Preprint accepted in Journal of Computational Physics (originally submitted in August, 2018).
Download the journal version at https://doi.org/10.1016/j.jcp.2019.04.070

On Large Start-Up Error of BDF2

Hiroaki Nishikawa*
National Institute of Aerospace, Hampton, VA 23666, USA

May 5, 2019

Abstract

The second-order backward-difference (BDF2) method requires solutions at two previous time steps and is often started with the first-order backward-difference (BDF1) method. The initial time step is typically taken to be small in order to minimize the effect of the first-order error. However, contrary to expectations, the solution computed in this way generates a very large error for stiff problems and the error grows as the initial time step is further reduced. As shown in this note, the problem is originated from the fact that the variable-coefficient BDF2 method combined with any first- or higher-order one-step method is asymptotically equivalent to the trapezoidal method and therefore not L-stable. Based on the study presented, it is strongly recommended that BDF2 method be started with a self-starting second-order implicit Runge-Kutta method, which guarantees second-order accuracy and L-stability at no additional cost.

1 Introduction

The second-order backward difference (BDF2) method is widely used for solving time-dependent problems in various practical applications [1, 2, 3, 4, 5, 6, 7]. It is a very efficient and robust method since it is second-order accurate and L-stable with only a single nonlinear solve per time step [8, 9]. For a general ordinary differential equation of the form:

$$\frac{du}{dt} = f(u,t), \tag{1}$$

the BDF2 method is given by

$$\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta t} = f(u^{n+1}, t^{n+1}), \tag{2}$$

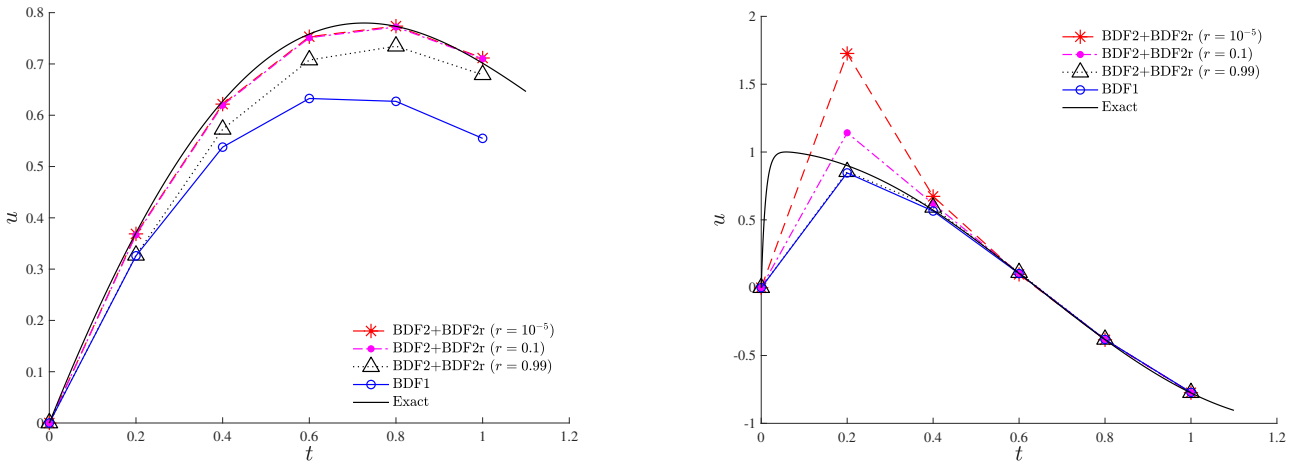
where Δt is a constant time step, n indicates the time level ($n = 1, 2, \cdots$), and t^n and u^n denote the physical time and the solution at the time level n, respectively. The BDF2 method is, however, not self-starting: the solution at the first time step needs to be obtained by some other means before the BDF2 method can be applied. Among various approaches, we consider the use of the first-order backward difference (BDF1) method, which is indeed very popular for its simplicity [10, 11]. In doing so, one would wish to minimize the effect of the first-order error by using a very small time step for BDF1. This strategy can be implemented with the variable-coefficient BDF2 method as follows:

