Optimal use of a numerical method for solving differential equations based on Taylor-series expansions

Citation for published version (APA):

Sonnemans, P. J. M., Goey, de, L. P. H., & Nieuwenhuizen, J. K. (1991). Optimal use of a numerical method for solving differential equations based on Taylor-series expansions. *International Journal for Numerical Methods in Engineering*, 32(3), 471-499. <https://doi.org/10.1002/nme.1620320303>

DOI:

[10.1002/nme.1620320303](https://doi.org/10.1002/nme.1620320303)

Document status and date:

Published: 01/01/1991

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

OPTIMAL USE OF A NUMERICAL METHOD FOR SOLVING DIFFERENTIAL EQUATIONS BASED ON TAYLOR SERIES EXPANSIONS

P. J. M. SONNEMANS, L. P. H. DE GOEY AND J. K. NIEUWENHUIZEN

Eindhoven University of Technology, Faculty of Mechanical Engineering, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

SUMMARY

Efficiency in solving differential equations is improved by increasing the order of a Taylor series approximation. Computing time can be reduced up to a factor of 40 and an amount of memory storage can be saved, up to a factor of 70.

The truncation error can be estimated not only by order but also by magnitude.

1. INTRODUCTION

Solving problems numerically in engineering today often requires huge amounts of computing time and memory storage: for example, three-dimensional problems involving non-stationary phenomena and complex geometry. Therefore it is of the utmost importance to develop numerical methods which are as efficient as possible for the problem to be tackled.

A method for solving differential equations is proposed. The method, based on the continuity of the solution, is not new.¹⁻⁵ Gibbons¹ formulated automatic programs to integrate ordinary differential equations which are reducible to rational form. Barton *et al.*^{2,3} showed the efficiency of the method in terms of required computing time, concluding that in some applications computing time may be reduced up to a factor of 40. For heat conduction the method is superior to all others in shortness and accuracy, according to Collatz.⁴ He generalized the application to two-dimensional systems and pointed out that instability was one of the problems still to be tackled. In a stability analysis, Small⁵⁻⁸ indicated that the method is Von Neumann stable over a wide class of equations, if the differential equations are linear. He also infers this stability for non-linear problems. Consistency and convergence were also proved by Small.

Bearing this in mind, we propose a general formulation of this Taylor series method to handle one-dimensional differential equations (DE). This one-dimensionality is no essential restriction of the method, it is just for simplicity. For two- and three-dimensional problems the presented advantages of this method are even more spectacular presumably.

The formulation is set up to be able to change the number of elements N , and to change the order of the Taylor series expansion k . Using equidistant elements, expressions to estimate gained accuracy and required computing time are derived. We use these expressions to optimize accuracy and computing time in terms of k and N .

Unlike the more conventional method of increasing the number of elements only, we conclude that the optimum combination of k and N is located towards regions of higher-order k , especially

for high accuracy demands and higher-order DE. Derivatives of the solution are available by hardly any effort. Non-linearity of the DE does not complicate the method fundamentally. These aspects are additional advantages.

The analytical work which has to be done preceding each different type of differential equation may be regarded as a disadvantage of the method. However, within the field of engineering, the same type of differential equations underlie a wide class of problems. Consequently, it is necessary only once to perform the analytical part of the problem. To increase the efficiency even more or to tackle (extremely) stiff DE, one can apply non-equidistant elements, as will be demonstrated in Section 7. Another possible improvement, with respect to the treatment of boundary conditions in more-dimensional problems, is a formulation of the problem in Body-Fitted Co-ordinates (BFC), an object of recent study.

We hope therefore that we can take advantage of this method in future, especially within the areas of stress analysis, fluid-flow and heat-conduction problems.

In the following sections we will present the method and optimize 'efficiency'.

In Section 2 we present the formalism to approximate the exact solution of a general one-dimensional (non-)linear DE numerically. The order of approximation k is introduced and the calculating domain is subdivided into N elements.

In Sections 3 and 4 we derive analytical expressions in terms of k and N for the relative truncation error ε of the approximation and for the computing time T needed to calculate the approximation. These expressions contain the order of the DE m and the radius of convergence of the series R as parameters.

In Section 5 the above mentioned analytical expressions are used to minimize required computing in terms of k and N at a given relative accuracy. This optimum combination (k_0, N_0) still depends on the two parameters m and R . The influence of both parameters is shown.

Demonstrations, using equidistant elements, are provided as illustration in Section 6 and an application of non-equidistant elements is shown in Section 7. Concluding remarks are summarized in Section 8.

2. FORMALISM

A method is proposed to approximate the solution of a certain class of one-dimensional differential equations (DE), not necessarily linear. Using the notation

$$\chi^{(i)} = \frac{d^i \chi}{d\tau^i} \quad (1)$$

as the i th derivative of $\chi(\tau)$, an m th-order DE can be denoted by

$$\mathcal{F}\{\chi^{(m)}, \chi^{(m-1)}, \dots, \chi^{(1)}, \chi^{(0)}, \tau\} = 0 \quad (2)$$

In general, the solution for an m th-order DE for a real function $\chi(\tau)$, $\tau \in [\tau_0, \tau_1]$ depends on m parameters $c_1, c_2, \dots, c_m$.

These parameters are determined by the m initial or boundary conditions

$$\mathcal{G}_b\{\chi^{(m-1)}(\tau_0), \dots, \chi^{(1)}(\tau_0), \chi^{(0)}(\tau_0), \tau_0, \chi^{(m-1)}(\tau_1), \dots, \chi^{(1)}(\tau_1), \chi^{(0)}(\tau_1), \tau_1\} = 0 \quad (3)$$

$$b = 1, 2, \dots, m$$

Several methods are known and have proved their use in practice in solving these DE. One of the methods, relatively uncommon in the engineering field, is a method based on power series. The

basic idea of this method is to approximate the exact solution by means of a Taylor series expansion.

The function $\chi(\tau)$ is expanded into a Taylor series around τ_e , a point within the calculation domain

$$\chi(\tau_e + t) = \sum_{i=0}^{\infty} x_i t^i \quad (4)$$

with

$$x_i = \frac{\chi^{(i)}(\tau_e)}{i!}$$

and t as local co-ordinate. According to Small,⁶ maximum stability is achieved if τ_e is taken at the middle of the calculation domain. Substituting these series in the DE and equating the factors in front of $(\tau - \tau_e)^i$ to zero for each i yields an infinite number of equations

$$F_i(x_0, x_1, \dots, x_{i+m}) = 0, \quad i = 0, 1, 2, \dots, \infty \quad (5)$$

because the DE must be satisfied for all $\tau \in [\tau_0, \tau_1]$. Similarly the m initial and boundary conditions are denoted by

$$G_b(x_0, x_1, x_2, \dots) = 0, \quad b = 1, 2, \dots, m \quad (6)$$

Note that $\mathcal{F}$, $\mathcal{G}_b$ and F_i ($i = 0, 1, \dots, \infty$), G_b ($b = 1, 2, \dots, m$) represent the DE and boundary conditions exactly. The equivalence between these two formulations for functions represented by converging series forms the basic idea of the method. These formulations are illustrated in Section 6.

$\mathcal{F}$ and $\mathcal{G}_b$ are functionals with the function χ as the solution of the DE and its first m derivatives as arguments. This is a general way of denoting a DE.

F_i and G_b are functions with the values of the solution of the DE and all its derivatives at the point $\tau = \tau_e$ as its arguments.

The solution (and its derivatives) in the neighbourhood of $\tau = \tau_e$ is known if all terms of the Taylor expansion x_i ($i = 0, 1, \dots, \infty$) in $\tau = \tau_e$ are known.

