VI International Conference on Adaptive Modeling and Simulation
ADMOS 2013
J. P. Moitinho de Almeida, P. Díez, C. Tiago and N. Parés (Eds)

ADAPTIVE TIMESTEP CONTROL FOR THE GENERALISED-α METHOD

JOACHIM RANG*

*Institute of Scientific Computing
TU Braunschweig
Hans-Sommer Str. 65, 38106 Braunschweig, Germany
e-mail: j.rang@tu-bs, web page: http://www.wire.tu-bs.de

Key words: Adaptivity, generalised-α methods

Abstract. In this article an adaptive timestep control for the generalised-α methods is introduced. If the methods for first and second order ODEs are formulated as onestep schemes a second solution can be computed with the backward Euler method, which costs no additional computing time. If the generalised-α methods are formulated as multistep methods an adaptive timestep control is only introduced for first order ODEs. In this case the method is formulated with variable coefficients and a second solution is computed with the Leapfrog method. Numerical examples show in the case of the onestep versions the advantages of the adaptive algorithms.

1 INTRODUCTION

In this article we consider the generalised-α methods, which are introduced for first order ODEs in [JWH00] and for second order ODEs in [CH93]. The generalised-α methods are usually of second order and allow the damping of high frequencies, which can be controlled by certain parameters. An analysis for first order problems can be found in [DP03]. In the case of second order ODEs many papers can be found, which analyse the generalised-α method, for example [EBB02]. It is well known that the generalised-α method for first order problems can be formulated as onestep and multistep methods. In the case of second order methods this statement is only true if the ODE is linear in the first derivative (see [EBB02]). For both classes of multistep methods second order can be achieved if a further order condition is satisfied. Together with stability conditions (see [EBB02]) a robust and effective class of methods is obtained. If these parameter sets are used for onestep methods theoretically only first order can be reached. But the error constant is very small so that the observed numerical order of convergence is two. Moreover in our experience the onestep versions achieve better results than the multistep versions.

For solving ODEs or DAEs a good time integration method needs an error estimator to increase efficiency. This error estimator suggests a new timestep size to reach a given accuracy. If the timestep size is too small a lot of unnecessary computational work has to be done. Otherwise, if the timestep size is too large, the results become less accurate. In [HNW93] two approaches for time-adaptive one-step methods are presented. The first one is called *Richardson extrapolation* and can be applied to every one-step method. In this case the calculations for computing an approximation of the solution at the next timestep are repeated with the timestep size $\tau/2$ and compared with the first result, i.e. the computational work increases by a factor of 3. Thus, the question of efficiency arises.

A more effective control of timesteps can be achieved with the so-called embedding technique, which can be used for many Runge–Kutta and Rosenbrock–Wanner methods [HW96, SW92]. In this case a second solution can be computed with almost the same coefficients and without solving a further linear or non-linear solution, i.e. there are almost no further computational costs. Applications can be found in [HHR12, JR10, Lan01, Ran04].

In this article different approaches are considered. In the case of the onestep formulation of the generalised-α method the backward Euler method can be used to compute a second solution without any further computation. As the numerical examples will show this is an effective way of computing adaptive timestep sizes.

In the case of the multistep formulation we need methods, which have variable coefficients. In the case of second order ODEs we get a formulation, which involves potentials of the mass and the damping matrix. Therefore we only develop for the onestep version a formulation with variable coefficients. In this case a second solution can be computed with the Leapfrog method. This approach can be found in [GS00] for a backward difference formular.

This paper is structured as follows. First we introduce the generalised-α methods for first and second order ODEs. A short analysis about convergency and stability is given. Then adaptive algorithms are explained and numerical examples illustrate the adavantages of the new adaptive methods.

2 THE GENERALISED-α METHOD FOR 1ST ORDER ODES

In the following we consider the ODE

$$\dot{\mathbf{u}} = \mathbf{f}(t, \mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0. \tag{1}$$

To determine the numerical solution of (1) we use the generalised-α method, which is given by the formulas (see [JWH00, DP03])

$$\dot{\mathbf{u}}_{n+\alpha_m} = \mathbf{f}(t_{n+\alpha_f}, \mathbf{u}_{n+\alpha_f}), \tag{2}$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau\dot{\mathbf{u}}_n + \tau\gamma(\dot{\mathbf{u}}_{n+1} - \dot{\mathbf{u}}_n), \tag{3}$$

$$\dot{\mathbf{u}}_{n+\alpha_m} = \dot{\mathbf{u}}_n + \alpha_m(\dot{\mathbf{u}}_{n+1} - \dot{\mathbf{u}}_n), \tag{4}$$

$$\mathbf{u}_{n+\alpha_f} = \mathbf{u}_n + \alpha_f(\mathbf{u}_{n+1} - \mathbf{u}_n). \tag{5}$$

It is well known that the generalised-α method can be formulated as a onestep and a twostep method.

