use the constant step DLN algorithm, MATLAB ode15s, ode23 and ode45 function (with relative tolerance $1.e-6$ and absolute tolerance $1.e-6$) for reference. The phase solutions are given in Figure 9. All the methods we apply for work well in terms of the phase solution. Then we check the results of Hamiltonian conservation with the Hamiltonian function

$$H(x, y) = x - \ln x + y - 2 \ln y.$$

The results are displayed in Figure 10 and the number of steps is summarized in Table 5. Adaptive DLN algorithm 2 has higher accuracy with many more steps as the tolerance is increased. Adaptive DLN algorithm 4 has higher accuracy with a larger number of steps as the parameter δ decreases. Ode 45 functions works better than ode15s and ode 23 functions but these functions take less number of steps than adaptive DLN algorithm 2 and 4. The constant step DLN method has relatively good performance but with larger oscillations and it has better simulation as the parameter δ increases.

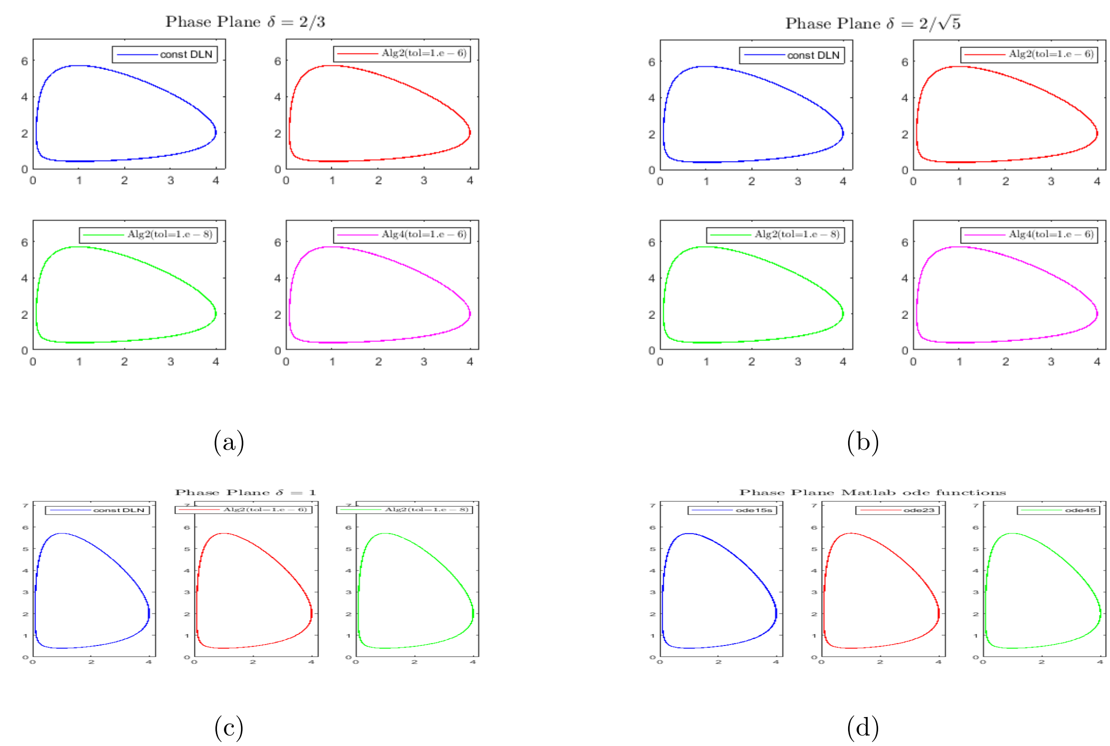
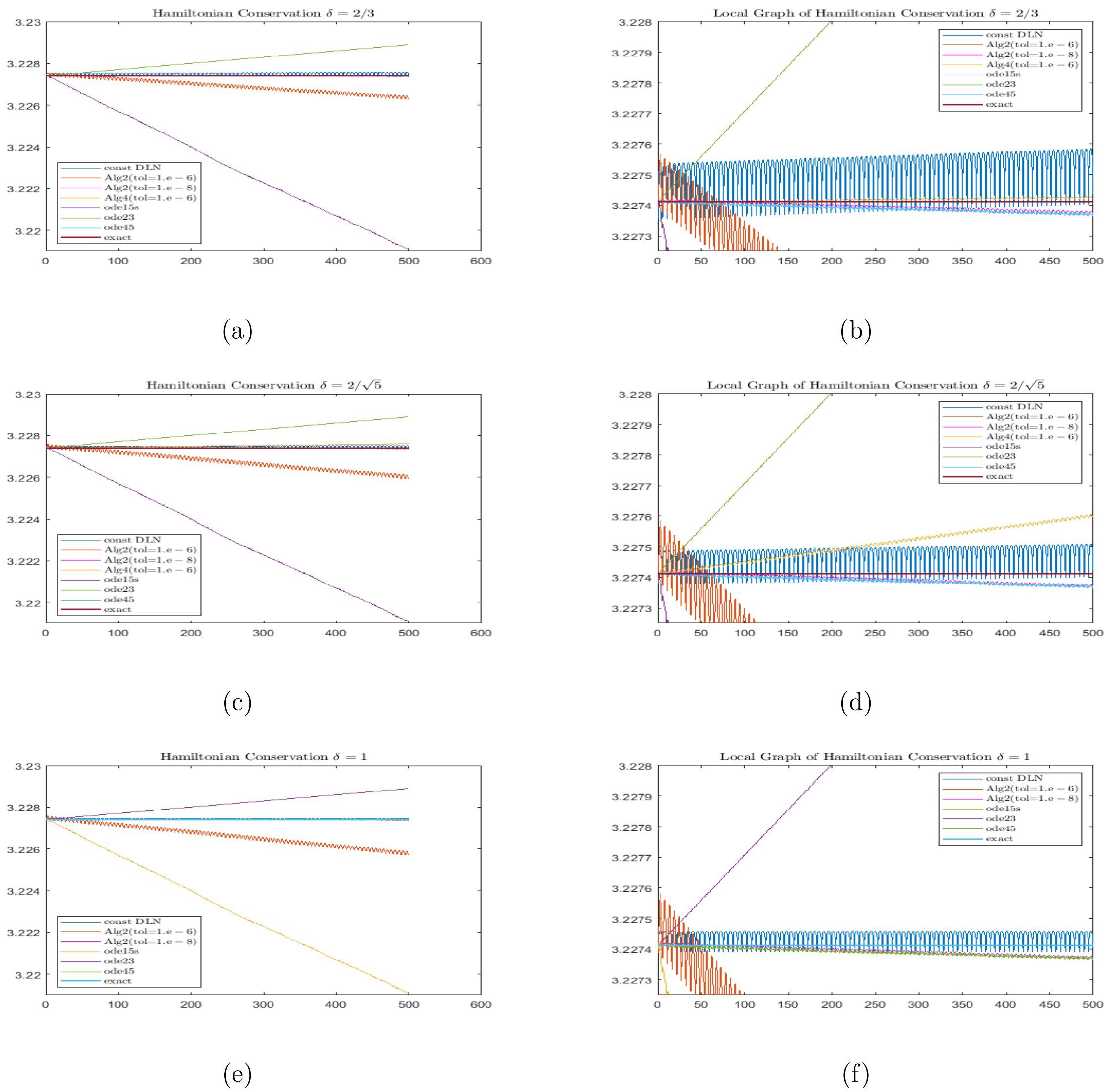


Figure 9: Lotka-Volterra System Phase Solutions by DLN Method and Matlab ODE Functions



(a) (b)

(c) (d)

(e) (f)

Figure 10: Lotka-Volterra System

Table 5: Number of steps of DLN algorithms and MATLAB ode functions

	Constant step DLN	Algorithm 2	Algorithm 2 (Tol $= 1.\mathrm{e}-8$)	Algorithm 4
$\delta = 2/3$	100000	79364	368352	900497
$\delta = 2/\sqrt{5}$	100000	58122	269765	924047
$\delta = 1$	100000	46619	216400	-
	ode15s	ode23	ode45	
	11879	29599	14149	

3.4.2.3 Kepler System

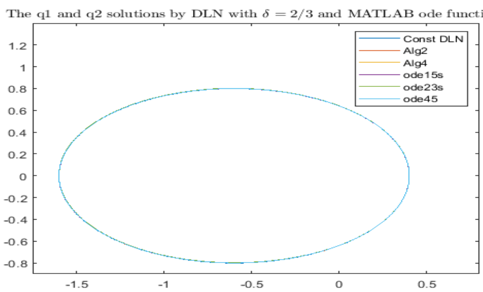
Consider the Kepler system

$$\begin{cases} q_1' = p_1 \\ q_2' = p_2 \\ p_1' = -\frac{q_1}{\sqrt{(q_1^2+q_2^2)^3}} \\ p_2' = -\frac{q_2}{\sqrt{(q_1^2+q_2^2)^3}} \end{cases}, \quad \begin{bmatrix} q_1(0) \\ q_2(0) \\ p_1(0) \\ p_2(0) \end{bmatrix} = \begin{bmatrix} 1-0.6 \\ 0 \\ 0 \\ \sqrt{\frac{1+0.6}{1-0.6}} \end{bmatrix},$$

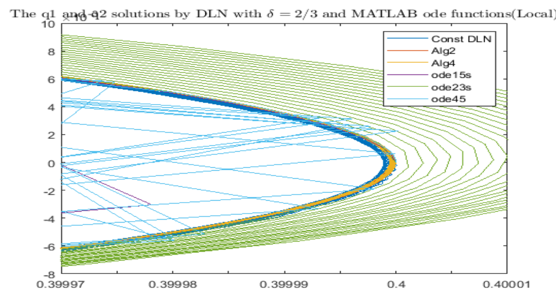
We still apply Algorithm 2, Algorithm 4, constant step DLN method and MATLAB ode functions to the system over time interval $[0, 120]$. We set the tolerance to be $1.\mathrm{e}-8$ for Algorithm 2, $1.\mathrm{e}-6$ for Algorithm 4. For MATLAB ode functions, we set relative tolerance to be $1.\mathrm{e}-6$ and absolute tolerance to be $1.\mathrm{e}-8$. The initial step size is $1.\mathrm{e}-4$ and the number of steps for the constant step DLN algorithm is 100000. Phase solutions of q_1, q_2 and p_1, p_2 by DLN with $\delta = 2/3$ and MATLAB ode functions are given in Figure 11. From Figure 11(b) and Figure 11(d), the adaptive DLN algorithm (Algorithm 2 and Algorithm 4) works better than constant step DLN method and MATLAB ode functions. For $\delta = 2/\sqrt{5}$ and $\delta = 1$, results are similar. Then we compare the Hamiltonian conservation function

$$H(q_1, q_2, p_1, p_2) = \frac{1}{2}\big(p_1^2 + p_2^2\big) - \big(q_1^2 + q_2^2\big)^{-1/2}.$$

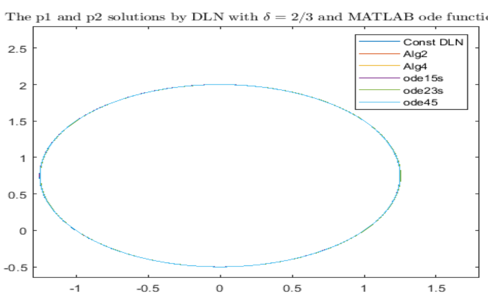
The Hamiltonian conservation results are given in Figure 12. Combining Figure 12 and Table 6, we have: Algorithm 4 with $\delta = 2/3$ works much better than that of two other parameters and constant step DLN algorithm. Solutions by Algorithm 2 are more accurate than solutions of constant step DLN algorithm and take much less number of steps. Under the tolerance above, MATLAB ode functions have worse performance than adaptive DLN algorithm but with a few steps.



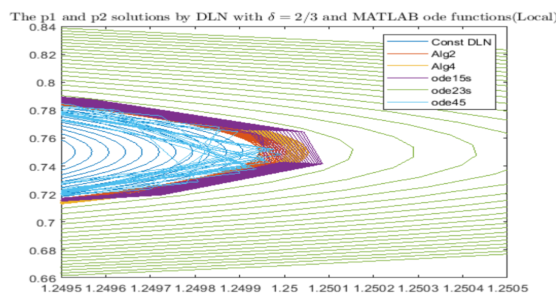
(a)



(b)



(c)



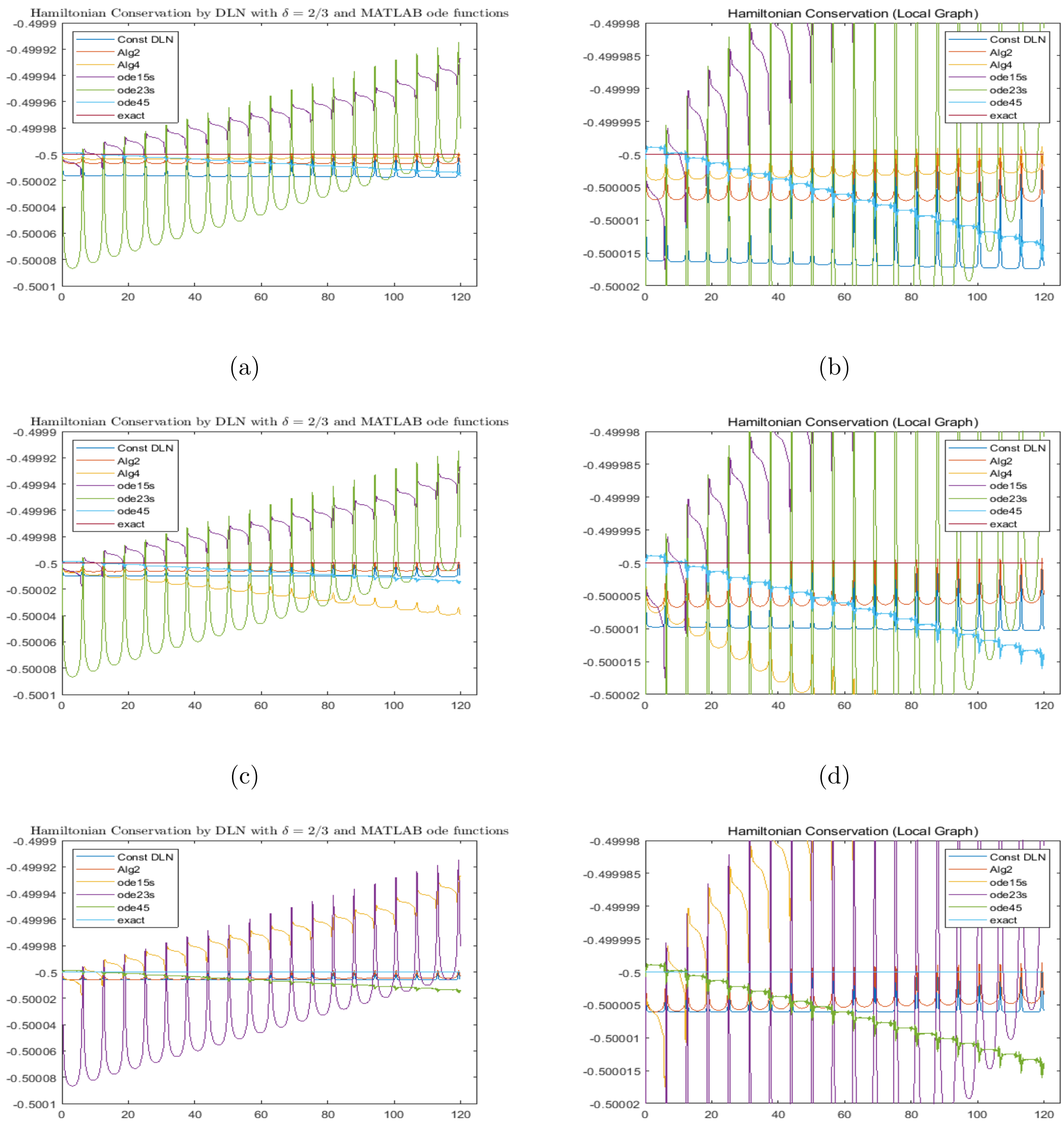
(d)

Figure 11: Phase solutions of Kepler system by DLN with $\delta = 2/3$ and MATLAB ode functions

3.4.2.4 Van der Pol's equation

Finally, we test Van der Pol's equation

$$x'' - \mu(1 - x^2)x' + x = 0, \quad x(0) = 2, \quad x'(0) = 0.$$



(a) (b)

(c) (d)

(e) (f)

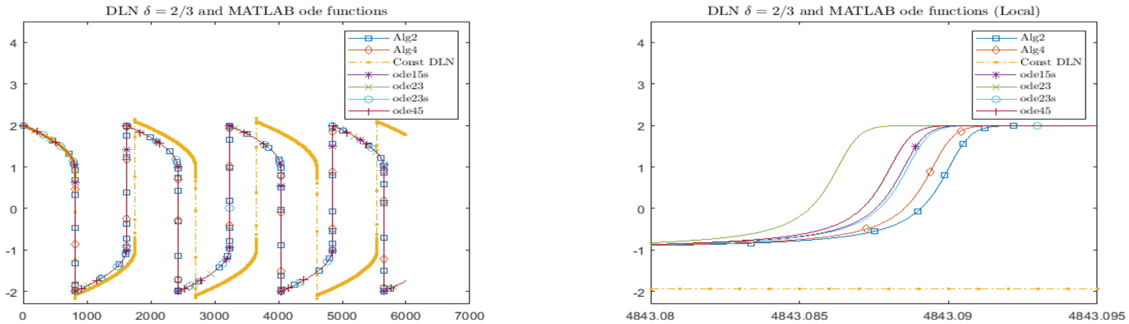
Figure 12: Hamiltonian Conservation of Kepler System

Table 6: Number of steps of DLN algorithms and MATLAB ode functions

	Constant step DLN	Algorithm 2	Algorithm 4
$\delta = 2/3$	100000	62337	154817
$\delta = 2/\sqrt{5}$	100000	47202	157626
$\delta = 1$	100000	38775	-
	ode15s	ode23	ode45
	3374	12235	3733

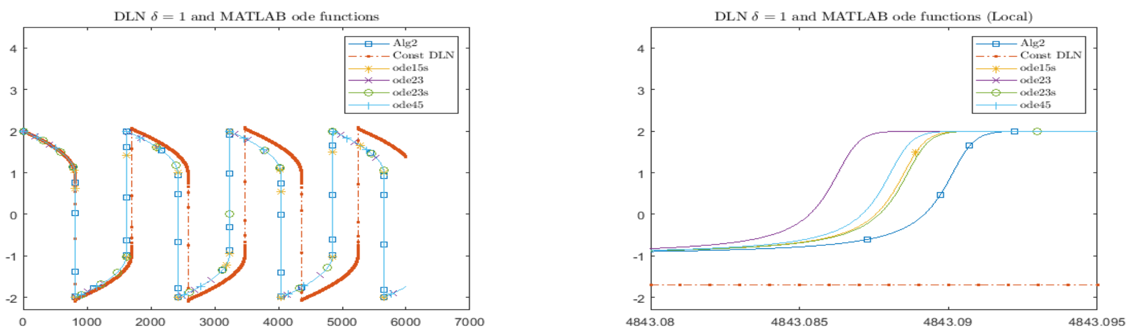
Van der Pol's equation with parameter $\mu = 1000$ is a common test problem for stiff solvers. We test the problem with constant step DLN method and adaptive DLN algorithm (Algorithm 2 and Algorithm 4). We use MATLAB ode functions (ode15s, ode23, ode23s and ode 45) for reference. We simulate this problem over time interval $[0, 6000]$. We first choose the parameter δ to be $2/3$ and 1. For Algorithm 2, we set the tolerance to be $1.\mathrm{e}-6$ and inital time step to be $1.\mathrm{e}-4$. Due to the stiffness of the test, we adjust the tolerance to be $1.3\mathrm{e}-6$ and the safety factor κ to be 0.65 to avoid pre-setting the minimum step size for Algorithm 4. But initial time step is still $1.\mathrm{e}-4$. For ode15s and ode23s, we set relative tolerance to be $1.\mathrm{e}-10$ and absolute tolerance to be $1.\mathrm{e}-15$. For ode23 and ode45, we set both relative tolerance and absolute tolerance to be $1.\mathrm{e}-6$.

Figure 13 and Figure 14 give the global and local approximate solutions of the first component and Table 7 summarizes the number of steps. From Figure 13(a) and 14(a), we see that constant step DLN solutions lag behind other solutions even under large number of steps. From Figure 13(b) and 14(b), adaptive DLN algorithms (Algorithm 2 and Algorithm 4) work well for this stiff problem if we use solutions of MATLAB ode functions as reference. From Table 7, adaptive DLN algorithms are more efficient than most MATLAB ode functions.



(a) (b)

Figure 13: Van der Pol's equation by DLN with $\delta = 2/3$ and MATLAB ode functions



(a) (b)

Figure 14: Van der Pol's equation by DLN with $\delta = 1$ and MATLAB ode functions

Table 7: Number of steps of DLN algorithms and MATLAB ode functions

	Constant step DLN	Algorithm 2	Algorithm 4	
$\delta = 2/3$	6000000	62806	769319	
$\delta = 1$	6000000	32379	-	
	ode15s	ode23	ode23s	ode45
	19714	4377590	1593283	13242893

3.4.3 Lindberg's Example with Step Floor

In this subsection, we consider a extremely stiff problem, Lindberg's example [126]

$$\begin{cases} y_1' = 10^4 y_1 y_3 + 10^4 y_2 y_4 \\ y_2' = -10^4 y_1 y_4 + 10^4 y_2 y_3 \\ y_3' = 1 - y_3 \\ y_4' = -0.5 y_3 - y_4 + 0.5 \end{cases}, \qquad \begin{bmatrix} y_1(0) \\ y_2(0) \\ y_3(0) \\ y_4(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ 0 \end{bmatrix}.$$

The exact solution for the above system is

$$\begin{cases} y_1 = e^{g_1(t)} \big[\cos(g_2(t)) + \sin(g_2(t)) \big] \\ y_2 = e^{g_1(t)} \big[\cos(g_2(t)) - \sin(g_2(t)) \big] \\ y_3 = 1 - 2e^{-t} \\ y_4 = te^{-t} \end{cases}, \qquad \begin{cases} g_1(t) = 10^4 (t + 2e^{-t} - 2) \\ g_2(t) = 10^4 (1 - e^{-t} - te^{-t}) \end{cases}.$$

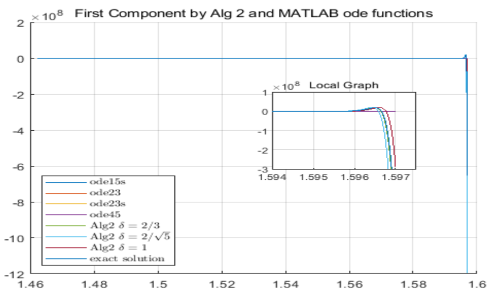
We test this system with Algorithm 2 (with estimator of LTE by AB2) and Algorithm 4. The third and fourth components consist of a linear system and all time-stepping schemes work well for the system. Therefore we test the system of y_1 and y_2, i.e.

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = A(t) \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad A(x) = 10^4 \begin{bmatrix} 1 - 2e^{-t} & te^{-t} \\ -te^{-t} & 1 - 2e^{-t} \end{bmatrix}, \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \qquad (3.20)$$

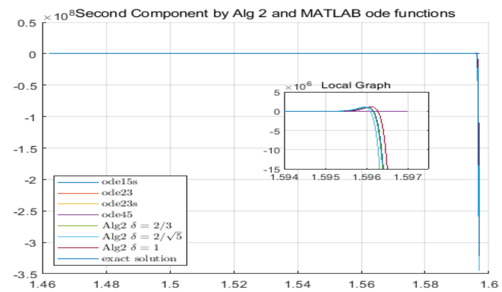
The above system in (3.20) is extremely stiff because both eigenvalues of the matrix $A(x)$ are -10^4 initially and approach $5946 \pm 3235i$ as $t = 1.596$. Both Algorithm 2 and Algorithm 4 simulate inaccurately starting from the initial time $t = 0$ without minimum step size (The simulations decay from 1 to zero for the two components). All MATLAB functions including ode15s, ode23, ode23s and ode45 (with relative tolerance $1.\text{e} - 11$ and absolute tolerance $1.\text{e} - 15$) also fail for this example.

Thus we simulate the problem on time interval $[1.4622, 1.597]$ for Algorithm 2 and $[1.4633, 1.597]$ for Algorithm 4 with minimum step size $1.\text{e} - 8$. We set the initial step size to be the same as the minimum step size. The tolerance of the two adaptive DLN algorithms is given in Table 8. The graphs of first and second components by Algorithm 2 are given in Figure 15. From Figure 15(c) and 15(d), the Algorithm 2 with $\delta = 2/3, 2/\sqrt{5}, 1$ simulate relative well while all MATLAB ode functions fail even under small tolerance. From

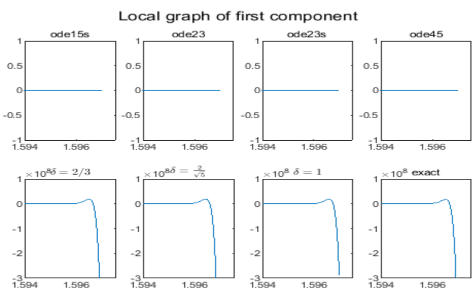
Figure 16, we can see that the simulations by Algorithm 3 work much better but with more steps and later initial time.