Equations (5) and (6) form a well-posed infinite set of equations for the infinite set of unknowns x_i , $i = 0, 1, 2, \dots, \infty$. To obtain a set suitable for numerical computation the series has to be truncated. Suppose we want to estimate the exact solution by the series

$$\chi^{(0)}(\tau_e + t) = \sum_{i=0}^{k+m} x_i t^i \quad (7)$$

using only the first $k + m + 1$ coefficients $x_0, x_1, \dots, x_{k+m}$. The choice of $k + m + 1$ coefficients might seem to be a bit strange at first sight. It will become clear by considering the estimation of the highest derivative in the m th-order DE

$$\chi^{(m)}(\tau_e + t) = \sum_{i=0}^k \frac{(i+m)!}{i!} x_{i+m} t^i \quad (8)$$

It is easy to see that this m th-order derivative is approximated by a k th-order series. Approximations of all lower derivatives contain more terms. The series representation of the n th-order derivative is

$$\chi^{(n)}(\tau_e + t) = \sum_{i=0}^{k+m-n} \frac{(i+n)!}{i!} x_{i+n} t^i \quad (9)$$

In order to determine the coefficients $x_0, x_1, \dots, x_{k+m}$ we need $k + m + 1$ equations. The boundary conditions yield m equations, while the remaining $k + 1$ equations are provided by equation (5) for $i = 0, 1, \dots, k$.

$$g_b(x_0, x_1, \dots, x_{k+m}) = 0, \quad b = 1, 2, \dots, m \quad (10a)$$

$$f_i(x_0, x_1, \dots, x_{i+m}) = 0, \quad i = 0, 1, \dots, k \quad (10b)$$

The numerical solution of this finite set is a k th-order approximation of the exact solution and all its derivatives, as we will show in Section 3. The approximation of the function χ , determined up to the k th-order, also contains some information on the higher coefficients $x_{k+1}, \dots, x_{k+m}$,

$$\sum_{i=k+1}^{k+m} x_i t^i \quad (11)$$

It causes a slight improvement on the numerical results in our examples.

The approximation according to equation (7) will give some idea of the behaviour of the variable in the neighbourhood of $\tau = \tau_e$. Naturally, nothing can be asserted generally about the extent of the region of convergence. We implicitly suppose that the Taylor series of the solution exists and is convergent. This is an important restriction of the method presented so far: the Taylor series must be convergent.

If the calculation domain exceeds the region of convergence, we must subdivide the calculation domain in (N) elements, and proceed in the same way as before for each element separately. Here we restrict ourselves to equidistant elements for simplicity. An application of non-equidistant elements is given in Section 7.

Formally, equation (10) is valid for each element separately, introducing a set of $k + m + 1$ unknowns $x_0^j, x_1^j, \dots, x_{k+m}^j$ for each element j :

$$f_i^j(x_0^j, x_1^j, \dots, x_{i+m}^j) = 0, \quad i = 0, 1, \dots, k \quad (12)$$

$$j = 1, 2, \dots, N$$

and of course the initial and boundary conditions

$$g_b(x_0^1, x_1^1, \dots, x_{k+m}^1, x_0^N, x_1^N, \dots, x_{k+m}^N), \quad b = 1, 2, \dots, m \quad (13)$$

Some bookkeeping shows that we still need $(N - 1)m$ equations to make the set of equations well posed. This number of equations will be provided by connecting the elements, because until now the approximations for the separate elements do not correlate at all. To obtain a well-posed set of equations we introduce m continuity conditions on the $(N - 1)$ common element boundaries by demanding the continuation of the m lower derivatives of the variable. These conditions connect the behaviour of the solution of two adjacent elements.

These continuity conditions will be denoted by

$$h_b^j(x_0^j, x_1^j, \dots, x_{k+m}^j, x_0^{j+1}, x_1^{j+1}, \dots, x_{k+m}^{j+1}) = 0 \quad (14)$$

$$b = 1, 2, \dots, m \quad j = 1, 2, \dots, N - 1$$

For ease of notation the unknowns of all the elements will be denoted by a vector

$$\mathbf{x} = (x_0^1, x_1^1, \dots, x_{k+m}^1, \dots, x_0^j, x_1^j, \dots, x_{k+m}^j, \dots, x_0^N, x_1^N, \dots, x_{k+m}^N), \quad j = 1, 2, \dots, N \quad (15)$$

and the set of equations (12), (13) and (14) by

$$\mathbf{a}(\mathbf{x}) = 0 \quad (16)$$

Here $\mathbf{a}$, in general, is a vector of the (non-)linear operators f_i^j , g_b and h_b^j arranged in a special order such that the resulting set of equations to be solved can be represented by a band matrix. The vector $\mathbf{0}$ contains zeros.

This well-posed set of equations can be solved by standard methods, e.g. the Newton–Raphson method.

In fact, this method is a repetitive procedure to improve an estimation of the solution by a determined correction.

Suppose $\mathbf{x}^*$ is an estimation of the solution $\mathbf{x}$ of equation (16). A new estimation $\mathbf{x}^{**} = \mathbf{x}^* + d\mathbf{x}$ can be determined by calculation of the correction vector $d\mathbf{x}$, using a linearization of the set of equations

$$\mathbf{a}(\mathbf{x}^{**}) = \mathbf{0} = \mathbf{a}(\mathbf{x}^* + d\mathbf{x}) = \mathbf{a}(\mathbf{x}^*) + \left. \frac{d\mathbf{a}}{d\mathbf{x}} \right|_{\mathbf{x}^*} d\mathbf{x} + \dots \quad (17)$$

The derivative of $\mathbf{a}$ with respect to $\mathbf{x}$ for the estimation $\mathbf{x} = \mathbf{x}^*$ is a matrix $\mathbf{A}$. Neglecting the higher-order derivatives of $\mathbf{a}$, and substituting the matrix $\mathbf{A}$ yields

$$\mathbf{A} d\mathbf{x} = -\mathbf{f}, \quad \text{or} \quad d\mathbf{x} = -\mathbf{A}^{-1}\mathbf{f} \quad (18)$$

using $\mathbf{f}$ as a notation for $\mathbf{a}(\mathbf{x}^*)$. This set of linear equations can be solved with a standard numerical method, e.g. *LU*-decomposition. As indicated earlier, we will arrange f_i^j , g_b and h_b^j in $\mathbf{a}$ in such a way to give matrix $\mathbf{A}$ a band structure, of which we take advantage during the solving process. Note that a linear DE is represented by a linear operator $\mathbf{a}$ and that the solution of equation (16) is found after only one iteration.

3. ERROR ESTIMATION

In order to qualify the approximation we have to study the difference between the exact analytical solution and the approximation. This difference, often referred to as the (truncation) error, is of order δ^{k+1} , as indicated by Small,⁶ where 2δ represents the length of one (equidistant) element.

This becomes obvious if we consider the truncation of the n th-derivative of the exact solution according to the formalism

$$\chi^{(n)}(\tau_e + \delta) = \sum_{i=0}^{k+m-n} \frac{(i+n)!}{i!} x_{i+n} \delta^i + E_n \quad (19)$$

The truncation error is known as the rest term of Lagrange,⁹

$$E_n = \frac{1}{(k+m+1-n)!} \chi^{(k+m+1)}(\tau_e + \xi) \delta^{k+m+1-n}, \quad 0 \leq \xi \leq \delta \quad (20)$$

It is easy to see that the truncation error of the highest derivative in the DE is of lowest order,

$$E_m = \frac{1}{(k+1)!} \chi^{(k+m+1)}(\tau_e + \xi) \delta^{k+1} \quad (21)$$

The lowest order in approximating the derivatives of the DE forms the restricting factor for all approximations since equation (2) has to be satisfied. Accordingly we find, in agreement with Small,⁶ that the truncation error of the function itself is also of order δ^{k+1} .

At this stage it seems unnecessary to expand the series representation of the (lower) derivatives beyond the term containing t^k , as is done in equation (9). Indeed, no improvement is made on the order of approximation but practice shows that numerical results move just a bit more towards

the exact solution if we do it as well. That is the reason for the expansion of the series representation of the lower derivatives beyond the term t^k . Since the bandwidth of the system matrix has not changed by including these extra terms no extra computing time is needed.

To quantify the error we still need the factor preceding δ^{k+1} . This factor must describe the rest term and consequently it must contain some information about the series behaviour of convergence.