2.1 The formulation as onestep method and its analysis

First we manipulate the formulas (2)–(5) to obtain a non-linear system consisting of two decoupled equations. For simplification we define $\mathbf{f}_{n+\alpha_f} := \mathbf{f}(t_{n+\alpha_f}, \mathbf{u}_{n+\alpha_f})$. A simple calculation gives us

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau\left(1 - \frac{\gamma}{\alpha_m}\right)\dot{\mathbf{u}}_n + \frac{\tau\gamma}{\alpha_m}\mathbf{f}_{n+\alpha_f}, \tag{6}$$

$$\dot{\mathbf{u}}_{n+1} = \frac{1}{\tau\gamma}\left(\mathbf{u}_{n+1} - \mathbf{u}_n - \tau(1-\gamma)\dot{\mathbf{u}}_n\right), \tag{7}$$

if $\alpha_m \neq 0$. We call the scheme (6)–(7) the *onestep generalised-α method*. The starting value $\dot{\mathbf{u}}_0$ can be computed from the ODE (1). Next we want to determine the order of consistency. Therefore the numerical solution $\mathbf{u}_{n+1}$ can be expanded in a Taylor series as follows

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau\dot{\mathbf{u}}_n + \frac{\tau^2\gamma\alpha_f}{\alpha_m}\ddot{\mathbf{u}}_n + \mathcal{O}(\tau^3).$$

For consistency of order 2 we get the condition $\dfrac{\gamma\alpha_f}{\alpha_m} = \dfrac{1}{2}$. Since $\mathbf{u}_{n+1}$ depends on $\dot{\mathbf{u}}_n$ we use equation (7) for expanding $\dot{\mathbf{u}}_{n+1}$ in a Taylor series and get

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \frac{\tau\alpha_f}{\alpha_m}\ddot{\mathbf{u}}_n + \mathcal{O}(\tau^2),$$

i. e. $\dot{\mathbf{u}}_{n+1}$ is of order 1 if $\dfrac{\alpha_f}{\alpha_m} = 1$. Summarising our results we have consistency of order 2 if $\alpha_m = \alpha_f$ and $\gamma = 1/2$. It can be easily shown that the generalised-α method is zero-stable if $\alpha_m > 1/2$. In other words our method is convergent if $\alpha_m > 1/2$.

2.2 Formulation as multistep method and its analysis

The generalised-α method can be formulated as a twostep method as follows

$$\mathbf{u}_{n+1} = \frac{2\alpha_m - 1}{\alpha_m}\mathbf{u}_n - \frac{\alpha_m - 1}{\alpha_m}\mathbf{u}_{n-1} + \frac{\tau(1-\gamma)}{\alpha_m}\mathbf{f}_{n-1+\alpha_f} + \frac{\tau\gamma}{\alpha_m}\mathbf{f}_{n+\alpha_f}. \tag{8}$$

For $\alpha_m = 3/2$, $\alpha_f = 1$ and $\gamma = 1$ we obtain the backward difference formula (BDF) from Gear (see [HW96]). Next we expand $\mathbf{u}_{n+1}$ in a Taylor expansion and compare it with the exact solution. Then we have

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau\dot{\mathbf{u}}_n + \frac{\tau^2}{2}\frac{2\alpha_f - \alpha_m + 2\gamma - 1}{\alpha_m}\ddot{\mathbf{u}}_n + \mathcal{O}(\tau^3).$$

Comparing the Taylor expansions for $\mathbf{u}(t_{n+1})$ and $\mathbf{u}_{n+1}$ leads to the condition for second order consistency

$$\gamma = \frac{1}{2} - \alpha_f + \alpha_m, \tag{9}$$

which is already known from [JWH00, DP03]). The generalised-α method in form (8) is convergent of order 2 if $\alpha_m > 1/2$ and condition (9) holds. For stability the setting

$$\alpha_f = \gamma = \frac{1}{1+\rho_\infty}, \qquad \alpha_m = \frac{3-\rho_\infty}{2(1+\rho_\infty)}. \tag{10}$$

is used (see [JWH00, DP03]). Note that condition (9) is automatically satisfied. For $\rho_\infty = 0$ we get the BDF-2 method.

