INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 19, Number 2-3, Pages 299–314

A STRESS TEST FOR THE MIDPOINT TIME-STEPPING METHOD

JOHN BURKARDT, WENLONG PEI, AND CATALIN TRENCHEA

This paper is dedicated to our mentor and friend Max Gunzburger

Abstract. The midpoint method can be implemented as a sequence of Backward Euler and Forward Euler solves with half time steps, allowing for improved performance of existing solvers for PDEs. We highlight the advantages of this refactorization by considering some specifics of implementation, conservation, error estimation, adaptivity, stability, and performance on several test problems.

Key words. Midpoint method, local error estimation, time step adaptivity, conservation, B-stable.

1. Introduction

In the paper [9], the midpoint method was reformulated in an unusual way, and numerous properties and advantages of the method and its implementation were claimed. Here, we propose to highlight these features by considering some specifics of implementation, conservation, error estimation, adaptivity, stability, and performance on several test problems. The refactorized method was successfully applied to partial differential equations and partitioning algorithms for fluid-structure interaction and magnetohydrodynamics [6–8, 52], and the idea was further developed for multistep methods in [34, 35].

2. The midpoint method and its "relatives"

We wish to estimate some quantity $y(t)$, for which we have an initial value y_0 at time t_0, and an evolution equation:

$$y'(t) = f(t, y(t)).$$

We shall produce estimates y_n at a discrete sequence of times $\{t_n\}_{n\geq 0}$, using stepsizes $\tau_n = t_{n+1} - t_n$. For convenience, we define $t_{n+1/2} = t_n + \frac{1}{2}\tau_n$.

The (implicit) midpoint method for this problem can now be defined by:

$$\begin{aligned}\frac{y_{n+1} - y_n}{\tau_n} &= f\left(t_{n+1/2}, \frac{y_{n+1} + y_n}{2}\right)\\ &= f\left(t_{n+1/2}, y_n + \frac{1}{2}\tau_n \frac{y_{n+1} - y_n}{\tau_n}\right).\end{aligned}$$

Unless the right hand side function $f(\cdot,\cdot)$ is linear in y, each time we want to take a step by applying this formula, we must solve an implicit nonlinear equation for the unknown value y_{n+1}. This implicit equation solution cost is part of the overhead of the midpoint method. This cost varies depending on the nonlinearity of $f(\cdot,\cdot)$ (associated with the problem), on the stepsize being used, (associated

Received by the editors October 7, 2021 and, in revised form, November 25, 2021.
2000 *Mathematics Subject Classification.* 34D20, 35L65, 65C20, 65D30, 65L07, 65L20, 65M12.

with the midpoint method), and on the robustness of the implicit equation solver (depending on the underlying nonlinear solver employed).

It should be noted that a number of ODE solution methods are loosely termed "midpoint methods". To avoid confusion, it may be helpful to first name several of these related methods and indicate their distinguishing features.

Title	**Formula for** $\frac{y_{n+1}-y_n}{\tau_n}$
Implicit midpoint:	$f(t_{n+1/2}, 1/2\,(y_n + y_{n+1}))$
Explicit midpoint:	$f(t_{n+1/2}, 1/2\,(y_n + y_n + \tau_n f(t_n, y_n)))$
Implicit trapezoidal:	$1/2\,(f(t_n, y_n) + f(t_{n+1}, y_{n+1}))$
Explicit trapezoidal:	$1/2\,(f(t_n, y_n) + f(t_{n+1}, y_n + \tau_n f(t_n, y_n)))$

TABLE 1. Four related one-step ODE methods.

It should be clear now that midpoint methods evaluate the right hand side at the midpoint of the interval $[t_n, t_{n+1}]$, while trapezoidal methods average values at the endpoints. Implicit methods invoke the unknown solution y_{n+1}, while explicit methods use an estimate, such as the Euler approximation, instead.

If we move to partial differential equations, a very popular procedure is known as the Crank-Nicolson [1] method, which involves both space and time discretization. For time stepping, it is possible to implement the Crank-Nicolson method using any one of the four above methods. Interestingly, in the original paper [12], it seems that the implicit midpoint method is described.

From now on in this discussion, the expression "midpoint method" will be used exclusively to refer to the implicit midpoint method.

3. Implementation

The midpoint method is a single step method; that is, the approximation of y_{n+1} at time t_{n+1} depends on the current values t_n and y_n, but not on any previous data. For the moment, we will assume that an appropriate stepsize $\tau_n = t_{n+1} - t_n$ has been specified, so that we only need to address the implicit equation that defines y_{n+1}.

While adaptivity and variable step sizes are a vital feature of modern ODE solvers, we will defer discussion of those matters (see e.g. [8, 9] and the references

[1]There is some confusion on what the Crank-Nicolson method actually is. For example, the wikipedia page (November 14, 2021) claims that Crank-Nicolson is based on the trapezoidal rule, and also Hundsdorfer and Verwer [28, page 125] state that "In the classic numerical PDE literature, backward Euler and the trapezoidal rule are also known under the names Laasonen scheme [31] and Crank-Nicolson [12] scheme, respectively". The Crank-Nicolson appellative for the trapezoidal rule is used in Ascher [2, page 41], Canuto, Hussaini, Quarteroni and Zhang's book [10, page 521], Hairer, Lubich and Wanner [23, page 28], Leveque [38, pages 121, 185], Quarteroni, Sacco and Saleri [44, page 483], Quarteroni, Valli [45, page 149], Volker [29, page 394]. Heywood and Rannacher [25, page 355] and Gunzburger [22, page 131] call trapezoidal rule as Crank-Nicolson, and then use the midpoint rule. In [33, page 162], Layton also refers to the midpoint rule by the trapezoidal cognomen. On the other hand, Glowinski [21, page 267], Hoffman and Johnson [27, page 216], Kalnay [30, page 83], Layton and Rebholz [36, page 137], refer to the midpoint rule as the Crank-Nicolson method. Finally, it is worth mentioning that the method which John Crank and Phyllis Nicolson literally used in their 1947 paper [12] on numerical solutions of a nonlinear partial differential equation for heat flow is the midpoint rule.