For an exact solution the $(k+1)$ th-order truncation error of the function itself is given by

$$E = \frac{1}{(k+1)!} \chi^{(k+1)}(\tau_e + \xi) \delta^{k+1}, \quad 0 \leq \xi \leq \delta \quad (22)$$

The error is exact whenever we are dealing with the exact solution. However, we estimate it by

$$E \approx \frac{1}{(k+1)!} \chi^{(k+1)}(\tau_e) \delta^{k+1} \approx x_{k+1} \delta^{k+1} \quad (23)$$

This estimation is good if the solution is known exactly in combination with sufficiently small elements. We use this very same expression as an indicator for the error in the approximation. We assume that the truncation error is small, implying that the approximation differs slightly from the exact solution. Consequently the difference between equation (23) and the true error will be small. We use

$$e \equiv |x_{k+1}| \delta^{k+1} \quad (24)$$

as indicator of the error. Here 2δ is the element length. Using N equidistant elements, δ satisfies

$$\delta = \frac{L}{2N} \quad (25)$$

L being the length of the calculation domain $(\tau_1 - \tau_0)$.

From equation (24) we see that for stronger converging series the rest term will be smaller, which is expressed in a smaller value of $|x_{k+1}|$. Furthermore, it can be seen that the error is indeed of order $k+1$.

Because we want to study the influence of the order of approximation and the number of elements we rewrite equation (24) as a function of k and N . First of all we normalize the error because we are interested in the relative errors only. Therefore we divide the error indicator by $|x_0|$, the order of magnitude of the function $\chi^{(0)}$ itself,

$$\varepsilon = \frac{e}{|x_0|} = \frac{|x_{k+1}|}{|x_0|} (\delta)^{k+1} = \left(\frac{\delta}{r_{k+1}} \right)^{k+1} = \varepsilon_0(\delta)^{k+1} \quad (26)$$

with

$$\frac{1}{r_k} = k \sqrt{\frac{|x_k|}{|x_0|}} \quad (27)$$

as an estimation for the radius of convergence z for which

$$\lim_{k \rightarrow \infty} \frac{1}{r_k} = \frac{1}{z} \quad (28)$$

is satisfied. Substituting equation (25) we rewrite the expression for the error indicator

$$\varepsilon = \left(\frac{L}{2r} \right)^{k+1} \left(\frac{1}{N} \right)^{k+1} \quad (29)$$

In equation (29) r expresses the information about the behaviour of convergence of the series, which we already mentioned. This relative truncation error is independent of co-ordinate scaling since L is scaled in the same way as r . We use this knowledge to write equation (29) more elegantly by introducing the normalized radius of convergence

$$\frac{1}{R} = \frac{L}{2r} \quad (30)$$

The quantity R can be interpreted as an estimation for the radius of convergence if we scaled the co-ordinate to $L = 2$. Substitution of (30) changes the relation for the indicator of relative error into

$$\varepsilon = \left(\frac{1}{R_{k+1}N} \right)^{k+1} = \varepsilon_0 \left(\frac{1}{N} \right)^{k+1} \quad (31a)$$

with

$$\varepsilon_0 = \left(\frac{1}{R_{k+1}} \right)^{k+1} \quad (31b)$$

With equation (26) we are able to give ε_0 a mathematical meaning.

The parameter ε_0 represents the ratio between $|x_{k+1}|$ and $|x_0|$. A value of $\varepsilon_0 = 1$ means x_{k+1} is of the same magnitude as x_0 . Smaller values of ε_0 imply less influence of the $(k+1)$ th derivative, while higher values suggest a stronger influence of this derivative. High values of ε_0 do not mean that the series does not converge. Convergence can be forced by taking very small elements so that

$$\frac{\delta}{r_{k+1}} < 1 \quad (32)$$

The parameter ε_0 merely tells us whether it will cost us much effort to achieve a certain amount of accuracy. So ε_0 contains information about the behaviour of convergence.

The truncation error according to equations (31) depends on the behaviour of convergence R and on the variables k and N . In Figure 1 this truncation error is plotted for $R = 1$. It is obvious that both increasing k and increasing N will give more accurate results. On the other hand, increasing k or N will increase the required computing time as well, which we will see in the next section. Given a required accuracy, an optimum combination of k and N exists for which computing time will be minimal. Finding this optimum is the aim of the next two sections.

Before that, we mention that, whenever an approximation of the solution is determined, equation (26) can be evaluated for each element to give an indication of the accuracy at once.

Now we are able to indicate the gained accuracy of the approximation within the whole calculation domain. Being able to indicate the accuracy of the solution not only by order but also in magnitude in such a fast way is rather unique, since conventional methods need recalculating the whole problem with a different number of elements or a different order of approximation, in order to get an idea of the accuracy.

4. COMPUTING TIME

In problems involving many elements, computing time is primarily spent on solving the set of equations

$$A \, dx = f \quad (33)$$

Figure 1. The relative truncation error ε as a function of k and N for $R = 1$

In one-dimensional problems the matrix A is very often banded. Some problems destroy this band structure; for example, 'circular' problems in which only the circumferential direction is of importance. Boundary conditions in this kind of problem connect the first and the last element, causing loss of band structure. For these problems special profile solvers should be used. We restrict ourselves to problems with a band structure, covering a great variety of engineering problems. The solver used is a procedure for LU -decomposition taking advantage of the band structure. The computing time we estimate is based on this solver. Other solvers give comparable results since the order of their computing time remains the same.

Computing time required by the band matrix solver used here, to solve a set of linear equations represented by a $(n \times n)$ matrix with p subdiagonals on both sides of the main diagonal, is proportional to

$$n(p^2 + 3p + 1) - (\frac{2}{3}p^3 + p^2) \quad (34)$$

according to Golub and van Loan.¹⁰

In our case the dimension and bandwidth appear to be

$$\begin{aligned} n &= N(k + m + 1) \\ p &= (k + 2m) \end{aligned} \quad (35)$$

in which N is the number of elements, k is the order of approximation and m is the order of the DE. Computing time C is determined by

$$\begin{aligned} C &= C_0 \{ N(k + m + 1) [(k + 2m)^2 + 3(k + 2m) + 1] \\ &\quad - [\frac{2}{3}(k + 2m)^3 + (k + 2m)^2] \} \end{aligned} \quad (36)$$

C_0 is a hardware dependent factor representing the time the computer needs for a floating point operation. High values of C_0 indicate a low performance in floating point arithmetic. In the next

section we will use a non-dimensional time

$$T = \frac{C(k, N)}{C_0} \quad (37)$$

Computing time is increased by higher values of both k and N . In Figure 2 computing time according to equation (37) is plotted for $m = 2$.

In two-dimensional problems, the subject of our recent study, both the number of variables and the number of equations will increase. In general a band structure can be achieved by numbering the elements appropriately. Very often in two-dimensional problems the band is sparse, which might be exploited to reduce the amount of computing time even more.

5. OPTIMIZING

As mentioned already in the first section, the Taylor series has to be convergent in order to approximate the exact solution by truncating the series. To what order the series has to be truncated depends on the magnitude of the relative truncation error, which is given by

$$\varepsilon = \varepsilon_0 \left(\frac{1}{N} \right)^{k+1} \quad (38a)$$

with

$$\varepsilon_0 = \frac{|x_{k+1}|}{|x_0|} = \left(\frac{1}{R} \right)^{k+1} \quad (38b)$$

Without loss of generality we have normalized the co-ordinate, i.e. $L = 2$.

Figure 2. Computing time T as a function of k and N for $m = 2$.

Several combinations of k and N , needing different amounts of computing time, are possible to achieve the same accuracy, which can be seen from equation (38). An optimum combination (k_0, N_0) exists, so that the computing time will be minimal to achieve a desired accuracy. To find this optimum we will use

$$\varepsilon = \varepsilon(k, N) = \left(\frac{1}{RN} \right)^{k+1} \quad (39a)$$

$$T = T(k, N) = N f(k, m) - g(k, m) \quad (39b)$$

$$f(k, m) = (k + m + 1)[(k + 2m)^2 + 3(k + 2m) + 1]$$

$$g(k, m) = \frac{2}{3}(k + 2m)^3 + (k + 2m)^2$$

which were explained in Sections 3 and 4 as the analytical relations describing the relative error and the non-dimensional required computing time respectively. We treat the quantities k and N as if they are continuous in order to get a better idea of the behaviour of the different relations. We keep in mind that k and N are discrete in reality.

