ORBIT 14
The propagator

A. Milani et al.
Dipmat, Pisa, Italy

Contents

1 Propagator - Interpolator

1.1 Propagation and interpolation

1.1.1 Introduction

The propagator-interpolator has the purpose of passing the state y_1 at time t_1 to the state y_2 at time t_2, solving the equation of motion and also the variation equation whenever the state includes some partial derivatives. Since the numerical integration takes place in steps, of variable length only with some difficulties, and since the time $t_2 - t_1$ will not equal, in general, an integer number of steps, the function of the propagator, working with constant step or rarely variable, is distinguished from the function of the interpolator, working on arbitrary fractions of the step.
Nonetheless, beyond this functional distinction, propagator and interpolator have to be in essence application of the same algorithm, to guarantee homogeneity of precision in the two functions.

1.1.2

The structure of the model is complicated by the presence of partly conflicting needs, that do no allow an univoque choice of a method, but require an hybrid structure in which co-exist:

- an implicit single step algorithm of the Runge-Kutta type;
- a multi-step algorithm of the Adams-Cowell type.

The reasons for this complex choice can be summarized as follows:

- the multi-step methods are not self-starting, therefore a RK is anyway necessary as initializator;
- the multi-step methods are much more economic in terms of CPU, due to the lower number of evaluation of the complex acceleration model;
- the implicit RK methods, an particularly some of them (derived from the Gaussian quadrature formulas), are much more stable and therefore usable also with very long steps and on long orbital arcs with less risk of dangerous numerical errors accumulations.

1.1.3

The procedure used is in essence the following: the first step is always executed with an implicit RK method, and similarly the following (with the help of an hybrid procedure) until the number of steps needed to start the multi-step. At this point the multi-step can take place, or the RK can be still used to control the stability. At the moment of going ahead with the interpolation, this is executed with a multi-step interpolator weather the RK or the multi-step were being used as the propagator.

1.2 Implicit Runge-Kutta initializator

If the orbit differential equation, reduced to the first order by assuming $y = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$, $\dot{x} = \frac{dx}{dt}$ is:

$$\dot{y} = f(t, y) \tag{1}$$

the RK algorithm for the values of $y(t)$ at intervals of amplitude h is the following: if Y_n is an approximate value of the solution $y(t)$ for $t = t_n = nh$, the approximate solution Y_{n+1} is calculated in the following way:

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_j k^{(j)} \tag{2}$$

where the $k^{(j)}$ are the solutions of the fixed point equation:

$$k^{(j)} = f\left(t_n + hc_j, y_n + h \sum_{\ell=1}^{s} a_{j\ell} k^{(\ell)}\right) \tag{3}$$

It the matrix $a_{j\ell}$ is sub-diagonal, Eq.(3) is solved by substitution and the method is called explicit; otherwise a solution must by sought by an iterative method. Every method is characterized by the number of steps s and by the coefficients $a_{j\ell}, c_j, b_j$ $(\ell, j = 1, \ldots, s)$ [5]

1.2.1

Since the coefficients a, b, c are solutions of complex non-linear equations, there is no algorithm that allows the creation of RK of arbitrary orders and of explicit type. Nonetheless, for some classes of implicit methods this algorithm exists, and between these there are the methods based on the Gaussian quadrature formulas. They are obtained from the approximation of the integral:

$$\int_{t_n}^{t_{n+1}} f(t, y(t))dt \simeq h \sum_{j=1}^{s} b_j f(t_n + hc_j, y(t_n + hc_j)) \tag{4}$$

by the integral of a polynomial of degree $2s - 1$; therefore the coefficients b_j are solutions of the equations:

$$\sum_{j=1}^{s} b_j c_j^{k-1} = \frac{1}{k} \qquad k = 1, \ldots, 2s \tag{5}$$

while, if $y_n + h \sum_{\ell=1}^{s} a_{j\ell} k^{(\ell)}$ has to be an approximation, valid for a degree s polynomial, to the value of $y(t_n + c_j h)$, the coefficients a_{jk} satisfy the relation:

$$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k} \qquad k, i = 1, \ldots, s\ . \tag{6}$$

Combining Eqs.(5) and (6) we have $s^2 + 2s$ equations in $s^2 + 2s$ unknowns a_{ij}, b_i, c_i; these equations are linear in a_{ij}, b_i and non-linear only in c_j. Now the explicit solution can be obtained by first Eq.(5) as a system of linear equations in the in the variables b_j overdetermined. The rank conditions and the conditions of the Rouché-Capelli theorem translates then into conditions on the matrix of the coefficients c_j^{k-1}. Then it can be shown [2] that, in order to satisfy these conditions, the coefficients c_j must be the roots of the Legendre polynomial of degree s (referred to the interval $[0, 1]$):

$$P_s(2c_j - 1) = 0 \qquad j = 1, \ldots, s \tag{7}$$

At this point the a_{jk}, b_j can be found from the inversion of the Van der Monde matrix $\left[c_j^{k-1}\right]$ $(j, k = 1, ..., s)$ [4].

1.2.2

The noticeable properties of this type of implicit RK methods, deduced from the Gaussian quadrature formulas, are three:

- the order of the method is $2s$, that is if $y_n = (nh)$, the difference between y_{n+1} given by Eq.(1) and $y(nh + h)$ (local truncation error) is of the order h^{2s+1}. No RK method can have higher order and, actually the explicit methods have very low orders ($\leq s$) [4].
- the method is A-stable, i.e., if applied to any equation of the type $\dot{y} = \lambda y$ with arbitrary step h, the results is convergent to zero with $n \to \infty$ if $Re(\lambda) < 0$. Therefore there are no stability problems in the choice of the step, that may be limited by other considerations (truncation, convergence of the iterations in Eq.(3)) [5].
- the method has a unique non-linear stability form, that can be defined as follows: if y_n, z_n are two different numerical solutions of Eq.(1) obtained from two different initial conditions $y_0 \neq z_0$, then, given $\Delta = |y - z|^2$ and applying the same method to the differential equation for Δ:
$$\dot{\Delta} = 2 < f(y) - f(z), y - z > ,$$
it results for the numerical solution (neglecting the rounding off) with $\Delta_0 = |y_0 - z_0|^2$:
$$\Delta_n = |y_n - z_n|$$
that is the numerical solution of the distance equation is the distance of the numerical solutions [3], [1].

