Stability of implicit multiderivative deferred correction methods

Peer-reviewed author version

ZEIFANG, Jonas; SCHUETZ, Jochen & Seal, David C (2022) Stability of implicit multiderivative deferred correction methods. In: BIT NUMERICAL MATHEMATICS,.

DOI: 10.1007/s10543-022-00919-x
Handle: http://hdl.handle.net/1942/37276

BIT manuscript No.
(will be inserted by the editor)

Stability of Implicit Multiderivative Deferred Correction Methods

Jonas Zeifang · **Jochen Schütz** · **David C. Seal**

Received: date / Accepted: date

Abstract Very recently, a novel class of parallelizable high-order time discretization schemes has been introduced in [Schütz J., Seal D., Zeifang J., Parallel-in-time high-order multiderivative IMEX solvers, arXiv preprint arXiv:2101.07846.]. In this current work, we analyze the stability properties of those schemes and introduce a small but effective modification which only necessitates minor modifications of existing implementations. It is shown how this modification leads to A(α)-stable schemes with α being close to 90°. Numerical examples illustrate an additional favorable influence of this modification on the accuracy of those schemes.

Keywords Multiderivative Schemes · Stability Analysis · Parallel-in-Time

Mathematics Subject Classification (2000) 65L20 · 65L04

1 Introduction

In this work, we consider algorithms to solve linear and nonlinear large differential equations of the form

$$w'(t) = \Phi(w), \quad t \in (0, T_{\text{end}}) \quad \text{with} \quad w(0) = w_0. \tag{1.1}$$

J. Zeifang
Faculty of Sciences, Universiteit Hasselt,
Agoralaan Gebouw D, BE-3590 Diepenbeek, Belgium
E-mail: jonas.zeifang@uhasselt.be

J. Schütz
Faculty of Sciences, Universiteit Hasselt,
Agoralaan Gebouw D, BE-3590 Diepenbeek, Belgium
E-mail: jochen.schuetz@uhasselt.be

D. C. Seal
United States Naval Academy, Department of Mathematics,
572C Holloway Road, Annapolis, MD 21402, USA
E-mail: seal@usna.edu

Such equations are e.g. obtained from the discretization of partial differential equations (PDEs) and there exist a variety of different approaches to solve this type of equations, see e.g. [9,10] for an overview. Methods that do not only consider $\Phi(w)$ but also higher temporal derivatives of the solution w, i.e. $\dot{\Phi}(w),\ \ddot{\Phi}(w),\ldots$ are called multi-derivative methods, see e.g. the early works [16] and [18]. The multi-derivative paradigm has been applied to the idea of general linear methods, see [3]. Runge-Kutta methods, which are a subclass of general linear methods, have also been extended to deal with multiple derivatives, see the non-exhaustive list [1,8,4,5,11,17,15] and the references therein. In the following, we consider two-derivative schemes, i.e. both $\Phi(w)$ and additionally

$$\dot{\Phi}(w) := \Phi'(w)\Phi(w)$$

are used in an algorithm. Here, $\Phi'(w)$ refers to the Jacobian of $\Phi(w)$ with respect to w, i.e., $\Phi'(w) \equiv \partial\Phi/\partial w$.

Recently, a predictor-corrector formulation for an implicit-explicit (IMEX) two-derivative Runge-Kutta method of fourth order has been introduced in [13]. This formulation avoids the necessity of solving very large implicit systems as it consists of a sequence of two-derivative Euler solves for the implicit system. The method can be seen from two perspectives: either as a reformulation of a two-derivative Runge-Kutta method into a predictor-corrector formulation to enable an efficient implementation, or as the extension of spectral deferred correction (SDC) methods, see [7,12], to incorporate two-derivatives in the quadrature formula. This idea has been transferred to take even more derivatives into account [6]. Other extensions of this idea include a combination with higher order accurate quadrature formulas by adding additional quadrature nodes [14] and the substitution of the predictor with a higher order accurate one [19]. In [14], it has also been shown how pipelining can be introduced to allow for a parallel-in-time execution.

In this work, we evaluate the stability properties of this class of schemes and introduce a simple but effective modification of the algorithm, which only necessitates minor adjustments to already existing implementations. Additionally to having a favorable influence on the accuracy of the scheme, the modification improves the stability properties significantly.

The paper is organized as follows: In Sec. 2 the predictor-corrector schemes from [13] and [14] are recalled and cast into a unified formulation. Following, the stability properties of the schemes are evaluated in Sec. 3 and the results of a stability optimization are reported in Sec. 3.2. The influence of this optimization on the accuracy of the methods is evaluated in Sec. 4. Finally, a summary and an outlook are given in Sec. 5.

2 Hermite-Birkhoff Predictor-Corrector scheme

In this section, we briefly recall the serial algorithm from [13] and its low-storage parallel-in-time modification presented in [14]. We directly introduce a small modification which will be used as a tuning parameter to improve the stability in the remainder of this paper. Both schemes are cast in a unified formulation and rewritten to make them accessible to a stability analysis. We introduce the notation $(\bullet) = (S)$ for the serial and $(\bullet) = (P)$ for the parallel algorithm. The algorithms describe a **p**redictor-**c**orrector approach with $k_{\max}$ correction steps to approximate a two-derivative **H**ermite-**B**irkhoff Runge-Kutta method of order q and are therefore labeled as HBPC$(q, k_{\max})$ and Parallel-HBPC$(q, k_{\max})$, respectively.