Lines of constant ε and constant T are plotted in the k, N -space in Figure 3. We see that higher accuracy (i.e. smaller values of ε) can be achieved by increasing k and N , as might be expected. Shorter computing time can be achieved by smaller values of k and N . In Figure 3 it can be seen that in an optimum both lines of constant ε and of constant T coincide at (k_0, N_0) and have the same tangent. We will now determine this optimum.

Figure 3. Projections of contours of constant accuracy and of constant computing time in the (k, N) area, with an optimum

First we eliminate the variable N from equation (39a) with ε given as a tolerated relative error. The resulting equation for N is substituted in equation (39b) yielding a function in k which is to be minimized.

The expressions locating the optimum combination (k_0, N_0) formally still contain two parameters, i.e. m and R .

The first parameter to be discussed is m , the order of the DE. In Figure 4 optimum curves for $m = 1, 2, 3$ and 4, for $R = 1$, are shown. It can be seen that for higher-order DE the optimum curves are located towards regions of higher k , as expected, since high-order DE express the importance of high derivatives.

The second parameter to be discussed is R . Its influence can be seen in Figure 5. The mathematical interpretation of this parameter has already been discussed in Section 3. It contained information about the behaviour of convergence. Assuming ε to be constant, we see that, for series with high values of R , the optimum is also shifted towards regions of higher k , as might be expected, since high values of R indicate that large elements can be used. Accuracy is gained more economically by increasing the order k rather than using more smaller elements. This will become clear when we compare the computing time of the optimal choice with that of first- and second-order approximation as used in most existing methods.

We define ratios of required computing time and memory storage used by an i th-order choice, i.e. $k = i$ and $N = N_i$ (the corresponding number of elements to achieve the same accuracy), and

Figure 4. Optimum curves in the (k, N) area for $R = 1$, with m as parameter

Figure 5. Optimum curves in the (k, N) area for $m = 2$, with R as parameter

by optimum choice (k_0, N_0) :

$$T_{i \text{ rat.}} = \frac{T(k = i, N_i)}{T(k_0, N_0)} \quad (40a)$$

$$M_{i \text{ rat.}} = \frac{M(k = i, N_i)}{M(k_0, N_0)} \quad (40b)$$

A comparison with 1st- and 2nd-order methods, as far as they are comparable with the same relationships equation (39) for relative truncation error and computing time, is made with respect to computing time and memory storage relative to this Taylor series method.

Quantities $T_{1 \text{ rat.}}$ and $T_{2 \text{ rat.}}$, and $M_{1 \text{ rat.}}$ and $M_{2 \text{ rat.}}$ are presented in Tables I and II respectively.

The computing time is calculated from (39b), and as an estimation for needed memory storage we use

$$M = N(k + m + 1)(3 + 2(k + 2m)) - (k + 2m)(k + 2m + 1) \quad (41)$$

representing the memory needed to store the non-zero elements of the matrix A and the vectors dx and f .

Spectacular amounts of saved computing time and memory storage are achievable for higher-order DE, in particular if a high accuracy is desired. The Taylor series method in these cases (e.g. $m = 4$, $R = 1$, $\epsilon = 10^{-6}$) is up to a factor 43 faster and uses up to a factor of 70 less memory storage than a 1st-order method, which can be seen from Table I.

Table I. Ratios of computing time needed by a 1st- and 2nd-order method: $T_{1 \text{ rat.}}$ and $T_{2 \text{ rat.}}$ for $R = 1$

ε	$m = 1$		$m = 2$		$m = 3$		$m = 4$	
	$T_{1 \text{ rat.}}$	$T_{2 \text{ rat.}}$	$T_{1 \text{ rat.}}$	$T_{2 \text{ rat.}}$	$T_{1 \text{ rat.}}$	$T_{2 \text{ rat.}}$	$T_{1 \text{ rat.}}$	$T_{2 \text{ rat.}}$
10^{-1}	1.0	1.3	1.0	1.0	1.1	1.0	1.3	1.1
10^{-2}	1.1	1.0	1.4	1.0	1.8	1.1	2.1	1.0
10^{-3}	1.7	1.1	2.5	1.2	3.2	1.5	4.1	1.7
10^{-4}	3.0	1.3	4.7	1.7	6.5	2.1	8.4	2.5
10^{-5}	5.8	1.7	9.7	2.4	14	3.0	18	3.7
10^{-6}	12	2.5	21	3.6	32	4.7	43	5.9
10^{-7}	27	3.7	49	5.6	74	7.6	102	9.7
10^{-8}	62	5.8	116	9.0	180	13	251	16
10^{-9}	146	9.4	284	15	447	21	632	28
10^{-10}	356	16	708	26	1130	37	1620	49

Table II. Ratios of memory needed by a 1st- and 2nd-order method: $M_{1 \text{ rat.}}$ and $M_{2 \text{ rat.}}$ for $R = 1$

ε	$m = 1$		$m = 2$		$m = 3$		$m = 4$	
	$M_{1 \text{ rat.}}$	$M_{2 \text{ rat.}}$	$M_{1 \text{ rat.}}$	$M_{2 \text{ rat.}}$	$M_{1 \text{ rat.}}$	$M_{2 \text{ rat.}}$	$M_{1 \text{ rat.}}$	$M_{2 \text{ rat.}}$
10^{-1}	1.0	1.0	1.1	1.0	1.2	1.0	1.4	1.1
10^{-2}	1.4	1.0	1.8	1.1	2.2	1.3	2.6	1.4
10^{-3}	2.4	1.2	3.4	1.5	4.4	1.8	5.4	2.1
10^{-4}	4.9	1.7	7.3	2.2	9.7	2.7	12	3.2
10^{-5}	11	2.6	17	3.5	23	4.4	28	5.3
10^{-6}	25	4.1	40	5.7	55	7.3	70	8.9
10^{-7}	60	6.7	98	9.6	137	13	177	15
10^{-8}	151	11	248	17	352	22	461	27
10^{-9}	387	20	644	29	924	39	1220	49
10^{-10}	1010	36	1700	53	2470	71	3280	90

6. DEMONSTRATIONS

In this section the method and its formalism as presented in Section 2 is illustrated by three examples, one linear and two non-linear.

First example

$$M \frac{\partial^2 \chi}{\partial \tau^2} + B \frac{\partial \chi}{\partial \tau} + K \chi = MG, \quad \tau \in [0, 1] \quad (42)$$

with initial conditions:

$$\chi(\tau = 0) = 0$$

$$\left. \frac{\partial \chi}{\partial \tau} \right|_{\tau=0} = 0$$

This DE is recognized as the equation of motion of a mass-spring system in a gravitational field.

Second example

$$\frac{\partial \chi}{\partial \tau} - a(\chi)^2 = 0, \quad \tau \in [0, 0.5] \quad (43)$$

with boundary condition:

$$\chi(\tau = 0.5) = 1$$

The DE is an equation without direct physical interpretation. Its relevancy lies in its non-linearity, and in the structure of the analytical solution. Since the analytical solution can be written as a geometrical series, the approximation r of the radius of convergence is exact. Consequently the accuracy analysis is easier to observe.

Third example

$$\frac{\partial^2 \chi}{\partial \tau^2} - \gamma \frac{\partial \chi}{\partial \tau} = S(\chi), \quad \tau \in [0, 1] \quad (44a)$$

with boundary conditions

$$\chi(\tau = 0) = 0.15$$

$$\left. \frac{\partial \chi}{\partial \tau} \right|_{\tau=1} = 0$$

The term on the right-hand side is strongly non-linear:

$$\begin{aligned} S(\chi) &= \alpha(1 - \chi)^2 \exp[\beta(\chi - 1)] \quad 0 \leq \chi \leq 1 \\ &= 0 \quad \text{elsewhere} \end{aligned} \quad (44b)$$

This last example may be regarded as the energy equation used to calculate the stationary state of the non-dimensional temperature profile $\chi(\tau)$ in a flat flame as a function of the non-dimensional distance τ to the burner.¹¹ It serves as a demonstration of the method for real-life problems.