$$\frac{u^* - u^0}{\Delta t^*} = f(u^*, t^*), \tag{3}$$

$$\frac{1}{\Delta t'}\frac{\Delta t^* + 2\Delta t'}{\Delta t^* + \Delta t'}u^1 - \left(\frac{1}{\Delta t^*} + \frac{1}{\Delta t'}\right)u^* + \frac{1}{\Delta t^*}\frac{\Delta t'}{\Delta t^* + \Delta t'}u^0 = f(u^1, t^1), \tag{4}$$

where $\Delta t = \Delta t^* + \Delta t'$, u^0 is a given initial solution, u^* is an intermediate solution computed by BDF1, i.e., Equation (3), at $t = t^* = \Delta t^*$, u^1 is the solution we seek at $t = \Delta t$. Equation (4) is the variable-coefficient BDF2 method [9] that maintains second-order accuracy, by varying the coefficients, for two different previous time-step sizes: Δt^* and $\Delta t'$; it reduces to the constant-coefficient form (2) when $\Delta t^* = \Delta t' = \Delta t$. It seems reasonable to expect better accuracy with a smaller Δt^* as it will reduce the effect of the first-order error. However, it actually generates a larger error for stiff problems. To illustrate this strange behavior, we applied

*Associate Research Fellow (hiro@nianet.org), 100 Exploration Way, Hampton, VA 23666 USA,



(a) Non-stiff problem

(b) Stiff problem

Figure 1: The reasons that Δt^* should be small (Left) and Δt^* should be large (Right).

the method to the following model problem (which is a modified version of the model problem (1.1) in Ref.[9]) with $\Delta t = 0.2$:

$$\frac{du}{dt} = -K\left\{u - \cos(2.5t)\right\} + 1.1\exp(-0.1t), \quad u^0 = 0, \tag{5}$$

where K is a positive constant. Figure 1(a) shows the results for a non-stiff case with $K = 1$, where the above start-up method is denoted by BDF2r with $r = \Delta t^*/\Delta t'$. As we expected, a better solution is obtained with a smaller value of Δt^* (i.e., smaller r). Note that the case $r = 0.99$ is nearly equivalent to computing the first solution entirely by the BDF1 method; the numerical solution is not very accurate due to the first-order error committed at the first step. This is the reason that we wish to take Δt^* to be very small. In contrast, for a moderately stiff case with $K = 100$, the method generates a larger error at $t = \Delta t$ with a smaller Δt^* as can be clearly seen in Figure 1(b). It is noted that the problem is not related to the instability of the variable-coefficient BDF2 method discussed in Ref.[12], which is associated with a constant step size ratio while here we are concerned with the combination with the BDF1 method and only at the first step. Inaccuracy of solution at the first time level presents a concern about non-physical solutions (e.g., negative pressure) or inaccurate evaluations of a functional defined as an integral over the entire time. The objective of this note is to show that the problem stems from the asymptotic equivalence between the variable-coefficient BDF2 combined with BDF1 and the non-L-stable trapezoidal method.

2 BDF2 with BDF1 is Not L-Stable

Consider a model problem:

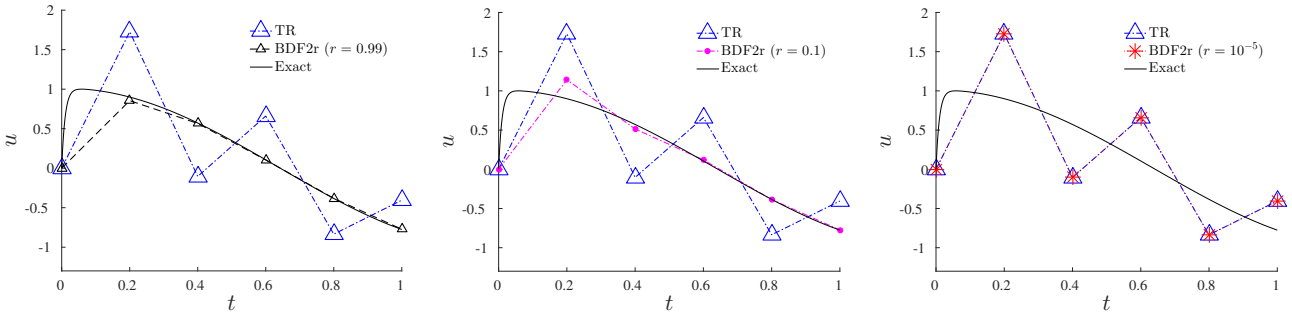
$$\frac{du}{dt} = \lambda u, \tag{6}$$

where λ is a constant. The BDF1 start-up method applied to the model problem is given by

