Numerical Analysis and Simulation I

—

Ordinary Differential Equations

Michael Günther

Lecture in Winter Term 2017/18

University of Wuppertal

Applied Mathematics/Numerical Analysis

Contents:

Literature:

J. Stoer, R. Bulirsch: Introduction to Numerical Analysis. Springer, Berlin 2002. (Chapter 7)

J. Stoer, R. Bulirsch: Numerische Mathematik 2. Springer, Berlin 2005. (Kapitel 7)

A. Quarteroni, R. Sacco, F. Saleri: Numerical Mathematics. Springer, New York 2007. (Chapter 11)

Contents

Chapter 1

Methods for Differential Algebraic Equations

We consider initial values problems of systems of differential algebraic equations (DAEs), i.e., a mixture of ordinary differential equations and algebraic equations. Such mathematical models are typically large in technical applications.

1.1 Implicit ODEs

We observe implicit systems of ordinary differential equations, since they represent a first step towards differential algebraic equations. Consider the initial value problem

$$My'(x) = f(x, y(x)), \quad y(x_0) = y_0 \tag{1.1}$$

with unknown solution $y : \mathbb{R} \to \mathbb{R}^n$ and right-hand side $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$. Let $M \in \mathbb{R}^{n \times n}$ be a constant matrix with $M \neq I$. Often M is called the mass matrix. If M is the identity matrix, then the system (1.1) represents explicit ODEs. We distinguish two cases:

M regular:	(1.1) is a system of implicit ordinary differential equations,
M singular:	(1.1) is a system of differential algebraic equations.

In this section, we assume the case of implicit ODEs. Consequently, we can transform the system (1.1) into the explicit system

$$y'(x) = M^{-1} f(x, y(x)). \tag{1.2}$$

Each evaluation of the new right-hand side demands the solution of a linear system with the matrix M now. For example, the explicit Euler method yields the formula

$$y_1 = y_0 + hM^{-1} f(x_0, y_0).$$

Thus a linear system with matrix M has to be solved in each step of the integration. A corresponding LU-decomposition has to be calculated just once. Using an explicit Runge-Kutta method, we obtain a sequence of linear systems, which have to be solved for each increment, i.e.,

$$Mk_i = f\left(x_0 + c_i h, y_0 + h \sum_{j=1}^{i-1} a_{ij} k_j\right) \quad \text{for } i = 1, \ldots, s.$$

However, implicit ODEs are often stiff. Hence implicit methods have to be used. For example, the implicit Euler method applied to the system (1.2) yields the nonlinear system

$$y_1 = y_0 + hM^{-1} f(x_1, y_1)$$

for the unknown value y_1. Considering the nonlinear system

$$y_1 - hM^{-1} f(x_1, y_1) - y_0 = 0,$$

the corresponding simplified Newton iteration reads

$$\begin{aligned} (I - hM^{-1}\mathrm{D}f(x_1, y_1^{(0)}))\Delta y_1^{(\nu)} &= -y_1^{(\nu)} + hM^{-1} f(x_1, y_1^{(\nu)}) + y_0, \\ y_1^{(\nu+1)} &= y_1^{(\nu)} + \Delta y_1^{(\nu)}, \end{aligned}$$

where $\mathrm{D}f = \frac{\partial f}{\partial y}$ denotes the Jacobian matrix of f. We multiply the equation of the Newton iteration with M and achieve the equivalent formulation

$$(M - h\mathrm{D}f(x_1, y_1^{(0)}))\Delta y_1^{(\nu)} = M(y_0 - y_1^{(\nu)}) + hf(x_1, y_1^{(\nu)}). \tag{1.3}$$

Thus one linear system has to be solved for both explicit and implicit ODEs in each step of the iteration. Just an additional matrix-vector multiplication is necessary on the right-hand side of (1.3).

Likewise, an implicit Runge-Kutta method applied to (1.1) or (1.2) exhibits the relations

$$Mk_i = f\left(x_0 + c_i h, y_0 + h\sum_{j=1}^{s} a_{ij}k_j\right) \quad \text{for } i = 1, \ldots, s. \tag{1.4}$$

Given a nonlinear function f, a nonlinear system of sn equations for the unknown increments has to be solved as for explicit ODEs.

Hence the computational effort for implicit ODEs is not significantly higher than for explicit ODEs in case of implicit methods. The situation becomes more complicated, if the matrix M is not constant but depends on the independent variable or the unknown solution.

We distinguish the following cases (with increasing complexity):

- linear-implicit system of ODEs with constant mass matrix:
  $My'(x) = f(x, y(x))$
- linear-implicit system of ODEs with non-constant mass matrix:
  $M(x)y'(x) = f(x, y(x))$
- quasilinear implicit system of ODEs:
  $M(y(x))y'(x) = f(x, y(x))$ or $M(x, y(x))y'(x) = f(x, y(x))$
- fully implicit system of ODEs:
  $F(y'(x), y(x), x) = 0$,
  $F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, $(z, y, x) \mapsto F(z, y, x)$, $\det\left(\frac{\partial F}{\partial z}\right) \neq 0$

For an example of an implicit system of ODEs, see the Colpitts oscillator introduced in Sect. **??**. The involved mass matrix is constant and regular. The system of ODEs exhibits a strongly stiff behaviour.

1.2 Linear DAEs

In this section, we consider linear systems of differential algebraic equations

$$Ay'(x) + By(x) = s(x), \qquad y(x_0) = y_0 \tag{1.5}$$

with unknown solution $y : \mathbb{R} \to \mathbb{R}^n$ and given input signal $s : \mathbb{R} \to \mathbb{R}^n$. We assume that the matrices $A, B \in \mathbb{R}^{n\times n}$ are constant. For $\det(A) \neq 0$, we obtain implicit ODEs, whereas $\det(A) = 0$ implies DAEs.

For simplicity, we assume $\det(B) \neq 0$ in the following. Stationary solutions of the DAEs (1.5) with some constant input $s \equiv s_0$ are characterised by $y' \equiv 0$. Hence a unique stationary solution is given by $y_0 = B^{-1}s_0$ in case of $\det(B) \neq 0$. We transform the system (1.5) to the equivalent system

$$B^{-1}Ay'(x) + y(x) = B^{-1}s(x). \tag{1.6}$$

We use $B^{-1}A = T^{-1}JT$ with the Jordan form J and the regular transformation matrix $T \in \mathbb{R}^{n\times n}$. Thus the system (1.6) is transformed to

$$\begin{aligned} TB^{-1}Ay'(x) + Ty(x) &= TB^{-1}s(x) \\ TB^{-1}AT^{-1}Ty'(x) + Ty(x) &= TB^{-1}s(x) \\ J(Ty(x))' + Ty(x) &= TB^{-1}s(x). \end{aligned} \tag{1.7}$$

The Jordan matrix J can be ordered such that it exhibits the form

$$J = \begin{pmatrix} R & 0 \\ 0 & N \end{pmatrix}, \qquad \begin{matrix} R \in \mathbb{R}^{n_1\times n_1}, \\ N \in \mathbb{R}^{n_2\times n_2}, \end{matrix} \qquad n_1 + n_2 = n, \tag{1.8}$$

where R contains all eigenvalues not equal to zero ($\det(R) \neq 0$) and N includes the eigenvalues equal to zero ($\det(N) = 0$). More precisely, N is a stricly upper triangular matrix. Hence N is nilpotent, i.e.,

$$N^{k-1} \neq 0, \quad N^k = 0 \qquad \text{for some } \; k \leq n_2. \tag{1.9}$$

We call k the nilpotency index of the linear DAE system (1.5). Since $\det(A) = 0$ holds, it follows $n_2 \geq 1$ and $k \geq 1$. The corresponding partitioning of the solution and the right-hand side reads

$$Ty = \begin{pmatrix} u \\ v \end{pmatrix}, \qquad TB^{-1}s = \begin{pmatrix} p \\ q \end{pmatrix} \tag{1.10}$$

with $u, p : \mathbb{R} \to \mathbb{R}^{n_1}$ and $v, q : \mathbb{R} \to \mathbb{R}^{n_2}$. Hence the system (1.5) is decoupled in two parts

$$\begin{aligned} Ru'(x) + u(x) &= p(x), \\ Nv'(x) + v(x) &= q(x). \end{aligned} \tag{1.11}$$

Since $\det(R) \neq 0$ holds, the first part represents an implicit ODE for the part u, which is equivalent to the linear explicit ODE

$$u'(x) = -R^{-1}u(x) + R^{-1}p(x).$$

The second part can be written as

$$\begin{aligned} v(x) &= q(x) - Nv'(x), \\ v^{(l)}(x) &= q^{(l)}(x) - Nv^{(l+1)}(x). \end{aligned}$$

We obtain successively together with $N^k = 0$

$$\begin{aligned} v(x) &= q(x) - Nv'(x), \\ &= q(x) - Nq'(x) + N^2v''(x) \\ &= q(x) - Nq'(x) + N^2q''(x) - N^3v^{(3)}(x) \\ &= \cdots \\ &= q(x) - Nq'(x) + N^2q''(x) - \cdots + (-1)^k N^k v^{(k+1)}(x) \\ &= \sum_{i=0}^{k-1} (-1)^i N^i q^{(i)}(x). \end{aligned} \tag{1.12}$$

Thus we achieve an algebraic relation for the part v depending on the higher derivatives of the input. The special case $N = 0$ yields $v(x) = q(x)$. We call u and v the differential and algebraic part, respectively. In particular, the initial value of the algebraic part follows from the input via

$$v(x_0) = \sum_{i=0}^{k-1} (-1)^i N^i q^{(i)}(x_0). \tag{1.13}$$

In contrast, the initial value $u(x_0) \in \mathbb{R}^{n_1}$ of the differential part can be chosen arbitrarily.

Differentiating the relation (1.12) one more time yields

$$v'(x) = \sum_{i=0}^{k-1} (-1)^i N^i q^{(i+1)}(x). \tag{1.14}$$

Hence by differentiating the system (1.5) k times, we obtain a system of ODEs for the part v.

If the source term includes a perturbation, i.e., the right-hand side changes into $\hat{s}(x) = s(x) + \delta(x)$, then the algebraic part reads

$$\hat{v}(x) = \sum_{i=0}^{k-1} (-1)^i N^i q^{(i)}(x) + \sum_{i=0}^{k-1} (-1)^i N^i \tilde{\delta}^{(i)}(x)$$

with transformed perturbations $\tilde{\delta} : \mathbb{R} \to \mathbb{R}^{n_2}$ due to (1.10). Thus also higher derivates of the perturbation influence the solution of the linear DAE system in case of $k > 1$.

Conclusions:

- To guarantee the existence of solutions of the linear DAEs (1.5), the right-hand side s has to be sufficiently smooth, namely $s \in C^{k-1}$. The algebraic part v may be just continuous and not smooth.
- Derivatives of perturbations in the right-hand side influence the solution of a perturbed system in case of nilpotency index $k \geq 2$.
- The initial values $y(x_0) = y_0$ of the system (1.5) cannot be chosen arbitrarily. A consistent choice is necessary regarding (1.13).

If the matrix B is singular, existence and uniqueness of solutions can still be obtained in case of a regular matrix pencil, i.e., $\det(\lambda A + B) \not\equiv 0$ holds. Take a fixed $\lambda \in \mathbb{R}$ such that $\det(\lambda A + B) \neq 0$. Now we transform the system (1.5) into

$$\begin{aligned} A(y'(x) - \lambda y(x)) + (\lambda A + B)y(x) &= s(x), \\ (\lambda A + B)^{-1} A(y'(x) - \lambda y(x)) + y(x) &= (\lambda A + B)^{-1} s(x). \end{aligned} \tag{1.15}$$

We use the Jordan form $(\lambda A+B)^{-1}A = T^{-1}JT$ with the structure (1.8). The transformation is analogue to (1.10). Consequently, the DAE system (1.5) is decoupled into the two parts

$$\begin{aligned} R(u'(x) - \lambda u(x)) + u(x) &= p(x), \\ N(v'(x) - \lambda v(x)) + v(x) &= q(x). \end{aligned} \tag{1.16}$$

The first part is equivalent to an explicit system of ODEs again. The second part can be written in the form

$$v(x) = (I - \lambda N)^{-1}q(x) - (I - \lambda N)^{-1}Nv'(x) = \tilde{q}(x) - \tilde{N}v'(x)$$

with $\tilde{q} := (I - \lambda N)^{-1}q$ and $\tilde{N} := (I - \lambda N)^{-1}N$. We arrange a von Neumann series to represent the inverse matrix

$$(I - \lambda N)^{-1} = \sum_{j=0}^{\infty} \lambda^j N^j = \sum_{j=0}^{k-1} \lambda^j N^j,$$

since $N^j = 0$ holds for all $j \geq k$. It follows

$$\tilde{N} = (I - \lambda N)^{-1}N = \sum_{j=0}^{k-2} \lambda^j N^{j+1}$$

and thus $\tilde{N}^{k-1} \neq 0,\ \tilde{N}^k = 0$ with the same k as in (1.9). Accordingly, we obtain the same results as in the case $\det(B) \neq 0$. However, we have not shown that the definition of the index k is unique in this case, i.e., k is independent of the choice of λ.

If $\det(\lambda A + B) \equiv 0$ holds, then either existence or uniqueness of solutions to the linear DAE system (1.5) is violated.

1.3 Index Concepts

The index of a system of DAEs represents an integer number, which characterises the qualitative differences of the DAE system in comparison to a system of ODEs. We distinguish the two cases

$$\begin{aligned} \text{index } k = 0: &\quad \text{system of ODEs,} \\ \text{index } k \geq 1: &\quad \text{system of DAEs.} \end{aligned}$$

The higher the index, the more the system of DAEs behaves different form a system of ODEs.

Several concepts for defining the index exist. We discuss two important approaches, namely the differential index and the perturbation index.

To define the index, we consider a general nonlinear system of differential algebraic equations

$$F(y'(x), y(x), x) = 0, \qquad y(x_0) = y_0 \tag{1.17}$$

with unknown solution $y : \mathbb{R} \to \mathbb{R}^n$ and $F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$. The predetermined initial values have to be consistent.

Differential Index

The system (1.17) represents ordinary differential equations, if the Jacobian matrix $\frac{\partial F}{\partial y'}$ is regular. We consider the extended system

$$\begin{aligned} F(y'(x), y(x), x) &= 0 \\ \frac{\mathrm{d}}{\mathrm{d}x} F(y'(x), y(x), x) &= 0 \\ \frac{\mathrm{d}^2}{\mathrm{d}x^2} F(y'(x), y(x), x) &= 0 \\ &\vdots \\ \frac{\mathrm{d}^k}{\mathrm{d}x^k} F(y'(x), y(x), x) &= 0 \end{aligned} \tag{1.18}$$

with $(k+1)n$ equations, which is achieved by a subsequent differentiation. In most cases, an explicit system of ODEs for the unknown solution in the form

$$y'(x) = G(y(x), x)$$

can be constructed from a larger system (1.18) by algebraic manipulations.

Definition 1 *The differential index of the system of DAEs (1.17) is the smallest integer $k \geq 0$ such that an explicit system of ODEs for the solution y can be constructed by algebraic manipulations using the extended system (1.18)*

The special case $k = 0$ implies that the system (1.17) is equivalent to an explicit system of ODEs, i.e., it is not a DAE.

As example, we discuss a semi-explicit system of DAEs

$$\begin{aligned} y'(x) &= f(y(x), z(x)), \quad y : \mathbb{R} \to \mathbb{R}^{n_1}, \quad f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_1}, \\ 0 &= g(y(x), z(x)), \quad z : \mathbb{R} \to \mathbb{R}^{n_2}, \quad g : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_2}. \end{aligned} \tag{1.19}$$

The differential index of this system is always $k \geq 1$ provided that $n_2 > 0$. Differentiating the second part of the system yields

$$0 = \frac{\partial g}{\partial y} \cdot y'(x) + \frac{\partial g}{\partial z} \cdot z'(x) = \frac{\partial g}{\partial y} \cdot f(y(x), z(x)) + \frac{\partial g}{\partial z} \cdot z'(x).$$

If the Jacobian matrix $\frac{\partial g}{\partial z} \in \mathbb{R}^{n_2 \times n_2}$ is regular, then we obtain

$$z'(x) = -\left(\frac{\partial g}{\partial z}\right)^{-1} \cdot \frac{\partial g}{\partial y} \cdot f(y(x), z(x)).$$

Thus we achieve an explicit ODE for the solution y, z and the differential index results to $k = 1$. If the Jacobian matrix $\frac{\partial g}{\partial z}$ is singular, then the differential index satisfies $k \geq 2$ and further examinations are necessary.

This example indicates that the differential index possibly does not depend on the underlying system of DAEs only but also on the considered solution. Thus the same system may exhibit two different solutions with according indexes.

Perturbation Index

We observe a system of ODEs and a corresponding perturbed system

$$\begin{aligned} y'(x) &= f(x, y(x)), & y(x_0) &= y_0, \\ \hat{y}'(x) &= f(x, \hat{y}(x)) + \delta(x), & \hat{y}(x_0) &= \hat{y}_0. \end{aligned} \tag{1.20}$$

Let the function f be Lipschitz-continuous. We perform a similar analysis as in Sect. **??**. However, we do not apply Gronwall's lemma now. The equivalent integral equations of (1.20) read

$$y(x) = y_0 + \int_{x_0}^{x} f(s, y(s)) \, \mathrm{d}s, \quad \hat{y}(x) = \hat{y}_0 + \int_{x_0}^{x} f(s, \hat{y}(s)) + \delta(s) \, \mathrm{d}s.$$

We consider an interval $I := [x_0, x_{\text{end}}]$. Let $R := x_{\text{end}} - x_0$. Subtracting the integral equations yields the estimate in the maximum norm

$$\begin{aligned}
\|\hat{y}(x) - y(x)\| &= \left\| \hat{y}_0 - y_0 + \int_{x_0}^{x} f(s, \hat{y}(s)) - f(s, y(s)) + \delta(s) \, \mathrm{d}s \right\| \\
&\leq \|\hat{y}_0 - y_0\| + \int_{x_0}^{x} \|f(s, \hat{y}(s)) - f(s, y(s))\| \, \mathrm{d}s + \left\| \int_{x_0}^{x} \delta(s) \, \mathrm{d}s \right\| \\
&\leq \|\hat{y}_0 - y_0\| + L \int_{x_0}^{x} \|\hat{y}(s) - y(s)\| \, \mathrm{d}s + \left\| \int_{x_0}^{x} \delta(s) \, \mathrm{d}s \right\| \\
&\leq \|\hat{y}_0 - y_0\| + L(x - x_0) \max_{s \in I} \|\hat{y}(s) - y(s)\| + \left\| \int_{x_0}^{x} \delta(s) \, \mathrm{d}s \right\| \\
&\leq \|\hat{y}_0 - y_0\| + LR \max_{s \in I} \|\hat{y}(s) - y(s)\| + \max_{s \in I} \left\| \int_{x_0}^{s} \delta(u) \, \mathrm{d}u \right\|
\end{aligned}$$

for all $x \in I$. Taking the maximum over all $x \in I$ on the left-hand side yields (provided that $LR < 1$)

$$\max_{x \in I} \|\hat{y}(x) - y(x)\| \leq \frac{1}{1 - LR} \left(\|\hat{y}_0 - y_0\| + \max_{s \in I} \left\| \int_{x_0}^{s} \delta(u) \, \mathrm{d}u \right\| \right).$$

Hence just the difference in the initial values and the integral of the perturbation give a contribution to the discrepancy of the two solutions. Furthermore, it holds the estimate

$$\max_{x \in I} \|\hat{y}(x) - y(x)\| \leq \frac{1}{1 - LR} \left(\|\hat{y}_0 - y_0\| + R \max_{s \in I} \|\delta(s)\| \right).$$

Given a general nonlinear system of DAEs (1.17) and a corresponding solution y on $I := [x_0, x_{\text{end}}]$, we consider the perturbed system

$$F(\hat{y}'(x), \hat{y}(x), x) = \delta(x), \qquad \hat{y}(x_0) = \hat{y}_0 \tag{1.21}$$

with sufficiently smooth perturbation $\delta : I \to \mathbb{R}^n$.

Definition 2 *The perturbation index of the system (1.17) corresponding to the solution y on an interval I is the smallest integer $k \geq 1$ such that an estimate*

$$\|\hat{y}(x) - y(x)\| \leq C \left(\|\hat{y}_0 - y_0\| + \sum_{l=0}^{k-1} \max_{s \in I} \|\delta^{(l)}(s)\| \right)$$

exists with a constant $C > 0$ for sufficiently small right-hand side. The perturbation index is $k = 0$ if an estimate of the form

$$\|\hat{y}(x) - y(x)\| \leq C \left(\|\hat{y}_0 - y_0\| + \max_{s \in I} \left\| \int_{x_0}^{s} \delta(u) \; du \right\| \right)$$

holds.

It can be shown that the perturbation index is $k = 0$ if and only if the system (1.17) represents explicit or implicit ODEs.

Remark that a perturbation can be small itself but exhibit large derivatives. For example, we discuss the function

$$\begin{aligned} \delta(x) &= \varepsilon \sin(\omega x), \\ \delta'(x) &= \varepsilon \omega \cos(\omega x). \end{aligned}$$

It holds $|\delta(x)| \leq \varepsilon$ for arbitrary $\omega \in \mathbb{R}$. However, we obtain $|\delta'(x)| \leq \varepsilon\omega$, which becomes large in case of $\omega \gg 1$ even if $\varepsilon > 0$ is tiny.

In view of this property, the numerical simulation of DAE models becomes critical in case of perturbation index $k \geq 2$, since derivatives of perturbations are involved. DAE systems of index $k = 1$ are well-posed, whereas DAE systems of index $k \geq 2$ are (strictly speaking) ill-posed. The higher the perturbation index becomes, the more critical is this situation. However, modelling electric circuits can be done by DAEs with index $k \leq 2$. The models of mechanical systems exhibit DAEs with index $k \leq 3$. In practice, mathematical models based on DAE systems with perturbation index $k > 3$ are avoided.

From the numerical point of view, the perturbation index is more interesting than the differential index, since it characterises the expected problems in

numerical methods. The result of a numerical technique can be seen as the exact solution of a perturbed system of DAEs (backward analysis). It is often difficult to determine the perturbation index of a system of DAEs, whereas the differential index is easier to examine.

For linear systems (1.5), the differential index and the perturbation index coincide and are equal to the nilpotency index. For a general nonlinear system (1.17), the two index concepts can differ arbitrarily. However, the differential index is equal to the perturbation index in many technical applications.

Examples: Electric Circuits

We discuss the differential index of two systems of DAEs, which result from modelling an electric circuit by a network approach. The two circuits are shown in Fig. 1.

The first circuit is an electromagnetic oscillator, which has already been introduced in Sect. **??**. It consists of a capacitance C, an inductance L and a linear resistor R in parallel. The unknowns are the three currents I_C, I_L, I_R through the basic elements and the node voltage U depending on time. Each basic element is modelled by a current-voltage relation. Furthermore, Kirchhoff's current law is added. We obtain a linear system of DAEs

$$\begin{aligned} CU' &= I_C \\ LI_L' &= U \\ 0 &= U - RI_R \\ 0 &= I_C + I_L + I_R. \end{aligned} \tag{1.22}$$

We can eliminate the unknowns I_C, I_R such that a linear system of ODEs is achieved

$$\begin{aligned} CU' &= -I_L - \tfrac{1}{R}U \\ LI_L' &= U. \end{aligned} \tag{1.23}$$

Systems of the form (1.22) are arranged automatically by tools of computer aided design (CAD). In contrast, the advantageous description by ODEs

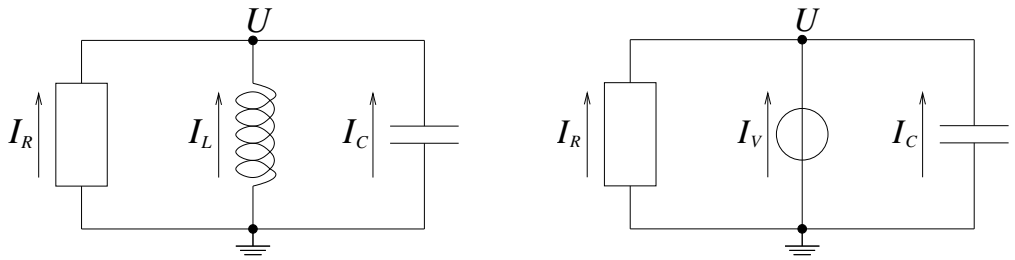


Figure 1: Example circuits.

like (1.23) has to be constructed by ourselves.

Differentiating the system (1.22) with respect to time yields

$$\begin{aligned} CU'' &= I_C' \\ LI_L'' &= U' \\ 0 &= U' - RI_R' \\ 0 &= I_C' + I_L' + I_R'. \end{aligned}$$

Hence we obtain an explicit system of ODEs for the unknowns

$$\begin{aligned} U' &= \tfrac{1}{C} I_C \\ I_L' &= \tfrac{1}{L} U \\ I_R' &= \tfrac{1}{R} U' = \tfrac{1}{RC} I_C \\ I_C' &= -I_L' - I_R' = -\tfrac{1}{L} U - \tfrac{1}{RC} I_C. \end{aligned}$$

Since just one differentiation is necessary to achieve this ODE system, the differential index of the DAE system (1.22) is $k = 1$.

Now we consider the second circuit, which consists of a capacitance C, an independent voltage source $V(t)$ and a linear resistor R. The corresponding DAE model reads

$$\begin{aligned} CU' &= I_C \\ 0 &= U - V(t) \\ 0 &= U - RI_R \\ 0 &= I_C + I_V + I_R. \end{aligned} \tag{1.24}$$

If the input voltage $V(t)$ is smooth, the solution can be calculated analytically

$$U = V(t), \quad I_R = \tfrac{1}{R}V(t), \quad I_C = CV'(t), \quad I_V = -CV'(t) - \tfrac{1}{R}V(t).$$

Furthermore, we arrange an explicit system of ODEs for the unknowns starting from the DAE system (1.24)

$$\begin{aligned} U' &= \tfrac{1}{C}I_C \\ I_R' &= \tfrac{1}{R}U' = \tfrac{1}{RC}I_C \\ I_C' &= CU'' = CV''(t) \\ I_V' &= -I_C' - I_R' = -CV''(t) - \tfrac{1}{RC}I_C. \end{aligned}$$

In this case, two differentiations of the system (1.24) with respect to time are required, since the relation $U'' = V''$ is used. Hence the differential index of the DAE system (1.24) is $k = 2$.

1.4 Methods for General Systems

In the next two sections, we outline the construction of numerical techniques for systems of DAEs. The numerical methods represent generalisations of corresponding schemes for systems of ODEs introduced in the previous chapters.

We consider initial value problems of fully implicit systems of DAEs (1.17), i.e., the most general form. The initial values have to be consistent with respect to the DAEs. We apply a grid $x_0 < x_1 < x_2 < \cdots < x_m$. Corresponding approximations $y_i \doteq y(x_i)$ of the solution are determined recursively by a numerical method.

Linear multistep methods

In case of systems of ODEs $y' = f(x, y)$, a linear multistep method is defined in (**??**) for equidistant step sizes. Since $y' = f$ holds, we can rewrite the

scheme as

$$\sum_{l=0}^{k} \alpha_l y_{i+l} = h \sum_{l=0}^{k} \beta_l z_{i+l}, \tag{1.25}$$

where $z_{i+l} = f(x_{i+l}, y_{i+l})$ represents an approximation of $y'(x_{i+l})$. In case of general DAE systems, this value is obtained by solving the nonlinear system (1.17). It follows the method

$$\begin{aligned} \sum_{l=0}^{k} \alpha_l y_{i+l} &= h \sum_{l=0}^{k} \beta_l z_{i+l} \\ F(z_{i+k}, y_{i+k}, x_{i+k}) &= 0 \end{aligned} \tag{1.26}$$

with the unknowns y_{i+k}, z_{i+k} in each step.

The BDF methods, see Sect. **??**, are suitable for solving systems of DAEs. The k-step BDF scheme reads

$$\sum_{l=0}^{k} \alpha_l y_{i+l} = h z_{i+k}.$$

(Remark that the numbering of the coefficients is opposite to (**??**)). Hence we can replace z_{i+l} in $F(z_{i+k}, y_{i+k}, x_{i+k})$ by this formula. Consequently, the method (1.26) exhibits the simple form

$$F\left(\frac{1}{h}\sum_{l=0}^{k} \alpha_l y_{i+l}, y_{i+k}, x_{i+k}\right) = 0$$

with then unknown y_{i+k}. The BDF methods for fully implicit DAE systems (1.17) are implemented in the FORTRAN code DASSL (Differential Algebraic System SoLver) by Petzold (1982).

Although the trapezoidal rule represents a one-step method, we can write it in the form (1.25)

$$y_{i+1} - y_i = h\left[\tfrac{1}{2} z_i + \tfrac{1}{2} z_{i+1}\right] \quad \Rightarrow \quad z_{i+1} = -z_i + \tfrac{2}{h}(y_{i+1} - y_i).$$

Inserting z_{i+1} in $F(z_{i+1}, y_{i+1}, x_{i+1})$ yields the scheme

$$F\left(-z_i + \tfrac{2}{h}(y_{i+1} - y_i), y_{i+1}, x_{i+1}\right) = 0 \tag{1.27}$$

with the unknown y_{i+1}. The value z_i is known from the previous step.

Runge-Kutta Methods

We consider a Runge-Kutta method given in (**??**) for systems of ODEs. An approximation of the solution at the intermediate points is achieved via

$$y(x_0 + c_i h) \doteq y_0 + h \sum_{j=1}^{s} a_{ij} k_j \qquad \text{for } \; i = 1, \ldots, s.$$

Due to $y' = f$, the increments k_i represent approximations of the derivatives $y'(x_0 + c_i h)$, i.e.,

$$y'(x_0 + c_i h) \doteq k_i = f\left(x_0 + c_i h, y_0 + h \sum_{j=1}^{s} a_{ij} k_j\right) \qquad \text{for } \; i = 1, \ldots, s.$$

Now we solve general DAE systems. We apply the nonlinear system (1.17) for the determination of these derivatives again. It holds exactly

$$F\left(y'(x_0 + c_i h), y(x_0 + c_i h), x_0 + c_i h\right) = 0 \qquad \text{for } \; i = 1, \ldots, s.$$

It follows the numerical method

$$\begin{aligned} &F\left(k_i, y_0 + h\sum_{j=1}^{s} a_{ij} k_j, x_0 + c_i h\right) = 0 \quad \text{for } \; i = 1, \ldots, s\ , \\ &y_1 = y_0 + h\sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{1.28}$$

In case of systems $My' = f(x, y)$, the technique (1.28) results just to (1.4).

As example, we consider the trapezoidal rule again. The coefficients of this Runge-Kutta scheme are $c_1 = 0$, $c_2 = 1$, $a_{11} = a_{12} = 0$, $a_{21} = a_{22} = b_1 = b_2 = \frac{1}{2}$. It follows

$$\begin{aligned} F(k_1, y_0, x_0) &= 0 \\ F(k_2, y_0 + h(\tfrac{1}{2}k_1 + \tfrac{1}{2}k_2), x_1) &= 0 \\ y_0 + h(\tfrac{1}{2}k_1 + \tfrac{1}{2}k_2) &= y_1. \end{aligned}$$

If we replace k_2 by the other values, then the second equation changes into

$$F\left(-k_1 + \frac{2}{h}(y_1 - y_0), y_1, x_1\right) = 0.$$

Hence the method coincides with (1.27).

1.5 Methods for Semi-Explicit Systems

In case of semi-explicit systems of DAEs (1.19), methods for systems of ODEs can be generalised immediately. Two approaches exist for this purpose.

Direct Approach (ε-embedding)

The semi-explicit system of DAEs (1.19) is embedded into a family of systems of ODEs

$$\begin{array}{rcl} y'(x) &=& f(y(x), z(x)), \\ \varepsilon z'(x) &=& g(y(x), z(x)), \end{array} \quad \Leftrightarrow \quad \begin{array}{rcl} y'(x) &=& f(y(x), z(x)), \\ z'(x) &=& \frac{1}{\varepsilon} g(y(x), z(x)). \end{array} \tag{1.29}$$

The original DAE is recovered for $\varepsilon \to 0$. Systems of the form (1.29) are also called singularly perturbed systems. Systems of DAEs can be seen as the limit case of stiff systems, where the amount of stiffness becomes infinite.

As an example, we consider the Van-der-Pol oscillator

$$y'' + \mu^2((y^2 - 1)y' + y) = 0 \quad \Leftrightarrow \quad \varepsilon y'' + (y^2 - 1)y' + y = 0$$

with parameter $\varepsilon = \frac{1}{\mu^2}$. The system becomes more and more stiff in case of $\varepsilon \to 0$. We investigate the corresponding system of first order

$$y' = z, \quad \varepsilon z' = -(y^2 - 1)z - y.$$

Setting $\varepsilon = 0$ implies the semi-explicit DAE system

$$y' = z, \quad 0 = -(y^2 - 1)z - y.$$

It follows

$$y' = z = \frac{y}{1 - y^2} \quad \text{for } y \neq \pm 1.$$

We can solve this ODE for y partly and achieve (with a constant $C \in \mathbb{R}$)

$$\ln|y(x)| - \tfrac{1}{2}y(x)^2 = x + C.$$

If a solution of the semi-explicit DAEs reaches a singularity $y = \pm 1$, then the existence of the solution is violated. In contrast, the solution of the ODE continues to exist and exhibits steep gradients at the singularity. This solution changes fastly from $y = 1$ to $y = -2$ and from $y = -1$ to $y = 2$. We apply the above relation to obtain an estimate of the period of the solution of the oscillator. Let $y(x_1) = 2$ and $y(x_2) = 1$, i.e., the solution changes slowly between x_1 and x_2. It follows

$$\ln 2 - 2 = x_1 + C, \quad \ln 1 - \tfrac{1}{2} = x_2 + C \quad \Rightarrow \quad x_2 - x_1 = -\ln 2 + \tfrac{3}{2}.$$

The period is $T \approx 2(x_2 - x_1) = 3 - 2\ln 2 \approx 1.6137$ in case of $\varepsilon \approx 0$. Numerical simulations confirm this estimate.