Each of these cases will be discussed in the sections below. The optimum combinations (k_0, N_0) are determined and compared with the analytical results. We already mentioned the general relation for the n th-derivative as a Taylor series using only the first $k + m + 1$ coefficients $x_0, x_1, \dots, x_{k+m}$ because we make extensive use of it in the next subsections. The n th-derivative is given by

$$\chi^{(n)}(\tau_e + t) = \sum_{i=0}^{k+m-n} \frac{(i+n)!}{i!} x_{i+n} t^i \quad (45)$$

6.1. The mass-spring system

Consider the equation of motion of a mass-spring system in a gravitational field, equation (42). We denote the DE according to the formalism by

$$\mathcal{F}: M\chi^{(2)} + B\chi^{(1)} + K\chi^{(0)} - MG \quad (46a)$$

with boundary conditions

$$\mathcal{G}_1: \chi^{(0)}(\tau = 0) \quad (46b)$$

$$\mathcal{G}_2: \chi^{(1)}(\tau = 0) \quad (46c)$$

Substituting equation (45) and equating the different order terms to zero yields

$$f_0: 2Mx_2^j + Bx_1^j + Kx_0^j - MG \quad (47a)$$

$$f_i: (i+2)(i+1)Mx_{i+2}^j + (i+1)Bx_{i+1}^j + Kx_i^j \quad (47b)$$

$$i = 0, 1, \dots, k \quad j = 1, 2, \dots, N-1$$

The boundary and continuity conditions result in

$$g_1: \sum_{i=0}^{k+m} x_i^1(-\delta)^i \quad (47c)$$

$$g_2: \sum_{i=0}^{k+m-1} (i+1)x_{i+1}^1(-\delta)^i \quad (47d)$$

$$h_1^j: \sum_{i=0}^{k+m} x_i^j(\delta)^i - x_{i+1}^{j+1}(-\delta)^i \quad (47e)$$

$$h_2^j: \sum_{i=0}^{k+m-1} (i+1)\{x_{i+1}^j(\delta)^i - x_{i+1}^{j+1}(-\delta)^i\}, \quad j = 1, 2, \dots, N-1 \quad (47f)$$

with $m = 2$. The physical quantities are chosen to be $M = 1$, $B = 5$, $K = 40$ and $G = 10$. The analytical solution for this problem is given by the equation

$$\chi(\tau) = a_1 \{1 - \exp(-a_2 \tau) [\cos(\omega \tau) + a_3 \sin(\omega \tau)]\} \quad (48)$$

$$\omega = \frac{\sqrt{4KM - B^2}}{2M} = \frac{\sqrt{135}}{2}$$

$$a_1 = \frac{MG}{K} = \frac{1}{4}$$

$$a_2 = \frac{B}{2M} = \frac{5}{2}$$

$$a_3 = \frac{a_2}{\omega} = \frac{5}{\sqrt{135}}$$

Observe that the set of equations is linear, due to the fact that the original DE is linear. The problem has been calculated with different numbers of elements ($N = 1, 2, \dots, 6$), and different orders of approximation ($k = 1, 2, \dots, 6$). The normalized radius of convergence is estimated numerically and its smallest value is about $R = 1.1$.

The indications of the error are satisfying, which can be seen from Table III. Suppose a relative error of $\varepsilon = 10^{-5}$ is desired. According to the optimum curve ($m = 2$ and $R = 1.1$) we find optimum values which differ slightly from numerical results.

Optimum values	Numerical values
$N_0 = 5.0$	$N_{\text{num}} = 4$
$k_0 = 5.7$	$k_{\text{num}} = 6$
$T_0 = 4.7$ kFlops	$T_{\text{num}} = 4.0$ kFlops

(49)

Again we mention that k_0 and N_0 are treated as if they were continuous quantities. Compared with 1st and 2nd-order methods based on the same relations of accuracy and time, we have

Table III. Exact (number above) and estimated (number below) relative error in the numerical solution of the 1st example

$k \backslash N$	1	2	3	4	5	6
1	2.6E-1	1.4E-1	4.3E-2	2.3E-2	1.4E-2	9.2E-3
	9.2E-1	1.0E-1	1.3E-1	1.5E-1	1.4E-1	1.3E-1
2	9.1E-2	1.9E-2	1.5E-2	6.3E-3	2.9E-3	1.6E-3
	2.0E-1	2.4E-1	1.3E-1	7.2E-2	4.8E-2	3.3E-2
3	1.8E-2	2.2E-2	5.6E-3	1.7E-3	6.6E-4	3.2E-4
	4.3E-1	1.1E-1	2.3E-2	7.0E-3	2.8E-3	1.2E-3
4	6.5E-2	5.9E-3	2.0E-4	4.2E-5	1.7E-5	6.6E-6
	6.6E-1	3.5E-3	4.4E-3	1.8E-3	7.9E-4	4.0E-4
5	2.2E-2	7.2E-4	1.3E-4	2.4E-5	6.9E-6	2.3E-6
	1.2E-1	1.0E-2	1.3E-3	2.9E-4	9.2E-5	3.6E-5
6	5.3E-3	4.4E-4	1.6E-5	1.4E-6	2.5E-7	5.7E-8
	1.3E-1	1.6E-3	7.6E-5	5.9E-6	2.4E-6	1.0E-6

reduced computing time by a factor of 9.9 and 2.4 respectively. These reductions are even more spectacular for higher accuracy and higher-order DE, as can be seen from Tables I and II. Another important factor is the amount of memory storage needed to store the matrix A . Here a reduction is achieved by a factor of 17 and 3.5, respectively, since fewer elements are needed.

Optimizing gives good estimates for optimum values of the relevant calculation quantities. One important cause of the small differences is that the estimations of $T(k, N)$ and $\varepsilon(k, N)$ are, of course, not exact. Another reason is the treatment of k_0 and N_0 as if they were continuous.

6.2. A non-linear differential equation

The second example to be examined in the non-linear DE.

According to the formalism we write

$$\mathcal{F}: \chi^{(1)} - a\chi^{(0)}\chi^{(0)} \quad (50a)$$

$$\mathcal{G}_1: \chi^{(0)}(\tau = 0.5) - 1 \quad (50b)$$

Substituting (45) yields

$$f_i: (i+1)x_{i+1}^j - a \sum_{l=0}^i x_l^j x_{i-l}^j, \quad i = 0, 1, \dots, k \quad (51a)$$

$$g_1: \left\{ \sum_{i=0}^{k+m} x_i^N(\delta)^i \right\} - 1 \quad (51b)$$

$$h_1^j: \sum_{i=0}^{k+m} x_i^j(\delta)^i - x_{i+1}^{j+1}(-\delta)^i, \quad j = 1, 2, \dots, N-1 \quad (51c)$$

for $m = 1$. To explain the last term in equation (51a) it is important to study a multiplication of two series. Suppose we want to calculate the product C of two series A and B . Multiplying the two series and re-collecting by power

$$\sum_{i=0}^{\infty} a_i t^i \sum_{j=0}^{\infty} b_j t^j = \sum_{k=0}^{\infty} \left(\sum_{l=0}^k a_l b_{k-l} \right) t^k = \sum_{k=0}^{\infty} c_k t^k \quad (52a)$$

yields

$$c_k = \sum_{l=0}^k a_l b_{k-l} \quad (52b)$$

This is the basic relation, which we will make use of extensively to calculate the powers and products of series.

The analytical solution of the DE is

$$\chi(\tau) = \frac{2}{2 + a(1 - 2\tau)} = \sum_{i=0}^{\infty} \left\{ \frac{a(2\tau - 1)}{2} \right\}^i \quad (53)$$

having a pole at $\tau = (2 + a)/2a$. We have chosen $a = 2$ to simplify the analytical solution

$$\chi(\tau) = \frac{1}{2} \left(\frac{1}{1 - \tau} \right) = \frac{1}{2} \sum_{i=0}^{\infty} \tau^i \quad (54)$$

The boundary condition is given at a distance from the singular point $\tau = 1$ to avoid numerical problems which could divert our attention.

Note that equations (51a) are non-linear because of the non-linearity of the DE.

This problem is also calculated for different numbers of elements ($N = 1, 2, \dots, 6$), and different orders of approximation ($k = 1, 2, \dots, 6$).