$$\frac{u^* - u^0}{\Delta t^*} = \lambda u^*, \tag{7}$$

$$\frac{1}{\Delta t'}\frac{\Delta t^* + 2\Delta t'}{\Delta t^* + \Delta t'}u^1 - \left(\frac{1}{\Delta t^*} + \frac{1}{\Delta t'}\right)u^* + \frac{1}{\Delta t^*}\frac{\Delta t'}{\Delta t^* + \Delta t'}u^0 = \lambda u^1. \tag{8}$$

It seems little known that this start-up strategy is not L-stable when Δt^* is small. A method is called L-stable if the stability polynomial $P(\lambda\Delta t)$, defined as in $u^{n+1} = P(\lambda\Delta t)u^n$ when it is applied to the model equation (6), has the following two properties: $|P| < 1$ for $Re(\lambda\Delta t) < 0$ (i.e., A-stable) and $|P| \to 0$ as $|\lambda\Delta t| \to \infty$, i.e., high-frequency modes are damped rapidly. For stiff problems, it is well known that A-stability is not sufficient because it can develop large oscillations as typically illustrated with the trapezoidal method (see Ref.[9] or results below), which is A-stable but not L-stable. We prove that the above start-up method is not L-stable by



(a) $r = 0.99$ (b) $r = 0.1$ (c) $r = 10^{-5}$

Figure 2: The BDF1-start-up method is applied at every time step to demonstrate the equivalence with the trapezoidal method (TR) in the limit $r \to 0$.

showing that the above start-up method is asymptotically equivalent to the trapezoidal method. Let us write it as a general two-stage method from n to $n+1$:

$$\frac{u^* - u^n}{\Delta t^*} = \lambda u^*, \tag{9}$$

$$\frac{au^{n+1} + bu^* + cu^n}{\Delta t'} = \lambda u^{n+1}, \tag{10}$$

where, again with $r = \Delta t^*/\Delta t'$,

$$a = \frac{r+2}{r+1}, \quad b = -\frac{1+r}{r}, \quad c = \frac{1}{r(r+1)}. \tag{11}$$

By eliminating u^* from Equation (10), we obtain

$$au^{n+1} + \frac{b}{1 - \Delta t^* \lambda} u^n + cu^n = \Delta t' \lambda u^{n+1}, \tag{12}$$

or

$$(a - \Delta t'\lambda)u^{n+1} + \left(\frac{b}{1 - r\Delta t'\lambda} + c\right) u^n = 0. \tag{13}$$

For a small r, we can expand $\frac{1}{1-r\Delta t'\lambda}$ to get

$$(a - \Delta t'\lambda)u^{n+1} + \left[\, b(1 + r\Delta t'\lambda) + c \,\right] u^n = 0, \tag{14}$$

which becomes by $b = -(a + c)$,

$$(a - \Delta t'\lambda)u^{n+1} - \left\{a + (ar + cr)\Delta t'\lambda\right\} u^n = 0. \tag{15}$$

In the limit $r \to 0$, we have $a \to 2$, $cr = 1/(r+1) \to 1$, and $\Delta t' \to \Delta t$, and therefore