2.1 The HBPC$(q, k_{\max})$ Procedure

Algorithm 1 ((Parallel-)HBPC$(q, k_{\max})$ [13,14]) *First, the initial conditions are filled with $w^{-1,[k],s} := w_0$. To advance the solution to Eq. (1.1) from time level t^n to time level t^{n+1}, fill the values $w^{n,[0],l}$ using a second-order implicit Taylor method:*

1. ***Predict.*** *Solve the following expression for $w^{n,[0],l}$ and $1 \le l \le s$:*
$$w^{n,[0],l} := w^{n-1,[k^*],s} + c_l \Delta t \Phi^{n,[0],l} - \frac{(c_l \Delta t)^2}{2} \dot{\Phi}^{n,[0],l}, \tag{2.1}$$
where $k^* = k_{\max}$ for the serial and $k^* = 1$ for the parallel-in-time algorithm. Subsequently:
2. ***Correct.*** *Solve the following for $w^{n,[k+1],l}$, for each $2 \le l \le s$ and each $0 \le k < k_{\max}$:*
$$\begin{aligned} w^{n,[k+1],1} &:= w^{n-1,[k^{**}],s}, \\ w^{n,[k+1],l} &:= w^{n-1,[k^{**}],s} + \Delta t \theta_1 \left(\Phi^{n,[k+1],l} - \Phi^{n,[k],l} \right) \\ &\qquad - \frac{\Delta t^2}{2} \theta_2 \left(\dot{\Phi}^{n,[k+1],l} - \dot{\Phi}^{n,[k],l} \right) + \mathscr{I}_l^{(\bullet)}, \end{aligned} \tag{2.2}$$
where $k^{**} = k_{\max}$ and $k^{**} = \min(k+2, k_{\max})$ for the serial and parallel-in-time algorithm, respectively.
3. ***Update.*** *Set $w^{n+1} := w^{n,[k_{\max}],s}$.*

The quadrature formulas $\mathscr{I}_l^{(\bullet)}$ for every stage $1 \le l \le s$ are defined by

$$\begin{aligned} \mathscr{I}_l^{(S)} :=& \Delta t \sum_{j=1}^{s} B_{lj}^{(1)} \Phi^{n,[k],j} + \Delta t^2 \sum_{j=1}^{s} B_{lj}^{(2)} \dot{\Phi}^{n,[k],j}, \\ \mathscr{I}_l^{(P)} :=& \Delta t \left(\sum_{j=1}^{l-1} B_{lj}^{(1)} \Phi^{n,[k+1],j} + \sum_{j=l}^{s} B_{lj}^{(1)} \Phi^{n,[k],j} \right) \\ &+ \Delta t^2 \left(\sum_{j=1}^{l-1} B_{lj}^{(2)} \dot{\Phi}^{n,[k+1],j} + \sum_{j=l}^{s} B_{lj}^{(2)} \dot{\Phi}^{n,[k],j} \right), \end{aligned}$$

for the serial and the parallel algorithm, respectively. $B^{(1)}$, $B^{(2)}$ and c are the Butcher tableaux that define the limiting Hermite-Birkhoff Runge-Kutta scheme. In this work, we use the same quadrature rules as used in [14, Eqs. (2)-(4)]. The difference between the serial and the parallel-in-time low-storage method lies in the modified start value of the iterations to allow for pipelining and the modification of the quadrature rule to use already known values to reduce the memory footprint.

Remark 2.1 Note that we have directly made a modification of the original methods by adding the parameters θ_1 and θ_2, which equal one in the original algorithm. These variables will serve as tuning parameters to improve the stability properties of the algorithm in the remainder of this paper.

2.2 One Step Equivalent of HBPC$(q, k_{\max})$

To facilitate the stability analysis, we rewrite Alg. 1 in a one step fashion. Although one would not pursue this implementation in a practical application, this allows us to study the stability properties of the schemes in a straight forward manner. Following the ideas outlined in [14], we define the solution vector Y^{n+1} as

$$Y^{n+1} := \left(w^{n,[0],1}, \cdots, w^{n,[0],s}, w^{n,[1],1}, \cdots, w^{n,[1],s}, \cdots, w^{n,[k_{\max}],s}\right)^T \in \mathbb{R}^{s\cdot(k_{\max}+1)},$$

with $Y^0 := (w_0, \cdots, w_0)^T \in \mathbb{R}^{s\cdot(k_{\max}+1)}$.

Definition 2.1 For the serial and the parallel algorithm, matrices $A^{(\bullet)}$, $C_1^{(\bullet)}$, $C_2^{(\bullet)}$ and D in $\mathbb{R}^{s(k_{\max}+1)\times s(k_{\max}+1)}$ can be found such that

$$D \cdot Y^{n+1} = A^{(\bullet)} \cdot Y^n + \Delta t C_1^{(\bullet)} \cdot \Phi\left(Y^{n+1}\right) + \frac{\Delta t^2}{2} C_2^{(\bullet)} \cdot \dot{\Phi}\left(Y^{n+1}\right). \tag{2.3}$$

Note that this formulation is a direct extension of the original general linear method formulation from [2] to two derivatives. The only difference between the matrices used in Eq. (2.3) and the ones used in [14] lies in the coefficients θ, which have to be taken into account in $C_1^{(\bullet)}$ and $C_2^{(\bullet)}$. Let us first define the matrices $\mathfrak{E}$, $\mathfrak{I} \in \mathbb{R}^{s\times s}$ with

$$\mathfrak{E} := \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \quad \text{and} \quad \mathfrak{I} := \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Then, the matrices $A^{(\bullet)} \in \mathbb{R}^{s\cdot(k_{\max}+1)\times s\cdot(k_{\max}+1)}$ are given by

$$A^{(S)} = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & \mathfrak{E} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mathfrak{E} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mathfrak{E} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mathfrak{E} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mathfrak{E} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mathfrak{E} \end{pmatrix}, \quad \text{and} \quad A^{(P)} = \begin{pmatrix} 0 & \mathfrak{E} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \mathfrak{E} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \mathfrak{E} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \mathfrak{E} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mathfrak{E} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mathfrak{E} \end{pmatrix}. \tag{2.4}$$

For both, the serial and parallel scheme, D corresponds to the $s(k_{\max}+1) \times s(k_{\max}+1)$ unity matrix. The matrices $C_1^{(S)}, C_2^{(S)} \in \mathbb{R}^{s\cdot(k_{\max}+1)\times s\cdot(k_{\max}+1)}$ are given by

$$
\begin{aligned}
C_1^{(S)} &= \begin{pmatrix}
\operatorname{diag}(c) & 0 & 0 & \cdots & \cdots & 0 \\
B^{(1)} - \theta_1 \mathfrak{I} & \theta_1 \mathfrak{I} & 0 & \cdots & \cdots & 0 \\
0 & B^{(1)} - \theta_1 \mathfrak{I} & \theta_1 \mathfrak{I} & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & B^{(1)} - \theta_1 \mathfrak{I} & \theta_1 \mathfrak{I}
\end{pmatrix}, \\
C_2^{(S)} &= \begin{pmatrix}
-\operatorname{diag}(c^2) & 0 & 0 & \cdots & \cdots & 0 \\
2B^{(2)} + \theta_2 \mathfrak{I} & -\theta_2 \mathfrak{I} & 0 & \cdots & \cdots & 0 \\
0 & 2B^{(2)} + \theta_2 \mathfrak{I} & -\theta_2 \mathfrak{I} & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & 2B^{(2)} + \theta_2 \mathfrak{I} & -\theta_2 \mathfrak{I}
\end{pmatrix}.
\end{aligned}
\tag{2.5}
$$