1.2.3

The problems encountered in the implementation of an algorithm such as the one described above are not those related to the calculation of the coefficients, for whom Eq.(7), and then Eqs.(5)–(6), are solved with no difficulties for every s of any practical use, but in the implementation of the solution of the non-linear Eq.(3).

Since Eq.(3) is a fixed point equation in $6s$ variables, it is possible to look for the solution by successive approximations. If f is lipschitzian with Lipschitz constant L, the contraction constant of the fixed point equation (3) is $L\|4\|$, where $\|a\| = \max_{\mathrm{i,j}} \mathrm{a_{ij}}$; in practice $\|a\|$ is comprised between $\frac{1}{2}$ and 1; therefore if h is small enough (with respect to the frequencies typical of the second member) there is a geometric convergence fast enough [4]. Nonetheless, in practice a faster method is the Gauss-Siedel one, for which the $k^{(j)}$ are updated one by one instead of all at once. As an alternative, or in sequence, to speed up the convergence a Newton-type method can be applied to the Eq.(3).
Nonetheless even if the iterative method used to solve Eq.(3) is convergent, it turns out to be very expensive in terms of CPU time since every iteration requires the calculation of the complex second member function f, for s times. For this reason the capability of the second member to execute reduced calculations, by using again the computations for the smaller perturbations done in the previous iterations, is exploited.

1.2.4

Another procedure, used here for the first time, is that of choosing with attention the initial values from which to start for the iterations of the $k^{(j)}$ in Eq.(3). Instead of putting the $k^{(j)}$ equal to a standard value at the first iteration (e.g., $k^{(j)} = 0$), we use an interpolator predicting, with truncation error of the order h^{s+1}, the $k^{(j)}$ of every step starting from the values at convergence of the $k^{(j)}$ of the previous step.
This procedure, transforming the R-K in a hybrid method, allows the reduction of the number of iterations in the computation of the $k^{(j)}$ to about one half in most of the cases. Nonetheless this procedure can be used only if the step h is not changing [1].
The interpolation formula that is used can be written as:

$$k^{(j)} = \sum_{i=1}^{s} \tilde{a}_{ij} \overline{k}^{(i)} \quad j = 1, \dots, s \tag{8}$$

where $\overline{k}^{(i)}$ are the previous step values and the coefficients a are calculated solving the system:

$$\sum_{i=1}^{s} \tilde{a}_{ij} c_j^{k-1} = (1 + c_i)^{k-1} \quad j = 1, \dots, s\,. \tag{9}$$

1.2.5

The adoption of these procedure (together with those used for the variation equation) allows to keep the CPU time of a completely implicit method ($a_{ij} \neq 0, \forall i, j$) within acceptable limits. Nonetheless the cost is higher than the one of the multi-step, so that the R-K has in any case to be used only as an initializator or in the case where a stability control is needed.
As an initializator the method used here is very good, overcomes the limitations and avoids the complications of the methods used in other programs (explicit R-K or iterative starters), also since it can have any order $2s$ (while the known explicit methods are of order ≤ 8).

1.3 Multi-step method

To integrate the orbit differential equation with a multi-step method, it is convenient to take into account the fact that is is of second order, that is to use (instead of the form reduced to the first order 1), the form:

$$\ddot{x} = g(t, x, \dot{x})\,. \tag{10}$$

A multi-step method will describe the approximation x_{n+1} for $x(nh + h)$ as a function of the previous steps with a formula of the explicit type, that is a Cowell predictor :

$$x_{n+1} = x_n + h\dot{x}_n + h^2 \sum_{j=0}^{m} \beta_j^{(m)} g_{n-j} \tag{11}$$

where g_{n-j} is the value of the second member j steps before, i.e.:

$$g_{n-j} = g\left((n-j)h, x_{n-j}, \dot{x}_{n-j}\right) \tag{12}$$

Coupled to the predictor (11), another predictor for first order equations (an Adams one) will compute the value of the derivative:

$$\dot{x}_{n+} = \dot{x}_n + h \sum_{j=0}^{m} \alpha_j^{(m)} g_{n-j} \tag{13}$$

At this point $g_{n+1} = g\left(nh, x_{n+1}, \dot{x}_{n+1}\right)$ can be computed and a corrector may be applied, i.e., an implicit formula:

$$x_{n+1} = x_n + h\dot{x}_n + \frac{1}{2}h^2 g_n + h\overline{\beta}_{-1}^{(m)} g_{n+1} + h^2 \sum_{j=0}^{m} \overline{\beta}_j^{(m)} g_{n-j} \tag{14}$$

$$\dot{x}_{n+1} = \dot{x}_n + h\overline{\alpha}_{-1}^{(m)} g_{n+1} + h \sum_{j=0}^{m} \overline{\alpha}_j^{(m)} g_{n-j}\ . \tag{15}$$

If the difference between the predictor (11) and (13) and the corrector obtained from this with Eqs.(14)–(15) is too large, the corrector can be iterated more times, eventually using a reduced second member calculation. Therefore the multi-step method to be used is entirely defined by:

- the order $m + 3$
- the predictor coefficients α_j, β_j $\quad j = 1, \ldots, m$
- the corrector coefficients $\overline{\alpha}_j$, $\overline{\beta}_j$ $\quad j = -1, 0, 1, \ldots, m$
- the sequence of the iterations predictor–correctors with complete or reduced calculations.

1.3.1

The calculation of the coefficients of the multi-step methods must be executed in such a way that the order k is the highest possible. That is, if $x_n = x(nh)$, x_{n+1} calculated with Eqs.(11), (13), (14), (15) has to differ from $x(nh+h)$ by a term containing h^{k+1} with $k+1$ as large as possible. Therefore it is required that the formulas are exact if applied to polynomial of degree $\leq k$; the local truncation error is therefore the rest of the Taylor formula, i.e., $\frac{h^{k+1}}{(k+1)!}$, multiplied by an appropriate $k+1$ derivative.
The most efficient way to compute the coefficient is to get them with the formal operator series method: if we define the following operators on an appropriate space of functions $f(t)$, given a fixed step h:

$$\begin{aligned} E^s f(t) &= f(t+sh) \\ Df(t) &= f'(t) \\ \nabla f(t) = f(t) - f(t-h) I f(t) &= f(t) \end{aligned} \tag{16}$$

they form a commutative algebra on $\mathbb{R}$, on whose surroundings the following relations hold:

$$\nabla = I - E^{-1} \ , \ E^s = (I-\nabla)^{-s} \ \ \text{(operator difference prime)} \tag{17}$$

$$\nabla^j = (I - E^{-1})^j = \sum_{k=0}^{j} \left(\begin{array}{c} j \\ k \end{array} \right) (-1)^k E^{-k} \ \ \text{(Newton binomial)} \tag{18}$$

$$\nabla^j E^1 = E^1 (I - E^{-1}) \nabla^{j-1} = \nabla^{j-1} E^1 - \nabla^{j-1} \ \ \text{(updating of the j} - \text{th difference operators)} \tag{19}$$

$$E^1 = \exp(hd) = I + hD + \frac{h^2}{2} D^2 + \dots \ \ (\text{Taylor}'\text{s formula}) \ . \tag{20}$$