The exact coefficients for the Taylor series around τ_e are given by

$$x_i = \frac{1}{2}(1 - \tau_e)^{-(i+1)} \quad (55)$$

The normalized radius of convergence is known exactly:

$$R = \frac{2\tau}{0.5} = 4(1 - \tau_e) \in [4, 2] \quad (56)$$

The worst value of R is the smallest value $R = 2$. Exact values and estimations of the error are given in Table IV.

The estimated values of the error give a good impression of the accuracy. To demonstrate the use of the optimization we pose the next question. Suppose we allow a computing time $T = 1.4$

Table IV. Exact (number above) and estimated (number below) relative error of the numerical solution of the 2nd example

$k \backslash N$	1	2	3	4	5	6
1	2.8E - 2	1.0E - 2	5.0E - 3	2.9E - 3	1.9E - 3	1.3E - 3
	1.2E - 1	3.1E - 2	1.4E - 2	7.8E - 3	5.0E - 3	3.5E - 3
2	8.6E - 3	7.0E - 4	1.5E - 4	4.7E - 5	2.0E - 5	9.5E - 6
	3.8E - 2	5.5E - 3	1.7E - 3	7.2E - 4	3.7E - 4	2.2E - 4
3	2.8E - 3	3.8E - 4	8.9E - 5	3.0E - 5	1.3E - 5	6.2E - 6
	1.2E - 2	1.0E - 3	2.1E - 4	6.9E - 5	2.9E - 5	1.4E - 5
4	9.2E - 4	3.1E - 5	3.0E - 6	5.8E - 7	1.6E - 7	5.2E - 8
	4.1E - 3	1.9E - 4	2.8E - 5	6.8E - 6	2.3E - 6	9.2E - 7
5	3.1E - 4	1.5E - 5	1.7E - 6	3.3E - 7	9.1E - 8	3.1E - 8
	1.4E - 3	3.6E - 5	3.6E - 6	6.9E - 7	1.8E - 7	6.3E - 8
6	1.0E - 4	1.3E - 6	6.6E - 8	6.8E - 9	1.3E - 9	3.0E - 10
	4.6E - 4	7.0E - 6	4.9E - 7	7.1E - 8	1.5E - 8	4.4E - 9

kFlops. What is the maximum accuracy feasible, and for which combination of k and N do we achieve this?

The (normalized) radius of convergence is chosen to be the smallest value $R = 2$. A good estimation of the optimum combination (k_0, N_0) for $m = 1$ is found. The minimal error feasible is $\varepsilon \approx 10^{-5}$ at

Optimal values	Numerical values	
$N_0 = 3.3$	$N_{\text{num}} = 3$	
$k_0 = 5.1$	$k_{\text{num}} = 5$	
$T_0 = 1.4$ kFlops	$T_{\text{num}} = 1.2$ kFlops	(57)

6.3. The energy equation of a flat flame

The last problem to be studied is

$$\frac{\partial^2 \chi}{\partial \tau^2} - \gamma \frac{\partial \chi}{\partial \tau} = S(\chi) \quad (58a)$$

with boundary conditions

$$\begin{aligned} \chi(\tau = 0) &= 0.15 \\ \left. \frac{\partial \chi}{\partial \tau} \right|_{\tau=1} &= 0 \end{aligned}$$

The term on the right-hand side stands for

$$S(\chi) = \alpha(1 - \chi)^2 \exp[\beta(\chi - 1)] \quad (58b)$$

with $\alpha = -1848$, $\beta = 10$ and $\gamma = 1.49$.

The DE can be regarded as the energy equation of a flat (1D) flame, prescribing the non-dimensional temperature χ as a function of the non-dimensional distance to the burner τ . The first two terms in equation (58a) describe the conductive and convective transport of heat in the flame. $S(\chi)$ can be interpreted as a source term due to burning of the fuel.¹¹ The source term is strongly temperature dependent. No analytical solution is available. A comparison with the solution of a Finite Difference (FD) method is made.

This problem is chosen to show the method's comprehensive range of application in real-life problems and to demonstrate a way of dealing with terms in the DE which themselves depend on the variable itself, like the source term.

The general idea remains the same, except for $S(\chi)$. The source term is expanded in a Taylor series of τ indirectly:

$$S(\chi(\tau_e + t)) = S(\chi(\tau_e)) + \left. \frac{dS}{d\chi} \right|_{\tau_e} d\chi + \frac{1}{2!} \left. \frac{d^2 S}{d\chi^2} \right|_{\tau_e} (d\chi)^2 + \dots = \sum_{i=0}^{\infty} S_i d\chi^i \quad (59)$$

In this series the powers of $d\chi$ are replaced by their Taylor series expansions

$$\begin{aligned} \chi(\tau_e + t) &= \chi(\tau_e) + \left. \frac{d\chi}{d\tau} \right|_{\tau_e} t + \frac{1}{2!} \left. \frac{d^2 \chi}{d\tau^2} \right|_{\tau_e} t^2 + \dots \\ &= x_0 + x_1 t + x_2 t^2 + \dots \end{aligned} \quad (60)$$

The Taylor series expansion of $d\chi$ itself is written as

$$\begin{aligned} d\chi(\tau_e + t) &= \chi(\tau_e + t) - \chi(\tau_e) = \sum_{i=0}^{\infty} x_{i+1} t^{i+1} \\ &= t \sum_{i=0}^{\infty} x_{i+1} t^i = t \sum_{i=0}^{\infty} dx_i^1 t^i \end{aligned} \quad (61)$$

For ease of notation we define a new coefficient dx_i^p which depends on the original unknowns.

The coefficient dx_i^p is the i th coefficient in the series representing $(d\chi)^p$. The powers of $d\chi$ in series formulation are generated by repeated multiplying. This multiplying can be done systematically, which we will demonstrate. Each series $(d\chi)^p$ is calculated as the product of the preceding power $(d\chi)^{p-1}$ and $d\chi$. The recurrent relation reads

$$(d\chi)^p = (d\chi)^{p-1} d\chi \quad (62)$$

or using equations (52a) and (52b),

$$t^p \sum_{k=0}^{\infty} dx_k^p t^k = t^{p-1} \sum_{i=0}^{\infty} dx_i^{p-1} t \sum_{j=0}^{\infty} dx_j^1 \quad (63)$$

with

$$dx_k^p = \sum_{l=0}^k dx_l^{p-1} dx_{k-l}^1 \quad (64)$$

Determination of the powers of $d\chi$ is done according to the above recurrent relation. Formally we define the coefficient of $(d\chi)^0$ by

$$dx_i^0 = \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{if } i \neq 0 \end{cases} \quad (65)$$

because now we are able to write the finite set of equations very elegantly by

$$f_i: (i+2)(i+1)x_{i+2} - \gamma(i+1)x_{i+1} - \sum_{j=0}^i S_j dx_{i-j}^j \quad (66a)$$

$$g_1: \left(\sum_{i=0}^{k+m} x_i^1 (-\delta)^i \right) - 0.15 \quad (66b)$$

$$g_2: \left(\sum_{i=0}^{k+m} (i+1)x_{i+1}^N (\delta)^i \right) - 1 \quad (66c)$$

$$h_1^j: \sum_{i=0}^{k+m} x_i^j (\delta)^i - x_{i+1}^{j+1} (-\delta)^i \quad (66d)$$

$$h_2^j: \sum_{i=0}^{k+m-1} (i+1)(x_{i+1}^j (\delta)^i - x_{i+1}^{j+1} (-\delta)^i) \quad (66e)$$

using the truncated series as an approximation.

Note that the equations $f_i = 0$ are non-linear because of the non-linearity of the DE. To set up these equations we need the derivatives of S with respect to χ . For this example the derivatives are determined easily, but in general one needs a numerical method to obtain derivatives of a function without causing instability problems. In the problem discussed here we use analytical derivatives in order to avoid diversion from the main subjects of this paper.

Figure 6. The source term $S(\chi)$ for $0 \leq \chi \leq 1$

Table V. Comparison of approximations computed by a Finite Difference and the Taylor-series method for the 3rd example

τ	$x_0(\tau)$ F.D.	$x_0(\tau)$ Taylor
0.05	0.24191	0.24191
0.15	0.44164	0.44162
0.25	0.64761	0.64770
0.35	0.81522	0.81537
0.45	0.90993	0.90998
0.55	0.95274	0.95274
0.65	0.97249	0.97250
0.75	0.98308	0.98309
0.85	0.99021	0.99021
0.95	0.99660	0.99660

Solving the set of equations has to be done with sufficiently small elements in order to describe the source term properly. In Figure 6 the source term $S(\chi)$ is shown.