3 THE GENERALISED-α METHOD FOR SECOND ORDER ODES

3.1 The formulation as onestep method

In the following we consider the second order ODE

$$\ddot{\mathbf{u}} = \mathbf{f}(t, \mathbf{u}, \dot{\mathbf{u}}), \quad \mathbf{u}(0) = \mathbf{u}_0, \dot{\mathbf{u}}(0) = \dot{\mathbf{u}}_0. \tag{11}$$

The generalised-α method can be written as

$$\mathbf{u}_{n+\alpha_f} = \alpha_f \mathbf{u}_{n+1} + (1-\alpha_f)\mathbf{u}_n, \tag{12}$$

$$\dot{\mathbf{u}}_{n+\alpha_f} = \alpha_f \dot{\mathbf{u}}_{n+1} + (1-\alpha_f)\dot{\mathbf{u}}_n, \tag{13}$$

$$\ddot{\mathbf{u}}_{n+\alpha_m} = \alpha_m \ddot{\mathbf{u}}_{n+1} + (1-\alpha_m)\ddot{\mathbf{u}}_n, \tag{14}$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau\dot{\mathbf{u}}_n + \tau^2\left[\left(\frac{1}{2}-\beta\right)\ddot{\mathbf{u}}_n + \beta\ddot{\mathbf{u}}_{n+1}\right], \tag{15}$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \tau\left[(1-\gamma)\ddot{\mathbf{u}}_n + \gamma\ddot{\mathbf{u}}_{n+1}\right], \tag{16}$$

$$\ddot{\mathbf{u}}_{n+\alpha_m} = \mathbf{f}(t_{n+\alpha_f}, \alpha_f\mathbf{u}_{n+1} + (1-\alpha_f)\mathbf{u}_n, \alpha_f\dot{\mathbf{u}}_{n+1} + (1-\alpha_f)\dot{\mathbf{u}}_n), \tag{17}$$

where $t_{n+\alpha_f} = t_n + \tau\alpha_f$. To abbreviate we write

$$\mathbf{f}_{n+\alpha_f} := \mathbf{f}(t_{n+\alpha_f}, \alpha_f\mathbf{u}_{n+1} + (1-\alpha_f)\mathbf{u}_n, \alpha_f\dot{\mathbf{u}}_{n+1} + (1-\alpha_f)\dot{\mathbf{u}}_n).$$

These equations can be simplyfied to

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau\dot{\mathbf{u}}_n + \tau^2\left[\left(\frac{1}{2}-\frac{\beta}{\alpha_m}\right)\ddot{\mathbf{u}}_n + \frac{\beta}{\alpha_m}\mathbf{f}_{n+\alpha_f}\right], \tag{18}$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \tau\left[\left(1-\frac{\gamma}{\alpha_m}\right)\ddot{\mathbf{u}}_n + \frac{\gamma}{\alpha_m}\mathbf{f}_{n+\alpha_f}\right]. \tag{19}$$

$$\ddot{\mathbf{u}}_{n+1} = \frac{1}{\alpha_m}\left[\ddot{\mathbf{u}}_{n+\alpha_m} - (1-\alpha_m)\ddot{\mathbf{u}}_n\right] = \frac{1}{\alpha_m}\left[\mathbf{f}_{n+\alpha_f} - (1-\alpha_m)\ddot{\mathbf{u}}_n\right]. \tag{20}$$

Next we expand these three expression into Taylor expansions and get

$$\begin{aligned}
\mathbf{u}_{n+1} &= \mathbf{u}_n + \tau\dot{\mathbf{u}}_n + \frac{1}{2}\tau^2\ddot{\mathbf{u}}_n + \mathcal{O}(\tau^3),\\
\dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n + \tau\ddot{\mathbf{u}}_n + \gamma\frac{\alpha_f}{\alpha_m}\tau^2\ddot{\mathbf{u}}_n + \mathcal{O}(\tau^3),\\
\ddot{\mathbf{u}}_{n+1} &= \ddot{\mathbf{u}}_n + \tau\frac{\alpha_f}{\alpha_m}\dddot{\mathbf{u}}_n + \mathcal{O}(\tau^2).
\end{aligned}$$

It follows that the method is of order 2 if $\alpha_f/\alpha_m = 1$ and $\gamma\alpha_f/\alpha_m = 1/2$. This is the same result as in the previous section.