Since for the formal series algebra hold the same rules of composition, multiplication, inversion, etc., as for convergent numerical series, it is enough to formally "invert" Eq.(20) using the logarithm series:

$$\begin{aligned} hD = \log(E^1) = \log(I-\nabla)^{-1} = -\log(I-\nabla) = \\ \nabla^1 + \frac{1}{2}\nabla^2 + \frac{1}{3}\nabla^3 + \dots \end{aligned} \tag{21}$$

from this the Adams multi-step method is obtained by setting:

$$E^s = E^s \left[-\log(I-\nabla) \right]^{-1} hD = (I-\nabla)^{-s} \left[-\log(I-\nabla) \right]^{-1} hD \ . \tag{22}$$

The expression in the last member of Eq.(22) is evaluated using the convergent numerical series (of Laurent):

$$\phi(x) = \frac{(1-x)^{-s}}{-\log(1-x)} = \frac{1}{x} \sum_{n=0}^{+\infty} \gamma'_n(s) x^n \ . \tag{23}$$

The coefficients of the series (23) are calculated with the normal procedure for the "undetermined coefficients" and of the " product according to Cauchy" used for the functions of one variable (either complex or real):

$$-\log(1-x) = \sum_{j=1}^{+\infty} \frac{x^j}{j} \, [-\log(1-x)]^{-1} = \frac{1}{x}\sum_{n=0}^{+\infty} \gamma'_n(0) x^n$$

$$1 = \frac{-\log(1-x)}{-\log(1-x)} = \frac{1}{x}\left(\sum_{j=1}^{+\infty}\frac{x^j}{j}\right)\left(\sum_{n=0}^{+\infty}\gamma'_n(0)x^n\right) = \frac{1}{x}\left(\sum_{k=1}^{+\infty} x^k\right)\left(\sum_{j+n=k}\frac{\gamma'_n(0)}{j}\right)$$

from which, by executing the Cauchy product and solving the recurrent equations:

$$\gamma'_k(0) = \sum_{j=0}^{k-1}\frac{\gamma'_j(0)}{k-j+1} \qquad \gamma'_k(0) = 1 \tag{24}$$

to find $\gamma'_n(s)$ the further Cauchy product for $(1-x)^{-s}$ is executed:

$$(1-x)^{-s} = \sum_{j=0}^{+\infty}\binom{-s}{j}(-1)^j x^j$$

$$\frac{(1-x)^{-s}}{-\log(1-x)} = \frac{1}{x}\left(\sum_{n=0}^{+\infty}\gamma'_n(0)x^n\right)\left(\sum_{j=0}^{+\infty}\binom{-s}{j}(-1)^j x^j\right) = \\ \frac{1}{x}\left(\sum_{k=0}^{+\infty}x^k\right)\left(\sum_{j+n=k}\gamma'_n(0)\binom{-s}{j}(-1)^j\right)$$

where the generalized binomial coefficients are defined by:

$$\binom{-s}{j} = \frac{(-s)(-s-1)(-s-2)\dots(-s-j+1)}{j!} = \frac{(-s-j+1)}{j}\binom{-s}{j-1} \\ \binom{-s}{0} = 1 \tag{25}$$

so that the coefficients of Eq.(23) are all computable as:

$$\gamma'_n(s) = \sum_{j=0}^{n}\binom{-s}{n-j}(-1)^{n-j}\gamma'_j(0) \tag{26}$$

in particular:

$$\gamma'_n(1) = \sum_{j=0}^{n}\gamma'_j(0)\ . \tag{27}$$

The Eq.(22) can be explicitated, in terms of the differential operator, as:

$$E^s = \nabla^{-1} hD + \sum_{j=0}^{+\infty} \gamma'_{j+1}(s) \nabla^j hD \; . \tag{28}$$

Since the operator ∇^{-1}, beyond being defined up to a constant function, is not explicitly computable, it is convenient to eliminate it from Eq.(28) using two equations with different s. E.g., from Eq.(28) with $s = 0$ we have:

$$\nabla^{-1} hD = E^0 + \sum_{j=0}^{+\infty} \gamma'_{j+1}(0) \nabla^j hD \tag{29}$$

and substituting into Eq.(28):

$$E^s = E^0 + \sum_{j=0}^{+\infty} \left[\gamma'_{j+1}(s) - \gamma'_{j+1}(0)\right] \nabla^j hD \tag{30}$$

that represents the predictor (for $s = 1$) or the Adams interpolator (for $s \neq 1$).
To reduce ourselves to the ordered form of Eq.(13) it is necessary first to truncate the infinite series, but imposing that the remainder is of the desired order. To do this is sufficient to consider that the j-th differences contain the j-th derivative so that:

$$\nabla^j t^k = 0 \quad \text{if j > k} \tag{31}$$

and therefore if we want that the formulas of the kind of Eq.(13) are exact for polynomials $x(t)$ of degree $\leq m + 2$, we will have $g(t, x(t), \dot{x}(t))$ polynomial of degree m and therefore:

$$\nabla^{m+1} g = 0 \qquad \text{(or better o(h}^\text{m}\text{))} \; . \tag{32}$$

Than the development of ∇^j (Newton's binomial of Eq.(18)) can be substituted in Eq.(13) truncated, by finding a finite sum:

$$\begin{aligned} E^s = E^0 + \sum_{j=0}^{m} \left[\gamma'_{j+1}(s) - \gamma'_{j+1}(0)\right] \nabla^j hD + o(h^m) = \\ E^0 + \sum_{j=0}^{m} \left[\gamma'_{j+1}(s) - \gamma'_{j+1}(0)\right] \sum_{k=0}^{j} \begin{pmatrix} j \\ k \end{pmatrix} (-1)^k E^{-k} hD = \\ E^0 + \sum_{k=0} m\alpha^{(m)} E^{-k} hD \qquad \text{(for s = 1)} \end{aligned} \tag{33}$$

from which, for $s = 1$, Eq.(13) with:

$$\alpha_k^{(m)} = (-1)^k \sum_{j=k}^{m} \left[\gamma'_{j+1}(1) - \gamma'_{j+1}(0)\right] \begin{pmatrix} j \\ k \end{pmatrix} \qquad k = 0, 1, \ldots, m \tag{34}$$

