A Brief Note on the Nonlinear Systems Arise from Stiff Equations

Yingbo Ma
mayingbo5@gmail.com

May 31, 2020

1 Diagonally Implicit Runge-Kutta Methods

Runge-Kutta methods and linear multistep methods are two major kinds of methods that numerically solve ordinary differential equations (ODE) and differential-algebraic equations (DAEs). We are going to focus on DAEs in the mass matrix form, since ODE is a subset of DAE. Mass matrix DAEs are in the form of

$$M\frac{\mathrm{d}u}{\mathrm{d}t} = f(u,t), \quad u(a) = u_a \in \mathbb{R}^m, \quad t \in [a,b]. \tag{1}$$

Fully implicit methods and the implementation details of Newton iteration are covered in the FIRK note. We will formulate the Newton iteration for diagonally implicit Runge-Kutta (DIRK) methods in this section.

An s-stage DIRK method in general can be written as

$$u_{n+1} = u_n + h\sum_{i=1}^{s} b_i k_i, \tag{2}$$

where

$$Mk_i = f\left(u_n + h\sum_{j=1}^{i} a_{ij}k_j, t_n + c_i h\right), \tag{3}$$

h is the step size, and a_{ij}, b_i, and c_j are scalar constants for each i and j.

We can isolate the implicit equation from constants and get

$$Mk_i = f\left(\underbrace{u_n + h\left(\sum_{j=1}^{i-1} a_{ij}k_j\right)}_{\text{constant part}} + \underbrace{ha_{ii}k_i}_{\text{implicit part}}, t_n + c_i h\right) \tag{4}$$

We can further simplify the RK expression by introducing $z_i = hk_i$ and let the constant part to be `innertmp`, then we have

$$u_{n+1} = u_n + \sum_{i=1}^{s} b_i z_i \tag{5}$$

$$Mz_i = hf(\texttt{innertmp} + a_{ii}z_i, t_n + c_i h). \tag{6}$$

Hence the Newton iteration for solving eq. (6) is

$$(-M + ha_{ii}J)\Delta_i^j = hf\Big(\texttt{innertmp} + a_{ii}z_i^j, t_n + c_i h\Big) - Mz_i^j \tag{7}$$

$$z_i^{j+1} = z_i^j - \Delta_i^j \tag{8}$$

After the W-transformation [1, Chapter IV.5], we obtain

$$\left(-\frac{1}{ha_{ii}}M + J\right)\Delta_i^j = \frac{1}{ha_{ii}}\Big(hf\Big(\texttt{innertmp} + a_{ii}z_i^j, t_n + c_i h\Big) - Mz_i^j\Big) \tag{9}$$

$$z_i^{j+1} = z_i^j - \Delta_i^j \tag{10}$$

The above formation is the generalization of the Newton iteration proposed by Hosea and Shampine [2].

2 Linear Multistep Methods

The general form for an s-step linear multistep method (LMM) is

$$M\sum_{i=0}^{s}\alpha_i u_{n-i} = h\sum_{i=0}^{s}\beta_i f(u_{n-i}, t_{n-i}). \tag{11}$$

We can separate the constant terms, and get

$$\alpha_0 M u_n = \underbrace{\left(h\sum_{i=1}^{s}\beta_i f(u_{n-i}, t_{n-i})\right) - \left(M\sum_{i=1}^{s}\alpha_i u_{n-i}\right)}_{\text{constant part}} + h\beta_0 f(u_n, t_n). \tag{12}$$

Let the constant part be `outertmp`, then we have

$$\alpha_0 M u_n = \texttt{outertmp} + h\beta_0 f(u_n, t_n). \tag{13}$$

Naturally, the Newton iteration is

$$(-\alpha_0 M + h\beta_0 J)\Delta^j = \texttt{outertmp} + h\beta_0 f(u_n^j, t_n) - \alpha_0 M u_n^j \tag{14}$$

$$u_n^{j+1} = u_n^j - \Delta^j. \tag{15}$$

After the W-transformation [1, Chapter IV.5], we obtain

$$\left(-\frac{\alpha_0}{h\beta_0}M + J\right)\Delta^j = \frac{1}{h\beta_0}\big(\texttt{outertmp} + h\beta_0 f(u_n^j, t_n) - \alpha_0 M u_n^j\big) \tag{16}$$

$$u_n^{j+1} = u_n^j - \Delta^j. \tag{17}$$

2.1 Nordsieck Formulation

For the sake of completeness, we shall demonstrate the Nordsieck formulation too. The fundamental object that we need to solve for is the difference between the predictor and the corrector, i.e. $u_n - u_{n(0)}$, where $\{\cdot\}_{n(0)}$ denotes the predictor. The general form of the nonlinear equation is

$$M(u_n - u_{n(0)}) = h\chi\left(f(u_n, t_n) - f(u_{n(0)}, t_n)\right), \tag{18}$$

where χ is a scalar constant. Let's separate the constant part,

$$Mu_n = h\chi f(u_n, t_n) + \underbrace{Mu_{n(0)} - h\chi f(u_{n(0)}, t_n)}_{\text{constant part}}, \tag{19}$$

The Newton iteration is then

$$(-M + h\chi J)\Delta^j = \texttt{outertmp} + h\chi f(u_n^j, t_n) - Mu_n^j, \tag{20}$$

$$u_n^{j+1} = u_n^j - \Delta^j. \tag{21}$$

After the W-transformation [1, Chapter IV.5], we obtain

$$\left(-\frac{1}{h\chi}M + J\right)\Delta^j = \frac{1}{h\chi}\big(\texttt{outertmp} + h\chi f(u_n^j, t_n) - Mu_n^j\big), \tag{22}$$

$$u_n^{j+1} = u_n^j - \Delta^j. \tag{23}$$

References

[1] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 2010.

[2] M. Hosea and L. Shampine, *Analysis and implementation of tr-bdf2*, Applied Numerical Mathematics, 20 (1996), pp. 21–37.