To ensure the low-storage property of the parallel-in-time variant we have to decompose the $B^{(1)}$ and the $B^{(2)}$ matrix into the lower and upper triangular matrices $B_L^{(\cdot)}$ and $B_U^{(\cdot)}$ of the Runge-Kutta tables with $(\cdot) = (1), (2)$, respectively. I.e.

$$
B_L^{(\cdot)} = \begin{pmatrix}
0 & \cdots & \cdots & 0 \\
B_{21}^{(\cdot)} & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
B_{s1}^{(\cdot)} & B_{s2}^{(\cdot)} & \cdots & 0
\end{pmatrix}, \quad
B_U^{(\cdot)} = \begin{pmatrix}
B_{11}^{(\cdot)} & B_{12}^{(\cdot)} & \cdots & B_{1s}^{(\cdot)} \\
0 & B_{22}^{(\cdot)} & \cdots & B_{2s}^{(\cdot)} \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & B_{ss}^{(\cdot)}
\end{pmatrix}.
$$

With this, $C_1^{(P)}$ and $C_2^{(P)} \in \mathbb{R}^{s\cdot(k_{\max}+1)\times s\cdot(k_{\max}+1)}$ can be written as

$$
\begin{aligned}
C_1^{(P)} &= \begin{pmatrix}
\operatorname{diag}(c) & 0 & 0 & \cdots & \cdots & 0 \\
B_U^{(1)} - \theta_1 \mathfrak{I} & B_L^{(1)} + \theta_1 \mathfrak{I} & 0 & \cdots & \cdots & 0 \\
0 & B_U^{(1)} - \theta_1 \mathfrak{I} & B_L^{(1)} + \theta_1 \mathfrak{I} & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & B_U^{(1)} - \theta_1 \mathfrak{I} & B_L^{(1)} + \theta_1 \mathfrak{I}
\end{pmatrix}, \\
C_2^{(P)} &= \begin{pmatrix}
-\operatorname{diag}(c^2) & 0 & 0 & \cdots & \cdots & 0 \\
2B_U^{(2)} + \theta_2 \mathfrak{I} & 2B_L^{(2)} - \theta_2 \mathfrak{I} & 0 & \cdots & \cdots & 0 \\
0 & 2B_U^{(2)} + \theta_2 \mathfrak{I} & 2B_L^{(2)} - \theta_2 \mathfrak{I} & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & 2B_U^{(2)} + \theta_2 \mathfrak{I} & 2B_L^{(2)} - \theta_2 \mathfrak{I}
\end{pmatrix}.
\end{aligned}
\tag{2.6}
$$

Remark 2.2 In [19], the second order Taylor predictor has been substituted with a successive application of a 4^{th} order two-derivative method. This means that the predictor in Alg. 1 (see Eq. (2.1)) is substituted by

1. **Predict.** Solve the following expression for $w^{n,[0],l}$ and $2 \leq l \leq s$:

$$\begin{aligned} w^{n,[0],1} &:= w^{n-1,[k^*],s} \\ w^{n,[0],l} &:= w^{n,[0],l-1} + \frac{\Delta c_l \Delta t}{2}\left(\Phi^{n,[0],l-1} + \Phi^{n,[0],l}\right) \\ &\quad + \frac{(\Delta c_l \Delta t)^2}{12}\left(\dot{\Phi}^{n,[0],l-1} - \dot{\Phi}^{n,[0],l}\right), \end{aligned}$$

   with $\Delta c_l := c_l - c_{l-1}$.

As the modifications to Alg. 1 and the matrices in Eq. (2.3) (i.e., A, $C_1^{(\bullet)}$, $C_2^{(\bullet)}$ and D) are straightforward, we also analyze the stability properties of those schemes. They will be denoted with (Parallel-)HBPC$(q, k_{\max}, p4)$.

3 Stability of Hermite-Birkhoff Predictor-Corrector Scheme

In this section, we analyze the linear stability properties of the HBPC$(q, k_{\max})$ scheme with the modification proposed in the previous section. We perform a linear stability analysis using Dahlquist's equation $w'(t) = \lambda w(t)$, with $\lambda \in \mathbb{C}$. For this equation, one obtains for the ODE (1.1)

$$\Phi(w) = \lambda w, \qquad \dot{\Phi}(w) = \Phi'(w)\Phi(w) = \lambda^2 w.$$

Definition 3.1 With the abbreviation $z := \lambda \Delta t$, the stability function of a one-step method that can be cast into the form given in Def. 2.1 reads

$$R(z) := \left(D - zC_1^{(\bullet)} - \frac{z^2}{2}C_2^{(\bullet)}\right)^{-1} A^{(\bullet)}. \tag{3.1}$$

To ensure stability, $\lim_{n\to\infty}(R(z))^n = 0$ must hold. Hence, the method is A(α)-stable iff for the spectral radius $\rho\,(R(z))$, there holds

$$\rho\,(R(z)) < 1 \qquad \forall z \in S_\alpha := \{z \in \mathbb{C};\ |\arg(-z)| < \alpha,\ z \neq 0\}.$$

The argument-function has been normalized such that $-180° \leq \arg(z) \leq 180°$. A method is called A-stable if it is A($\alpha = 90°$)-stable.

To numerically determine the stability angle of a HBPC$(q, k_{\max})$ method with a specific parameter pair θ, we define a bisection procedure.