$$(2 - \Delta t\lambda)u^{n+1} - (2 + \Delta t\lambda)\, u^n = 0, \tag{16}$$

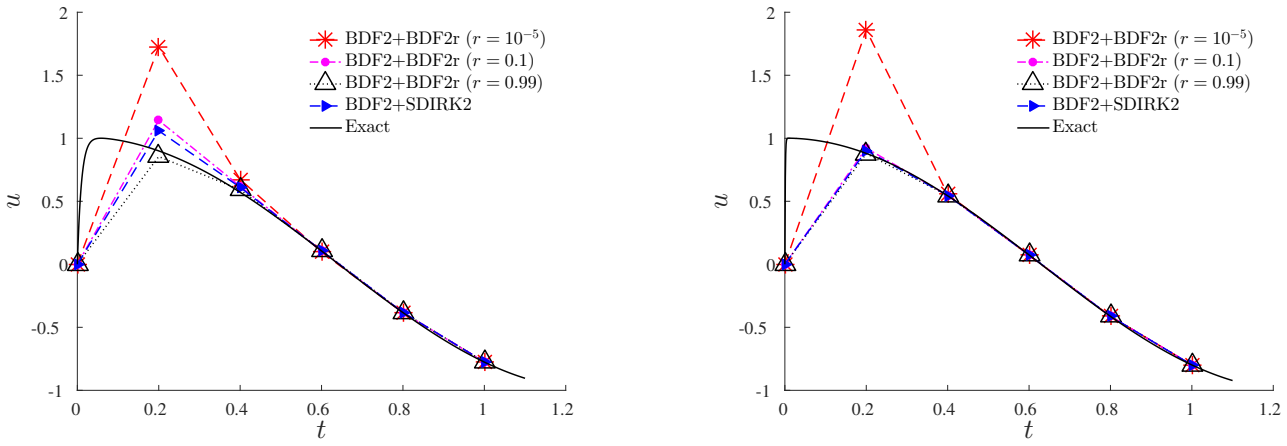
which is the trapezoidal method:

$$\left(1 - \frac{1}{2}\Delta t\lambda\right) u^{n+1} - \left(1 + \frac{1}{2}\Delta t\lambda\right) u^n = 0, \tag{17}$$

and not L-stable:

$$\left|\frac{u^{n+1}}{u^n}\right| = \left|\frac{1 + \frac{1}{2}\Delta t\lambda}{1 - \frac{1}{2}\Delta t\lambda}\right| \to 1 \quad \text{as} \quad \Delta t\lambda \to \infty. \tag{18}$$

To confirm the analysis, we solved the test problem (5) with $K = 100$ by the general two-stage method (9) and (10) applied at every time step. Solutions obtained with $r = 0.99$, 0.1, and 10^{-5} are shown in Figure 2. The method approaches from the BDF1 method to the trapezoidal method as r is reduced from 0.99 to 10^{-5}.



(a) $K = 100$ (b) $K = 2000$

Figure 3: Comparison of star-up methods: SDIRK2 and BDF1 with various r for a stiff problem.

The overlapping solutions in Figure 2(c) confirm the asymptotic equivalence between the BDF2 with the BDF1 start-up and the trapezoidal method. Both exhibit a typical 'ringing' instability due to the lack of L-stability, which causes, for example, an adaptive step-size control to stall [13]. These results imply that the overshoot observed in the example calculation is originated from the non-L-stable trapezoidal method. As one might have noticed, Equation (14) indicates that the BDF2 method is equivalent to the trapezoidal method for any first- or higher-order one-step method because all such will give $u^* \to (1 + r\Delta t'\lambda)u^n$ as $r \to 0$ (even an exact solution at $t = r\Delta t'$).

3 Use SDIRK Instead

The above analysis suggests that the BDF2 method with any start-up method will suffer from the same problem and the initial time step Δt^* should be large enough to avoid the problem. A practical compromise might be $r = 0.1$, which yields reasonable accuracy for both non-stiff and stiff problems, at least for the test case considered here. A more sensible approach is to employ a self-starting second-order L-stable method such as the second-order singly-diagonal implicit Runge-Kutta (SDIRK) method [14, 15]:

$$u^{n+\alpha} = u^n + \alpha\Delta t f(u^{n+\alpha}, t^n + \alpha\Delta t), \tag{19}$$

$$u^{n+1} = u^n + (1-\alpha)\Delta t f(u^{n+\alpha}, t^n + \alpha\Delta t) + \alpha\Delta t f(u^{n+1}, t^n + \Delta t), \tag{20}$$

where $\alpha = (2-\sqrt{2})/2$. In general, two nonlinear equations need to be solved, but they can be solved sequentially: first solve Equation (19) for $u^{n+\alpha}$, and then solve Equation (20) for u^{n+1}. Therefore, the cost is comparable to the BDF2 method with the BDF1 start-up, which also requires two nonlinear solves to get u^1. We implemented the SDIRK method as a method for generating the solution at the first time step (the BDF2 is used afterwards). Figures 3 and 4 show that it produces reasonably accurate solutions, as expected, to the stiff and non-stiff problems considered in Section 1 and a highly stiff case with $K = 2000$.