3.2 Formulation as multistep method

As in the previous section the generalised-α method can be written as a multistep method if the ODE (11) is linear in $\dot{\mathbf{u}}$. Therefore we consider the problem as in [EBB02]

$$M\ddot{\mathbf{u}} + C\dot{\mathbf{u}} + \mathbf{S}(\mathbf{u}) = \mathbf{F}(t), \mathbf{u}(0) = \mathbf{u}_0, \quad \dot{\mathbf{u}}_0 = \mathbf{v}_0. \tag{21}$$

Then equation (17) reads as

$$M\ddot{\mathbf{u}}_{n+\alpha_m} = \mathbf{F}(t_{n+\alpha_f}) - S(\alpha_f\mathbf{u}_{n+1} + (1-\alpha_f)\mathbf{u}_n) - C(\alpha_f\dot{\mathbf{u}}_{n+1} + (1-\alpha_f)\dot{\mathbf{u}}_n). \tag{22}$$

The generalised-α method can be formulated as a multistep method with the help of (15), (16), and (22). These formulas are evaluated at time t_n, t_{n+1}, and t_{n+2} (see for example [EBB02]). Then we get

$$\sum_{j=0}^{3}[M\alpha_j + \tau C\gamma_j]\mathbf{u}_{n+j} + \tau^2\sum_{j=0}^{2}\delta_j[\mathbf{S}_{n+j+\alpha_f} - \mathbf{F}(t_{n+j+\alpha_f})] = 0, \tag{23}$$

where

$$\begin{aligned}
&\alpha_0 = 1-\alpha_m, \quad \alpha_1 = 3\alpha_m - 2, \quad \alpha_2 = 1 - 3\alpha_m, \quad \alpha_3 = \alpha_m,\\
&\gamma_0 = (1-\alpha_f)(\gamma-1), \quad \gamma_1 = 1 - 2\alpha_f - 2\gamma + 3\gamma\alpha_f, \gamma_2 = \alpha_f + \gamma - 3\gamma\alpha_f, \quad \gamma_3 = \alpha_f\gamma,\\
&\delta_0 = \frac{1}{2} + \beta - \gamma, \quad \delta_1 = \frac{1}{2} - 2\beta + \gamma, \quad \delta_2 = \beta
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{F}_{n+j-\alpha_f} &= \mathbf{F}(\alpha_f t_{n+j+1} + (1-\alpha_f)t_{n+j}) = \mathbf{F}(t_{n+j} + \alpha_f\tau)\\
\mathbf{S}_{n+j+\alpha_f} &= \alpha_f\mathbf{S}(\mathbf{u}_{n+j+1}) + (1-\alpha_f)\mathbf{S}(\mathbf{u}_{n+j}).
\end{aligned}$$

The method has consistency order 2 if $\gamma = \frac{1}{2} + \alpha_m - \alpha_f$. The method is zero-stable and convergent if $\alpha_m \geq 1/2$, $\alpha_f \leq 1/2$ and $\gamma \leq 1/2$ (see [EBB02]). For stability often the setting

$$\beta = \frac{(1+\alpha_m-\alpha_f)^2}{4}, \alpha_f = \frac{1}{1+\rho_\infty}, \alpha_m = \frac{2-\rho_\infty}{1+\rho_\infty}$$

is used (see [CH93]).

4 ADAPTIVITY

4.1 Adaptivity for the onestep version

If the generalised-α methods are formulated as onestep methods the so-called PI-controller from Gustafsson et. al. [GLS88] can be used. To suggest a new timestep size we need solutions of order p and $p-1$. The approximation of the generalised-α method can be used as a second order approximation since the error constant is very small and the methods behave in our numerical experiments as a second order method. As the second solution with order 1 we use the backward Euler method. The next timestep size τ_{n+1} is proposed to be

$$\tau_{n+1} = \rho \frac{\tau_n^2}{\tau_{n-1}} \left(\frac{TOL \cdot r_n}{r_{n+1}^2} \right)^{1/2}, \tag{24}$$

where $\rho \in (0,1]$ is a safety factor, $TOL > 0$ is a given tolerance, and $r_{n+1} := \|\mathbf{u}_{n+1} - \hat{\mathbf{u}}_{n+1}\|$. In [HNW93, HW96, SW92] different error measures can be found, which use a combination of relative and absolute errors. For further details about the numerical error and the implementation of automatic steplength control we refer to [HW96, Lan01]. The algorithm reads as follows:

- Compute the numerical solution $(\mathbf{u}_{n+1}, \dot{\mathbf{u}}_{n+1})^\top$ with the help of the generalised-α method (6), (7).
- Compute the second solution with the backward Euler method and use $\dot{\mathbf{u}}_{n+1}$ as approximation for $\mathbf{f}(t_{n+1}, \mathbf{u}_{n+1})$, i. e. $\hat{\mathbf{u}}_{n+1} = \mathbf{u}_n + \tau_n \dot{\mathbf{u}}_{n+1}$.
- Compute the numerical error with r_{n+1} and approximate the new timestep length τ_{n+1} with (24).
- If the numerical error is smaller than the given tolerance the timestep is accepted, otherwise it is rejected and has to be recomputed with the new timestep length τ_{n+1}.