Algorithm 2 (Bisection to determine stability angle α) *First, initialize the upper and lower angles $\alpha_0^{max} = 90°$ and $\alpha_0^{min} = 0°$. Then, repeat for $j = 1, \cdots, J$, with $J = 20$:*

1. *Calculate stability angle angle via $\alpha_j = \frac{1}{2}\left(\alpha_{j-1}^{max} - \alpha_{j-1}^{min}\right)$.*
2. *Define $\{z_L \in \mathbb{C} : -25 < Re(z_L) < 0, Im(z_L) = |Re(z_L)|\tan(\alpha_j)\}$ and select equidistant points z_l from this set, $1 \leq l \leq 10^5$.*

3. Evaluate $\rho(R(z_l))$ for all z_l and set

$$\alpha_j^{max} = \begin{cases} \alpha_{j-1}^{max} & \text{if } \rho(R(z_l)) < 1 \ \forall z_l, \\ \alpha_j & \text{else,} \end{cases} \quad , \quad \alpha_j^{min} = \begin{cases} \alpha_j & \text{if } \rho(R(z_l)) < 1 \ \forall z_l, \\ \alpha_{j-1}^{min} & \text{else.} \end{cases}$$

Finally, set $\alpha = \frac{1}{2}\left(\alpha_J^{max} - \alpha_J^{min}\right)$ and check that there holds $\lim_{z \to -\infty} \rho(R(z)) \leq 1$.

3.1 Stability of the Original Schemes: $\theta_1 = \theta_2 = 1$

Theorem 3.1 *Alg. 1 with $\theta_1 = \theta_2 = 1$ and the quadrature formulas given in [14, Eqs. (2)-(4)] are A(α)-stable with $\alpha < 90°$ if $k_{\max} > 0$. This also holds for the HBPC$(q, k_{\max}, p4)$ schemes, except for the Parallel-HBPC$(8, k_{\max}, p4)$ method, which has an enclosed stability region.*

The stability function given in Eq. (3.1) is numerically analyzed in terms of the maximum opening angle α of the stability region for

- three different HBPC$(q, k_{\max})$ methods (4^{th}, 6^{th} and 8^{th} order),
- three different Parallel-HBPC$(q, k_{\max})$ methods (4^{th}, 6^{th} and 8^{th} order),
- two different HBPC$(q, k_{\max}, p4)$ schemes (6^{th} and 8^{th} order), and
- the Parallel-HBPC$(6, k_{\max}, p4)$ scheme,

for an increasing number of correction steps up to $k_{\max} = 50$. We select $\theta = (1,1)$ as it has been done in the original publications. The results of this analysis are shown in Fig. 3.1. One can see that the stability angle α heavily depends on the number of correction steps and can be in the order of $\alpha \approx 70°$ in some cases. Such a stability behavior is e.g. known in the spectral deferred correction (SDC) community [7], where the schemes also have a predictor-corrector flavor. As an illustrative example, a visualization of the stability region of the Parallel-HBPC$(8,1)$ scheme is provided in Fig. 3.2 (left). One can see that there are two pikes in the left complex half plane which cause a stability angle $\alpha < 90°$. As the stability angle often is not even close to $\alpha = 90°$, a modification of the schemes is desirable to improve the stability properties.

Remark 3.1 Note that the Parallel-HBPC$(8, k_{\max}, p4)$ scheme is not A(α)-stable for $k_{\max} > 0$. This can directly be seen in Fig. 3.2 (right), where the boundary of the stability region is visualized in the left complex half plane for $k_{\max} = 1$. One can see that the stability region is an enclosed area in the left complex half plane.

3.2 Stability Optimization of the HBPC$(q, k_{\max})$ Schemes

3.2.1 Stability of HBPC$(4, k_{\max})$

Theorem 3.2 *The serial HBPC$(4, k_{\max})$ scheme is A-stable if $\theta = \left(\frac{1}{2}, \frac{1}{6}\right)$ is chosen. For those parameters, the HBPC$(4, k_{\max} > 0)$ scheme reduces to the limiting HBRK4 method.*

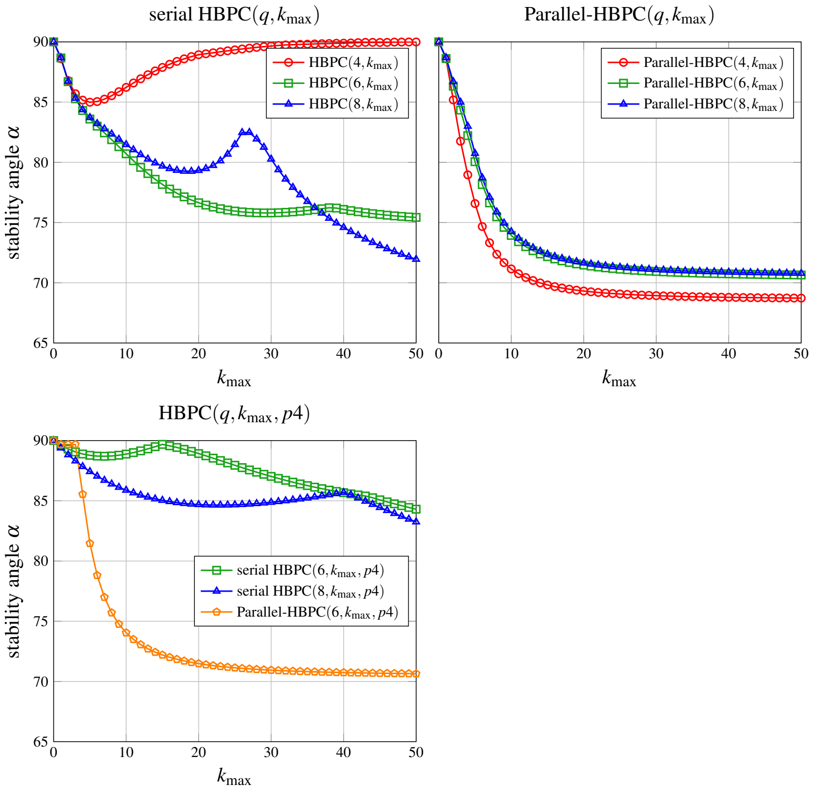

Fig. 3.1 Maximum opening angle α of stability region in left complex half plane for HBPC$(4,k_{\max})$, HBPC$(6,k_{\max})$ and HBPC$(8,k_{\max})$ for an increasing number of correction steps. Results of serial algorithm (top left) and parallel-in-time algorithm (top right), both with 2^{nd} order Taylor predictor. α for schemes with 4^{th} order HBRK4 predictor are visualized on the bottom left. Note that there is no HBPC$(4,k_{\max},p4)$ algorithm as the HBRK4 predictor directly gives the limiting scheme.