1.3.2

The formula of the corrector, Eq.(15) is obtained in the same way, but applying Eq.(28) for $s = 0$ step later. That is from:

$$E^1 = E^0 E^1 = \nabla^{-1} E^1 hD + \sum_{j=0}^{+\infty} \gamma'_{j+1}(0) \nabla^j E^1 hD \tag{35}$$

the first term in the second member is simplified using the formula:

$$\nabla^{-1} E^1 = \nabla^{-1} + E^1 \tag{36}$$

(formally analogous to Eq.(19); to prove this it can be verified $\nabla \left[\nabla^{-1} E^1 - E^1\right] = E^0$). Then $\nabla^{-1} E^1 hD = \nabla^{-1} hD + E^1 hD$, and using for $\nabla^{-1} hD$ the Eq.(29):

$$E^1 = E^0 + E^1 hD + \sum_{j=0}^{+\infty} \gamma'_{j+1}(0) \left[\nabla^j E^1 - \nabla^j\right] hD \ . \tag{37}$$

Now using again Eq.(19) it is obtained:

$$E^1 = E^0 + E^1 hD + \sum_{j=0} \gamma'_{j+1}(0) \nabla^{j+1} E^1 hD \ . \tag{38}$$

At this point the series is truncated deciding the order. Since, as usual, it is desired that the corrector is of one order higher than the corresponding predictor, the exact equation for polynomial of degree $\leq m + 3$ is sought and therefore it is supposed that:

$$\nabla^{m+2} g = 0 \qquad (\text{or better o}(\text{h}^{\text{m}+1})) \tag{39}$$

and therefore Eq.(18) is substituted into Eq.(38) truncated, finding:

$$E^1 = E^0 + E^1 hD + \sum_{j=0}^{+\infty} \gamma'_{j+1}(0) \sum_{k=0}^{j+1} \left(\begin{array}{c} j+1 \\ k \end{array} \right) (-1)^k E^{-k+1} hD \tag{40}$$

from which, separating the terms with $k = 0$ (that are calculated in the point t_{n+1}) and exchanging the order of the sum:

$$E^1 = E^0 + \left(1 + \sum_{j=0}^{m} \gamma'_{j+1}(0) \right) E^1 hD + \sum_{\ell=0}^{m} \left[\sum_{j=\ell}^{m} \left(\begin{array}{c} j+1 \\ \ell+1 \end{array} \right) \gamma'_{j+1}(0) \right] (-1)^{\ell+1} E^{-\ell} hD =$$

$$E^0 + \overline{\alpha}_{-1}^{(m)} E^1 hD + \sum_{\ell=0}^{m} \overline{\alpha}_{\ell}^{(m)} E^{-\ell} hD \ . \tag{41}$$

Therefore the coefficient of the corrector in Eq.(15) are determined in the following way:

$$\overline{\alpha}_{-1}^{(m)} = 1 + \sum_{j=0}^{m} \gamma'_{j+1}(0) = \sum_{j=0}^{m+1} \gamma'_j(0) \tag{42}$$

$$\overline{\alpha}_{\ell}^{(m)} = \sum_{j=\ell}^{m} \left(\begin{array}{c} j+1 \\ \ell+1 \end{array} \right) \gamma'_{j+1}(0)(-1)^{\ell+1} \ . \tag{43}$$

The procedure for the Cowell method is analogous, starting from Eq.(21). I.e., we use:

$$E^s = (I - \nabla^{-s}) \left[-\log(I-\nabla) \right]^{-2} h^2 D^2 \ . \tag{44}$$

It follows that for $s = 0$:

$$E^0 = \left[-\log(I-\nabla) \right]^{-2} h^2 D^2 = \sum_{j=0}^{+\infty} \gamma''_j(0) \nabla^{j-2} h^2 D^2 \tag{45}$$

where the γ'' are computed using Laurent's series, computed doing Cauchy's product:

$$\left[\frac{1}{-\log(1-x)} \right]^2 = \frac{1}{x} \left(\sum_{m=0} \gamma'_m(0) x^m \right) =$$
$$= \frac{1}{x^2} \sum_{k=0}^{+\infty} x^k \sum_{n+m=k} \gamma'_n(0) \gamma'_m(0) = \frac{1}{x^2} \sum_{k=0}^{+\infty} \gamma''_k(0) x^k$$

from which:

$$\gamma''_k(0) = \sum_{n=0}^{k} \gamma'_n(0) \gamma'_{k-n}(0) \tag{46}$$

and in particular for the first coefficients:

$$\gamma''_0(0) = 1 \qquad \gamma''_1(0) = 2\gamma'_1(0)\gamma'_0(0) = -1 \tag{47}$$

from which, taking out the part with negative exponents:

$$E^0 = \nabla^{-2} h^2 D^2 - \nabla^{-1} h^2 D^2 + \sum_{j=0}^{+\infty} \gamma''_{j+2}(0) \nabla^j h^2 D^2 \ . \tag{48}$$

For the computation of E^s it is put, like for the Adam's case:

$$E^s = \sum_{j=0}^{+\infty} \gamma''_j(s) \nabla^{j-2} h^2 D^2 \tag{49}$$

and making again the Cauchy's product for $(1-x)^{-s}$ it is obtained:

$$\gamma''_n(s) = \sum_{j=0}^{n} \left(\begin{array}{c} -s \\ n-j \end{array} \right) (-1)^{n-j} \gamma''_j(0) \tag{50}$$

where the coefficients are again defined by Eq.(25), and in particular:

$$\gamma''_n(1) = \sum_{j=0}^{n} \gamma''_j(0) \ . \tag{51}$$

Note that $\gamma''_1(0) = 0$, that is for every s:

$$E^s = \nabla^{-2} h^2 D^2 + (s-1) \nabla^{-1} h^2 D^2 + \sum_{j=0}^{+\infty} \gamma''_{j+2}(s) \nabla^j h^2 D^2 \ . \tag{52}$$