This problem has been calculated with $N = 10$ and $k = 2$ using under-relaxation to avoid uncontrolled behaviour of the iteration process. A comparison with the outcome of a FD method is shown in Table V.

7. NON-EQUIDISTANT ELEMENTS

7.1. Introduction

In many practical problems the solution of a one-dimensional differential equation (DE) is not as smooth as in the first three examples presented in Section 6. In these cases regions with different 'activity' of the solution can be distinguished. These differences in 'activity' find expression in the radius of convergence changing with the co-ordinate. Using equidistant elements, like we did in the first three examples, would be uneconomic in problems involving the radius of convergence changing over a wide range of magnitude. The accuracy would be worse in regions with a smaller radius of convergence.

These problems have different scales, that is the dependent variable changes over a distance much smaller than the scale of the system. In the additional example presented here, we will apply the Taylor series methods with non-equidistant elements to a problem exhibiting two extremely different scales. This example considers a stationary one-dimensional shock wave.

The Taylor series method requires considerable continuity within each element. This, however, will not be a serious problem in principle, since solutions of engineering problems described by a DE are always continuous. The reader should bear this in mind in spite of the fact that a solution could appear to be discontinuous at places where the radius of convergence changes suddenly and extremely, as in the example which is presented here.

7.2. Stationary shock wave

Consider the flow of an ideal gas through a duct. In a very small region the velocity changes from the upstream value v_1 to a downstream value v_2 . There is no change of velocity outside the region.

It can be shown¹² that, under the proper conditions, almost all of the changes of gas condition take place in an extremely thin region called a 'shock wave'.

The stationary one-dimensional equations of change for the system under consideration are

$$\frac{\partial}{\partial x}(\rho u) = 0 \quad (67)$$

$$\rho u \frac{\partial u}{\partial x} = -\frac{\partial p}{\partial x} + \frac{4}{3} \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) \quad (68)$$

$$\rho u \hat{C}_p \frac{\partial T}{\partial x} = \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + u \frac{\partial p}{\partial x} + \frac{4}{3} \mu \left(\frac{\partial u}{\partial x} \right)^2 \quad (69)$$

where part of the notation is taken from Reference 12. These equations may be written as a single relation containing only u and x as variables:

$$\frac{4}{3} \frac{\mu}{\rho_1 v_1} u \frac{\partial u}{\partial x} - \frac{\gamma + 1}{2\gamma} u^2 + \frac{C_{III}}{\rho_1 v_1} u = \frac{\gamma - 1}{\gamma} C_I \quad (70a)$$

$$\gamma = \frac{\hat{C}_p}{\hat{C}_v} \quad (70b)$$

$$C_I = \frac{1}{2}(v_1)^2 + \hat{C}_p T_1 \quad (70c)$$

$$C_{III} = p_1 + \rho_1 (v_1)^2 = \rho_1 ((v_1)^2 + RT_1/M) \quad (70d)$$

if we use the ideal gas law and assume the Prandtl number of the ideal gas to be $Pr = \frac{3}{4}$.

With dimensionless velocity $\phi = u/v_1 \in [0, 1]$ and the values

$$M = 2 \times 10^{-3} \quad [\text{kg mol}^{-1}] \quad (71a)$$

$$C_p = 1.039 \times 10^4 \quad [\text{J kg}^{-1} \text{K}^{-1}] \quad (71b)$$

$$\gamma = \frac{5}{3} \quad [-] \quad (71c)$$

$$\mu = 8.35 \times 10^{-6} \quad [\text{N s m}^{-2}] \quad (71d)$$

$$\rho_1 = 0.5216 \quad [\text{kg m}^{-3}] \quad (71e)$$

$$v_1 = 5.494 \times 10^3 \quad [\text{m s}^{-1}], \quad (71f)$$

$$T_1 = 697.15 \quad [\text{K}] \quad (71g)$$

representing supersonic ($Ma_1 = 2.5$) flow of hydrogen gas, equation (70a) can be written as

$$\phi \frac{\partial \phi}{\partial x} + B_1(\phi - 1)(\phi - 0.37) = 0 \quad (72a)$$

with

$$B_1 = -\frac{3(\gamma + 1)\rho_1 v_1}{8\gamma\mu} \quad (72b)$$

In addition the co-ordinate is scaled,

$$d\tilde{x} = \beta dx \quad (73)$$

with β as a scaling parameter being constant, but possibly different, for each element separately. Substitution in the DE (72a) yields

$$\beta \phi \frac{\partial \phi}{\partial \tilde{x}} + B_1(\phi - 1)(\phi - 0.37) = 0 \quad (74)$$

We will refer to the range of the x -co-ordinate as the 'physical domain', while the $\tilde{x}$ -range is referred to as the 'calculating domain'. Furthermore, the situation with $\beta = 1$ will be referred to as the 'unscaled' situation, despite the velocity scaling.

In this latter case the physical and calculating domains will be identical.

Note that from equations (74) and (72b) it can be seen that $\beta \approx 10^8$ is an interesting scaling value ($B_1 = -2.059 \times 10^8$), near the shock position, where ϕ and $\tilde{x}$ are of order 1. This value for β is interpreted as a very small region in the physical domain being stretched to map a wider region in the calculating domain, as will occur in the immediate neighbourhood of the shock. For a large distance from the shock, we have that $\phi \rightarrow 1$ or $\phi \rightarrow 0.37$, so that β becomes smaller.

For sufficiently small values of β , the first term of equation (74) may be neglected. Now the solution is constant, being determined by a polynomial of degree 2,

$$B_1(\phi - 1)(\phi - 0.37) = 0 \quad (75a)$$

with roots

$$\phi_1 = 1.0 \quad (75b)$$

$$\phi_2 = 0.37 \quad (75c)$$

which we will see are the asymptotic values of the solution, before and after the 'shock' region.

The analytical solution of equation (72a) can be found by integration,

$$\frac{(1 - \phi)}{(\phi - 0.37)^{0.37}} = K \exp(-B_1 \{1 - 0.37\} x) \quad (76)$$

where K is an integration constant. K can be solved by using the boundary condition. As boundary condition we take the middle of the shock, where $\phi = 1/Ma_1$, to be $x = 0$. The name 'boundary condition' is somehow a bit confusing because the condition is not referring to an actual boundary. 'Inner condition' might be a better name:

$$\phi(x = 0) = \frac{1}{Ma_1} \quad (77)$$

This gives for K ,

$$K = \frac{\left(1 - \frac{1}{Ma_1}\right)}{\left(\frac{1}{Ma_1} - 0.37\right)^{0.37}} \quad (78)$$

7.3. Consequences of scaling

Application of the Taylor series method will result in a set of non-linear equations which we solve by the Newton-Raphson method. Without scaling the co-ordinate properly, numerical singularity of the Jacobian occurs. Scaling the co-ordinate improves the condition of the Jacobian, as we will show in this section. We remind the reader that a regular matrix is well conditioned if all elements are of comparable magnitude.