Proof Considering the last stage of the corrector step (Eq. (2.2)) of the serial algorithm with the fourth order quadrature rule and $\theta = \left(\frac{1}{2}, \frac{1}{6}\right)$ one obtains

$$
\begin{aligned}
w^{n,[k+1],2} := w^n &+ \frac{\Delta t}{2}\left(\Phi^{n,[k+1],2} - \Phi^{n,[k],2}\right) - \frac{\Delta t^2}{12}\left(\dot{\Phi}^{n,[k+1],2} - \dot{\Phi}^{n,[k],2}\right) \\
&+ \Delta t\left(\frac{1}{2}\Phi^{n,[k],1} + \frac{1}{2}\Phi^{n,[k],2}\right) + \Delta t^2\left(\frac{1}{12}\dot{\Phi}^{n,[k],1} - \frac{1}{12}\dot{\Phi}^{n,[k],2}\right).
\end{aligned}
$$

For the first stage it holds $w^{n,[k],1} = w^n$ and hence this reduces to

$$
w^{n,[k+1],2} = w^n + \frac{\Delta t}{2}\left(\Phi^n + \Phi^{n,[k+1],2}\right) + \frac{\Delta t^2}{12}\left(\dot{\Phi}^n - \dot{\Phi}^{n,[k+1],2}\right),
$$

which directly gives the A-stable limiting method. □

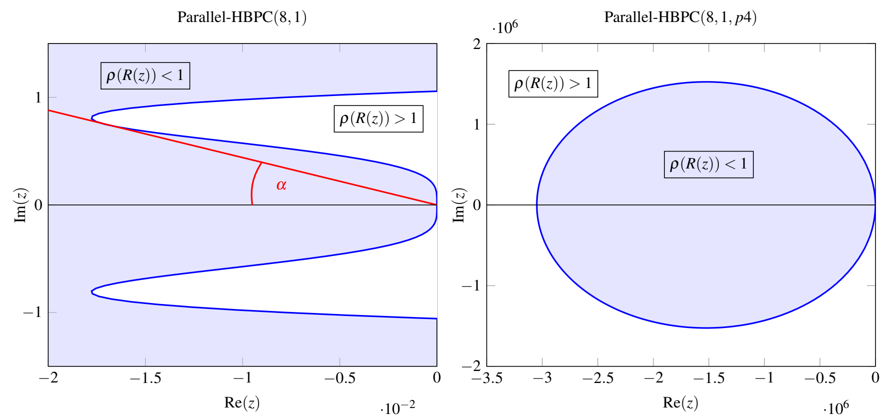

Fig. 3.2 Stability region (blue) for Parallel-HBPC$(8,1)$ schemes with 2^{nd} order Taylor predictor (left) and 4^{th} order HBRK4 predictor (right); please note the different scales of the axes. An illustration of the stability angle α is provided on the left.

Proposition 3.1 *For the same reasoning as for the serial algorithm, an A-stable scheme is obtained for the fourth order Parallel-HBPC*$(4, k_{\max} > 0)$ *method if* $\theta = \left(\frac{1}{2}, \frac{1}{6}\right)$ *is chosen.*

Proposition 3.2 *For the same reasoning as for the HBPC*$(4, k_{\max})$ *scheme, coefficients for the p-derivative* $2p^{th}$*-order schemes presented in [6] can be found to achieve A-stability. Using the coefficients of the p-derivative* $2p^{th}$*-order quadrature formula defines the p coefficients* $\theta_1, \ldots, \theta_p$.

Remark 3.2 If one uses the fully implicit serial method there is no advantage in preferring the predictor-corrector approach over the limiting method. Nevertheless, if one pursues a mixed implicit-explicit approach as it is done in [14] and [6], the HBPC$(4, k_{\max})$ scheme with $\theta = \left(\frac{1}{2}, \frac{1}{6}\right)$ and its higher order p-derivative extensions with optimized θ can be beneficial.

3.2.2 Stability of HBPC$(6, k_{\max})$ *and HBPC*$(8, k_{\max})$

For the HBPC$(6, k_{\max})$ and the HBPC$(8, k_{\max})$ method it is not possible to find θ such that they reduce to their limiting schemes. To obtain the optimized coefficients for the different schemes, we rely on an iterative procedure based on the bisection described in Alg. 2. We sample the parameter space $\theta \in \Omega$ and determine the minimum stability angle on the sampling points for $k_{\max} = 0, \ldots, 50$. We manually refine the sampling interval to determine the optimum parameters. Note that we have restricted the parameter space to $\Omega = [0,1] \times [0,1]$ as we have observed that using larger values for θ has an unfavorable influence on the solution quality due to an increased stiffness. Caused by the very steep gradients of the stability angle in the parameter space and the presence of several local maxima, automated refinement strategies did not succeed. We can therefore only state that the obtained values are close to the global maximum. Note that for the Parallel-HBPC$(8,k_{\max},p4)$ scheme no parameter pair θ has been found to achieve A(α)-stability.

The results of the optimization for the remaining schemes are summarized in Tbl. 3.1, which provides an overview on the optimal parameters and the resulting minimum stability angles. For reference, the minimum stability angles for the non-optimized case $\theta = (1,1)$ are also given in Tbl. 3.1. The corresponding visualization of the stability angle for a varying number of correction steps is shown in Fig. 3.3. The stability angles of the original schemes are visualized in the previous section, see Fig. 3.1. The results illustrate that the stability properties of the schemes can be improved significantly by optimizing the parameters θ_1 and θ_2. Except for the serial HBPC$(8,k_{\max},p4)$ scheme, the obtained stability angles are very close to 90°.

	θ_1	θ_2	optimized minimum stability angle α	minimum stability angle α with $\theta = (1,1)$
HBPC$(4,k_{\max})$	$1/2$	$1/6$	90°	85.00°
HBPC$(6,k_{\max})$	0.283	0.0528	89.72°	75.43°
HBPC$(8,k_{\max})$	0.395	0.0375	88.75°	71.95°
Parallel-HBPC$(4,k_{\max})$	$1/2$	$1/6$	90°	68.73°
Parallel-HBPC$(6,k_{\max})$	0.296	0.0527	89.56°	70.68°
Parallel-HBPC$(8,k_{\max})$	0.239	0.0246	89.20°	70.80°
HBPC$(6,k_{\max},p4)$	1.0	0.496	88.58°	84.30°
HBPC$(8,k_{\max},p4)$	1.0	0.689	84.63°	83.25°
Parallel-HBPC$(6,k_{\max},p4)$	0.266	0.0590	89.90°	70.65°

Table 3.1 Optimized parameters for correction step and resulting minimum stability angle for serial HBPC$(q,k_{\max})$ and low-storage Parallel-HBPC$(q,k_{\max})$ schemes with 2^{nd} order Taylor predictor (top) and 4^{th} order HBRK4 predictor (bottom). For reference, the minimum stability angles of the original schemes are displayed on the right.