As another example, we consider the following system is taken from Ref.[8]:

$$du(t)/dt = -2u + v + 2\sin(t), \tag{21}$$

$$dv(t)/dt = 998u - 999v + 999(\cos(t) - \sin(t)), \tag{22}$$

whose exact solution is given by

$$u(t) = \kappa_1 \exp(-t) + \kappa_2 \exp(-1000t) + \sin(t), \quad v(t) = \kappa_1 \exp(-t) - 998\kappa_2 \exp(-1000t) + \cos(t), \tag{23}$$

$$\kappa_1 = u(0) - \kappa 2, \quad \kappa_2 = (u(0) - v(0) + 1)/999. \tag{24}$$

The solution exhibits a rapid transient for the initial condition: $u(0) = 2.0$ and $v(0) = 3.999$, but the stiff component vanishes for $u(0) = 2.0$ and $v(0) = 3.0$. Numerical results obtained by the set of methods as in the previous example are shown in Figure 5, where only the second component of the solution v is plotted. Observe that the BDF1 start-up with a very small time step $r = 10^{-5}$ generates a large error even in the non-stiff case.

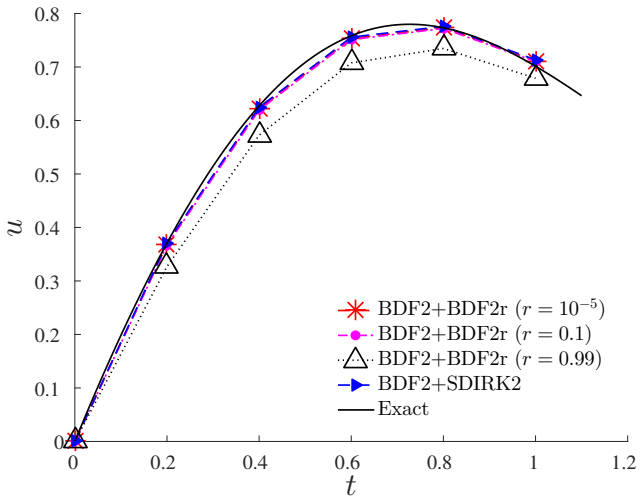
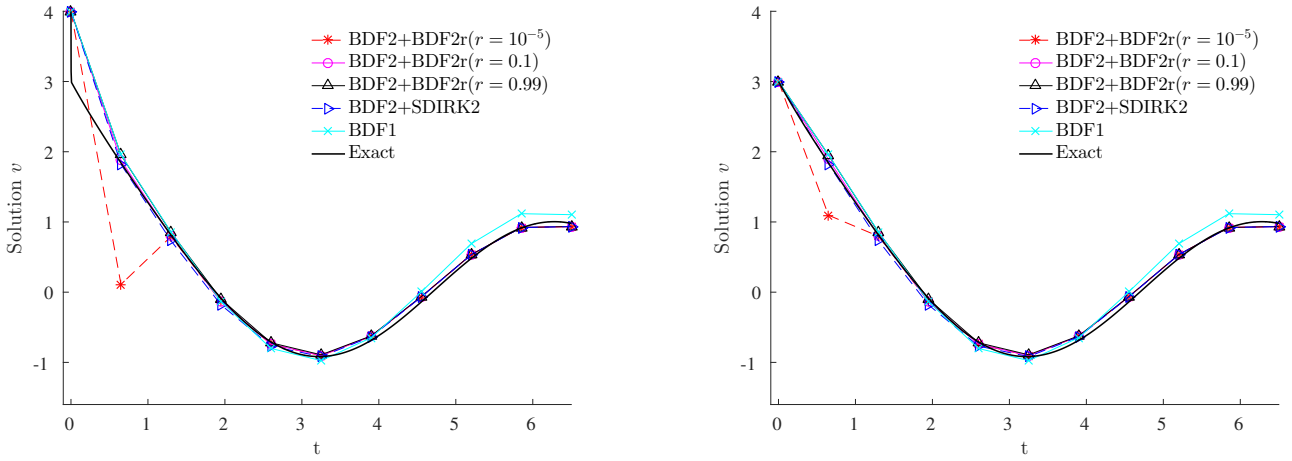


Figure 4: Comparison of star-up methods: SDIRK2 and BDF1 with various r for a non-stiff problem.