Now we can apply a numerical method for ODEs to the system (1.29). Implicit techniques typically have to be considered, since DAEs represent the limit of stiff systems of ODEs. Performing the limit $\varepsilon \to 0$ yields a method for the semi-explicit DAEs (1.19).

For example, the implicit Euler method implies

$$\begin{aligned} y_1 &= y_0 + hf(y_1, z_1), \\ z_1 &= z_0 + h\tfrac{1}{\varepsilon}g(y_1, z_1). \end{aligned}$$

The second equation is equivalent to

$$\varepsilon z_1 = \varepsilon z_0 + hg(y_1, z_1).$$

In the limit $\varepsilon \to 0$, we obtain the numerical method

$$\begin{aligned} y_1 &= y_0 + hf(y_1, z_1), \\ 0 &= g(y_1, z_1), \end{aligned} \tag{1.30}$$

which represents a nonlinear system for the unknown approximation y_1, z_1.

Indirect Approach (state space form)

For the semi-explicit DAEs (1.19), we consider the component z as the solution of a nonlinear system for given y, i.e.,

$$z(x) = \Phi(y(x)), \qquad g(y(x), \Phi(y(x))) = 0. \tag{1.31}$$

Due to the implicit function theorem, the regularity of the Jacobian matrix $\frac{\partial g}{\partial z}$ is sufficient for the existence and the local uniqueness of a continuous function $\Phi : U \to V$ with $U \subset \mathbb{R}^{n_1}, V \subset \mathbb{R}^{n_2}$. This condition corresponds to a semi-explicit DAE of differential index 1. Consequently, the differential part of the DAE depends only on y

$$y'(x) = f(y(x), \Phi(y(x))). \tag{1.32}$$

This system is called the state space form of the problem. Now we are able to use a method for ODEs directly to this system. In a numerical method, we have to evaluate the right-hand side of (1.32) for given values y. Each evaluation demands the solution of a nonlinear system (1.31).

As example, we apply the implicit Euler method again. It follows

$$\begin{aligned} y_1 &= y_0 + hf(y_1, \Phi(y_1)), \\ 0 &= g(y_1, \Phi(y_1)). \end{aligned} \tag{1.33}$$

Hence the resulting technique (1.33) is equivalent to the scheme (1.30) obtained by the direct approach in case of the implicit Euler method.

The direct and indirect approach represent just techniques to obtain a suggestion for a numerical method. The properties of the corresponding method for ODEs do not necessarily hold for the resulting scheme to solve DAEs. Hence an analysis of consistency and stability of the constructed numerical methods has still to be performed.

Runge-Kutta Methods

We investigate Runge-Kutta methods now, see Sect. **??**. The indirect approach is straightforward to apply. We obtain the formula

$$\begin{aligned} Y_i &= y_0 + h\sum_{j=1}^{s} a_{ij} f(Y_j, Z_j) \\ 0 &= g(Y_i, Z_i) \qquad\qquad \text{for } i = 1, \ldots, s \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i f(Y_i, Z_i). \end{aligned}$$

The value z_1 can be computed by solving the nonlinear system $g(y_1, z_1) = 0$.

The direct approach yields

$$\begin{aligned} Y_i &= y_0 + h\sum_{j=1}^{s} a_{ij} f(Y_j, Z_j) \\ \varepsilon Z_i &= \varepsilon z_0 + h\sum_{j=1}^{s} a_{ij} g(Y_j, Z_j) \qquad \text{for } \; i = 1, \ldots, s \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i f(Y_i, Z_i) \\ \varepsilon z_1 &= \varepsilon z_0 + h\sum_{i=1}^{s} b_i g(Y_i, Z_i). \end{aligned}$$

We assume that the matrix $A = (a_{ij})$ is regular in the following. Let $A^{-1} = (\omega_{ij})$. We transform the second equation into

$$h g(Y_i, Z_i) = \varepsilon \sum_{j=1}^{s} \omega_{ij} (Z_i - z_0) \qquad \text{for } \; i = 1, \ldots, s.$$

Accordingly, the fourth equation becomes

$$\varepsilon z_1 = \varepsilon z_0 + \varepsilon \sum_{i=1}^{s} b_i \left(\sum_{j=1}^{s} \omega_{ij} (Z_i - z_0) \right).$$

The limit $\varepsilon \to 0$ yields the method

$$\begin{aligned} Y_i &= y_0 + h\sum_{j=1}^{s} a_{ij} f(Y_j, Z_j) \\ 0 &= g(Y_i, Z_i) \qquad\qquad\qquad \text{for } \; i = 1, \ldots, s \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i f(Y_i, Z_i) \\ z_1 &= \left(1 - \sum_{i,j=1}^{s} b_i \omega_{ij} \right) z_0 + \sum_{i,j=1}^{s} b_i \omega_{ij} Z_j. \end{aligned} \tag{1.34}$$

The scheme (1.34) of the direct approach coincides with the method (1.28) applied to semi-explicit DAEs in case of a regular coefficient matrix A.

In (1.34), the involved coefficient satisfies

$$1 - \sum_{i,j=1}^{s} b_i \omega_{ij} = \lim_{z \to \infty} R(z) =: R(\infty)$$

with the stability function $R(z) = 1 + zb^\top (I - zA)^{-1} \mathbb{1}$ of the Runge-Kutta method from (**??**).

A Runge-Kutta method is called stiffly accurate, if it holds

$$a_{si} = b_i \qquad \text{for } \; i = 1, \ldots, s.$$

For example, the RadauIIa schemes are stiffly accurate ($s = 1$: implicit Euler). In this case, it follows $y_1 = Y_s$ and $z_1 = Z_s$, i.e., the direct approach coincides with the indirect approach.

Given a Runge-Kutta method with order of consistency p in case of ODEs, we are interested in the order of convergence in case of semi-explicit DAEs. Let q be the stage order of the method, i.e., $Y_i - y(x_0 + c_i h) = \mathcal{O}(h^{q+1})$ for all i in case of ODEs. We consider semi-explicit DAEs (1.19) with differential index 1. Using the indirect approach, the order of convergence is equal p for both differential part y and algebraic part z. The direct approach implies the global errors

$$y_N - y(x_{\text{end}}) = \mathcal{O}(h^p), \qquad z_N - z(x_{\text{end}}) = \mathcal{O}(h^r)$$

with

(i) $r = p$ for stiffly accurate methods ($R(\infty) = 0$),

(ii) $r = \min(p, q+1)$ for $-1 \leq R(\infty) < 1$,

(iii) $r = \min(p-1, q)$ for $R(\infty) = 1$,

(iv) divergence if $|R(\infty)| > 1$.

For methods, which are not stiffy accurate, an order reduction appears ($r < p$). The A-stability of a Runge-Kutta technique is sufficient (not necessary) for the convergence of the algebraic part.

Now we consider DAEs of index 2. Thereby, we analyse semi-explicit DAEs of the form

$$\begin{aligned} y' &= f(y,z), \\ 0 &= g(y). \end{aligned} \tag{1.35}$$

The system cannot have the differential index 1, since it holds $\frac{\partial g}{\partial z} \equiv 0$. The system (1.35) exhibits the differential index $k = 2$ if the matrix $\frac{\partial g}{\partial y}\frac{\partial f}{\partial z}$ is always regular. It can be shown that a system of the form (1.35) has the differential index $k = 2$ if and only if the perturbation index is $k = 2$.

The indirect approach cannot be applied to (1.35), since the function Φ from (1.31) is not defined. In contrast, the direct approach yields the same Runge-Kutta method (1.34) as in the case of index 1 (just replace $g(y,z)$ by the special case $g(y)$). The analysis of convergence becomes more complicated in case of differential index 2. We just cite the results for the Gauss and the Radau methods with s stages. The following table illustrates the orders of the local errors and the global errors.

	local error for ODEs	global error for ODEs	local error		global error	
			y	z	y	z
Gauss, s odd	$2s+1$	$2s$	$s+1$	s	$s+1$	$s-1$
Gauss, s even	$2s+1$	$2s$	$s+1$	s	s	$s-2$
RadauIA	$2s$	$2s-1$	s	$s-1$	s	$s-1$
RadauIIA	$2s$	$2s-1$	$2s$	s	$2s-1$	s

We recognise that the behaviour of the methods is much more complex than in the case of index 1. The RadauIIA schemes exhibits the best convergence properties within these examples, since these techniques are stiffly accurate.

For further reading on numerical methods for systems of DAEs, see
E. Hairer, G. Wanner: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. (2nd Ed.) Springer, Berlin, 1996.

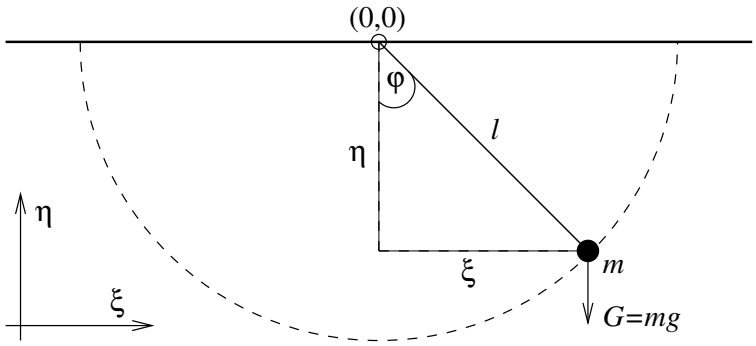


Figure 2: Mathematical Pendulum.

1.6 Illustrative Example: Mathematical Pendulum

Fig. 2 demonstrates the problem of the mathematical pendulum. We desire a mathematical model, which describes the positions ξ, η of the mass m with respect to time. On the one hand, Newton's law states $F = mx''$ for the force F acting on the mass m and for the space variables $x := (\xi, \eta)^\top$. On the other hand, the force F is the sum of the gravitational force $G = (0, mg)^\top$ with gravitation constant g and the force $F_r = -2\lambda x$ in direction of the rope, where λ represents a time-dependent scalar. The force F_r causes that the mass moves on a circle with radius l, since the constant l denotes the length of the rope. It follows

$$\begin{aligned} m\xi''(t) &= -2\lambda(t)\xi(t) \\ m\eta''(t) &= -2\lambda(t)\eta(t) - mg. \end{aligned}$$

A semi-explicit system of DAEs including five equations results

$$\begin{aligned} \xi'(t) &= u(t) \\ \eta'(t) &= v(t) \\ u'(t) &= -\tfrac{2}{m}\lambda(t)\xi(t) \\ v'(t) &= -\tfrac{2}{m}\lambda(t)\eta(t) - g \\ 0 &= \xi(t)^2 + \eta(t)^2 - l^2 \end{aligned} \tag{1.36}$$

with the unknowns ξ, η, u, v, λ. The components u, v are the components of the velocity of the mass, i.e., $x' = (u, v)^\top$. The last equation of the

system (1.36) represents the constraint that the mass moves on a circle with radius l only. The unknown λ characterises the magnitude of the force, which keeps the mass on this circle.

The most appropriate model of the mathematical pendulum results by considering the angle φ. It holds $\sin\varphi = \xi/l$ and $\cos\varphi = \eta/l$. Consequently, we achieve an ordinary differential equation of second order

$$\varphi''(t) = -\frac{g}{l}\sin(\varphi(t)), \qquad \varphi(t_0) = \varphi_0, \quad \varphi'(t_0) = \varphi_0'.$$

Hence the problem can be modelled by an explicit system of two ODEs of first order. In contrast, the system (1.36) represents a system of five DAEs. However, computer aided design is able to construct mathematical models based on DAEs automatically. A model for large technical problems involving just ODEs cannot be found by the usage of existing software codes.

Differentiating the algebraic constraint of the system (1.36) with respect to time yields the relation

$$2\xi(t)\xi'(t) + 2\eta(t)\eta'(t) = 0 \qquad \Leftrightarrow \qquad \xi(t)u(t) + \eta(t)v(t) = 0. \tag{1.37}$$

Thus we obtain an additional algebraic relation, which the exact solution of (1.36) satisfies. The equation (1.37) represents a hidden constraint, since it is not included directly in the system (1.36). A further differentiation in time shows the relation

$$u(t)^2 + \xi(t)u'(t) + v(t)^2 + \eta(t)v'(t) = 0. \tag{1.38}$$

Multiplying the third and fourth equation of (1.36) by ξ and η, respectively, it follows

$$\begin{aligned} \xi(t)u'(t) &= -\tfrac{2}{m}\lambda(t)\xi(t)^2 \\ \eta(t)v'(t) &= -\tfrac{2}{m}\lambda(t)\eta(t)^2 - g\eta(t). \end{aligned}$$

Summing up these two equations and using (1.38) implies an algebraic relation for the unknown λ

$$\lambda(t) = \frac{m}{2l^2}\left(u(t)^2 + v(t)^2 - g\eta(t)\right). \tag{1.39}$$

Differentiating this equation with respect to time results to

$$\lambda'(t) = \frac{m}{2l^2}\left(2u(t)u'(t) + 2v(t)v'(t) - gv(t)\right). \tag{1.40}$$

Inserting the ODEs (1.36) and using (1.37) yields

$$\lambda'(t) = -\frac{3mg}{2l^2}v(t). \tag{1.41}$$

If we replace the algebraic constraint in (1.36) by the equation (1.41), then we achieve a system of five ODEs for the five unknowns. Three differentiations of the original system (1.36) with respect to time are necessary to derive this ODE system. Thus the differential index of the DAE system (1.36) is $k = 3$. It can be shown that the perturbation index is also $k = 3$.

Now we perform a numerical simulation of the mathematical pendulum using the DAE model (1.36) as well as the regularised model with (1.41), which represents an ODE model. We apply the parameters $m = 1$, $l = 2$, $g = 9.81$. The initial values are

$$\xi(0) = \sqrt{2}, \quad \eta(0) = -\sqrt{2}, \quad u(0) = 0, \quad v(0) = 0$$

The initial value $\lambda(0)$ follows from (1.39). The numerical solutions are computed in the time interval $t \in [0, 20]$.

The BDF methods are damping the amplitude of oscillations in a numerical simulation. In contrast, trapezoidal rule conserves the energy of a system and thus the amplitude of oscillations is reproduced correctly. We solve the ODE model by trapezoidal rule with adaptive step size control. Thereby, two different demands of relative accuracy are applied, namely 10^{-3} and 10^{-6}, whereas the absolute accuracy is set to 10^{-6}. The number of necessary integration steps is 610 and 4778, respectively. Fig. 3 illustrates the solution of the coordinates ξ, η by phase diagrammes. We recognise that the solution leaves the circle significantly in case of the lower accuracy.

To analyse this effect more detailed, we compute the values of the circle condition (last equation of (1.36)) and of the hidden constraint (1.37). For

the exact solution, these values are equal to zero, since the constraints are satisfied. On the contrary, the numerical solution causes an error in these constraints. Fig. 4 shows the corresponding discrepancies. We see that the error increases in time for each accuracy demand. Thus the numerical solution will leave a given neighbourhood of the circle at a later time. The reason is that the simulated ODE system does not include the information of the circle explicitly. This phenomenon is called drift off: the numerical solution of the regularised DAE, i.e., the ODE, drifts away from the manifold, where the true solution is situated.

Alternatively, we simulate the DAE model (1.36) directly using the trapezoidal rule with constant step sizes. We apply 1000 integration steps in the interval $t \in [0, 20]$. In each integration step, we perform just one step of the Newton method to solve the involved nonlinear system of algebraic equations.

The resulting solutions for ξ, η as well as the corresponding errors in the constraints are illustrated in Fig. 5. Both the circle condition and the hidden constraint exhibit an oscillating error, whose amplitude remains constant in time. Since the system (1.36) includes the circle condition directly, the error in this constraint depends just on the accuracy demand in solving the nonlinear system in each integration step. Hence the DAE model generates a significantly better numerical approximation than the corresponding ODE formulation using (1.41).

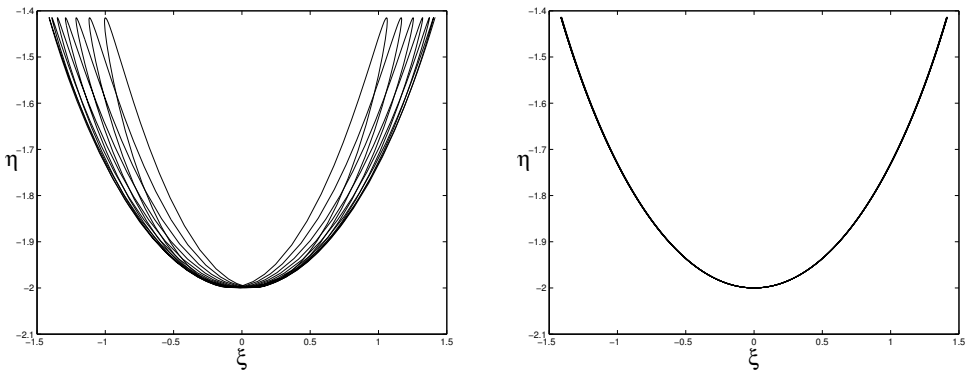


Figure 3: Phase diagramme of solution of ODE model for mathematical pendulum computed by trapezoidal rule with relative tolerance 10^{-3} (left) and 10^{-6} (right).

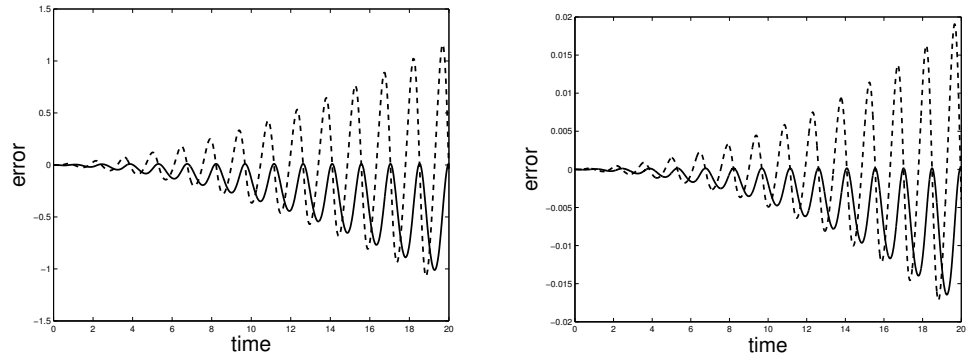


Figure 4: Error in circle condition (solid line) and hidden constraint (dashed line) for solution of ODEs corresponding to relative tolerance 10^{-3} (left) and 10^{-6} (right).

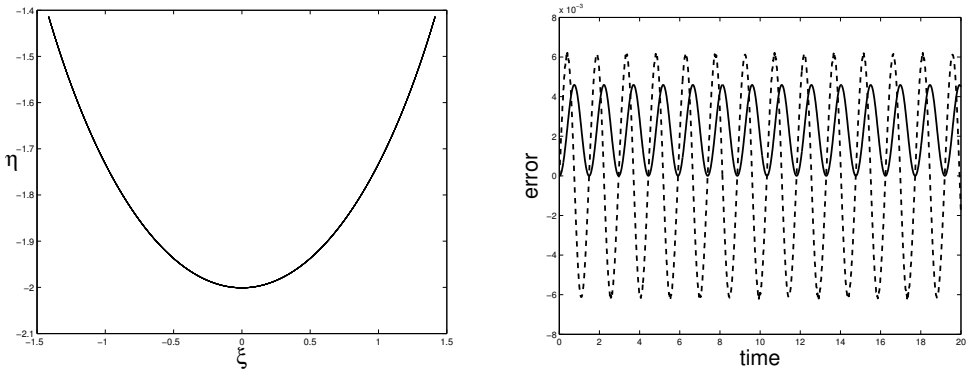


Figure 5: Phase diagramme of solution of DAE model for mathematical pendulum computed by trapezoidal rule (left) and corresponding errors (right) in circle condition (solid line) as well as hidden constraint (dashed line).

Chapter 2

Geometric integration

In many technical applications, solutions to differential equations fulfill additional properties, so-called invariants of the system, e.g., conservation of mass and momentum. One aims at transfering such properties to the numerical approximations for these systems. For this, let us consider the autonomeous differential equation

$$y' = f(y). \tag{2.1}$$

Definition 2.1 (First integral)
A non-constant function $I(y)$ is called a first integral or invariant of (2.1), if

$$\frac{d}{dt} I(y) = I'(y) f(y) = 0$$

for all y.

The Robertson example

$$\begin{aligned} y_1' &= -0.04 y_1 + 10^4 y_2 y_3, \\ y_2' &= 0.04 y_1 - 10^4 y_2 y_3 - 3 \cdot 10^7 y_2^2 \\ y_3' &= 3 \cdot 10^7 y_2^2 \end{aligned}$$

has the (linear) invariant $I(y) = d^\top y = y_1 + y_2 + y_3$ with $d = (1, 1, 1)^\top$ (why?).

Are there numerical schemes which preserve such invariants numerically? In this chapter we restrict ourselves to RK schemes.

Theorem 2.1 (Linear invariants)
All RK schemes preserve linear invariants.

Proof: Linear invariants are given by $I(y) = d^\top y$=const, i.e., we have

$$\frac{d}{dt} I(y) = d^\top y' = d^\top f(y) = 0.$$

As RK schemes are given by

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i k_i, \quad k_i = f(x_0 + hc_i, y_0 + h \sum_{j=1}^{s} a_{ij} k_j),$$

we get

$$I(y_1) = d^\top y_1 = d^\top y_0 + h d^\top \left(\sum_{i=1}^{s} b_i k_i \right) = d^\top y_0 = I(y_0),$$

and all RK schemes preserve linear invariants by construction. □

The mass conservation of the Robertson example defines a linear invariant with $d = (1, 1, 1)^\top$ and will be preserved by all RK schemes!

Theorem 2.2 (Quadratic invariants)
If the RK coefficients fulfill the conditions

$$M := (b_i a_{ij} + b_j a_{ji} - b_i b_j)_{i,j=1:s} = 0, \qquad i, j = 1, \ldots, s, \tag{2.2}$$

the RK scheme preserves quadratic invariants.

Proof: Quadratic invariants can be written as $I(y) = y^\top C y$ with a symmetric matrix C. By definition of a RK scheme we have

$$y_1^\top C y_1 = y_0^\top C y_0 + h \sum_{i=1}^{s} b_i k_i^\top C y_0 + h \sum_{j=1}^{s} b_j y_0^\top C k_j + h^2 \sum_{i,j=1}^{s} b_i b_j k_i^\top C k_j.$$

Using $k_i = f(Y_i)$ with internal stages $Y_i = y_0 + h \sum_{j=1}^{s} a_{ij} k_j$, we can solve the latter for y_0. Together with the symmetry of C we get

$$y_1^\top C y_1 = y_0^\top C y_0 + 2h \sum_{i=1}^{s} b_i f(Y_i)^\top C Y_i + h^2 \sum_{i,j=1}^{s} (b_i b_j - b_i a_{ij} - b_j a_{ji}) k_i^\top C k_j.$$

Because of $d/dt I(y) = 2y^\top C y' = 2y^\top C f(y) = 0$ for all y, (2.2) implies the theorem. □

As symplecticity defined in Chapter 2.2 is a quadratic invariant, schemes fulfilling condition (2.2) are called symplectic schemes.

Example 2.1 *The implicit mid-point rule defined by the Butcher tableau*

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}$$

is symplectic, but not the trapezoidal rule.

Explicit RK schemes cannot be symplectic.

Proof: *For the diagonal elements condition (2.2) reads*

$$b_i a_{ii} + b_i a_{ii} - b_i b_i = 0 \qquad i = 1, \ldots, s.$$

For explicit RK methods we have $a_{ii} = 0$ for all $i = 1, \ldots, s$, which yields the condition $b_i = 0$ for all $i = 1, \ldots, s$. This is a contradiction to the consistency condition $\sum b_i = 1$!

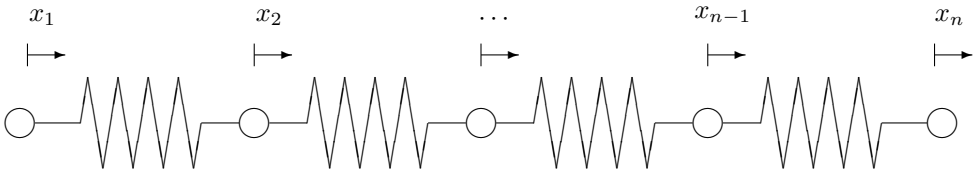
It can be shown that other invariants cannot be preserved by RK methods!

2.1 Isospectral flows

Isosepctrical flows, modelled by matrix differential equations that can be used to compute the eigen values of a given matrix, are another (more fancy) occurrence of quadratic invariants.

In the following we consider the case of a symmetric, tridiagonal matrix A.

Motivation: Toda flows. Let us consider a free mechanical system, a spring consisting of lumped mass points and nonlinear springs, which is not fixed at the boundaries:



If exponentially decaying forces between adjacent mass points are assumed, one speaks about toda lattices. Such systems serve as models for investigating nonlinear phenomena, for example, oscillations in crystals or heat conduction stimulated by external sources. If all masses are set to a unit mass ($m = 1$), one gets, by using the kinetic and potential energy

$$T = \frac{1}{2}\sum_{k=1}^{n} \dot{x}_k^2 \quad , \quad U = \sum_{k=0}^{n+1} \exp(x_k - x_{k+1})$$

and the formal boundary conditions $x_0 = -\infty$, $x_{n+1} = \infty$, the equations of motion (using the Lagrangian approach of chapter 1)

$$\ddot{x}_k = \exp(x_{k-1} - x_k) - \exp(x_k - x_{k+1}), \qquad (k = 1, \dots, n)\,. \tag{2.3}$$

The nonlinear transformation

$$\begin{aligned} a_k &= -\frac{1}{2}\dot{x}_k \qquad (k = 1, \dots, n) \\ b_k &= \frac{1}{2}\exp((x_k - x_{k+1})/2) \qquad (k = 2, \dots, n-1) \\ b_0 &= b_n = 0 \end{aligned}$$

due to Flaschka leads to the nonlinear differential equation

$$\begin{aligned} \dot{a}_k &= 2(b_k^2 - b_{k-1}^2) \qquad (k = 1, \dots, n) \\ \dot{b}_k &= b_k(a_{k+1} - a_k) \qquad (k = 1, \dots, n-1) \end{aligned}$$

of first order. By using the tridiagonal matrices

$$A(t) := \begin{pmatrix} a_1(t) & b_1(t) & & & \\ b_1(t) & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1}(t) \\ & & & b_{n-1}(t) & a_n(t) \end{pmatrix},$$

$$B(t) := \begin{pmatrix} 0 & -b_1(t) & & & \\ b_1(t) & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -b_{n-1}(t) \\ & & & b_{n-1}(t) & 0 \end{pmatrix},$$

one gets (with the commutator $[A, B] := AB - BA$) the equivalent system

$$\frac{d}{dt}A(t) = [A(t), B(t)] \tag{2.4}$$

of matrix differential equations, the Toda differential equations.

Is there really a connection between this system and the computation of eigen values, as asserted above? An answer to this question is given by the following

Theorem 2.3 (Toda differential equation and eigen values) *The solution to the Toda initial value problem*

$$\dot{A}(t) = [A(t), B(t)], \qquad A(0) = A_0 \tag{2.5}$$

is isospectral: $A(t) = Q(t)^\top A_0 Q(t)$, with the orthogonal matrix $Q(t)$ given as solution of the initial value problem (so-called orthogonal flow)

$$\dot{Q}(t) = Q(t)B(t)\,, \qquad Q(0) = I.$$

Asymptotically, $A(t)$ converges to a diagonal matrix with entries $\lambda_1, \ldots, \lambda_n$, the eigen values of the initial value A_0 in sorted order:

$$\lim_{t\to\pm\infty} A(t) = diag(\lambda_1, \ldots, \lambda_n)\,.$$

Proof: The initial value in transposed formulation

$$\frac{d}{dt}Q^\top(t) = -B(t)Q^\top(t)$$

implies $d(Q(t)Q^\top(t))/dt = 0$, and thus orthogonality of $Q(t)$. Thus the similarity of $A(t)$ and A_0 follows from $d(Q(t)A(t)Q^\top(t))/dt = 0$. This finishes the first part of the theorem.

Due to the isospectral flow we have $\|A(t)\|_2 = \|A_0\|_2$, and thus $|a_k(t)|, |b_k(t)|$ is uniformly bounded in t. With Lipschitz continuity of the b_k and the differential equation $\dot{a}_k = 2(b_k^2 - b_{k-1}^2)$, the latter result leads to $\lim_{t\to\pm\infty} b_k = 0$ by integration. □

Besides that, there is a connection between Toda differential equations and the QR method without shift: The k-th step of the QR method without shift is equivalent to evaluating the solution to the Toda differential equation at time point $t = k$. Or more formally, we have

Theorem 2.4 (Toda differential equation and QR method) *Let* $k \in \mathbb{N}$. *The matrix* $\exp(A(k))$ *coincides with the* k*-th iterate of the QR method without shift, applied to* $\exp(A_0)$.

The proof of this theorem can be found in

T.Nanda: Differential Equations and the QR-Algorithm. SIAM J. Numer. Anal. 22 (1985), 310–321.

In general, the choice of the skew-symmetric matrix B defines the type of the isospectral flow, in our case the continuous generalization of the discrete QR algorithm. By choosing other matrices B, one gets analogous statements for other discrete algorithms, for example, LU, Cholesky etc.

Is there any advantage in detecting such a connection between the QR algorithm without shift and the Toda differential equation?

Realizing the QR scheme by using the Toda differential equation allows for large savings in computation time! Two approaches are feasible:

Software based implementation: Instead of exploiting the isospectral structure of the flow (2.5), the numerical computation of the eigen values of A_0 is based on solving the initial value problem

$$\dot{Q}(t) = Q(t)B(t)\,, \qquad Q(0) = I$$

defing an orthogonal flow numerically by using geometric integration:

1. Assume that A_n is an approximation to the exact flow $A(t_n)$ at time $t = t_n$ and that it is isospectrally similar to L_0;

2. Let Q_{n+1} be the numerical approximation at time $t = t_{n+1}$ to the solution of the orthogonal flow

$$Q' = QB, \quad Q(t_n) = I, \quad t_n \leq t \leq t_{n+1}.$$

3. Set

$$A_{n+1} = Q_{n+1}^\top A_n Q_{n+1}$$

as the numerical approximation $A_{n+1} \approx A(t_{n+1})$.

Provided that Q_{n+1} is an orthogonal matrix, the matrix A_{n+1} above is symmetric and isospectrally similar to A_n, and by induction to A_0. The main demand on the numerical approximation Q_{n+1} of the orthogonal flow $Q(t_{n+1})$ is to preserve the orthogonal structure of the flow, i.e., we require $Q_{n+1}^\top Q_{n+1} = Q_{n+1} Q_{n+1}^\top = I$. As the latter defines a quadratic invariant, we can use any RK scheme with coefficients fulfilling $M = 0$! As we are only interested on the asymptotic behaviour os the system, we can use arbitrarily large step sizes.

Here we have to deal with matrix differential equations. Of course, we can implement RK schemes for matrix differential equations just by embedding $\mathbb{R}^{n\times n}$ into $\mathbb{R}^{n^2}$, i.e, we write a matrix $C \in \mathbb{R}^{n\times n}$ as a long vector $(a_1, a_2, \ldots, a_n)^\top$ with the columns a_i of A. This allows for defining easily RK schemes for matrix differential equations, but invariants of type $Q^\top Q = I$ cannot easily be regarded as quadratic invariants in $\mathbb{R}^{n^2}$.

An alternative is to derive RK schemes for matrix differential equations $A' = f(x, A)$, $A(x_0) = A_0$ directly as

$$\begin{aligned} A_1 &= A_0 + h \sum_{i=1}^{s} b_i k_i, \\ k_i &= f(x_0 + c_i h, A_0 + h \sum_{j=1}^{s} a_{ij} k_j). \end{aligned}$$

For this approach, the orthoganlity relations $Q^\top Q = QQ^\top = I$ define quadratic invariants, and preservation of the orthogonal flow by algebraically stable RK scheme (i.e. RK schemes with $M = 0$) can be shown directly (see exercise).

Hardware based implementation: In general, any initial value problem can be realized as electric circuit using basic elements such as multiplicators, adders and operational amplifiers. In the case of tridiagonal matrices,

only local connections between nodes are required, whose node potentials correspond to the values a_k and b_k. A downwelling composition using the same cells allows for solving eigen value problems of any dimension. This approach is characterized by a simple implementation of nonlinearities, low energy consumption and an inherent parallelism: a high processing speed, which does not depend on the dimension n of the system!

An efficient application of this approach is the task of filtering, for example, the use of a median filter with 256 shades of grey. For this, the values $a_1, \ldots, a_n$ that are to be filtered ere embedded into a diagonal matrix $A_0 := \mathrm{diag}(a_1, \ldots\ a_n)$. A slight perturbation converts this matrix into a tridiagonal matrix, by allocating the lower and upper diagonal with $\delta \ll 1$; applying the Toda flow to this slightly perturbed symmetric tridiagonal matrix supplies us with the eigen values $a_1, \ldots, a_n$ of A_0 in sorted order and very high accuracy (Gerschgorin!). It only remains to grip the wanted value in the circuit (as node potential).