In case of second order ODEs we use equations (12)–(17) to compute the numerical approximation $\mathbf{u}_{n+1}$. As in the case of the first order ODEs the backward Euler can be used for computing the first order solution.

The chemical reaction E5 This chemical reaction problem is called E5 and can be found in the collection by Enright, Hull, and Lindberg [EHL75]. The equations are given by

$$\begin{aligned}
\dot{u}_1 &= -Au_1 - Bu_1u_3, \\
\dot{u}_2 &= Au_1 - MCu_2u_3, \\
\dot{u}_3 &= Au_1 - Bu_1u_3 - MCu_2u_3 + Cu_4, \\
\dot{u}_4 &= Bu_1u_3 - Cu_4
\end{aligned}$$

with the initial conditions $u_1(0) = 1,76 \times 10^{-3}$ and $u_i(0) = 0$, $i \in \{2,3,4\}$. Moreover we set as in [HW96] $A = 7,89 \times 10^{-10}$, $B = 1,1 \times 10^7$, $C = 1,13 \times 10^3$, and $M = 10^6$. The equations should be solved in the time interval $[0, 10^{13}]$. Note that the variables u_2, u_3, and u_4 satisfy the equation $u_2 - u_3 - u_4 = 0$. The parameter ρ is chosen to be 0, 1/4, 1/2, 3/4, and 9/10, resp. For ρ tending to 1 the algorithm becomes instable. We compare the generalised-α methods with other implicit and linear-implicit second order solvers like ROS2 (see [VSBH99]), ROS2S (see [HHR12]), and the method of Ellsiepen (see [EH01]). It can be observed from Figure 1 that the generalised-α methods with the new stepsize controller are more effective than the other second order methods.

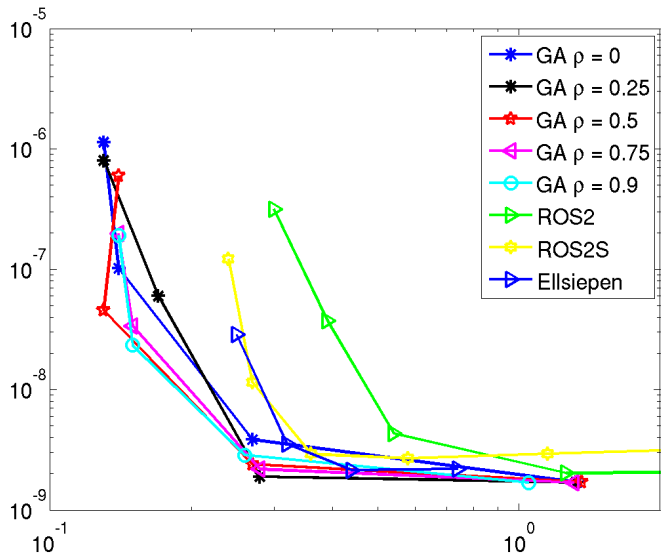

Figure 1: Comparison of generalised-α methods for first order ODEs: CPU time versus error

Kepler's problem Consider the second order ODE

$$\ddot{y}_i = -\frac{y_i}{(y_1^2 + y_2^2)^{3/2}}, \quad i = 1, 2.$$

The initial conditions are given by $\mathbf{u}_0 = \left(0, \sqrt{\frac{1+e}{1-e}}, 1-e, 0\right)^\top$, where $e \in [0,1)$ is a given parameter. In our numerical example we choose $e = 1/2$. We solve the problem in the interval $[0, 20000]$ with the generalised-α methods for second order ODEs and use the new adaptive timestep control. The parameter ρ is chosen to be 0, 1/4, 1/2, 3/4, and 9/10, resp. We compare the generalised-α methods with other implicit and linear-implicit second order solvers like ROS2 (see [VSBH99]), ROS2S (see [HHR12]), and the method of

Ellsiepen (see [EH01]). It can be observed from Figure 2 that the generalised-α methods with the new stepsize controller are more effective than the other second order methods. In this case we get better results for a larger ρ.