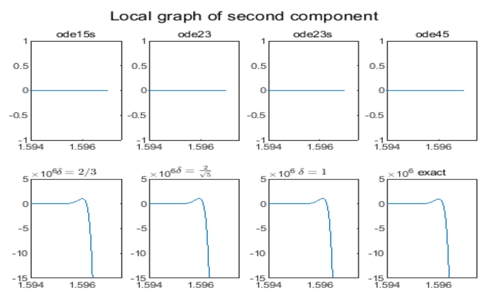
(a)



(b)



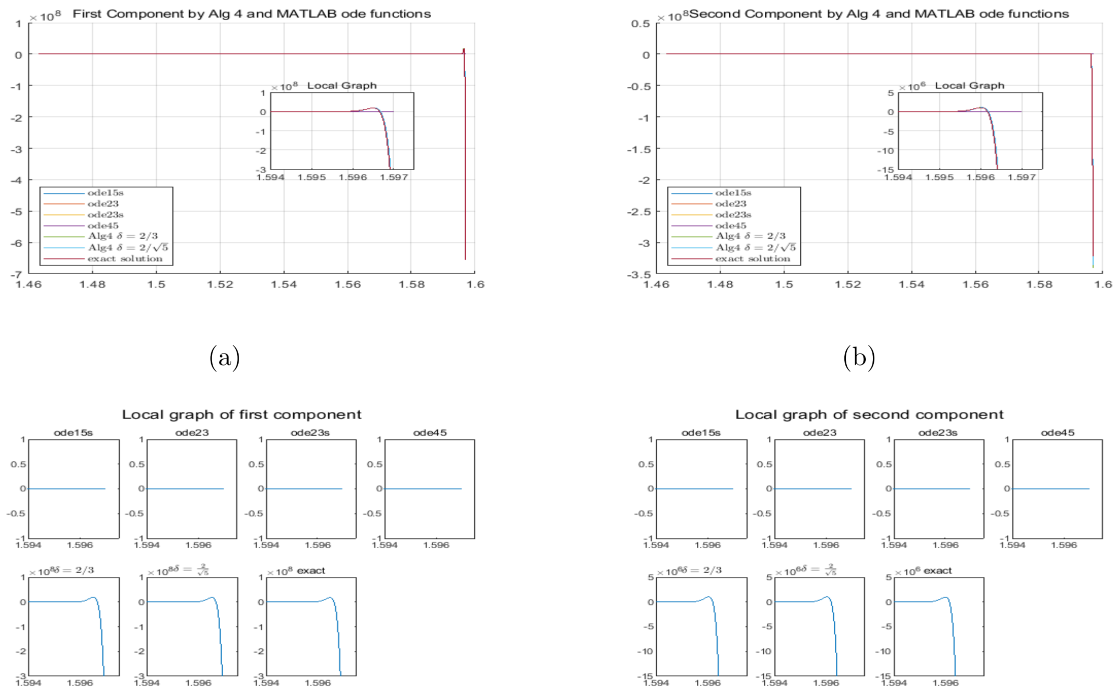
(c)



(d)

Figure 15: First and second components of Lindberg's example by Algorithm 2 and MATLAB ode functions

Then we try adaptive DLN algorithm on time interval $[0, 1.597]$ and find that Algorithm 2 works to some extent (with different order) with tolerance around $1.e-15$. We also use MATLAB ode15s, ode23, and ode23s for comparison. We neglect the ode45 function because it diverges as time goes to the endpoint. From Figure 17, the values by adaptive DLN algorithm increase to 1.e31, much larger than the value of exact solution (1.e8). However all MATLAB ode functions keep zero even under extremely small tolerance (absolute tolerance is $1.e-15$ and relative tolerance is $1.e-11$). The tolerance for the adaptive DLN algorithm is given in Table 9

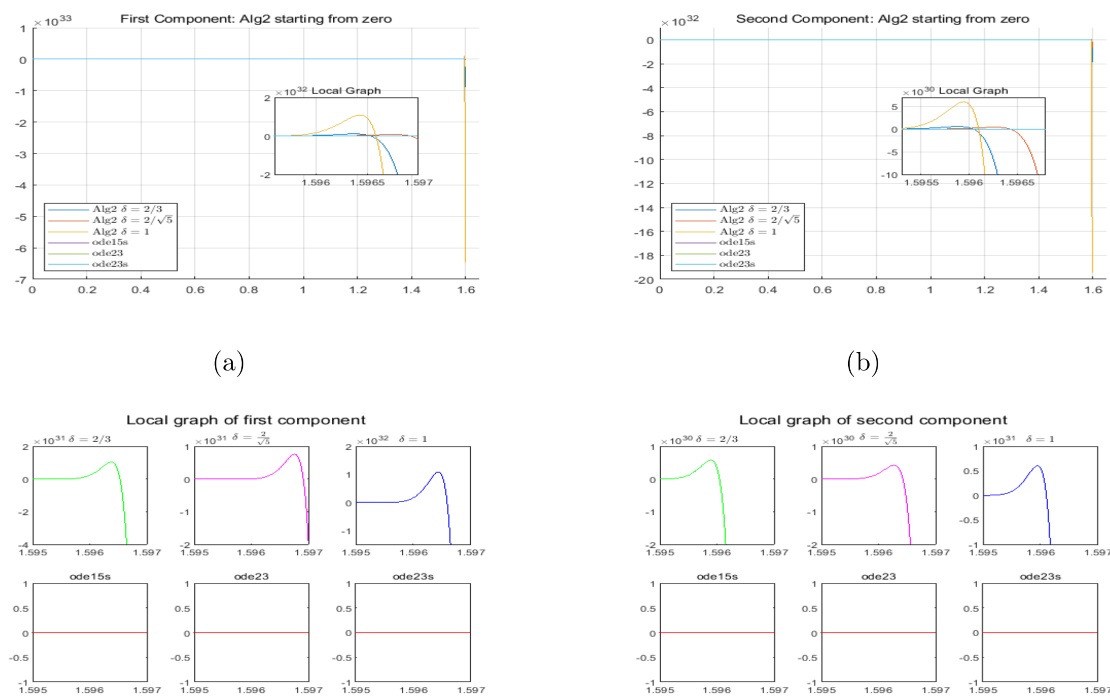


(a) (b) (c) (d)

Figure 16: First and second components of Lindberg's example by Algorithm 4 and MATLAB ode functions

Table 8: Tolerance of DLN algorithms for Lindberg's example

	$\delta = 2/3$	$\delta = 2/\sqrt{5}$	$\delta = 1$
Algorithm 2	$0.635 * (1.e - 14)$	$5.268 * (1.e - 15)$	$4.627 * (1.e - 15)$
Algorithm 4	$1.e - 14$	$1.e - 14$	-



(a) (b)

(c) (d)

Figure 17: First and second components of Lindberg's example by Algorithm 2 starting from $t = 0$.

Table 9: Tolerance of Algorithm 2 for Lindberg's example starting at $t = 0$

	$\delta = 2/3$	$\delta = 2/\sqrt{5}$	$\delta = 1$
Algorithm 2	$0.79 * (1.e - 15)$	$0.719 * (1.e - 15)$	$1.01 * (1.e - 14)$

3.5 Conclusions

The DLN method is an unconditionally G-stable, second-order accurate time-stepping scheme for both constant and variable time steps. In this chapter, we designed two adaptive algorithms: first via a certain second order explicit scheme like AB2 method, second through the DLN refactorization process in [84]. The numerous numerical tests of both constant step and variable step DLN methods, by the adaptivity algorithms proposed in section 3.2 and 3.3, verify the properties of the DLN method, even in extreme stiff cases. In our numerical studies, the constant timestep DLN methods work best for problems with smooth solutions. The symplectic midpoint method has the smallest errors, also conserving all quadratic Hamiltonians. For stiff problems, the adaptive DLN methods outperform the constant time step algorithms, with minimal computational cost. The preferred step size estimator is the one based on the Milnes device, using the AB2 approximate solution in (3.7). In the near future, we would like to apply the adaptive algorithms in Section 3.2 and 3.3 to incompressible fluid models. The difficulty of Algorithm 2 (adaptivity by explicit methods) is that the local truncation error for the fluid model would be different from that of the evolutionary equation and thus the estimator in (3.7) and (3.10) can not be applied directly. To apply Algorithm 4 (refactorization adaptivity algorithm), we need to prove the first order accurate solution $\widetilde{u}_n$ of velocity in fluid model.

4.0 The DLN Algorithm for the Unsteady Stokes/Darcy Model

4.1 Introduction

The Stokes/Darcy model, simulating the coupling between surface and subsurface motions of fluid, deserves great interest in geophysics and related areas. Mathematical theory and numerical schemes for both steady and unsteady Stokes/Darcy model [5, 38, 44, 49, 67, 85, 133] have been well developed in recent years. Nevertheless, the time discretization for the unsteady Stokes/Darcy model is always a big problem, where various constant time-stepping algorithms give accuracy and efficiency of computation to a different level. Some simulations use first order, fully implicit scheme for simplicity, e.g. [21, 22, 97, 106, 112, 113], while many others implement higher order time-stepping algorithms to increase accuracy, e.g. [24, 25, 87, 88, 105]. Moreover, time adaptivity for variable time-stepping schemes is an ideal way of solving the conflict between time accuracy and computational complexity. Due to the limitations of most existing methods (e.g. BDF2 is not A-stable under increasing step size), variable time-stepping analysis for the unsteady Stokes/Darcy model is promising but technically challenging.

To solve this issue, we refer to the one-parameter family of two-step, one-leg method proposed by Dahlquist, Liniger, and Nevanlinna (the DLN method) [33] and apply the method to the time-dependent Stokes/Darcy model for variable time-stepping analysis. The DLN algorithm maintains G-stability [29, 30, 32, 62] under any arbitrary sequence of time steps and keeps second-order accuracy at the same time. To begin, consider the initial value problem

$$x'(t) = f\left(t, x(t)\right), \quad x(0) = x_0, \tag{4.1}$$

where $x : [0, T] \to \mathbb{R}^d$ and $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ are vector-valued functions. Let $\{t_n\}_{n=0}^N$ be the grid on time interval $[0, T]$ and $k_n := t_{n+1} - t_n$ be local step size. Now given the two initial values x_0 and x_1, the one parameter DLN algorithm (with parameter $\theta \in [0,1]$) for the problem (4.1) is

$$\sum_{j=0}^{2} \alpha_j x_{n-1+j} = (\alpha_2 k_n - \alpha_0 k_{n-1}) f\Big(\sum_{j=0}^{2} \beta_{j,n} t_{n-1+j}, \sum_{j=0}^{2} \beta_{j,n} x_{n-1+j} \Big), \tag{4.2}$$

where coefficients $\{\alpha_j\}_{j=0:2}$ and coefficients $\{\beta_{j,n}\}_{j=0:2}$ (ϵ_n-dependent) are

$$\begin{bmatrix} \alpha_2 \\ \alpha_1 \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} \frac{\theta+1}{2} \\ \theta \\ \frac{\theta-1}{2} \end{bmatrix}, \quad \begin{bmatrix} \beta_{2,n} \\ \beta_{1,n} \\ \beta_{0,n} \end{bmatrix} = \begin{bmatrix} \frac{1}{4}\Big(1 + \frac{1-\theta^2}{(1+\epsilon_n\theta)^2} + \epsilon_n^2 \frac{\theta(1-\theta^2)}{(1+\epsilon_n\theta)^2} + \theta\Big) \\ \frac{1}{2}\Big(1 - \frac{1-\theta^2}{(1+\epsilon_n\theta)^2}\Big) \\ \frac{1}{4}\Big(1 + \frac{1-\theta^2}{(1+\epsilon_n\theta)^2} - \epsilon_n^2 \frac{\theta(1-\theta^2)}{(1+\epsilon_n\theta)^2} - \theta\Big) \end{bmatrix}.$$

The coefficients of $\{\alpha_j\}_{j=0:2}$, $\{\beta_{j,n}\}_{j=0:2}$ and the average time step $\alpha_2 k_n - \alpha_0 k_{n-1}$ are constructed to ensure the G-stability and second order accuracy of the method. Combining these fine properties, and existing numerical schemes for spatial discretization (e.g. finite element method [39, 40, 74, 108], two grid decoupled method [41, 66, 104, 131, 132], multi-grid decoupled method [2, 130], domain decomposition method [45, 64], etc.), this chapter provides complete variable time-stepping analysis for unsteady Stokes/Darcy Model (stability and error analysis) [107].

The remaining part of this chapter is organized as follows: we review the time-dependent Stokes/Darcy model (including necessary notations) in Section 4.2. Preliminaries and two lemmas about properties of the DLN algorithm (4.2) are presented in Section 4.3. In Section 4.4 and Section 4.5, we apply the variable time-stepping DLN algorithm (4.2) to the unsteady Stokes/Darcy model and offer detailed proofs of unconditional stability and second-order convergence of approximate solutions, which are rarely done in other papers. Two numerical tests are given in Section 4.6. The variable time-stepping test is aimed to verify the stability of the approximate solutions and is followed by a constant time-stepping example to confirm second-order convergence.

4.1.1 Related Work

Even though the DLN method has shown its success in the variable time-stepping analysis of Navier-Stokes equations [83], the standard methods (backward Euler, BDF2, Crank-Nicolson Leap Frog, etc.) would be a priority in numerical analysis of fluid flow. Recently artificial compression algorithm, changing the mass conservation condition $\nabla \cdot u = 0$ a little by $\delta p_t + \nabla \cdot u = 0$ $(0 < \delta \ll 1)$ and thus advancing the pressure explicitly, has been employed to improve the efficiency in time-stepping analysis of flow problems [23, 82, 90]. Additionally, it is possible that adding time filters on certain standard methods increases the order of convergence while keeping the conditional stability [58]. Furthermore, the mixture of time filter and artificial compression becomes a pioneering technique in computational fluid dynamics [59].

4.2 The Time-dependent Stokes/Darcy Model

In this section, we consider the unsteady Stokes/Darcy model in the region $\Omega = \Omega_f \cup \Omega_p$, where Ω_f is the incompressible fluid region and Ω_p is the porous media region. The two regions are separated by the interface denoted by $\Gamma = \overline{\Omega}_f \cap \overline{\Omega}_p$ and $\mathbf{n}_f$ and $\mathbf{n}_p$ are the unit outward normal vectors on $\partial\Omega_f$ and $\partial\Omega_p$. The schematic representation is displayed in Figure 18.

For the finite time interval $[0, T]$, the fluid motion in Ω_f is governed by the time-dependent Stokes equations, i.e. the fluid velocity $\mathbf{u}_f(\mathbf{x}, t)$ and the pressure $p(\mathbf{x}, t)$ satisfy

$$\begin{aligned} \frac{\partial \mathbf{u}_f}{\partial t} - \nabla \cdot \mathbb{T}\left(\mathbf{u}_f, p\right) = \boldsymbol{F}_1(\mathbf{x}, t) \qquad & \text{in } \Omega_f \times (0, T), \\ \nabla \cdot \mathbf{u}_f = 0 \qquad & \text{in } \Omega_f \times (0, T), \\ \mathbf{u}_f(\mathbf{x}, 0) = \mathbf{u}_f^0(\mathbf{x}) \qquad & \text{in } \Omega_f, \end{aligned} \tag{4.3}$$

where the stress tensor $\mathbb{T}$ and the deformation rate tensor $\mathbb{D}$ are defined as

$$\mathbb{T}(\mathbf{u}_f, p) = -p\mathbb{I} + 2\nu\mathbb{D}(\mathbf{u}_f), \quad \mathbb{D}(\mathbf{u}_f) = \frac{1}{2}(\nabla \mathbf{u}_f + \nabla^{\mathrm{tr}} \mathbf{u}_f)^1,$$

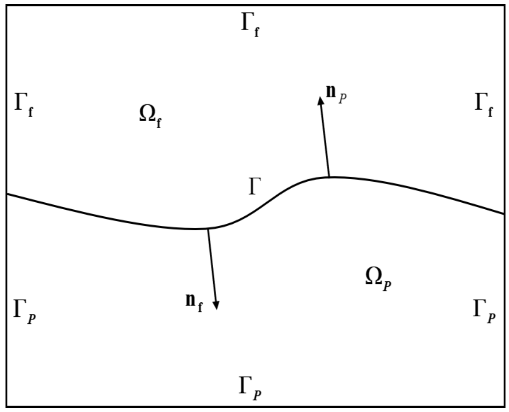


Figure 18: A global domain Ω consisting of a fluid flow region Ω_f and a porous media flow region Ω_p separated by an interface Γ.

$\nu > 0$ is the kinetic viscosity and $\boldsymbol{F}_1$ is the external force.

The velocity $\mathbf{u}_p(\mathbf{x}, t)$ and hydraulic head $\phi(\mathbf{x}, t)$ in porous media region are governed by Darcy's law and the saturated flow model

$$\begin{aligned} \mathbf{u}_p &= -\mathbf{K}\nabla\phi && \text{in } \Omega_p \times (0, T), && (4.4)\\ S_0\phi_t + \nabla \cdot \mathbf{u}_p &= F_2(\mathbf{x}, t) && \text{in } \Omega_p \times (0, T), && (4.5)\\ \phi(\mathbf{x}, 0) &= \phi^0(\mathbf{x}) && \text{in } \Omega_p, \end{aligned}$$

where positive symmetric tensor $\mathbf{K}$ denotes the hydraulic conductivity in Ω_p and is allowed to vary in space. S_0 is the specific mass storativity coefficient and F_2 is a source term. Combining (4.4) and (4.5), we obtain the Darcy equation which describes the hydraulic head:

$$S_0\phi_t - \nabla \cdot (\mathbf{K}\nabla\phi) = F_2(x, t), \qquad \text{in } \Omega_p \times (0, T). \tag{4.6}$$

Now we introduce the boundary conditions:

$$\begin{aligned} \mathbf{u}_f &= 0 \quad \text{on } (\partial\Omega_f \setminus \Gamma) \times (0, T),\\ \phi &= 0 \quad \text{on } (\partial\Omega_p \setminus \Gamma) \times (0, T), \end{aligned} \tag{4.7}$$

[1] tr means transpose of matrix or vector.

and the necessary interface conditions for the coupled Stokes/Darcy model:

$$\begin{aligned}
\mathbf{u}_f \cdot \mathbf{n}_f - \mathbf{K}\nabla\phi \cdot \mathbf{n}_p &= 0, && \text{on } \Gamma \times (0,T), \\
-\mathbf{n}_f \cdot \Big(\mathbb{T}(\mathbf{u}_f, p) \cdot \mathbf{n}_f\Big) &= g\phi, && \text{on } \Gamma \times (0,T), \\
-\boldsymbol{\tau}_i \cdot \Big(\mathbb{T}(\mathbf{u}_f, p) \cdot \mathbf{n}_f\Big) &= \frac{\mu_{BJS}\nu\sqrt{d}}{\sqrt{\mathrm{trace}(\Pi)}}\boldsymbol{\tau}_i \cdot \mathbf{u}_f, && \text{on } \Gamma \times (0,T),
\end{aligned} \tag{4.8}$$

where g is the gravitational constant and $\{\boldsymbol{\tau}_i\}_{i=1}^{d-1}$ are the orthonormal system of tangential vectors along Γ, μ_{BJS} is an experimentally determined parameter, Π represents the permeability and satisfies $\mathbf{K} = \frac{\Pi g}{\nu}$.

For the weak formulation of the unsteady Stokes/Darcy model, we define some function spaces:

$$\begin{aligned}
H_f &= \Big\{\mathbf{v} \in (H^1(\Omega_f))^d : \mathbf{v}|_{\Omega_f \backslash \Gamma} = 0\Big\}, \\
H_p &= \Big\{\psi \in H^1(\Omega_p) : \psi|_{\Omega_p \backslash \Gamma} = 0\Big\}, \\
\mathbf{U} &= H_f \times H_p, \qquad\qquad Q_f = L^2(\Omega_f).
\end{aligned}$$

We associate the space $\mathbf{U}$ with the following three norms: for all $\underline{\mathbf{v}} = (\mathbf{v}, \psi) \in \mathbf{U}$ and $1 \le s < \infty$

$$\begin{aligned}
\|\underline{\mathbf{v}}\|_0 = \sqrt{(\mathbf{v},\mathbf{v})_{\Omega_f} + gS_0(\psi,\psi)_{\Omega_p}}, \quad \|\underline{\mathbf{v}}\|_s = \sqrt{\|\mathbf{v}\|_s^2 + \|\psi\|_s^2}, \\
\|\underline{\mathbf{v}}\|_{\mathbf{U}} = \sqrt{\nu(\nabla\mathbf{v}, \nabla\mathbf{v})_{\Omega_f} + g(\mathbf{K}\nabla\psi, \nabla\psi)_{\Omega_p}}.
\end{aligned}$$

Here $(\cdot,\cdot)_\Omega$ and $\|\cdot\|$ denote the L^2-inner product and L^2-norm on function space $L^2(\Omega)$ respectively. $\|\cdot\|_s$ represents $\|\cdot\|_s$-norm of Sobolev space H^s.

By the positive definiteness of tensor $\mathbf{K}$ and Poincaré inequality, there exists constant $C_{0,\mathbf{U}} > 0$ such that

$$\|\underline{\mathbf{v}}\|_0 \le C_{0,\mathbf{U}}\|\underline{\mathbf{v}}\|_{\mathbf{U}}. \tag{4.9}$$

Now we combine (4.3), (4.6), (4.7) and (4.8) to derive the weak form of the time dependent Stokes/Darcy model: given $\mathbf{F} = (\mathbf{F}_1, \mathrm{F}_2) \in \mathrm{L}^2\big(0,\mathrm{T};(\mathrm{L}^2(\Omega_\mathrm{f}))^\mathrm{d}\big) \times \mathrm{L}^2\big(0,\mathrm{T};\mathrm{L}^2(\Omega_\mathrm{p})\big)$, find $\underline{\mathbf{u}} = (\mathbf{u}_f, \phi) \in \mathbf{U}$ and $p \in Q_f$ such that for all $\underline{\mathbf{v}} = (\mathbf{v}, \psi) \in \mathbf{U}$, $q \in Q_f$

$$\begin{aligned}
\Big\langle \frac{\partial \underline{\mathbf{u}}}{\partial t}, \underline{\mathbf{v}} \Big\rangle_0 + a(\underline{\mathbf{u}}, \underline{\mathbf{v}}) + b(\underline{\mathbf{v}}, p) &= \big\langle \mathbf{F}, \underline{\mathbf{v}} \big\rangle_{\mathbf{U}'}, \\
b(\underline{\mathbf{u}}, q) &= 0, \\
\underline{\mathbf{u}}(\mathbf{x}, 0) &= \underline{\mathbf{u}}^0(\mathbf{x}),
\end{aligned} \tag{4.10}$$

where

$$\begin{aligned}
\Big\langle \frac{\partial \underline{\mathbf{u}}}{\partial t}, \underline{\mathbf{v}} \Big\rangle_0 &= \Big(\frac{\partial \mathbf{u}_f}{\partial t}, \mathbf{v} \Big)_{\Omega_f} + g S_0 \Big(\frac{\partial \phi}{\partial t}, \psi \Big)_{\Omega_p}, \\
a(\underline{\mathbf{u}}, \underline{\mathbf{v}}) &= a_\Omega(\underline{\mathbf{u}}, \underline{\mathbf{v}}) + a_\Gamma(\underline{\mathbf{u}}, \underline{\mathbf{v}}), \\
a_\Omega(\underline{\mathbf{u}}, \underline{\mathbf{v}}) &= a_{\Omega_f}(\mathbf{u}, \mathbf{v}) + a_{\Omega_p}(\phi, \psi), \\
a_{\Omega_f}(\mathbf{u}, \mathbf{v}) &= \nu \big(\mathbb{D}(\mathbf{u}), \mathbb{D}(\mathbf{v}) \big)_{\Omega_f} + \Big(\frac{\mu_{BJS} \nu \sqrt{d}}{\sqrt{\mathrm{trace}(\Pi)}} P_{\boldsymbol{\tau}}(\mathbf{u}), P_{\boldsymbol{\tau}}(\mathbf{v}) \Big)_\Gamma, \\
a_{\Omega_p}(\phi, \psi) &= g(\mathbf{K} \nabla \phi, \nabla \psi)_{\Omega_p}, \\
a_\Gamma(\underline{\mathbf{u}}, \underline{\mathbf{v}}) &= g(\phi, \mathbf{v} \cdot \mathbf{n}_f)_\Gamma - g(\psi, \mathbf{u}_f \cdot \mathbf{n}_f)_\Gamma, \\
b(\underline{\mathbf{v}}, p) &= -(p, \nabla \cdot \mathbf{v})_{\Omega_f}, \\
\big\langle \mathbf{F}, \underline{\mathbf{v}} \big\rangle_{\mathbf{U}'} &= (\mathbf{F}_1, \mathbf{v})_{\Omega_f} + g(\mathrm{F}_2, \psi)_{\Omega_\mathrm{p}}, \\
\underline{\mathbf{u}}^0(\mathbf{x}) &= (\mathbf{u}_f^0(\mathbf{x}), \phi^0(\mathbf{x})).
\end{aligned}$$

$\mathbf{U}'$ is the dual space of $\mathbf{U}$ with the norm

$$\|\mathbf{F}\|_{\mathbf{U}'} = \sup_{\underline{\mathbf{v}} \in \mathbf{U} \setminus 0} \frac{\big\langle \mathbf{F}, \underline{\mathbf{v}} \big\rangle_{\mathbf{U}'}}{\|\underline{\mathbf{v}}\|_{\mathbf{U}}},$$

and $P_{\boldsymbol{\tau}}(\cdot)$ is the projection onto the local tangential plane, i.e. $P_{\boldsymbol{\tau}}(\mathbf{v}) = \mathbf{v} - (\mathbf{v} \cdot \mathbf{n}_f)\mathbf{n}_f$. The biliear form $a(\cdot, \cdot)$ is continuous and coercive: for all $\underline{\mathbf{u}}, \underline{\mathbf{v}} \in \mathbf{U}$,

$$\begin{aligned}
a(\underline{\mathbf{u}}, \underline{\mathbf{v}}) &\leq C_1 \|\underline{\mathbf{u}}\|_{\mathbf{U}} \|\underline{\mathbf{v}}\|_{\mathbf{U}}, \\
a(\underline{\mathbf{u}}, \underline{\mathbf{u}}) &\geq C_2 \|\underline{\mathbf{u}}\|^2_{\mathbf{U}}.
\end{aligned} \tag{4.11}$$

Here the above constants $C_1, C_2 > 0$ are independent of functions.

4.3 Preliminaries

For spatial discretization, we construct regular triangulations of Ω, Ω_f and Ω_p with diameter $h > 0$ and choose any finite element spaces $H_{fh} \subset H_f, Q_{fh} \subset Q_f, H_{ph} \subset H_p$ such that the pair (H_{fh}, Q_{fh}) satisfies the discrete LBB^h condition. Typical examples of such pair include Taylor-Hood (P2-P1) and MINI (P1b-P1). Then we define $\mathbf{U}_h = H_{fh} \times H_{ph}$ to be finite element space of $\mathbf{U}$. The discretely divergence-free subspace of H_{fh} is defined to be

$$V_{fh} := \Big\{ \mathbf{v}_h \in H_{fh} : \big(\nabla \cdot \mathbf{v}_h, q_h\big) = 0, \ \forall q_h \in Q_{fh} \Big\},$$

and the divergence free space of $\mathbf{U}_h$ to be $\mathbf{V}_h = V_{fh} \times H_{ph}$.

We define the linear projection operator (see [97]) $P_h = \big(P_h^{\mathbf{u}}, P_h^p\big)$ from $\mathbf{U} \times Q_f$ onto $\mathbf{U}_h \times Q_{fh}$: given $t \in (0, T]$ and $(\underline{\mathbf{u}}(t), p(t)) \in (\mathbf{U}, Q_f)$, $\big(P_h^{\mathbf{u}} \mathbf{u}(t), P_h^p p(t)\big)$ satisfies

$$\begin{aligned}
a(P_h^{\mathbf{u}} \underline{\mathbf{u}}(t), \underline{\mathbf{v}}_h) + b(\underline{\mathbf{v}}_h, P_h^p p(t)) &= a(\underline{\mathbf{u}}(t), \underline{\mathbf{v}}_h) + b(\underline{\mathbf{v}}_h, p(t)), \\
b(P_h^{\mathbf{u}} \underline{\mathbf{u}}(t), q_h) &= 0.
\end{aligned} \tag{4.12}$$

for all $\underline{\mathbf{v}}_h \in \mathbf{U}_h$, $q_h \in Q_{fh}$. Furthermore, the linear projection P_h defined above satisfies

$$\begin{aligned}
&\| P_h^{\mathbf{u}} \underline{\mathbf{u}}(t) - \underline{\mathbf{u}}(t) \|_0 \leq C_3 h^2 \| \underline{\mathbf{u}}(t) \|_2, \\
&\| P_h^{\mathbf{u}} \underline{\mathbf{u}}(t) - \underline{\mathbf{u}}(t) \|_{\mathbf{U}} \leq C_4 h \| \underline{\mathbf{u}}(t) \|_2, \\
&\| P_h^p p(t) - p(t) \| \leq C_5 h \| p(t) \|_1,
\end{aligned} \tag{4.13}$$

if the pair $(\underline{\mathbf{u}}(t), p(t))$ is smooth enough.

For the rest of the paper, $P = \{t_n\}_{n=0}^N$ is the partition on time interval $[0, T]$ with $t_0 = 0, t_N = T$ and $k_n = t_{n+1} - t_n$ is the time step size. Let $\underline{\mathbf{u}}_h^n$ and p_h^n denotes the approximate solutions of $\underline{\mathbf{u}}(t_n)$ and $p(t_n)$ by the DLN method (4.2) and for convenience we denote

$$\begin{aligned}
\underline{\mathbf{u}}_{h,\beta}^n &= \beta_{2,n} \underline{\mathbf{u}}_h^{n+1} + \beta_{1,n} \underline{\mathbf{u}}_h^n + \beta_{0,n} \underline{\mathbf{u}}_h^{n-1}, \\
\mathbf{F}_\beta^n &= \beta_{2,n} \mathbf{F}(t_{n+1}) + \beta_{1,n} \mathbf{F}(t_n) + \beta_{0,n} \mathbf{F}(t_{n-1}).
\end{aligned}$$

Then we have the discrete weak formulation for the unsteady Stokes/Darcy model by variable time-stepping DLN algorithm: given $\underline{\mathbf{u}}_h^n$, $\underline{\mathbf{u}}_h^{n-1}$, p_h^n, p_h^{n-1} , find $\underline{\mathbf{u}}_h^{n+1}$ and p_h^{n+1} such that for all $\underline{\mathbf{v}}_h \in \mathbf{U}_h$ and $q_h \in Q_{fh}$,

$$\Big\langle \frac{\alpha_2 \underline{\mathbf{u}}_h^{n+1} + \alpha_1 \underline{\mathbf{u}}_h^n + \alpha_0 \underline{\mathbf{u}}_h^{n-1}}{\alpha_2 k_n - \alpha_0 k_{n-1}}, \underline{\mathbf{v}}_h \Big\rangle_0 + a(\underline{\mathbf{u}}_{h,\beta}^n, \underline{\mathbf{v}}_h) + b(\underline{\mathbf{v}}_h, p_{h,\beta}^n) = \big\langle \mathbf{F}_\beta^n, \underline{\mathbf{v}}_h \big\rangle_{\mathbf{U}'},$$
$$b(\underline{\mathbf{u}}_h^{n+1}, q_h) = 0. \tag{4.14}$$

(4.14) has the equivalent form: for all $\underline{\mathbf{v}}_h \in \mathbf{V}_h$

$$\Big\langle \frac{\alpha_2 \underline{\mathbf{u}}_h^{n+1} + \alpha_1 \underline{\mathbf{u}}_h^n + \alpha_0 \underline{\mathbf{u}}_h^{n-1}}{\alpha_2 k_n - \alpha_0 k_{n-1}}, \underline{\mathbf{v}}_h \Big\rangle_0 + a(\underline{\mathbf{u}}_{h,\beta}^n, \underline{\mathbf{v}}_h) = \big\langle \mathbf{F}_\beta^n, \underline{\mathbf{v}}_h \big\rangle_{\mathbf{U}'}. \tag{4.15}$$

For stability and error analysis, we recall the following consistency results [31, 76, 77] on the interpolation and differentiation defects.