Chips equipped with this functionality had been derived at Munich University of Technology (Chair Nossek) and can be purchased. Speed advantage up to 10000–100000!

Both approaches lead to current fields of research in numerical analysis: On one hand, the paradigm shift from quantitative to qualitative integration schemes (geometric integration as strucure-preserving schemes), on the other hand the attempt to arrive at real-time simulations by deriving a hardware based numerical analysis. Neural nets are part of the latter approach, which are strongly linked to nonlinear least square problems.

Reference: Mari Paz Calvo, Arieh Iserles and Antonelle Zenna: Numerical solution of isospectral flows. Mathematics of Computation, Volume 66, Number 220, 1461–1485 (1997).

2.2 Hamiltonian dynamics

A class of models, which contain invariants already at the modelling level, are Hamiltonian systems

$$\begin{aligned} \dot{p} &= -\frac{\partial}{\partial q} H(p,q), \\ \dot{q} &= \frac{\partial}{\partial p} H(p,q), \end{aligned} \tag{2.6}$$

with the Hamiltonian $H : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ twice continuously differentiable. In short hand:

$$\dot{y} = J^{-1} \nabla H(y)$$

with

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

and $y := (p, q)$.

Example 2.2 *The mathematical pendulum with mass 1, length 1 and gravitational constant 1 has the energy*

$$H(p,q) = \frac{1}{2} p^2 - \cos q$$

with displacement $q = \alpha$ and momentum $p = \dot{\alpha}$. The equations of motion (Hamiltonian formulation) read

$$p' = -\sin q, \quad q' = p.$$

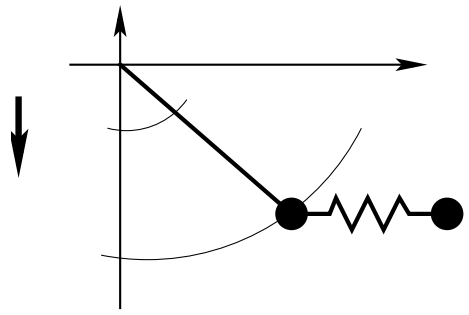


Figure 6: Mathematical pendulum

Hamiltonian systems are characterized by the following properties, which we will discuss in the following:

- Energy conservation
- Volume preservation
- Symplecticity
- Time reversibility

Energy conservation

As can be seen from

$$\frac{d}{dt}H(p,q) = H_p p' + H_q q' = -H_p H_q + H_q H_p = 0,$$

the Hamiltonian itself is an invariant of the system. In many cases, it models the energy conserved in a system.

Volume preservation

Besides energy conservation, Hamiltonian systems posses an additional invariant. For this, let us consider the flow $\varphi_t(y_0)$ defined by

$$\varphi_t(y_0) := y(t; t_0, y_0),$$

which describes the solution with respect to the initial value $y(t6_0) = y_0$. This definition is now generalized to a set of initial values. We define

$$\varphi_t\mathcal{Y} := \{y | y = y(t; t_0, y_0), y_0 \in \mathcal{Y}\}.$$

Definition 2.2 (Volume-preserving flows)
The flow φ_t and the differential equation (2.1), resp., are called volume-preserving, if

$$Vol(\varphi_t\mathcal{Y}) = Vol(\mathcal{Y})$$

holds for all $t > t_0$, with $Vol(\mathcal{Y})$ denoting the volume and surface, resp., of $\mathcal{Y}$.

The volume is computed by

$$\mathrm{Vol}(\varphi_t\mathcal{Y}) = \int_{\varphi_t\mathcal{Y}} dy = \int_{\mathcal{Y}} \left|\det\left(\frac{\partial y}{\partial y_0}(t;t_0,y_0)\right)\right| dy_0,$$

which is equivalent to

$$\mathrm{Vol}(\varphi_t\mathcal{Y}) = \int_{\mathcal{Y}} \exp\left(\int_{t_0}^{t} \mathrm{trace}\left(f_y(y(s;t_0,y_0))\right) ds\right) dy_0 \tag{2.7}$$

with the Jacobian f_y (see Hairer/Norsett/Wanner p. 99). Equation (2.7) shows that $\mathrm{trace}(f_y(y)) = 0$ implies volume preservation for the flow φ_t due to

$$\mathrm{Vol}(\varphi_t\mathcal{Y}) = \int_{\mathcal{Y}} \exp(0) dy_0 = \mathrm{Vol}\mathcal{Y}.$$

Hamiltonian systems are characterized by the Jacobian

$$f_y = \begin{pmatrix} -H_{pq} & -H_{qq} \\ H_{pp} & H_{pq} \end{pmatrix}$$

with $\mathrm{trace}(f_y) = 0$, and thus are volume-preserving by construction.

Symplectic structure

Hamiltonian systems fulfill a more general property, called symplecticity, which yields volume preservation as a natural consequence. Symplecticity is defined by fulfilling a quadratic invariant, i.e., can be written as $I(y) = y^\top Cy$ =const. with a given matrix C, as we will see in the following.

Theorem 2.5 (Symplectic structure of Hamiltonian flow) *Let $H(p,q)$ be a twice continuously differentiable function on $U \in \mathbb{R}^{2d}$. Then, for each fixed t, the flow φ_t is a symplectic transformation*

$$\left(\frac{\partial \varphi_t}{\partial y_0}\right)^\top J \left(\frac{\partial \varphi_t}{\partial y_0}\right) = J \tag{2.8}$$

wherever it is defined.

Proof: The derivative $\partial\varphi_t/\partial y_0$ (with $y_0 = (p_0, q_0)$) is a solution of the variational equation which, for the Hamiltonian system (2.6), is of the form $\dot{\Psi} = J^{-1}\nabla^2 H(\varphi_t(y_0)\Psi$, where $\nabla H(p,q)$ is the Hessian matrix of $H(p,q)$ ($\nabla^2 H(p,q)$ is symmetric). We therefore obtain

$$\begin{aligned}
\frac{d}{dt}\left(\left(\frac{\partial\varphi_t}{\partial y_0}\right)^\top J\left(\frac{\partial\varphi_t}{\partial y_0}\right)\right) &= \left(\frac{d}{dt}\frac{\partial\varphi_t}{\partial y_0}\right)^\top J\left(\frac{\partial\varphi_t}{\partial y_0}\right) + \left(\frac{\partial\varphi_t}{\partial y_0}\right)^\top J\left(\frac{d}{dt}\frac{\partial\varphi_t}{\partial y_0}\right) \\
&= \left(\frac{\partial\varphi_t}{\partial y_0}\right)^\top \nabla^2 H(\varphi_t(y_0)) J^{-\top} J\left(\frac{\partial\varphi_t}{\partial y_0}\right) + \\
&\quad + \left(\frac{\partial\varphi_t}{\partial y_0}\right)^\top \nabla^2 H(\varphi_t(y_0))\left(\frac{\partial\varphi_t}{\partial y_0}\right) \\
&= 0,
\end{aligned}$$

because $J^\top = -J$ and $J^{-\top}J = -I$. Since the relation

$$\left(\frac{\partial\varphi_t}{\partial y_0}\right)^\top J\left(\frac{\partial\varphi_t}{\partial y_0}\right) = J$$

is satisfied for $t = 0$ (φ_0 is the identity map), it is satisfied for all t and all (p_0, q_0), as long as the solution remains in the domain of definition of H. □

If we consider the Hamiltonian system augmented by the sensitivity equation

$$\begin{aligned}
\dot{y} &= J^{-1}\nabla H(y), \\
\dot{\Psi} &= J^{-1}\nabla^2 H(y)\Psi,
\end{aligned}$$

then we get that the quadratic function $\Psi^\top J\Psi$ is constant, i.e., symplecticity is a quadratic invariant.

Note that volume preservation is a direct consequence of symplecticity:

$$\det(\Psi^\top J\Psi) = \det J = 1 \Rightarrow |\det\Psi| = 1.$$

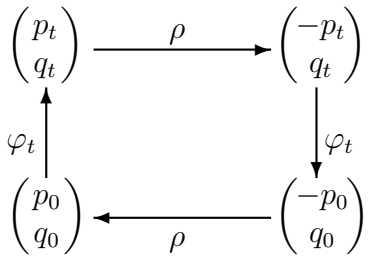


Figure 7: Time-reversibility of the Hamiltonian flow φ_t: applying the transformations $\varphi_t, \rho, \varphi_t$ and finally ρ to the initial value (p_0, q_0) yields the initial value again.

Time reversibility

The Hamiltonian flow is time reversible, i.e., it holds

$$\rho \cdot \varphi_t(\rho \cdot \varphi_t(y_0)) = y_0 \qquad \text{for } \rho = \begin{pmatrix} I_n & 0 \\ 0 & -I_n \end{pmatrix} \tag{2.9}$$

see Fig. 7. This is equivalent to

$$\rho\varphi_t = \varphi_{-t}\rho \quad \Leftrightarrow \quad \begin{pmatrix} -p(t; t_0, p_0, q_0) \\ q(t; t_0, p_0, q_0) \end{pmatrix} = \begin{pmatrix} p(-t; t_0, -p_0, q_0) \\ q(-t; t_0, -p_0, q_0) \end{pmatrix},$$

as $\varphi_{-t} = (\varphi_t)^{-1}$.

Why do we have time reversibility of the Hamiltonian flow? This follows directly from a symmetry property of the Hamiltonian $H(p, q) = H(-p, q)$. Namely, if the Hamiltonian satisfies this condition, then the equations of motion (2.6) are invariant under the transformation

$$R(p, q, t) := (-p, q, t).$$

In turn, this implies that when $(p(t), q(t))$ is a trajectory in phase space describing a possible motion of the system with initial momentum and position (p_0, q_0), then so is $-p(-t), q(-t))$ with initial condition $(-p_0, q_0)$. In configuration (position) space this means that if we have a trajectory $q(t)$, then we also have a trajectory $q(-t)$. This is precisely what we see when we play a film of a time-reversible system in reverse.

Geometric integration schemes for Hamiltonian systems

Let us now consider numerical approximation schemes given by the numerical flow $y_1 := \Phi_h(y_0)$, integrating the flow numerically from $t = 0$ to $t = h$ from the initial-value y_0 to the new approximate $y_1 \approx y(h)$ at $t = h$. We demand now the following:

- volume preservation of the numerical flow:

$$\left| \det \frac{\partial \Phi_h(y_0)}{\partial y_0} \right| = 1.$$

- symplectic numerical flow:

$$\left(\frac{\partial \Phi_h}{\partial y_0} \right)^\top J \left(\frac{\partial \Phi_h}{\partial y_0} \right) = J$$

- time reversible numerical flow:

$$\rho \cdot \Phi_h(\rho \cdot \Phi_h(y_0)) = y_0$$

  This is equivalent to $P\varphi_t^h = \varphi_t^{-h} P$ for symmetric schemes $\Phi_h \Phi_{-h} = I$.

Due to the symmetry of the scheme, the most simple symplectic and symmetric numerical method has at least order two, i.e., the difference between exact solution and numerical approximation at time point T after n steps of step size h ($T = nh$) is of order $\mathcal{O}(h^2)$ for h being sufficiently small. It is given by the Störmer-Verlet method (or Leap-Frog scheme), which can be written as an explicit scheme for separable Hamiltonians $H(q, p) = V(p) + U(q)$. One step, starting from initial values (p_0, q_0), to obtain numerical approximations (p_1, q_1) at time point $t_0 + h$ reads

$$\begin{aligned} p_{1/2} &= p_0 - \frac{h}{2} U_q(q_0)\,, && (2.10) \\ q_1 &= q_0 + h V_p(p_{1/2})\,, && (2.11) \\ p_1 &= p_{1/2} - \frac{h}{2} U_q(q_1)\,, && (2.12) \end{aligned}$$

with short-hands U_q and V_p for $\partial U/\partial q$ and $\partial V/\partial p$, respectively. Symplecticity of the scheme follows directly from the fact that it is defined by the composition of three symplectic mappings

$$\begin{aligned}(q_0, p_0) &\rightarrow p_{h/2}(q_0, p_0) = (q_0, p_{1/2})\,, &(2.13)\\ (q_0, p_{1/2}) &\rightarrow q_h(q_0, p_{1/2}) = (q_1, p_{1/2})\,, &(2.14)\\ (q_1, p_{1/2}) &\rightarrow p_{h/2}(q_1, p_{1/2}) = (q_1, p_1)\,, &(2.15)\end{aligned}$$

so-called p- and q-updates with step sizes $h/2$, h and $h/2$, resp., which enables us to rewrite the leap-frog scheme as

$$p_{h/2} \circ q_h \circ p_{h/2}(q_0, p_0)\,. \tag{2.16}$$

Symmetry follows directly by changing the sign of h and replacing (q_0, p_0) by (q_1, p_1). Time reversibility is then equivalent to $\rho\Phi_h = \Phi_{-h}\rho$, which reads

$$\begin{aligned} q_0 + hV_p(p_{1/2}) &= q_0 - hV_p(-p_{1/2}),\\ -p_{1/2} + \frac{h}{2}U_q(q_0 + hV_p(p_{1/2})) &= -p_{1/2} + \frac{h}{2}U_q(q_0 - hV_p(-p_{1/2})).\end{aligned}$$

This holds for symmetrical Jacobians $V_p = \frac{1}{2}p^\top M^{-1}p$ fulfilling $V_p(p) = -V_p(-p)$.

An easy way to derive symplectic and time-reversible higher-order schemes $\widetilde{\Phi}_h$ is based on the composition of m symplectic and time-reversible basic schemes Φ_h:

$$\widetilde{\Phi}_h = \Phi_{\gamma_1 h} \circ \Phi_{\gamma_2 h} \circ \ldots \circ \Phi_{\gamma_m h}\,. \tag{2.17}$$

Besides the time-reversibility of the underlying basic schemes Φ_h, the coefficients have to be symmetric, too, to get an overall time-reversible system:

$$\gamma_{m-k} = \gamma_k \quad \text{for} \quad k = 1, 2, \ldots, m-1. \tag{2.18}$$

It can easily be shown that the composition scheme has order $p+1$ (with the underlying scheme having order p), if the following two conditions hold for the free parameters $\gamma_1, \ldots, \gamma_m$:

$$\sum_{j=1}^{m} \gamma_j = 1, \qquad \sum_{j=1}^{m} \gamma_j^p = 0. \tag{2.19}$$

Symplecticity and time-revesibility of the composition scheme follow directly from symplecticity and time-revesibility of the underlying scheme.

This approach allows us to construct symplectic and time-reversible schemes of any arbitrary (even) high order. We start with the Störmer-Verlet scheme Φ_h and define

$$\widetilde{\Phi}_h = \Phi_{\gamma_1 h} \circ \Phi_{\gamma_2 h} \circ \Phi_{\gamma_3 h}$$

with

$$\gamma_1 = \gamma_3 = \frac{1}{2 - 2^{\frac{1}{3}}}, \gamma_2 = 1 - 2\gamma_1.$$

These coefficients fulfill both conditions above, which gives at least order 3 for the composition scheme $\widetilde{\Phi}_h$. As the order of symmetric methods is even, we get order 4 for $\widetilde{\Phi}_h$. We can now repeat this process, only replacing Φ_h by $\widetilde{\Phi}_h$, and we get schemes of order 6,8, etc.

2.3 Differential equations on Lie groups

In this chapter we will discuss differential equations on manifolds given by Lie groups. Before defining such differential equations, we have to remember some concepts from algebra.

Definition 2.3 (Lie group) *Let G be a differentiable manifold and $G \times G \to G$ a differentialble mapping which turns G into a group. This group is then called a Lie group.*

Definition 2.4 (lie algebra) *The Lie algebra $\mathfrak{g}$ of a Lie group G is the tangent space at the identity, i.e, it holds $\mathfrak{g} = T_I G$. The mapping $\mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, which turns $\mathfrak{g}$ into an algebra, is defined be the commutator (or Lie bracket)*

$$[A, B] = AB - BA.$$

The mapping induced by the Lie bracket is bilnear, skew-symmetric and fulfills the Jacobi identity

$$[A, [B, C]] + [C, [A, B]] + [B, [C, A]] = 0.$$

We get now

Lemma 2.6 *Let U be an element of the Lie group G and A an element of its Lie algebra $\mathfrak{g} = T_I G$. Then AU is an element of the tangent space $T_U G$ and $\dot{U} = AU$ is a differential equation on the manifold G.*

Proof: For a given manifold $\mathcal{M}$, the tangent space $T_a\mathcal{M}$ at point a is defined by the set of all vectors v for which the differentiable path $\alpha : (-\epsilon, \epsilon) \to \mathcal{M}$ with $\alpha(0) = a$ and $\dot{\alpha}(0) = v$ exists. Hence we get for $A \in \mathfrak{g} = T_I G$ that there exists a differentiable path $\alpha(t) \in G$ such that $\alpha(0) = I$ and $\dot{\alpha}(0) = A$ holds. For $U \in G$ fixed, another path is defined by $\gamma(t) := \alpha(t)U \in G$ for which $\gamma(0) = U$ and $\dot{\gamma}(0) = AU$ holds. Thus AU is an element of the tangent space $T_U G$. The remaining fact that $\dot{U} = AU$ defines a differential equation on the Lie group G follows directly from

Theorem 2.7 (Differential equations on manifolds) *Let $\mathcal{M}$ be a manifold. Then $\dot{U} = AU$ is a differential equation on the manifold if and only if*

$$AU \in T_U\mathcal{M} \qquad \forall U \in \mathcal{M}.$$

Proof: That $AU \in T_U\mathcal{M}$ has to hold for all $U \in \mathcal{M}$ follows directly from the definition of the tangent space $T_U\mathcal{M}$, as $AU \in T_U\mathcal{M}$ for the exact solution $U \in \mathcal{M}$. Assume now that $AU \in T_U\mathcal{M} \qquad \forall U \in \mathcal{M}$ holds. Let $\mathcal{M}$ be locally defined by $\mathcal{M} = \{U = \Psi(\Omega)\}$, with Ψ being a differentiable parameterization and Ω a local coordinate of the manifold. Then we can write the solution of the initial-value problem

$$\dot{U} = AU, \qquad U(t_0) = U_0 = \Psi(\Omega_0)$$

as $U(t) = \Psi(\Omega(t))$. We get

$$\left(\frac{d}{d\Omega}\Psi(\Omega(t))\right)\dot{\Omega} = A\Psi(\Omega(t)).$$

If we define for $H \in \mathfrak{g}$ implicitly $d\Psi_\Omega$ via the equation

$$\left(\frac{d}{d\Omega}\Psi(\Omega(t))\right) H = (d\Psi_\Omega(H))\Psi(\Omega),$$

which leads to

$$d\Psi_\Omega(\dot{\Omega})\Psi(\Omega) = A\Psi(\Omega),$$

we can solve for $\dot{\Omega}$ by introducing the inverse $d\Psi_\Omega^{-1}$ (or pseudoinverse in the case of Ψ not being bijective) and get the differential equation

$$\dot{\Omega} = d\Psi_\Omega^{-1} A. \tag{2.20}$$

Defining now $\Omega(t)$ as solution of the initial-value problem given by the ODE (2.20) and initial value $\Omega(t_0) = \Omega_0$, we get $U(t) = \Psi(\Omega(t))$ as solution of the initial-value problem $\dot{U} = AU, U(t_0) = U_0$. Therefore the solution $U(t)$ remains in the manifold $\mathcal{M}$ and $\dot{U} = AU$ is a differential equation on this manifold. $\square$

This characterization allows for defining an algorithms to solve numerically the differential equation

$$\dot{U}(t) = A(t) \cdot U(t) \tag{2.21}$$

with $U(t)$ being an element of a Lie group G and $A(t)$ of the corresponding Lie algebra $\mathfrak{g}$: based on the proof of theorem 2.7, we can build a solution of $U(t) \in G$ by using a local parameterization ψ, if we first get the expression $\Omega(t)$ by solving the ODE (2.20) in the parameter space. Finally we get the desired solution $U(t)$ by backprojection of $\Omega(t)$ onto the Lie group.

If the parameter space is given by the Lie algebra $\mathfrak{g}$, then we seek a mapping Ψ with

$$\Psi : \mathfrak{g} \to G, \qquad U(t) = \Psi(\Omega(t)),$$

with $\Omega(t)$ given as solution of the initial-value problem $\dot{\Omega} = d\Psi_\Omega^{-1} A(t)$, $\Omega(t_0) = \Omega_0$. Setting $\widetilde{\Psi}(\Omega(t)) := \Psi(\Omega(t))U^{-1}(t_0)$, we can reformulate this task into: we seek a mapping $\widetilde{\Psi}$ with

$$\widetilde{\Psi} : \mathfrak{g} \to G, \qquad U(t) = \widetilde{\Psi}(\Omega(t)) \cdot U(t_0),$$

with $\Omega(t)$ given as solution of the initial-value problem $\dot{\Omega} = d\widetilde{\Psi}_{\Omega}^{-1} A(t)$, $\Omega(t_0) = \Omega_0$.

The algorithm now reads as follows.

Algorithmus 2.1 (Solving ODEs on Lie groups via the Lie algebra) *Let the differential equation $\dot{U} = A(t)U(t)$ with $A(t) \in \mathfrak{g}$ and $U(t)$ in the corresponing Lie group G be given. Let $\Psi : \mathfrak{g} \to G$ be a mapping from the Lie algebra $\mathfrak{g}$ to the Lie group G with $\Psi(0) = I$. A numerical approximation U_{n+1} at t_{n+1} can be computed as folows provided that an approximation U_n for $U(t_n)$ at time point t_n is given:*

1. *Define the auxiliary ODE for $\Omega(t)$ as*
$$\dot{\Omega} = d\Psi_{\Omega}^{-1} f(A(t)), \qquad \Omega(t_0) = 0.$$
2. *Compute $\Omega_{t_{n+1}} \approx \Omega(t_n + h)$ numerically with step size $h := t_{n+1} - t_n$.*
3. *Define the numerical approximation of the ODE $\dot{U} = A(t)U(t)$ at time point t_{n+1} by*
$$U_{n+1} = \Psi(\Omega_{n+1})U_n.$$

Why not directly solving the Lie group differential equation? The answer is quite simple: the Lie group is a multiplicative group. Applying numerical schemes such as Runge-Kutta schemes we have used for ODEs in $\mathbb{R}^n$ directly will not work: the Lie group is only closed with respect to multiplicative, but not with respect to additive operations. Hence the approximation will not lie in the Lie group manifold. However, the Lie algebra is closed with respect to additive operations, and hence the numerical approximation of step 2 will remain in the Lie group and thus U_{n+1} in the Lie group.

Another question arises? Which parameterization should one use? For matrix Lie groups, i.e., Lie groups with elements in $GL(n)$, the exponential mapping is a natural choice. An alternative for quadratic Lie groups, i.e., Lie groups of the form $G = \{U : U^H P U = P\}$ with a given constant matrix P is given by the Cayley transform. We will discuss both in the following.

Exponential map

Lemma 2.8 (Exponential map) *Consider a matrix Lie group G and its Lie algebra $\mathfrak{g}$. The matrix exponential is a map*

$$\exp : \mathfrak{g} \to G, \qquad \exp(\Omega) = \sum_{k \geq 0} \frac{1}{k!} \Omega^k$$

i.e., for $A \in \mathfrak{g}$ we have $\exp(A) \in G$. Moreover, $\exp$ is a local diffeomorphism in a neighborhood of $A = 0$.

Proof: For $A \in \mathfrak{g}$, if follows from the defintion of the tangent space $\mathfrak{g} = T_I G$ that there exists a differentiable path $\alpha(t)$ in G satisfying $\alpha(0) = I$ and $\dot{\alpha}(0) = A$. For a fixed $Y \in G$, the path $\gamma(t) := \alpha(t)Y$ is in G and satisfies $\gamma(0) = Y$ and $\dot{\gamma}(0) = AY$. Consequently, $AY \in T_Y G$ and $\dot{Y} = AY$ defines a differential equation on the manifold G. The solution $Y(t) = \exp(tA)$ is therefore in G for all t.

Since $\exp(H) - \exp(0) = H + \mathcal{O}(H^2)$, the derivative of the exponential map at $A = 0$ is the identity, and it follows from the inverse function theorem that exp is a local diffeomorphism close to $A = 0$. □

To apply our algorithm, we have to derive the derivative of the exponential map and its inverse. Elegant formulas for the derivative of exp and for its inverse can be obtained by the use of matrix commutators $[\Omega, A] = \Omega A - A\Omega$. If we suppose Ω fixed, this expression defines a linear operator $A \to [\Omega, A]$

$$\mathrm{ad}_\Omega(A) = [\Omega, A], \tag{2.22}$$

which is called the adjoint operator. Let us start by computing the derivatives of Ω^k . The product rule for differentiation becomes

$$\left(\frac{d}{d\Omega} \Omega^k \right) H = H\Omega^{k-1} + \Omega H \Omega^{k-2} + \ldots \Omega^{k-1} H, \tag{2.23}$$

and this equals $kH\Omega^{k-1}$ if Ω and H commute. Therefore, it is natural to write (2.23) as $kH\Omega^{k-1}$ to which are added correction terms involving

commutators and iterated commutators. In the cases $k = 2$ and $k = 3$we have

$$\begin{aligned} H\Omega + \Omega H &= 2H\Omega + \mathrm{ad}_\Omega(H), \\ H\Omega^2 + \Omega H\Omega + \Omega^2 H &= = 3H\Omega^2 + 3(\mathrm{ad}_\Omega(H))\Omega + \mathrm{ad}_\Omega^2(H), \end{aligned}$$

where ad_Ω^i denotes the iterated application of the linear operator ad_Ω . With the convention $\mathrm{ad}_\Omega^0(H) = H$ we obtain by induction on k that

$$\left(\frac{d}{d\Omega}\Omega^k\right) H = \sum_{i=0}^{k-1} \binom{k}{i+1} \left(\mathrm{ad}_\Omega^i(H)\right) \Omega^{k-i-1}. \tag{2.24}$$

This is seen by applying Leibniz rule to $\Omega^{k+1} = \Omega \cdot \Omega^k$ and by using the identity $\Omega(\mathrm{ad}_\Omega^i(H)) = (ad_\Omega^i(H))\Omega + \mathrm{ad}_\Omega^{i+1}(H)$.

Lemma 2.9 (Derivative of exponential map) *The derivative of* $\exp\Omega = \sum_{k\geq 0} \frac{1}{k!}\Omega^k$ *is given by*

$$\left(\frac{d}{d\Omega}\exp\Omega\right) H = (d\exp_\Omega(H))\exp\Omega, \tag{2.25}$$

where

$$d\exp_\Omega(H) = \sum_{k\geq 0} \frac{1}{(k+1)!} ad_\Omega^k(H).$$

The series (2.25) *converges for all matrices* Ω.

Proof: Multiplying (2.24) by $(k!)^{-1}$ and summing, then exchanging the sums and putting $j = k - i - 1$ yields

$$\begin{aligned} \left(\frac{d}{d\Omega}\exp\Omega\right) H &= \sum_{K\geq 0} \frac{1}{k!} \sum_{i=0}^{k-1} \binom{k}{i+1} \left(\mathrm{ad}_\Omega^i(H)\right) \Omega^{k-i-1} \\ &= \sum_{i\geq 0}\sum_{j\geq 0} \frac{1}{(i+1)!j!} \left(\mathrm{ad}_\Omega^i(H)\right) \Omega^j. \end{aligned}$$

The convergence of the series follows from the boundedness of the linear operator ad_Ω (we have $\|\mathrm{ad}_\Omega\| \leq 2\|\Omega\|$). $\square$

Lemma 2.10 *If the eigenvalues of the linear operator ad_Ω are different from $2l\pi i$ with $l \in \{\pm 1, \pm 2, \ldots\}$, then $d\exp_\Omega$ is invertible. Furthermore, we have for $\|\Omega\| \leq \pi$ that*

$$d\exp_\Omega^{-1}(H) = \sum_{k\geq 0} \frac{B_k}{k!} ad_\Omega^k(H), \tag{2.26}$$

where B_k are the Bernoulli numbers, defined by $\sum_{k\geq 0}(B_k/k!)x^k = x/(e^x - 1)$.

Proof: The eigenvalues of $d\exp_\Omega$ are $\mu = \sum_{k\geq 0} \lambda^k/(k+1)! = (e^\lambda - 1)/\lambda$, where λ is an eigenvalue of ad_Ω. By our assumption, the values μ are non-zero, so that $d\exp_\Omega$ is invertible. By definition of the Bernoulli numbers, the composition of (2.26) with (2.25) gives the identity. Convergence for $\|\Omega\| < \pi$ follows from $\|\mathrm{ad}_\Omega\| \leq 2\|\Omega\|$ and from the fact that the radius of convergence of the series for $x/(e^x - 1)$ is 2π. □

The following theorem now states that we can use the exponential mapping to solve the differential equation(2.21).

Theorem 2.11 (Magnus, 1954) *The solution of the linear matrix differential equation $\dot{U} = A(t)U(t)$ with $A(t) \in \mathfrak{g}$ and $U(t) \in G$ can be written as $U(t) = \exp(\Omega(t))U_0$ with $U_0 \in G$ with Ω defined by*

$$\dot{\Omega} = d\exp_\Omega^{-1}\big(A(t)\big). \tag{2.27}$$

As long as $\|\Omega\| < \pi$, the convergence of the $d\exp_\Omega^{-1}$ exapnsion (2.26) is assured.

Proof: Comparing the derivative of $\dot{U} = A(t)U(t)$,

$$\begin{aligned} \dot{U} &= \left(\frac{d}{d\Omega}\exp\Omega(t)\right)\dot{\Omega}(t)U_0 \overset{(2.25)}{=} \Big(d\exp_{\Omega(t)}\big(\dot{\Omega}(t)\big)\Big)\exp\Omega(t)U_0 \\ &= \Big(d\exp_{\Omega(t)}\big(\dot{\Omega}(t)\big)\Big)U(t), \end{aligned}$$

with(2.21) we obtain $A(t) = d\exp_{\Omega(t)}\big(\dot{\Omega}(t)\big)$. Applying the inverse operator $d\exp_\Omega^{-1}$ to this relation yields the differential equation (2.27) for $\Omega(t)$. The statement on the convergence is a consequence of lemma 2.10.

Thus, with $\Psi := \exp$, (2.20) yields the differential equation

$$\dot{\Omega} = d\exp_\Omega^{-1}(A(t)).$$

However, the infinite series

$$d\exp_\Omega^{-1}(A(t)) = \sum_{k\geq 0} \frac{B_k}{k!}\mathrm{ad}_\Omega^k(A(t)) = A(t) - \frac{1}{2}[\Omega, A(t)] + \frac{1}{12}\big[\Omega, [\Omega, A(t)]\big] + \dots \tag{2.28}$$

cannot be computed in finite time. We need a criterion when to truncate the summation for a given accuracy requirement in the Munthe-Kaas algorithm 2.1, i.e., we have to replace ∞ in the summation by an appropriate truncation index q:

$$\dot{\Omega} = A(t)U_0) + \sum_{k=0}^{q} \frac{B_k}{k!}\mathrm{ad}_\Omega^k(A(t)U_0)), \qquad \Omega_0 = 0. \tag{2.29}$$

How to choose q minimal is given by the following

Theorem 2.12 (Appropriate termination criterion) *If the Runge-Kutta method is of (classical) order p and if the truncation index in* (2.29) *satisifes $q \geq p-2$, then the method of algorithm 2.1(with $\Psi := \exp$) is of order p.*

Proof: For sufficiently smooth $A(t)$ we have $\Omega(t) = tA(t_0) + \mathcal{O}(t^2)$ and $[\Omega(t), A(t)] = \mathcal{O}(t^2)$. This implies that $\mathrm{ad}_{\Omega(t)}^k(A(t)) = \mathcal{O}(t^{k+1})$, so that the truncation of the series in (2.29) induces an error of size $\mathcal{O}(t^{q+2})$ for $|t| < h$. Hence, for $q+2 \geq p$, this truncation does not affect the order of convergence. □

If we use the Störmer-Verlet scheme, which is of order $p = 2$, the truncation index $q = 0$ is sufficient. With $B_0 = 1$ we get a numerical approximation of order two by applying Störmer-Verlet on

$$\dot{\Omega} = d\exp_\Omega^{-1}\big(A(t)\big) = B_0 A(t) = A(t)$$

instead of (2.27).

Cayley transform

An alternative to the exponential map, which does not demand to truncate an infinite series for numerical compuation, is given by the Cayley-transform, which is defined for quadratic Lie groups $G = \{U : U^H P U = P\}$ with a given constant matrix P. The corresponding Lie-Algebra is given by $\mathfrak{g} = \{\Omega : P\Omega + \Omega^H P = 0\}$.