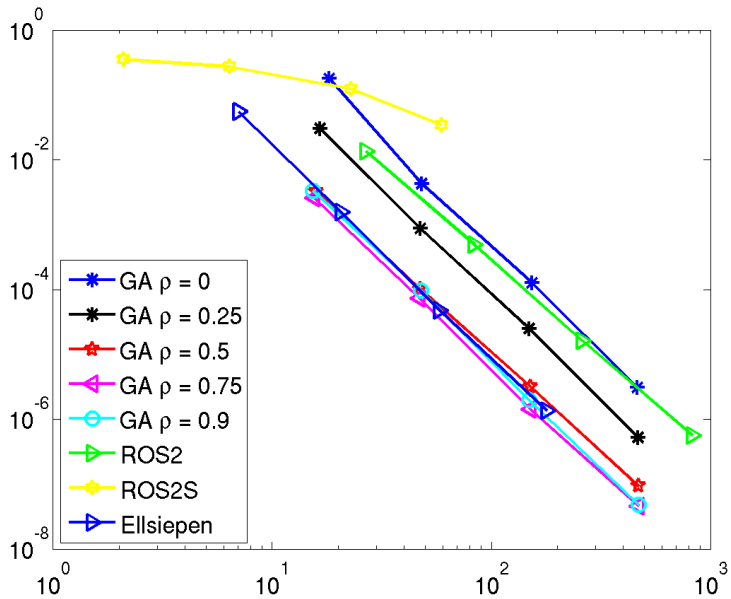

Figure 2: Comparison of generalised-α methods for second order ODEs: CPU time versus error

4.2 Adaptivity for the multistep version

In this section we derive first a multistep formula for the generalised-α method, which has variable coefficients. Note that this idea works only for the generalised-α method for first order problems. In the case of the generalised-α method for second order problems matrix potentials must be computed.

Let us start with the generalised-α method for first order problems. We want to formulate this method as a twostep method with variable coefficients. We consider equations (6) for t_n and t_{n+1} and (7) for t_n. A simple calculation leads to

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \frac{\tau_{n+1}}{\tau_n}\frac{\alpha_m - 1}{\alpha_m}(\mathbf{u}_n - \mathbf{u}_{n-1}) - \tau_{n+1}\frac{\gamma - 1}{\alpha_m}\mathbf{f}_{n-1+\alpha_f} + \tau_{n+1}\frac{\gamma}{\alpha_m}\mathbf{f}_{n+\alpha_f}.$$

The BDF-2 method with variable timesteps is a special case with the setting $\gamma = \alpha_f = 1$ and $\alpha_m = (2\tau_{n-1} + \tau_n)(\tau_{n-1} + \tau_n)$.

Next we want to derive the condition for order 2. Therefore we compute a Taylor expansion of $\mathbf{u}_{n+1}$ and obtain

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau_{n+1}\dot{\mathbf{u}}_n + \frac{\tau_{n+1}}{\alpha_m}\left[-\frac{\tau_n}{2}(\alpha_m - 1) - \tau_n(\gamma - 1)(\alpha_f - 1) + \tau_{n+1}\gamma\right]\ddot{\mathbf{u}}_n + \mathcal{O}(\tau_{n+1}^3)$$

It follows

$$t_n(1-\alpha_m) - 2\tau_n(\gamma-1)(\alpha_f-1) + 2\gamma\alpha_f\tau_{n+1} = \tau_{n+1}\alpha_m. \tag{25}$$

Next we consider the problem $\dot{u} = 0$. Applying the adaptive generalised-α method we obtain

$$u_{n+1} - \left(1 + \omega_n \frac{\alpha_m - 1}{\alpha_m}\right) u_n + \omega_n \frac{\alpha_m - 1}{\alpha_m} u_{n-1} = 0.$$

It follows

$$\xi^2 - \left(1 + \omega_n \frac{\alpha_m - 1}{\alpha_m}\right) \xi + \omega_n \frac{\alpha_m - 1}{\alpha_m} = 0,$$

which has the solutions $\xi_1 = 1$ and $\xi_2 = \omega_n(\alpha_m - 1)/\alpha_m$. It follows

$$\frac{\omega_n}{\omega_n + 1} \leq \alpha_m \leq \frac{\omega_n}{|\omega_n - 1|}.$$

For the A-stability of the method we consider the problem $\dot{u} = \lambda u$, $\lambda < 0$. Using the adapative generalised-α method we get

$$\begin{aligned} u_{n+1} = &\left(1 + \omega_n \frac{\alpha_m - 1}{\alpha_m}\right) u_n - \omega_n \frac{\alpha_m - 1}{\alpha_m} u_{n-1} + \frac{\tau_{n+1}(1-\gamma)}{\alpha_m} \lambda((1-\alpha_f)u_{n-1} + \alpha_f u_n)) \\ &+ \frac{\tau_n \gamma}{\alpha_m} \lambda((1-\alpha_f)u_n + \alpha_f u_{n+1}). \end{aligned}$$