The computation of the predictor or interpolator coefficients requires therefore the elimination of ∇^{-2} (which is moreover an operator not defined over polynomials of degree 0 and 1). For this purpose $\nabla^{-2} h^2 D^2$ is eliminated from the couple of Eqs.(48) and (52) with $s = 1$, so that, from Eq.(48):

$$\nabla^{-2} h^2 D^2 = E^0 + \nabla^{-1} h^2 D^2 + \sum_{j=0}^{+\infty} \gamma''_{j+2}(0) \nabla^j h^2 D^2 \ . \tag{53}$$

and substituting in Eq.(52):

$$E^s = E^0 + s \nabla^{-1} h^2 D^2 + \sum_{j=0}^{+\infty} \left[\gamma''_{j+2}(s) - \gamma''_{j+2}(0) \right] \nabla^j h^2 D^2 \ . \tag{54}$$

It is now necessary to eliminate $s \nabla^{-1} h^2 D^2$ obtaining it from the formula already used for Adams, i.e., Eq.(29) multiplied on the right hand side by hD:

$$E^s = E^0 + s \left[E^0 - \sum_{j=0}^{+\infty} \gamma'_{j+1}(0) \nabla^j hD \right] hD + \sum_{j=0}^{+\infty} \left[\gamma''_{j+2}(s) - \gamma''_{j+2}(0) \right] \nabla^j h^2 D^2 \tag{55}$$

from which, gathering:

$$E^s = E^0 + s E^0 hD + \sum_{j=0}^{+\infty} \left[\gamma''_{j+2}(s) - \gamma''_{j+2}(0) - s \gamma'_{j+1}(0) \right] \nabla^j h^2 D^2 \ . \tag{56}$$

At this point we supposed, as usual, $\nabla^{m+1} h^2 D^2 = 0$ (or better Eq.(32)) and therefore:

$$E^s = E^0 + s E^0 hD + \sum_{j=0}^{m} \left[\gamma''_{j+2}(s) - \gamma''_{j+2}(0) - s \gamma'_{j+1}(0) \right] \nabla^j h^2 D^2 =$$

$$= E^0 + s E^0 hD + \sum_{k=0}^{m} \left[\sum_{j=0}^{k} \left(\begin{array}{c} j \\ k \end{array} \right) (-1)^k \left\{ \gamma''_{j+2}(s) - \gamma''_{j+2}(0) - s \gamma'_{j+2}(0) \right\} \right] E^{-k} h^2 D^2 \tag{57}$$

that gives the predictor, for $s = 1$, with:

$$\beta_k^{(m)} = (-1)^k \sum_{j=k}^{m} \begin{pmatrix} j \\ k \end{pmatrix} \left[\gamma''_{j+2}(1) - \gamma''_{j+2}(0) - \gamma'_{j+1}(0)\right] \ . \tag{58}$$

The Cowell's corrector is obtained from the formula of E^0 computed one step later, i.e., from Eq.(48):

$$E^1 = E^0 E^1 = \nabla^{-2} h^2 D^2 E^1 - \nabla^{-1} h^2 D^2 E^1 + \sum_{j=0}^{+\infty} \gamma''_{j+2}(0) \nabla^j h^2 D^2 E^1 \ . \tag{59}$$

Now the formula similar to Eq.(36) is used:

$$\nabla^{-2} E^1 = \nabla^{-2} + \nabla^{-1} E^1 \tag{60}$$

using then Eq.(53):

$$\nabla^{-2} E^1 h^2 D^2 = \left(E^0 + \nabla^{-1} h^2 D^2 - \sum_{j=0}^{+\infty} \gamma''_{j+2}(0) \nabla^j h^2 D^2 \right) + \nabla^{-1} E^1 h^2 D^2 \tag{61}$$

therefore, since $\nabla^{-1} E^1 h^2 D^2$ cancels out:

$$E^1 = E^0 + \nabla^{-1} h^2 D^2 + \sum_{j=0}^{+\infty} \gamma''_{j+2}(0) \left[\nabla^j E^1 - \nabla^j\right] h^2 D^2 \ . \tag{62}$$

Using again Eq.(29), multiplied on the right hand side by hD:

$$E^1 = E^0 + E^0 hD + \sum_{j=0}^{+\infty} \gamma''_{j+2}(0) \left[\nabla^j E^1 - \nabla^j\right] h^2 D^2 - \sum_{j=0}^{+\infty} \gamma'_{j+2}(0) \nabla^j h^2 D^2 \ . \tag{63}$$

Using Eq.(19) the two terms in square brackets give $\nabla^{j+1} e^1$; gathering the similar terms:

$$E^1 = E^0 + E^0 hD - \gamma'_1(0) h^2 D^2 + \sum_{j=1}^{+\infty} \left[\gamma''_{j+1}(0) \nabla^j E^1 - \gamma'_{j+1}(0) \nabla^j\right] h^2 D^2 \tag{64}$$

the term between square brackets has to be computed with Eq.(19), from which:

$$a \nabla^j E^1 - b \nabla^j = b \nabla^{j+1} E^1 + (a - b) \nabla^j E^1$$

that is, in our case:

$$E^1 = E^0 + E^0 hD - \gamma'_1(0) h^2 D^2 + \sum_{j=1}^{+\infty} \gamma'_{j+1}(0) \nabla^{j+1} E^1 h^2 D^2 + \sum_{j=1}^{+\infty} \left[\gamma''_{j+1}(0) - \gamma'_{j+1}(0)\right] \nabla^j E^1 h^2 D^2 \tag{65}$$

and gathering the similar terms (separating the term with $j = 1$ which is different):

$$E^1 = E^0 + E^0 hD - \gamma_1'(0)h^2D^2 + [\gamma_2''(0) - \gamma_2'(0)]\, \nabla^j E^1 h^2 D^2 + \sum_{j=2}^{+\infty} \left[\gamma_{j+1}''(0) - \gamma_{j+1}'(0) + \gamma_j'(0)\right] \nabla^j E^1 h^2 D^2 \tag{66}$$

it is finally possible to go to the truncation, that assumes to be able to neglect $\nabla^{m+2}h^2D^2$, so that:

$$E^1 = E^0 hD - \gamma_1'(0)h^2D^2 + [\gamma_2''(0) - \gamma_2'(0)]\left[h^2D^2E^1 - h^2D^2\right] + + \sum_{j=2}^{+\infty} \gamma_{j+2}''(0) - \gamma_{j+2}'(0) + \gamma_j'(0) \sum_{k=0}^{j} \left(\begin{array}{c} j \\ k \end{array} \right) (-1)^k E^{-k+1} h^2 D^2 \ . \tag{67}$$