Lemma 2.13 (Cayley transform) *Let G be a quadratic Lie group. The Cayley transform*

$$cay\colon \mathfrak{g} \to G, \quad cay(\Omega) = (I - \Omega)^{-1}(I + \Omega)$$

maps elements of the Lie algebra $\mathfrak{g}$ to the corresponding group G. In addition, cay *is a local diffeomorphism in a neighborhood of $\Omega = 0$.*

Proof: Let $\Omega \in \mathfrak{g}$, i.e., $P\Omega = -\Omega^H P$. Therefore we have

$$\text{(i) } P(I + \Omega) = (I - \Omega)^H P \quad \text{and} \quad \text{(ii) } P(I - \Omega)^{-1} = (I + \Omega)^{-H} P.$$

With $U = (I - \Omega)^{-1}(I + \Omega) = (I + \Omega)(I - \Omega)^{-1}$ (note that the matrices are commuting) we get

$$\begin{aligned}
U^H P U &= \big((I - \Omega)^{-1}(I + \Omega)\big)^H P (I - \Omega)^{-1}(I + \Omega) \\
&= \big((I - \Omega)^{-1}(I + \Omega)\big)^H (I + \Omega)^{-H} P (I + \Omega) && \text{(due to (ii))} \\
&= \big((I - \Omega)^{-1}(I + \Omega)\big)^H (I + \Omega)^{-H} (I - \Omega)^H P && \text{(due to (i))} \\
&= \big((I + \Omega)^{-H}(I - \Omega)^H\big)^{-1} (I + \Omega)^{-H} (I - \Omega)^H P \\
&= P
\end{aligned}$$

□

To use the Cayley transform as local parameterization in algorithm 2.1 we need its inverse:

Lemma 2.14 *The derivate of the Cayley transform $cay(\Omega)$ is given by*

$$\left(\frac{d}{d\Omega} cay(\Omega)\right) H = \big(d cay_\Omega(H)\big)\, cay(\Omega),$$

with

$$dcay_{\Omega}(H) = 2(I-\Omega)^{-1}H(I+\Omega)^{-1}.$$

For its inverse we have

$$dcay_{\Omega}^{-1}(H) = \frac{1}{2}(I-\Omega)H(I+\Omega).$$

Proof:

Using the chain rule for the derivative of Ω^k

$$\left(\frac{d}{d\Omega}\Omega^k\right)H = H\Omega^{k-1} + \Omega H\Omega^{k-2} + \ldots + \Omega^{k-1}H$$

we get

$$\left(\frac{d}{d\Omega}\mathrm{cay}(\Omega)\right)H = (I-\Omega)^{-1}H(I-\Omega)^{-1}(I+\Omega) + (I-\Omega)^{-1}H.$$

Hence we have to show the equivalence

$$\big(d\mathrm{cay}_{\Omega}(H)\big)\mathrm{cay}(\Omega) = (I-\Omega)^{-1}H(I-\Omega)^{-1}(I+\Omega) + (I-\Omega)^{-1}H$$

i.e.,

$$\Big(2(I-\Omega)^{-1}H(I+\Omega)^{-1}\Big)\Big((I-\Omega)^{-1}(I+\Omega)\Big) = (I-\Omega)^{-1}H(I-\Omega)^{-1}(I+\Omega)+(I-\Omega)^{-1}H.$$

Using the commutativity of the matrices we can simplify the equation:

$$2(I-\Omega)^{-1}H(I-\Omega)^{-1} = (I-\Omega)^{-1}H(I-\Omega)^{-1}(I+\Omega) + (I-\Omega)^{-1}H.$$

Mulitplying the left-hand side by $(I-\Omega)$ and changing again the matrices yields

$$2H(I-\Omega)^{-1} = H(I+\Omega)(I-\Omega)^{-1} + H.$$

A last right-hande multiplication by $(I-\Omega)$ shows the equivalence of both sides, as

$$2H = H(I+\Omega) + H(I-\Omega),$$
$$2H = H + H\Omega + H - H\Omega.$$

□

In analogy to theorem 2.11 for the exponential mapping we get

Theorem 2.15 *The solution of the differential equation $\dot{U} = A(t)U(t)$ with $U(t)$ being an element of the quadratic Lie group G and $A(t) \in \mathfrak{g}$ can be written as $U(t) = cay\big(\Omega(t)\big)U_0$ with $U_0 \in G$ and $\Omega(t)$ defined by*

$$\dot{\Omega} = dcay_{\Omega}^{-1}\big(A(t)\big) = \frac{1}{2}\big(I - \Omega(t)\big)A(t)\big(I + \Omega(t)\big), \quad \Omega(t_0) = 0. \tag{2.30}$$

Thus, with $\Psi := \text{cay}$, (2.20) the differential equation

$$\dot{\Omega} = d\text{cay}_{\Omega}^{-1}(A(t))$$

can be used in algorithm 2.1 instead of the exponential map. No infinite series arise, no truncation is needed. One can apply numerical schemes directly to this ODE system.

Application to lattice QCD

One application of Lie group differential equations is given by the lattice QCD equation of motion

$$\dot{U} = \frac{\partial \mathcal{H}(U,A)}{\partial A} = A(t) \cdot U(t), \quad \dot{A} = -\frac{\partial \mathcal{H}(U,A)}{\partial U} = g(U)$$

for a given Hamiltonian field $\mathcal{H}$. Here the differential equation for U describes an ODE on a manifold, given by the special unitary Lie group

$$\text{SU}(3,\mathbb{C}) = \{X \in \text{GL}(3,\mathbb{C}) : \det(X) = 1,\, X^H = X^{-1}\}$$

and the corresponding Lie-Algebra $\mathfrak{su}(3,\mathbb{C})$ of skew-symmetric Hermitian matrices. The differential equation for A is a differential equation living in the additive Lie algebra, which does not pose the problems of a multiplicative Lie group.

If we apply the algorith 2.1 to this setting, we get by using the Leap frog scheme as discretization scheme

- for the exponential mapping:

$$\begin{aligned} A_{n+\frac{1}{2}} &= A_n + \frac{h}{2} g(U_n), \\ \Omega_{n+1} &= \Omega_n + h \cdot d\exp_{\Omega}^{-1}(A_{n+\frac{1}{2}}) = \Omega_n + h \cdot A_{n+\frac{1}{2}}, \\ U_{n+1} &= \exp(\Omega_{n+1}) U_n, \\ A_{n+1} &= A_{n+\frac{1}{2}} + \frac{h}{2} g(U_{n+1}). \end{aligned} \tag{2.31}$$

  Eliminating the auxiliary Ω variable, the update (U_n, A_n) to (U_{n+1}, A_{n+1}) is given by

$$\begin{aligned} U_{n+1} &= \exp(h \cdot (A_n + \frac{h}{2} g(U_n))) U_n, \\ A_{n+1} &= A_n + \frac{h}{2} g(U_n) + \frac{h}{2} g(\exp(h \cdot (A_n + \frac{h}{2} g(U_n)))) \end{aligned}$$

- for the Cayley transform:

$$\begin{aligned} A_{n+\frac{1}{2}} &= A_n + \frac{h}{2} g(U_n), \\ \Omega_{n+1} &= \Omega_n + h \cdot d\mathrm{cay}_{\Omega}^{-1}(A_{n+\frac{1}{2}}) = \Omega_n + \frac{h}{2}(I - \Omega_n) A_{n+\frac{1}{2}} (I + \Omega_n), \\ U_{n+1} &= \mathrm{cay}(\Omega_{n+1}) U_n = (I - \Omega_{n+1})^{-1} (I + \Omega_{n+1}) U_n, \\ A_{n+1} &= A_{n+\frac{1}{2}} + \frac{h}{2} g(U_{n+1}). \end{aligned} \tag{2.32}$$

  Eliminating again the auxiliary Ω variable, the update (U_n, A_n) to (U_{n+1}, A_{n+1}) is given by

$$\begin{aligned} U_{n+1} &= \mathrm{cay}(h \cdot d\mathrm{cay}_{\Omega}^{-1}(A_n + \frac{h}{2} g(U_n))) U_n, \\ A_{n+1} &= A_n + \frac{h}{2} g(U_n) + \frac{h}{2} g(\mathrm{cay}(h \cdot d\mathrm{cay}_{\Omega}^{-1}(A_n + \frac{h}{2} g(U_n)))) \end{aligned}$$

The geometric properties to be preserved are now structure preservation (numerical approximations have to remain in the Lie group and Lie algebra, resp.), volume preservation and time reversibility. Whereas structure preservation is given by construction for both approaches, the latter have to be verified.

Volume preservation

The update of the variables U, A and Ω in (2.32) and (2.31), resp., can be written as subsequent single updates follows (note that the Ω-update will be thrown away later on, as Ω is only an auxiliary variable):

$$\begin{pmatrix} U_n \\ A_n \\ \Omega_n \end{pmatrix} \xrightarrow{\alpha} \begin{pmatrix} U_n \\ A_{n+\frac{1}{2}} \\ \Omega_n \end{pmatrix} \xrightarrow{\beta} \begin{pmatrix} U_n \\ A_{n+\frac{1}{2}} \\ \Omega_{n+1} \end{pmatrix} \xrightarrow{\gamma} \begin{pmatrix} U_{n+1} \\ A_{n+\frac{1}{2}} \\ \Omega_{n+1} \end{pmatrix} \xrightarrow{\delta} \begin{pmatrix} U_{n+1} \\ A_{n+1} \\ \Omega_{n+1} \end{pmatrix}$$

Denoting by

$$\Theta := \frac{\partial(U_{n+1}, A_{n+1}, \Omega_{n+1})}{\partial(U_n, A_n, \Omega_n)}$$

the Jacobian of the overall step and by

$$\alpha := \left(\frac{\partial(U_n, A_{n+1/2}, \Omega_n)}{\partial(U_n, A_n, \Omega_n)} \right), \quad \beta := \left(\frac{\partial(U_n, A_{n+1/2}, \Omega_{n+1})}{\partial(U_n, A_{n+1/2}, \Omega_n)} \right),$$

$$\gamma := \left(\frac{\partial(U_{n+1}, A_{n+1/2}, \Omega_{n+1})}{\partial(U_n, A_{n+1/2}, \Omega_{n+1})} \right), \quad \delta := \left(\frac{\partial(U_{n+1}, A_{n+1}, \Omega_{n+1})}{\partial(U_{n+1}, A_{n+1/2}, \Omega_{n+1})} \right)$$

the Jacobians of the subsequent steps, we have to show that for both the exponential and the Cayley map $\Theta = \delta \cdot \gamma \cdot \beta \cdot \alpha$ it holds

$$|\det \Theta| = |\det \delta \cdot \det \gamma \cdot \det \beta \cdot \det \alpha| = 1.$$

Exponential map: In this case, the Jacobians read

$$\alpha = \begin{pmatrix} I & 0 & 0 \\ \frac{h}{2}g'(U_n) & I & 0 \\ 0 & 0 & I \end{pmatrix} \qquad \Rightarrow \det \alpha = 1,$$

$$\beta = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & hI & I \end{pmatrix} \qquad \Rightarrow \det \beta = 1,$$

$$\gamma = \begin{pmatrix} \exp(\Omega_{n+1}) & 0 & d\exp_\Omega(\Omega_{n+1})U_n \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \Rightarrow \det \gamma = \det \exp(\Omega_{n+1}),$$

$$\delta = \begin{pmatrix} I & 0 & 0 \\ \frac{h}{2}g'(U_{n+1}) & I & 0 \\ 0 & 0 & I \end{pmatrix} \qquad \Rightarrow \det \delta = 1.$$

Caylay map: In this case, the Jacobians read

$$\alpha = \begin{pmatrix} I & 0 & 0 \\ \frac{h}{2}g'(U_n) & I & 0 \\ 0 & 0 & I \end{pmatrix} \qquad \Rightarrow \det \alpha = 1,$$

$$\beta = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & h \cdot d^2\mathrm{cay}^{-1}_{\Omega A}(A_{n+1/2}) & I \end{pmatrix} \qquad \Rightarrow \det \beta = 1,$$

$$\gamma = \begin{pmatrix} \mathrm{cay}(\Omega_{n+1}) & 0 & d\mathrm{cay}_\Omega(\Omega_{n+1})U_n \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \Rightarrow \det \gamma = \det \mathrm{cay}(\Omega_{n+1}),$$

$$\delta = \begin{pmatrix} I & 0 & 0 \\ \frac{h}{2}g'(U_{n+1}) & I & 0 \\ 0 & 0 & I \end{pmatrix} \qquad \Rightarrow \det \delta = 1.$$

As exp and cay, resp., map $\mathfrak{g}$ to $G = \mathrm{SU}(3,\mathbb{C}) = \{X \in \mathrm{GL}(3,\mathbb{C}) : \det(X) = 1,\ X^H = X^{-1}\}$, we have $\det \exp(\Omega_{n+1}) = 1$ and $\det \mathrm{cay}(\Omega_{n+1}) = 1$, resp., and both schemes are volume preserving.

Time reversibility

To show time reversibility, we have to show that for the mapping

$$\Phi_h : (U_n, A_n) \to (U_{n+1}, A_{n+1})$$

given by the exponential and Cayley map, resp.,

$$\Phi_h \circ \rho \circ \Phi_h = \rho$$

holds. If the scheme is symmetric, i.e., $\Phi_h^{-1} = \Phi_{-h}$, then time-reversibility is equivalent to

$$\rho \circ \Phi_h = \Phi_{-h} \circ \rho \tag{2.33}$$

Symmetry for both schemes is given if exchanging the subscripts $n \leftrightarrow n+1$ and step size $h \leftrightarrow -h$ leaves the methods unaltered. This is the case for both the exponential and the Cayley map, as can be easily verified.

The condition (2.33) holds for both schemes. This is a consequence of the fact that all symmetric partitioned Runge-Kutta schemes are time reversible for time reversible systems. Or it can be shown directly: for the exponential approach, for example, we have

$$\rho\Phi_h(U_n, A_n) = \; = \; \begin{pmatrix} \exp\left(h(A_n + \frac{h}{2}g(U_n))\right) U_n \\ -\left(A_n + \frac{h}{2}\left(g(U_n) + g(\exp\left(h(A_n + \frac{h}{2}g(U_n))\right) U_n)\right)\right) \end{pmatrix}.$$

On the other hand, we get for

$$\Phi_{-h}\rho(U_n, A_n) \; = \; \begin{pmatrix} \exp\left((-h)((-A_n) + \frac{(-h)}{2}g(U_n))\right) U_n \\ (-A_n) + \frac{(-h)}{2}\left(g(U_n) + g(\exp\left((-h)((-A_n) + \frac{(-h)}{2}g(U_n))\right) U_n)\right) \end{pmatrix},$$

and thus $\rho \circ \Phi_h = \Phi_{-h} \circ \rho$ is verified. The corresponding proof for the Cayley transfor will be an exercise.

Chapter 3

Model Order Reduction

In many applications, the user is not directly interested in the solution of an initial-value problem

$$f(x, \dot{x}, u(t)) = 0, \qquad x(0) = x_0,$$

written as a general DAE system with given input function $u : [0, \infty) \to \mathbb{R}^m$, but on an output function depending on the state $x : [0, \infty) \to \mathbb{R}^n$ and possible input u:

$$y(t) = h(x(t), u(t)), \qquad y : [0, \infty) \to \mathbb{R}^p$$

with usually $p \ll n$.

To start with, we restrict ourselves to a linear input-output system of the type

$$\begin{aligned} 0 &= E\dot{x}(t) + Ax(t) + Bu(t), && (3.1) \\ y(t) &= Cx(t) + Du(t) && (3.2) \end{aligned}$$

with $A, E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. In many applications, there is not direct feedthrough of the input to the output, i.e.,

$$D = 0 \in \mathbb{R}^{p\times m},$$

and $p = m$ with $C = B^\top \in \mathbb{R}^{p\times n}$.

One is not interested in the state $x(t)$ at a given time point t, but in the output $y(t)$ depending on the input u. The task of model order reduction is now to approximate y by an approximate function $\tilde{y}$ based on a dynamical system of lower dimension, i.e.,

$$\begin{aligned} 0 &= \tilde{E}\dot{\tilde{x}}(t) + \tilde{A}\tilde{x}(t) + \tilde{B}u(t), && (3.3)\\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t) + \tilde{D}u(t) && (3.4) \end{aligned}$$

with $A, E \in \mathbb{R}^{r\times r}$, $B \in \mathbb{R}^{r\times m}$, $C \in \mathbb{R}^{p\times r}$ and $D \in \mathbb{R}^{p\times m}$ with $r \ll n$.

A characterisation of the approximation error $\tilde{y} - y$ can be derived by transfering the system from the time to the frequency domain by using the Laplace transform. Remember that the Laplace transform for a function $f : [0, \infty) \to \mathbb{C}$ with $f(0) = 0$ is defined by

$$F(s) := \mathfrak{L}\{f\}(s) = \int_0^\infty f(t) \exp(-st) dt.$$

For a vector-valued function $F = (f_1, \ldots, f_q)^\top$, the Laplace transform is defined component-wise: $F(s) = (\mathfrak{L}\{f_1\}(s), \ldots, \mathfrak{L}\{f_q\}(s))^\top$.

Taking now the Laplace transform of the time domain representation of the linear problem (3.1) we obtain the following frequence domain representation (with $s = i\omega$ where $\omega \geq 0$ is referred to as the (angular) frequency):

$$\begin{aligned} 0 &= sEX(s) + AX(s) + BU(s), && (3.5)\\ Y(s) &= CX(s) + Du(s), && (3.6) \end{aligned}$$

where $X(s), U(s), Y(s)$ are the Laplace transforms of the states, the input and the output, respectively. Note that we assumed zero initial conditions, i.e., $x(0) = 0, u(0) = 0$ and $y(0) = 0$.

Eliminating the variable $X(s)$ in the frequency domain representation we see that the system's response to the input $U(s)$ in the frequency domain is fiven by

$$Y(s) = H(s)U(s)$$

with the matrix-valued transfer function

$$H(s) = -C(sE + A)^{-1}B + D \in \mathbb{C}^{p\times m}. \quad (3.7)$$

For the approximate function $\tilde{y}(t)$ we get the approximate transfer function

$$\tilde{H}(s) = -\tilde{C}(s\tilde{E} + \tilde{A})^{-1}\tilde{B} + \tilde{D} \in \mathbb{C}^{p\times m}. \tag{3.8}$$

The approximation error between $\tilde{y}$ and y in the time domain can now be estimated by the approximation error of the transfer functions in the frequency domain:

Theorem 1 (Approximation error) *Let* $||u||_{L^2([0,\infty))} < \infty$ *and* $U(i\omega) = 0$ *for* $\omega \neq I_\omega$. *If the system* (3.1) *consists of ODEs, then we have the estimate*

$$\max_{t>0} |y(t) - \tilde{y}(t)| \leq \left(\frac{1}{2\pi}\int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)|^2 d\omega\right)^{\frac{1}{2}} \cdot ||u||_{L^2([0,\infty))} \tag{3.9}$$

Proof. We obtain by using the Cauchy-Schwarz inequality in $L^2(I_\omega)$

$$\begin{aligned}
\max_{t>0} |y(t) - \tilde{y}(t)| &< \max_{t>0} \left|\frac{1}{2\pi}\int_{\mathbb{R}} (Y(i\omega) - \tilde{Y}(i\omega))e^{i\omega t} d\omega\right| \\
&< \frac{1}{2\pi}\int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)| \cdot |U(i\omega)| d\omega \\
&< \frac{1}{2\pi}\left(\int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)| d\omega\right)^{\frac{1}{2}} \left(\int_{I_\omega} |U(i\omega)|^2 d\omega\right)^{\frac{1}{2}} \\
&< \left(\frac{1}{2\pi}\int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)|^2 d\omega\right)^{\frac{1}{2}} \cdot ||u||_{L^2([0,\infty))}. \qquad \square
\end{aligned}$$

The question is now how to derive an approximation $\tilde{y}$ with an approximate transfer function $\tilde{H}$, which makes the right-hand side of (3.9) small enough for our accuracy demands? Here a variety of methods is available. We will discuss projection based MOR techniques only in the following.

3.1 Projection based MOR

The concept of all projection based MOR techniques is to approximate the high dimensional state space vector $x(t) \in \mathbb{R}^n$ with the help of a vector

$z(t) \in \mathbb{R}^r$ of reduced dimension $r \ll n$, within the meaning of

$$x(t) \approx \tilde{x}(t) := Vz(t) \quad \text{with } V \in \mathbb{R}^{n \times r}.$$

Note that the first approximation may be interpretet as a wish. We will only aim for $y(t) \approx \tilde{y}(t) = CVz(t) + \tilde{D}u(t)$. The columns of the matrix V are a basis of a subspace $\tilde{\mathcal{M}} \subseteq \mathbb{R}^n$, i. e., the state space $\mathcal{M}$, the solution $x(t)$ of the differential equation (3.1) resides in, is projected on $\tilde{\mathcal{M}}$. A reduced order model, representing the full problem (3.1) results from deriving a state space equation that determines the reduced state vector $z(t)$ such that $\tilde{x}(t)$ is a reasonable approximation to $x(t)$.

If we insert $\tilde{x}(t)$ on the right-hand side of the dynamic part of the input-output problem (3.1), it will not vanish identically. Instead we get a residual:

$$r(t) := EV\dot{z}(t) + AVz(t) + Bu(t) \in \mathbb{R}^n.$$

We can not demand $r(t) \equiv 0$ in general as this would state an overdetermined system for $z(t)$. Instead we apply the Petrov-Galerkin technique, i. e., we demand the residual to be orthogonal to some testspace $\mathcal{W}$. Assuming that the columns of a matrix $W \in \mathbb{R}^{n \times r}$ span this testspace, the mathematical formulation of this orthogonality becomes

$$0 = W^\top r(t) = W^\top (EV\dot{z}(t) + AVz(t) + Bu(t)) \in \mathbb{R}^r,$$

which states a differential equation for the reduced state $z(t)$.

Defining

$$\begin{aligned} &\tilde{E} := W^\top EV \in \mathbb{R}^{r \times r}, \quad \tilde{A} := W^\top AV \in \mathbb{R}^{r \times r}, \\ &\tilde{B} := W^\top B \in \mathbb{R}^{r \times m}, \quad \tilde{C} := CV \in \mathbb{R}^{p \times r}, \\ &\tilde{D} := D \in \mathbb{R}^{p \times m}, \end{aligned} \tag{3.10}$$

we arrive at the reduced order model (3.3).

To relate V and W we demand biorthogonality of the spaces $\mathcal{V}$ and $\mathcal{W}$ spanned by the columns of the two matrices, respectively, i. e. $W^\top V = I_r$. With this, the reduced problem (3.3) is the projection of the full problem (3.1) onto $\mathcal{V}$ along $\mathcal{W}$. If an orthonormal V and $W = V$ is chosen, we speak

of a orthogonal projection on the space $\mathcal{V}$ (and we come down to a Galerkin method).

Now, MOR projection methods are characterised by the way of how to construct the matrices V and W that define the projection. In the following we find a short introduction of Krylov methods and POD approaches. The former starts from the frequency domain representation, the latter from the time domain formulation of the input-output problem.

3.2 Krylov method

Krylov-based methods to MOR are based on a series expansion of the transfer function H. The idea is to construct a reduced order model such that the series expansions of the transfer function $\tilde{H}$ of the reduced model and the full problem's transfer function agree up to a certain index of summation.

In the following we will assume that the system under consideration does not have a direct feedthrough, i. e., $D = 0$ is satisfied. Furthermore we restrict to SISO systems, i. e, single input single output systems. In this case we have $p = m = 1$, i. e., $B = b$ and $C = C^H$ where $b, c \in \mathbb{R}^n$, and the (scalar) transfer function becomes:

$$H(s) = -c^H (sE + A)^{-1} b \quad \in \mathbb{C},$$

As $\{E, A\}$ is a regular matrix pencil we find some frequency s_0 such that $s_0E + A$ is regular. Then the transfer function can be reformulated as

$$H(s) = l\,(I_n - (s - s_0)F)^{-1}\, r, \tag{3.11}$$

with $l := -c^H$, $r := (s_0E + A)^{-1}b$ and $F := -(s_0E + A)^{-1}E$.

In a neighbourhood of s_0 one can replace the matrix inverse in (3.11) by the corresponding Neumann series. Hence, a series expansion of the transfer function is

$$H(s) = \sum_{k=0}^{\infty} m_k (s - s_0)^k \quad \text{with} \quad m_k := l\, F^k\, r \quad \in \mathbb{C}. \tag{3.12}$$

The quantities m_k for $k = 0, 1, \ldots$ are called moments of the transfer function.

A different model, of lower dimension, can now be considered to be an approximation to the full problem, if the moments $\tilde{m}_k$ of the new model's transfer function $\tilde{H}(s)$ agree with the moments m_k defined above, for $k = 1, \ldots, q$ for some $q \in \mathbb{N}$.

Expressions like F^k or lF^k arise also in methods, namely in Krylov-subspace-methods, which are used for the iterative solution of large algebraic equations. Here the Lanczos- and the Arnoldi-method are algorithms that compute biorthogonal bases W, V or a orthonormal basis V of the μth left and/or right Krylov subspaces

$$\mathcal{K}_l(F^\top, l^\top, \mu) := \text{span}\left(l^\top, F^\top l^\top, \ldots, \left(F^\top\right)^{\mu-1} l^\top\right),$$
$$\mathcal{K}_r(F, r, \mu) := \text{span}\left(r, Fr, \ldots, F^{\mu-1} r\right),$$

for $\mu \in \mathbb{N}$, respectively in a numerically robust way.

The Krylov subspaces, thus "contain" the moments m_k of the transfer function and it can be shown that from applying Krylov-subspace methods, reduced order models can be created. These reduced order models, however, did not arise from a projection approach. In fact, the Lanczos- and the Arnold-algorithm produces besides the matrices W and/or V whose columns span the Krylov subspaces $\mathcal{K}_l$ and/or $\mathcal{K}_r$, respectively, a tridiagonal or an upper Hessenbergmatrix $\mathcal{T}$, respectively. This matrix is then used to postulate a dynamical system whose transfer function has the desired matching property.

Concerning the moment matching property there is a difference for reduced order models created from a Lanczos- and those created from an Arnoldi-based process.

For a fixed q, the Lanczos-process constructs the qth left and the qth right Krylov-subspace, hence biorthogonal matrices $W, V \in \mathbb{R}^{n\times q}$. A reduced order model of order q, arising from this procedure possesses a transfer function $\tilde{H}(s)$ whose first $2q$ moments coincide with the first $2q$ moments

of the original problem's transfer function $H(s)$, i. e. $\tilde{m}_k = m_k$ for $k = 0, \ldots, 2q-1$. Hence, the Lanczos MOR model yields a Padé approximation.

The Arnoldi method on the other hand is a one sided Krylov subspace method. For a fixed q only the qth right Krylov subspace is constructed. As a consequence, here only the first q moments of the original system's and the reduced system's transfer function match.

The main drawbacks of these methods are, in general, lack of provable error bounds for the extracted reduced models.

3.3 Proper Orthogonal Decomposition

While the Krylov approaches are based on the matrices, i. e., on the system itself, the method of Proper Orthogonal Decomposition (POD) is based on the trajectory $x(t)$, i. e., the outcome of the system (3.1). One could also say that the Krylov methods are based on the frequency domain, whereas POD is based on the time domain formulation of the input output system to be modelled.

POD first collects data $\{x_1, \ldots, x_K\}$. The datapoints are snapshots of the state space solution $x(t)$ of the network equation (3.1) at different timepoints t or for different input signals $u(t)$. They are usually constructed by a numerical time simulation, but may also arise from measurements of a real physical system.

From analysing this data, a subspace is created such that the data points as a whole are approximated by corresponding points in the subspace in a optimal least-squares sense. The basis of this approach is also known as Principal Component Analysis and Karhunen–Loève Theorem from picture and data analysis.

The mathematical formulation of POD is as follows: Given a set of K datapoints $X := \{x_1, \ldots, x_K\}$ a subspace $\mathcal{S}_r \subset \mathbb{R}^n$ of dimension r is searched

for that minimizes

$$\|X - \varrho_r X\|_2^2 := \sum_{i=1}^{n} \mu_i \tag{3.13}$$

where $\varrho_r : \mathbb{R}^n \to \mathcal{S}_r$ is the orthogonal projection onto $\mathcal{S}_r$ and $\mu_1 \geq \ldots \geq \mu_n$ are the eigenvalues of the semi positive-definite matrix $(X - \varrho X)^\top (X - \varrho X)$.

We will not describe POD in full detail here, as in literature this is well explained. However, the key to solving this minimization problem is the computation of the eigenvalues λ_i and eigenvectors φ_i (for $i = 1, \ldots, n$) of the correlation matrix XX^T:

$$XX^T \varphi_i = \lambda_i \varphi_i,$$

where the eigenvalues and eigenvectors are sorted such that $\lambda_1 \geq \cdots \geq \lambda_n$. The matrix X is defined as $X := (x_1, \ldots, x_K) \in \mathbb{R}^{n \times K}$ and is called snapshot matrix.

Intuitively the correlation matrix detects the principal directions in the data cloud that is made up of the snapshots $x_1, \ldots, x_K$. The eigenvectors and eigenvalues can be thought of as directions and radii of axes of an ellipsoid that incloses the cloud of data. Then, the smaller the radii of one axis is, the less information is lost if that direction is neglected.

The question arises, how many directions r should be kept and how many can be neglected. There is no a-priori error bound for the POD reduction. However, the eigenvalues are a measure for the relevance of the dimensions of the state space. Hence, it seems reasonable to choose the dimension r of the reduced order model in such a way, that the relative information content of the reduced model with respect to the full system is high. The measure for this content, used in the literature cited above is

$$\mathcal{I}(r) = \frac{\lambda_1 + \cdots \lambda_r}{\lambda_1 + \cdots \lambda_r + \lambda_{r+1} + \cdots \lambda_n}.$$

Clearly, a high relative information content means to have $\mathcal{I}(r) \approx 1$. Typically r is chosen such that this measure is around 0.99 or 0.995.

If the eigenvalues and eigenvectors are available and a dimension r has been

chosen, the projection matrices V and W are taken as

$$V := W := (\varphi_1, \ldots, \varphi_r) \in \mathbb{R}^{n \times r}.$$

leading to an orthogonal projection $\varrho_r = VV^\top$ on the space $\mathcal{S}_r$ spanned by $\varphi_1, \ldots, \varphi_r$.

The procedure described so far relies on the eigenvalue decomposition of the $n \times n$ matrix XX^T. This direct approach is feasible only for problems of moderate size. For high dimensional problems, i. e., for dimensions $n \gg 1$, the eigenvalues and eigenvectors are derived form the Singular Value Decomposition (SVD) of the snapshot matrix $X \in \mathbb{R}^{n \times K}$.

The SVD provides three matrices:

$$\begin{aligned}
\Phi &= (\varphi_1, \cdots, \varphi_n) \in \mathbb{R}^{n \times n} \quad \text{orthogonal,} \\
\Psi &= (\psi_1, \cdots, \psi_K) \in \mathbb{R}^{K \times K} \quad \text{orthogonal,} \\
\Sigma_\nu &= \operatorname{diag}(\sigma_1, \ldots, \sigma_\nu) \in \mathbb{R}^{\nu \times \nu} \quad \text{with } \sigma_1 \geq \cdots \geq \sigma_\nu > \sigma_{\nu+1} = \ldots = \sigma_K = 0,
\end{aligned}$$

such that

$$X = \Phi \begin{pmatrix} \Sigma_\nu & 0 \\ 0 & 0 \end{pmatrix} \Psi^T \top \tag{3.14}$$

where the columns of Φ and Ψ are the left and right singular eigenvectors, respectively, and $\sigma_1, \ldots, \sigma_\nu$ are the singular values of X (σ_ν being the smallest non-zero singular value; this also defines the index ν). It follows that $\varphi_1, \ldots, \varphi_n$ are eigenvectors of the correlation matrix XX^T with the n eigenvalues $\sigma_1^2, \ldots, \sigma_\nu^2, 0, \ldots, 0$.

With the help of the SVD of X, we can show that the choice $\varrho_r = VV^\top$

indeed minimizes the norm:

$$\begin{aligned} X - \varrho X &= X - VV^\top X = \\ &= X - \left(\Phi \begin{pmatrix} I_r \\ 0 \end{pmatrix}\right)\left(\Phi \begin{pmatrix} I_r \\ 0 \end{pmatrix}\right)^\top X \\ &= \Phi \begin{pmatrix} \Sigma_\nu & 0 \\ 0 & 0 \end{pmatrix} \Psi^\top - \Phi \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \Psi^\top \Rightarrow \\ \|X - \varrho X\|_2^2 &= \left\| \begin{pmatrix} \Sigma_\nu - \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \right\|_2^2 \\ &= \sum_{i=r+1}^{\nu} \sigma_i^2 = \sum_{i=r+1}^{\nu} \lambda_i. \end{aligned}$$

Consequently, the singular values of X and ϱX coincide up to r: $\mu_1 = \lambda_1, \ldots, \mu_r = \lambda_r$, $\mu_{r+1} = \ldots = \mu_K = 0$.

It remains to show to define now the reduced system: approximating x by Vx, $x \approx V(x)$, in (3.1) and multiplying from the left-hand side with $V^\top$, we get

$$\begin{aligned} 0 &= \tilde{E}\dot{\tilde{x}}(t) + \tilde{A}x(t) + \tilde{B}u(t), && (3.15) \\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t) + \tilde{D}u(t). && (3.16) \end{aligned}$$

with $\tilde{E} := V^\top EV$, $\tilde{A} := V^\top AV$, $\tilde{B} := V^\top B$, $\tilde{C} := CV$ and $\tilde{D} = D$. In the linear ODE case, i.e. $E = I_n$, we have due to $V^\top V = I_r$

$$\begin{aligned} 0 &= \dot{\tilde{x}}(t) + \tilde{A}x(t) + \tilde{B}u(t), && (3.17) \\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t) + \tilde{D}u(t). && (3.18) \end{aligned}$$

3.4 The nonlinear case

Model order reductions schemes for nonlinear systems of the type

$$\begin{aligned} \dot{x} &= f(x(t), u(t)) && (3.19) \\ y &= g(x(t), u(t) && (3.20) \end{aligned}$$

are still in the core of current research. In the following we will briefly sketch how to generalize the POD approach to this nonlinear setting. Again, we decompose the corrolation matrix $XX^\top$ of snapshots as

$$X = \Phi \begin{pmatrix} \Sigma_\nu & 0 \\ 0 & 0 \end{pmatrix} \Psi^T \top$$

and approximate x by $x = V\tilde{x}$ with V consisting of the first r columns of Φ. With $V^\top V = I_r$ we get than the approximative ODE system

$$\dot{\tilde{x}} = V^\top f(V\tilde{x}(t), u(t))$$

for $\tilde{x}$. However, we are now faced by the problem that evaluating the nonlinear term $V^\top f(V\tilde{x}(t), u(t))$ still contains the dimension n, as V maps from $\mathbb{R}^n$ to $\mathbb{R}^r$ and $f(V\tilde{x}(t), u(t))$ is a vector in $\mathbb{R}^n$. To overcome this problem, we apply Discrete Empirical Interpolation, for short DEIM. This works as follows.