We are interested in the case $\lambda \to -\infty$ and get

$$0 = \tau_n \gamma \alpha_f \xi^2 + (\tau_{n+1}(1-\gamma)\alpha_f + \tau_n\gamma(1-\alpha_f))\, \xi + \tau_{n+1}(1-\gamma)(1-\alpha_f).$$

The solutions of this equation are given by

$$\xi_1 = \frac{\alpha_f - 1}{\alpha_f}, \quad \xi_2 = \tau_{n+1} \frac{\gamma - 1}{\tau_n \gamma}.$$

As in the case of constant coefficients we solve $\xi_1 = -\rho_\infty$ and $\xi_1 = -\rho_\infty$ together with (25) and get

$$\gamma = \frac{\tau_{n+1}}{\tau_{n+1} + \rho_\infty \tau_n}, \alpha_f = \frac{1}{\rho_\infty + 1}, \alpha_m = \frac{2\tau_{n+1}\tau_n - \tau_n \rho_\infty}{\tau_n + \rho_\infty \tau_n + \tau_{n+1} + \tau_{n+1}\rho_\infty}.$$

For adaptivity we want to use a so-called *predictor-corrector scheme* (see for example [GS00]). The predictor is a scheme, which needs no solution of a linear or nonlinear system, for example an explicit method. The corrector is the desired method, in our case the generalised-α-method.

Let us assume that the predictor and the corrector are of order p, i. e. it holds

$$\mathbf{u}_{n+1}^p - \mathbf{u}(t_{n+1}) = C_p \frac{\tau_{n+1}^{p+1}}{(p+1)!} \mathbf{f}^{(p+1)}(t_n), \tag{26}$$

$$\mathbf{u}_{n+1}^c - \mathbf{u}(t_{n+1}) = C_c \frac{\tau_{n+1}^{p+1}}{(p+1)!} \mathbf{f}^{(p+1)}(t_n), \tag{27}$$

where $\mathbf{u}_{n+1}^p$ is the approximation of the predictor and $\mathbf{u}_{n+1}^c$ is the approximation of the corrector. In the system (26)–(27) the quantities $\mathbf{u}(t_{n+1})$ and $\mathbf{f}^{(p+1)}(t_n)$ are unknown. Therefore we solve equation (26) w.r.t. $\mathbf{u}(t_{n+1})$ and insert it into equation (27). Then we get the following approximation of the local truncation error

$$\mathbf{d}_{n+1} = \mathbf{u}_{n+1}^c - \mathbf{u}(t_{n+1}) = C_c \frac{\tau_{n+1}^{p+1}}{(p+1)!} \mathbf{f}^{(p+1)}(t_n) = \frac{C_c}{C_c - C_p} (\mathbf{u}_{n+1}^c - \mathbf{u}_{n+1}^p). \tag{28}$$

As predictor we want to use the leapfrog method (see [GS00]) given by

$$\mathbf{u}_{n+1}^p = \mathbf{u}_n + \left(1 + \frac{\tau_n}{\tau_{n+1}}\right) \tau_{n+1} \dot{\mathbf{u}}_n - \left(\frac{\tau_{n+1}}{\tau_n}\right)^2 (\mathbf{u}_n - \mathbf{u}_{n-1}).$$

Expanding this formula into a Taylor expansion gives us the constant C_p, which reads as

$$C_p = -\left(1 + \frac{\tau_n}{\tau_{n+1}}\right).$$

The error constant C_c we receive from the Taylor expansion of the generalised-α method. We obtain

$$C_c = \left(\frac{\tau_n}{\tau_{n+1}}\right)^2 \frac{\alpha_m - 1 + 3(1-\gamma)(\alpha_f - 1)^2}{\alpha_m} + 3\frac{\gamma \alpha_f^2}{\alpha_m} - 1.$$

The chemical reaction E5 As numerical example we again choose the chemical reaction problem E5. The parameter ρ is chosen to be 0, 1/4, and 1/2, resp. For $\rho > 1/2$ we obtain too many stepsize rejections and the algorithm becomes ineffective. We compare the generalised-α methods with other implicit and linear-implicit second order solvers like ROS2 (see [VSBH99]), ROS2S (see [HHR12]), and the method of Ellsiepen (see [EH01]). It can be observed from Figure 3 that the generalised-α with the new stepsize controller is not as effective as the other second order methods.