Remark 3.3 Note that for the serial HBPC$(8,k_{\max},p4)$ method (see Fig. 3.3 (right, onset)), one can see that if one would choose $k_{\max} > 50$ for the optimization, one would probably obtain other coefficients θ as the stability angle decreases for $k_{\max} > 50$ for the chosen θ. However, we have chosen those values as one typically would not choose $k_{\max} > 50$ in a practical application.

4 Validation

After having shown the improved stability properties of the schemes, this section gives numerical evidence that the introduced modification $\theta \neq 1$ does

- not deteriorate the observed order of convergence, and
- generally improves the accuracy of the schemes.

We start by analyzing the order of convergence in Sec. 4.1 and continue with numerical results to illustrate the influence on the accuracy for stiff and non-stiff problems in Sec. 4.2.

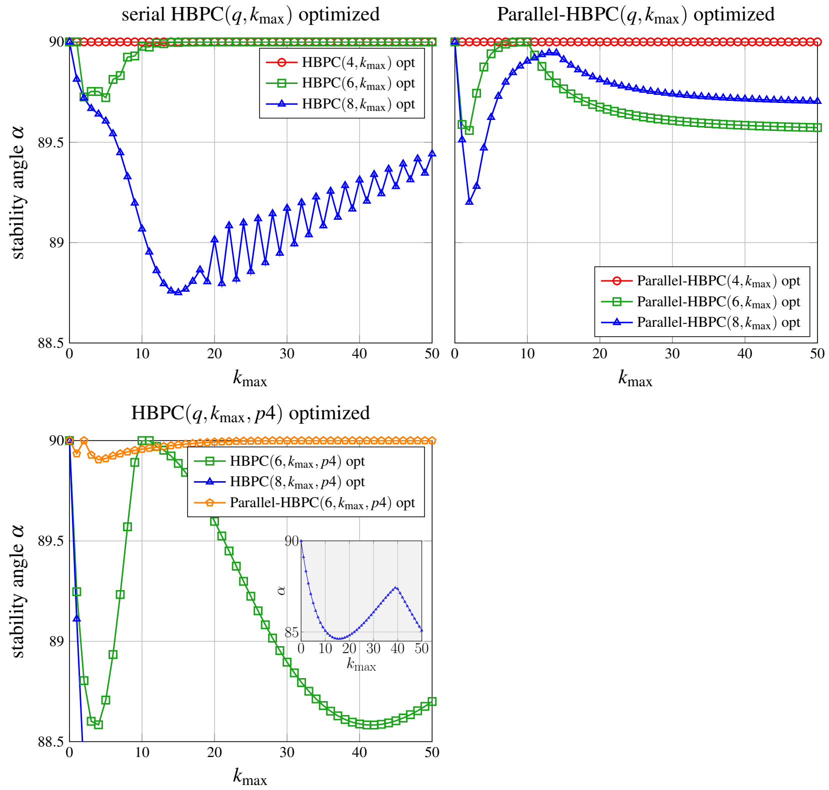

Fig. 3.3 Maximum opening angle α of stability region in left complex half plane with optimized parameters θ from Tbl. 3.1 for an increasing number of correction steps $k_{\max} = 0,\dots,50$. Results of serial algorithm (top left) and parallel-in-time algorithm (top right), both with 2^{nd} order Taylor predictor. α for schemes with 4^{th} order HBRK4 predictor are visualized on the bottom left. Note that due to the relatively low stability angle of the HBPC($6,k_{\max},p4$) method, an additional total view is provided as an onset figure on the bottom left.

4.1 Order of Convergence

We start by investigating the experimental order of convergence of the scalar model problem

$$w'(t) = -w^{\frac{5}{2}}, \quad w_0 = 1,$$

which has also been considered in [14]. We evaluate the error at $T_{end} = 0.25$ and compare the schemes with $\theta = (1,1)$ and with the optimized parameter sets obtained in Sec. 3.2. Performing a set of calculations with different step sizes allows us to validate the order of convergence, see Fig. 4.1 and Fig. 4.2 for the schemes using the 2^{nd} order Taylor predictor and the 4^{th} order HBRK4 predictor, respectively. As it is to be expected (see [13, 14, 19]), the obtained order of accuracy of the algorithms is $\min(k_{\max} + p, q)$, where q denotes the order of the quadrature rule and p the order of the predictor. We start by considering the HBPC($q,k_{\max}$) schemes, see Fig. 4.1.

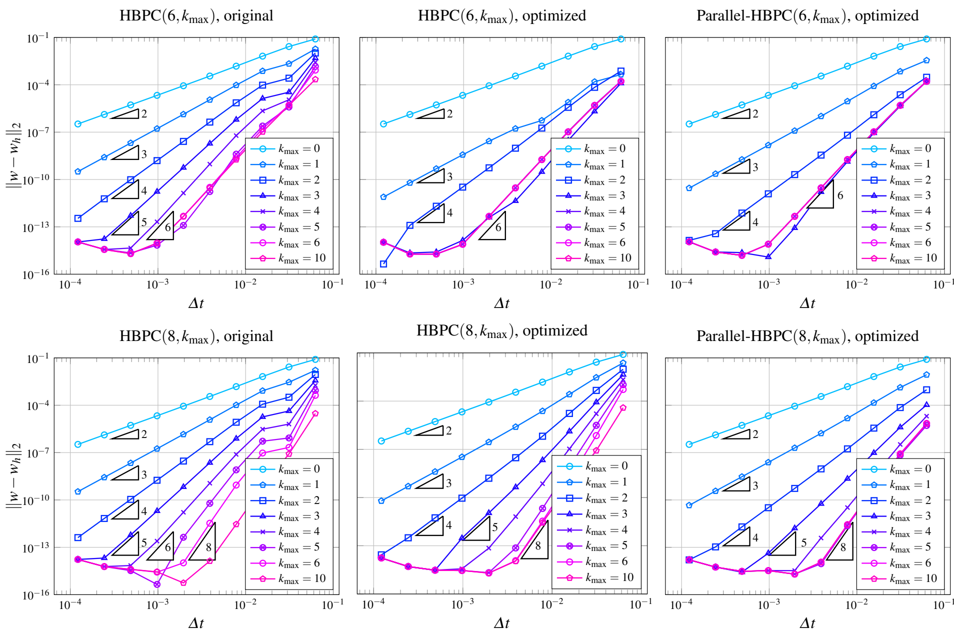

Fig. 4.1 Errors of ODE $w' = -w^{-\frac{5}{3}}$ at $T_{end} = 0.25$ for 6^{th} order (top) and 8^{th} order HBPC$(q, k_{\max})$ scheme (bottom). Left column shows results of serial methods with $\theta = 1$, middle column shows results for serial algorithms with optimized θ and right column shows results of parallel-in-time algorithms with optimized θ. Note that for $k_{\max} = 0$ all schemes are identical and reduce to the implicit second order Taylor method.