Lemma 6. *Let $y : \Omega \times [0, T] \to \mathbb{R}^d$ be arbitrary function smooth enough, then for $\theta \in [0, 1)$*

$$\Big\| \sum_{j=0}^{2} \beta_{j,n} y(t_{n-1+j}) - y(t_{n,\beta}) \Big\|^2 \le C(\theta)(k_n + k_{n-1})^3 \int_{t_{n-1}}^{t_{n+1}} \|y_{tt}\|^2 dt,$$

and for $\theta \in [0, 1)$

$$\Big\| \frac{\alpha_2 y(t_{n+1}) + \alpha_1 y(t_n) + \alpha_0 y(t_{n-1})}{\alpha_2 k_n - \alpha_0 k_{n-1}} - y_t(t_{n,\beta}) \Big\|^2 \le C(\theta)(k_n + k_{n-1})^3 \int_{t_{n-1}}^{t_{n+1}} \|y_{ttt}\|^2 dt,$$

where $t_{n,\beta} = \beta_{2,n} t_{n+1} + \beta_{1,n} t_n + \beta_{0,n} t_{n-1}$.

Proof. Apply Taylor theorem with integral reminder to $y(t_{n+1})$, $y(t_{n-1})$ and $y\,(t_{n,\beta})$ and expand these functions at point t_n. □

4.4 Stability Analysis

Now we apply G-stability of the DLN method to derive the following theorem about stability of variable time-stepping DLN algorithm on Stokes/Darcy model (4.15).

Theorem 7. *(Unconditional Stability) For any $N \geq 2$, the approximate solutions of the unsteady Stokes/Darcy model by the algorithm* (4.15) *satisfy*

$$\frac{1}{4}(1+\theta)\|\underline{\mathbf{u}}_h^N\|_0^2 + \frac{1}{4}(1-\theta)\|\underline{\mathbf{u}}_h^{N-1}\|_0^2 + \sum_{n=1}^{N-1}\Big\|\sum_{j=0}^{2}\lambda_{j,n}\underline{\mathbf{u}}_h^{n-1+j}\Big\|_0^2 + C(\theta)\sum_{n=1}^{N-1}(k_n+k_{n-1})\|\underline{\mathbf{u}}_{h,\beta}^n\|_U^2$$
$$\leq \frac{1}{4}(1+\theta)\|\underline{\mathbf{u}}_h^1\|_0^2 + \frac{1}{4}(1-\theta)\|\underline{\mathbf{u}}_h^0\|_0^2 + \widetilde{C}(\theta)\sum_{n=1}^{N-1}(k_n+k_{n-1})\|\mathbf{F}_\beta^n\|_{U'}^2. \tag{4.16}$$

Here, the constants $C(\theta), \widetilde{C}(\theta) \geq 0$ are independent of the diameter h and time step k_n.

Proof. Let $\underline{\mathbf{v}}_h = \underline{\mathbf{u}}_{h,\beta}^n$ in (4.15) and multiply both sides of the equation by $\alpha_2 k_n - \alpha_0 k_{n-1}$,

$$\Big(\sum_{j=0}^{2}\alpha_j\underline{\mathbf{u}}_h^{n-1+j}, \sum_{j=0}^{2}\beta_{j,n}\underline{\mathbf{u}}_h^{n-1+j}\Big)_0 + (\alpha_2 k_n - \alpha_0 k_{n-1})a(\underline{\mathbf{u}}_{h,\beta}^n, \underline{\mathbf{u}}_{h,\beta}^n)$$
$$= (\alpha_2 k_n - \alpha_0 k_{n-1})\big\langle \mathbf{F}_\beta^n, \underline{\mathbf{u}}_{h,\beta}^n\big\rangle_{\mathbf{U}'}. \tag{4.17}$$

Apply the Proposition 2 and replace L^2 space by $\mathbf{U}$ and L^2-norm by $\|\cdot\|_0$-norm, we obtain

$$\Big(\sum_{j=0}^{2}\alpha_j\underline{\mathbf{u}}_h^{n-1+j}, \sum_{j=0}^{2}\beta_{j,n}\underline{\mathbf{u}}_h^{n-1+j}\Big)_0 = \left\|\begin{matrix}\underline{\mathbf{u}}_h^{n+1}\\ \underline{\mathbf{u}}_h^n\end{matrix}\right\|_{G(\theta)}^2 - \left\|\begin{matrix}\underline{\mathbf{u}}_h^{n}\\ \underline{\mathbf{u}}_h^{n-1}\end{matrix}\right\|_{G(\theta)}^2 + \Big\|\sum_{j=0}^{2}\lambda_{j,n}\underline{\mathbf{u}}_h^{n-1+j}\Big\|_0^2, \tag{4.18}$$

where $\{\lambda_{j,n}\}_{j=0}^2$ correspond to the coefficients $\{\gamma_j^{(n)}\}_{j=0}^2$ in the Proposition 2. By the definition of the $G(\theta)$-norm in (2.8)

$$\left\|\begin{matrix}\underline{\mathbf{u}}_h^{n+1}\\ \underline{\mathbf{u}}_h^n\end{matrix}\right\|_{G(\theta)}^2 := \frac{1}{4}(1+\theta)\|\underline{\mathbf{u}}_h^{n+1}\|_0^2 + \frac{1}{4}(1-\theta)\|\underline{\mathbf{u}}_h^n\|_0^2. \tag{4.19}$$

We combine (4.11), (4.17), (4.18) and use Cauchy Schwarz inequality:

$$\left\| \begin{matrix} \underline{\mathbf{u}}_h^{n+1} \\ \underline{\mathbf{u}}_h^{n} \end{matrix} \right\|_{G(\theta)}^2 - \left\| \begin{matrix} \underline{\mathbf{u}}_h^{n} \\ \underline{\mathbf{u}}_h^{n-1} \end{matrix} \right\|_{G(\theta)}^2 + \Big\| \sum_{j=0}^{2} \lambda_{j,n} \underline{\mathbf{u}}_h^{n-1+j} \Big\|_0^2 + C_2(\alpha_2 k_n - \alpha_0 k_{n-1}) \|\underline{\mathbf{u}}_{h,\beta}^n\|_{\mathbf{U}}^2$$
$$\leq \frac{C_2}{2}(\alpha_2 k_n - \alpha_0 k_{n-1}) \|\underline{\mathbf{u}}_{h,\beta}^n\|_{\mathbf{U}}^2 + \frac{1}{2C_2}(\alpha_2 k_n - \alpha_0 k_{n-1}) \|\mathbf{F}_\beta^n\|_{\mathbf{U}'}^2. \tag{4.20}$$

Note that

$$\frac{1-\theta}{2}(k_n + k_{n-1}) \leq \alpha_2 k_n - \alpha_0 k_{n-1} \leq \frac{1+\theta}{2}(k_n + k_{n-1}), \tag{4.21}$$

then (4.20) becomes

$$\left\| \begin{matrix} \underline{\mathbf{u}}_h^{n+1} \\ \underline{\mathbf{u}}_h^{n} \end{matrix} \right\|_{G(\theta)}^2 - \left\| \begin{matrix} \underline{\mathbf{u}}_h^{n} \\ \underline{\mathbf{u}}_h^{n-1} \end{matrix} \right\|_{G(\theta)}^2 + \Big\| \sum_{j=0}^{2} \lambda_{j,n} \underline{\mathbf{u}}_h^{n-1+j} \Big\|_0^2 + \frac{C_2(1-\theta)}{4}(k_n + k_{n-1}) \|\underline{\mathbf{u}}_{h,\beta}^n\|_{\mathbf{U}}^2$$
$$\leq \frac{1+\theta}{4C_2}(k_n + k_{n-1}) \|\mathbf{F}_\beta^n\|_{\mathbf{U}'}^2. \tag{4.22}$$

Summing over (4.22) from $n = 1, \cdots, N-1$ and applying (4.19), we obtain the result. □

4.5 Error Analysis

In this section, we utilize the G-stability (Proposition 2) and consistency (Lemma 6) of the DLN algorithm to show the second-order convergence of approximate solutions to unsteady Stokes/Darcy model. We denote $\underline{\mathbf{u}}^n = (\mathbf{u}^n, \phi^n)$ and p^n be the exact solutions of the coupled Stokes/Darcy model (4.10) at time t_n and define the error functions

$$\begin{aligned} \mathbf{e}^n &= \underline{\mathbf{u}}_h^n - \underline{\mathbf{u}}^n = (\underline{\mathbf{u}}_h^n - P_h^{\mathbf{u}} \underline{\mathbf{u}}^n) - (\underline{\mathbf{u}}^n - P_h^{\mathbf{u}} \underline{\mathbf{u}}^n) = \eta^n - \xi^n. \\ e_p^n &= p_h^n - p^n = (p_h^n - P_h^p p^n) - (p^n - P_h^p p^n) = \eta_p^n - \xi_p^n, \end{aligned} \tag{4.23}$$

and $\eta^0 = \eta^1 = 0$.

For variable time-stepping analysis, we need to define some continuous and discrete norms. Given $\underline{\mathbf{v}} \in \mathbf{U}$, $q \in Q_f$, $\mathbf{G} \in \mathbf{U}'$ and $1 \le m, s < \infty$, we define continuous norms

$$\|\underline{\mathbf{v}}\|_{m,0} := \Big(\int_0^T \|\underline{\mathbf{v}}(t)\|_0^m dt \Big)^{\frac{1}{m}}, \ \|\underline{\mathbf{v}}\|_{m,s} := \Big(\int_0^T \|\underline{\mathbf{v}}(t)\|_s^m dt \Big)^{\frac{1}{m}}, \ \|\underline{\mathbf{v}}\|_{m,\mathbf{U}} := \Big(\int_0^T \|\underline{\mathbf{v}}(t)\|_{\mathbf{U}}^m dt \Big)^{\frac{1}{m}},$$
$$\|q\|_{m,L^2} := \Big(\int_0^T \|q(t)\|^m dt \Big)^{\frac{1}{m}}, \quad \|\mathbf{G}\|_{m,\mathbf{U}'} := \Big(\int_0^T \|G(t)\|_{\mathbf{U}'}^m dt \Big)^{\frac{1}{m}}.$$

and new discrete norms

$$|||\underline{\mathbf{v}}|||_{m,0} := \Big(\sum_{n=0}^{N-1} k_n \|\underline{\mathbf{v}}^{n+1}\|_0^m \Big)^{\frac{1}{m}}, \quad |||\underline{\mathbf{v}}|||_{m,s} := \Big(\sum_{n=0}^{N-1} k_n \|\underline{\mathbf{v}}^{n+1}\|_s^m \Big)^{1/m},$$
$$|||\underline{\mathbf{v}}_{\beta}|||_{m,s} := \Big(\sum_{n=1}^{N-1} (k_{n-1} + k_n) \|\underline{\mathbf{v}}(t_{n,\beta})\|_s^m \Big)^{1/m}.$$

Now we have the main theorem for error analysis.

Theorem 8. *(Second Order Convergence) The approximate solutions* $\{\underline{\mathbf{u}}_h^n\}_{n=0}^N$ *by the variable timestepping DLN scheme* (4.15) *with parameter* $\theta \in [0, 1)$ *satisfies*

$$\begin{aligned} |||\underline{\mathbf{u}}_h - \underline{\mathbf{u}}|||_{2,0} \le & C(\theta, T) \Big\{ \max_{1 \le n \le N-1} \big\{ (k_n + k_{n-1})^2 \big\} \big(\|p_{tt}\|_{2,L^2} + \|\underline{\mathbf{u}}_{ttt}\|_{2,0} + \|\underline{\mathbf{u}}_{tt}\|_{2,\mathbf{U}} + \|\mathbf{F}_{tt}\|_{2,\mathbf{U}'} \big) \\ & + h^2 \|\underline{\mathbf{u}}_t\|_{2,2} + h^2 |||\underline{\mathbf{u}}|||_{2,2} \Big\}, \end{aligned} \tag{4.24}$$

and

$$\begin{aligned} & \Big(\sum_{n=1}^{N-1} (\alpha_2 k_n - \alpha_0 k_{n-1}) \|\underline{\mathbf{u}}(t_{n,\beta}) - \underline{\mathbf{u}}_{h,\beta}^n\|_{\mathbf{U}}^2 \Big)^{1/2} \\ & \le C(\theta) \max_{1 \le n \le N-1} \big\{ (k_n + k_{n-1})^2 \big\} \big(\|p_{tt}\|_{2,L^2} + \|\underline{\mathbf{u}}_{ttt}\|_{2,0} + \|\underline{\mathbf{u}}_{tt}\|_{2,\mathbf{U}} + \|\mathbf{F}_{tt}\|_{2,\mathbf{U}'} \big) \\ & + C(\theta, T) h^2 \|\underline{\mathbf{u}}_t\|_{2,2} + C(\theta) h \max_{1 \le n \le N-1} \big\{ (k_n + k_{n-1})^2 \big\} \|\underline{\mathbf{u}}_{tt}\|_{2,2} + C(\theta) h |||\underline{\mathbf{u}}_{\beta}|||_{2,2}. \end{aligned} \tag{4.25}$$

Proof. By (4.10), the true solution of unsteady Stokes/Darcy model at time $t_{n,\beta}$ satisfies

$$\Big\langle \frac{\partial \underline{\mathbf{u}}}{\partial t}(t_{n,\beta}), \underline{\mathbf{v}}_h \Big\rangle_0 + a(\underline{\mathbf{u}}(t_{n,\beta}), \underline{\mathbf{v}}_h) + b(\underline{\mathbf{v}}_h, p(t_{n,\beta})) = \big\langle \mathbf{F}(t_{n,\beta}), \underline{\mathbf{v}}_h \big\rangle_{\mathbf{U}'}, \quad \text{for all} \ \ \underline{\mathbf{v}}_h \in \mathbf{V}_h. \tag{4.26}$$

Equivalently, (4.26) can be rewritten as

$$\begin{aligned} \Big\langle \frac{\alpha_2 \underline{\mathbf{u}}^{n+1} + \alpha_1 \underline{\mathbf{u}}^n + \alpha_0 \underline{\mathbf{u}}^{n-1}}{\alpha_2 k_n - \alpha_0 k_{n-1}}, \underline{\mathbf{v}}_h \Big\rangle_0 &+ a(\underline{\mathbf{u}}_\beta^n, \underline{\mathbf{v}}_h) + b(\underline{\mathbf{v}}_h, p_\beta^n) \\ &= \big\langle \mathbf{F}_\beta^n, \underline{\mathbf{v}}_h \big\rangle_{\mathbf{U}'} + \tau\big(\underline{\mathbf{u}}(t_{n,\beta}), p(t_{n,\beta}), \underline{\mathbf{v}}_h\big), \end{aligned} \tag{4.27}$$

where

$$\begin{aligned} \underline{\mathbf{u}}_\beta^n &= \beta_{2,n} \underline{\mathbf{u}}^{n+1} + \beta_{1,n} \underline{\mathbf{u}}^n + \beta_{0,n} \underline{\mathbf{u}}^{n-1}, \quad p_\beta^n = \beta_{2,n} p^{n+1} + \beta_{1,n} p^n + \beta_{0,n} p^{n-1}, \\ \tau\big(\underline{\mathbf{u}}(t_{n,\beta}), p(t_{n,\beta}), \underline{\mathbf{v}}_h\big) &= \Big\langle \frac{\alpha_2 \underline{\mathbf{u}}^{n+1} + \alpha_1 \underline{\mathbf{u}}^n + \alpha_0 \underline{\mathbf{u}}^{n-1}}{\alpha_2 k_n - \alpha_0 k_{n-1}} - \frac{\partial \underline{\mathbf{u}}}{\partial t}(t_{n,\beta}), \underline{\mathbf{v}}_h \Big\rangle_0 + a(\underline{\mathbf{u}}_\beta^n - \underline{\mathbf{u}}(t_{n,\beta}), \underline{\mathbf{v}}_h) \\ &\quad + b(\underline{\mathbf{v}}_h, p_\beta^n - p(t_{n,\beta})) - \big\langle \mathbf{F}_\beta^n - \mathbf{F}(t_{n,\beta}), \underline{\mathbf{v}}_h \big\rangle_{\mathbf{U}'}. \end{aligned}$$

We subtract (4.27) from first equation of (4.14) and use the definition of error function in (4.23) to obtain: for all $\underline{\mathbf{v}}_h \in \mathbf{V}_h$,

$$\begin{aligned} &\Big\langle \frac{\alpha_2 \eta^{n+1} + \alpha_1 \eta^n + \alpha_0 \eta^{n-1}}{\alpha_2 k_n - \alpha_0 k_{n-1}}, \underline{\mathbf{v}}_h \Big\rangle_0 + a(\eta_\beta^n, \underline{\mathbf{v}}_h) + b(\underline{\mathbf{v}}_h, \eta_{p,\beta}^n) \\ = &\Big\langle \frac{\alpha_2 \xi^{n+1} + \alpha_1 \xi^n + \alpha_0 \xi^{n-1}}{\alpha_2 k_n - \alpha_0 k_{n-1}}, \underline{\mathbf{v}}_h \Big\rangle_0 + a(\xi_\beta^n, \underline{\mathbf{v}}_h) + b(\underline{\mathbf{v}}_h, \xi_{p,\beta}^n) - \tau\big(\underline{\mathbf{u}}(t_{n,\beta}), p(t_{n,\beta}), \underline{\mathbf{v}}_h\big), \end{aligned} \tag{4.28}$$

where

$$\begin{aligned} \eta_\beta^n &= \beta_{2,n} \eta^{n+1} + \beta_{1,n} \eta^n + \beta_{0,n} \eta^{n-1}, \qquad \xi_\beta^n = \beta_{2,n} \xi^{n+1} + \beta_{1,n} \xi^n + \beta_{0,n} \xi^{n-1}, \\ \eta_{p,\beta}^n &= \beta_{2,n} \eta_p^{n+1} + \beta_{1,n} \eta_p^n + \beta_{0,n} \eta_p^{n-1}, \quad \xi_{p,\beta}^n = \beta_{2,n} \xi_p^{n+1} + \beta_{1,n} \xi_p^n + \beta_{0,n} \xi_p^{n-1}. \end{aligned}$$

By definition of discrete divergence-free space $\mathbf{V}_h$ and the definition of projection operator P_h, we have

$$b(\underline{\mathbf{v}}_h, \eta_{p,\beta}^n) = 0 \quad \text{and} \quad a(\xi_\beta^n, \underline{\mathbf{v}}_h) + b(\underline{\mathbf{v}}_h, \xi_{p,\beta}^n) = 0. \tag{4.29}$$

Now we choose $\underline{\mathbf{v}}_h = \eta^n_\beta$ in (4.28) and then apply (4.29) and Proposition 2 to the equation (4.28) to obtain

$$\begin{aligned}
&\left\|\begin{matrix}\eta^{n+1}\\ \eta^n\end{matrix}\right\|^2_{G(\theta)} - \left\|\begin{matrix}\eta^{n}\\ \eta^{n-1}\end{matrix}\right\|^2_{G(\theta)} + \Big\|\sum_{j=0}^{2}\lambda_{j,n}\eta^{n-1+j}\Big\|^2_0 + C_2(\alpha_2 k_n - \alpha_0 k_{n-1})\|\eta^n_\beta\|^2_{\mathbf{U}}\\
&\le (\alpha_2\xi^{n+1} + \alpha_1\xi^n + \alpha_0\xi^{n-1}, \eta^n_\beta) + (\alpha_2 k_n - \alpha_0 k_{n-1})\tau\big(\underline{\mathbf{u}}(t_{n,\beta}), p(t_{n,\beta}), \eta^n_\beta\big). \qquad (4.30)
\end{aligned}$$

Using the Taylor theorem with integral reminder,

$$\underline{\mathbf{u}}^n = \underline{\mathbf{u}}^{n+1} + \int_{t_{n+1}}^{t_n}\underline{\mathbf{u}}_t dt \quad \text{and} \quad \underline{\mathbf{u}}^{n-1} = \underline{\mathbf{u}}^{n+1} + \int_{t_{n+1}}^{t_{n-1}}\underline{\mathbf{u}}_t dt. \qquad (4.31)$$

Then by (4.31) and the fact $\alpha_2 + \alpha_1 + \alpha_0 = 0$, we have

$$\begin{aligned}
\|\alpha_2\xi^{n+1} + \alpha_1\xi^n + \alpha_0\xi^{n-1}\|_0 &= \Big\|\alpha_1\int_{t_{n+1}}^{t_n}(P_h^{\mathbf{u}} - Id)\underline{\mathbf{u}}_t dt + \alpha_0\int_{t_{n+1}}^{t_{n-1}}(P_h^{\mathbf{u}} - Id)\underline{\mathbf{u}}_t dt\Big\|_0\\
&\le C(\theta)\int_{t_{n-1}}^{t_{n+1}}\|(P_h^{\mathbf{u}} - Id)\underline{\mathbf{u}}_t\|_0 dt, \qquad (4.32)
\end{aligned}$$

where Id is the identity mapping. Thus by (4.9), (4.32), Cauchy Schwarz inequality and Young's inequality,

$$\begin{aligned}
\big\langle \alpha_2\xi^{n+1} + \alpha_1\xi^n + \alpha_0\xi^{n-1}, \eta^n_\beta\big\rangle_0 \le & C(\theta, T)\int_{t_{n-1}}^{t_{n+1}}\|(P_h^{\mathbf{u}} - Id)\underline{\mathbf{u}}_t\|^2_0 dt\\
&+ \frac{C_2(\alpha_2 k_n - \alpha_0 k_{n-1})}{2}\|\eta^n_\beta\|^2_{\mathbf{U}}. \qquad (4.33)
\end{aligned}$$

Summing over (4.30) from $n = 1, 2, ..., M$ $(1 \le M \le N-1)$ and using (4.33),

$$\begin{aligned}
&\left\|\begin{matrix}\eta^{M+1}\\ \eta^M\end{matrix}\right\|^2_{G(\theta)} - \left\|\begin{matrix}\eta^{1}\\ \eta^{0}\end{matrix}\right\|^2_{G(\theta)} + \sum_{n=1}^{M}\Big\|\sum_{j=0}^{2}\lambda_{j,n}\eta^{n-1+j}\Big\|^2_0 + \sum_{n=1}^{M}\frac{C_2(\alpha_2 k_n - \alpha_0 k_{n-1})}{2}\|\eta^n_\beta\|^2_U\\
\le & C(\theta, T)\sum_{n=1}^{M}\int_{t_{n-1}}^{t_{n+1}}\|(P_h^{\mathbf{u}} - Id)\underline{\mathbf{u}}_t\|^2_0 dt + \sum_{n=1}^{M}(\alpha_2 k_n - \alpha_0 k_{n-1})\tau\big(\underline{\mathbf{u}}(t_{n,\beta}), p(t_{n,\beta}), \eta^n_\beta\big). \qquad (4.34)
\end{aligned}$$

Then we deal with four terms of $\tau\left(\underline{\mathbf{u}}(t_{n,\beta}), p(t_{n,\beta}), \eta^n_\beta\right)$ respectively. Combining (4.9), (4.11), Lemma 6 and using Cauchy Schwarz inequality, Young's inequality again, we obtain

$$\begin{aligned}
\Big\langle \frac{\alpha_2\underline{\mathbf{u}}^{n+1} + \alpha_1\underline{\mathbf{u}}^n + \alpha_0\underline{\mathbf{u}}^{n-1}}{\alpha_2 k_n - \alpha_0 k_{n-1}} - \frac{\partial\underline{\mathbf{u}}}{\partial t}(t_{n,\beta}), \eta^{n+1}_\beta\Big\rangle_0 \le & C(\theta)(k_n + k_{n-1})^3\int_{t_{n-1}}^{t_{n+1}}\|\underline{\mathbf{u}}_{ttt}\|^2_0 dt\\
&+ \frac{C_2}{16}\|\eta^n_\beta\|^2_{\mathbf{U}},
\end{aligned}$$

$$
\begin{aligned}
a(\underline{\mathbf{u}}_\beta^n - \underline{\mathbf{u}}(t_{n,\beta}), \eta_\beta^n) \leq & C_1 \|\underline{\mathbf{u}}_\beta^n - \underline{\mathbf{u}}(t_{n,\beta})\|_{\mathbf{U}} \|\eta_\beta^n\|_{\mathbf{U}} \\
\leq & C(\theta)(k_n + k_{n-1})^3 \int_{t_{n-1}}^{t_{n+1}} \|\underline{\mathbf{u}}_{tt}\|_{\mathbf{U}}^2 dt + \frac{C_2}{16} \|\eta_\beta^n\|_{\mathbf{U}}^2,
\end{aligned}
$$

$$
\begin{aligned}
b(\eta_\beta^{n+1}, p_\beta^n - p(t_{n,\beta})) \leq & C\|p_\beta^{n+1} - p(t_{n,\beta})\| \|\eta_\beta^n\|_{\mathbf{U}} \\
\leq & C(k_n + k_{n-1})^3 \int_{t_{n-1}}^{t_{n+1}} \|p_{tt}\|^2 dt + \frac{C_2}{16} \|\eta_\beta^n\|_{\mathbf{U}}^2,
\end{aligned}
$$

$$
\begin{aligned}
\left\langle \mathbf{F}_\beta^n - \mathbf{F}(t_{n,\beta}), \eta_\beta^n \right\rangle_{\mathbf{U}'} \leq & \|\mathbf{F}_\beta^n - \mathbf{F}(t_{n,\beta})\|_{\mathbf{U}'} \|\eta_\beta^n\|_{\mathbf{U}} \\
\leq & C(k_n + k_{n-1})^3 \int_{t_{n-1}}^{t_{n+1}} \|\mathbf{F}_{tt}\|_{\mathbf{U}'}^2 + \frac{C_2}{16} \|\eta_\beta^n\|_{\mathbf{U}}^2.
\end{aligned} \tag{4.35}
$$

Since $\eta^0 = \eta^1 = 0$ and by (4.13), (4.21), the definition of $G(\theta)$-norm and estimators in (4.35), (4.34) becomes

$$
\begin{aligned}
&\frac{1+\theta}{4}\|\eta^{M+1}\|_0^2 + \frac{1-\theta}{4}\|\eta^M\|_0^2 + \sum_{n=1}^{M} \Big\| \sum_{j=0}^{2} \lambda_{j,n} \eta^{n-1+j} \Big\|_0^2 + \frac{C_2}{4} \sum_{n=1}^{M} (\alpha_2 k_n - \alpha_0 k_{n-1}) \|\eta_\beta^n\|_{\mathbf{U}}^2 \\
&\quad \leq C(\theta) \max_{1 \leq n \leq N-1} \left\{ (k_n + k_{n-1})^4 \right\} \Big(\|p_{tt}\|_{2,L^2}^2 + \|\underline{\mathbf{u}}_{ttt}\|_{2,0}^2 + \|\underline{\mathbf{u}}_{tt}\|_{2,\mathbf{U}}^2 + \|\mathbf{F}_{tt}\|_{2,\mathbf{U}'}^2 \Big) \\
&\qquad\qquad + \sum_{n=1}^{M-1} C(\theta, T) \int_{t_{n-1}}^{t_{n+1}} h^4 \|\underline{\mathbf{u}}_t\|_2^2 dt \\
&\quad \leq C(\theta) \max_{1 \leq n \leq N-1} \left\{ (k_n + k_{n-1})^4 \right\} \Big(\|p_{tt}\|_{2,L^2}^2 + \|\underline{\mathbf{u}}_{ttt}\|_{2,0}^2 + \|\underline{\mathbf{u}}_{tt}\|_{2,\mathbf{U}}^2 + \|\mathbf{F}_{tt}\|_{2,\mathbf{U}'}^2 \Big) \\
&\qquad\qquad + C(\theta, T) h^4 \|\underline{\mathbf{u}}_t\|_{2,2}^2.
\end{aligned} \tag{4.36}
$$