To go further it is convenient to put $\ell = j - 1$, $r = k - 1$; separating the terms with $k = 0$ (i.e., those that makes the corrector implicit):

$$\begin{aligned} E^1 &= E^0 + E^0 hD - \gamma_1'(0)h^2D^2 \\ &+ \left[\{\gamma_2''(0) - \gamma_2'(0)\} + \sum_{\ell=1}^{m} \left\{\gamma_{\ell+2}''(0) - \gamma_{\ell+2}'(0) + \gamma_{\ell+1}'(0)\right\} \right] E^1 h^2 D^2 + \\ &+ \sum_{r=0}^{m} (-1)^{r+1} \sum_{\ell=r}^{m} \left(\begin{array}{c} \ell+1 \\ r+1 \end{array} \right) \left\{\gamma_{\ell+2}''(0) - \gamma_{\ell+2}'(0) + \gamma_{\ell+1}'(0)\right\} E^{-r} h^2 D^2 \end{aligned} \tag{68}$$

which is exactly the formula that was looked for, i.e.,:

$$E^1 = E^0 + E^0 hD + \frac{1}{2} h^2 D^2 + \overline{\beta}_{-1}^{(m)} E^1 h^2 D^2 + \sum_{r=0}^{m} \overline{\beta}_r^{(m)} E^{-1} h^2 D^2 \tag{69}$$

where:

$$\overline{\beta}_{-1}^{(m)} = \gamma_2''(0) - \gamma_2'(0) + \sum_{\ell=1}^{m} \left\{\gamma_{\ell+2}''(0) - \gamma_{\ell+2}'(0) + \gamma_{\ell+1}'(0)\right\} \tag{70}$$

$$\overline{\beta}_r^{(m)} = (-1)^{r+1} \sum_{\ell=r}^{m} \left(\begin{array}{c} \ell+1 \\ r+1 \end{array} \right) \left\{\gamma_{\ell+2}''(0) - \gamma_{\ell+2}'(0) + \gamma_{\ell+1}'(0)\right\} \tag{71}$$

1.4 Multi-step interpolator

Eqs.(34) and (57) give, as we saw, the Adams and Cowell's interpolators, respectively. Nonetheless, while the coefficients to be used for $s = 1$ (predictor) are the same for every step, and therefore can be computed once for all with Eqs.(34), (58) (and in the same way for the corrector), the interpolator coefficients for varying s are polynomials in s, to be re-calculated at every occasion in which one wants to interpolate, since s is varying from case to case ($0 < s < 1$).

For simplicity it is used, as a base in the space of the polynomials in s, the one composed by the $\left(\begin{array}{c} -s \\ i \end{array}\right)$ (that is a polynomial of degree i). Then, starting again from Eq.(30) for Adams, and using the truncation of Eq.(32), the equation of the $\gamma'_j(s)$, Eq.(26), and Eq.(12) of Newton's binomial:

$$E^s = E^0 + \sum_{i=0}^{m} \left[\gamma'_{i+1}(s) - \gamma'_{i+1}(0)\right] \nabla^i hD = \tag{72}$$

$$E^0 + \sum_{i=0}^{m} \sum_{j=0}^{i} \left(\begin{array}{c} -s \\ i+1-j \end{array}\right) (-i)^{i+1-j} \gamma'_j(0) \sum_{k=0}^{i} (-i)^k \left(\begin{array}{c} i \\ k \end{array}\right) E^{-k} hD =$$

$$= E^0 + \sum_{\ell=1}^{m+1} \left(\begin{array}{c} -s \\ \ell \end{array}\right) \sum_{k=0}^{m} \left[\sum_{i=\max(k,\ell-1)}^{m} (-i)^{\ell+k} \left(\begin{array}{c} i \\ k \end{array}\right) \gamma'_{i+1-\ell}(0) \right] E^{-k} hD$$

where the term in square brackets is the coefficient d, dependent from the degree m and the indexes ℓ, k, that defines the interpolator as a bilinear form in the base polynomials $\left(\begin{array}{c} -s \\ \ell \end{array}\right)$ and in the previous values of the derivative $E^{-k}hD$:

$$d^{(m)}_{\ell k} = \sum_{i=\max(k,\ell-1)}^{m} (-i)^{\ell+k} \left(\begin{array}{c} i \\ k \end{array}\right) \gamma'_{i+1-\ell}(0) \; . \tag{73}$$

For Cowell one has to restart from Eq.(56), using the truncation of Eq.(32), Eq.(50) for the $\gamma''_j(s)$ and Eq.(18):

$$E^s = E^0 + sE^0 hD + \sum_{i=0}^{m} \left[\gamma''_{i+2}(s) - \gamma''_{i+2}(0) - s\gamma'_{i+1}(0)\right] \nabla^i h^2 D^2 =$$

$$= E^0 + sE^0 hD + \sum_{i=0}^{m} \left[\sum_{j=1}^{i+2} \left(\begin{array}{c} -s \\ i+2-j \end{array}\right) (-i)^{i+2-j} \gamma''_j(0) - \gamma'_{i+1}(0) \right] \nabla^i h^2 D^2 =$$

(putting $i + 2 - j = \ell$ and exchanging the indexes)

$$= E^0 + sE^0 hD + \sum_{i=0}^{m} \left[\sum_{j=1}^{i+2} \left(\begin{array}{c} -s \\ \ell \end{array}\right) (-i)\gamma''_{i+2-\ell}(0) - s\gamma'_{i+1}(0) \right] \sum_{k=0}^{i} (-1)^k \left(\begin{array}{c} i \\ k \end{array}\right) E^{-k} h^2 D^2 =$$

$$= E^0 + sE^0 hD + \sum_{k=0}^{m} \sum_{\ell=1}^{m+2} \sum_{i=\max(0,\ell-2,k)}^{m} \left[(-i)^{\ell-s} \gamma''_{i+2-\ell}(0) - s\gamma'_{i+1}(0)\right] (-1)^k \left(\begin{array}{c} i \\ k \end{array}\right) E^{-k} h^2 D^2 \tag{74}$$