We approximate the nonlinear function $f(t) := f(V\tilde{x}(t), u(t))$ (note that we distinguish two different functions f depending on whether they depend on one or two arguments!) by projecting onto a lower dimensional manifold defined by $U \in \mathbb{R}^{n\times m}$ with $m \ll n$:

$$f(t) \approx \hat{f}(t) := Uc(t).$$

Then we get for the right-hand side

$$V^\top f(t) \approx V^\top U \cdot c(t).$$

The matrix $V^\top U \in \mathbb{R}^{r\times m}$ can be computed a priori, and one only has to do matrix-vector multiplications of lower dimension in the following.

Now two questions arise. How to choose U and how to choose c?

- U can be defined, in analogy to the linear case, by expanding the correlation matrix $FF^\top$, with F consisting of snapshots of f, and taking the leading m left eigenvectors of the SVD to define U.

- c is defined by first selecting the most important m rows $p_1, \ldots, p_m$ of U. This defines the selection matrix $P = (e_{p_1}, \ldots, e_{p_m}) \in \mathbb{N}^{n \times m}$. The core of DEIM is the construction of this matrix. A set of indices $p_1, \ldots, p_m \subset \{1, \ldots, n\}$ define the selection matrix, meaning that P has a 1 in the i-th column and p_i-th row (for $i = 1, \ldots, m$) and 0 elsewhere. The first index, p_1 is chosen to be the index of the largest (in absolute value) entry in u_1 . In step $l = 2, \ldots, m$ the residual

  $$r_l = u_{l+1} - U_l(P_l^\top U_l)^{-1} P_l^\top u_{l+1}$$

  of the best-approximation of u_{l+1} in the subspace spanned by the columns given by $U_l = (u_1, \ldots, u_l)$ is computed , i.e,, $u_{l+1} = U_l\alpha$ with α given by $P_l^\top u_{l+1} = P_l^\top U_l \alpha$, where $P_l = (e_{p_1}, \ldots, e_{p_l})$ is constructed from the indices $p_1, \ldots, p_l$. Then, the index corresponding to entry of the residualr_l with the largest magnitude is taken as index p_{l+1} . Setting up the selection matrix with this algorithm, $P^\top U$ is guaranteed to be regular.

  c is then given as the unique solution of the m-dimensional linear system

  $$\left(P^\top U\right) c(t) = P^\top f(t).$$

Summing up, the approximate lower dimension ODE system is then given by

$$\dot{\tilde{x}} = V^\top \underbrace{U \left(P^\top U\right)^{-1} P^\top}_{\mathbb{P} :=} f(t)$$

with the projection operator $\mathbb{P} := U(P^\top U)^{-1}P^\top$ onto the submanifold spanned by the columns of U. Indeed, $\mathbb{P}$ defines an interpolation, as the approximation is exact at entries $p_1, \ldots, p_m$:

$$P^\top \hat{f}(t) = P^\top U c(t)) = P^\top \mathbb{P} f(t) = P^\top f(t),$$

as $P^\top \mathbb{P} = P^\top$ holds.

Reference:
M. Günter (ed.): Coupled Multiscale Simulation and Optimization in Nanoelectronics. Springer, Heidelberg, 2015 (Chapters 4 and 6).

Chapter 4

Multirate Schemes

Throughout this chapter we consider dynamical systems described by the initial-value problem of ordinary differential equations (ODEs)

$$\dot{\mathbf{y}}(t) = f\big(t, \mathbf{y}(t)\big), \quad t \in [t_0, t_{end}], \quad \mathbf{y}(t_0) = \mathbf{y}_0, \quad \mathbf{y}(t) \in \mathbb{R}^n, \tag{4.1}$$

with $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ Lipschitz continuous in $\mathbf{y}$ to obtain a unique solution.

4.1 Types of multirate behaviour

Many dynamical systems (4.1) display a multiple time scale dynamics, with some parts of the system evolving at faster pace and other evolving at a slower pace. Different time scales may be associated with different activity levels of various components (e.g., fast signals in active transistors and slow voltage changes in latent parts of an integrated circuit), or with different processes that drive the dynamics (e.g., fast chemical reactions and slow tracer transport driving pollutant concentrations in the atmosphere).

Multirate time discretization schemes exploit the different time scales in the dynamics of a differential equation model by adapting the computational costs to different activity levels of the system. The goal is to considerably improve the overall computational efficiency.

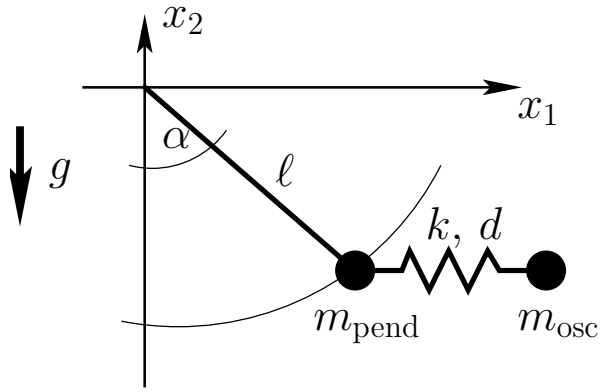


Figure 8: Mathematical pendulum coupled to an oscillator (taken after [**?**]).

4.1.1 Multiscale dynamics with partitioned components

An ODE system (4.1) where different components display a multiscale behavior, (4.1) can be partitioned into slow and fast components $\mathbf{y}_{\text{S}}$ and $\mathbf{y}_{\text{F}}$

$$\begin{aligned} \dot{\mathbf{y}}_{\text{S}} &= \mathbf{f}_{\text{S}}(t, \mathbf{y}_{\text{S}}, \mathbf{y}_{\text{F}}), \qquad \mathbf{y}_{\text{S}}(t_0) = \mathbf{y}_{\text{S},0}, \\ \dot{\mathbf{y}}_{\text{F}} &= \mathbf{f}_{\text{F}}(t, \mathbf{y}_{\text{S}}, \mathbf{y}_{\text{F}}), \qquad \mathbf{y}_{\text{F}}(t_0) = \mathbf{y}_{\text{F},0}. \end{aligned}$$

Here the multirate potential lies in numerically integrating the slow components by using large step sizes, whereas small step sizes have to be used to approximate the fast components accurately enough.

As an example of a system with component-wise multiscale behavior we consider a mathematical pendulum of constant length ℓ that is coupled to a damped oscillator with a horizontal degree of freedom, as illustrated in Figure 8. The system consists of two rigid bodies: the first mass m_{pend} is connected to a second mass m_{osc} by a soft spring.

The minimal set of coordinates $q = [\alpha, x_1]^T$ uniquely describe the position of both bodies. The equations of motion given by the Euler-Lagrange equation read:

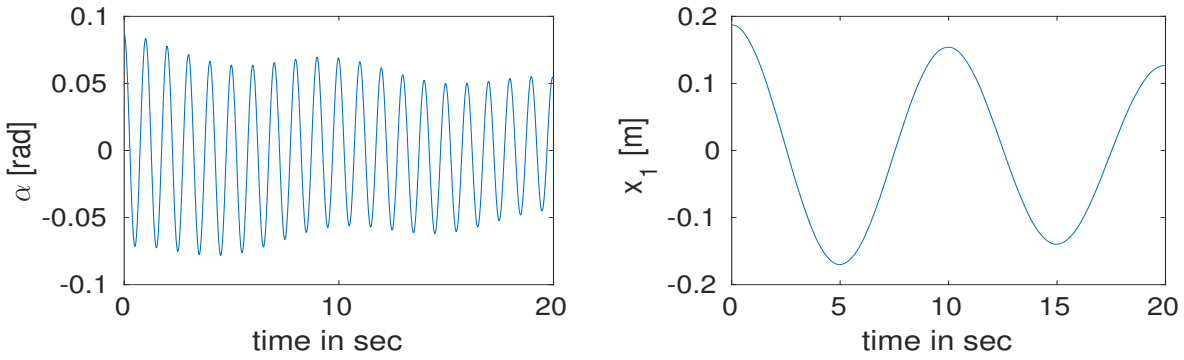
$$\begin{pmatrix} m_{\text{pend}}\,\ell & 0 \\ 0 & m_{\text{osc}} \end{pmatrix} \ddot{q} = \begin{pmatrix} -m_{\text{pend}}\,g\,\sin(\alpha) + \cos(\alpha\,F) \\ -F \end{pmatrix} =: f(q),$$

with the spring-damper force F modeled by

$$F = k\left(x_1 - \ell\,\sin(\alpha)\right) + d\left(\dot{x}_1 - \ell\,\dot{\alpha}\,\cos(\alpha)\right),$$

where k denotes the spring stiffness and d is the coefficient of friction. This two-dimensional second order system of differential equations can be easily transformed into a four-dimensional ODE system (4.1) by introducing the derivatives of q as additional unknowns.

The time evolution of the two components is illustrated in Figure 9. We see that $\alpha(t)$ oscillates quickly, while $x_1(t)$ oscillates slowly.



(a) First angle $\alpha_1(t)$ (b) First coordinate $x_1(t)$

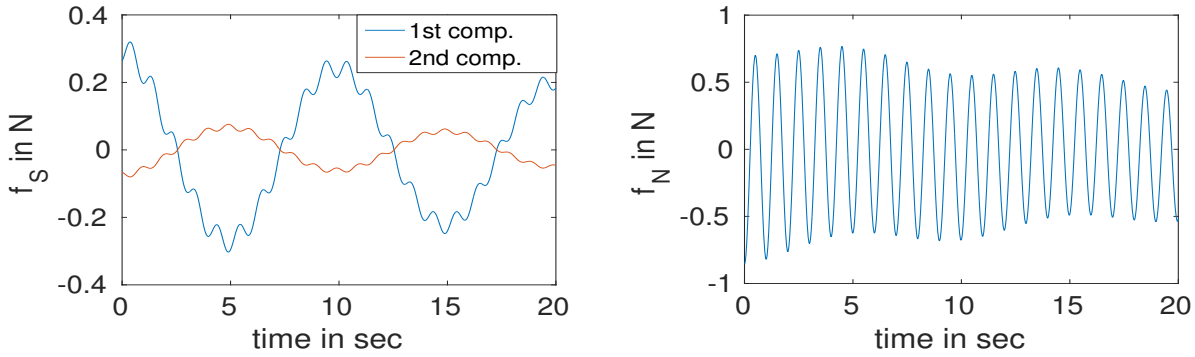
Figure 9: Time evolution of components of the two-body problem .

4.1.2 Multiscale dynamics with multiple physical processes

The right-hand-side $\mathbf{f}(\mathbf{y})$ of (4.1) represents the physical processes that drive the evolution of the system. A second cause of multiscale dynamics is due to the system being driven by multiple physical processes with different dynamics. In this case the right hand side $\mathbf{f}(\mathbf{y})$ can be split into two parts of different activity levels:

$$\mathbf{f}(t, \mathbf{y}) = \mathbf{f}_{\text{S}}(t, \mathbf{y}) + \mathbf{f}_{\text{F}}(t, \mathbf{y}).$$

Here $\mathbf{f}_{\text{F}}$ defines a fast changing process that is inexpensive to evaluate, whereas $\mathbf{f}_{\text{S}}$ is a slowly changing process, but is expensive to evaluate. The multirate potential lies in evaluating the cheap and fast part $\mathbf{f}_{\text{F}}$ more often then the slow and expensive part $\mathbf{f}_{\text{S}}$.



(a) Components of slow force $\mathbf{f}_{\mathrm{S}}(t, \mathbf{y}(t))$

(b) Fast force (1st component of $\mathbf{f}_{\mathrm{F}}(t, \mathbf{y}(t))$)

Figure 10: Time evolution of slow and fast forces acting on the two-body system.

Recall the rigid body example. The right hand side is also a source of multirate behavior. Specifically, the right hand side $f(q)$ can be split into two parts $f(q) = \mathbf{f}_{\mathrm{S}}(q) + \mathbf{f}_{\mathrm{F}}(q)$ with

$$\mathbf{f}_{\mathrm{S}}(q) = \begin{pmatrix} \frac{\cos(\alpha) F}{m_{\mathrm{pend}} l} \\ -\frac{F}{m_{\mathrm{osc}}} \end{pmatrix} \quad \text{and} \quad \mathbf{f}_{\mathrm{F}}(q) = \begin{pmatrix} -g \sin(q_1) \\ 0 \end{pmatrix}$$

of different activity. As seen in Figure 10, $\mathbf{f}_{\mathrm{F}}$ defines a fast changing force, whereas $\mathbf{f}_{\mathrm{S}}$ is slowly changing.

4.1.3 Multiscale dynamics due to forcing

A third source of multiscale behavior may occur in initial-value problems of the form

$$\dot{\mathbf{y}} = \mathbf{f}\big(t, \mathbf{y}, s(t)\big),$$

with a multitone input (forcing) function $s(t)$ comprised of modulated signals of different frequencies. One example is the amplitude modulated signal

$$s(t) = \big(1 + \alpha \sin\left(2\,\pi\, t/T_1\right)\big) \cdot \sin\left(2\,\pi\, t/T_2\right), \quad T_1 \gg T_2.$$

Here a high-frequency oscillation with period T_2 is modulated in amplitude by a low-frequency oscillation with period T_1. The fixed parameter $\alpha \in (0, 1)$ defines the size of the modulation.

A standard numerical discretization will require many time steps to resolve all oscillations induced by the forcing in case of widely separated scales. The time steps will be driven by the highest frequency in the system that needs to be resolved accurately. However, this signal can be represented very efficiently by the bivariate function

$$\hat{s}(t_1, t_2) = \left(1 + \alpha \sin\left(2\,\pi\, t_1/T_1\right)\right) \sin\left(2\,\pi\, t_2/T_1\right),$$

where separate but dependent time variables have been introduced for each of the time scales. This representation is biperiodic and completely defined by its values in the rectangle $[0, T_1] \times [0, T_2]$: a coarse grid in time domain is sufficient for resolving this representation. As the original signal can be reconstructed by setting $s(t) := \hat{s}(t, t)$, the bivariate representation achieves an efficient multidimensional model of amplitude modulated signals.

In order to exploit the *multiscale behavior in the signals* we remodel the ODE system into a PDE system of multivariate functions

$$\begin{aligned} \frac{\partial \hat{\mathbf{y}}(t_1, t_2)}{\partial t_1} + \frac{\partial \hat{\mathbf{y}}(t_1, t_2)}{\partial t_2} &= \mathbf{f}(\hat{\mathbf{y}}(t_1, t_2), \hat{s}(t_1, t_2)), \\ \hat{\mathbf{y}}(0, t_2) &= \hat{\mathbf{y}}(T_1, t_2) \quad \forall t_2 \in [0, T_2], \\ \hat{\mathbf{y}}(t_1, 0) &= \hat{\mathbf{y}}(t_1, T_2) \quad \forall t_1 \in [0, T_1], \end{aligned}$$

which can be solved more efficiently than the original ODE problem. The sought ODE solution $\mathbf{y}(t)$ can be reconstructed from the solution of this biperiodic boundary-value problem by setting $\mathbf{y}(t) := \hat{\mathbf{y}}(t \mod T_1, t \mod T_2)$.

Exploiting the multiscale behavior of components, processes, and signals can be naturally generalized to DAE and PDE systems, as we will see by inspecting some instructive examples at the end of this chapter.

4.2 Multirate Euler schemes - the singlerate case revisited

We discuss the numerical solution of multiscale systems introduced above in Chapter. Multirate time discretization schemes exploit the different dynamical time scales in order to improve the overall computational efficiency

w ithout sacrificing the overall solution accuracy. To achieve this different time steps are used for different parts of the system, according to their activity levels: whereas faster parts are solved with smaller time steps, the slower parts use large time steps. We start with a presentation of the multirate approach in the context of the simple Euler integration schemes.

Numerical methods for solving the initial-value problem (4.1) are categorized as being either *explicit* or *implicit*, and the two families are represented by the explicit (forward) Euler and the implicit (backward) Euler methods, respectively. Assume we have already computed the numerical approximations $\mathbf{y}_i \approx \mathbf{y}(t_i)$ at the time points $t_i = t_0 + ih$ $(i = 1, \ldots, n)$. The approximation at the next time point t_{n+1} computed by the forward Euler method is

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\,\mathbf{f}\,(t_n, \mathbf{y}_n)\,. \tag{4.2}$$

The scheme (4.2) is explicit, i.e., $\mathbf{y}_{n+1}$ is explicitly given as a function of $\mathbf{y}_n$ and the corresponding function value. The intuitive interpretation is that the solution at the next step $\mathbf{y}_{n+1}$ is the result of freezing the velocity field $\mathbf{f}(t, \mathbf{y}(t))$ in (4.1) at $t = t_n$ and following it for a time h.

The backward Euler method computes the solution approximation at the next time point t_{n+1} by the formula

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\,\mathbf{f}\,(t_{n+1}, \mathbf{y}_{n+1})\,. \tag{4.3}$$

The scheme (4.3) is implicit, i.e., $\mathbf{y}_{n+1}$ is implicitly defined by solving the nonlinear equation (4.3) which has a unique solution provided that the step size h is small enough. The intuitive interpretation is that the velocity field $\mathbf{f}(t, \mathbf{y}(t))$ in (4.1) is frozen at $t = t_{n+1}$; the frozen field depends on the unknown solution $\mathbf{y}_{n+1}$. At each step one needs to solve the generally nonlinear algebraic system of equations (4.3) for $\mathbf{y}_{n+1}$. For example, a solution procedure based on a simplified Newton approach performs iterations of the form

$$\begin{aligned} \mathbf{y}_{n+1}^{(\nu+1)} \;=\; & \mathbf{y}_{n+1}^{(\nu)} - \left(\mathbf{I} - h\,\mathbf{f}_{\mathbf{y}}(t_n, \mathbf{y}_n)\right)^{-1} \left(\mathbf{y}_{n+1}^{(\nu)} - \mathbf{y}_n - h\,\mathbf{f}(t_{n+1}, \mathbf{y}_{n+1}^{(\nu)})\right), \\ & \nu = 0, 1, 2, \ldots, \end{aligned} \tag{4.4}$$

where $\mathbf{f}_\mathbf{y} = \partial\mathbf{f}/\partial\mathbf{y} \in \mathbb{R}^{d\times d}$ is the Jacobian of the right-hand side function (4.1). The solution process requires the (typically expensive) solution of linear systems involving the matrix $(\mathbf{I} - h\,\mathbf{f}_\mathbf{y})$ at each step.

The performance of a numerical scheme depends on two properties, accuracy and stability. They are discussed next.

4.2.1 Accuracy of Euler schemes

Accuracy refers to the size of the approximation error made when the differential equation (4.1) is replaced by a difference equation such as (4.2) or (4.3). The *local truncation error* is the approximation error introduced during a single step (the $n+1$-st step) of (4.2) when initialized with the exact solution:

$$\boldsymbol{\delta}_{n+1}^{\mathrm{FE}} = \mathbf{y}_{n+1}^{\mathrm{FE}} - \mathbf{y}(t_{n+1}) = \mathbf{y}(t_n) + h\,\mathbf{f}\big(t_n, \mathbf{y}(t_n)\big) - \mathbf{y}(t_{n+1}). \tag{4.5}$$

An expansion in Taylor series about t_n

$$-\boldsymbol{\delta}_{n+1}^{\mathrm{FE}} = \mathbf{y}(t_{n+1}) - \mathbf{y}(t_n) - h\,\dot{\mathbf{y}}(t_n) = \int_0^h (h-\tau)\ddot{\mathbf{y}}(t_n+\tau)\,d\tau \tag{4.6}$$

allows to bound the magnitude of the local error as follows:

$$\|\boldsymbol{\delta}_{n+1}^{\mathrm{FE}}\| \le \frac{h^2}{2} \max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}(\tau)\|. \tag{4.7}$$

Similarly, the local truncation error of the backward Euler scheme (4.3) is

$$\boldsymbol{\delta}_{n+1}^{\mathrm{BE}} = \mathbf{y}_{n+1}^{\mathrm{BE}} - \mathbf{y}(t_{n+1}) = \mathbf{y}(t_n) + h\,\mathbf{f}\big(t_{n+1}, \mathbf{y}(t_{n+1})\big) - \mathbf{y}(t_{n+1}\big). \tag{4.8}$$

A Taylor series about t_n

$$-\boldsymbol{\delta}_{n+1}^{\mathrm{BE}} = \mathbf{y}(t_{n+1}) - \mathbf{y}(t_n) - h\,\dot{\mathbf{y}}(t_{n+1}) = \int_0^h \tau\,\ddot{\mathbf{y}}(t_n+\tau)\,d\tau \tag{4.9}$$

allows to bound the magnitude of local truncation error as follows:

$$\|\boldsymbol{\delta}_{n+1}^{\mathrm{BE}}\| \le \frac{h^2}{2} \max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}(\tau)\|. \tag{4.10}$$

A numerical scheme has order p of consistency (has order p for short) if its local truncation error is $\boldsymbol{\delta}_n = \mathcal{O}(h^{p+1})$ [**?**]. From (4.6) and (4.9) we conclude that the Euler schemes (4.2) and (4.3) have *order one.*

In order to ensure that the error is smaller than the admissible level, $\|\boldsymbol{\delta}_{n+1}\| \leq \texttt{tol}$, the step size has to be chosen such that:

$$h \leq \sqrt{\frac{2\,\texttt{tol}}{\max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}(\tau)\|}}. \tag{4.11}$$

4.2.2 Stability of Euler schemes

Stability refers to local truncation errors not being amplified in subsequent steps [**?**]. To assess the stability of the forward Euler method consider the global errors, i.e., the differences between the numerical and the exact solutions at each time step $\boldsymbol{\varepsilon}_n^{\text{FE}} := \mathbf{y}_n - \mathbf{y}(t_n)$. Subtracting (4.6) from (4.2) leads to:

$$\begin{aligned}
\boldsymbol{\varepsilon}_{n+1}^{\text{FE}} &= \boldsymbol{\varepsilon}_n^{\text{FE}} + h\,\mathbf{f}\left(t_n, \mathbf{y}_n\right) - h\,\mathbf{f}\left(t_n, \mathbf{y}(t_n)\right) + \boldsymbol{\delta}_{n+1}^{\text{FE}} \\
&\approx \boldsymbol{\varepsilon}_n^{\text{FE}} + h\,\mathbf{f}_{\mathbf{y}}\left(t_n, \mathbf{y}(t_n)\right)\cdot\boldsymbol{\varepsilon}_n^{\text{FE}} + \boldsymbol{\delta}_{n+1}^{\text{FE}} \\
&= \left(\mathbf{I} - h\,\mathbf{f}_{\mathbf{y}}\left(t_n, \mathbf{y}(t_n)\right)\right)\cdot\boldsymbol{\varepsilon}_n^{\text{FE}} + \boldsymbol{\delta}_{n+1}^{\text{FE}}.
\end{aligned}$$

The current global error consists of the previous global error transported to the current time, $(\mathbf{I} - h\,\mathbf{f}_{\mathbf{y}})\cdot\boldsymbol{\varepsilon}_n$, plus the local error added in the current step $\boldsymbol{\delta}_{n+1}$. Assuming that the Jacobian matrix $\mathbf{f}_{\mathbf{y}}$ is diagonalizable with eigenvalues λ_i and independent eigenvectors $\mathbf{v}_i$, $i = 1, \ldots, d$, the local and global errors can be decomposed in components along each eigenvector, $\boldsymbol{\delta}_n = \sum_{i=1}^{d} (\boldsymbol{\delta}_n)_i\,\mathbf{v}_i$ and $\boldsymbol{\varepsilon}_n = \sum_{i=1}^{d} (\boldsymbol{\varepsilon}_n)_i\,\mathbf{v}_i$. The error evolution equation can be written in component-wise manner

$$\left(\boldsymbol{\varepsilon}_{n+1}^{\text{FE}}\right)_i = (1 - h\,\lambda_i)\cdot\left(\boldsymbol{\varepsilon}_n^{\text{FE}}\right)_i + \left(\boldsymbol{\delta}_{n+1}^{\text{FE}}\right)_i, \quad i = 1, \ldots, d.$$

The forward Euler scheme is stable if it does not amplify previous errors, i.e., if the step size h is chosen such that $|1 + h\,\lambda_i| \leq 1$ for all Jacobian eigenvalues λ_i.

These considerations lead to the following simple analysis of the linear stability of the Euler schemes. Consider the scalar linear test problem

$$\dot{\mathbf{y}}(t) = \lambda\, \mathbf{y}(t), \qquad \mathbf{y}(t_0) = \mathbf{y}_0, \tag{4.12}$$

where λ plays the role of a Jacobian eigenvalue and $\mathbf{y}$ plays the role of a component of the global error. Application of (4.2) leads to

$$\mathbf{y}_1 = \mathbf{y}_0 + h\,\lambda\,\mathbf{y}_0 \quad \Rightarrow \quad \mathbf{y}_1 = R^{\text{FE}}(z)\,\mathbf{y}_0, \quad z = h\lambda, \quad R^{\text{FE}}(z) = 1 + z. \tag{4.13}$$

The stability domain of the method is the following set $\mathbb{S} \subset \mathbb{C}$

$$\mathbb{S} := \{z \in \mathbb{C} \ : \ |R(z)| \leq 1\}\,.$$

The forward Euler method is linearly stable if the step size is chosen such that $h\lambda_i \in \mathbb{S}^{\text{FE}}$ for all Jacobian eigenvalues λ_i.

Similarly, application of the backward Euler scheme (4.3) to the test problem (4.12) leads to

$$\mathbf{y}_1 = \mathbf{y}_0 + h\,\lambda\,\mathbf{y}_1 \quad \Rightarrow \quad \mathbf{y}_1 = R^{\text{BE}}(z)\,\mathbf{y}_0, \quad z = h\lambda, \quad R^{\text{BE}}(z) = \frac{1}{1-z}. \tag{4.14}$$

We note that if $Re(\lambda) \leq 0$ then $|R(z) \leq 1|$ for any step size h. The backward Euler method is A-stable since $\mathbb{C}^- \subset \mathbb{S}^{\text{BE}}$.

4.3 Multirate explicit Euler method

4.3.1 Multiscale partitioned initial value problems

Consider the partitioned initial-value problem

$$\begin{bmatrix} \dot{\mathbf{y}}_{\text{S}}(t) \\ \dot{\mathbf{y}}_{\text{F}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\text{S}}\big(t, \mathbf{y}_{\text{S}}(t), \mathbf{y}_{\text{F}}(t)\big) \\ \mathbf{f}_{\text{F}}\big(t, \mathbf{y}_{\text{S}}(t), \mathbf{y}_{\text{F}}(t)\big) \end{bmatrix}, \quad t \in [t_0, t_f], \quad \begin{bmatrix} \mathbf{y}_{\text{S}}(t_0) \\ \mathbf{y}_{\text{F}}(t_0) \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{\text{S},0} \\ \mathbf{y}_{\text{F},0} \end{bmatrix}, \tag{4.15}$$

with $\mathbf{y}_{\text{S}} \in \mathbb{R}^{d_{\text{S}}}$ are the slow varying components of the solution, $\mathbf{y}_{\text{F}} \in \mathbb{R}^{d_{\text{F}}}$ are the fast varying components, and $\mathbf{f}_{\text{S}}$ and $\mathbf{f}_{\text{F}}$ are the slow and fast change rates, respectively.

Since the rates of change are different we want to solve the slow components with a large step size H and the fast components with a small step size $h = H/\mathtt{m}$. This idea is illustrated in Figure 11 for $\mathtt{m} = 3$.

In this book we will consider the approximation times t_n, t_{n+1}, t_{n+2} are given by the large step size H. The intermediate approximation points corresponding to the small step size h are considered fractions of the full steps. Thus we will use the notation:

$$t_{n+1} = t_n + H \quad \text{and} \quad t_{n+\ell/\mathtt{m}} = t_n + \ell\, h, \quad \ell = 0, \ldots, \mathtt{m}. \tag{4.16}$$

Application of the forward Euler method (4.2) with a large step size H to solve the slow components gives:

$$\mathbf{y}_{\mathrm{S},n+1} = \mathbf{y}_{\mathrm{S},n} + H\, \mathbf{f}_{\mathrm{S}}\big(t_n, \mathbf{y}_{\mathrm{S},n}, \mathbf{y}_{\mathrm{F},n}\big). \tag{4.17a}$$

Application of the forward Euler method (4.2) with a small step size $h = H/\mathtt{m}$ to solve the fast components gives:

$$\begin{aligned} \mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}} \;&=\; \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + h\, \mathbf{f}_{\mathrm{F}}\big(t_{n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}}\big) \\ & \qquad t_{n+\ell/\mathtt{m}} = t_n + \ell\, h, \quad \ell = 0, \ldots, \mathtt{m}-1. \end{aligned} \tag{4.17b}$$

Note that the intermediate slow variables $\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}}$ for $\ell = 1, \ldots, \mathtt{m}-1$ have not been calculated by (4.17a) and need to be approximated. The way this approximation is carried out defines the particular multirate strategy one employs.

4.3.2 Multiscale split initial value problems

Consider the additively split initial-value problem

$$\dot{\mathbf{y}} = \mathbf{f}_{\mathrm{S}}\big(t, \mathbf{y}(t)\big) + \mathbf{f}_{\mathrm{F}}\big(t, \mathbf{y}(t)\big), \quad t \in [t_0, t_f], \quad \mathbf{y}(t_0) = \mathbf{y}_0. \tag{4.18}$$

The dynamics is driven by the simultaneous action of slow processes described by $\mathbf{f}_{\mathrm{S}}$ and of fast processes described by $\mathbf{f}_{\mathrm{F}}$. Following the idea illustrated in Figure 11 we want to discretize the slow rates of change with a

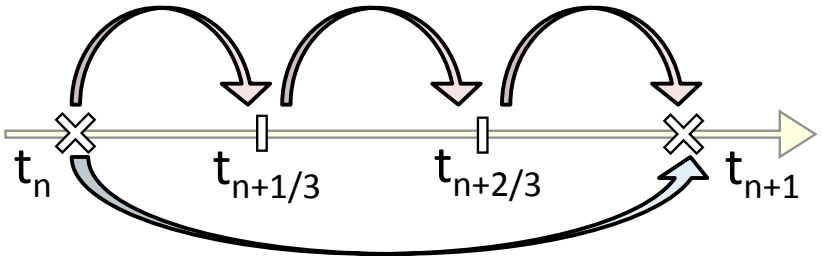


Figure 11: Application of multi rate time integration. The slow components are integrated with a large time step $H = t_{n+1} - t_n$. The fast components are integrated with a small time step $h = H/3$.

large step size H and the fast rates of change with a small step size $h = H/\mathtt{m}$ for $\mathtt{m} = 3$.

The first possibility is to apply operator splitting and incorporate the slow and the fast terms in different steps. In a slowest–first approach the forward Euler method (4.2) with a large step size H is applied to the slow rate:

$$\mathbf{y}^*_{n+1} = \mathbf{y}_n + H\,\mathbf{f}_{\mathrm{S}}\big(t_n, \mathbf{y}_n\big). \tag{4.19a}$$

Application of the forward Euler method (4.2) with a small step size $h = H/\mathtt{m}$ to solve the fast components gives:

$$\begin{aligned} \widetilde{\mathbf{y}}_n &= \mathbf{y}^*_{n+1}, \\ \widetilde{\mathbf{y}}_{n+(\ell+1)/\mathtt{m}} &= \widetilde{\mathbf{y}}_{n+\ell/\mathtt{m}} + h\,\mathbf{f}_{\mathrm{F}}\big(t_{n+\ell/\mathtt{m}}, \widetilde{\mathbf{y}}_{n+\ell/\mathtt{m}}\big), \quad \ell = 0, \ldots, \mathtt{m}-1, \\ \mathbf{y}_{n+1} &= \widetilde{\mathbf{y}}_{n+1}. \end{aligned} \tag{4.19b}$$

A second possibility is to compute the slow rate $\mathbf{f}_{\mathrm{S}}\big(t_n, \mathbf{y}_n\big)$ once and to incorporate it in each fast step as follows:

$$\begin{aligned} \mathbf{y}_{n+(\ell+1)/\mathtt{m}} &= \mathbf{y}_{n+\ell/\mathtt{m}} + h\,\mathbf{f}_{\mathrm{S}}\big(t_n, \mathbf{y}_n\big) + h\,\mathbf{f}_{\mathrm{F}}\big(t_{n+\ell/\mathtt{m}}, \mathbf{y}_{n+\ell/\mathtt{m}}\big), \\ &\qquad \ell = 0, \ldots, \mathtt{m}-1. \end{aligned} \tag{4.20}$$

The method (4.20) is multirate since it employs one evaluation of $\mathbf{f}_{\mathrm{S}}$ and $\mathtt{m}$ evaluations of $\mathbf{f}_{\mathrm{F}}$ per step.