Note that

$$
|||\mathbf{e}|||_{2,0} \leq |||\xi|||_{2,0} + |||\eta|||_{2,0}, \tag{4.37}
$$

by (4.13) and (4.36), we have

$$
|||\xi|||_{2,0} = \Big(\sum_{n=0}^{N-1} k_n \|\xi^{n+1}\|_0^2 \Big)^{1/2} \leq \Big(\sum_{n=0}^{N-1} C_3 h^4 k_n \|\underline{\mathbf{u}}^{n+1}\|_2^2 \Big)^{1/2} \leq Ch^2 |||\underline{\mathbf{u}}|||_{2,2}, \tag{4.38}
$$

$$
\begin{aligned}
|||\eta|||_{2,0} \leq & C(\theta,T)\big(\sum_{n=0}^{N-1} k_n\big)^{1/2}\Big\{\max_{1\leq n\leq N-1}\{(k_n+k_{n-1})^2\}\Big(\|p_{tt}\|_{2,L^2}+\|\underline{\mathbf{u}}_{ttt}\|_{2,0}+\|\underline{\mathbf{u}}_{tt}\|_{2,\mathbf{U}} \\
& \qquad + \|\mathbf{F}_{tt}\|_{2,\mathbf{U}'}\Big) + h^2\|\underline{\mathbf{u}}_t\|_{2,2}\Big\} \\
\leq & C(\theta,T)\sqrt{T}\Big\{\max_{1\leq n\leq N-1}\{(k_n+k_{n-1})^2\}\Big(\|p_{tt}\|_{2,L^2}+\|\underline{\mathbf{u}}_{ttt}\|_{2,0}+\|\underline{\mathbf{u}}_{tt}\|_{2,\mathbf{U}}+\|\mathbf{F}_{tt}\|_{2,\mathbf{U}'}\Big) \\
& \qquad + h^2\|\underline{\mathbf{u}}_t\|_{2,2}\Big\}.
\end{aligned} \tag{4.39}
$$

Combining (4.37), (4.38) and (4.39), we have (4.24). For the second part, we have

$$
\begin{aligned}
& \sum_{n=1}^{N-1}(\alpha_2 k_n - \alpha_0 k_{n-1})\|\underline{\mathbf{u}}(t_{n,\beta}) - \underline{\mathbf{u}}_{h,\beta}^n\|_{\mathbf{U}}^2 \\
\leq & C(\theta)\sum_{n=1}^{N-1}(k_n+k_{n-1})\|\underline{\mathbf{u}}(t_{n,\beta}) - \underline{\mathbf{u}}_\beta^n\|_{\mathbf{U}}^2 + \sum_{n=1}^{N-1}(\alpha_2 k_n - \alpha_0 k_{n-1})\|\underline{\mathbf{u}}_\beta^n - \underline{\mathbf{u}}_{h,\beta}^n\|_{\mathbf{U}}^2.
\end{aligned} \tag{4.40}
$$

Using Lemma 6,

$$
C(\theta)\sum_{n=1}^{N-1}(k_n+k_{n-1})\|\underline{\mathbf{u}}(t_{n,\beta}) - \underline{\mathbf{u}}_\beta^n\|_{\mathbf{U}}^2 \leq C(\theta)\max_{1\leq n\leq N-1}\{(k_n+k_{n-1})^4\}\|\underline{\mathbf{u}}_{tt}\|_{2,\mathbf{U}}^2. \tag{4.41}
$$

And

$$
\sum_{n=1}^{N-1}(\alpha_2 k_n - \alpha_0 k_{n-1})\left\|\underline{\mathbf{u}}_\beta^n - \underline{\mathbf{u}}_{h,\beta}^n\right\|_{\mathbf{U}}^2 \leq C(\theta)\sum_{n=1}^{N-1}(k_n+k_{n-1})\left\|\xi_\beta^n\right\|_{\mathbf{U}}^2 + \sum_{n=1}^{N-1}(\alpha_2 k_n - \alpha_0 k_{n-1})\left\|\eta_\beta^n\right\|_{\mathbf{U}}^2. \tag{4.42}
$$

By (4.13) and linearity of the projection operator P_h,

$$
\|\xi_\beta^n\|_{\mathbf{U}}^2 = \|P_h^{\mathbf{u}}\underline{\mathbf{u}}_\beta^n - \underline{\mathbf{u}}_\beta^n\|_{\mathbf{U}}^2 \leq Ch^2\|\underline{\mathbf{u}}_\beta^n\|_2^2 \leq Ch^2\|\underline{\mathbf{u}}_\beta^n - \underline{\mathbf{u}}(t_{n,\beta})\|_2^2 + Ch^2\|\underline{\mathbf{u}}(t_{n,\beta})\|_2^2. \tag{4.43}
$$

Applying Lemma 6 again to (4.43),

$$
C(\theta)\sum_{n=1}^{N-1}(k_n+k_{n-1})\|\xi_\beta^n\|_{\mathbf{U}}^2 \leq C(\theta)h^2\max_{1\leq n\leq N-1}\{(k_n+k_{n-1})^4\}\|\underline{\mathbf{u}}_{tt}\|_{2,2}^2 + C(\theta)h^2|||\underline{\mathbf{u}}_\beta|||_{2,2}^2. \tag{4.44}
$$

Combining (4.36), (4.40), (4.41), (4.42) and (4.44), we obtain

$$
\begin{aligned}
&\sum_{n=1}^{N-1}(\alpha_2 k_n - \alpha_0 k_{n-1})\|\underline{\mathbf{u}}(t_{n,\beta}) - \underline{\mathbf{u}}_{h,\beta}^n\|_{\mathbf{U}}^2 \\
\leq\ & C(\theta) \max_{1\leq n\leq N-1}\left\{(k_n + k_{n-1})^4\right\}\Big(\|p_{tt}\|_{2,L^2}^2 + \|\underline{\mathbf{u}}_{ttt}\|_{2,0}^2 + \|\underline{\mathbf{u}}_{tt}\|_{2,\mathbf{U}}^2 + \|\mathbf{F}_{tt}\|_{2,\mathbf{U}'}^2\Big) \\
&+C(\theta,T)h^4\|\underline{\mathbf{u}}_t\|_{2,2}^2 + C(\theta)h^2 \max_{1\leq n\leq N-1}\left\{(k_n + k_{n-1})^4\right\}\|\underline{\mathbf{u}}_{tt}\|_{2,2}^2 + C(\theta)h^2|||\underline{\mathbf{u}}_{\beta}|||_{2,2}^2,
\end{aligned}
$$

which leads to (4.25).

□

4.6 Numerical Tests

In this section, we use two numerical experiments to verify two distinct properties of the DLN algorithm (stability and consistency). Both numerical tests are implemented by FreeFEM++. The first test confirms that the variable time-stepping DLN algorithm is stable with different values of parameter $\theta \in [0, 1]$. In the second experiment, we apply the DLN algorithm with constant time step to check the second-order convergence of the approximate solutions as well as compare it with the BDF2 scheme.

4.6.1 Test of Variable Time-stepping DLN algorithm

In this experiment, we use the example mentioned in [19, 73]. Considering the model problem on $\Omega_f = [0, \pi] \times [0, 1]$ and $\Omega_p = [0, \pi] \times [-1, 0]$ with the interface $\Gamma = [0, \pi] \times [0]$:

$$
\begin{aligned}
\mathbf{u} &= \Big[\tfrac{1}{\pi}\sin(2\pi y)\cos(x)e^t, (-2 + \tfrac{1}{\pi^2}\sin^2(\pi y))\sin(x)e^t\Big]^{\mathrm{tr}}, \\
p &= 0, \qquad \phi = (e^y - e^{-y})\sin(x)e^t.
\end{aligned}
$$

For this test, we set the physical parameters ρ, g, ν, $\mathbf{K}$, S_0 and μ_{BJS} all equal to 1 and consider the cases of parameters $\theta = 0.2, 0.5, 0.7$ in DLN scheme. The initial conditions, boundary conditions, and the source terms follow from the exact solution. We use the well-known Taylor-Hood element (P2-P1) for the fluid equation and the piecewise quadratic polynomials (P2) for the porous equation. To see the effect on the results by change of time steps, we set the diameters $h = 1/100$ for space triangulation. We apply the DLN algorithm to this test problem for 40 time steps and refer to the time step size k_n similar to that in [23]:

$$k_n = \begin{cases} 0.1 & 0 \leq n \leq 10, \\ 0.1 + 0.05\sin(10t_n) & n > 10. \end{cases} \tag{4.45}$$

The graph of the time step function (4.45) are given in Figure 19. Figure 20 shows speed

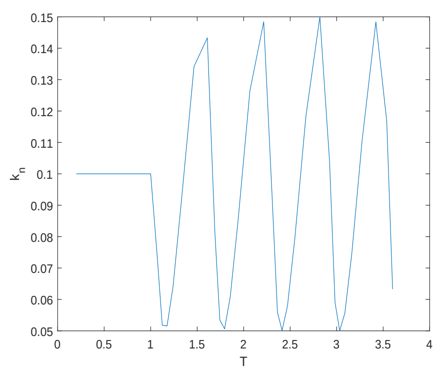
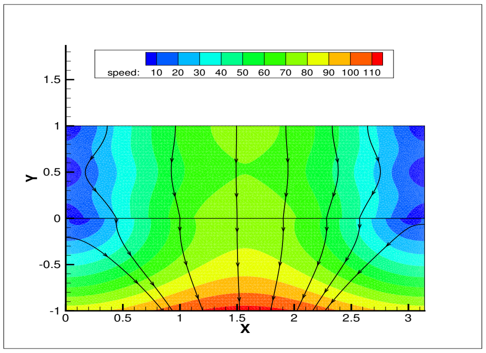


Figure 19: Change of step size k_n.

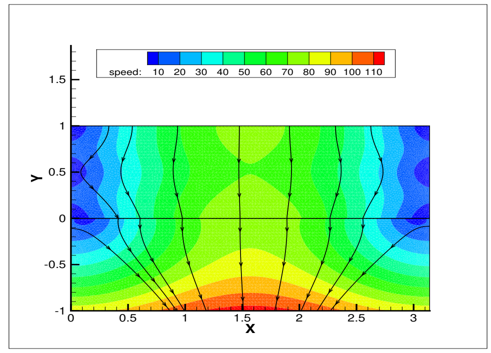
contours and velocity streamlines with parameter $\theta = 0.2, 0.5, 0.7$ respectively. From the graphs, we observe that good performance can be obtained for all three cases. Figure 21(a) and Figure 21(b) respectively show the comparison between the approximate solutions and the true solutions of the incompressible fluid velocity $\mathbf{u}_f$ and porous media fluid hydraulic head ϕ with different θ. The variable time-stepping DLN algorithm approximates exact solutions well, which confirms the stability of the DLN algorithm.

4.6.2 Test of constant Time-stepping DLN algorithm

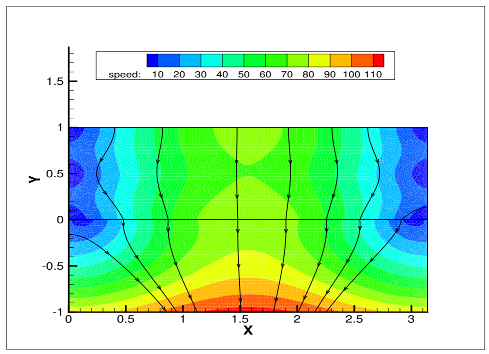
For the constant step test, we refer to the numerical example in [97]. Let the computational domain Ω be composed of $\Omega_f = (0, 1) \times (1, 2)$ and $\Omega_p = (0, 1) \times (0, 1)$ with the



(a) $\theta = 0.2$

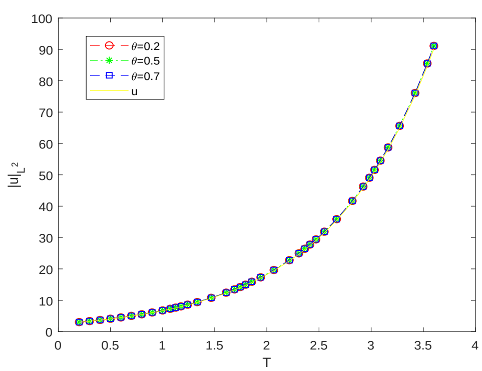


(b) $\theta = 0.5$

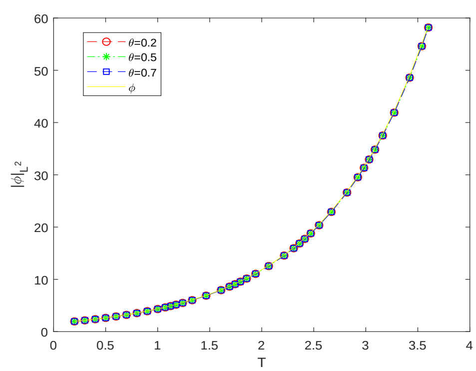


(c) $\theta = 0.7$

Figure 20: Speed contours and velocity streamlines with $\theta = 0.2, 0.5, 0.7$.



(a) Comparison for velocity $\mathbf{u}_f$



(b) Comparison for hydraulic head ϕ

Figure 21: Comparison between approximate solutions and exact solutions with different parameter θ.

interface $\Gamma = (0,1) \times \{1\}$. We set the total time $T = 1$. The exact solution is:

$$
\begin{aligned}
\mathbf{u} &= \Big[(x^2(y-1)^2 + y)\cos(t), -\tfrac{2}{3}x(y-1)^3\cos(t) + (2 - \pi\sin(\pi x))\cos(t)\Big]^{\mathrm{tr}}, \\
p &= \big(2 - \pi\sin(\pi x)\big)\sin\big(\frac{1}{2}\pi y\big)\cos(t), \\
\phi &= (2 - \pi\sin(\pi x))(1 - y - \cos(\pi y))\cos(t).
\end{aligned}
$$

For this test, MINI (P1b-P1) space and piecewise linear polynomials (P1) space are used for the approximation of the incompression fluid and the porous equation respectively. To confirm the consistency of the DLN algorithm, we set $h = \Delta t$ and calculate the errors and convergence rates for the functions $\mathbf{u}$, p, and ϕ. The rate of convergence r is calculated by

$$
r = \ln(e(\Delta t_1)/e(\Delta t_2))/\ln(\Delta t_1/\Delta t_2),
$$

where $e(\Delta t)$ is the error computed by the DLN algorithm with time stepsize Δt.

Table 10, Table 11, and Table 12 show the fluid velocity $\mathbf{u}$, hydraulic head ϕ and pressure p errors of DLN algorithm when $\theta = 0.2, 0.5, 0.7$. By comparison, the results are almost the same, but as θ increases, the errors of $\mathbf{u}$ decrease slightly, while the errors of ϕ increase. Thus how to choose the best parameters leaves an open question. Moreover Table 13, Table 14, and Table 15 show the convergence rate of velocity $\mathbf{u}$, pressure p and hydraulic head ϕ with different θ and therefore verify the second-order convergence of DLN algorithm. Finally, Table 16 shows the corresponding errors obtained by the common BDF2 method. By comparison, we can see that the DLN algorithm obtains a better hydraulic head ϕ than the BDF2 method.

4.7 Conclusions

This chapter has shown that the DLN algorithm has advantages in variable time-stepping analysis for the unsteady Stokes/Darcy model due to unconditional, long time G-stability and second-order accuracy under variable time steps. The stability of the approximate solutions comes from G-stability of the DLN algorithm and the second-order accuracy of the numerical

Table 10: The errors for DLN scheme with $\theta = 0.2$.

$\Delta t = h$	$\|\|\|\mathbf{e_u}\|\|\|_{2,0}$	$\|\|\|\mathbf{e_u}\|\|\|_{2,1}$	$\|\|\|\mathbf{e}_\phi\|\|\|_{2,0}$	$\|\|\|\mathbf{e}_\phi\|\|\|_{2,1}$	$\|\|\|e_p\|\|\|_{2,0}$
1/10	0.0163655	0.599657	0.0143625	0.552125	0.175753
1/16	0.00657067	0.354318	0.00587243	0.359717	0.0785158
1/22	0.00353871	0.255182	0.00317754	0.268333	0.0490189
1/28	0.00218857	0.191492	0.00198363	0.2117	0.0306542
1/34	0.00150194	0.160602	0.00135819	0.177254	0.0213342

Table 11: The errors for DLN scheme with $\theta = 0.5$.

$\Delta t = h$	$\|\|\|\mathbf{e_u}\|\|\|_{2,0}$	$\|\|\|\mathbf{e_u}\|\|\|_{2,1}$	$\|\|\|\mathbf{e}_\phi\|\|\|_{2,0}$	$\|\|\|\mathbf{e}_\phi\|\|\|_{2,1}$	$\|\|\|e_p\|\|\|_{2,0}$
1/10	0.01615	0.506002	0.0146238	0.551755	0.138243
1/16	0.00652393	0.311263	0.00599802	0.359655	0.0637115
1/22	0.00351853	0.22917	0.00324735	0.268314	0.04083
1/28	0.00218086	0.176397	0.00202875	0.211693	0.0260884
1/34	0.00149633	0.148517	0.0013883	0.177249	0.0184629

Table 12: The errors for DLN scheme with $\theta = 0.7$.

$\Delta t = h$	$\|\|\|\mathbf{e_u}\|\|\|_{2,0}$	$\|\|\|\mathbf{e_u}\|\|\|_{2,1}$	$\|\|\|\mathbf{e}_\phi\|\|\|_{2,0}$	$\|\|\|\mathbf{e}_\phi\|\|\|_{2,1}$	$\|\|\|e_p\|\|\|_{2,0}$
1/10	0.0161161	0.488013	0.0150263	0.551591	0.128276
1/16	0.00652022	0.30443	0.00616699	0.359622	0.0604363
1/22	0.00351759	0.225303	0.00333733	0.268301	0.0393132
1/28	0.00218125	0.174198	0.00208573	0.211687	0.0252779
1/34	0.00149674	0.14679	0.00142616	0.177246	0.0179642

Table 13: The convergence order of errors for DLN scheme with $\theta = 0.2$.

$\Delta t = h$	$r_{\mathbf{u},0}$	$r_{\mathbf{u},1}$	$r_{\phi,0}$	$r_{\phi,1}$	$r_{p,0}$
1/10	-	-	-	-	-
1/16	1.9416	1.11949	1.90286	0.911604	1.71441
1/22	1.94331	1.03066	1.92857	0.920353	1.47931
1/28	1.99249	1.19062	1.95378	0.982976	1.94657
1/34	1.93911	0.906045	1.9509	0.914669	1.86683

Table 14: The convergence order of errors for DLN scheme with $\theta = 0.5$.

$\Delta t = h$	$r_{\mathbf{u},0}$	$r_{\mathbf{u},1}$	$r_{\phi,0}$	$r_{\phi,1}$	$r_{p,0}$
1/10	-	-	-	-	-
1/16	1.92895	1.03382	1.8962	0.910541	1.64818
1/22	1.93885	0.961447	1.92678	0.920035	1.39722
1/28	1.98342	1.08527	1.95063	0.982834	1.85737
1/34	1.94019	0.886068	1.9538	0.914617	1.78066

Table 15: The convergence order of errors for DLN scheme with $\theta = 0.7$.

$\Delta t = h$	$r_{\mathbf{u},0}$	$r_{\mathbf{u},1}$	$r_{\phi,0}$	$r_{\phi,1}$	$r_{p,0}$
1/10	-	-	-	-	-
1/16	1.92532	1.00404	1.89485	0.910111	1.60126
1/22	1.93791	0.945172	1.92819	0.919886	1.35037
1/28	1.98157	1.06674	1.94911	0.982746	1.83126
1/34	1.93971	0.881715	1.95786	0.914595	1.75915

Table 16: The errors for BDF2 scheme.

$\Delta t = h$	$\|\|\|\mathbf{e_u}\|\|\|_{2,0}$	$\|\|\|\mathbf{e_u}\|\|\|_{2,1}$	$\|\|\|\mathbf{e}_\phi\|\|\|_{2,0}$	$\|\|\|\mathbf{e}_\phi\|\|\|_{2,1}$	$\|\|\|e_p\|\|\|_{2,0}$
1/10	0.0160291	0.450396	0.0165148	0.551278	0.116047
1/16	0.00650765	0.290462	0.00680715	0.359553	0.0561277
1/22	0.00351566	0.2176	0.0036845	0.268273	0.0373131
1/28	0.00218218	0.169732	0.00230674	0.211677	0.024088
1/34	0.00149872	0.143413	0.00157485	0.177236	0.0171673

simulations is derived from the combination of G-stability and consistency properties of the DLN algorithm. The complexity of the DLN method prevents its testing in flow models in which a method with DLN's excellent properties should be valued. Refactorization with time filters would be a natural way to simplify the implementation of the method and propel its common use in computational fluid dynamics. For simulations of variable time-stepping DLN algorithm, it is favorable to adapt time steps based on the criterion of minimum numerical dissipation or allowed local truncation error. To deal with the coupling problem of different regions (like the Stokes/Darcy model), we may consider decoupling the problem, applying the DLN algorithm with different time steps and corresponding adaptivity in each sub-domain.

5.0 The DLN Algorithm for the Navier-Stokes equations

5.1 Introduction

In [33] Dahlquist, Liniger, and Nevanlinna give a family of one-leg 2-step methods that are G-stable for *any* sequence of time steps. Consider the differential equation $y'(t) = g(t, y)$, with $t \in [0, T]$, where $y(t) \in X$, a Banach space, and $g : [0, T] \times X \to X'$ is a sufficiently smooth function. The family of one-leg, 2-step methods proposed by Dahlquist, Liniger, and Nevanlinna (DLN) takes the form

$$\sum_{\ell=0}^{2} \alpha_\ell y_{n-1+\ell} = \widehat{k}_n g\Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} t_{n-1+\ell}, \sum_{\ell=0}^{2} \beta_\ell^{(n)} y_{n-1+\ell} \Big), \qquad n = 1, \dots, N-1, \tag{DLN}$$

where $\widehat{k}_n$ is an weighted average of the time steps k_n, k_{n-1}, and the generating polynomials are $\rho(\zeta) = \sum_{\ell=0}^{2} \alpha_\ell \zeta^\ell$, $\sigma_n(\zeta) = \sum_{\ell=0}^{2} \beta_\ell^{(n)} \zeta^\ell$ (see (5.1) in Section 5.2).

Let Ω be the flow domain in $\mathbb{R}^d$ ($d = 2$ or 3), and denote by $u(x, t)$ the fluid velocity, $p(x, t)$ the pressure, and $f(x, t)$ body force. Herein we give an analysis of the (DLN) method for the Navier-Stokes Equations:

$$\begin{aligned} & u_t + u \cdot \nabla u - \nu \Delta u + \nabla p = f, \quad x \in \Omega, \quad 0 < t \le T, \\ & \nabla \cdot u = 0, \quad x \in \Omega \ \text{ for } \ 0 < t \le T, \quad u(x, 0) = u_0(x), \quad x \in \Omega, \\ & u = 0 \ \text{ on } \ \partial\Omega, \quad \int_\Omega p \, dx = 0 \ \text{ for } \ 0 < t \le T. \end{aligned} \tag{NSE}$$

Section 5.2 presents (DLN)'s critical property of variable step G-stability with the *G-matrix independent of the time step ratio.* Notations and preliminaries are presented in Section 5.3. Section 5.4 gives a proof of variable time step, unconditional, long time, bound of the energy for the one-leg (DLN) method for (NSE). This analysis shows that the natural kinetic energy, $\mathcal{E}(t_n)$, and numerical dissipation rate, $\mathcal{D}(t_n)$, of the DLN approximation are

$$\begin{aligned} \mathcal{E}(t_n) &= \frac{1}{4}(1+\theta)\|u_n^h\|^2 + \frac{1}{4}(1-\theta)\|u_{n-1}^h\|^2, \quad \theta = \text{method parameter}, \\ \mathcal{D}(t_n) &= \frac{1}{\widehat{k}_n} \Big\| \sum_{\ell=0}^{2} a_\ell^{(n)} u_{n-1+\ell}^h \Big\|^2, \ \text{where the coefficients } a_\ell^{(n)} \text{ are given in (5.2),} \end{aligned}$$

where $\|\cdot\|$ denotes the $L^2(\Omega)$ norm.

Section 5.5 provides the variable step error analysis. The DLN method is proven second-order for any sequence of time steps. Numerical tests are presented in Section 5.6. The first example confirms the theoretical prediction of second-order accuracy. The second test shows that DLN has stability advantages over BDF2 for variable time steps. There is a recent idea by Capuano, Sanderse, De Angelis, and Coppola [20] to adapt the time step to control the ratio of numerical to physical dissipation. Rather than testing a standard approach to error estimation and adaptivity, we also test this idea, which uses the above explicit formula for the method's numerical dissipation, in Section 5.6.

Remark 9. *We focus herein on the variable step DLN time discretization. It is impossible to draw clear conclusions when varying more than one thing. Thus, at each non-time discretization decision, we select the most classical one, e.g. standard Galerkin, well-known finite element spaces, standard nonlinearity, no stabilizations, no turbulence models, and so on. Each of these can be further optimized using the properties of DLN, developed here.*

5.1.1 Related work

The number of papers studying time-stepping methods for flow problems is very large. The general (2 parameter two-legs) linear 2-step A-stable method was analyzed for the NSE for *constant* time steps in Girault and Raviart [48], and developed further in [72]. Time adaptive discretizations of the NSE have been limited by the Dahlquist barrier, storage limitations, and the cognitive complexity of extending to the NSE many of the standard methods for systems of ordinary differential equations [12, 15, 125]. One early and important work is Kay, Gresho, Griffiths, and Silvester [75]. It presents an adaptive algorithm based on the trapezoid scheme / linearized midpoint rule (with error estimation done using an explicit AB2 type method) that is memory and computation efficient. It is well-known [80] for systems of ODEs that variable step, variable order (VSVO) methods are efficient choices, and have been considered for the NSE in [37, 63].

5.2 The Variable Step DLN method

The (DLN) method is a 1-parameter ($0 \le \theta \le 1$) family of A-stable, one-leg 2-step G-stable methods. When $\theta = 1$, (DLN) reduces to the one-step, one-leg implicit midpoint scheme [15], in which case is also symplectic, and conserves all linear and quadratic Hamiltonians. The (DLN)'s key property is that its G-matrix depends on the parameter θ, but not on the timestep ratio. For a comparison in terms of stability regions, in the constant time step case, Figure 22 shows the DLN with $\theta = 0.5$ and the BDF2 root locus curves. Let

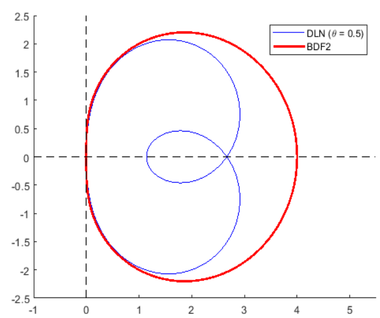


Figure 22: Boundaries of Stability Region for constant DLN ($\theta = 0.5$) and BDF2.

$\{t_n\}_{n\ge 0}$ denote the time mesh points, $k_n = t_{n+1} - t_n$ the timestep, and $\varepsilon_n = \frac{k_n - k_{n-1}}{k_n + k_{n-1}}$ the step size variability (notice $\varepsilon_n \in (-1, 1)$). Then the $\{\alpha_\ell, \beta_\ell\}_{\ell=0,1,2}$ coefficients in (DLN) are

$$\begin{pmatrix} \alpha_2 & \beta_2^{(n)} \\ \alpha_1 & \beta_1^{(n)} \\ \alpha_0 & \beta_0^{(n)} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(\theta+1) & \frac{1}{4}\Big(1 + \frac{1-\theta^2}{(1+\varepsilon_n\theta)^2} + {\varepsilon_n}^2 \frac{\theta(1-\theta^2)}{(1+\varepsilon_n\theta)^2} + \theta\Big) \\ -\theta & \frac{1}{2}\Big(1 - \frac{1-\theta^2}{(1+\varepsilon_n\theta)^2}\Big) \\ \frac{1}{2}(\theta-1) & \frac{1}{4}\Big(1 + \frac{1-\theta^2}{(1+\varepsilon_n\theta)^2} - {\varepsilon_n}^2 \frac{\theta(1-\theta^2)}{(1+\varepsilon_n\theta)^2} - \theta\Big) \end{pmatrix}, \tag{5.1}$$

and the averaged time step $\widehat{k}_n$ is

$$\widehat{k}_n = \alpha_2 k_n - \alpha_0 k_{n-1} = \frac{1}{2}(1+\theta)k_n + \frac{1}{2}(1-\theta)k_{n-1} = \theta\frac{k_n - k_{n-1}}{2} + \frac{k_n + k_{n-1}}{2}.$$

The α_ℓ-coefficients are independent of the time-step ratio, but β_ℓ depend on the time-step ratios via the variability coefficients ε_n. For expression of the numerical dissipation, we also define

$$a_1^{(n)} = -\frac{\sqrt{\theta\left(1-\theta^2\right)}}{\sqrt{2}(1+\varepsilon_n\theta)}, \quad a_2^{(n)} = -\frac{1-\varepsilon_n}{2}a_1^{(n)}, \quad a_0^{(n)} = -\frac{1+\varepsilon_n}{2}a_1^{(n)}, \tag{5.2}$$

which correspond to $\{\gamma_\ell^{(n)}\}_{\ell=0}^{2}$ in the Proposition 2 and depend on the time-step ratios through the variability coefficients ε_n.