Then separating the term of degree 1 in s:

$$E^s = E^0 + sE^0hD + \begin{pmatrix} -s \\ 1 \end{pmatrix} \sum_{k=0}^{m} (-1)^{k+1} \sum_{i=k}^{m} \begin{pmatrix} i \\ k \end{pmatrix} \left[\gamma''_{i+1}(0) - \gamma'_{i+1}(0)\right] E^{-k} h^2 D^2 +$$
$$\sum_{k=0}^{m} \left[\sum_{\ell=2}^{k+1} \left\{ (-1)^{k+2} \sum_{i=\max(k,\ell-2)}^{m} \begin{pmatrix} i \\ k \end{pmatrix} \gamma''_{i+2-\ell}(0) \right\} \begin{pmatrix} -s \\ \ell \end{pmatrix} \right] E^{-k} h^2 D^2 \qquad (75)$$

from which the interpolator coefficient is:

$$f^{(m)}_{\ell k} = (-1)^{k+2} \sum_{i=\max(k,\ell-2)}^{m} \begin{pmatrix} i \\ k \end{pmatrix} \gamma''_{i+2-\ell}(0) \qquad (76)$$

for $\ell = 2, \ldots, m+1$; for $\ell = 1$ instead:

$$f^{(m)}_{1k} = (-1)^{k+1} \sum_{i=k}^{m} \left[\gamma''_{i+1}(0) - \gamma'_{i+1}(0)\right] \; . \qquad (77)$$

Summarizing, the interpolators that are used are simply:

$$E^s = E^0 + \sum_{\ell=1}^{m+1} \begin{pmatrix} -s \\ \ell \end{pmatrix} \sum_{k=0}^{m} d^{(m)}_{\ell k} E^{-k} hD \qquad (78)$$

$$E^s = E^0 + sE^0hD + \sum_{\ell=1}^{m+1} \begin{pmatrix} -s \\ \ell \end{pmatrix} \sum_{k=0}^{m} f^{(m)}_{\ell k} E^{-k} h^2 D^2 \qquad (79)$$

where f and d are computed once for all, while the binomial coefficients $\begin{pmatrix} -s \\ \ell \end{pmatrix}$ are computed time by time.

Note: to save memory space, in the code the triple tensors $f^{(m)}_{\ell k}$ and $d^{(m)}_{\ell k}$, that are not null only for:

$$1 \leq \ell \leq m+1 \quad 0 \leq k \leq m$$

are memorized in a single three indexes tensor *cub* with the rule:

$$\begin{aligned} cub(j, m+2, \ell) &= d^{(m)}_{\ell\, j-1} \\ cub(j, \ell, m+2) &= f^{(m)}_{\ell\, j-1} \\ \ell = 1, \ldots, m+1 \qquad & j = 1, \ldots, m+1 \end{aligned} \qquad (80)$$

2 Numerical manipulation of the variation equations

Let's suppose that the initial conditions problem to be handled depends laso from some parameters μ:

$$\begin{aligned} \ddot{x} &= g(t, x, \dot{x}, \mu) \\ x(t_0) = x_0 \qquad & \dot{x}(t_o) = \dot{x}_0 \end{aligned} \tag{81}$$

or in the form reduced to first order with $y = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$:

$$\begin{aligned} \dot{y} &= f(t, y, \mu) \\ t(t_0) &= y_0 \ . \end{aligned} \tag{82}$$

Then the partial derivatives with respect to the initial conditions or to the parameters μ satisfy the following linear differential equations: in the case of the second order, of Eq.(81):

$$\frac{d^2}{dt^2}\left(\frac{\partial x}{\partial x_0}\right) = \frac{\partial g}{\partial x}\left(\frac{\partial x}{\partial x_0}\right) + \frac{\partial g}{\partial \dot{x}}\frac{d}{dt}\left(\frac{\partial x}{\partial x_0}\right) \qquad \frac{\partial x}{\partial \ddot{x}_0}(0) = I \qquad \frac{\partial \dot{x}}{\partial x_0}(0) = 0 \tag{83}$$

$$\frac{d^2}{dt^2}\left(\frac{\partial x}{\partial \dot{x}_0}\right) = \frac{\partial g}{\partial x}\left(\frac{\partial x}{\partial \dot{x}_0}\right) + \frac{\partial g}{\partial \dot{x}}\frac{d}{dt}\left(\frac{\partial x}{\partial \dot{x}_0}\right) \qquad \frac{\partial x}{\partial \dot{x}_0}(0) = I \qquad \frac{\partial \dot{x}}{\partial \dot{x}_0}(0) = 0 \tag{84}$$

$$\frac{d^2}{dt^2}\left(\frac{\partial x}{\partial \mu}\right) = \frac{\partial g}{\partial x}\left(\frac{\partial x}{\partial \mu}\right) + \frac{\partial g}{\partial \dot{x}}\frac{d}{dt}\left(\frac{\partial x}{\partial \mu}\right) + \frac{\partial g}{\partial \mu} \qquad \frac{\partial x}{\partial \mu}(0) = 0 \tag{85}$$

where the derivative commutation relation has been used:

$$\frac{d}{dt}\left(\frac{\partial x}{\partial z}\right) = \frac{\partial \dot{x}}{\partial z} \qquad z = x_0 \, , \dot{x}_0 \, , \mu \tag{86}$$

while in the reduced form at first order:

$$\frac{d}{dt}\left(\frac{\partial y}{\partial y_0}\right) = \frac{\partial f}{\partial y}\left(\frac{\partial f}{\partial y_0}\right) \qquad \frac{\partial y}{\partial y_0}(0) = I \tag{87}$$

$$\frac{d}{dt}\left(\frac{\partial y}{\partial \mu}\right) = \frac{\partial f}{\partial y}\left(\frac{\partial f}{\partial \mu}\right) + \frac{\partial f}{\partial \mu} \qquad \frac{\partial y}{\partial \mu}(0) = 0 \ . \tag{88}$$