Concerning the accuracy, one can observe that the modification of θ leads to smaller errors until machine accuracy is reached for both, the serial and parallel-in-time methods as they show very similar results. For the HBPC$(q, k_{\max}, p4)$ schemes there are only slight differences between the original and the optimized serial algorithms. This is most probably due to θ being rather close to $\theta = (1,1)$ for the optimized scheme. For the sixth order parallel-in-time method, one can observe slightly clearer accuracy improvements by choosing the optimized θ.

Summing up, those calculations show that the modification of θ to optimize the stability with the parameters given in Tbl. 3.1 does not deteriorate the order of convergence and, at the same time, has a favorable influence on the accuracy.

4.2 Influence of the Corrector Modification on the Accuracy

Next, we evaluate the influence of the proposed modifications on the accuracy for stiff problems. For that purpose, we consider the van-der-Pol equation

$$\begin{pmatrix} w_1'(t) \\ w_2'(t) \end{pmatrix} = \begin{pmatrix} w_2 \\ \frac{(1-w_1^2)w_2 - w_1}{\varepsilon} \end{pmatrix}, \quad w_0 = \begin{pmatrix} 2 \\ -\frac{2}{3} + \frac{10}{81}\varepsilon \end{pmatrix},$$

with $T_{end} = 0.5$. We use different ε to evaluate the influence of the stiffness of the problem on the accuracy of the schemes. We do not report the results of the

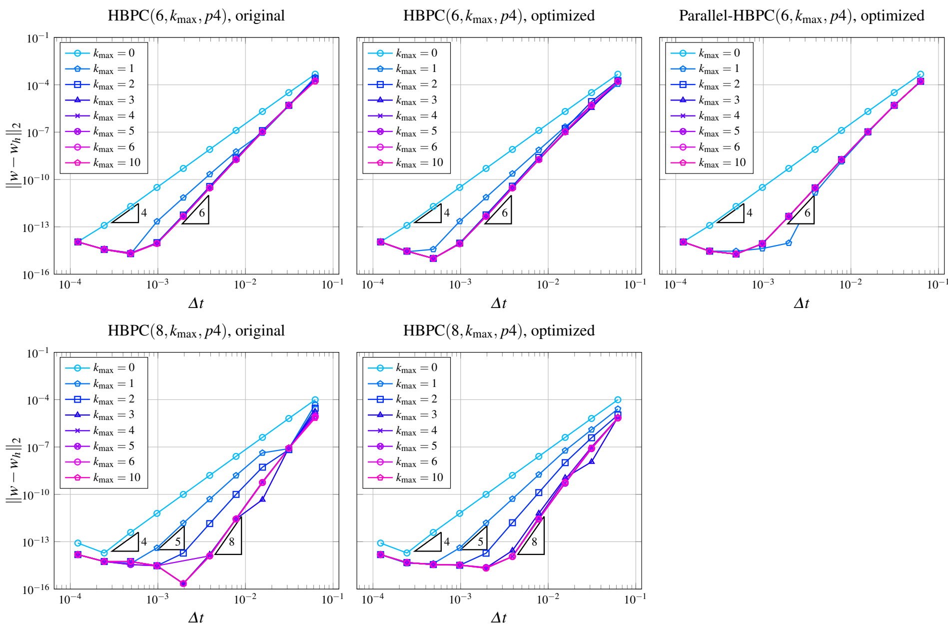

Fig. 4.2 Errors of ODE $w' = -w^{-\frac{5}{3}}$ at $T_{end} = 0.25$ for 6^{th} order (top) and 8^{th} order HBPC($q, k_{\max}, p4$) scheme (bottom) using the 4^{th} order predictor. Left column shows results of serial methods with $\theta = 1$, middle column shows results for serial algorithms with optimized θ and right column shows results of parallel-in-time algorithm with optimized θ. Note that no results for the Parallel-HBPC($8, k_{\max}, p4$) are shown as the method is not A(α)-stable.

parallel-in-time algorithm as we have observed similar errors for the serial and parallel method. The results shown in Fig. 4.3 illustrate the favorable influence of the optimized θ on the accuracy also for stiff problems. Especially for moderately stiff problems the improvements are quite large. One can see that the results obtained with the optimized θ are much closer to the solution obtained with the limiting method. Additionally, one can see that preferring the 4^{th} order HBRK4 predictor over the 2^{nd} order Taylor predictor has a favorable influence on the accuracy for stiff and non-stiff problems.

5 Conclusion and Outlook

In this work, we have introduced a small modification to the novel class of two-derivative predictor corrector schemes from [13] and [14] which has a large influence on the stability and accuracy. A stability analysis for the original algorithms revealed that the algorithms are A(α)-stable with the stability angle being only $\alpha \approx 70°$ for some cases. It has been shown that the introduced modification

- allows to improve the stability angle up to almost 90° for the serial and parallel-in-time algorithm.

– does not deteriorate the order of convergence and, in general, has a favorable influence on the accuracy.

Numerical experiments with stiff and non-stiff nonlinear ODEs illustrate this observation. As the modification requires only minor rework of already existing implementations, the authors highly recommend to use the proposed modification in a practical application.

Future work will be twofold: On the one hand, other quadrature rules can be used to design different methods offering e.g. L-stability and potentially taking higher derivatives into account. One can e.g. consider Gauß-Turan-type Runge-Kutta schemes for an odd number of derivatives. On the other hand, further studies of the algorithm's performance with different PDE discretizations will be done, also with a special focus on the parallel-in-time capabilities of the algorithm.