5.3 Preliminaries and Notations

The discussion of the DLN method connects to stability theory in numerical ODEs. Much of this theory addresses the response of the method when applied to $y' = \lambda y$, $y(0) = 1$, e.g., [56]. Recall that a method is 0-stable if, when applied to $y' = 0$, the approximate solution does not grow. For more general problems, 0-stability allows exponential growth but excludes rate constants that blow up as $\Delta t \to 0$. A method is A-stable if, when applied to $y' = \lambda y$, $y(0) = 1$, the constant time step approximation $y_n \to 0$ as $n \to \infty$ for any Δt and any λ with $\mathrm{Re}(\lambda) < 0$. A-stability addresses time asymptotics ($t_n \to \infty$) and is thus stronger than $0-$ stability. A method is L-stable if it is A-stable and satisfies the additional condition that $y_n \to 0$ for n fixed for any Δt as (real) $\lambda \to -\infty$. L-stability means that the method will not experience the ± 1 type oscillation (called *ringing*) of the trapezoid rule for constant Δt and large, negative λ. G-stability is an extension of A-stability to unconditional, long time, energetic stability of nonlinear problems.

Let Ω be any domain in $\mathbb{R}^d$ ($d = 2$ or 3). For $1 \le p < \infty$ and $r \in \mathbb{N}$, $\|\cdot\|_{L^p}$ and $\|\cdot\|_{W_p^r}$ are norms on function spaces $L^p(\Omega)$ and $W_p^r(\Omega)$ respectively. When $p = 2$, we denote $\|\cdot\|$ be $L^2(\Omega)$ norm, and the $L^2(\Omega)$ inner product $(\cdot,\cdot)$. Moreover, $H^r(\Omega)$ denotes the Sobolev space $W_2^r(\Omega)$ with norm $\|\cdot\|_r$. For the velocity u and pressure p, we define the spaces

$$\begin{aligned}
X &= \Big\{v : \Omega \to \mathbb{R}^d : v \in L^2(\Omega), \nabla v \in L^2(\Omega) \text{ and } v = 0 \text{ on } \partial\Omega\Big\}, \\
Q &= \Big\{q : \Omega \to \mathbb{R} : v \in L^2(\Omega) \text{ and } \int_\Omega q\,dx = 0\Big\}.
\end{aligned}$$

The space of divergence-free functions is denoted

$$V = \Big\{ v \in X : (\nabla \cdot v, q) = 0, \ \forall q \in Q \Big\}.$$

The space X^* is the dual space of X with the dual norm

$$\|f\|_* = \sup_{\substack{v \in X \\ v \neq 0}} \frac{(f, \nabla v)}{\|\nabla v\|}, \quad \forall f \in X^*.$$

For convenience, we denote

$$\|v\|_{p,r} = \|v\|_{L^p(0,T,H^r)},$$

for any function $v(t,x)$ and $1 \leq p \leq \infty$. For $u, v, w \in X$, we define the explicitly skew symmetrized trilinear form

$$b(u,v,w) := \frac{1}{2}(u \cdot \nabla v, w) - \frac{1}{2}(u \cdot \nabla w, v).$$

This is the most common form of nonlinearity in use so it is selected herein. In passing, we note that the momentum and angular momentum conserving (EMAC) form of the nonlinearity from Olshanskii and Rebholz [102], although less common, provides better results. Obviously we have $b(u,v,v) = 0$ and by the divergence theorem, we have $b(u,v,w) = (u \cdot \nabla v, w)$ for all $u \in V$ and $v, w \in X$. Moreover, $b(u,v,w)$ satisfies the bounds (see e.g., [122])

$$b(u,v,w) \leq C(\Omega)\|\nabla u\|\|\nabla v\|\|\nabla w\|, \qquad b(u,v,w) \leq C(\Omega)\|u\|^{1/2}\|\nabla u\|^{1/2}\|\nabla v\|\|\nabla w\|. \tag{5.3}$$

For spatial discretization, we use the finite element method (FEM). The approximate solutions for the velocity and pressure are in the finite element spaces $X_h \subset X, Q_h \subset Q$, based on an edge to edge triangulation Ω (with maximum triangle diameter h). We assume that X_h and Q_h satisfy the usual discrete inf-sup condition (LBB^h condition). The Taylor-Hood elements, which satisfy the condition, are used in the numerical tests. The discretely divergence-free subspace of X_h is

$$V_h := \{v_h \in X_h : (\nabla \cdot v_h, q_h) = 0, \ \forall q_h \in Q_h\}.$$

We also assume that X_h and Q_h have degree r and s respectively ($r, s \in \mathbb{N}$), and the following interpolation error estimate for the velocity u and pressure p holds (see e.g., [9, p.108]):

$$\begin{aligned}
&\|u - I^h u\|_m \leq Ch^{r+1-m}\|u\|_{r+1}, \quad u \in H^{r+1}(\Omega)^d, \quad 0 \leq m \leq r \\
&\|p - I^h p\|_m \leq Ch^{s+1-m}\|p\|_{s+1}, \quad p \in H^{s+1}(\Omega), \quad 0 \leq m \leq s, \quad m \in \mathbb{N},
\end{aligned} \tag{5.4}$$

where $I^h u$ and $I^h p$ are the L^2 projection of u and p into V^h and Q^h respectively. (These estimates can follow, for example, from standard ones for the Stokes projection for H^2 regular domains.)

For any given sequence $\{z_n\}_{n\geq 1}$, we denote by

$$z_{n,\beta} = \beta_2^{(n)} z_{n+1} + \beta_1^{(n)} z_n + \beta_0^{(n)} z_{n-1}$$

the convex combination of the three adjacent terms in the sequence. For example, $\{t_{n,\beta}\}$ is the set of time-values and $u_{n,\beta}$ are the implicit values where the equation is evaluated

$$t_{n,\beta} = \beta_2^{(n)} t_{n+1} + \beta_1^{(n)} t_n + \beta_0^{(n)} t_{n-1}, \qquad u_{n,\beta} = \beta_2^{(n)} u_{n+1} + \beta_1^{(n)} u_n + \beta_0^{(n)} u_{n-1}.$$

The variational formulation of the one-leg (DLN) method for the NSE is as follows. Given $u_n^h, u_{n-1}^h \in X_h$ and $p_n^h, p_{n-1}^h \in Q_h$, find u_{n+1}^h and p_{n+1}^h satisfying

$$\begin{aligned}
\frac{1}{\widehat{k}_n}\Big(\alpha_2 u_{n+1}^h + \alpha_1 u_n^h + \alpha_0 u_{n-1}^h, v^h\Big) + \nu(\nabla u_{n,\beta}^h, \nabla v^h) + b(u_{n,\beta}^h, u_{n,\beta}^h, v^h)& \\
-(p_{n,\beta}^h, \nabla \cdot v^h) = (f(t_{n,\beta}), v^h),& \\
(\nabla \cdot u_{n+1}^h, q^h) = 0,&
\end{aligned} \tag{5.5}$$

for all $v^h \in X^h, q^h \in Q^h$. Under the discrete inf-sup condition, (5.5) is equivalent to

$$\begin{aligned}
\frac{1}{\widehat{k}_n}(\alpha_2 u_{n+1}^h + \alpha_1 u_n^h + \alpha_0 u_{n-1}^h, v^h) + \nu(\nabla u_{n,\beta}^h, \nabla v^h) + b(u_{n,\beta}^h, u_{n,\beta}^h, v^h)& \\
= \Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} f(t_{n-1+\ell}), v^h\Big), \quad \forall v^h \in V^h.&
\end{aligned} \tag{5.6}$$

5.4 Stability of DLN Method for the NSE

In this section, we prove the unconditional, long time, variable time step bound of the energy for (5.5), using the G-stability property in (2.7) of the method.

Theorem 10 (Unconditional, Long Time Stability)**.** *The one-leg 2-step* (DLN) *algorithm* (5.5) *satisfies unconditionally the following long-time energy bounds: for any integer $M > 1$,*

$$\frac{1}{4}(1+\theta)\|u_M^h\|^2 + \frac{1}{4}(1-\theta)\|u_{M-1}^h\|^2 + \sum_{n=1}^{M-1}\left\|\sum_{\ell=0}^{2} a_\ell^{(n)} u_{n-1+\ell}^h\right\|^2 + \frac{\nu}{2}\sum_{n=1}^{M-1}\widehat{k}_n\|\nabla u_{n,\beta}^h\|^2$$
$$\leq \frac{1}{2\nu}\sum_{n=1}^{M-1}\widehat{k}_n\|f(t_{n,\beta})\|_*^2 + \frac{1}{4}(1+\theta)\|u_1^h\|^2 + \frac{1}{4}(1-\theta)\|u_0^h\|^2,$$

where $\{a_i^{(n)}\}_{i=0,1,2}$ are defined in (5.2).

Proof. For $n \geq 1$, set $v^h = u_{n,\beta}^h$ in (5.5). Using the skew-symmetry property of b

$$\frac{1}{\widehat{k}_n}\Big(\sum_{\ell=0}^{2}\alpha_\ell u_{n-1+\ell}^h, u_{n,\beta}^h\Big) + \nu\|\nabla u_{n,\beta}^h\|^2 = (f(t_{n,\beta}), \nabla u_{n,\beta}^h).$$

Using definition of $\|\cdot\|_*$ norm and Young's inequality

$$\big(f\left(t_{n,\beta}\right), \nabla u_{n,\beta}^h\big) \leq \|f\left(t_{n,\beta}\right)\|_*\|\nabla u_{n,\beta}^h\| \leq \frac{\nu}{2}\|\nabla u_{n,\beta}^h\|^2 + \frac{1}{2\nu}\|f\left(t_{n,\beta}\right)\|_*^2 .$$

Thus for $n = 1, 2, \cdots M-1$ and the Cauchy-Schwarz inequality we obtain

$$\Big(\sum_{\ell=0}^{2}\alpha_\ell u_{n-1+\ell}^h\ , u_{n,\beta}^h\Big) + \frac{\nu}{2}\widehat{k}_n\|\nabla u_{n,\beta}^h\|^2 \leq \frac{1}{2\nu}\widehat{k}_n\|f(t_{n,\beta})\|_*^2.$$

Then the G-stability relation (2.7) implies

$$\left\|\begin{matrix} u_{n+1}^h \\ u_n^h \end{matrix}\right\|_{G(\theta)}^2 - \left\|\begin{matrix} u_n^h \\ u_{n-1}^h \end{matrix}\right\|_{G(\theta)}^2 + \left\|\sum_{\ell=0}^{2} a_\ell^{(n)} u_{n-1+\ell}^h\right\|^2 + \frac{\nu}{2}\widehat{k}_n\|\nabla u_{n,\beta}^h\|^2 \leq \frac{1}{2\nu}\widehat{k}_n\|f(t_{n,\beta})\|_*^2.$$

Finally, summation over n from 1 to $M-1$, the G-stability equality in (2.7) and (5.2) yield

$$\left\|\begin{matrix} u_M^h \\ u_{M-1}^h \end{matrix}\right\|_{G(\theta)}^2 - \left\|\begin{matrix} u_1^h \\ u_0^h \end{matrix}\right\|_{G(\theta)}^2 + \sum_{n=1}^{M-1}\left\|\sum_{\ell=0}^{2} a_\ell^{(n)} u_{n-1+\ell}^h\right\|^2 + \sum_{n=1}^{M-1}\frac{\nu}{2}\widehat{k}_n\left\|\nabla u_{n,\beta}^h\right\|^2 \leq \sum_{n=1}^{M-1}\frac{1}{2\nu}\widehat{k}_n\|f\left(t_{n,\beta}\right)\|_*^2 .$$

The above inequality results in the proof. □

The above energy equality result identifies the DLN method's kinetic energy and numerical energy dissipation rates:

$$\mathcal{E}_n = \frac{1}{4}(1+\theta)\|u_n^h\|^2 + \frac{1}{4}(1-\theta)\|u_{n-1}^h\|^2, \qquad \mathcal{D}_n = \frac{1}{\widehat{k}_n}\Big\| \sum_{\ell=0}^{2} a_\ell^{(n)} u_{n-1+\ell}^h \Big\|^2.$$

5.5 Variable Time-Step Error Analysis

In this section, we analyze the error in the approximate solutions by the one-leg DLN method for variable time steps. The discrete-time error analysis requires norms that are discrete-time analogs of the norms used in the continuous-time case. As before, let $[0,T]$ denote the whole time interval, $P = \{t_n\}_{n=0}^{M}$ be a partition on $[0,T]$ and $\{k_n\}_{n=0}^{M-1}$ be the set of time step sizes. For a function $v\,(x,t)$ and $1 \le p < \infty$, $r \in \mathbb{N}$, we define

$$\begin{gathered} |||v|||_{\infty,r} = \max_{0\le n\le M} \|v\,(t_n)\,\|_r, \quad |||v|||_{\beta,\infty,r} = \max_{1\le n\le M-1} \|v\,(t_{n,\beta})\,\|_r, \\ |||v|||_{\beta,p,r} = \Big(\sum_{n=1}^{M-1} (k_{n-1}+k_n)\|v(t_{n,\beta})\|_r^p\Big)^{1/p}, \quad |||v|||_{\beta,p,*} = \Big(\sum_{n=1}^{M-1} (k_{n-1}+k_n)\|v(t_{n,\beta})\|_*^p\Big)^{1/p}. \end{gathered} \tag{5.7}$$

In the above definitions, $|||v|||_{\beta,p,r}$ and $|||v|||_{\beta,p,*}$ are forms of Riemann sums in which the function v is evaluated at point $t_{n,\beta} \in [t_{n-1}, t_{n+1}]$.

Now we introduce the main theorem about error analysis under the following timestep condition:

$$\sum_{\ell=0}^{2} \Big(\frac{C(\theta)}{\nu^3} \big(k_{\max}^7 \|\nabla u_{tt}\|_{4,0}^4 + |||\nabla u|||_{\beta,\infty,0}^4\big) + 1\Big)\widehat{k}_{n-1+\ell} \le 1, \qquad \text{for all} \quad 2 \le n \le M-2. \tag{5.8}$$

Theorem 11. *Let* $(u(t), p(t))$ *be a sufficiently smooth, strong solution of the* (NSE)*. Under the timestep condition* (5.8)*, there exists a constant* $C > 0$ *such that the solution to the DLN algorithm* (5.5) *satisfies the following error estimates*

$$|||u - u^h|||_{\infty,0} \leq Ch^{r+1}|||u|||_{\infty,r+1} + F(h, k_{\max}), \tag{5.9}$$

and

$$\Big(\nu \sum_{n=1}^{M-1} \widehat{k}_n \big\|\nabla\big(u(t_{n,\beta}) - u_{n,\beta}^h\big)\big\|^2\Big)^{\frac{1}{2}} \leq C(\theta)\sqrt{\nu}k_{\max}^2\|\nabla u_{tt}\|_{2,0} + F(h, k_{\max}), \tag{5.10}$$

where $k_{\max} = \max\{k_n\}_{n=0}^{M-1}$ *and*

$$\begin{aligned} F(h, k_{\max}) &= C(\theta)\sqrt{\nu}h^r k_{\max}^2\|u_{tt}\|_{2,r+1} + C(\theta)\sqrt{\nu}h^r|||u|||_{\beta,2,r+1} \\ &+ C(\theta)\frac{h^{r+1/2}}{\sqrt{\nu}}\big\{k_{\max}^4(\|u_{tt}\|_{4,r+1}^2 + \|\nabla u_{tt}\|_{4,0}^2) + |||u|||_{\beta,4,r+1}^2 + |||\nabla u|||_{\beta,4,0}^2\big\} \\ &+ \frac{C(\theta)}{\sqrt{\nu}}(k_{\max}^2\|p_{tt}\|_{2,0} + h^{s+1}|||p|||_{\beta,2,s+1}) + \frac{C(\theta)}{\sqrt{\nu}}k_{\max}^4\|\nabla u_{tt}\|_{4,0}^2 \\ &+ C(\theta)\frac{h^r}{\sqrt{\nu}}\big(\frac{1}{\nu}|||f|||_{\beta,2,*} + \frac{1}{\sqrt{\nu}}\|u_1^h\| + \frac{1}{\sqrt{\nu}}\|u_0^h\| + k_{\max}^4\|u_{tt}\|_{4,r+1}^2 + |||u|||_{\beta,4,r+1}^2\big) \\ &+ C(\theta)k_{\max}^2\big\{\|u_{ttt}\|_{2,0} + \|f_{tt}\|_{2,0} + \frac{1}{\sqrt{\nu}}\|p_{tt}\|_{2,0} + \sqrt{\nu}\|\nabla u_{tt}\|_{2,0} + \frac{1}{\sqrt{\nu}}|||\nabla u|||_{\beta,4,0}^2 \\ &\qquad + \frac{1}{\sqrt{\nu}}\|\nabla u_{tt}\|_{4,0}^2\big\}. \end{aligned} \tag{5.11}$$

Remark 12. *Time step restrictions* (5.8) *like* $\Delta t \leq O(\nu^{-3})$ *arise from the discrete Gronwall inequality in the analysis of fully implicit methods. To bound the error above the discrete Gronwall inequality requires the linearized discrete problem to be positive definite. If a suitable linearly implicit (semi-implicit) method is used instead, the discrete problem automatically satisfies this and no similar time step restriction occurs, see the treatment of Crank-Nicolson in Ingram [68] for details. We comment below in Remark 13 on what this linearly implicit realization is for DLN.*

Proof. For $\theta = 1$, the one-leg 2-step (DLN) method becomes the implicit rule and the conclusions of the theorem have been proved in many places, hence we will examine the case $\theta \in [0, 1)$. The proof is relative long, thus we separate the proof in the following steps

1. Combining NSE at time $t_{n,\beta}$ and the DLN algorithm (5.6) to derive the equation of pointwise error $e_n := u(t_n) - u_n^h$ and the truncation error τ in (5.13).
2. Decomposing e_n by sum of η_n (the difference of $u(t_n)$ and its L^2 projection onto V_h) and ϕ_n^h and then transfering the error equation derived in step 1 into the equation of η_n and ϕ_n^h.
3. Giving bound for ϕ_n^h in the error equation obtained in step 2
4. Combining interpolant approximation theorem in (5.4) and conclusion in step 3 to obtain the convergence of the DLN solution in L^2-norm and H^1-norm.

Step 1. Consider (NSE) at time $t_{n,\beta}$ $(1 \leq n \leq M-1)$. For any $v^h \in V^h$, we have

$$(u_t(t_{n,\beta}), v^h) + \nu(\nabla u(t_{n,\beta}), \nabla v^h) + b(u(t_{n,\beta}), u(t_{n,\beta}), v^h) - (p(t_{n,\beta}), \nabla \cdot v^h) = (f(t_{n,\beta}), v^h),$$

and equivalently

$$\begin{aligned}
&\frac{1}{\widehat{k}_n}\Big(\sum_{\ell=0}^{2} \alpha_\ell u(t_{n-1+\ell}), v^h\Big) + b\Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), v^h\Big) - \tau(u,p,v^h) \\
&+\nu\Big(\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \nabla v^h\Big) - \Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}), \nabla \cdot v^h\Big) = \Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} f(t_{n-1+\ell}), v^h\Big),
\end{aligned} \tag{5.12}$$

where $\tau(u,p,v^h)$ is the truncation error

$$\begin{aligned}
\tau(u,p,v^h) =& \Big(\frac{1}{\widehat{k}_n}\Big(\sum_{\ell=0}^{2} \alpha_\ell u(t_{n-1+\ell})\Big) - u_t(t_{n,\beta}), v^h\Big) + \nu\Big(\nabla\Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}) - u(t_{n,\beta})\Big), \nabla v^h\Big) \\
&+ b\Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), v^h\Big) - b(u(t_{n,\beta}), u(t_{n,\beta}), v^h) \\
&- \Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - p(t_{n,\beta}), \nabla \cdot v^h\Big) + \Big(f(t_{n,\beta}) - \sum_{\ell=0}^{2} \beta_\ell^{(n)} f(t_{n-1+\ell}), v^h\Big).
\end{aligned} \tag{5.13}$$

Next by the definition of pointwise error e_n, we subtract (5.12) from the fully discrete one-leg 2-step DLN equation (5.6) to obtain

$$\begin{aligned}
&\frac{1}{\widehat{k}_n}\Big(\sum_{\ell=0}^{2} \alpha_\ell e_{n-1+\ell}, v^h\Big) + b\Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), v^h\Big) - b\big(u_{n,\beta}^h, u_{n,\beta}^h, v^h\big) \\
&+\nu(\nabla e_{n,\beta}, \nabla v^h) = \Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}), \nabla \cdot v^h\Big) + \tau(u,p,v^h), \quad \forall v^h \in V^h.
\end{aligned} \tag{5.14}$$

Step 2. As usual, let U_n be L^2 projection of $u(t_n)$ onto V^h, and we decompose e_n as

$$e_n = u(t_n) - U_n - (u_n^h - U_n) := \eta_n - \phi_n^h.$$

Setting $v^h = \phi_{n,\beta}^h \equiv \sum_{\ell=0}^{2} \beta_\ell^{(n)} \phi_{n-1+\ell}^h$, then (5.14) writes

$$\begin{aligned}
&\Big(\frac{\sum_{\ell=0}^{2} \alpha_\ell \phi_{n-1+\ell}^h}{\widehat{k}_n}, \phi_{n,\beta}^h\Big) + \nu \|\nabla \phi_{n,\beta}^h\|^2 + b(u_{n,\beta}^h, u_{n,\beta}^h, \phi_{n,\beta}^h) \\
&\quad - b\big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \phi_{n,\beta}^h\big) \\
= &\Big(\frac{\sum_{\ell=0}^{2} \alpha_\ell \eta_{n-1+\ell}}{\widehat{k}_n}, \phi_{n,\beta}^h\Big) + \nu(\nabla \eta_{n,\beta}, \nabla \phi_{n,\beta}^h) - \big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}), \nabla \cdot \phi_{n,\beta}^h\big) - \tau(u, p, v^h).
\end{aligned}$$

Using $(q^h, \nabla \cdot \phi_{n,\beta}^h) = 0$ for any $q^h \in Q^h$ and multiplying the above equation by $\widehat{k}_n$, we obtain

$$\begin{aligned}
&\big(\sum_{\ell=0}^{2} \alpha_\ell \phi_{n-1+\ell}^h \,, \phi_{n,\beta}^h\big) + \nu \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 \qquad (5.15)\\
= &\big(\sum_{\ell=0}^{2} \alpha_\ell \eta_{n-1+\ell}^h \,, \phi_{n,\beta}^h\big) + \nu \widehat{k}_n (\nabla \eta_{n,\beta}, \nabla \phi_{n,\beta}^h) + \widehat{k}_n b\big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \phi_{n,\beta}^h\big) \\
&- \widehat{k}_n b(u_{n,\beta}^h, u_{n,\beta}^h, \phi_{n,\beta}^h) - \widehat{k}_n \big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - q^h, \nabla \cdot \phi_{n,\beta}^h\big) - \widehat{k}_n \tau(u, p, \phi_{n,\beta}^h), \quad \forall q^h \in Q^h.
\end{aligned}$$

Now we analyze the terms on the right-hand side of (5.15). By the property of projection operators and the linearity of inner products, we have

$$\big(\sum_{\ell=0}^{2} \alpha_\ell \eta_{n-1+\ell}, \phi_{n,\beta}^h\big) = 0.$$

Using the skew-symmetry of the trilinear form b, using (5.3), the Cauchy-Schwarz and Young inequalities, and the G-stability relation (2.7), we obtain in a typical manner

$$\begin{aligned}
&\widehat{k}_n b\big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \phi_{n,\beta}^h\big) - \widehat{k}_n b(u_{n,\beta}^h, u_{n,\beta}^h, \phi_{n,\beta}^h) \\
= &\widehat{k}_n b\Big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} \big(u(t_{n-1+\ell}) - u_{n-1+\ell}^h\big), \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \phi_{n,\beta}^h\Big) \\
&+ \widehat{k}_n b\Big(u_{n,\beta}^h, \sum_{\ell=0}^{2} \beta_\ell^{(n)} \big(u(t_{n-1+\ell}) - u_{n-1+\ell}^h\big), \phi_{n,\beta}^h\Big) \\
= &\widehat{k}_n b\big(\eta_{n,\beta}, \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \phi_{n,\beta}^h\big) - \widehat{k}_n b\big(\phi_{n,\beta}^h, \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \phi_{n,\beta}^h\big) + \widehat{k}_n b(u_{n,\beta}^h, \eta_{n,\beta}, \phi_{n,\beta}^h).
\end{aligned}$$

For any $\varepsilon > 0$, using (5.3),

$$\begin{aligned}
\widehat{k}_n b\big(\eta_{n,\beta}, \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \phi_{n,\beta}^h\big) &\le C(\Omega)\widehat{k}_n \|\eta_{n,\beta}\|^{\frac{1}{2}} \|\nabla \eta_{n,\beta}\|^{\frac{1}{2}} \big\|\nabla\big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big)\big\| \|\nabla \phi_{n,\beta}^h\| \\
&\le \varepsilon \nu \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 \\
&\quad + C(\varepsilon,\Omega)\widehat{k}_n \nu^{-1} \|\eta_{n,\beta}\| \|\nabla \eta_{n,\beta}\| \big\|\nabla\big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big)\big\|^2, \\
\widehat{k}_n b\big(\phi_{n,\beta}^h, \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \phi_{n,\beta}^h\big) &\le C(\Omega)\widehat{k}_n \|\phi_{n,\beta}^h\|^{\frac{1}{2}} \|\nabla \phi_{n,\beta}^h\|^{\frac{1}{2}} \big\|\nabla\big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big)\big\| \|\nabla \phi_{n,\beta}^h\| \\
&\le \varepsilon \nu \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 \\
&\quad + C(\varepsilon,\Omega)\widehat{k}_n \nu^{-3} \|\phi_{n,\beta}^h\|^2 \big\|\nabla\big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big)\big\|^4, \\
\widehat{k}_n b(u_{n,\beta}^h, \eta_{n,\beta}, \phi_{n,\beta}^h) &\le C(\Omega)\widehat{k}_n \|u_{n,\beta}^h\|^{\frac{1}{2}} \|\nabla u_{n,\beta}^h\|^{\frac{1}{2}} \|\nabla \eta_{n,\beta}\| \|\nabla \phi_{n,\beta}^h\| \\
&\le \varepsilon \nu \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 + C(\varepsilon,\Omega)\widehat{k}_n \nu^{-1} \|u_{n,\beta}^h\| \|\nabla u_{n,\beta}^h\| \|\nabla \eta_{n,\beta}\|^2.
\end{aligned}$$

Now using the Cauchy-Schwarz and Young inequalities gives

$$\begin{aligned}
\nu \widehat{k}_n (\nabla \eta_{n,\beta}, \nabla \phi_{n,\beta}^h) &\le \nu \widehat{k}_n \|\nabla \eta_{n,\beta}\| \|\nabla \phi_{n,\beta}^h\| \\
&\le \varepsilon \nu \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 + C(\varepsilon)\nu \widehat{k}_n \|\nabla \eta_{n,\beta}\|^2, \\
\widehat{k}_n \big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - q^h, \nabla \cdot \phi_{n,\beta}^h\big) &\le \widehat{k}_n \big\|\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - q^h\big\| \|\nabla \cdot \phi_{n,\beta}^h\| \\
&\le \varepsilon \nu \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 + C(\varepsilon)\widehat{k}_n \nu^{-1} \big\|\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - q^h\big\|^2.
\end{aligned}$$

Now we set $\varepsilon = 1/16$, combine the analysis above and apply the G-stability relation (2.7) to (5.15). This becomes

$$\begin{aligned}
&\left\| \begin{matrix} \phi_{n+1}^h \\ \phi_n^h \end{matrix} \right\|_{G(\theta)}^2 - \left\| \begin{matrix} \phi_n^h \\ \phi_{n-1}^h \end{matrix} \right\|_{G(\theta)}^2 + \frac{11\nu}{16}\widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 + \big\|\sum_{\ell=0}^{2} a_\ell^{(n)} \phi_{n-1+\ell}^h\big\|^2 \\
&\le C\frac{\widehat{k}_n}{\nu^3} \|\phi_{n,\beta}^h\|^2 \big\|\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big\|^4 + C\frac{\widehat{k}_n}{\nu} \|\eta_{n,\beta}\| \|\nabla \eta_{n,\beta}\| \big\|\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big\|^2 \\
&\quad + C\nu \widehat{k}_n \|\nabla \eta_{n,\beta}\|^2 + C\frac{\widehat{k}_n}{\nu} \|u_{n,\beta}^h\| \|\nabla u_{n,\beta}^h\| \|\nabla \eta_{n,\beta}\|^2 + C\frac{\widehat{k}_n}{\nu} \big\|\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - q^h\big\|^2 \\
&\quad + \widehat{k}_n |\tau(u, p, \phi_{n,\beta}^h)|.
\end{aligned}$$

Summing up from $n=1$ to $n=M-1$, we get

$$\left\| \begin{matrix} \phi_M^h \\ \phi_{M-1}^h \end{matrix} \right\|_{G(\theta)}^2 - \left\| \begin{matrix} \phi_1^h \\ \phi_0^h \end{matrix} \right\|_{G(\theta)}^2 + \sum_{n=1}^{M-1} \Big\| \sum_{\ell=0}^{2} a_\ell^{(n)} \phi_{n-1+\ell}^h \Big\|^2 + \frac{\nu}{2} \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 \tag{5.16}$$
$$\leq \sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu} \|\eta_{n,\beta}\| \|\nabla \eta_{n,\beta}\| \Big\|\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\Big\|^2 + \sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu} \|u_{n,\beta}^h\| \|\nabla u_{n,\beta}^h\| \|\nabla \eta_{n,\beta}\|^2$$
$$+ \sum_{n=1}^{M-1} C\nu \widehat{k}_n \|\nabla \eta_{n,\beta}\|^2 + \sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu^3} \|\phi_{n,\beta}^h\|^2 \Big\|\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\Big\|^4$$
$$+ \sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu} \| \sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - q^h\|^2 + \sum_{n=1}^{M-1} \widehat{k}_n |\tau(u, p, \phi_{n,\beta}^h)|.$$

We set the approximate solution of u at two initial time-steps t_0 and t_1 to be L^2 projection of u into V^h:

$$\phi_i^h = u_i^h - U_i = 0, \; i = 0, 1.$$

Using the definition of the G-norm (2.8), the estimate (5.16) becomes

$$\frac{1}{4}(1+\theta)\|\phi_M^h\|^2 + \frac{1}{4}(1-\theta)\|\phi_{M-1}^h\|^2 + \sum_{n=1}^{M-1} \Big\| \sum_{\ell=0}^{2} a_\ell^{(n)} \phi_{n-1+\ell}^h \Big\|^2 + \frac{\nu}{2} \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}\|^2 \tag{5.17}$$
$$\leq \sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu} \|\eta_{n,\beta}\| \|\nabla \eta_{n,\beta}\| \Big\|\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\Big\|^2 + \sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu} \|u_{n,\beta}^h\| \|\nabla u_{n,\beta}^h\| \|\nabla \eta_{n,\beta}\|^2$$
$$+ \sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu^3} \|\phi_{n,\beta}^h\|^2 \Big\|\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\Big\|^4 + \sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu} \| \sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - q^h\|^2$$
$$+ \sum_{n=1}^{M-1} C\nu \widehat{k}_n \|\nabla \eta_{n,\beta}\|^2 + \sum_{n=1}^{M-1} \widehat{k}_n |\tau(u, p, \phi_{n,\beta}^h)|.$$

Step 3. We now bound each term in the right-hand side.