Note that the partitioned system (4.15) and the split system (4.18) are formulations that can be transformed into one another. For example (4.15)

can be written in the form (4.18) as follows:

$$\mathbf{y}(t) = \begin{bmatrix} \dot{\mathbf{y}}_{\mathrm{S}}(t) \\ \dot{\mathbf{y}}_{\mathrm{F}}(t) \end{bmatrix}, \quad \dot{\mathbf{y}}(t) = \begin{bmatrix} \mathbf{f}_{\mathrm{S}}\big(t, \mathbf{y}(t)\big) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{f}_{\mathrm{F}}\big(t, \mathbf{y}(t)\big) \end{bmatrix}, \quad t \in [t_0, t_f]. \tag{4.21}$$

Similarly, (4.18) can be written in the form (4.15) as follows:

$$\dot{\mathbf{y}}_{\mathrm{S}} = \mathbf{f}_{\mathrm{S}}\big(t, \mathbf{y}_{\mathrm{S}} + \mathbf{y}_{\mathrm{F}}\big), \quad \dot{\mathbf{y}}_{\mathrm{F}} = \mathbf{f}_{\mathrm{F}}\big(t, \mathbf{y}_{\mathrm{S}} + \mathbf{y}_{\mathrm{F}}\big), \quad \mathbf{y}(t) := \mathbf{y}_{\mathrm{S}}(t) + \mathbf{y}_{\mathrm{F}}(t), \quad t \in [t_0, t_f]. \tag{4.22}$$

We note that when a split system is rewritten in partitioned form (4.22) the full solution is the sum of individual solutions. This leads to some ambiguity about the choice of initial conditions for the two individual subsystems: any choice of $\mathbf{y}_{\mathrm{S}}(t_0)$ and $\mathbf{y}_{\mathrm{F}}(t_0)$ such that $\mathbf{y}(t_0) = \mathbf{y}_{\mathrm{S}}(t_0) + \mathbf{y}_{\mathrm{F}}(t_0)$ leads to legitimate partitioned solutions (4.22). On the other hand the reformulation of a split system in the partitioned form (4.21) is completely well defined.

We will use the partitioned system (4.15) as our default formulation in this book. Since the two formulations are equivalent it suffices to perform the accuracy and stability analyses for one form.

4.3.3 Slowest–first solution strategy

In the *slowest–first strategy* one starts with solving (4.17a) to obtain the slow solution $\mathbf{y}_{\mathrm{S},n+1}$. Next, intermediate approximation of the slow solution are obtained by either zeroth order interpolation

$$\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}} = \mathbf{y}_{\mathrm{S},n}, \quad \ell = 0, 1, \ldots, \mathtt{m} - 1, \tag{4.23}$$

or by first order interpolation

$$\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}} = \frac{\mathtt{m} - \ell}{\mathtt{m}} \mathbf{y}_{\mathrm{S},n} + \frac{\ell}{\mathtt{m}} \mathbf{y}_{\mathrm{S},n+1}, \quad \ell = 0, 1, \ldots, \mathtt{m} - 1. \tag{4.24}$$

These approximations are used in (4.17b) and one solves next for the fast components.

4.3.4 Fastest–first solution strategy

In the *fastest–first strategy* one starts with solving (4.17b) to advance the fast components to $\mathbf{y}_{\mathrm{F},n+1}$. The intermediate slow solutions can be obtained by zeroth oder interpolation (4.23). Note that (4.24) cannot be used since $\mathbf{y}_{\mathrm{S},n+1}$ is unavailable. To apply first order interpolation we can proceed along different ways:

- We use the previous value $\mathbf{y}_{\mathrm{S},n-1}$ and define $\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}}$ as the the evaluation of the linear polynomial given by $(t_{n-1}, \mathbf{y}_{\mathrm{S},n-1})$ and $(t_n, \mathbf{y}_{\mathrm{S},n})$. This turns the one-step scheme into a two-step scheme whose stability analysis is more difficult.
- We use a Hermite approach and use the differential equation itself to obtain a first order interpolation. This is equivalent to performing an explicit Euler step:

$$\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}} = \mathbf{y}_{\mathrm{S},n} + \frac{\ell}{\mathtt{m}} H\, \mathbf{f}_{\mathrm{S}}(t_n, \mathbf{y}_{\mathrm{S},n}, \mathbf{y}_{\mathrm{F},n}), \quad \ell = 0, 1, \ldots, \mathtt{m}-1. \tag{4.25}$$

Note that with zeroth order interpolation the fastest–first and slowest–first multirate forward Euler methods coincide.

4.3.5 Accuracy analysis of multirate explicit Euler

We assume that the functions $\mathbf{f}_{\mathrm{S}}$, $\mathbf{f}_{\mathrm{F}}$ are smooth and that all their first and second order partial derivatives are uniformly bounded. The error analysis is similar to the one leading to equation (4.6): we compare Taylor series (about t_n) of the exact and the numerical solutions. For the exact solutions we have:

$$\begin{aligned} \mathbf{y}_{\mathrm{S}}(t_{n+1}) &= \mathbf{y}_{\mathrm{S}}(t_n) + \dot{\mathbf{y}}_{\mathrm{S}}(t_n) + \mathcal{O}(H^2) \\ &= \mathbf{y}_{\mathrm{S}}(t_n) + H\, \mathbf{f}_{\mathrm{S}}\big(t_n, \mathbf{y}_{\mathrm{S}}(t_n), \mathbf{y}_{\mathrm{F}}(t_n)\big) + \mathcal{O}(H^2), \end{aligned} \tag{4.26a}$$

$$\begin{aligned} \mathbf{y}_{\mathrm{F}}(t_{n+1}) &= \mathbf{y}_{\mathrm{F}}(t_n) + \dot{\mathbf{y}}_{\mathrm{F}}(t_n) + \mathcal{O}(H^2) \\ &= \mathbf{y}_{\mathrm{F}}(t_n) + H\, \mathbf{f}_{\mathrm{F}}\big(t_n, \mathbf{y}_{\mathrm{S}}(t_n), \mathbf{y}_{\mathrm{F}}(t_n)\big) + \mathcal{O}(H^2). \end{aligned} \tag{4.26b}$$

Slowest–first approach In the slowest–first strategy (Section 4.3.3) one first applies one large Euler step to the slow system (4.17a). To study the accuracy we start this step from the exact solution at time t_n. A comparison with (4.26a)

$$\mathbf{y}_{\mathrm{S},n+1} \;=\; \mathbf{y}_{\mathrm{S}}(t_n) + H\,\mathbf{f}_{\mathrm{S}}\big(t_n, \mathbf{y}_{\mathrm{S}}(t_n), \mathbf{y}_{\mathrm{F}}(t_n)\big) = \mathbf{y}_{\mathrm{S}}(t_{n+1}) + \boldsymbol{\delta}_{\mathrm{S},n+1}.$$

From (4.5) and (4.6) we see that the local truncation error is $\mathcal{O}(H^2)$, therefore the slow component is integrated with first order accuracy:

$$\|\boldsymbol{\delta}_{\mathrm{S},n+1}\| \;\le\; \frac{H^2}{2} \max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\|.$$

Since $\mathbf{y}_{\mathrm{S}}$ is slowly evolving, its time derivatives are small, and a desired accuracy level `tol` for slow components can be achieved for relatively large values of the macrostep H:

$$H = \sqrt{\frac{2\,\mathtt{tol}}{\max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\|}} \quad\Rightarrow\quad \|\boldsymbol{\delta}_{\mathrm{S},n+1}\| \le \mathtt{tol}. \tag{4.27}$$

Next, `m` small Euler steps are applied to the fast system (4.17b), starting from the exact solution at time t_n. Each small step is corrupted by the forward Euler local truncation error $\boldsymbol{\delta}_{\mathrm{F},n+\ell/\mathtt{m}} \sim \mathcal{O}(h^2)$:

$$\mathbf{y}_{\mathrm{F}}(t_{n+(\ell+1)/m}) \;=\; \mathbf{y}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}}) + h\,\mathbf{f}_{\mathrm{F}}\big(t_{n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{S}}(t_{n+\ell/\mathtt{m}}), \mathbf{y}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}})\big) - \boldsymbol{\delta}_{\mathrm{F},n+\ell/\mathtt{m}}.$$

The succession of numerical steps will accumulate these errors. The accumulated error in the fast solution is denoted by

$$\Delta\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} = \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} - \mathbf{y}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}}),$$

where the initial error is zero since we start from the exact solution, and the final error is the local truncation error over `m` steps of the fast computational solution:

$$\Delta\mathbf{y}_{\mathrm{F},n} = 0 \quad\text{and}\quad \boldsymbol{\delta}_{\mathrm{F},n+1} := \Delta\mathbf{y}_{\mathrm{F},n+1}.$$

The errors in the fast solution accumulate as follows:

$$\begin{aligned}
\mathbf{y}_{\mathrm{F},n+(\ell+1)/m} &= \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + h\,\mathbf{f}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}}),\\
\Delta\mathbf{y}_{\mathrm{F},n+(\ell+1)/m} &= \Delta\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + h\,\mathbf{f}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}})\\
&\quad - h\,\mathbf{f}_{\mathrm{F}}\big(t_{n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{S}}(t_{n+\ell/\mathtt{m}}), \mathbf{y}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}})\big) + \boldsymbol{\delta}_{\mathrm{F},n+\ell/\mathtt{m}}\\
&= \int_0^1 \left(\mathbf{I} + h\,\frac{\partial \mathbf{f}_{\mathrm{F}}}{\partial \mathbf{y}_{\mathrm{F}}}\right)\Big(t_{n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{S}}(t_{n+\ell/\mathtt{m}}) + \sigma\,\Delta\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}}, \dots\\
&\qquad \mathbf{y}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}}) + \sigma\,\Delta\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}}\Big)\cdot\Delta\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}}\,d\sigma\\
&\quad + \int_0^1 \frac{\partial \mathbf{f}_{\mathrm{F}}}{\partial \mathbf{y}_{\mathrm{S}}}\Big(t_{n+\ell/\mathtt{m}}, \mathbf{y}_{\mathrm{S}}(t_{n+\ell/\mathtt{m}}) + \sigma\,\Delta\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}}, \dots\\
&\qquad \mathbf{y}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}}) + \sigma\,\Delta\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}}\Big)\cdot\Delta\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}}\,d\sigma + \boldsymbol{\delta}_{\mathrm{F},n+\ell/\mathtt{m}}.
\end{aligned} \tag{4.28}$$

The last equality above was obtained by applying the mean value theorem. We denote the maximum norms of the Jacobians by:

$$\begin{aligned}
L_{\mathrm{F,S}} &= \max_{\tau\in[t_n,t_{n+1}],\,\|\mathbf{y}_{\mathrm{S}}-\mathbf{y}_{\mathrm{S}}(\tau)\|\le\epsilon,\,\|\mathbf{y}_{\mathrm{F}}-\mathbf{y}_{\mathrm{F}}(\tau)\|\le\epsilon} \left\|\frac{\partial \mathbf{f}_{\mathrm{F}}}{\partial \mathbf{y}_{\mathrm{S}}}(\tau,\mathbf{y}_{\mathrm{S}},\mathbf{y}_{\mathrm{F}})\right\|,\\
L_{\mathrm{F,F}} &= \max_{\tau\in[t_n,t_{n+1}],\,\|\mathbf{y}_{\mathrm{S}}-\mathbf{y}_{\mathrm{S}}(\tau)\|\le\epsilon,\,\|\mathbf{y}_{\mathrm{F}}-\mathbf{y}_{\mathrm{F}}(\tau)\|\le\epsilon} \left\|\frac{\partial \mathbf{f}_{\mathrm{F}}}{\partial \mathbf{y}_{\mathrm{F}}}(\tau,\mathbf{y}_{\mathrm{S}},\mathbf{y}_{\mathrm{F}})\right\|,\\
L_{\mathrm{F},t} &= \max_{\tau\in[t_n,t_{n+1}],\,\|\mathbf{y}_{\mathrm{S}}-\mathbf{y}_{\mathrm{S}}(\tau)\|\le\epsilon,\,\|\mathbf{y}_{\mathrm{F}}-\mathbf{y}_{\mathrm{F}}(\tau)\|\le\epsilon} \left\|\frac{\partial \mathbf{f}_{\mathrm{F}}}{\partial t}(\tau,\mathbf{y}_{\mathrm{S}},\mathbf{y}_{\mathrm{F}})\right\|,
\end{aligned} \tag{4.29}$$

where the maxima are taken over all values of $\mathbf{y}_{\mathrm{S}}, \mathbf{y}_{\mathrm{F}}$ in a neighborhood of the exact solution. The corresponding norm bounds $L_{\mathrm{S,S}}, L_{\mathrm{S,F}}, L_{\mathrm{S},t}$ for the Jacobians of the slow component are defined similarly.

Taking norms in (4.28) gives:

$$\begin{aligned}\|\Delta\mathbf{y}_{\mathrm{F},n+(\ell+1)/m}\| \le (1+h\,L_{\mathrm{F,F}})\,\|\Delta\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}}\| + h\,L_{\mathrm{F,S}}\,\|\Delta\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}}\| \\ + \|\boldsymbol{\delta}_{\mathrm{F},n+\ell/\mathtt{m}}\|.\end{aligned} \tag{4.30}$$

Iterating for $\ell = 0,\ldots,\mathtt{m}-1$ and using $\Delta\mathbf{y}_{\mathrm{F},n} = 0$ leads to the estimate:

$$\begin{aligned}\|\Delta\mathbf{y}_{\mathrm{F},n+1}\| &\le \sum_{k=0}^{\mathtt{m}-1}(1+h\,L_{\mathrm{F,F}})^k\cdot\left(h\,L_{\mathrm{F,S}}\,\|\Delta\mathbf{y}_{\mathrm{S},n+(\mathtt{m}-k-1)/\mathtt{m}}\| + \|\boldsymbol{\delta}_{\mathrm{F},n+(\mathtt{m}-k-1)/\mathtt{m}}\|\right)\\ &\le e^{H\,L_{\mathrm{F,F}}}\,L_{\mathrm{F,S}}\,h\sum_{k=0}^{\mathtt{m}-1}\|\Delta\mathbf{y}_{\mathrm{S},n+(\mathtt{m}-k-1)/\mathtt{m}}\| \\ &\quad + e^{H\,L_{\mathrm{F,F}}}\sum_{k=0}^{\mathtt{m}-1}\|\boldsymbol{\delta}_{\mathrm{F},n+(\mathtt{m}-k-1)/\mathtt{m}}\|.\end{aligned} \tag{4.31}$$

This error (bound) has two components:

- One is the cumulated local truncation errors (LTE) of the sequence of $\mathtt{m}$ small steps:

$$e^{H\,L_{\mathrm{F,F}}}\sum_{k=0}^{\mathtt{m}-1}\|\boldsymbol{\delta}_{\mathrm{F},n+(\mathtt{m}-k-1)/\mathtt{m}}\| \;=\; e^{H\,L_{\mathrm{F,F}}}\,\frac{\mathtt{m}\,h^2}{2}\max_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|.$$

- The other is the effect of the error in the slow solution onto the fast numerical solution. This term depends on the type of slow interpolation performed, and is discussed below.

Constant interpolation of the slow component. In case of constant interpolation of the slow solution (4.23):

$$\begin{aligned}\left\|\Delta\mathbf{y}_{\mathrm{S},n+k/\mathtt{m}}\right\| &\le \frac{k}{\mathtt{m}}\,H\max_{t\in[t_n,t_{n+k/\mathtt{m}}]}\|\dot{\mathbf{y}}_{\mathrm{S}}(t)\|,\\ \sum_{k=0}^{\mathtt{m}-1}\left\|\Delta\mathbf{y}_{\mathrm{S},n+k/\mathtt{m}}\right\| &\le \frac{\mathtt{m}-1}{2}\,H\max_{\tau\in[t_n,t_{n+1}]}\|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\|,\\ h\sum_{k=0}^{\mathtt{m}-1}\left\|\Delta\mathbf{y}_{\mathrm{S},n+k/\mathtt{m}}\right\| &\le \frac{H^2}{2}\max_{\tau\in[t_n,t_{n+1}]}\|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\|.\end{aligned}$$

Therefore (4.31) becomes:

$$\|\Delta \mathbf{y}_{\mathrm{F},n+1}\| \leq e^{H\,L_{\mathrm{F,F}}}\,\frac{H^2}{2}\,\left(L_{\mathrm{F,S}}\,\max_{\tau\in[t_n,t_{n+1}]}\|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + \frac{1}{\mathtt{m}}\,\max_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|\right). \tag{4.32}$$

Both error terms, the LTE of the fast steps and the interpolation of the slow solution, have the same asymptotic order. The overall error in the fast component is first order accurate.

Even when $\mathtt{m}\to\infty$ the fast component is affected by an error that depends on the large step size H. To keep the fast local error below the desired accuracy level $\|\Delta \mathbf{y}_{\mathrm{F},n+1}\| \leq \mathtt{tol}$, it is not sufficient to choose (4.27) for the accuracy of the slow component, and then increase the number of small time steps for attaining the desired accuracy of the fast component.

One possible strategy is to balance the two error components in (4.32), i.e., make each of them smaller that $\mathtt{tol}/2$. First choose the large step size H as the minimum that ensures the accuracy of the slow component (4.27), as well as keeps the interpolation error component in (4.32) below $\mathtt{tol}/2$:

$$e^{H\,L_{\mathrm{F,F}}}\,H^2 \leq \frac{\mathtt{tol}}{L_{\mathrm{F,S}}\,\max_{\tau\in[t_n,t_{n+1}]}\|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\|}.$$

Next, choose $\mathtt{m}$ such that the fast local truncation error component in (4.32) is smaller than $\mathtt{tol}/2$:

$$\mathtt{m} \geq \frac{e^{H\,L_{\mathrm{F,F}}}\,H^2}{\mathtt{tol}}\,\max_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| \geq 2\,e^{H\,L_{\mathrm{F,F}}}\,\frac{\max_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|}{\max_{t\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{S}}(t)\|}.$$

The second relation comes from using (4.27) and shows that the ratio of the step sizes is driven by the ratio of the dynamics of the fast to slow components.

Linear interpolation of the slow component. In case of linear interpolation of the slow solution (4.24)

$$\left\|\Delta \mathbf{y}_{\mathrm{S},n+k/\mathtt{m}}\right\| \leq \frac{H^2}{8}\,\max_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + \frac{k}{\mathtt{m}}\,\boldsymbol{\delta}_{\mathrm{S},n+1},$$

where the first term is the maximal linear interpolation error, and the second comes from the fact that we use the numerical solution value $\mathbf{y}_{\mathrm{S},n+1}$, not the exact value $\mathbf{y}_{\mathrm{S}}(t_{n+1})$, as data for interpolation. We then have:

$$\begin{aligned} h \sum_{k=0}^{\mathtt{m}-1} \left\| \Delta \mathbf{y}_{\mathrm{S},n+k/\mathtt{m}} \right\| &\leq \frac{\mathtt{m}\, h\, H^2}{8} \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + \frac{\mathtt{m}-1}{2\,\mathtt{m}}\, H\, \boldsymbol{\delta}_{\mathrm{S},n+1} \\ &\leq \frac{H^3}{8} \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + \frac{H^3}{4} \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| \\ &= \frac{3H^3}{8} \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\|. \end{aligned}$$

We also see that for linear interpolation the LTE of the fast steps dominates, asymptotically, the error due to the interpolation of the slow components. Because of this we will only consider the LTE error in what follows. Since the fast steps are solved with forward Euler, the overall fast solution is first order accurate.

We seek to ensure that the accuracy of the fast solution is smaller than the acceptable level $\|\Delta \mathbf{y}_{\mathrm{F},n+1}\| \leq \mathtt{tol}$:

$$\|\Delta \mathbf{y}_{\mathrm{F},n+1}\| \;\leq\; e^{H\, L_{\mathrm{F,F}}}\, \frac{\mathtt{m}\, h^2}{2} \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| = e^{H\, L_{\mathrm{F,F}}}\, \frac{H^2}{2\,\mathtt{m}} \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| \leq \mathtt{tol}.$$

This leads to the following sufficient condition:

$$\mathtt{m} \;\geq\; e^{H\, L_{\mathrm{F,F}}}\, \frac{H^2 \, \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|}{2\,\mathtt{tol}}.$$

Using the relation (4.27) for the step H we find that, as expected, the ratio of the step sizes is driven by the ratio of the dynamics of the fast to slow components:

$$\mathtt{m} \geq e^{H\, L_{\mathrm{F,F}}}\, \frac{\max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|}{\max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\|}.$$

Comment 4.1 (Sharpness of the error estimates) *The estimates of H and $\mathtt{m}$ derived in this section are not sharp. They are meant to be only qualitative, and we will not use them for implementation in an adaptive code.*

Fastest–first approach As in the slowest–first strategy, the numerical approximation of the slow component is independent of the approximations of the fast components in the fastest–first strategy (Section 4.3.4). Hence one gets the same results for the slow components in the accuracy analysis of the slowest-first approach.

Recapitulating the accuracy analysis for the fast components, one ends up with the estimate (4.31), where $\|\Delta \mathbf{y}_{\mathrm{S},n+(m-k-1)/m}\|$ denotes again the interpolation errors in the slow components. However, as we start with computing the fast variables, interpolation can only be based on information at time point t_n. For constant interpolation, we again obtain the estimate (4.32), as slowest and fastest-first approach coincide in this case. For Hermite interpolation (4.25) the error is:

$$\begin{aligned} \|\Delta \mathbf{y}_{\mathrm{S},n+k/\mathtt{m}}\| &= \left\| \left(\mathbf{y}_{\mathrm{S}}(t_n) + \frac{k}{\mathtt{m}} H \mathbf{f}_{\mathrm{S}}(t_n, \mathbf{y}_{\mathrm{S},n}, \mathbf{y}_{\mathrm{F},n}) \right) - \mathbf{y}_{\mathrm{S}}(t_n + h\,k/\mathtt{m}) \right\| \\ &\leq \frac{1}{2} \left(\frac{k}{\mathtt{m}} \right)^2 H^2 \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| \,, \end{aligned}$$

which turns the estimate (4.32) into

$$\|\Delta \mathbf{y}_{\mathrm{F},n+1}\| \leq e^{H\,L_{\mathrm{F,F}}} \frac{H^2}{2} \left(\frac{2}{3} h\, L_{\mathrm{F,S}} \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + \frac{1}{\mathtt{m}} \max_{\tau \in [t_n, t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| \right).$$

Again, the LTE of the fast steps dominates, asymptotically, the error due to the interpolation of the slow components.

4.3.6 Linear stability analysis of multirate explicit Euler

A two-dimensional test problem

The test problem (4.12) needs to be generalized to account for two time scale problems in order to investigate the stability of multirate schemes. Following the analysis done by Kværnø [**?**], we consider the following generalized

linear test problem:

$$\begin{bmatrix} \dot{\mathbf{y}}_{\mathrm{S}}(t) \\ \dot{\mathbf{y}}_{\mathrm{F}}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \lambda_{\mathrm{S}} & \eta_{\mathrm{F}} \\ \eta_{\mathrm{S}} & \lambda_{\mathrm{F}} \end{bmatrix}}_{A} \begin{bmatrix} \mathbf{y}_{\mathrm{S}}(t) \\ \mathbf{y}_{\mathrm{F}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathrm{S}}\big(t, \mathbf{y}_{\mathrm{S}}(t), \mathbf{y}_{\mathrm{F}}(t)\big) \\ \mathbf{f}_{\mathrm{F}}\big(t, \mathbf{y}_{\mathrm{S}}(t), \mathbf{y}_{\mathrm{F}}(t)\big) \end{bmatrix} . \tag{4.33}$$

The test system (4.33) is assumed to have the following properties:

1. The system has real coefficients,

$$\lambda_{\mathrm{S}},\, \lambda_{\mathrm{F}},\, \eta_{\mathrm{S}},\, \eta_{\mathrm{F}} \,\in\, \mathbb{R}. \tag{4.34a}$$

2. The system has two underlying time scales associated with the slow (λ_{S}) and the fast (λ_{F}) dynamics. Each subsystem is stable when run in decoupled mode, therefore the two diagonal terms are negative

$$\lambda_{\mathrm{S}} < 0, \qquad \lambda_{\mathrm{F}} < 0. \tag{4.34b}$$

3. The dynamics is characterized by the following coefficients:

$$\text{scale ratio:} \qquad \mu = \frac{|\lambda_{\mathrm{F}}|}{|\lambda_{\mathrm{S}}|}, \tag{4.34c}$$

$$\text{coupling coefficient:} \qquad \mathtt{k} = \frac{\eta_{\mathrm{F}}\, \eta_{\mathrm{S}}}{\lambda_{\mathrm{F}}\, \lambda_{\mathrm{S}}}. \tag{4.34d}$$

   To sample the fast part accurately enough one may require that $\mathtt{m} \geq \mu$. The system is weakly coupled for $|\mathtt{k}| \approx 0$.

4. The coupled system (4.33) is assumed to be stable. The coupling between these two components is represented by η_{F} and η_{S}. The system (4.33) is stable if the real part of the eigenvalues of A is negative. The characteristic polynomial of this matrix is:

$$p(z) = (z - \lambda_{\mathrm{S}})(z - \lambda_{\mathrm{F}}) - \eta_{\mathrm{S}}\, \eta_{\mathrm{F}} = z^2 - \mathrm{tr}(A) z + \det(A).$$

   According to the continuous Routh-Hurwitz criterion (Lemma 4.1) all the roots are in the left half plane iff all the coefficients of the characteristic polynomial are positive:

$$\mathrm{tr}(A) < 0 \quad \Leftrightarrow \quad \lambda_{\mathrm{S}} + \lambda_{\mathrm{F}} < 0, \tag{4.34ea}$$

$$\det(A) > 0 \quad \Leftrightarrow \quad \lambda_{\mathrm{S}}\, \lambda_{\mathrm{F}} > \eta_{\mathrm{S}}\, \eta_{\mathrm{F}} \quad \Leftrightarrow \quad \mathtt{k} < 1. \tag{4.34eb}$$

The trace relation (4.34ea) is fulfilled since the separate dynamics of the two subsystems are stable (4.34b). The determinant relation (4.34eb) can be expressed in terms of the coupling coefficient (4.34d); note that the linear test system is stable for $\mathtt{k} \to -\infty$.

Comment 4.2 (Limitations of the linear problem (4.33)**)** *The fact that the test system has real coefficients* (4.34a) *implies some limitations of the dynamics it models. For decoupled systems with* $\eta_{\rm S} = \eta_{\rm F} = 0$ *only damping dynamics is allowed, since* $\lambda_{\rm S}$ *and* $\lambda_{\rm F}$ *have no imaginary parts.*

Comment 4.3 (Enforcing that the linear problem (4.33) **has two scales)** *In order to guarantee that the test problem* (4.33) *has two distinct scales like the original system* (4.15) *one can ask that each coupling coefficient is not much larger than the corresponding internal dynamics coefficients:*

$$|\eta_{\rm F}| \le C_1\,|\lambda_{\rm S}|, \quad \text{and} \quad |\eta_{\rm S}| \le C_1\,| \quad \Rightarrow \quad -C_1\,C_2 \le \mathtt{k} < C_1\,C_2, \tag{4.1}$$

for a moderate constants $C_1, C_2 \sim \mathcal{O}(1)$*. Under this assumption the coupling coefficient for a stable system* (4.33) *is bounded below,* $-C_1\,C_2 \le \mathtt{k} < 1$*.*

Lemma 4.1 (Routh-Hurwitz continuous stability criterion) *The matrix* $A \in \mathbb{R}^{2\times 2}$ *has all eigenvalues in the negative complex plane if and only if the following conditions hold:*

$$a) \qquad \operatorname{tr}(A) < 0, \quad \text{and} \tag{4.2a}$$

$$b) \qquad \det(A) > 0. \tag{4.2b}$$

Multirate forward Euler with constant interpolation

Application of the multirate forward Euler scheme (4.17) with zeroth order interpolation (4.23) gives:

$$\begin{aligned} \mathbf{y}_{{\rm S},n+1} &= \mathbf{y}_{{\rm S},n} + H\,\mathbf{f}_{\rm S}\big(t_n, \mathbf{y}_{{\rm S},n}, \mathbf{y}_{{\rm F},n}\big) \\ &= \mathbf{y}_{{\rm S},n} + H\,\lambda_{\rm S}\,\mathbf{y}_{{\rm S},n} + H\,\eta_{\rm F}\,\mathbf{y}_{{\rm F},n} \\ \mathbf{y}_{{\rm F},n+(\ell+1)/\mathtt{m}} &= \mathbf{y}_{{\rm F},n+\ell/\mathtt{m}} + h\,\mathbf{f}_{\rm F}\big(t_n, \mathbf{y}_{{\rm S},n}, \mathbf{y}_{{\rm F},n+\ell/\mathtt{m}}\big) \\ &= \mathbf{y}_{{\rm F},n+\ell/\mathtt{m}} + h\,\eta_{\rm S}\,\mathbf{y}_{{\rm S},n} + h\,\lambda_{\rm F}\,\mathbf{y}_{{\rm F},n+\ell/\mathtt{m}}, \quad \ell = 0,\ldots,\mathtt{m}-1. \end{aligned} \tag{4.3}$$

We introduce the following variables:

$$z_{\mathrm{S}} = H\,\lambda_{\mathrm{S}}, \quad z_{\mathrm{F}} = H\,\lambda_{\mathrm{F}}, \quad w_{\mathrm{F}} = H\,\eta_{\mathrm{F}}, \quad w_{\mathrm{S}} = H\,\eta_{\mathrm{S}}.$$

From the properties of the test problem (4.34b) and (4.34e) we have that:

$$z_{\mathrm{S}} < 0, \qquad z_{\mathrm{F}} < 0, \qquad \text{and} \qquad z_{\mathrm{S}}\,z_{\mathrm{F}} > w_{\mathrm{S}}\,w_{\mathrm{F}}. \tag{4.4}$$

The multirate forward Euler applied to the test problem (4.3) reads:

$$\begin{aligned}
\mathbf{y}_{\mathrm{S},n+1} &= (1 + z_{\mathrm{S}})\,\mathbf{y}_{\mathrm{S},n} + w_{\mathrm{F}}\,\mathbf{y}_{\mathrm{F},n} && \text{(4.5a)}\\
\mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}} &= (1 + z_{\mathrm{F}}/\mathtt{m})\,\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + (w_{\mathrm{S}}/\mathtt{m})\,\mathbf{y}_{\mathrm{S},n}, \quad \ell = 0,\ldots,\mathtt{m}-1 && \text{(4.5b)}
\end{aligned}$$

From equation (4.5b) we have

$$\begin{aligned}
\mathbf{y}_{\mathrm{F},n+1} &= (1 + z_{\mathrm{F}}/\mathtt{m})^{\mathtt{m}}\,\mathbf{y}_{\mathrm{F},n} + \sum_{\ell=0}^{\mathtt{m}-1}(1 + z_{\mathrm{F}}/\mathtt{m})^{\ell}\,(w_{\mathrm{S}}/\mathtt{m})\,\mathbf{y}_{\mathrm{S},n}\\
&= (1 + z_{\mathrm{F}}/\mathtt{m})^{\mathtt{m}}\,\mathbf{y}_{\mathrm{F},n} + \frac{(1 + z_{\mathrm{F}}/\mathtt{m})^{\mathtt{m}} - 1}{z_{\mathrm{F}}/\mathtt{m}}\,(w_{\mathrm{S}}/\mathtt{m})\,\mathbf{y}_{\mathrm{S},n}.
\end{aligned}$$

Therefore

$$\begin{bmatrix}\mathbf{y}_{\mathrm{S},n+1}\\ \mathbf{y}_{\mathrm{F},n+1}\end{bmatrix} = \underbrace{\begin{bmatrix} 1 + z_{\mathrm{S}} & w_{\mathrm{F}}\\ \big((1 + z_{\mathrm{F}}/\mathtt{m})^{\mathtt{m}} - 1\big)(w_{\mathrm{S}}/z_{\mathrm{F}}) & (1 + z_{\mathrm{F}}/\mathtt{m})^{\mathtt{m}}\end{bmatrix}}_{\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}} \cdot \begin{bmatrix}\mathbf{y}_{\mathrm{S},n}\\ \mathbf{y}_{\mathrm{F},n}\end{bmatrix}. \tag{4.6}$$

Comment 4.4 (Linear stability for a decoupled system) *When the forward Euler method is applied to a decoupled test problem* (4.3) *where* $w_{\mathrm{F}} = w_{\mathrm{S}} = 0$ *one takes one step with the slow system and* $\mathtt{m}$ *steps with the fast system, and* (4.5) *becomes:*

$$\mathbf{y}_{\mathrm{S},n+1} = \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}} \cdot \mathbf{y}_{\mathrm{S},n}, \quad \mathbf{y}_{\mathrm{F},n+1} = \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}} \cdot \mathbf{y}_{\mathrm{F},n},$$

where the slow and fast stability functions of the forward Euler method over a macro-step are defined as:

$$\mathbf{R}^{\mathrm{S}}_{\mathrm{FE}} := 1 + z_{\mathrm{S}} \quad \text{and} \quad \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}} := \left(1 + \frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}}, \tag{4.7}$$

respectively. In this case the stability of the multirate integration is equivalent to the stability of each of the base schemes:

$$
\begin{aligned}
&|\,\mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\,| \in (0,1] \;\; |\,1+z_{\mathrm{S}}\,| \le 1 \quad\Leftrightarrow\quad |\,H\lambda_{\mathrm{S}}\,| \le 2 \quad\Leftrightarrow\quad H \le \frac{2}{|\,\lambda_{\mathrm{S}}\,|},\\
&|\,\mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}\,| \in (0,1] \;\; |\,1+z_{\mathrm{F}}/\mathtt{m}\,| \le 1 \quad\Leftrightarrow\quad |\,H\lambda_{\mathrm{F}}/\mathtt{m}\,| = |\,h\lambda_{\mathrm{F}}\,| \le 2 \quad\Leftrightarrow\quad h \le \frac{2}{|\,\lambda_{\mathrm{F}}\,|}.
\end{aligned}
\tag{4.8}
$$

Definition 3 (Linear stability) *The multirate explicit Euler method is linearly stable if all the eigenvalues of the transfer matrix* $\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}$ *have absolute values smaller than one.*

The eigenvalues of $\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}$ are:

$$
\frac{\operatorname{tr}\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}}{2} \pm \sqrt{\left(\frac{\operatorname{tr}\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}}{2}\right)^2 - \det\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}}.
$$

Using the definitions (4.7), we get

$$
\begin{aligned}
\operatorname{tr}(\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}) &= (1+z_{\mathrm{S}}) + (1+z_{\mathrm{F}}/\mathtt{m})^{\mathtt{m}} = \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}},\\
\det(\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}) &= (1+z_{\mathrm{S}})(1+z_{\mathrm{F}}/\mathtt{m})^{\mathtt{m}} + \big(1-(1+z_{\mathrm{F}}/\mathtt{m})^{\mathtt{m}}\big) w_{\mathrm{F}} w_{\mathrm{S}}/z_{\mathrm{F}}\\
&= \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\,\mathbf{R}^{\mathrm{F}}_{\mathrm{FE}} + \mathtt{k}\left(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}\right) z_{\mathrm{S}}\\
&= \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\,\mathbf{R}^{\mathrm{F}}_{\mathrm{FE}} - \mathtt{k}(1-\mathbf{R}^{\mathrm{S}}_{\mathrm{FE}})(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}),
\end{aligned}
$$

and the eigenvalues can be rewritten as

$$
\frac{\mathbf{R}^{\mathrm{S}}_{\mathrm{FE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}}{2} \pm \sqrt{\left(\frac{\mathbf{R}^{\mathrm{S}}_{\mathrm{FE}} - \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}}{2}\right)^2 + \mathtt{k}(1-\mathbf{R}^{\mathrm{S}}_{\mathrm{FE}})(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{FE}})}
$$

in terms of the stability functions $\mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}$ and $\mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}$ of the underlying explicit Euler schemes and the coupling coefficient $\mathtt{k}$.