(a) $u(0) = 2.0$ and $v(0) = 3.999$.

(b) $u(0) = 2.0$ and $v(0) = 3.0$

Figure 5: Comparison of star-up methods: SDIRK2 and BDF1 with various r for stiff and non-stiff solutions.

4 Conclusions

As shown, the variable-coefficient BDF2 method started with the BDF1 (or any first- and higher-order) method with a small initial time step is asymptotically equivalent to the trapezoidal method and therefore not L-stable. As a consequence, reducing the initial time step results, contrary to expectations, in a very large error for stiff problems. It is strongly recommended that the second-order SDRIK method be employed to obtain the solution at the first time step (e.g., as in Ref.[16], for example) because it guarantees second-order accuracy and L-stability at no additional cost.

Acknowledgments

The author gratefully acknowledges support from Software CRADLE.

References

[1] Michael Hauth and Olaf Etzmuss. A High Performance Solver for the Animation of Deformable Objects using Advanced Numerical Methods. *Computer Graphics Forum*, 2001.

[2] O. Scherf and B. Simeon. Differential-algebraic equations in elasto-viscoplasticity. In K. Hutter and H. Baaser, editors, *Deformation and Failure in Metallic Materials. Lecture Notes in Applied and Computational Mechanics*, volume 10, pages 799–804. Springer, 2003.

[3] E. Brocke, M. Djurfeldt, U. S. Bhalla, J. H. Kotaleski, and M. Hanke. Multirate method for co-simulation of electrical-chemical systems in multiscale modeling. *Journal of computational neuroscience*, 42(3):245–256, 2017.

[4] J. Alikhani, B. Shoghli, U. K. Bhowmik, and A. Massoudieh. An adaptive time-step backward differentiation algorithm to solve stiff ordinary differential equations: Application to solve activated sludge models. *American Journal of Computational Mathematic*, 6:298–312, 2016.

[5] A. Sandu, J. G. Verwert an d M. Van Loons, G. R. Carmichael, F. A. Potrat, D. Dabdubii, and J. H. Seinfeld. Benchmarking stiff ode solvers for atmospheric chemistry problems-I. implicit vs explicit. *Atmospheric Environment*, 31(19):3151–3166, 1997.

[6] Veer N. Vatsa, Mark H. Carpenter, and David P. Lockard. Re-evaluation of an optimized second order backward difference (BDF2OPT) scheme for unsteady flow applications. In *48th AIAA Aerospace Sciences Meeting*, AIAA Paper 2010-0122, Orlando, FL, 2010.

[7] Y. Nakashima, N. Watanabe, and H. Nishikawa. Development of an effective implicit solver for general-purpose unstructured CFD software. In *The 28th Computational Fluid Dynamics Symposium*, C08-1, Tokyo, Japan, 2014.

[8] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems*. Wiley, Chichester, UK, 1991.

[9] Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II: Nonstiff Problems*, volume 14 of Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2nd edition, 2010.

[10] Adrian Sandu. Positive numerical integration methods for chemical kinetic systems. *J. Comput. Phys.*, 170:589–602, 2001.

[11] J. Rosama, P. K. Jimack, and A. Mullis. A fully implicit, fully adaptive time and space discretisation method for phase-field simulation of binary alloy solidification. *J. Comput. Phys.*, 225:1271–1287, 2007.

[12] Etienne Emmrich. Stability and error of the variable two-step BDF for semilinear parabolic problems. *J. Appl. Math. & Computing*, 19:33–55, 2005.

[13] J. Alex Lee, Jaewook Nam, and Matteo Pasquali. A new stabilization of adaptive step trapezoid rule based on finite difference interrupts. *SIAM J. Sci. Comput.*, 37(2):A725–A746, 2015.

[14] Roger Alexander. Diagonally implicit runge-kutta methods for stiff O.D.E.'s. *SIAM J. Numer. Anal.*, 14(6):1006–1021, 1977.

[15] Christopher A. Kennedy and Mark H. Carpenter. Diagonally implicit Runge-Kutta methods for ordinary differential equations. A review. *NASA/TM 2016-219173*, 2016.

[16] M. Tabesh and D. W. Aingg. Efficient implicit time-marching methods using a Newton-Krylov algorithm. AIAA Paper 2009-164, Orlando, Florida, January 2009.