1. For the first term, we use the linearity of the L^2 projection, the interpolation error estimates (5.4) and Young's inequality to obtain

$$\sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu} \|\eta_{n,\beta}\| \|\nabla \eta_{n,\beta}\| \Big\|\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\Big\|^2 \tag{5.18}$$
$$\leq C(\theta) \frac{h^{2r+1}}{\nu} \sum_{n=1}^{M-1} (k_{n-1} + k_n) \Big\| \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}) \Big\|_{r+1}^2 \Big\|\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\Big\|^2$$
$$\leq C(\theta) \frac{h^{2r+1}}{\nu} \sum_{n=1}^{M-1} (k_n + k_{n-1}) \Big(\Big\| \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}) \Big\|_{r+1}^4 + \Big\|\nabla \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\Big\|^4 \Big).$$

By the triangle inequality

$$\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\Big\|_{r+1}^4 \le C\Big(\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\Big\|_{r+1}^4+\|u(t_{n,\beta})\|_{r+1}^4\Big).$$

Then by Lemma 6 and Hölder's inequality

$$\begin{aligned}\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\Big\|_{r+1}^4 \le& C(\theta)(k_n+k_{n-1})^6\Big(\int_{t_{n-1}}^{t_{n+1}}1\cdot\|u_{tt}\|_{r+1}^2dt\Big)^2 \\ \le& C(\theta)(k_n+k_{n-1})^7\int_{t_{n-1}}^{t_{n+1}}\|u_{tt}\|_{r+1}^4dt.\end{aligned} \tag{5.19}$$

Thus by the definition of the discrete norm (5.7)

$$\sum_{n=1}^{M-1}(k_n+k_{n-1})\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\Big\|_{r+1}^4 \le C(\theta)k_{\max}^8\|u_{tt}\|_{4,r+1}^4+C\||u|\|t_{\beta,4,r+1}^4. \tag{5.20}$$

Similarly

$$\sum_{n=1}^{M-1}(k_n+k_{n-1})\Big\|\nabla\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\Big\|^4 \le C(\theta)k_{\max}^8\|\nabla u_{tt}\|_{4,0}^4+C\||\nabla u|\|_{\beta,4,0}^4. \tag{5.21}$$

Combining (5.18), (5.20) and (5.21), we obtain

$$\begin{aligned}&\sum_{n=1}^{M-1}C\frac{\widehat{k}_n}{\nu}\|\eta_{n,\beta}\|\|\nabla\eta_{n,\beta}\|\Big\|\nabla\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\Big\|^2 \\ &\le C(\theta)\frac{h^{2r+1}}{\nu}\Big\{k_{\max}^8\big(\|u_{tt}\|_{4,r+1}^4+\|\nabla u_{tt}\|_{4,0}^4\big)+\||u|\|_{\beta,4,r+1}^4+\||\nabla u|\|_{\beta,4,0}^4\Big\}.\end{aligned} \tag{5.22}$$

2. For the second term we use the linearity of the L^2 projection, the interpolation error estimates (5.4), to yield

$$\sum_{n=1}^{M-1}\widehat{k}_n\|\nabla\eta_{n,\beta}\|^2 \le Ch^{2r}\sum_{n=1}^{M-1}\widehat{k}_n\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\Big\|_{r+1}^2$$
$$\le C(\theta)\,h^{2r}\sum_{n=1}^{M-1}(k_{n-1}+k_n)\left(\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\Big\|_{r+1}^2+\|u(t_{n,\beta})\|_{r+1}^2\right).$$

By Lemma 6, and the definition of the discrete norm (5.7), we have

$$C\nu\sum_{n=1}^{M-1}\widehat{k}_n\|\nabla\eta_{n,\beta}\|^2 \le C(\theta)\,\nu h^{2r}\sum_{n=1}^{M-1}(k_{n-1}+k_n)^4\int_{t_{n-1}}^{t_{n+1}}\|u_{tt}\|_{r+1}^2\,dt$$
$$+C(\theta)\,\nu h^{2r}\sum_{n=1}^{M-1}(k_{n-1}+k_n)\,\|u(t_{n,\beta})\|_{r+1}^2$$
$$\le C(\theta)\,\nu h^{2r}k_{\max}^4\,\|u_{tt}\|_{2,r+1}^2+C(\theta)\,\nu h^{2r}\,|||u|||_{\beta,2,r+1}^2\,.$$

$$C\nu\sum_{n=1}^{M-1}\widehat{k}_n\|\nabla\eta_{n,\beta}\|^2 \le \nu Ch^{2r}\sum_{n=1}^{M-1}\widehat{k}_n\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\Big\|_{r+1}^2 \tag{5.23}$$
$$\le \nu C(\theta)h^{2r}\sum_{n=1}^{M-1}(k_{n-1}+k_n)\left(\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\Big\|_{r+1}^2+\|u(t_{n,\beta})\|_{r+1}^2\right).$$
$$\le C(\theta)\nu h^{2r}\sum_{n=1}^{M-1}(k_{n-1}+k_n)^4\int_{t_{n-1}}^{t_{n+1}}\|u_{tt}\|_{r+1}^2dt+C(\theta)\nu h^{2r}\sum_{n=1}^{M-1}(k_{n-1}+k_n)\|u(t_{n,\beta})\|_{r+1}^2$$
$$\le C(\theta)\nu h^{2r}k_{\max}^4\|u_{tt}\|_{2,r+1}^2+C(\theta)\nu h^{2r}|||u|||_{\beta,2,r+1}^2.$$

3. Using the a priori bounds from Theorem 10, we have

$$\sum_{n=1}^{M-1}\frac{\widehat{k}_n}{\nu}\|u_{n,\beta}^h\|\|\nabla u_{n,\beta}^h\|\|\nabla\eta_{n,\beta}\|^2 \le \frac{Ch^{2r}}{\nu}\sum_{n=1}^{M-1}\widehat{k}_n\|\nabla u_{n,\beta}^h\|\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\Big\|_{r+1}^2$$
$$\le \frac{Ch^{2r}}{\nu}\sum_{n=1}^{M-1}\widehat{k}_n\left(\left\|\nabla u_{n,\beta}^h\right\|^2+\Big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\Big\|_{r+1}^4\right).$$

Also by Theorem 10

$$\sum_{n=1}^{M-1}\widehat{k}_n\left\|\nabla u_{n,\beta}^h\right\|^2 \le \frac{C(\theta)}{\nu^2}\,|||f|||_{\beta,2,*}^2+\frac{1}{\nu}\left\|u_1^h\right\|^2+\frac{1}{\nu}\left\|u_0^h\right\|^2. \tag{5.24}$$

By (5.20), (5.26) and (5.24),

$$\sum_{n=1}^{M-1} C\frac{\widehat{k}_n}{\nu}\|u_{n,\beta}^h\|\|\nabla u_{n,\beta}^h\|\|\nabla\eta_{n,\beta}\|^2 \tag{5.25}$$
$$\leq C(\theta)\frac{h^{2r}}{\nu}\big(\frac{C(\theta)}{\nu^2}|||f|||_{\beta,2,*}^2+\frac{1}{\nu}\|u_1^h\|^2+\frac{1}{\nu}\|u_0^h\|^2+C(\theta)k_{\max}^8\|u_{tt}\|_{4,r+1}^4+C|||u|||_{\beta,4,r+1}^4\big).$$

$$\sum_{n=1}^{M-1}\frac{\widehat{k}_n}{\nu}\|u_{n,\beta}^h\|\|\nabla u_{n,\beta}^h\|\|\nabla\eta_{n,\beta}\|^2 \tag{5.26}$$
$$\leq\frac{Ch^{2r}}{\nu}\sum_{n=1}^{M-1}\widehat{k}_n\big\|\nabla u_{n,\beta}^h\big\|\big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\big\|_{r+1}^2$$
$$\leq\frac{Ch^{2r}}{\nu}\sum_{n=1}^{M-1}\widehat{k}_n\Big(\big\|\nabla u_{n,\beta}^h\big\|^2+\big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\big\|_{r+1}^4\Big)$$
$$\leq C(\theta)\frac{h^{2r}}{\nu}\big(\frac{C(\theta)}{\nu^2}|||f|||_{\beta,2,*}^2+\frac{1}{\nu}\|u_1^h\|^2+\frac{1}{\nu}\|u_0^h\|^2+C(\theta)k_{\max}^8\|u_{tt}\|_{4,r+1}^4+C|||u|||_{\beta,4,r+1}^4\big).$$

4. Recall that inequality derived for the first term,

$$\big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\big\|_{r+1}^4\leq C\big(\big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\big\|_{r+1}^4+\|u(t_{n,\beta})\|_{r+1}^4\big),$$

hence similarly

$$\big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}\nabla u(t_{n-1+\ell})\big\|^4\leq C\big(\big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}\nabla u(t_{n-1+\ell})-\nabla u(t_{n,\beta})\big\|^4+\|\nabla u(t_{n,\beta})\|^4\big),$$

which like for (5.19) gives

$$\big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}\nabla u(t_{n-1+\ell})\big\|_{r+1}^4\leq\big\|\sum_{\ell=0}^{2}\beta_\ell^{(n)}\nabla u(t_{n-1+\ell})-\nabla u(t_{n,\beta})\big\|_{r+1}^4+\|\nabla u(t_{n,\beta})\|_{r+1}^4$$
$$\leq C(\theta)(k_n+k_{n-1})^7\int_{t_{n-1}}^{t_{n+1}}\|\nabla u_{tt}\|_{r+1}^4\,dt+\|\nabla u(t_{n,\beta})\|_{r+1}^4.$$

We bound the fourth term in a similar way to the derivation of (5.22)

$$\sum_{n=1}^{M-1}C\frac{\widehat{k}_n}{\nu^3}\|\phi_{n,\beta}^h\|^2\big\|\nabla\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\big\|^4 \tag{5.27}$$
$$\leq\sum_{n=1}^{M-1}C\frac{\widehat{k}_n}{\nu^3}\|\phi_{n,\beta}^h\|^2\Big((k_n+k_{n-1})^7\int_{t_{n-1}}^{t_{n+1}}\|\nabla u_{tt}\|^4dt+\|\nabla u(t_{n,\beta})\|^4\Big)$$
$$\leq\sum_{n=1}^{M-1}C(\theta)\frac{\widehat{k}_n}{\nu^3}\|\phi_{n,\beta}^h\|^2\big(k_{\max}^7\|\nabla u_{tt}\|_{4,0}^4+|||\nabla u|||_{\beta,\infty,0}^4\big).$$

5. Using the triangle inequality again

$$\| \sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - q^h \|^2 \le C\Big(\| \sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - p\left(t_{n,\beta}\right) \|^2 + \| p\left(t_{n,\beta}\right) - q^h \|^2 \Big).$$

Using the interpolation error estimate in (5.4) and the consistency errors Lemma 6 for pressure p, we have

$$\sum_{n=1}^{M-1} C \frac{\widehat{k}_n}{\nu} \| \sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - q^h \|^2 \le \frac{C(\theta)}{\nu} \big(k_{\max}^4 \|p_{tt}\|_{2,0}^2 + h^{2s+2} \| |p| \|_{\beta,2,s+1}^2 \big). \tag{5.28}$$

6. Let us now treat the truncation error $|\tau(u, p, \phi_{n,\beta}^h)|$. Using the Cauchy-Schwarz inequality, we have

$$\Big(\frac{1}{\widehat{k}_n} \sum_{\ell=0}^{2} \alpha_\ell u(t_{n-1+\ell}) - u_t\left(t_{n,\beta}\right), \phi_{n,\beta}^h \Big) \le \frac{1}{2} \|\phi_{n,\beta}^h\|^2 + \frac{1}{2} \Big\| \frac{1}{\widehat{k}_n} \sum_{\ell=0}^{2} \alpha_\ell u(t_{n-1+\ell}) - u_t(t_{n,\beta}) \Big\|^2,$$

and applying again Lemma 6, for $\theta \in [0, 1)$ to the last term above

$$\sum_{n=1}^{M-1} \widehat{k}_n \Big\| \frac{\sum_{\ell=0}^{2} \alpha_\ell u(t_{n-1+\ell})}{\widehat{k}_n} - u_t(t_{n,\beta}) \Big\|^2 \le C(\theta) k_{\max}^4 \|u_{ttt}\|_{2,0}^2.$$

Thus we have

$$\sum_{n=1}^{M-1} \widehat{k}_n \Big(\frac{\sum_{\ell=0}^{2} \alpha_\ell u(t_{n-1+\ell})}{\widehat{k}_n} - u_t(t_{n,\beta}), \phi_{n,\beta}^h \Big) \le \frac{1}{2} \sum_{n=1}^{M-1} \widehat{k}_n \|\phi_{n,\beta}^h\|^2 + C(\theta) k_{\max}^4 \|u_{ttt}\|_{2,0}^2.$$

Similarly,

$$\sum_{n=1}^{M-1} \widehat{k}_n \big(f(t_{n,\beta}) - \sum_{\ell=0}^{2} \beta_\ell^{(n)} f(t_{n-1+\ell}), \phi_{n,\beta}^h \big) \le \frac{1}{2} \sum_{n=1}^{M-1} \widehat{k}_n \|\phi_{n,\beta}^h\|^2 + C(\theta) k_{\max}^4 \|f_{tt}\|_{2,0}^2,$$

and also

$$\sum_{n=1}^{M-1} \widehat{k}_n \big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} p(t_{n-1+\ell}) - p(t_{n,\beta}), \nabla \cdot \phi_{n,\beta}^h \big) \le \varepsilon\nu \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 + \frac{C(\varepsilon, \theta)}{\nu} k_{\max}^4 \|p_{tt}\|_{2,0}^2,$$

$$\begin{aligned} \sum_{n=1}^{M-1} \nu \widehat{k}_n \big(\nabla \big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}) - u(t_{n,\beta}) \big), \nabla \phi_{n,\beta}^h \big) \le{}& \varepsilon\nu \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 \\ &+ C(\varepsilon, \theta) \nu k_{\max}^4 \|\nabla u_{tt}\|_{2,0}^2. \end{aligned}$$

Moreover

$$
\begin{aligned}
&b\big(\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell}),\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell}),\phi_{n,\beta}^h\big)-b\big(u\left(t_{n,\beta}\right),u\left(t_{n,\beta}\right),\phi_{n,\beta}^h\big)\\
&=b\big(\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta}),\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell}),\phi_{n,\beta}^h\big)\\
&\qquad\qquad\qquad\qquad+b\big(u(t_{n,\beta}),\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta}),\phi_{n,\beta}^h\big)\\
&\le C\big\|\nabla\big(\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\big)\big\|\|\nabla\phi_{n,\beta}^h\|\big(\|\nabla\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\|+\|\nabla u(t_{n,\beta})\|\big)\\
&\le\varepsilon\nu\|\nabla\phi_{n,\beta}^h\|^2+\frac{C(\varepsilon)}{\nu}\big\|\nabla\big(\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\big)\big\|^2\big(\|\nabla\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\|^2\\
&\qquad\qquad\qquad\qquad+\|\nabla u(t_{n,\beta})\|^2\big).
\end{aligned}
$$

Then by triangle inequality again

$$
\begin{aligned}
&\big\|\nabla\big(\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\big)\big\|^2\big(\|\nabla\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})\|^2+\|\nabla u(t_{n,\beta})\|^2\big)\\
&\le C\big\|\nabla\big(\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\big)\big\|^4\\
&\qquad\qquad+C\big\|\nabla\big(\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\big)\big\|^2\|\nabla u(t_{n,\beta})\|^2.
\end{aligned}
$$

Similar to (5.19) we have that

$$
\big\|\nabla\big(\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\big)\big\|^4\le C(\theta)(k_n+k_{n-1})^7\int_{t_{n-1}}^{t_{n+1}}\|\nabla u_{tt}\|^4dt,
$$

and also using Lemma 6 and Young's inequality we obtain

$$
\begin{aligned}
&\big\|\nabla\big(\sum_{\ell=0}^{2}\beta_\ell^{(n)}u(t_{n-1+\ell})-u(t_{n,\beta})\big)\big\|^2\|\nabla u(t_{n,\beta})\|^2\\
&\le C(\theta)(k_n+k_{n-1})^3\int_{t_{n-1}}^{t_{n+1}}\|\nabla u(t_{n,\beta})\|^2\|\nabla u_{tt}\|^2dt\\
&\le C(\theta)(k_n+k_{n-1})^3\int_{t_{n-1}}^{t_{n+1}}\big(\|\nabla u(t_{n,\beta})\|^4+\|\nabla u_{tt}\|^4\big)dt\\
&\le C(\theta)(k_n+k_{n-1})^4\|\nabla u(t_{n,\beta})\|^4+C(\theta)(k_n+k_{n-1})^3\int_{t_{n-1}}^{t_{n+1}}\|\nabla u_{tt}\|^4dt.
\end{aligned}
$$

Thus

$$\sum_{n=1}^{M-1} \widehat{k}_n \Big(b\big(\sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}), \phi_{n,\beta}^h \big) - b\big(u(t_{n,\beta}), u(t_{n,\beta}), \phi_{n,\beta}^h \big) \Big)$$
$$\leq \varepsilon\nu \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 + \frac{C(\varepsilon,\theta)}{\nu} k_{\max}^8 \|\nabla u_{tt}\|_{4,0}^4 + \frac{C(\varepsilon,\theta)}{\nu} k_{\max}^4 \big(|||\nabla u|||_{\beta,4,0}^4 + \|\nabla u_{tt}\|_{4,0}^4 \big).$$

Setting $\varepsilon = 1/16$, we obtain the following estimate for the truncation error term

$$\sum_{n=1}^{M-1} \widehat{k}_n |\tau(u,p,\phi_{n,\beta}^h)| \leq \sum_{n=1}^{M-1} \widehat{k}_n \|\phi_{n,\beta}^h\|^2 + \frac{3\nu}{16} \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 + \frac{C(\theta)}{\nu} k_{\max}^8 \|\nabla u_{tt}\|_{4,0}^4$$
$$+ C(\theta) k_{\max}^4 \big\{ \|u_{ttt}\|_{2,0}^2 + \|f_{tt}\|_{2,0}^2 + \frac{1}{\nu} \|p_{tt}\|_{2,0}^2 + \nu \|\nabla u_{tt}\|_{2,0}^2 + \frac{1}{\nu} |||\nabla u|||_{\beta,4,0}^4 + \frac{1}{\nu} \|\nabla u_{tt}\|_{4,0}^4 \big\}. \tag{5.29}$$

Now collecting the terms from (5.22), (5.23), (5.26), (5.27), (5.28) and (5.29), the inequality (5.17) becomes

$$\frac{1}{4} \|\phi_M^h\|^2 + \frac{\nu}{4} \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 \leq \sum_{n=1}^{M-1} \big(\frac{C}{\nu^3} (k_{\max}^7 \|\nabla u_{tt}\|_{4,0}^4 + |||\nabla u|||_{\beta,\infty,0}^4) + 1 \big) \widehat{k}_n \|\phi_{n,\beta}^h\|^2$$
$$+ \widetilde{F}(h, k_{\max}), \tag{5.30}$$

where

$$\begin{aligned} \widetilde{F}(h,k_{\max}) = {} & C(\theta)\nu h^{2r} k_{\max}^4 \|u_{tt}\|_{2,r+1}^2 + C(\theta)\nu h^{2r} |||u|||_{\beta,2,r+1}^2 + \frac{C(\theta)}{\nu} k_{\max}^8 \|\nabla u_{tt}\|_{4,0}^4 \\ & + C(\theta) \frac{h^{2r+1}}{\nu} \Big\{ k_{\max}^8 \big(\|u_{tt}\|_{4,r+1}^4 + \|\nabla u_{tt}\|_{4,0}^4 \big) + |||u|||_{\beta,4,r+1}^4 + |||\nabla u|||_{\beta,4,0}^4 \Big\} \\ & + \frac{C(\theta)}{\nu} \big(k_{\max}^4 \|p_{tt}\|_{2,0}^2 + h^{2s+2} |||p|||_{\beta,2,s+1}^2 \big) \\ & + C(\theta) \frac{h^{2r}}{\nu} \big(\frac{1}{\nu^2} |||f|||_{\beta,2,*}^2 + \frac{1}{\nu} \|u_1^h\|^2 + \frac{1}{\nu} \|u_0^h\|^2 + k_{\max}^8 \|u_{tt}\|_{4,r+1}^4 + |||u|||_{\beta,4,r+1}^4 \big) \\ & + C(\theta) k_{\max}^4 \big\{ \|u_{ttt}\|_{2,0}^2 + \|f_{tt}\|_{2,0}^2 + \frac{1}{\nu} \|p_{tt}\|_{2,0}^2 + \nu \|\nabla u_{tt}\|_{2,0}^2 + \frac{1}{\nu} |||\nabla u|||_{\beta,4,0}^4 \\ & \qquad + \frac{1}{\nu} \|\nabla u_{tt}\|_{4,0}^4 \big\}. \end{aligned}$$

For convenience, we define the sequence $\{D_n\}_{n=1}^{M-1}$

$$D_n := \big(\frac{C(\theta)}{\nu^3} (k_{\max}^7 \|\nabla u_{tt}\|_{4,0}^4 + |||\nabla u|||_{\beta,\infty,0}^4) + 1 \big) \widehat{k}_n, \quad 1 \leq n \leq M-1,$$

and the sequence $\{d_n\}_{n=0}^{M}$ to be

$$d_n = \begin{cases} D_1 & \text{if } n = 0 \\ D_1 + D_2 & \text{if } n = 1 \\ \sum_{\ell=0}^{2} D_{n-1+\ell} & \text{if } 2 \le n \le M-2 \\ D_{M-2} + D_{M-1} & \text{if } n = M-1 \\ D_{M-1} & \text{if } n = M \end{cases}$$

We use triangle inequality in (5.30) to obtain

$$\|\phi_M^h\|^2 + \nu \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}^h\|^2 \le C(\theta) \sum_{n=0}^{M} d_n \|\phi_n^h\|^2 + \widetilde{F}(h, k_{\max}).$$

Step 4. Under the timestep condition (5.8), using the discrete Gronwall inequality (see e.g., [65]), the inequality (5.30) yields

$$\|\phi_M^h\|^2 + \nu \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}\|^2 \le \exp\big(\sum_{n=1}^{M-1} C(\theta) \frac{d_n}{1-d_n}\big) \widetilde{F}(h, k_{\max}). \tag{5.31}$$

From (5.31), we have

$$\|\phi_M^h\| \le C(\theta) \sqrt{\widetilde{F}(h, k_{\max})}. \tag{5.32}$$

Finally, combining (5.4) and (5.31), we obtain (5.9)

$$\begin{aligned} &|||u - u^h|||_{\infty,0} := \max_{0 \le n \le M} \|u(t_n) - u_n^h\| \le \max_{0 \le n \le M} \|\eta_n\| + \max_{0 \le n \le M} \|\phi_n^h\| \\ &\le \max_{0 \le n \le M} C h^{r+1} \|u_n\|_{r+1} + C(\theta) \sqrt{\widetilde{F}(h, k_{\max})} \le C h^{r+1} |||u|||_{\infty, r+1} + F(h, k_{\max}), \end{aligned}$$

concluding the proof of the first part of the theorem. For the second part, in order to prove (5.10), we begin by noticing that

$$\begin{aligned} \sum_{n=1}^{M-1} \widehat{k}_n \big\|\nabla\big(u(t_{n,\beta}) - u_{n,\beta}^h\big)\big\|^2 \le & \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla\big(u(t_{n,\beta}) - \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big)\|^2 \\ & + \sum_{n=1}^{M-1} \widehat{k}_n \big\|\nabla\big(u_{n,\beta}^h - \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big)\big\|^2. \end{aligned}$$

We then apply Lemma 6 to the first term in the right-hand side

$$\nu \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla\big(u(t_{n,\beta}) - \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big)\|^2 \le C(\theta)\nu k_{\max}^4 \|\nabla u_{tt}\|_{2,0}^2 \ ,$$

and use the triangle inequality,

$$\nu \sum_{n=1}^{M-1} \widehat{k}_n \big\|\nabla\big(u_{n,\beta}^h - \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell})\big)\big\|^2 \le C\nu \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \eta_{n,\beta}\|^2 + C\nu \sum_{n=1}^{M-1} \widehat{k}_n \|\nabla \phi_{n,\beta}\|^2 \ .$$

The last term inhere can be bounded by (5.31), while for the first term we use (5.23). Thus

$$\nu \sum_{n=1}^{M-1} \widehat{k}_n \big\|\nabla(u_{n,\beta}^h - \sum_{\ell=0}^{2} \beta_\ell^{(n)} u(t_{n-1+\ell}))\big\|^2 \le C\nu \sum_{n=1}^{M-1} \widehat{k}_n (\|\nabla \eta_{n,\beta}\|^2 + \|\nabla \phi_{n,\beta}\|^2) \le C(\theta) \widetilde{F}(h, k_{\max}),$$

which implies (5.10) and complete the proof. □

Remark 13. *For the linearly implicit method, we need the following second-order approximation for* $u_{n,\beta}^h$

$$\widetilde{u}_n^h = \begin{cases} \big(1 + \frac{t_{n,\beta} - t_{n-1,\beta}}{t_{n-1,\beta} - t_{n-2,\beta}}\big) u_{n-1,\beta}^h - \frac{t_{n,\beta} - t_{n-1,\beta}}{t_{n-1,\beta} - t_{n-2,\beta}} u_{n-2,\beta}^h & \text{if } n \ge 3 \\ \\ \Big[\beta_2^{(n)}\big(1 + \frac{k_n}{k_{n-1}}\big) + \beta_1^{(n)}\Big] u_n^h + \big(\beta_0^{(n)} - \beta_2^{(n)} \frac{k_n}{k_{n-1}}\big) u_{n-1}^h & \text{if } n = 1, 2 \end{cases} . \tag{5.33}$$

Then we replace the non-linear term $b(u_{n,\beta}^h, u_{n,\beta}^h, v)$ *by* $b(\widetilde{u}_n^h, u_{n,\beta}^h, v)$ *in the DLN algorithm (5.5) and (5.6). One issue behind the linear DLN algorithm is that we need* $(t_{n,\beta} - t_{n-1,\beta}) > 0$ *for all* n*. For* $\theta = 0, 1$*, this condition always holds and it's easy to check that for* $\theta \in (0, 1)$*, there exist upper bound* $K_1(\theta) \ge 3$ *and lower bound* $0 < K_2(\theta) \le \frac{1}{3}$ *for ratio of steps* k_n/k_{n-1} *such that* $t_{n,\beta} - t_{n-1,\beta} > 0$ *for all* n*. Under this simple step restriction, we have stability and second order accuracy (in time) of the solutions. Moreover for constant time-stepping case,* $(t_{n,\beta} - t_{n-1,\beta}) > 0$ *for all* $\theta \in [0, 1]$ *and thus no time step restriction is needed for stability and error analysis.*

5.6 Numerical Tests

For numerical tests we use FreeFem++ and Taylor-Hood ($P2-P1$) finite elements. We verify the second-order convergence and stability of the DLN algorithm with variable time steps through three numerical experiments.