This complicated formula is not easily amenable to a closed-form analysis. The stability can be easily analyzed with the help of the following result.

Lemma 4.2 (Routh-Hurwitz discrete stability criterion [?, ?]) *The matrix $\mathbf{R} \in \mathbb{R}^{2\times 2}$ has a spectral radius bounded by one if and only if the following three conditions hold:*

$$a) \qquad 1 + \mathrm{tr}(\mathbf{R}) + \det(\mathbf{R}) > 0, \tag{4.9a}$$

$$b) \qquad \det(\mathbf{R}) < 1, \qquad \text{and} \tag{4.9b}$$

$$c) \qquad 1 - \mathrm{tr}(\mathbf{R}) + \det(\mathbf{R}) > 0. \tag{4.9c}$$

Comment 4.5 (One-way coupled system) *Note that in case of one-way coupling ($w_{\mathrm{F}} = 0$ or $w_{\mathrm{S}} = 0$) the stability of the multirate scheme is given by the stability of the Euler scheme applied to each component with the corresponding step* (4.8)*.*

Theorem 2 (Constant interpolation) *Assume that the step sizes H and h are sufficiently small such that both base schemes, applied to the decoupled slow and fast systems, are linearly stable* (4.8)*. Then the multirate forward Euler scheme applied to the two-dimensional system* (4.33) *with constant interpolation is stable for*

$$\mathtt{k} \in \left(\frac{\mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\,\mathbf{R}^{\mathrm{F}}_{\mathrm{FE}} - 1}{\big(1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\big)\big(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}\big)}, \frac{\big(1 + \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\big)\big(1 + \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}\big)}{\big(1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\big)\big(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}\big)} \right) \cap (-\infty, 1),$$

unstable otherwise.

Proof: We use the three criteria (4.9) discussed in Lemma 4.2.

- For the first criterium (4.9a) in Lemma 4.2 we have:

  $$1 + \mathrm{tr}(\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}) + \det(\mathbf{R}^{\mathrm{CON}}_{\mathrm{MRFE}}) \;=\; \big(1 + \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\big)\big(1 + \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}\big) - \mathtt{k}\,\big(1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\big)\big(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}\big),$$

  which yields the condition

  $$\mathtt{k} < \min\left\{ \frac{\big(1 + \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\big)\big(1 + \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}\big)}{\big(1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}}\big)\big(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}}\big)}, 1 \right\}$$

  where we have used that $\mathtt{k} < 1$ and

  $$1 + \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}} > 0, \quad 1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{FE}} \geq 0, \quad 1 + \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}} > 0, \quad 1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{FE}} \geq 0. \tag{4.10}$$

- We next check the second criterium (4.9b) in Lemma 4.2. For all $\mathtt{k} < 0$ we have $-\mathtt{k}\left(1 - \mathbf{R}_{\mathrm{FE}}^{\mathrm{S}}\right)\left(1 - \mathbf{R}_{\mathrm{FE}}^{\mathrm{F}}\right) > 0$. Thus the determinant can get arbitrarily large for $H > 0$:

$$\lim_{\mathtt{k}\to-\infty} \det(\mathbf{R}_{\mathrm{MRFE}}^{\mathrm{CON}}) = \infty.$$

  For stability we need:

$$\det(\mathbf{R}_{\mathrm{MRFE}}^{\mathrm{CON}}) < 1 \quad \Leftrightarrow \quad \mathtt{k} > \frac{\mathbf{R}_{\mathrm{FE}}^{\mathrm{S}}\,\mathbf{R}_{\mathrm{FE}}^{\mathrm{F}} - 1}{\left(1 - \mathbf{R}_{\mathrm{FE}}^{\mathrm{S}}\right)\left(1 - \mathbf{R}_{\mathrm{FE}}^{\mathrm{F}}\right)}$$

- Finally, we check the third criterium (4.9c) in Lemma 4.2:

$$\begin{aligned} 1 - \mathrm{tr}(\mathbf{R}_{\mathrm{MRFE}}^{\mathrm{CON}}) + \det(\mathbf{R}_{\mathrm{MRFE}}^{\mathrm{CON}}) &= 1 - \mathbf{R}_{\mathrm{FE}}^{\mathrm{S}} - \mathbf{R}_{\mathrm{FE}}^{\mathrm{F}} + \mathbf{R}_{\mathrm{FE}}^{\mathrm{S}}\,\mathbf{R}_{\mathrm{FE}}^{\mathrm{F}} - \mathtt{k}\left(1 - \mathbf{R}_{\mathrm{FE}}^{\mathrm{S}}\right)\left(1 - \mathbf{R}_{\mathrm{FE}}^{\mathrm{F}}\right) \\ &= (1 - \mathtt{k})\left(1 - \mathbf{R}_{\mathrm{FE}}^{\mathrm{S}}\right)\left(1 - \mathbf{R}_{\mathrm{FE}}^{\mathrm{F}}\right) \\ &> 0, \end{aligned}$$

  since $1 - \mathtt{k} > 0$.

□

Comment 4.6 *In any case, stability is given for* $\mathtt{k}$ *in a neighborhood of zero, i.e., there exists some* $\epsilon > 0$ *depending on* $\mathbf{R}_{\mathrm{FE}}^{\mathrm{S}}$ *and* $\mathbf{R}_{\mathrm{FE}}^{\mathrm{F}}$ *such that the multirate explicit Euler schemes are stable for all* $\mathtt{k} \in [-\epsilon, \epsilon]$.

4.4 Multirate implicit Euler method

4.4.1 Multiscale partitioned initial value problems

Consider the partitioned initial-value problem (4.15). We apply the implicit Euler method to the slow and fast components. If the classical implicit Euler method is used then a system of $d = d_{\mathrm{S}} + d_{\mathrm{F}}$ nonlinear equations is solved at each time step. Assuming that a Newton-like method is employed, and that a direct linear algebra solver is used, then the cost per time step is $\mathcal{O}(d^3)$.

The cost of advancing the entire system from t_n to t_{n+1} using $\mathtt{m}$ implicit Euler steps with a small time step h is therefore:

$$\text{cost of traditional implicit Euler} \sim \mathcal{O}\big(\mathtt{m}\,(d_{\mathrm{S}} + d_{\mathrm{F}})^3\big). \tag{4.11}$$

In a multirate approach the slow and the fast subsystems are solved with different time steps. The goal is to perform the integration at a cost smaller than (4.11), without sacrificing accuracy. There are several different possibilities to couple the two subsystems. These approaches are discussed next.

The fully-coupled approach

In the spirit of single-rate backward Euler method we take implicit steps (4.3) for the slow component

$$\mathbf{y}_{\mathrm{S},n+1} = \mathbf{y}_{\mathrm{S},n} + H\,\mathbf{f}_{\mathrm{S}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}_{\mathrm{F},n+1}\big) \tag{4.12a}$$

as well as for the fast components:

$$\begin{aligned} \mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}} \;&=\; \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + h\,\mathbf{f}_{\mathrm{F}}\big(t_{n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}}\big), \\ &\qquad \ell = 0, \ldots, \mathtt{m}-1. \end{aligned} \tag{4.12b}$$

Note that the value of the argument $\mathbf{y}_{\mathrm{F},n+1}$ in (4.12a) is the solution (4.12b) after $\mathtt{m}$ small steps. The argument $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}$ in (4.12b) is obtained by interpolation, for which several options are possible, as follows:

$$\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}} = \begin{cases} \mathbf{y}_{\mathrm{S},n} & \text{(constant at its } t_n \text{ value)}, \qquad (4.13a) \\ \mathbf{y}_{\mathrm{S},n+1} & \text{(constant at its } t_{n+1} \text{ value)}, \qquad (4.13b) \\ \dfrac{\mathtt{m}-\ell}{\mathtt{m}}\,\mathbf{y}_{\mathrm{S},n} + \dfrac{\ell}{\mathtt{m}}\,\mathbf{y}_{\mathrm{S},n+1} & \text{(linear)}, \qquad (4.13c) \\ \mathbf{y}_{\mathrm{S},n} + \ell\,h\,\mathbf{f}_{\mathrm{S}}\big(t_n, \mathbf{y}_{\mathrm{S},n}, \mathbf{y}_{\mathrm{F},n}\big) & \text{(Hermite)}. \qquad (4.13d) \end{cases}$$

The fully coupled approach uses either a constant interpolant with the t_{n+1} value or a linear interpolant, such that $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}$ depends on the solution of (4.12a). It is clear from this discussion that in the fully coupled approach the computational process (4.12) leads to a very large system of nonlinear equations that solves simultaneously $d_{\mathrm{S}} + \mathtt{m} \times d_{\mathrm{F}}$ equations for the slow solution $\mathbf{y}_{\mathrm{S},n+1}$ and for all intermediate fast solutions $\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}}$, $\ell = 1, \ldots, \mathtt{m}$. The total cost

$$\text{cost of fully coupled MRBE} \sim \mathcal{O}\big((d_{\mathrm{S}} + \mathtt{m}\, d_{\mathrm{F}})^3\big). \tag{4.14}$$

exceeds that of taking small steps with the classical implicit Euler method (4.11). *Therefore the fully coupled implicit multirate approach is impractical due to its very large computational cost.*

The decoupled slowest-first approach

In order to reduce computational costs a simple idea is to apply the backward Euler method to solve the slow and fast variables in a decoupled way. The decoupling is realized by using in the slow solution only known past values of the fast variable, and vice-versa. Using (4.3) with a large step size H to solve the slow components

$$\mathbf{y}_{\mathrm{S},n+1} = \mathbf{y}_{\mathrm{S},n} + H\, \mathbf{f}_{\mathrm{S}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}_{\mathrm{F},n}\big), \tag{4.15a}$$

leads to a system of nonlinear equations in $\mathbf{y}_{\mathrm{S},n+1}$. This formula in implicit in the slow variables and explicit in the fast variables. The fast solution is obtained by applying the backward Euler method (4.3) with a small step size $h = H/\mathtt{m}$

$$\begin{aligned} \mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}} = &\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + h\, \mathbf{f}_{\mathrm{F}}\big(t_{n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}}\big), \\ &\ell = 0, \ldots, \mathtt{m}-1. \end{aligned} \tag{4.15b}$$

The intermediate slow values $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}$ are obtained from the known $\mathbf{y}_{\mathrm{S},n}$ and $\mathbf{y}_{\mathrm{S},n+1}$ by applying one of the interpolation formulas (4.13). Since the fast components are treated explicitly in the slow formula (4.15a) the decoupled slowest-first approach raises stability concerns. The total cost of

the slowest-first approach is that of one implicit slow step (4.15a) plus $\mathtt{m}$ implicit fast steps (4.15b):

$$\text{cost of decoupled slowest-first MRBE} \sim \mathcal{O}\big(\mathtt{m}\, {d_{\mathrm{F}}}^3 + {d_{\mathrm{S}}}^3\big). \tag{4.16}$$

The decoupled fastest-first approach

This approach proceeds with solving the fast variable as in the decoupled approach, since (4.15b) depends only on the past value of the slow variable.

$$\begin{aligned} \mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}} = \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + h\, \mathbf{f}_{\mathrm{F}}\big(t_{n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}}\big), \\ \ell = 0, \ldots, \mathtt{m}-1. \end{aligned} \tag{4.17a}$$

The intermediate values of the slow variable are computed either by constant interpolation with the value at t_n (4.13a) or by linear Hermite interpolation (4.13d). In both cases the slow variable values $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}$ depend only on the known past value $\mathbf{y}_{\mathrm{S},n}$. The slow variable is treated explicitly, while the fast variable is treated implicitly in (4.17a).

The slow variable is then computed using:

$$\mathbf{y}_{\mathrm{S},n+1} = \mathbf{y}_{\mathrm{S},n} + H\, \mathbf{f}_{\mathrm{S}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}_{\mathrm{F},n+1}\big). \tag{4.17b}$$

The fast variable value is the one obtained from the integration (4.17a). The total cost of the fastest-first approach is that of $\mathtt{m}$ implicit fast steps (4.17a) plus one implicit slow step (4.17b):

$$\text{cost of decoupled fastest-first MRBE} \sim \mathcal{O}\big(\mathtt{m}\, {d_{\mathrm{F}}}^3 + {d_{\mathrm{S}}}^3\big). \tag{4.18}$$

The coupled slowest-first approach

In the coupled slowest–first approach both components are solved together:

$$\begin{aligned} \mathbf{y}_{\mathrm{S},n+1} &= \mathbf{y}_{\mathrm{S},n} + H\, \mathbf{f}_{\mathrm{S}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}^*_{\mathrm{F},n+1}\big), \\ \mathbf{y}^*_{\mathrm{F},n+1} &= \mathbf{y}_{\mathrm{F},n} + H\, \mathbf{f}_{\mathrm{F}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}^*_{\mathrm{F},n+1}\big). \end{aligned} \tag{4.19a}$$

The fast component $\mathbf{y}^*_{\mathrm{F},n+1}$ is inaccurate for large H and is discarded. The fast solution is obtained by applying the backward Euler method (4.3) with a small step size $h = H/\mathtt{m}$

$$\begin{aligned}\mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}} = \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + h\,\mathbf{f}_{\mathrm{F}}\big(t_{n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}}\big),\\ \ell = 0,\dots,\mathtt{m}-1.\end{aligned} \tag{4.19b}$$

The intermediate slow variables $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}$ for $\ell = 1,\dots,\mathtt{m}-1$ can be approximated by any interpolation formula in (4.13).

The cost is given by one large Euler step with the full system (4.19a) plus $\mathtt{m}$ small steps with the fast subsystem (4.19b):

$$\text{cost of coupled slowest-first MRBE} \sim \mathcal{O}\big((d_{\mathrm{F}}+d_{\mathrm{S}})^3 + \mathtt{m}\,{d_{\mathrm{F}}}^3\big). \tag{4.20}$$

The coupled-first-step approach

In order to avoid computing and discarding a fast solution in (4.19a) we can couple the slow backward Euler step with the first fast backward Euler step, and use zeroth order interpolation in both formulas:

$$\mathbf{y}_{\mathrm{S},n+1} = \mathbf{y}_{\mathrm{S},n} + H\,\mathbf{f}_{\mathrm{S}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\big), \tag{4.21a}$$

$$\mathbf{y}_{\mathrm{F},n+1/\mathtt{m}} = \mathbf{y}_{\mathrm{F},n} + h\,\mathbf{f}_{\mathrm{F}}\big(t_{n+1/\mathtt{m}}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\big). \tag{4.21b}$$

The fast steps (4.19b) are then carried out for $\ell = 1,\dots,\mathtt{m}-1$. The cost is given by:

$$\text{cost of coupled-first-step MRBE} \sim \mathcal{O}\big((d_{\mathrm{F}}+d_{\mathrm{S}})^3 + (\mathtt{m}-1)\,{d_{\mathrm{F}}}^3\big). \tag{4.22}$$

4.4.2 Accuracy analysis of multirate implicit Euler

We assume that the functions $\mathbf{f}_{\mathrm{S}}$, $\mathbf{f}_{\mathrm{F}}$ are smooth and that all their first and second order partial derivatives are uniformly bounded. The analysis is similar to the one carried out in Section 4.3.5. We will not consider the fully-coupled approach further as it is not advantageous from a computational point of view.

Before discussing the different approaches of implementing implicit multirate Euler scheme, we note that the approximation formula for the fast components is always the same (besides the first fast step in the coupled first-step approach). It reads:

$$\mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}} = \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + h\,\mathbf{f}_{\mathrm{F}}\big(t_{n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}}\big), \\ \ell = 0, \ldots, \mathtt{m}-1.$$

Only the evaluation of the intermediate slow components $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}$ is influenced by the interpolation formula used, which may depend on whether new updates of $\mathbf{y}_{\mathrm{S}}$ at time point t_{n+1} are available or not when computing the fast approximations.

Accuracy of the fast components

Recapitulating the analysis of Section 4.3.5 for the error in the fast variables, instead of (4.30) we obtain the estimate:

$$\begin{aligned}\|\Delta\mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}}\| \;\leq\; & \frac{1}{1-h\,L_{\mathrm{F,F}}}\big(h\,L_{\mathrm{F,S}}\,\|\Delta\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}\| + \|\Delta\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}}\| + \\ & \|\boldsymbol{\delta}_{\mathrm{F},n+\ell/\mathtt{m}}\|\big) \end{aligned} \tag{4.23}$$

for sufficiently small step sizes h such that $0 < 1 - h\,L_{\mathrm{F,F}} < 1$ holds. Note that in this case the LTE $\boldsymbol{\delta}_{\mathrm{F},n+\ell/\mathtt{m}}$ is defined by:

$$\begin{aligned}\mathbf{y}_{\mathrm{F}}(t_{n+(\ell+1)/m}) \;=\; & \mathbf{y}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}}) + h\,\mathbf{f}_{\mathrm{F}}\big(t_{n+(\ell+1)/\mathtt{m}}, \mathbf{y}_{\mathrm{S}}(t_{n+(\ell+1)/\mathtt{m}}), \mathbf{y}_{\mathrm{F}}(t_{n+(\ell+1)/\mathtt{m}})\big) - \\ & \boldsymbol{\delta}_{\mathrm{F},n+\ell/\mathtt{m}}.\end{aligned}$$

With $\mathbf{y}_{\mathrm{F}}(t_{n+\ell/\mathtt{m}}) = \mathbf{y}_{\mathrm{F}}(t_{n+(\ell+1)/m} - h)$ we expand about $t_{n+(\ell+1)/m}$, and bound the LTE as follows:

$$\|\boldsymbol{\delta}_{\mathrm{F},n+\ell/\mathtt{m}}\| \;\leq\; \frac{h^2}{2} \max_{\tau\in[t_{n+\ell/\mathtt{m}}, t_{n+(\ell+1)/\mathtt{m}}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|.$$

Using this bound for the LTE in (4.23) gives the updated estimate:

$$\begin{aligned}\|\Delta\mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}}\| \;\leq\; & \frac{1}{1-h\,L_{\mathrm{F,F}}}\Big(h\,L_{\mathrm{F,S}}\,\|\Delta\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}}\| + \|\Delta\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}}\| \\ & + \frac{h^2}{2} \max_{\tau\in[t_{n+\ell/\mathtt{m}}, t_{n+(\ell+1)/\mathtt{m}}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|\Big). \end{aligned} \tag{4.24}$$

Iterating for $\ell = 0, \ldots, \mathtt{m}-1$ and using $\Delta \mathbf{y}_{\mathrm{F},n} = 0$ leads to the estimate:

$$\begin{aligned}
\|\Delta \mathbf{y}_{\mathrm{F},n+1}\| \;\leq\; & \sum_{k=0}^{\mathtt{m}-1} \left(\frac{1}{1-h\,L_{\mathrm{F,F}}}\right)^{k+1} \cdot \Big(h\,L_{\mathrm{F,S}}\,\|\Delta \mathbf{y}_{\mathrm{S},n+(\mathtt{m}-k)/\mathtt{m}}\| \\
& + \frac{h^2}{2} \max_{\tau\in[t_{n+(\mathtt{m}-k-1)/\mathtt{m}},\, t_{n+(\mathtt{m}-k)/\mathtt{m}}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| \Big) \\
\leq\; & \frac{1}{1-H\,L_{\mathrm{F,F}}} \sum_{k=0}^{\mathtt{m}-1} \Big(h\,L_{\mathrm{F,F}}\,\|\Delta \mathbf{y}_{\mathrm{S},n+(\mathtt{m}-k)/\mathtt{m}}\| \\
& + \frac{h^2}{2} \max_{\tau\in[t_{n+(\mathtt{m}-k-1)/\mathtt{m}},\, t_{n+(\mathtt{m}-k)/\mathtt{m}}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| \Big) \\
\leq\; & \frac{1}{1-H\,L_{\mathrm{F,F}}} \Big(\underbrace{\frac{H^2}{2\mathtt{m}} \max_{\tau\in[t_n t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|}_{E_1} + \underbrace{\sum_{k=0}^{\mathtt{m}-1} h\,L_{\mathrm{F,S}}\,\|\Delta \mathbf{y}_{\mathrm{S},n+(\mathtt{m}-k)/\mathtt{m}}\|}_{E_2} \Big).
\end{aligned} \tag{4.25}$$

The first term E_1 in the last inequality represents the LTE of the fast variables. The second term E_2 depends on the interpolation formula used for the slow components. It takes the following values:

- For constant interpolation with $\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}} = \mathbf{y}_{\mathrm{S},n}$:

$$E_2 = \frac{H^2}{2}\,L_{\mathrm{F,S}} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\|.$$

- For constant interpolation with $\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}} = \mathbf{y}_{\mathrm{S},n+1}$:

$$E_2 = \frac{H^2}{2}\,L_{\mathrm{F,S}} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + H\|\Delta \mathbf{y}_{\mathrm{S},n+1}\| \approx \frac{H^2}{2} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\|.$$

- For linear interpolation with $\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}} = \frac{\mathtt{m}-\ell}{\mathtt{m}}\mathbf{y}_{\mathrm{S},n} + \frac{\ell}{\mathtt{m}}\mathbf{y}_{\mathrm{S},n+1}$:

$$E_2 = (H+h)\,L_{\mathrm{F,S}} \left(\frac{H^2}{2} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + \frac{1}{2}\|\Delta \mathbf{y}_{\mathrm{S},n+1}\| \right).$$

- For Hermite interpolation with $\mathbf{y}_{\mathrm{S},n+\ell/\mathtt{m}} = \mathbf{y}_{\mathrm{S},+}\ell h \mathbf{f}_{\mathrm{S}}(t_n, \mathbf{y}_{\mathrm{S},n}, \mathbf{y}_{\mathrm{F},n})$:

$$E_2 = \frac{1}{12}H^3\,L_{\mathrm{F,S}} \left(2 + \frac{3}{\mathtt{m}} + \frac{1}{\mathtt{m}^2}\right) \max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\|.$$

It now remains to estimate the approximation errors in the slow variables for different implicit multirate Euler schemes.

The decoupled slowest-first approach

For the decoupled slowest-first approach the error in the slow components is:

$$\begin{aligned}\Delta\mathbf{y}_{\mathrm{S},n+1} &= \mathbf{y}_{\mathrm{S},n+1} - \mathbf{y}_{\mathrm{S}}(t_{n+1}) \\ &= H\mathbf{f}_{\mathrm{S}}(t_{n+1}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}_{\mathrm{F},n}) - H\mathbf{f}_{\mathrm{S}}(t_n, \mathbf{y}_{\mathrm{S},n}, \mathbf{y}_{\mathrm{F},n}) - \frac{H^2}{2}\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\end{aligned}$$

for an appropriate $\tau \in [t_n, t_{n+1}]$. Using the mean value theorem and writing $\mathbf{y}_{\mathrm{S},n+1} - \mathbf{y}_{\mathrm{S},n}$ as $\Delta\mathbf{y}_{\mathrm{S},n+1} + \mathbf{y}_{\mathrm{S}}(t_{n+1}) - \mathbf{y}_{\mathrm{S}}(t_n)$, we obtain the estimate

$$\|\Delta\mathbf{y}_{\mathrm{S},n+1}\| \leq \frac{H^2}{1 - H\,L_{\mathrm{S,S}}}\left(L_{\mathrm{S},t} + L_{\mathrm{S,S}} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + \frac{1}{2}\max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\|\right) \tag{4.26}$$

for sufficiently small macro step sizes H.

The decoupled fastest-first approach

The accuracy analysis of the slow variables is different than in the slowest-first approach, as these are computed after the fast ones and can make use of $\mathbf{y}_{\mathrm{F},n+1}$. We use a similar approach as for the decoupled slowest-first approach to obtain the following error estimate for the slow variables:

$$\begin{aligned}\|\Delta\mathbf{y}_{\mathrm{S},n+1}\| \;\leq\; \frac{H^2}{1 - H\,L_{\mathrm{S,S}}}\Bigg(&L_{\mathrm{S},t} + L_{\mathrm{S,S}} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + \frac{1}{2}\max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| \\ &+ L_{\mathrm{S,F}} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{F}}(\tau)\| + \frac{L_{\mathrm{S,F}}}{H}\|\Delta\mathbf{y}_{\mathrm{F},n+1}\|\Bigg).\end{aligned}$$

Since $\|\Delta\mathbf{y}_{\mathrm{F},n+1}\| \sim \mathcal{O}(H^2)$, the last term in the estimate is of order H^3 and can be neglected.

The coupled slowest-first approach

In the coupled slowest-first approach (4.19) we first compute an implicit Euler step with macro-step size H for the entire system to obtain the approximation for the slow variables, and then throw away the approximation for the fast variables.

From (4.19a) we obtain the error equation:

$$\begin{aligned}
\Delta\mathbf{y}_{\mathrm{S},n+1} &= \Delta\mathbf{y}_{\mathrm{S},n} + H\,\mathbf{f}_{\mathrm{S}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}^*_{\mathrm{F},n+1}\big) - H\,\mathbf{f}_{\mathrm{S}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S}}(t_{n+1}), \mathbf{y}_{\mathrm{F}}(t_{,n+1})\big) \\
&\quad - \int_0^H \tau\,\ddot{\mathbf{y}}_{\mathrm{S}}(t_n+\tau)\,d\tau \\
\Delta\mathbf{y}^*_{\mathrm{F},n+1} &= \Delta\mathbf{y}_{\mathrm{F},n} + H\,\mathbf{f}_{\mathrm{F}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S},n+1}, \mathbf{y}^*_{\mathrm{F},n+1}\big) - H\,\mathbf{f}_{\mathrm{F}}\big(t_{n+1}, \mathbf{y}_{\mathrm{S}}(t_{n+1}), \mathbf{y}_{\mathrm{F}}(t_{,n+1})\big) \\
&\quad - \int_0^H \tau\,\ddot{\mathbf{y}}_{\mathrm{F}}(t_n+\tau)\,d\tau.
\end{aligned}$$

For the local truncation error analysis we start the current step from the exact solution, i.e., we assume that the errors at the beginning of the time step are zero: $\Delta\mathbf{y}_{\mathrm{S},n} = 0$ and $\Delta\mathbf{y}_{\mathrm{F},n} = 0$. Taking norms, and using the Jacobian norm bounds (4.29) leads to the following estimates:

$$\begin{aligned}
\|\Delta\mathbf{y}_{\mathrm{S},n+1}\| &\le H\,L_{\mathrm{S,S}}\,\|\Delta\mathbf{y}_{\mathrm{S},n+1}\| + H\,L_{\mathrm{S,F}}\,\|\Delta\mathbf{y}^*_{\mathrm{F},n+1}\| \\
&\quad + \frac{H^2}{2}\max_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| \\
\|\Delta\mathbf{y}^*_{\mathrm{F},n+1}\| &\le H\,L_{\mathrm{F,S}}\,\|\Delta\mathbf{y}_{\mathrm{S},n+1}\| + H\,L_{\mathrm{F,F}}\,\|\Delta\mathbf{y}^*_{\mathrm{F},n+1}\| \\
&\quad + \frac{H^2}{2}\max_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|,
\end{aligned}$$

which are written in matrix form as follows:

$$\begin{bmatrix} 1 - H\,L_{\mathrm{S,S}} & -H\,L_{\mathrm{S,F}} \\ -H\,L_{\mathrm{F,S}} & 1 - H\,L_{\mathrm{F,F}} \end{bmatrix} \cdot \begin{bmatrix} \|\Delta\mathbf{y}_{\mathrm{S},n+1}\| \\ \|\Delta\mathbf{y}_{\mathrm{F},n+1}{}^{\star}\| \end{bmatrix} \le \frac{H^2}{2} \begin{bmatrix} \max\limits_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| \\ \max\limits_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| \end{bmatrix}.$$

In the asymptotic regime where H is sufficiently small such that $H\,L_{\mathrm{S,S}} < 1$ and $H\,L_{\mathrm{F,F}} < 1$ the inverse of the matrix on the left hand side has all entries

positive. Multiplication by this matrix with positive entries leads to the following estimate for the slow errors:

$$\begin{aligned}\|\Delta\mathbf{y}_{\mathrm{S},n+1}\| \;\leq\; & \frac{1-H\,L_{\mathrm{F,F}}}{(1-H\,L_{\mathrm{S,S}})(1-H\,L_{\mathrm{F,F}})-H^2L_{\mathrm{S,F}}L_{\mathrm{F,S}}}\cdot\Big(\frac{H^2}{2}\max_{\tau\in[t_n,t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\|\Big)\\ &+\frac{L_{\mathrm{S,F}}}{(1-H\,L_{\mathrm{S,S}})(1-H\,L_{\mathrm{F,F}})-H^2L_{\mathrm{S,F}}L_{\mathrm{F,S}}}\cdot\Big(\frac{H^3}{2}\max_{\tau\in[t_n,t_{n+1}]}\|\dddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|\Big).\end{aligned}$$

The coupled-first-step approach

The coupled-first-step approach (4.21) starts with an implicit Euler step with macro-step size H for the slow, and micro-step size $h = H/\mathtt{m}$ for the fast variable, to obtain the approximation for the slow variables at t_{n+1} and the fast ones at $t_{n+1/\mathtt{m}}$. We proceed similarly to the slowest-first case to obtain:

$$\begin{aligned}\|\Delta\mathbf{y}_{\mathrm{S},n+1}\| \;\leq\; & \|\Delta\mathbf{y}_{\mathrm{S},n}\|+HL_{\mathrm{S,S}}\|\Delta\mathbf{y}_{\mathrm{S},n+1}\|+HL_{\mathrm{S,F}}\|\mathbf{y}_{\mathrm{F}}(t_{n+1})-\mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\|\\ &+\frac{H^2}{2}\max_{t_n\leq\tau\leq t_{n+1}}\|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\|.\end{aligned}$$

The difference in the fast variables is bounded as follows:

$$\|\mathbf{y}_{\mathrm{F}}(t_{n+1})-\mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\|\leq\|\Delta\mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\|+H\max_{t_n\leq\tau\leq t_{n+1}}\|\dot{\mathbf{y}}_{\mathrm{F}}(\tau)\|.$$

Similarly, we obtain the following bound for $\|\Delta\mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\|$:

$$\begin{aligned}\|\Delta\mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\| \;\leq\; & \|\Delta\mathbf{y}_{\mathrm{F},n}\|+hL_{\mathrm{F,F}}\|\Delta\mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\|+\frac{h^2}{2}\max_{t_n\leq\tau\leq t_{n+1/\mathtt{m}}}\|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|\\ &+hL_{\mathrm{F,S}}\left(\|\Delta\mathbf{y}_{\mathrm{S},n+1}\|+H\max_{t_n\leq\tau\leq t_{n+1}}\|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\|\right).\end{aligned}$$

Starting the step from the exact solution implies that $\|\Delta \mathbf{y}_{\mathrm{S},n}\| = \|\Delta \mathbf{y}_{\mathrm{F},n}\| = 0$, and leads to the following local error inequality:

$$\begin{bmatrix} 1 - H\,L_{\mathrm{S,S}} & -H\,L_{\mathrm{S,F}} \\ -h\,L_{\mathrm{F,S}} & 1 - h\,L_{\mathrm{F,F}} \end{bmatrix} \cdot \begin{bmatrix} \|\Delta \mathbf{y}_{\mathrm{S},n+1}\| \\ \|\Delta \mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\| \end{bmatrix} \leq \\ \begin{bmatrix} \dfrac{H^2}{2} \max\limits_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + H^2 L_{\mathrm{S,F}} \max\limits_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{F}}(\tau)\| \\ \dfrac{h^2}{2} \max\limits_{\tau\in[t_n,t_{n+1/\mathtt{m}}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| + hHL_{\mathrm{F,S}} \max\limits_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\| \end{bmatrix}.$$

This leads to the estimate:

$$\begin{aligned} \|\Delta \mathbf{y}_{\mathrm{S},n+1}\| \;\leq\; & \frac{1}{(1 - H\,L_{\mathrm{S,S}})(1 - h\,L_{\mathrm{F,F}}) - hH\,L_{\mathrm{S,F}}L_{\mathrm{F,S}}} \cdot \\ & \Big\{ (1 - hL_{\mathrm{F,F}}) \Big(\frac{H^2}{2} \max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + H^2 L_{\mathrm{S,F}} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{F}}(\tau)\| \Big) \\ & + HL_{\mathrm{S,F}} \Big(\frac{h^2}{2} \max_{\tau\in[t_n,t_{n+1/\mathtt{m}}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| + hHL_{\mathrm{F,S}} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\| \Big) \Big\}, \\ \|\Delta \mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\| \;\leq\; & \frac{1}{(1 - h\,L_{\mathrm{F,F}}) - hH\,L_{\mathrm{S,F}}L_{\mathrm{F,S}}} \cdot \\ & \Big\{ hL_{\mathrm{F,S}} \Big(\frac{H^2}{2} \max_{\tau\in[t_n,t_{n+1}]} \|\ddot{\mathbf{y}}_{\mathrm{S}}(\tau)\| + H^2 L_{\mathrm{S,F}} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{F}}(\tau)\| \Big) \\ & + (1 - H\,L_{\mathrm{S,S}}) \Big(\frac{h^2}{2} \max_{\tau\in[t_n,t_{n+1/\mathtt{m}}]} \|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\| + hHL_{\mathrm{F,S}} \max_{\tau\in[t_n,t_{n+1}]} \|\dot{\mathbf{y}}_{\mathrm{S}}(\tau)\| \Big) \Big\}. \end{aligned}$$

For the slow variables the dominant error term is of size $\mathcal{O}(H^2)$.