Acknowledgements J. Zeifang was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - project no. 457811052. D. Seal was funded by the Office of Naval Research, grant number N0001419WX01523 and N0001420WX00219. The computing resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

References

1. Aiguobasimwin, I.B., Okuonghae, R.I.: A class of two-derivative two-step Runge-Kutta methods for non-stiff ODEs. Journal of Applied Mathematics **Art. ID 2459809, 9** (2019)
2. Butcher, J.C.: On the convergence of numerical solutions to ordinary differential equations. Mathematics of Computation **20**, 1–10 (1966)
3. Butcher, J.C., Hojjati, G.: Second derivative methods with RK stability. Numerical Algorithms **40**(4), 415–429 (2005)
4. Chan, R., Tsai, A.: On explicit two-derivative Runge-Kutta methods. Numerical Algorithms **53**, 171–194 (2010)
5. Christlieb, A.J., Gottlieb, S., Grant, Z.J., Seal, D.C.: Explicit strong stability preserving multistage two-derivative time-stepping schemes. Journal of Scientific Computing **68**, 914–942 (2016)
6. Dittmann, A.J.: High-order multiderivative IMEX schemes. Applied Numerical Mathematics **160**, 205 – 216 (2021)
7. Dutt, A., Greengard, L., Rokhlin, V.: Spectral deferred correction methods for ordinary differential equations. BIT. Numerical Mathematics **40**(2), 241–266 (2000)
8. Gekeler, E.W.: On implicit Runge-Kutta methods with higher derivatives. BIT **28**, 809–816 (1988)
9. Hairer, E., Norsett, S.P., Wanner, G.: Solving ordinary differential equations I. Springer Series in Computational Mathematics (1987)
10. Hairer, E., Wanner, G.: Solving ordinary differential equations II. Springer Series in Computational Mathematics (1991)
11. Kastlunger, K., Wanner, G.: On Turan type implicit Runge-Kutta methods. Computing **9**, 317–325 (1972)
12. Ong, B.W., Spiteri, R.J.: Deferred correction methods for ordinary differential equations. Journal of Scientific Computing **83**(3), Paper No. 60, 29 (2020)
13. Schütz, J., Seal, D.: An asymptotic preserving semi-implicit multiderivative solver. Applied Numerical Mathematics **160**, 84–101 (2021)
14. Schütz, J., Seal, D.C., Zeifang, J.: Parallel-in-time high-order multiderivative IMEX solvers. arXiv preprint arXiv:2101.07846 (2021)
15. Seal, D., Güçlü, Y., Christlieb, A.: High-order multiderivative time integrators for hyperbolic conservation laws. Journal of Scientific Computing **60**, 101–140 (2014)
16. Stroud, A.H., Stancu, D.D.: Quadrature formulas with multiple Gaussian nodes. SIAM Journal on Numerical Analysis **2**, 129–143 (1965)

17. Ökten Turacı, M., Öziş, T.: Derivation of three-derivative Runge-Kutta methods. Numerical Algorithms **74**(1), 247–265 (2017)
18. Turán, P.: On the theory of the mechanical quadrature. Acta Universitatis Szegediensis Acta Scientiarum Mathematicarum **12**, 30–37 (1950)
19. Zeifang, J., Schütz, J.: Two-derivative deferred correction time discretization for the discontinuous Galerkin method. arXiv preprint arXiv:2109.04804 (2021)

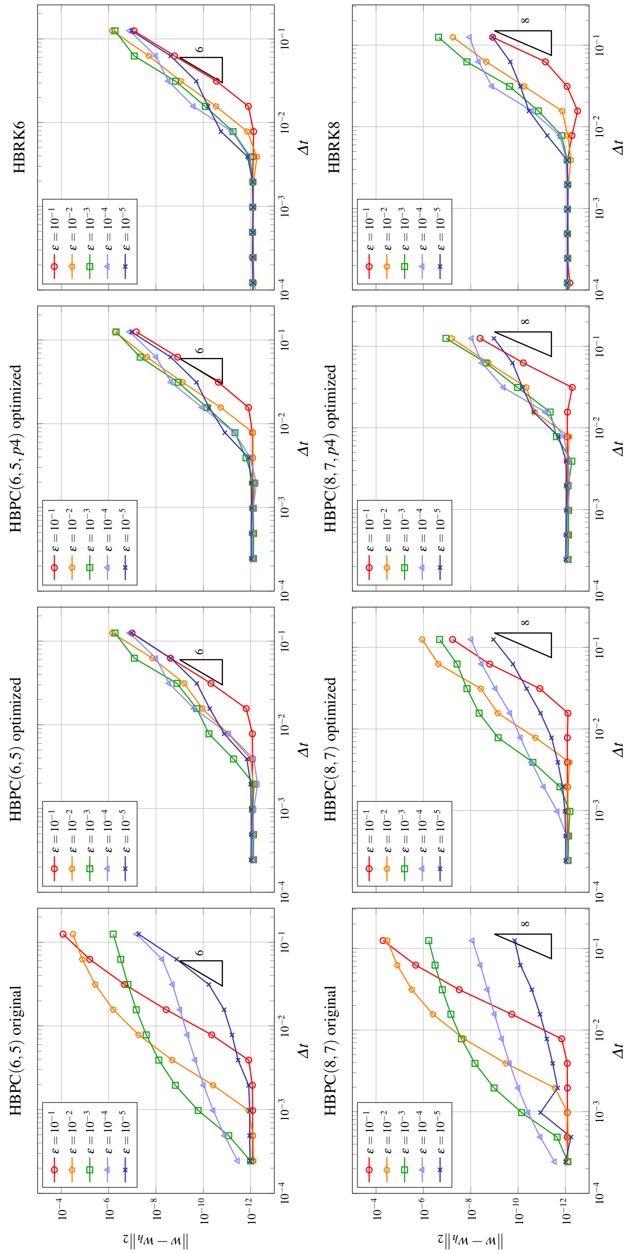

Fig. 4.3 Errors for van der Pol problem at $T_{end} = 0.5$ for original method with Taylor predictor, optimized HBPC$(q, k_{\max})$ method with Taylor predictor, optimized HBPC$(q, k_{\max}, p4)$ method with HBRK4 predictor with coefficients from Tbl. 3.1 and the limiting HBRK methods (from left to right).