5.6.1 Convergence Test (constant time step size)

The second-order convergence of the DLN algorithm is verified on the Taylor-Green benchmark problem, see e.g., [121]. In the domain $\Omega = (0,1) \times (0,1)$, the true solution is

$$\begin{aligned} u_1(x,y,t) &= -\cos(w\pi x)\sin(w\pi y)\exp(-2w^2\pi^2 t/\tau), \\ u_2(x,y,t) &= \sin(w\pi x)\cos(w\pi y)\exp(-2w^2\pi^2 t/\tau), \\ p(x,y,t) &= -\frac{1}{4}(\cos(2w\pi x) + \cos(2w\pi y))\exp(-4w^2\pi^2 t/\tau), \end{aligned}$$

and we take the final time $T = 1$, $w = 1$ and $\tau = Re = 100$. The body force f, initial condition, and boundary condition are determined by the true solution. Setting $\Delta t = h$ to calculate the convergence order R by the error e at two successive values of Δt via

$$R = \ln(e(\Delta t_1)/e(\Delta t_2))/\ln(\Delta t_1/\Delta t_2).$$

Tables 17, 18, Tables 19, 20 and Tables 21, 22 correspond to $\theta = 0.2, 0.5, 0.7$, respectively. The results illustrate that the DLN algorithm has second-order convergence for both velocity and pressure. In the tests, the convergence of velocity is better. It's quite common to observe higher rate of convergence for velocity than for velocity gradient. However proving this result requires a fairly long additional duality argument and its details are open.

Table 17: The errors and convergence order of the DLN scheme at time $T = 1$ for the velocity and pressure of L^2-norm with $\theta = 0.2$.

$h = \Delta t$	$\|\|e_u\|\|_{2,0}$	R	$\|\|\nabla e_u\|\|_{2,0}$	R	$\|\|e_p\|\|_{2,0}$	R
$\frac{1}{16}$	0.000740428	-	0.0610604	-	0.00169375	-
$\frac{1}{32}$	8.89412e-05	3.05	0.0141961	2.10	0.000359889	2.33
$\frac{1}{64}$	1.25455e-05	2.82	0.00382655	1.89	8.33864e-05	2.11
$\frac{1}{128}$	1.47688e-06	3.09	0.000897414	2.09	4.47778e-06	2.12
$\frac{1}{256}$	1.63539e-07	3.17	0.000199742	2.17	0.000152877	2.09

Table 18: The errors and convergence order of the DLN scheme at time $T = 1$ for the velocity and pressure of L^∞-norm with $\theta = 0.2$.

$h = \Delta t$	$\|\|e_u\|\|_{\infty}$	R	$\|\|\nabla e_u\|\|_{\infty}$	R	$\|\|e_p\|\|_{\infty}$	R
$\frac{1}{16}$	0.00122596	-	0.101825	-	0.00254809	-
$\frac{1}{32}$	0.000162022	2.92	0.025876	1.98	0.000638476	1.99
$\frac{1}{64}$	0.000162022	2.72	0.00757327	1.77	0.000164576	1.96
$\frac{1}{128}$	2.45897e-05	3.00	0.00187049	2.02	4.04611e-05	2.02
$\frac{1}{256}$	3.50466e-07	3.13	0.000427984	2.13	9.95382e-06	2.02

Table 19: The errors and convergence order of the DLN scheme at time $T = 1$ for the velocity and pressure of L^2-norm with $\theta = 0.5$.

$h = \Delta t$	$\|\|e_u\|\|_{2,0}$	R	$\|\|\nabla e_u\|\|_{2,0}$	R	$\|\|e_p\|\|_{2,0}$	R
$\frac{1}{16}$	0.000700594	-	0.0570129	-	0.00134003	-
$\frac{1}{32}$	8.53722e-05	3.04	0.0135313	2.07	0.000305539	2.13
$\frac{1}{64}$	1.22744e-05	2.80	0.00373636	1.86	7.58267e-05	2.01
$\frac{1}{128}$	1.45834e-06	3.07	0.000885626	2.08	1.8122e-05	2.06
$\frac{1}{256}$	1.6236e-07	3.16	0.000198276	2.16	4.3458e-06	2.06

Table 20: The errors and convergence order of the DLN scheme at time $T = 1$ for the velocity and pressure of L^{∞}-norm with $\theta = 0.5$.

$h = \Delta t$	$\|\|e_u\|\|_{\infty}$	R	$\|\|\nabla e_u\|\|_{\infty}$	R	$\|\|e_p\|\|_{\infty}$	R
$\frac{1}{16}$	0.00110053	-	0.0898315	-	0.00236018	-
$\frac{1}{32}$	0.000147375	2.90	0.0241532	1.89	0.000595252	1.99
$\frac{1}{64}$	2.3207e-05	2.67	0.00716777	1.75	0.000153932	1.95
$\frac{1}{128}$	2.91792e-06	2.99	0.00178222	2.01	3.79152e-05	2.02
$\frac{1}{256}$	3.34876e-07	3.12	0.000409336	2.12	9.32863e-06	2.02

Table 21: The errors and convergence order of the DLN scheme at time $T = 1$ for the velocity and pressure of L^{2}-norm with $\theta = 0.7$.

$h = \Delta t$	$\|\|e_u\|\|_{2,0}$	R	$\|\|\nabla e_u\|\|_{2,0}$	R	$\|\|e_p\|\|_{2,0}$	R
$\frac{1}{16}$	0.000689478	-	0.0560293	-	0.00127634	-
$\frac{1}{32}$	8.45301e-05	3.03	0.0133912	2.06	0.000296992	2.10
$\frac{1}{64}$	1.22087e-05	2.79	0.00371588	1.85	7.46964e-05	1.99
$\frac{1}{128}$	1.45411e-06	3.07	0.000883034	2.07	1.79769e-05	2.05
$\frac{1}{256}$	1.62107e-07	3.17	0.000197972	2.16	4.32679e-06	2.05

Table 22: The errors and convergence order of the DLN scheme at time $T = 1$ for the velocity and pressure of L^{∞}-norm with $\theta = 0.7$.

$h = \Delta t$	$\|\|e_u\|\|_{\infty}$	R	$\|\|\nabla e_u\|\|_{\infty}$	R	$\|\|e_p\|\|_{\infty}$	R
$\frac{1}{16}$	0.00101829	-	0.0878696	-	0.00241273	-
$\frac{1}{32}$	0.000146272	2.80	0.0240831	1.87	0.000611728	1.98
$\frac{1}{64}$	2.32681e-05	2.65	0.00720402	1.74	0.000158518	1.95
$\frac{1}{128}$	2.94608e-06	2.98	0.00180182	2.00	3.91189e-05	2.02
$\frac{1}{256}$	3.39458e-07	3.12	0.000415227	2.12	9.63668e-06	2.02

5.6.2 2D Offset Circles Problem (with preset variable time step size)

This is a test problem from Jiang and Layton [71] that is inspired by the flow between offset cylinders. The domain is a disk with a smaller off-center obstacle inside. Let $\Omega_1 = \{(x,y) : x^2 + y^2 \leq 1\}$ and $\Omega_2 = \{(x,y) : (x-\frac{1}{2})^2 + y^2 \geq 0.01\}$. The flow is driven by a rotational body force:

$$f(x,y,t) = (-4y(1-x^2-y^2), 4x(1-x^2-y^2))^T.$$

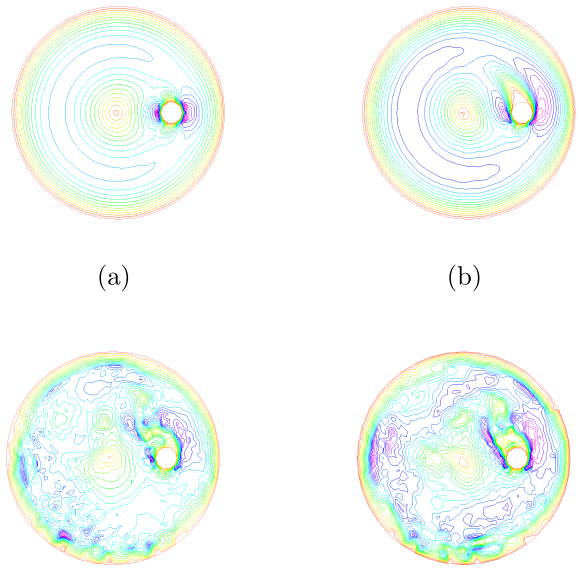
with no-slip boundary conditions imposed on both circles. The body force $f = 0$ is on the outer circle. The flow rotates about $(0,0)$ and the inner circle induces a von Kármán vortex street which re-interacts with the immersed circle creating more complex structures. Figure 23 and Figure 24 show this situation.

For this test, we set $Re = 200$, the number of mesh points around the inner circle and the mesh points around the outer circle to be 10 and 40 respectively. The parameter $\theta = 0.5$ in DLN scheme, for the variable time step size, the number of computations is $n = 1000$. We let the time step size change as the function used in Chen, Layton, and Mclaughlin [23] to test the stability of different methods:

$$k_n = \begin{cases} 0.05 & 0 \leq n \leq 10, \\ 0.05 + 0.002\sin(10t_n) & n > 10. \end{cases}$$

For comparison, we also solve this problem with standard (variable step) BDF2, BDF3, and BDF4 time discretization. We calculate the $\frac{1}{2}\|u\|^2$(energy), $\|u\|$ and $\|\nabla u\|$ using BDF2, BDF3, BDF4 and DLN algorithms respectively. Here, let the number of mesh points on boundary of outside circle and inner circle be 160 and 40 respectively and time step be $k_0 = 0.05$ and $k_n = k_{n-1} + 0.001$. We stop the simulation when the time step size reaches 0.5 since the time step size greater than that value will result in the inaccuracy of computation. In Figure 25(b) and Figure 25(c), *energy and* $\|u\|$ *by BDF2* have the trend to increase with increasing time step, while the *energy and* $\|u\|$ *by DLN* remain at low level. BDF3 and BDF4 (more accurate in time) have energy and $\|u\|$ value between those of BDF2 and DLN. For H^1-norm, Figure 25(d) shows that $\|\nabla u\|$ by the four algorithms are at the same level while

DLN and BDF4 have larger oscillations than BDF2 and BDF3 as time step size increases. In summary, this test verifies that the DLN algorithm has greater stability for energy than BDF2.



(a) (b)

(c) (d)

Figure 23: Speed Contours of DLN.

5.6.3 Adapting the time step

Finally, we use this example to perform a simple adaptivity experiment. For this test, we adapt the time step using the minimum dissipation criterion of Capuano, Sanderse, De Angelis and Coppola [20]. Our goal is to test if adapting the time step produces a significant difference in the solution. Other criteria/estimators are under study. Their idea is to adapt the time step to keep the numerical dissipation, ϵ^{DLN} from the dominating physical dissipation, ϵ^{ν}. Thus we adapt for

$$\chi = \left| \frac{\epsilon^{DLN}}{\epsilon^{\nu}} \right| < \delta.$$

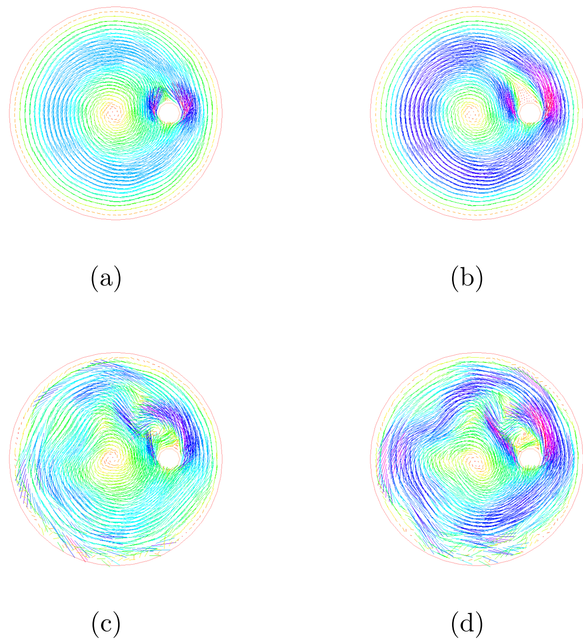


Figure 24: Velocity Streamlines of DLN.

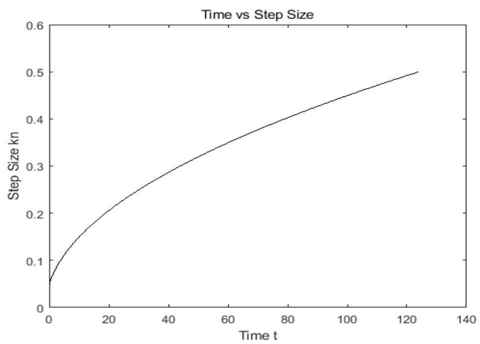
Here ϵ^{DLN} is the numerical dissipation and ϵ^{ν} is the viscous dissipation. These are given by:

$$\epsilon^{DLN} = \left\| \frac{\sum_{\ell=0}^{2} a_{\ell}^{n} u_{n-1+\ell}^{h}}{\sqrt{\hat{k}_n}} \right\|^2, \qquad \epsilon^{\nu} = \nu \left\| \nabla u_{n,\beta}^{h} \right\|^2 .$$

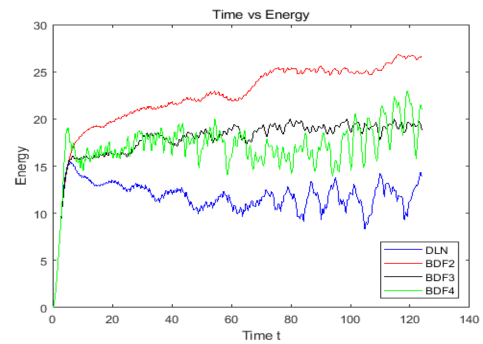
In the test, we set the tolerance for the dissipation ratio δ to be 0.002. The time step size is then adapted by the simplest strategy of halving or doubling according to

$$\Delta t^{n+1} = \min\{2 * \Delta t^n,\ 0.5\}; \quad \text{if } \ \chi < \delta, \qquad \Delta t^n = \max\{0.5 * \Delta t^n,\ 0.01\}; \qquad \text{if } \ \chi \geq \delta.$$

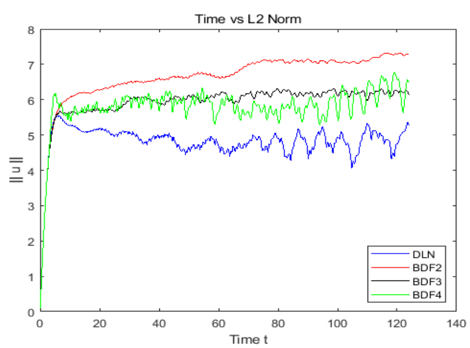
We adapted the next time step when the dissipation ratio was out of range. Naturally, other strategies for varying Δt could be tested, such as formula (16) in p.2317 of Capuano, Sanderse, De Angelis and Coppola [20] (which is $\Delta t^{n+1} = \Delta t^n |\delta/\chi|^{1/2}$). We select the final time $T = 60$, minimal time step size and maximal time step size to be 0.01 and 0.5 respectively. The adaptive algorithm completed in 5687 steps. Figure 26 and Figure 27 are line diagrams of time step size k_n, energy $\frac{1}{2}\|u\|^2$, numerical dissipation $\sqrt{\epsilon^{DLN}}$ and ratio χ changing with time t, respectively.



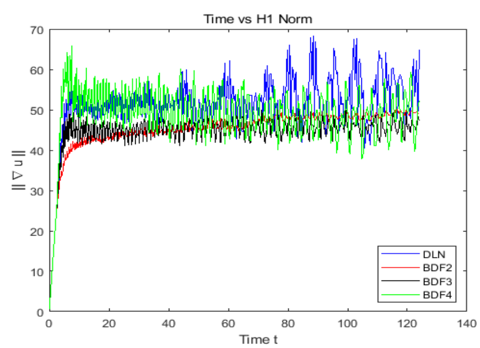
(a)



(b)



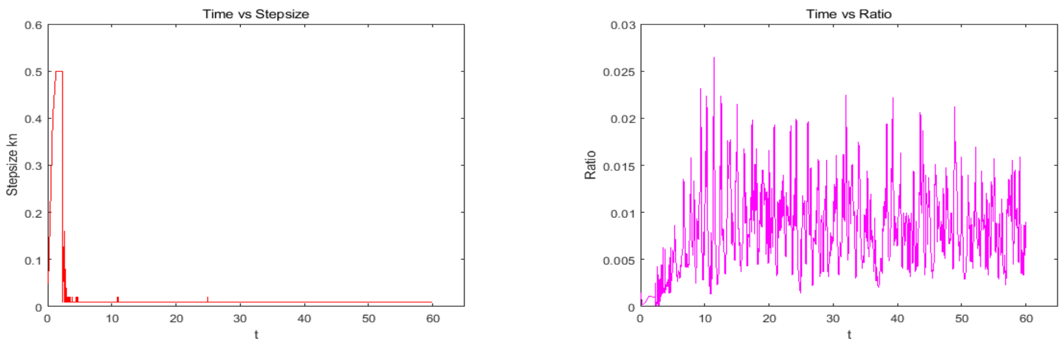
(c)



(d)

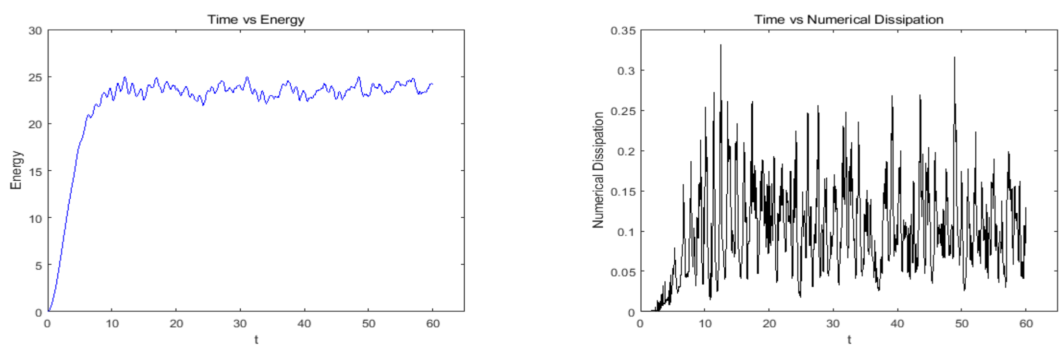
Figure 25: Energy, $\|u\|$ and $\|\nabla u\|$ of DLN, BDF2, BDF3 and BDF4 with variable time step size.

Then we select the same final time $T = 60$, the same calculated steps 5687 and use the constant time step $k = T/5687$ to calculate to obtain the line diagram of energy $\frac{1}{2}\|u\|^2$, numerical dissipation $\sqrt{\epsilon^{DLN}}$ and ratio χ changing with time t, see Figure 28 and Figure 29. We now compare the constant time step size results in Figure 28 and Figure 29 with



(a) (b)

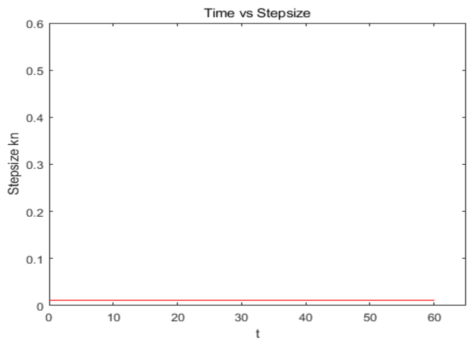
Figure 26: The time step size k_n and ratio χ changing with adaptive time step size.



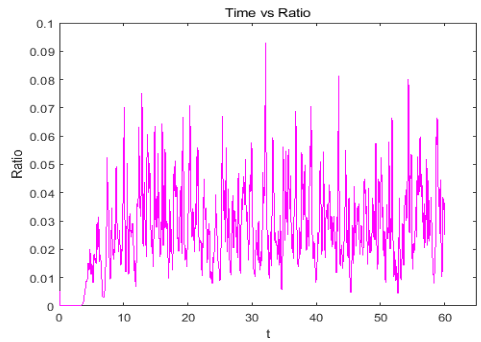
(a) (b)

Figure 27: The energy $\frac{1}{2}\|u\|^2$ and numerical dissipation $\sqrt{\epsilon^{DLN}}$ changing with adaptive time step size.

the adaptive results in Figure 26 and Figure 27. Time step size under adaptivity reaches the maximum value 0.5 in a few steps then goes down sharply to the minimum step size 0.01 thereafter. In the test represented in Figure 26(a), the time step alternates between the minimum step size and twice the minimum. This is due to the preset algorithmic choice.

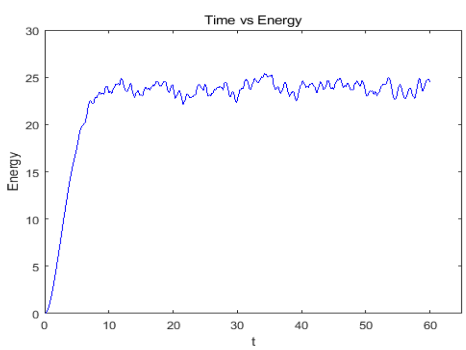


(a)

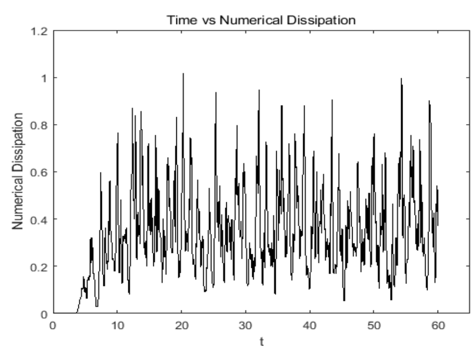


(b)

Figure 28: The time step size k and ratio χ changing with constant time step size.



(a)



(b)

Figure 29: The energy $\frac{1}{2}\|u\|^2$ and numerical dissipation $\sqrt{\epsilon^{DLN}}$ changing with constant time step size.

DLN under constant step size takes 777 time steps to reach kinetic energy of approximately 23, a level which adaptive DLN algorithm reaches in 544 time steps. In the comparison of numerical dissipation, Figure 27(b) and 29(b) show the numerical dissipation with adaptive time step size evolves smoothly with a peak value below 0.35. Similarly the ratio χ has a *order of magnitude smaller for adaptive time step size*, Figure 26(b), than constant time step size, Figure 29(b). Comparing Figure 27 and Figure 29, we can see that energy pattens are similar for both variable and constant step algorithms but numerical dissipation by the variable step DLN algorithm is much smaller than that of constant case.

5.7 Conclusions

Based on the theory and the simple numerical tests for time discretization of flow problems the 2-step DLN method is to be preferred over the common BDF2 method. It is second-order, unconditionally, long-time, and nonlinearly stable. For increasing step sizes, BDF2 injects nonphysical kinetic energy in the discrete solution (disrupting long time behavior and statistical equilibrium) while DLN does not. Important open questions include how to select the DLN parameter θ. At this point, we have no systematic (either universal or application-specific) method to choose an optimal DLN parameter θ balancing stability and accuracy. How to perform error estimation in a memory and computationally efficient (and effective) way is also an important open problem. In particular, finding a memory efficient estimator, as was done in Gresho, Sani, and Engelman [51] for the trapezoid rule, is a necessary step. It would be useful if the DLN method could be embedded in a family of different orders with good properties or if it could be induced from simpler methods by added time filters. Both are open problems.

6.0 Conclusion and Future Work

Well-developed finite element methods for spatial discretization, accompanied by time discretization of low accuracy, are employed in most CFD simulations. Easily implemented time-accurate algorithms with low storage, little-explored but highly expected, would strengthen the reliability of CFD simulation. Time step adaptivity is an effective way of balancing time accuracy and computational efficiency, which results in great interest in variable time-stepping analysis for fluid problems. Dahlquist, Liniger, and Nevanlinna [33] have proposed a one-parameter family, which is G-stable (nonlinearly, energetically stable) and second-order accurate for any arbitrary sequence of time steps. To my knowledge, this method is the unique one that possesses such two excellent properties. **My work is to analyze the method of Dahlquist, Liniger, and Nevanlinna (the DLN method), unearthing the properties of the method and applying it to fluid models**.

The complicated form of the DLN method deters its testing in CFD where its excellent properties should be valued. To solve this issue, I refactorize the DLN method by adding pre-filter and post-filter on the backward Euler method and obtain the following algorithm for each step computation [84]

Algorithm: Refactorization of the DLN Method

Input: y_n, y_{n-1} ;

$y_n^{\text{old}} \Leftarrow a_1^{(n)} y_n + a_0^{(n)} y_{n-1}$;

$k_n^{\text{new}} \Leftarrow b^{(n)} \widehat{k}_n$;

$t_{n+1}^{\text{new}} \Leftarrow \beta_2^{(n)} t_{n+1} + \beta_1^{(n)} t_n + \beta_0^{(n)} t_{n-1}$; `// pre-filter for` y_n , k_n `and` t_{n+1}

Solve for y_{n+1}^{temp}: $\frac{y_{n+1}^{\text{temp}} - y_n^{\text{old}}}{k_n^{\text{new}}} = f\left(t_{n+1}^{\text{new}}, y_{n+1}^{\text{temp}}\right)$; `// backward Euler algorithm`

$y_{n+1} \Leftarrow y_{n+1}^{\text{temp}} + \left(c_2^{(n)} y_{n+1}^{\text{temp}} + c_1^{(n)} y_n + c_0^{(n)} y_{n-1}\right)$; `// post-filter for` $y_{n+1}^{\texttt{temp}}$

To further develop the DLN method, I have obtained the expressions of numerical dissipation in G-stability identity and local truncation error (LTE) for the DLN algorithm, two important criteria for measuring the effect of time-stepping algorithms on fluid models. To adapt time steps, I extend Gresho's idea [18, 47, 50] to derive some estimators of LTE for variable time-stepping DLN method with aid of some explicit methods and the general time step controller proposed by Söderlind [116].

Equipped with these detailed properties, **I have applied the variable time-stepping DLN method to some flow problems (unsteady Stokes/Darcy model and Navier Stokes equations (NSE)) and performed a completed stability and error analysis of approximate solutions** in Chapter 4 and 5, see also [83,107]. The approximate solutions are unconditionally, long-time stable, and second-order accurate under variable time steps. I have implemented the DLN method to Taylor-Green benchmark problem [121] to confirm the second-order convergence rate and adjusted time step using the minimum-dissipation criteria of Capuano, Sanderse, De Angelis and Coppola [20] for the variable step test problem from Jiang [71] that is inspired by the flow between offset cylinders. The minimum-dissipation strategy, adding only a few lines of code, can be implemented simply to suppress the time-integration error with desired tolerance and increase efficiency dramatically.