All these variation equations have the characteristic to be linear in the partial derivatives (homogeneus in the case of the derivatives with respect to the initial conditions; non-homogeneus the others). While choosing a numerica integration method, it is essential to exploit these properties. Nonetheless it has to be noted that the coefficients of the linear equations are function not only of t, but of the solution $x(t)$ (or $y(t)$) of the equation of motion (81) or (82), and non-linear function of these arguments. Therefore what it is done is to first integrate one step of the equation of motion, and only after having choosen the next point x_n (or y_n), to execute

one step of the variation equautions exploiting their linearity in the other arguments to make explicits the implicit methods.
For the case reduced to the first order (82) the variation equations are written by noting with Y a column of partial derivatives with respect to one of the y or μ:

$$\dot{Y} = F(t)Y + G \tag{89}$$

with $F = \frac{\partial f}{\partial y}(y(t))$, $G = \partial f \partial \mu$ or 0, according to the cases of Eq.(87) or Eq.(88).
Then applying to Eq.(89) the Runge-Kutta formulas (2) and (3) it comes out:

$$Y_{n+1} = Y_n + h_j \sum_{j=1}^{s} b_j K^{(j)} \tag{90}$$

$$K^{(j)} = F(t_n + hc_j)Y_n + F(t_n = hc_j)h \sum_{\ell=1}^{s} a_j K^{(\ell)} + G(t_n + hc_j) \tag{91}$$

from which it can be seen that Eq.(91), differently from Eq.(3), is linear and can be solved by inverting the matrix of the system:

$$\sum_{\ell=1}^{s} \left[\delta_j - hF(t_n + hc_j)a_{j\ell}\right] K^{(\ell)}) = F(t_n + hc_j)Y_n + G(t_n + hc_j) \ . \tag{92}$$

Note that the system (92) has $s \times n$ unknown (if y has n components, in practice $n = 6$) $k_i^{(\ell})$ with $\ell = 1, \ldots, n$.
The noticeable thing in the system (92 is the it is linked to the system (3) by a complex relation: in fact let's try to solve the system (3) with an iterative method of the Newton-Raphson type. Let's write the system (3) in the form $\phi = 0$:

$$\phi_j(k) = k^{(j)} - f\left(t_n + hc_j, y_n + h \sum_{\ell=1}^{s} a_{j\ell} k^{(\ell)}\right) = 0 \tag{93}$$

then the derivative of ϕ with respect to $k^{(\ell)}$ is calculated in k:

$$\frac{\partial \phi_j(k)}{\partial k^{(\ell)}} = \delta_{J\ell} I - h \frac{\partial f}{\partial y}\left(t_n + hc_j, y_n + h_i \sum_{i=1}^{s} a_{ji} k^{(i)}\right) a_{j\ell} \tag{94}$$

therefore, at convergence in the k, the matrix to be inverted for the Newton-Raphson method in Eq.(3) for the k (relative to the equation of motion) is the same to be inverted to solve the problem (92) for the K in the variation equations. Therefore it is enough, at convergence of the k, to execute one N-R iteration to get the inverse matrix that gives the K, if multiplied by a second member given by Eq.(92). Of course the Newton method is not convenient to solve Eq.(3), since it also requires the computation of the partial derivatives $\frac{\partial f}{\partial y}$ and the inversion of

the matrix (94) that is of order $s \times n$. Nonetheless if many columns of partial derivatives have to be computed , the same computations are useful for the variation eqautions and the cost of the computations grows by a very small amount with the growing of the number of columns of partial derivatives.
The case of the multi-step method is quite similar: if the varation equation to be solved is:

$$\ddot{X} = A(t)X + B(t)\dot{X} + C(t) = D[t, X] \tag{95}$$

where X is a column of partial derivatives, $A = \frac{\partial g}{\partial x}$, $b = \frac{\partial g}{\partial \dot{x}}$, $c = \frac{\partial g}{\partial \mu}$ or 0. Then the equations of the corrector (14)–(15) are, applied to this case:

$$\begin{aligned} X_{n+1} &= X_n + h\dot{X}_n + \frac{1}{2}h^2\ddot{X}_n + \beta_{-1}^{(m)}h^2\ddot{X}_{n+1} + h^2\sum_{j=0}^{m}\beta_j^{(m)}\ddot{X}_{n-j} \\ \dot{X}_{n+1} &= \dot{X}_n + \alpha_{-1}^{(m)}h\ddot{X}_{n+1} + h\sum j = 0^m\alpha_j^{(m)}\ddot{X}_{n-1} \quad . \end{aligned} \tag{96}$$

Then, let's compute the second member in the next step, by using Eq.(96):

$$\begin{aligned} \ddot{X}_{n+1} = D\left[t_{n+1}, X_{n+1}\right] &= A(t_{n+1})X_{n+1} + B(t_{n+1})\dot{X}_{n+1} + C(t_{n+1}) = \\ &= A(t_{n+1})h^2\beta_{-1}^{(m)}\ddot{X}_{n+1} + B(t_{n+1})h\alpha_{-1}^{(m)}\ddot{X}_{n+1} + D[t_{n+1}, X_n] + \\ +A(t_{n+1})\left[h\dot{X}_{n+1} + \frac{1}{2}h^2\ddot{X}_n\right] &+ \sum_{j=0}^{m}\left(h^2A(t_{n+1})\beta_j^{(m)} + hB(t_{n+1})\alpha_j^{(m)}\right)\ddot{X}_{n-j} \end{aligned} \tag{97}$$

and the system (97) is linear and can be solved in the 3 ($= n/2$) unknown $\ddot{X}_{n+1}$ by inverting the matrix:

$$I - A(t_{n+1})\beta_{-1}^{(m)}h^2 - B(t_{n+1})\alpha_{-1}^{(m)} = H \quad . \tag{98}$$

At this stage the value of $\ddot{X}_{n+a}$ can be substituted in Eq.(96) giving the value of the corrector in a single operation, with no iterations. This procedure requires the computetation of the second member of the variation equation just once, taht is the computation of $D[t_{n+1}, X_n]$ (which actually is not the real second member of the variation eqaution in any point, since A, B and C are computed in t_{n+1} and X, $\dot{X}$ in t_n).

[1] Bevilacqua R., Gianni P. and Milani A., Nonlinear stability of implicit Runge-Kutta methods derived from gaussian quadrature, *dovesta*, 19XX. 1.2.2, 1.2.4

[2] Burlisch J. and Stoer R., *Introduction to numerical analysis*, Springer, 1980. 1.2.1

[3] Burrage R. and Buthcher J.C., Stability criteria for implicit Runge-Kutta methods, *Siam Journal Numerical Analysis*, **16**, 46, 1979. 1.2.2

[4] Butcher J.C., Implicit Runge-Kutta processes, *Math. Comp*, **18**, 50, 1964. 1.2.1, 1.2.2, 1.2.3

[5] Lapidus L. and Seinfeld J., *Numerical solution of ordinary differential equations*, Academic Press, 1971.

1.2, 1.2.2