5 Summary and Outlook

In this article we gave a short analysis of the generalised-α method for first and second order ODEs and introduced a new adaptive timestep control. In case of the onestep versions this controller is better than other adaptive second order methods. For multistep methods the adaptivity is more complicated. We have seen that only in the case $\rho \in [0, 1/2]$ an effective method is achieved. In future works adaptivity should also be possible if $\rho > 1/2$.

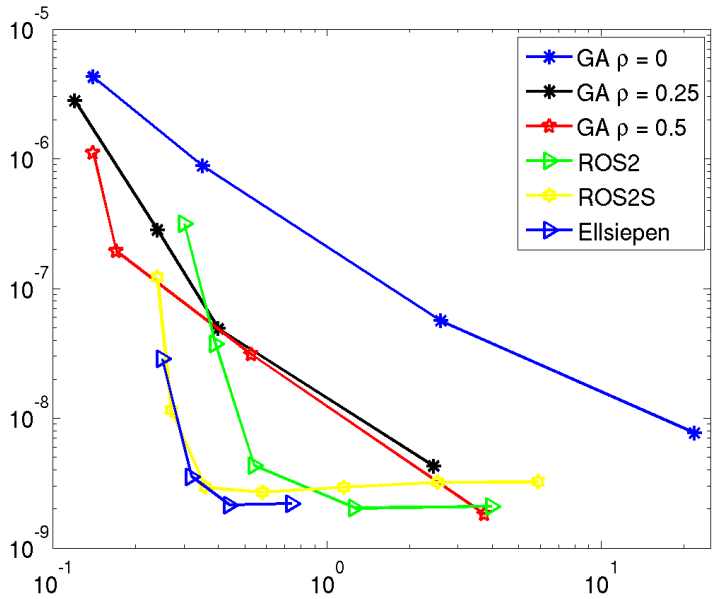

Figure 3: Comparison of generalised-α methods for first order ODEs: CPU time versus error

REFERENCES

[CH93] J. Chung and G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *J. Appl. Mech.*, 60(2):371–375, 1993.

[DP03] W. Dettmer and D. Perić. An analysis of the time integration algorithms for the finite element solutions of incompressible Navier-Stokes equations based on a stabilised formulation. *Comput. Methods Appl. Mech. Engrg.*, 192(9-10):1177–1226, 2003.

[EBB02] S. Erlicher, L. Bonaventura, and O. S. Bursi. The analysis of the generalized-α method for non-linear dynamic problems. *Comput. Mech.*, 28(2):83–104, 2002.

[EH01] P. Ellsiepen and S. Hartmann. Remarks on the interpretation of current non-linear finite-element-analyses as differential-algebraic equations. *International Journal for Numerical Methods in Engineering*, 51:679–707, 2001.

[EHL75] W.H. Enright, T.E. Hull, and B. Lindberg. Comparing numerical methods for stiff systems of O.D.E:s. *BIT, Nord. Tidskr. Inf.-behandl.*, 15:10–48, 1975.

[GLS88] K. Gustafsson, M. Lundh, and G. Söderlind. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT*, 28(2):270–287, 1988.

[GS00] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method*. Wiley, Chichester, 2000.

[HHR12] Ahmad-Wahadj Hamkar, Stefan Hartmann, and Joachim Rang. A stiffly accurate Rosenbrock-type method of order 2. *Appl. Num. Math.*, 62(12):1837–1848, 2012.

[HNW93] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations. I: Nonstiff problems.*, volume 8 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1993.

[HW96] E. Hairer and G. Wanner. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1996.

[JR10] Volker John and Joachim Rang. Adaptive time step control for the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Eng.*, 199:514–524, 2010.

[JWH00] Kenneth E. Jansen, Christian H. Whiting, and Gregory M. Hulbert. A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Comput. Methods Appl. Mech. Eng.*, 190(3-4):305–319, 2000.

[Lan01] J. Lang. *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems*, volume 16 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2001.

[Ran04] Joachim Rang. *Stability estimates and numerical methods for degenerate parabolic differential equations*. PhD thesis, Institut für Mathematik, TU Clausthal, 2004. appeared also as book from Papierflieger Verlag Clausthal-Zellerfeld.

[SW92] K. Strehmel and R. Weiner. *Linear-implizite Runge–Kutta-Methoden und ihre Anwendung*, volume 127 of *Teubner-Texte zur Mathematik*. Teubner, Stuttgart, 1992.

[VSBH99] J.G. Verwer, E.J. Spee, J.G. Blom, and W. Hundsdorfer. A second-order Rosenbrock method applied to photochemical dispersion problems. *SIAM J. Sci. Comput.*, 20(4):1456–1480, 1999.