6.1 Future Work

6.1.1 The DLN-ensemble Algorithm for Navier Stokes equations

To solve J ($J > 1$) NSEs simultaneously, Jiang and Layton [71] combine backward Euler method and ensemble averaging technique to obtain the following algorithm for each jth NSE ($j = 1, 2, \cdots, J$)

$$\frac{u_{n+1}^{j,h} - u_n^{j,h}}{k_n} + \langle u_n^h \rangle \cdot \nabla u_{n+1}^{j,h} + \big(u_n^{j,h} - \langle u_n^h \rangle \big) \cdot \nabla u_n^{j,h} - \nu \Delta u_{n+1}^{j,h} + \nabla p_{n+1}^{j,h} = f_{n+1}^j,$$

$$\nabla \cdot u_{n+1}^{j,h} = 0,$$

where $u_n^{j,h}$ and $p_n^{j,h}$ are the approximate solutions to jth NSEs at time t_n and $\langle u_n^h \rangle := \frac{1}{J}\big(\sum_{j=1}^J u_n^{j,h} \big)$ is the ensemble average. The above algorithm at each step is equivalent to the following block linear system

$$\begin{bmatrix} \frac{1}{k_n} M_u + \nu S_u + N_u(\langle u_n^h \rangle) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u_{n+1}^{j,h} \\ p_{n+1}^{j,h} \end{bmatrix} = \begin{bmatrix} f_{n+1}^j + \big(\frac{1}{k_n} M_u + N_u(u_n^{j,h} - \langle u_n^h \rangle) \big) u_n^{j,h} \\ 0 \end{bmatrix}, \tag{6.1}$$

where M_u is the mass matrix, S_u is the diffusion matrix, N_u is the convection matrix and B is the continuity matrix. The resulting coefficient matrix in (6.1), denoted by A, is independent of j. Denote the solution vector and the vector on the right hand side by x_j and b_j respectively, the ensemble algorithms of J NSE is reduced to

$$\Big[A\Big] \Big[x_1|x_2|\cdots|x_J\Big] = \Big[b_1|b_2|\cdots|b_J\Big],$$

which solves J NSE at the same time as well as significantly reduces the storage due to the shared coefficient matrix.

Applying the DLN algorithm to NSEs, we have the DLN-ensemble algorithm

$$\frac{\alpha_2 u_{n+1}^{j,h} + \alpha_1 u_n^{j,h} + \alpha_0 u_{n-1}^{j,h}}{\widehat{k}_n} + \big\langle \widetilde{u}_n^h \big\rangle \cdot \nabla u_{n,\beta}^{j,h} + \big(\widetilde{u}_n^{j,h} - \big\langle \widetilde{u}_n^h \big\rangle\big) \cdot \nabla \widetilde{u}_n^{j,h} - \nu \Delta u_{n,\beta}^{j,h} + \nabla p_{n,\beta}^{j,h} = f_{n,\beta}^{j},$$

$$\nabla \cdot u_{n+1}^{j,h} = 0,$$

where

$$\widetilde{u}_n^{j,h} := \Big[\beta_2^{(n)}\big(1 + \frac{k_n}{k_{n-1}}\big) + \beta_1^{(n)}\Big] u_n^{j,h} + \big(\beta_0^{(n)} - \beta_2^{(n)} \frac{k_n}{k_{n-1}}\big) u_{n-1}^{j,h}$$

is the second-order approximation (in time) to $u_{n,\beta}^{j,h}$.

We expect the unconditional stability of the solution and second-order convergence under certain time step condition.

6.1.2 The Semi-implicit DLN Algorithm for Navier Stokes equations

The error estimate in DLN algorithm in (5.5) or (5.6) involves the time step condition

$$\Delta t < \mathcal{O}\big(\frac{1}{\nu^3}\big)$$

which is prohibitively restrictive in practice if the flow is highly viscous. In addition, the non-linear system solver in the DLN algorithm in (5.5) or (5.6) keeps the computation cost relatively high.

We replace the non-linear term $u_{n,\beta}^h \cdot \nabla u_{n,\beta}^h$ by $\widetilde{u}_n^h \cdot \nabla u_{n,\beta}^h$, $\widetilde{u}_n^h$ is the second-order approximation (in time) to $u_{n,\beta}^h$, i.e.

$$\widetilde{u}_n^h := \Big[\beta_2^{(n)}\big(1 + \frac{k_n}{k_{n-1}}\big) + \beta_1^{(n)}\Big] u_n^h + \big(\beta_0^{(n)} - \beta_2^{(n)} \frac{k_n}{k_{n-1}}\big) u_{n-1}^h.$$

Thus we have the semi-implicit DLN Algorithm for NSEs

$$\frac{\alpha_2 u_{n+1}^h + \alpha_1 u_n^h + \alpha_0 u_{n-1}^h}{\widehat{k}_n} + \widetilde{u}_n^h \cdot \nabla u_{n,\beta}^h - \nu \Delta u_{n,\beta}^h + \nabla p_{n,\beta}^h = f_{n,\beta},$$
$$\nabla \cdot u_{n+1}^h = 0. \tag{6.2}$$

We expect that the approximate solution by the semi-implicit DLN algorithm in (6.2) is unconditionally stable. Moreover, we would like to obtain the second-order error estimate without the time step restriction.

Bibliography

[1] M. AKBAŞ. An adaptive time filter based finite element method for the velocity-vorticity-temperature model of the incompressible non-isothermal fluid flows. *Gazi University Journal of Science*, pages 1–1, 2020.

[2] T. Arbogast and M. S. M. Gomez. A discretization and multigrid solver for a Darcy–Stokes system of three dimensional vuggy porous media. *Computational Geosciences*, 13(3):331–348, 2009.

[3] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1st edition, 1998.

[4] R. Asselin. Frequency filter for time integrations. *Monthly Weather Review*, 100(6):487–490, 1972.

[5] L. Badea, M. Discacciati, and A. Quarteroni. Numerical analysis of the Navier–Stokes/Darcy coupling. *Numerische Mathematik*, 115(2):195–227, 2010.

[6] S. Badia, F. Nobile, and C. Vergara. Robin–Robin preconditioned Krylov methods for fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 198(33-36):2768–2784, 2009.

[7] G. A. Baker, V. A. Dougalis, and O. A. Karakashian. On a higher order accurate fully discrete Galerkin approximation to the Navier-Stokes equations. *Mathematics of Computation*, 39(160):339–375, 1982.

[8] B. Boutelje and A. Hill. Nonautonomous stability of linear multistep methods. *IMA Journal of Numerical Analysis*, 30(2):525–542, 4 2010.

[9] S. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*. Texts in Applied Mathematics. Springer New York, 2007.

[10] M. Bristeau, R. Glowinski, and J. Periaux. Numerical methods for the Navier-Stokes equations. applications to the simulation of compressible and incompressible viscous flows. *Computer Physics Reports*, 6(1):73 – 187, 1987.

[11] M. Bukač, A. Seboldt, and C. Trenchea. Refactorization of Cauchy's method: A second-order partitioned method for fluid–thick structure interaction problems. *Journal of Mathematical Fluid Mechanics*, 23(3):1–25, 2021.

[12] M. Bukač and C. Trenchea. Adaptive, second-order, unconditionally stable partitioned method for fluid-structure interaction. Technical report, University of Pittsburgh, 2020.

[13] M. Bukač and C. Trenchea. Boundary update via resolvent for fluid–structure interaction. *Journal of Numerical Mathematics*, 29(1):1–22, 2021.

[14] M. Bukač and C. Trenchea. Adaptive, second-order, unconditionally stable partitioned method for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 393:114847, 2022.

[15] J. Burkardt and C. Trenchea. Refactorization of the midpoint rule. *Applied Mathematics Letters*, 107:106438, 2020.

[16] E. Burman and M. A. Fernández. Stabilization of explicit coupling in fluid–structure interaction involving fluid incompressibility. *Computer Methods in Applied Mechanics and Engineering*, 198(5-8):766–784, 2009.

[17] J. C. Butcher. A stability property of implicit Runge-Kutta methods. *BIT Numerical Mathematics*, 15(4):358–361, Dec 1975.

[18] J. C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[19] Y. Cao, M. Gunzburger, X. He, and X. Wang. Parallel, non-iterative, multi-physics domain decomposition methods for time-dependent Stokes-Darcy systems. *Mathematics of Computation*, 83(288):1617–1644, 2014.

[20] F. Capuano, B. Sanderse, E. De Angelis, and G. Coppola. A minimum-dissipation time-integration strategy for large-eddy simulation of incompressible turbulent flows. In *AIMETA 2017 Proceedings of the XXIII Conference of the Italian Association of Theoretical and Applied Mechanics*, pages 2311–2323, sep 2017.

[21] A. Çeşmelio**u**glu and B. Rivière. Analysis of time-dependent Navier–Stokes flow coupled with Darcy flow. *Journal of Numerical Mathematics*, 16(4):249–280, 2008.

[22] A. Çeşmelioglu and B. Rivière. Primal discontinuous Galerkin methods for time-dependent coupled surface and subsurface flow. *Journal of Scientific Computing*, 40(1-3):115–140, 2009.

[23] R. Chen, W. J. Layton, and M. McLaughlin. Analysis of variable-step/non-autonomous artificial compression methods. *Journal of Mathematical Fluid Mechanics*, 21, 06 2019.

[24] W. Chen, M. Gunzburger, D. Sun, and X. Wang. Efficient and long-time accurate second-order methods for the Stokes–Darcy system. *SIAM Journal on Numerical Analysis*, 51(5):2563–2584, 2013.

[25] W. Chen, M. Gunzburger, D. Sun, and X. Wang. An efficient and long-time accurate third-order algorithm for the Stokes–Darcy system. *Numerische Mathematik*, 134(4):857–879, 2016.

[26] A. Cibik, F. G. Eroglu, and S. Kaya. Analysis of second order time filtered backward Euler method for MHD equations. *Journal of Scientific Computing*, 82(2):1–25, 2020.

[27] G. G. Dahlquist. On stability and error analysis for stiff non-linear problems, Part I. *Department of Computation Science, Royal Institute of Technology*, Report TRITA-NA-7508, 1975.

[28] G. G. Dahlquist. Error analysis for a class of methods for stiff non-linear initial value problems. In G. A. Watson, editor, *Numerical Analysis*, pages 60–72, Berlin, Heidelberg, 1976. Springer Berlin Heidelberg.

[29] G. G. Dahlquist. On the relation of G-stablity to other stability concepts for linear multistep methods. Technical report, CM-P00069426, 1976.

[30] G. G. Dahlquist. G-stability is equivalent to A-stability. *BIT Numerical Mathematics*, 18(4):384–401, Dec 1978.

[31] G. G. Dahlquist. On one-leg multistep methods. *SIAM Journal on Numerical Analysis*, 20(6):1130–1138, 1983.

[32] G. G. Dahlquist. Positive functions and some applications to stability questions for numerical methods. In *Recent Advances in Numerical Analysis: Proceedings of a Symposium Conducted by the Mathematics Research Center, the University of Wisconsin-Madison*, May 22-24,1978.

[33] G. G. Dahlquist, W. Liniger, and O. Nevanlinna. Stability of two-step methods for variable integration steps. *SIAM Journal on Numerical Analysis*, 20(5):1071–1085, 1983.

[34] P. A. de Sampaio, P. H. Hallak, A. L. Coutinho, and M. S. Pfeil. A stabilized finite element procedure for turbulent fluid–structure interaction using adaptive time–space refinement. *International Journal for Numerical Methods in Fluids*, 44(6):673–693, 2004.

[35] V. DeCaria, S. Gottlieb, Z. J. Grant, and W. J. Layton. A general linear method approach to the design and optimization of efficient, accurate, and easily implemented time-stepping methods in CFD. *Journal of Computational Physics*, page 110927, 2022.

[36] V. DeCaria and M. Schneier. An embedded variable step IMEX scheme for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 376:113661, 2021.

[37] V. P. DeCaria. *Variable Stepsize, Variable Order Methods for Partial Differential Equations*. PhD thesis, University of Pittsburgh, 2019.

[38] M. Discacciati, E. Miglio, and A. Quarteroni. Mathematical and numerical models for coupling surface and groundwater flows. *Applied Numerical Mathematics*, 43(1-2):57–74, 2002.

[39] G. Du and L. Zuo. Local and parallel finite element method for the mixed Navier-Stokes/Darcy model with Beavers-Joseph interface conditions. *Acta Mathematica Scientia*, 37(5):1331–1347, 2017.

[40] G. Du and L. Zuo. Local and parallel finite element methods for the coupled S-tokes/Darcy model. *Numerical Algorithms*, pages 1–19, 2020.

[41] G. Du and L. Zuo. A two-grid method with backtracking for the mixed Stokes/Darcy model. *Journal of Numerical Mathematics*, 1(ahead-of-print), 2020.

[42] D. R. Durran. The third-order Adams-Bashforth method: An attractive alternative to Leapfrog time differencing. *Monthly weather review*, 119(3):702–720, 1991.

[43] E. Emmrich. Error of the two-step BDF for the incompressible Navier-Stokes problem. *M2AN. Mathematical Modelling and Numerical Analysis*, 38(5):757–764, 2004.

[44] V. Ervin, E. Jenkins, and S. Sun. Coupled generalized nonlinear Stokes flow with flow through a porous medium. *SIAM Journal on Numerical Analysis*, 47(2):929–952, 2009.

[45] W. Feng, X. He, Z. Wang, and X. Zhang. Non-iterative domain decomposition methods for a non-stationary Stokes–Darcy model with Beavers–Joseph interface condition. *Applied Mathematics and Computation*, 219(2):453–463, 2012.

[46] C. W. Gear. Algorithm 407: DIFSUB for solution of ordinary differential equations [D2]. *Communications of the ACM*, 14(3):185–190, 1971.

[47] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Automatic Computation Series. Prentice-Hall, 1971.

[48] V. Girault and P.-A. Raviart. *Finite element approximation of the Navier-Stokes equations*, volume 749. Springer Berlin, 1979.

[49] V. Girault and B. Rivière. DG approximation of coupled Navier–Stokes and Darcy equations by Beaver–Joseph–Saffman interface condition. *SIAM Journal on Numerical Analysis*, 47(3):2052–2089, 2009.

[50] P. Gresho and R. Sani. *Incompressible flow and the finite element method. Volume 2: Isothermal Laminar Flow.* John Wiley, Chichester, UK, 2000.

[51] P. Gresho, R. Sani, and M. Engelman. *Incompressible flow and the finite element method: advection-diffusion and isothermal laminar flow.* Incompressible Flow & the Finite Element Method. Wiley, 1998.

[52] P. Gresho, R. Sani, and M. Engelman. *Incompressible flow and the finite element method, Volume 2: Isothermal Laminar Flow.* Incompressible Flow & the Finite Element Method. Wiley, 1998.

[53] P. M. Gresho, D. F. Griffiths, and D. J. Silvester. Adaptive time-stepping for incompressible flow part I: Scalar advection-diffusion. *SIAM Journal on Scientific Computing*, 30(4):2018–2054, 2008.

[54] P. M. Gresho, R. L. Lee, S. T. Chan, and R. L. Sani. Solution of the time-dependent incompressible Navier-Stokes and Boussinesq equations using the Galerkin finite element method. In *Approximation methods for Navier-Stokes problems*, pages 203–222. Springer, 1980.

[55] P. M. Gresho, R. L. Lee, R. L. Sani, and T. Stullich. Time-dependent FEM solution of the incompressible Navier–Stokes equations in two-and three-dimensions. Technical report, California University, 1978.

[56] D. Griffiths and D. Higham. *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. Springer Undergraduate Mathematics Series. Springer London, 2010.

[57] R. D. Grigorieff. Stability of multistep-methods on variable grids. *Numerische Mathematik*, 42(3):359–377, Oct. 1983.

[58] A. Guzel and W. Layton. Time filters increase accuracy of the fully implicit method. *BIT Numerical Mathematics*, 58:301–315, 2018.

[59] A. Guzel, W. Layton, M. McLaughlin, and Y. Rong. Time filters and spurious acoustics in artificial compression methods. *Numerical Methods for Partial Differential Equations*, pages 1–21, 2021.

[60] A. Guzel and C. Trenchea. The Williams step increases the stability and accuracy of the hoRA time filter. *Applied Numerical Mathematics*, 131:158–173, 2018.

[61] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Solving ordinary differential equations. Springer, 1993.

[62] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Solving Ordinary Differential Equations. Springer, 1993.

[63] A. Hay, S. Etienne, D. Pelletier, and A. Garon. hp-Adaptive time integration based on the BDF for viscous flows. *Journal of Computational Physics*, 291:151–176, 2015.

[64] X. He, J. Li, Y. Lin, and J. Ming. A domain decomposition method for the steady-state Navier–Stokes–Darcy model with Beavers–Joseph interface condition. *SIAM Journal on Scientific Computing*, 37(5):S264–S290, 2015.

[65] J. G. Heywood and R. Rannacher. Finite-element approximation of the non-stationary Navier-Stokes problem. Part IV: Error analysis for second-order time discretization. *SIAM Journal on Numerical Analysis*, 27(2):353–384, 1990.

[66] Y. Hou. Optimal error estimates of a decoupled scheme based on two-grid finite element for mixed Stokes–Darcy model. *Applied Mathematics Letters*, 57:90–96, 2016.

[67] Y. Hou and Y. Qin. On the solution of coupled Stokes/Darcy model with Beavers–Joseph interface condition. *Computers & Mathematics with Applications*, 77(1):50–65, 2019.

[68] R. Ingram. Unconditional convergence of high-order extrapolations of the Crank-Nicolson, finite element method for the Navier-Stokes equations. *International Journal of Numerical Analysis & Modeling*, 10(2), 2013.

[69] N. Jiang. A higher order ensemble simulation algorithm for fluid flows. *Journal of Scientific Computing*, 64(1):264–288, 2015.

[70] N. Jiang. A second-order ensemble method based on a blended backward differentiation formula timestepping scheme for time-dependent Navier–Stokes equations. *Numerical Methods for Partial Differential Equations*, 33(1):34–61, 2017.

[71] N. Jiang and W. Layton. An algorithm for fast calculation of flow ensembles. *International Journal for Uncertainty Quantification*, 4(4), 2014.

[72] N. Jiang, M. Mohebujjaman, L. G. Rebholz, and C. Trenchea. An optimally accurate discrete regularization for second order timestepping methods for naviercstokes equations. *Computer Methods in Applied Mechanics and Engineering*, 310:388 – 405, 2016.

[73] N. Jiang and C. Qiu. An efficient ensemble algorithm for numerical approximation of stochastic Stokes-Darcy equations. *Computer Methods in Applied Mechanics and Engineering*, 343:249 – 275, 2019.

[74] G. Kanschat and B. Riviere. A strongly conservative finite element method for the coupling of Stokes and Darcy flow. *Journal of Computational Physics*, 229(17):5933–5943, 2010.

[75] D. A. Kay, P. M. Gresho, D. F. Griffiths, and D. J. Silvester. Adaptive time-stepping for incompressible flow part II: Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 32(1):111–128, 2010.

[76] G. Y. Kulikov and S. K. Shindin. One-leg integration of ordinary differential equations with global error control. *Computational Methods in Applied Mathematics*, 5(1):86–96, 2005.

[77] G. Y. Kulikov and S. K. Shindin. One-leg variable-coefficient formulas for ordinary differential equations and local–global step size control. *Numerical Algorithms*, 43(1):99–121, 2006.

[78] A. Labovsky, W. Layton, C. Manica, M. Neda, and L. Rebholz. The stabilized extrapolated trapezoidal finite-element method for the Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 198(9-12):958–974, 2009.

[79] J. D. Lambert. *Computational methods in ordinary differential equations*. John Wiley & Sons, London-New York-Sydney, 1973. Introductory Mathematics for Scientists and Engineers.

[80] J. D. Lambert. *Numerical methods for ordinary differential systems*. John Wiley & Sons, Ltd., Chichester, 1991. The initial value problem.

[81] W. Layton, N. Mays, M. Neda, and C. Trenchea. Numerical analysis of modular regularization methods for the BDF2 time discretization of the Navier-Stokes equations. *ESAIM. Mathematical Modelling and Numerical Analysis*, 48(3):765–793, 2014.

[82] W. Layton and M. McLaughlin. Doubly-adaptive artificial compression methods for incompressible flow. *Journal of Numerical Mathematics*, 28(3):175–192, 2020.

[83] W. Layton, W. Pei, Y. Qin, and C. Trenchea. Analysis of the variable step method of Dahlquist, Liniger and Nevanlinna for fluid flow. *Numerical Methods for Partial Differential Equations*, pages 1–25, 2021.

[84] W. Layton, W. Pei, and C. Trenchea. Refactorization of a variable step, unconditionally stable method of Dahlquist, Liniger and Nevanlinna. *Applied Mathematics Letters*, 125:107789, 2022.

[85] W. Layton, F. Schieweck, and I. Yotov. Coupling fluid flow with porous media flow. *SIAM Journal on Numerical Analysis*, 40(6):2195–2218, 2002.

[86] W. Layton, H. Tran, and C. Trenchea. Numerical analysis of two partitioned methods for uncoupling evolutionary MHD flows. *Numerical Methods for Partial Differential Equations*, 30(4):1083–1102, 2014.

[87] W. Layton and C. Trenchea. Stability of two IMEX methods, CNLF and BDF2-AB2, for uncoupling systems of evolution equations. *Applied Numerical Mathematics*, 62(2):112–120, 2012.

[88] Y. Li and Y. Hou. A second-order partitioned method with different subdomain time steps for the evolutionary Stokes-Darcy system. *Mathematical Methods in the Applied Sciences*, 41(5):2178–2208, 2018.

[89] Y. Li, Y. Hou, W. Layton, and H. Zhao. Adaptive partitioned methods for the time-accurate approximation of the evolutionary Stokes–Darcy system. *Computer Methods in Applied Mechanics and Engineering*, 364:112923, 2020.

[90] Y. Li, Y. Hou, and Y. Rong. A second-order artificial compression method for the evolutionary Stokes-Darcy system. *Numerical Algorithms*, pages 1–30, 2019.

[91] Y. Li and C. Trenchea. A higher-order Robert–Asselin type time filter. *Journal of Computational Physics*, 259:23–32, 2014.

[92] H.-l. Liao and Z. Zhang. Analysis of adaptive BDF2 scheme for diffusion equations. *Mathematics of Computation*, 90(329):1207–1226, 2021.

[93] W. Liniger. The A-contractive second-order multistep formulas with variable steps. *SIAM Journal on Numerical Analysis*, 20(6):1231–1238, 1983.

[94] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[95] M. Mayr, W. Wall, and M. Gee. Adaptive time stepping for fluid-structure interaction solvers. *Finite Elements in Analysis and Design*, 141:55–69, 2018.

[96] W. E. Milne. Numerical Integration of Ordinary Differential Equations. *American Mathematical Monthly*, 33(9):455–460, 1926.

[97] M. Mu and X. Zhu. Decoupled schemes for a non-stationary mixed Stokes-Darcy model. *Mathematics of Computation*, 79(270):707–731, 2010.

[98] O. Nevanlinna and W. Liniger. Contractive methods for stiff differential equations. I. *BIT*, 18(4):457–474, 1978.

[99] O. Nevanlinna and W. Liniger. Contractive methods for stiff differential equations. II. *BIT*, 19(1):53–72, 1979.

[100] A. Nordsieck. On numerical integration of ordinary differential equations. *Mathematics of computation*, 16(77):22–49, 1962.

[101] O. Ø sterby. Five ways of reducing the Crank-Nicolson oscillations. *BIT*, 43(4):811–822, 2003.

[102] M. A. Olshanskii and L. G. Rebholz. Longer time accuracy for incompressible Navier–Stokes simulations with the EMAC formulation. *Computer Methods in Applied Mechanics and Engineering*, 372:113369, 2020.

[103] O. Oyekole, C. Trenchea, and M. Bukac. A second-order in time approximation of fluid-structure interaction problem. *SIAM Journal on Numerical Analysis*, 56(1):590–613, 2018.

[104] Y. Qin and Y. Hou. Optimal error estimates of a decoupled scheme based on two-grid finite element for mixed Navier-Stokes/Darcy model. *Acta Mathematica Scientia*, 38(4):1361–1369, 2018.

[105] Y. Qin and Y. Hou. The time filter for the non-stationary coupled Stokes/Darcy model. *Applied Numerical Mathematics*, 146:260–275, 2019.

[106] Y. Qin, Y. Hou, P. Huang, and Y. Wang. Numerical analysis of two grad–div stabilization methods for the time-dependent Stokes/Darcy model. *Computers & Mathematics with Applications*, 79(3):817–832, 2020.

[107] Y. Qin, Y. Hou, W. Pei, and J. Li. A variable time-stepping algorithm for the unsteady Stokes/Darcy model. *Journal of Computational and Applied Mathematics*, page 113521, 2021.

[108] B. Riviére. Analysis of a discontinuous finite element method for the coupled Stokes and Darcy problems. *Journal of Scientific Computing*, 22(1-3):479–500, 2005.

[109] A. Robert. The integration of a spectral model of the atmosphere by the implicit method. In *Proc. WMO/IUGG Symposium on NWP, Tokyo, Japan Meteorological Agency*, volume 7, pages 19–24, 1969.

[110] A. Seboldt and M. Bukač. A non-iterative domain decomposition method for the interaction between a fluid and a thick structure. *Numerical Methods for Partial Differential Equations*, 37(4):2803–2832, 2021.

[111] L. F. Shampine and M. K. Gordon. *Computer solution of ordinary differential equations*. W. H. Freeman and Company, San Francisco, California, 1975. The initial value problem.

[112] L. Shan and H. Zheng. Partitioned time stepping method for fully evolutionary Stokes–Darcy flow with Beavers–Joseph interface conditions. *SIAM Journal on Numerical Analysis*, 51(2):813–839, 2013.

[113] L. Shan, H. Zheng, and W. J. Layton. A decoupling method with different subdomain time steps for the nonstationary Stokes–Darcy model. *Numerical Methods for Partial Differential Equations*, 29(2):549–583, 2013.

[114] J. C. Simo, F. Armero, and C. A. Taylor. Stable and time-dissipative finite element methods for the incompressible Navier-Stokes equations in advection dominated flows. *International Journal for Numerical Methods in Engineering*, 38(9):1475–1506, 1995.

[115] R. D. Skeel. Equivalent forms of multistep formulas. *Mathematics of Computation*, 33(148):1229–1250, 1979.

[116] G. Söderlind. Digital filters in adaptive time-stepping. *ACM Transactions on Mathematical Software (TOMS)*, 29(1):1–26, 2003.

[117] G. Söderlind, I. Fekete, and I. Farag. On the zero-stability of multistep methods on smooth nonuniform grids. *BIT Numerical Mathematics*, 04 2018.

[118] G. Söderlind and L. Wang. Adaptive time-stepping and computational stability. *Journal of Computational and Applied Mathematics*, 185(2):225–243, 2006.

[119] H. J. Stetter. *Analysis of discretization methods for ordinary differential equations*. Springer-Verlag, New York-Heidelberg, 1973. Springer Tracts in Natural Philosophy, Vol. 23.

[120] M. Sussman. A stability example. Technical report, University of Pittsburgh, 2010.

[121] G. I. Taylor and A. E. Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 158(895):499–521, 1937.

[122] R. Temam. *Navier–Stokes Equations and Nonlinear Functional Analysis*. SIAM, 1995.

[123] G. Teschl. *Ordinary differential equations and dynamical systems*, volume 140. American Mathematical Soc., 2012.

[124] C. Trenchea. Second-order unconditionally stable ImEx schemes: implicit for local effects and explicit for nonlocal effects. *ROMAI Journal*, 12(1):163–178, 2016.

[125] C. Trenchea. Partitioned conservative, variable step, second-order method for magneto-hydrodynamics in Elsässer variables. *ROMAI Journal*, 15(2):117–137, 2019.

[126] D. S. Watanabe and Q. M. Sheikh. One-leg formulas for stiff ordinary differential equations. *SIAM journal on scientific and statistical computing*, 5(2):489–496, 1984.

[127] P. D. Williams. Achieving seventh-order amplitude accuracy in leapfrog integrations. *Monthly Weather Review*, 141(9):3037–3051, 2013.

[128] H. Xu, D. Baroli, F. Di Massimo, A. Quaini, and A. Veneziani. Backflow stabilization by deconvolution-based large eddy simulation modeling. *Journal of Computational Physics*, 404:109103, 2020.

[129] Y. Zeng, P. Huang, and Y. He. A time filter method for solving the double-diffusive natural convection model. *Computers & Fluids*, page 105265, 2021.

[130] L. Zuo and G. Du. A multi-grid technique for coupling fluid flow with porous media flow. *Computers & Mathematics with Applications*, 75(11):4012–4021, 2018.

[131] L. Zuo and G. Du. A parallel two-grid linearized method for the coupled Navier-Stokes-Darcy problem. *Numerical Algorithms*, 77(1):151–165, 2018.

[132] L. Zuo and Y. Hou. A decoupling two-grid algorithm for the mixed Stokes-Darcy model with the Beavers-Joseph interface condition. *Numerical Methods for Partial Differential Equations*, 30(3):1066–1082, 2014.

[133] L. Zuo and Y. Hou. Numerical analysis for the mixed Navier–Stokes and Darcy problem with the Beavers–Joseph interface condition. *Numerical Methods for Partial Differential Equations*, 31(4):1009–1030, 2015.