For the fast variables after the first micro-step the dominant error term has size $\mathcal{O}(h^2)$. This error is amplified with $\mathtt{m}$ for the derivatives of the slow components, which does not cause problems, as these derivatives are small.

The computation of the next micro-steps via (4.19b) is different then the computation of $\mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}$, and we have to update the estimate for $\Delta \mathbf{y}_{\mathrm{F},n+1}$.

The estimate (4.25) becomes:

$$\|\Delta \mathbf{y}_{\mathrm{F},n+1}\| \;\leq\; \frac{1}{1-H\,L_{\mathrm{F,F}}}\cdot\left(\frac{H^2}{2\mathtt{m}}\max_{\tau\in[t_n t_{n+1}]}\|\ddot{\mathbf{y}}_{\mathrm{F}}(\tau)\|+ \sum_{k=0}^{\mathtt{m}-2} h\,L_{\mathrm{F,S}}\,\|\Delta \mathbf{y}_{\mathrm{S},n+(\mathtt{m}-k)/\mathtt{m}}\|+\|\Delta \mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\|\right), \tag{4.27}$$

with $\|\Delta \mathbf{y}_{\mathrm{F},n+1/\mathtt{m}}\|$ given above, and $\|\Delta \mathbf{y}_{\mathrm{S},n+(\mathtt{m}-k)/\mathtt{m}}\|$ estimated by equation (4.25).

4.4.3 Linear stability analysis of multirate implicit Euler methods

The linear stability analysis follows the one developed in Section 4.3.6. To this end we apply the implicit schemes to solve the linear test problem (4.33). This gives an iteration of the form:

$$\begin{bmatrix}\mathbf{y}_{\mathrm{S},n+1}\\ \mathbf{y}_{\mathrm{F},n+1}\end{bmatrix}=\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}\cdot\begin{bmatrix}\mathbf{y}_{\mathrm{S},n}\\ \mathbf{y}_{\mathrm{F},n}\end{bmatrix},\quad \mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}\in\mathbb{R}^{2\times 2}. \tag{4.28}$$

The multirate backward Euler method is linearly stable if both eigenvalues of the matrix $\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}$ have absolute values smaller than or equal to one.

Definition 4 (Unconditional stability) *A multirate method is unconditionally stable if it is stable for any step sizes $H>0$ and $h>0$.*

Comment 4.7 (Linear stability for a decoupled system) *When the backward Euler method is applied to a decoupled test problem* (4.3) *where $w_{\mathrm{F}}=w_{\mathrm{S}}=0$ one takes one step with the slow system and $\mathtt{m}$ steps with the fast system, and* (4.5) *becomes:*

$$\mathbf{y}_{\mathrm{S},n+1}=\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\cdot\mathbf{y}_{\mathrm{S},n},\quad \mathbf{y}_{\mathrm{F},n+1}=\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\cdot\mathbf{y}_{\mathrm{F},n},$$

where the slow and fast stability functions of the backward Euler method over a macro-step are defined as:

$$\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}:=(1+z_{\mathrm{S}})^{-1}\in(0,1]\quad\text{and}\quad \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}:=\left(1+\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}}\in(0,1], \tag{4.29}$$

respectively. The decoupled schemes are stable for any $z_{\mathrm{F}},z_{\mathrm{S}}<0$.

The fully coupled approach

Application of the scheme (4.12) with constant interpolation of the slow variable at its t_{n+1} value (4.13b) to the linear test problem (4.33) gives:

$$\begin{aligned}
\mathbf{y}_{\mathrm{S},n+1} &= \mathbf{y}_{\mathrm{S},n} + z_{\mathrm{S}}\,\mathbf{y}_{\mathrm{S},n+1} + w_{\mathrm{F}}\,\mathbf{y}_{\mathrm{F},n+1} && (4.30)\\
&= (1-z_{\mathrm{S}})^{-1}\,(\mathbf{y}_{\mathrm{S},n} + w_{\mathrm{F}}\,\mathbf{y}_{\mathrm{F},n+1})\\
&= \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,\mathbf{y}_{\mathrm{S},n} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,w_{\mathrm{F}}\,\mathbf{y}_{\mathrm{F},n+1}; && (4.31)\\
\mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}} &= \mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + \frac{w_{\mathrm{S}}}{\mathtt{m}}\,\mathbf{y}_{\mathrm{S},n+1} + \frac{z_{\mathrm{F}}}{\mathtt{m}}\,\mathbf{y}_{\mathrm{F},n+(\ell+1)/\mathtt{m}} && (4.32)\\
&= \left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-1}\left(\mathbf{y}_{\mathrm{F},n+\ell/\mathtt{m}} + \frac{w_{\mathrm{S}}}{\mathtt{m}}\,\mathbf{y}_{\mathrm{S},n+1}\right),\\
&\qquad \ell = 0,\ldots,\mathtt{m}-1.
\end{aligned}$$

For the fast variables we obtain:

$$\begin{aligned}
\mathbf{y}_{\mathrm{F},n+1} &= \left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}}\mathbf{y}_{\mathrm{F},n} + \sum_{\ell=1}^{\mathtt{m}}\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\ell}\frac{w_{\mathrm{S}}}{\mathtt{m}}\,\mathbf{y}_{\mathrm{S},n+1} && (4.33)\\
&= \left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}}\mathbf{y}_{\mathrm{F},n} + \left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-1}\frac{1-\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}}}{1-\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-1}}\frac{w_{\mathrm{S}}}{\mathtt{m}}\,\mathbf{y}_{\mathrm{S},n+1}\\
&= \left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}}\mathbf{y}_{\mathrm{F},n} + \left(\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}}-1\right)\frac{w_{\mathrm{S}}}{z_{\mathrm{F}}}\,\mathbf{y}_{\mathrm{S},n+1}\\
&= \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\,\mathbf{y}_{\mathrm{F},n} + (\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}-1)\,\frac{w_{\mathrm{S}}}{z_{\mathrm{F}}}\,\mathbf{y}_{\mathrm{S},n+1}. && (4.34)
\end{aligned}$$

Putting it all together we have that

$$\begin{bmatrix} 1 & -\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,w_{\mathrm{F}} \\ (1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})\,\frac{w_{\mathrm{S}}}{z_{\mathrm{F}}} & 1 \end{bmatrix}\cdot\begin{bmatrix}\mathbf{y}_{\mathrm{S},n+1}\\ \mathbf{y}_{\mathrm{F},n+1}\end{bmatrix} = \begin{bmatrix}\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,\mathbf{y}_{\mathrm{S},n}\\ \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\,\mathbf{y}_{\mathrm{F},n}\end{bmatrix} \qquad (4.35)$$

Theorem 3 (Stability of fully coupled approach) *The fully coupled multirate backward Euler method* (4.12) *with constant interpolation of the slow variable, i.e.,* $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}} = \mathbf{y}_{\mathrm{S},n+1}$ *in* (4.12b)*, is unconditionally stable.*

Proof: The stability matrix is:

$$\mathbf{R}^{\mathrm{FC}}_{\mathrm{MRBE}} = [1+\mathtt{k}\,z_{\mathrm{S}}\,\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})]^{-1}\begin{bmatrix}\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} & \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\,\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,w_{\mathrm{F}}\\ -\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})\,\frac{w_{\mathrm{S}}}{z_{\mathrm{F}}} & \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\end{bmatrix}$$

which gives

$$\begin{aligned}
\mathrm{tr}(\mathbf{R}^{\mathrm{FC}}_{\mathrm{MRBE}}) &= \frac{\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}}{1 + \mathtt{k}\, z_{\mathrm{S}}\, \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})} = \frac{\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}}{1 - \mathtt{k}\, (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})}, \\
\det(\mathbf{R}^{\mathrm{FC}}_{\mathrm{MRBE}}) &= \frac{\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}}{1 + \mathtt{k}\, z_{\mathrm{S}}\, \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})} = \frac{\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}}{1 - \mathtt{k}\, (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})},
\end{aligned}$$

where we used the fact that:

$$\begin{aligned}
z_{\mathrm{S}}\, \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} = \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} - (1 - z_{\mathrm{S}})\, \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} = \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} - 1, \\
1 + \mathtt{k}\, z_{\mathrm{S}}\, \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) = 1 - \mathtt{k}\, (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})
\end{aligned}$$

Since $\mathtt{k} < 1$ and $0 \leq \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \leq 1$:

$$1 - \mathtt{k}\, (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) > 1 - (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})\,. \geq 0. \qquad (4.36)$$

- Check (4.9a):

$$\begin{aligned}
1 + \mathrm{tr}(\mathbf{R}^{\mathrm{FC}}_{\mathrm{MRBE}}) + \det(\mathbf{R}^{\mathrm{FC}}_{\mathrm{MRBE}}) &= \frac{1 - \mathtt{k}\, (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}}{1 - \mathtt{k}\, (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})} \\
&> 0.
\end{aligned}$$

  Stability follows from (4.36) and from:

$$\begin{aligned}
0 &\leq \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \\
&= 1 - (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \\
&< 1 - \mathtt{k}\, (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}.
\end{aligned}$$

- Check (4.9b). Since $\mathtt{k} < 1$ and $0 \leq \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \leq 1$:

$$\begin{aligned}
& \det(\mathbf{R}^{\mathrm{FC}}_{\mathrm{MRBE}}) < 1 \\
\Leftrightarrow\ & \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} < 1 - \mathtt{k}\, (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) \\
\Leftrightarrow\ & \mathtt{k} < \frac{1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}}{(1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})}
\end{aligned}$$

  Since $1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \geq (1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\, (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})$ for all $0 < \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \leq 1$ and since $\mathtt{k} < 1$ criterion (4.9b) holds.

- Check (4.9c):

$$\begin{aligned} 1 - \mathrm{tr}(\mathbf{R}^{\mathrm{FC}}_{\mathrm{MRBE}}) + \det(\mathbf{R}^{\mathrm{FC}}_{\mathrm{MRBE}}) &= \frac{1 - \mathtt{k}\,(1-\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\,(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}}{1 - \mathtt{k}\,(1-\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\,(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})} \\ &> 0. \end{aligned}$$

  Stability follows from (4.36) and from:

$$\begin{aligned} 0 &= \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \\ &\leq 1 - (1-\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\,(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \\ &< 1 - \mathtt{k}\,(1-\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})\,(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}. \end{aligned}$$

□

The decoupled slowest-first approach with constant interpolation

Theorem 4 (Stability of decoupled slowest-first approach) *The decoupled slowest-first multirate backward Euler method* (4.15) *with constant interpolation of the slow variable, i.e.,* $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}} = \mathbf{y}_{\mathrm{S},n}$ *in* (4.15b)*, is unconditionally stable if the system is weakly coupled,* $-1 \leq \mathtt{k} < 1$*; it becomes unstable for* $\mathtt{k} \to -\infty$*.*

Proof: For the decoupled approach (4.15), the recursion (4.35) turns into

$$\begin{bmatrix} 1-z_{\mathrm{S}} & 0 \\ 0 & \left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{y}_{\mathrm{S},n+1} \\ \mathbf{y}_{\mathrm{F},n+1} \end{bmatrix} = \begin{bmatrix} 1 & w_{\mathrm{F}} \\ \left(1-\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}}\right)\,\frac{w_{\mathrm{S}}}{z_{\mathrm{F}}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{y}_{\mathrm{S},n} \\ \mathbf{y}_{\mathrm{F},n} \end{bmatrix},$$

which yields the recursion matrix

$$\mathbf{R}^{\mathrm{DSFC}}_{\mathrm{MRBE}} = \begin{bmatrix} \frac{1}{1-z_{\mathrm{S}}} & \frac{w_{\mathrm{F}}}{1-z_{\mathrm{S}}} \\ \frac{w_{\mathrm{S}}}{z_{\mathrm{F}}}\,\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}}\left(1-\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}}\right) & \left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}} \end{bmatrix},$$

and

$$\begin{aligned} \mathrm{tr}(\mathbf{R}^{\mathrm{DSFC}}_{\mathrm{MRBE}}) &= (1-z_{\mathrm{S}})^{-1} + \left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}} = \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}, \\ \det(\mathbf{R}^{\mathrm{DSFC}}_{\mathrm{MRBE}}) &= (1-z_{\mathrm{S}})^{-1}\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{-\mathtt{m}}\left(1-\mathtt{k}\,z_{\mathrm{S}}\left(1-\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}}\right)\right) \\ &= \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\,(\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathtt{k}\,z_{\mathrm{S}}\,(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}))\,. \end{aligned}$$

The three Routh-Hurwitz criteria (4.9) have to hold.

- To check (4.9a):

$$\begin{aligned}
1 + \mathrm{tr}(\mathbf{R}^{\mathrm{DSFC}}_{\mathrm{MRBE}}) + \det(\mathbf{R}^{\mathrm{DSFC}}_{\mathrm{MRBE}}) &= 1 + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} \left(\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathtt{k}\, z_{\mathrm{S}} \left(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\right)\right) \\
(\text{since } \mathtt{k} z_{\mathrm{S}} > z_{\mathrm{S}}) \quad &\geq 1 + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + z_{\mathrm{S}}\, \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} \left(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\right) \\
(\text{since } z_{\mathrm{S}}\, \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} > -1) \quad &\geq 1 + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} - \left(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\right) \\
&= \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + 2\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \\
&\geq 0.
\end{aligned}$$

- We check (4.9b):

$$\det(\mathbf{R}^{\mathrm{DSFC}}_{\mathrm{MRBE}}) < 1 \quad \Leftrightarrow \quad \mathtt{k}\, z_{\mathrm{S}} < \frac{1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}}{\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} \left(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\right)} \quad \Leftrightarrow \quad \mathtt{k} > -\frac{1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}}{\left(1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\right) \left(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\right)}.$$

  For fixed $z_{\mathrm{S}}, z_{\mathrm{F}}$ this inequality will cease to hold when the system is tightly coupled, i.e., $\mathtt{k} \to -\infty$. However, the rightmost term is greater than -1 for any values $\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}, \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \in (0, 1]$. Consequently, the criterion (4.9b) is always satisfied for $-1 \leq \mathtt{k} < 1$.

- To check (4.9c):

$$\begin{aligned}
1 - \mathrm{tr}(\mathbf{R}^{\mathrm{DSFC}}_{\mathrm{MRBE}}) + \det(\mathbf{R}^{\mathrm{DSFC}}_{\mathrm{MRBE}}) &= 1 - \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} \left(\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathtt{k}\, z_{\mathrm{S}} \left(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\right)\right) \\
&= \left(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\right) \cdot \left(1 - \left(1 - \mathtt{k}\, z_{\mathrm{S}}\right) \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}}\right) \\
&= \left(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}\right) \cdot \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} \cdot \left(\mathtt{k} - 1\right) z_{\mathrm{S}} \\
&> 0.
\end{aligned}$$

  The last criterium (4.9c) is fulfilled as $\mathtt{k} < 1$.

□

The decoupled fastest-first approach with constant interpolation

Theorem 5 (Stability of decoupled fastest-first approach) *The decoupled fastest-first multirate backward Euler method* (4.17) *with constant interpolation of the slow variable, i.e.,* $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}} = \mathbf{y}_{\mathrm{S},n}$ *in* (4.17a)*, is unconditionally stable for* $-1 \leq \mathtt{k} \leq 1$*; it becomes unstable for* $\mathtt{k} \to -\infty$*.*

Proof: For the decoupled approach (4.15), the recursion (4.35) turns into

$$\begin{bmatrix} 1-z_{\mathrm{S}} & -w_{\mathrm{F}} \\ 0 & \left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{y}_{\mathrm{S},n+1} \\ \mathbf{y}_{\mathrm{F},n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \left(1-\left(1-\frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}}\right) & \frac{w_{\mathrm{S}}}{z_{\mathrm{F}}} \quad 1 \end{bmatrix} \begin{bmatrix} \mathbf{y}_{\mathrm{S},n} \\ \mathbf{y}_{\mathrm{F},n} \end{bmatrix}, \quad (4.37)$$

which yields the recursion matrix:

$$\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}} = \begin{bmatrix} \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \frac{w_{\mathrm{S}} w_{\mathrm{F}}}{z_{\mathrm{F}}} \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} \left(\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} - 1\right) & \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} w_{\mathrm{F}} \\ \left(\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} - 1\right) \frac{w_{\mathrm{S}}}{z_{\mathrm{F}}} & \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} \end{bmatrix}.$$

We have:

$$\begin{aligned} \det\left(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}\right) &= \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} < 1, \\ \mathrm{tr}\left(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}\right) &= \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + \mathtt{k}\, z_{\mathrm{S}} \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} \left(\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} - 1\right). \end{aligned}$$

We start with the equivalence:

$$1 + \mathrm{tr}(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}) + \det(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}) > 0 \quad \Leftrightarrow \quad \mathtt{k} > -\frac{(1+\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})(1+\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})}{(1-\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}})(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})},$$

and note that the second inequality holds for $-1 \leq \mathtt{k} \leq 1$. However, for $\mathtt{k} \to -\infty$ we get instability. Using $\mathtt{k} < 1$ we obtain:

$$-\mathrm{tr}(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}) > -\mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} + z_{\mathrm{S}} \mathbf{R}^{\mathrm{S}}_{\mathrm{BE}} (1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) = -1 - \det(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}),$$

and consequently:

$$1 - \mathrm{tr}(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}) + \det(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}) = 1 - 1 - \det(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}) + \det(\mathbf{R}^{\mathrm{DFFC}}_{\mathrm{MRBE}}) = 0.$$

□

The coupled slowest-first approach with constant interpolation

Theorem 6 (Stability of coupled slowest-first approach) *The coupled slowest–first multirate backward Euler method* (4.19) *with constant interpolation of the slow variable at* t_{n+1}*, i.e.,* $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}} = \mathbf{y}_{\mathrm{S},n+1}$ *in* (4.19b)*, is unconditionally stable.*

Proof: The macro-step (4.19) reads

$$\begin{bmatrix} 1-z_{\rm S} & -w_{\rm F} \\ -w_{\rm S} & 1-z_{\rm F} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{y}_{{\rm S},n+1} \\ \mathbf{y}^*_{{\rm F},n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{{\rm S},n} \\ \mathbf{y}_{{\rm F},n} \end{bmatrix}, \tag{4.38}$$

which yields

$$\mathbf{y}_{{\rm S},n+1} = \frac{(1-z_{\rm F})\mathbf{y}_{{\rm S},n} + w_{\rm F}\mathbf{y}_{{\rm F},n}}{(1-z_{\rm S})(1-z_{\rm F}) - w_{\rm S}w_{\rm F}}.$$

The micro-step solution reads:

$$\mathbf{y}_{{\rm F},n+1} = (\mathbf{R}^{\rm F}_{\rm BE} - 1)\frac{w_{\rm S}}{z_{\rm F}}\mathbf{y}_{{\rm S},n+1} + \mathbf{R}^{\rm F}_{\rm BE}\mathbf{y}_{{\rm F},n}$$

We have that:

$$\begin{bmatrix} \mathbf{y}_{{\rm S},n+1} \\ \mathbf{y}_{{\rm F},n+1} \end{bmatrix} = \mathbf{R}^{\rm CSFC}_{\rm MRBE} \cdot \begin{bmatrix} \mathbf{y}_{{\rm S},n} \\ \mathbf{y}_{{\rm F},n} \end{bmatrix}, \quad \text{with}$$

$$\mathbf{R}^{\rm CSFC}_{\rm MRBE} = \begin{bmatrix} \frac{1-z_{\rm F}}{(1-z_{\rm S})(1-z_{\rm F})-w_{\rm S}w_{\rm F}} & \frac{w_{\rm F}}{(1-z_{\rm S})(1-z_{\rm F})-w_{\rm S}w_{\rm F}} \\ (\mathbf{R}^{\rm F}_{\rm BE}-1)\frac{w_{\rm S}}{z_{\rm F}}\frac{1-z_{\rm F}}{(1-z_{\rm S})(1-z_{\rm F})-w_{\rm S}w_{\rm F}} & \mathbf{R}^{\rm F}_{\rm BE} + (\mathbf{R}^{\rm F}_{\rm BE}-1)\frac{w_{\rm S}}{z_{\rm F}}\frac{w_{\rm F}}{(1-z_{\rm S})(1-z_{\rm F})-w_{\rm S}w_{\rm F}} \end{bmatrix},$$

and therefore:

$$\begin{aligned} \det(\mathbf{R}^{\rm CSFC}_{\rm MRBE}) &= \mathbf{R}^{\rm F}_{\rm BE}\frac{1-z_{\rm F}}{(1-z_{\rm S})(1-z_{\rm F})-w_{\rm S}w_{\rm F}} = \mathbf{R}^{\rm F}_{\rm BE}\frac{1-z_{\rm F}}{1-z_{\rm S}-z_{\rm F}+(1-\mathtt{k})z_{\rm S}z_{\rm F}}, \\ {\rm tr}(\mathbf{R}^{\rm CSFC}_{\rm MRBE}) &= \mathbf{R}^{\rm F}_{\rm BE} + \frac{1-z_{\rm F}+(\mathbf{R}^{\rm F}_{\rm BE}-1)w_{\rm F}w_{\rm S}/z_{\rm F}}{(1-z_{\rm S})(1-z_{\rm F})-w_{\rm S}w_{\rm F}} = \mathbf{R}^{\rm F}_{\rm BE} + \frac{1-z_{\rm F}+(\mathbf{R}^{\rm F}_{\rm BE}-1)w_{\rm F}w_{\rm S}/z_{\rm F}}{1-z_{\rm S}-z_{\rm F}+(1-\mathtt{k})z_{\rm S}z_{\rm F}}. \end{aligned}$$

The three Hurwitz criteria can be verified as follows. The estimate

$$(1-z_{\rm S})(1-z_{\rm F}) - w_{\rm S}w_{\rm F} > 1-z_{\rm S}-z_{\rm F} > 1-z_{\rm F}$$

directly implies $\det(\mathbf{R}^{\rm CSFC}_{\rm MRBE}) < 1$, as $0 < \mathbf{R}^{\rm F}_{\rm BE} < 1$ holds. In addition, we get $\det(\mathbf{R}^{\rm CSFC}_{\rm MRBE}) > 0$. To check the first Hirwitz criterion, we see that

$$\begin{aligned} 1 + {\rm tr}(\mathbf{R}^{\rm CSFC}_{\rm MRBE}) + \det(\mathbf{R}^{\rm CSFC}_{\rm MRBE}) & \\ &= \frac{(2-2z_{\rm F}-z_{\rm S}+(1-\mathtt{k})z_{\rm S}z_{\rm F})(1+\mathbf{R}^{\rm F}_{\rm BE}) + (\mathbf{R}^{\rm F}_{\rm BE}-1)\mathtt{k}z_{\rm S}}{(1-z_{\rm S})(1-z_{\rm F})-w_{\rm S}w_{\rm F}} \\ &> \frac{(\mathbf{R}^{\rm F}_{\rm BE}-1)\mathtt{k}z_{\rm S}}{(1-z_{\rm S})(1-z_{\rm F})-w_{\rm S}w_{\rm F}} > 0 \end{aligned}$$

holds for $\mathtt{k} \geq 0$. To verify the criterion for $k < 0$, we only have to show that the nominator

$$(2 - 2z_{\mathrm{F}} - z_{\mathrm{S}} + (1 - \mathtt{k})z_{\mathrm{S}}z_{\mathrm{F}})(1 + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) + (\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} - 1)\mathtt{k}z_{\mathrm{S}}$$

is positive, which is equivalent to

$$\mathtt{k} < \underbrace{\frac{(2 - 2z_{\mathrm{F}} - z_{\mathrm{S}} + z_{\mathrm{S}}z_{\mathrm{S}})(1 + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})}{z_{\mathrm{S}}z_{\mathrm{F}}(1 + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) + z_{\mathrm{S}}(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})}}_{> 0}$$

provided that $z_{\mathrm{S}}z_{\mathrm{F}}(1+\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})+z_{\mathrm{S}}(1-\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}})$ is positive. This can be seen from

$$z_{\mathrm{S}}z_{\mathrm{F}}(1 + \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) + z_{\mathrm{S}}(1 - \mathbf{R}^{\mathrm{F}}_{\mathrm{BE}}) > 0 \Rightarrow \left(1 - \frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}} (z_{\mathrm{F}} + 1) < 1 - z_{\mathrm{F}}.$$

The latter inequality holds, since the left-hand side is monotonically increasing for $z_{\mathrm{F}} < 0$. Hence we have

$$\left(1 - \frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}} (z_{\mathrm{F}} + 1) < \left(\left(1 - \frac{z_{\mathrm{F}}}{\mathtt{m}}\right)^{\mathtt{m}} (z_{\mathrm{F}} + 1)\right)\Bigg|_{z_{\mathrm{F}}=0} = 1 < 1 - z_{\mathrm{F}}.$$

For the third Hurwitz criterium we check the expression:

$$1 - \mathrm{tr}(\mathbf{R}^{\mathrm{CSFC}}_{\mathrm{MRBE}}) + \det(\mathbf{R}^{\mathrm{CSFC}}_{\mathrm{MRBE}}) \;=\; (1 - z_{\mathrm{F}})\frac{z_{\mathrm{S}}(1 - \mathtt{k})(\mathbf{R}^{\mathrm{F}}_{\mathrm{BE}} - 1)}{(1 - z_{\mathrm{S}})(1 - z_{\mathrm{F}}) - w_{\mathrm{S}}w_{\mathrm{F}}},$$

which is positive as both nominator and denominator are positive.□

The coupled first-step approach with constant interpolation

Theorem 7 (Stability of coupled first-step approach) *The coupled first-step multirate backward Euler method* (4.21) *with constant interpolation of the slow variable at t_{n+1}, i.e., $\mathbf{y}_{\mathrm{S},n+(\ell+1)/\mathtt{m}} = \mathbf{y}_{\mathrm{S},n+1}$ in* (4.21b)*, is unconditionally stable.*

Proof: The macro-step (4.21) reads

$$\begin{bmatrix} 1 - z_{\mathrm{S}} & -w_{\mathrm{F}} \\ -\frac{w_{\mathrm{S}}}{\mathtt{m}} & 1 - \frac{z_{\mathrm{F}}}{\mathtt{m}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{y}_{\mathrm{S},n+1} \\ \mathbf{y}_{\mathrm{F},n+1/\mathtt{m}} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{\mathrm{S},n} \\ \mathbf{y}_{\mathrm{F},n} \end{bmatrix}, \tag{4.39}$$

which yields

$$\begin{aligned}\mathbf{y}_{\mathrm{S},n+1} &= \frac{(1-\frac{z_\mathrm{F}}{\mathtt{m}})\mathbf{y}_{\mathrm{S},n} + w_\mathrm{F}\mathbf{y}_{\mathrm{F},n}}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}}) - \frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}},\\ \mathbf{y}_{\mathrm{F},n+1/\mathtt{m}} &= \frac{\frac{w_\mathrm{S}}{\mathtt{m}}\mathbf{y}_{\mathrm{S},n} + (1-z_\mathrm{S})\mathbf{y}_{\mathrm{F},n}}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}}) - \frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}}.\end{aligned}$$

The micro-step solution reads:

$$\begin{aligned}\mathbf{y}_{\mathrm{F},n+1} &= \mathbf{R}^\mathrm{F}_\mathrm{BE}\left(1-\frac{z_\mathrm{F}}{\mathtt{m}}\right)\mathbf{y}_{\mathrm{F},n+1/\mathtt{m}} + \sum_{l=1}^{m-1}\left(1-\frac{z_\mathrm{F}}{\mathtt{m}}\right)^{-l}\frac{w_\mathrm{S}}{\mathtt{m}}\mathbf{y}_{\mathrm{S},n+1}\\ &= \mathbf{R}^\mathrm{F}_\mathrm{BE}\frac{(1-\frac{z_\mathrm{F}}{\mathtt{m}})(1-z_\mathrm{S})\mathbf{y}_{\mathrm{F},n}}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}}) - \frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}} + \mathbf{R}^\mathrm{F}_\mathrm{BE}\frac{(1-\frac{z_\mathrm{F}}{\mathtt{m}})\frac{w_\mathrm{S}}{\mathtt{m}}\mathbf{y}_{\mathrm{S},n}}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}}) - \frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}}\\ &\quad + \left(\mathbf{R}^\mathrm{F}_\mathrm{BE}\left(1-\frac{z_\mathrm{F}}{\mathtt{m}}\right)-1\right)\frac{w_\mathrm{S}}{z_\mathrm{F}}\mathbf{y}_{\mathrm{S},n+1}\end{aligned}$$

We have:

$$\begin{bmatrix}\mathbf{y}_{\mathrm{S},n+1}\\ \mathbf{y}_{\mathrm{F},n+1}\end{bmatrix} = \begin{bmatrix}\frac{1-\frac{z_\mathrm{F}}{\mathtt{m}}}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}})-\frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}} & \frac{w_\mathrm{F}}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}})-\frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}}\\ \frac{(1-\frac{z_\mathrm{F}}{\mathtt{m}})(\mathbf{R}^\mathrm{F}_\mathrm{BE}\frac{w_\mathrm{S}}{\mathtt{m}}+(\mathbf{R}^\mathrm{F}_\mathrm{BE}(1-\frac{z_\mathrm{F}}{\mathtt{m}})-1)\frac{w_\mathrm{S}}{z_\mathrm{F}})}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}})-\frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}} & \frac{\mathbf{R}^\mathrm{F}_\mathrm{BE}(1-\frac{z_\mathrm{F}}{\mathtt{m}})((1-z_\mathrm{S})+\frac{w_\mathrm{S}w_\mathrm{F}}{z_\mathrm{F}})-\frac{w_\mathrm{S}w_\mathrm{F}}{z_\mathrm{F}}}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}})-\frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{y}_{\mathrm{S},n}\\ \mathbf{y}_{\mathrm{F},n}\end{bmatrix},$$

and therefore

$$\begin{aligned}\det(\mathbf{R}^\mathrm{C1C}_\mathrm{MRBE}) &= \mathbf{R}^\mathrm{F}_\mathrm{BE}\frac{1-\frac{z_\mathrm{F}}{\mathtt{m}}}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}}) - \frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}},\\ \mathrm{tr}(\mathbf{R}^\mathrm{C1C}_\mathrm{MRBE}) &= \mathbf{R}^\mathrm{F}_\mathrm{BE} + \frac{(1-\frac{z_\mathrm{F}}{\mathtt{m}}) + (\mathbf{R}^\mathrm{F}_\mathrm{BE}-1)\frac{w_\mathrm{S}w_\mathrm{F}}{z_\mathrm{F}}}{(1-z_\mathrm{S})(1-\frac{z_\mathrm{F}}{\mathtt{m}}) - \frac{w_\mathrm{S}w_\mathrm{F}}{\mathtt{m}}}.\end{aligned}$$

One notes that both quantities are equivalent to the corresponding quantities of the coupled slowest-first approach, if z_F and w_S are replaced by $z_\mathrm{F}/\mathtt{m}$ and $w_\mathrm{S}/\mathtt{m}$, respectively. Consequently, the same conclusions hold. □