The elements of the Jacobian come from two kinds of equations: first of all the equations representing the DE. Formally the scaled DE (74) is represented by

$$\tilde{f}_i(\tilde{\phi}_0, \tilde{\phi}_1, \dots, \tilde{\phi}_{i+m}) = 0, \quad i = 0, 1, \dots, k \quad (79)$$

with

$$\tilde{\phi}_i = \frac{1}{i!} \frac{\partial^{(i)} \phi}{\partial \tilde{x}^i} \quad (80)$$

For further explanation we remind the reader of the definition of the radius of convergence equation (27). It is easy to verify that the subsequent relations hold:

$$\tilde{\phi}_i = \phi_i \left(\frac{1}{\beta}\right)^i \quad (81a)$$

$$\frac{1}{\tilde{r}_k} = \frac{1}{\beta r_k} \quad (81b)$$

Equation (79) gives elements in the Jacobian of the form

$$\frac{\partial \tilde{f}_i}{\partial \tilde{\phi}_j} \quad j = 0, 1, \dots, i + 1 \quad (82)$$

For a fixed value of i (the same row in the Jacobian) the elements of the matrix can be approximated by using equation (27),

$$\frac{\partial \tilde{f}_i}{\partial \tilde{\phi}_0} (\tilde{r}_j)^j, \quad j = 0, 1, \dots, i + 1 \quad (83)$$

which causes numerical singularity of the Jacobian if the scaled radius of convergence $\tilde{r}_j (\approx \beta r_j)$ takes very small values. Applying the Taylor series method rigorously in the unscaled shock region ($\beta = 1$ and $\tilde{r} = r \approx 10^{-8}$) will cause such numerical singularity. As the reader already might suspect, this singularity will be omitted by scaling the distance, for each element, in a way so that

$$\beta_e r_j = \tilde{r} \quad (84)$$

will have a bounded value, close to unity, for a suitable j . Relation (84) justifies the treatment of β_e as an indicator of solution activity. It is easy to see that a large scaling parameter has to be used close to the shock region (very small radius of convergence), and outside this region very small values (or even no scaling) are appropriate (large radius of convergence).

Bounding $\tilde{r}$ causes the elements of one row, for a fixed value of i , to become of comparable magnitude. The elements in different rows are also of comparable magnitude, because the magnitudes of $\tilde{f}_i$ for different values of i are comparable. This can be verified by considering equation (79),

$$\tilde{f}_i = \frac{\partial^i \tilde{f}_0}{\partial \tilde{x}^i} = \sum_{j=0}^{i+m} \frac{\partial \tilde{f}_0}{\partial \phi_j} \frac{\partial^i \phi_j}{\partial \tilde{x}^i} = \sum_{j=0}^{i+m} \frac{\partial \tilde{f}_0}{\partial \phi_j} \phi_{j+i} \quad (85)$$

in which, for each term, the order of magnitude can be estimated by

$$|\tilde{f}_0| \left| \frac{\phi_{j+i}}{\phi_j} \right| = |\tilde{f}_0| \frac{|\tilde{r}_i|^i}{|\tilde{r}_{i+j}|^{i+j}} \quad (86)$$

using equation (27). From this we see that all $\tilde{f}_i$ are of comparable magnitude, because $\tilde{r}$ is close to unity.

Not only the equations representing the DE, but, of course, also the continuity equations and boundary conditions must be considered. These are the second kind of equations the elements of the Jacobian come from.

The scaled continuity equations and boundary conditions are relations of the form

$$\sum_{i=0}^{k+m-n} \frac{(n+i)!}{n!} \{ \tilde{\phi}_{i+n}^e (\tilde{\delta}_e)^i - \tilde{\phi}_{i+n+1}^e (\tilde{\delta}_{e+1})^i \} = 0 \quad (87)$$

with

$$\tilde{\delta}_e = \beta_e \delta_e \quad (88)$$

in which n indicates the equation considers the n th-derivative of the solution, and the e -index indicates the involved elements.

It is easy to explain the influence of scaling by considering one element e which needs excessively strong scaling ($\beta_e \gg 1$). In this case coupling of this scaled element also causes numerical singularity because of the row-dominating matrix element $(\beta_e \delta_e)^{k+m-n}$. This dominating element occurs at least in two rows in the same column, since either the element must be coupled to its two neighbour(s) or to one neighbour and in addition at least one over-all boundary condition must be satisfied.

This singularity occurs for excessively large values of $\beta_e \delta_e$. An example is scaling in the shock region ($\beta_e \approx 10^8$).

This singularity can also be omitted by adjusting the element size, according to equation (88), with $\tilde{\delta}_e$, just like $\tilde{r}$, an appropriate bounded value. We will use equidistant elements $\tilde{\delta}$ in the scaled situation.

We saw that scaling improves the condition of the Jacobian if the elements are redistributed accordingly. The values of $\tilde{r}$ and $\tilde{\delta}$, still to be specified, cannot be chosen freely because the criterion of convergence has to be satisfied,

$$\frac{\tilde{\delta}}{\tilde{r}} \leq 1 \quad (89)$$

This criterion, which is in agreement with equation (32), indicates that the element size should never exceed the radius of convergence.

Until now we did not choose a specific β value. There are, of course, several ways to specify the scaling.

The possibility we choose uses the DE (74) to specify the scaling factor

$$\beta = \left| \frac{B_1(\tilde{\phi}_0 - 1)(\tilde{\phi}_0 - 0.37)}{\tilde{\phi}_0 \tilde{\phi}_1} \right| \quad (90)$$

The precise element size $\tilde{\delta}$ is still to be determined. An optimization strategy, not the same, but similar to the one in Section 5, can provide us with $\tilde{\delta}$.

In this example we used a more arbitrary choice for $\tilde{\delta}$ because our main intention is to demonstrate the possibility of handling problems involving different scales with the Taylor series method.

7.4. Results

In our example we divide the calculation domain $\tilde{x} \in [-0.1, 0.1]$ into $N + 2$ equidistant elements. Of course, two of them will be boundary elements while N of them are inner elements. We want the final solution in element $N - 1$ to satisfy the boundary condition, cf. equation (77).

As initial guess we will provide the boundary elements with the asymptotic values from equations (75a) and (75b). For the inner elements we use a linear behaviour of the solution as initial guess, which even does not have to satisfy the 'inner' boundary condition. Furthermore, we start with $\beta = 1$ for each element. Updating of the β 's for the inner elements is done according to equation (90). The β -values for the boundary elements β_1 and β_{N+2} are updated to guarantee the range of the physical domain at both sides of the shock to be 0.1,

$$\beta_{N+1} \delta_{N+1} + \sum_{e=1}^N 2\beta_e \delta_e = 0.1 \quad (91a)$$

$$\beta_{N+1} \delta_{N+1} + 2\beta_{N+2} \delta_{N+2} = 0.1 \quad (91b)$$

This very simple scale updating is not the most elegant one but it does not harm the purpose of this demonstration.

The solving procedure is started with one update of the β -values according to equations (90) and (91). This updating will be performed only once, and further, the β -values will remain fixed. The Newton-Raphson method is used to improve the numerical approximation until the relative correction is small enough (10^{-13}). Only a few (< 10) Newton-Raphson iterations are necessary to find the final numerical solution.

To qualify the numerical solution a comparison is made with the analytical solution ϕ given by equation (76). We define an average relative error ε of the inner elements as

$$\varepsilon = \frac{1}{10N} \sum_{n=1}^{10N} \left| \frac{\phi(\tilde{x}_n) - \phi(\tilde{x}_n)}{\phi(\tilde{x}_n)} \right| \quad (92)$$

where, for each inner element, ten points $\tilde{x}_n$ are chosen equidistantly in order to calculate the relative error. Note that the boundary elements are not included in this definition because their dominating influence on the error (10^{-3}) is due to their scaling parameter being determined according to equation (91), which is rather *ad hoc*.

For $N = 5$ and $N = 10$ ($k = 0, 1, \dots, 5$) the error according to equation (92) is given in Table VI. Comparing the different situations, a similar influence of k , as in the first three examples of the previous section, can be observed. Increasing the order of approximation will increase the accuracy, which can be seen from Figures 7 and 8. For $k \geq 2$ the difference between the numerical and exact solution is hardly noticeable.

Furthermore, it is quite remarkable that the accuracy is almost independent of the number of (inner) elements N (Table VI). This is another phenomenon which is caused by the *ad hoc* scaling procedure for the two boundary elements. Because of this scaling method doubling the number of elements will not reduce $\tilde{\delta}$ (nor ε) but instead the inner elements will occupy a wider neighbourhood of the shock region, which can be seen from Figures 7 and 8. Until the region occupied by the inner elements approaches the range of the calculating domain, the boundary elements will maintain their dominating influence on the error, because their size remains almost the same. That is the reason why the boundary elements were excluded from the definition of ε in order to qualify the numerical solution in the neighbourhood of the shock.

Figure 7. The dimensionless velocity $\phi(x)$ with $N = 5$, $k = 0, 2, \infty$

Figure 8. The dimensionless velocity $\phi(x)$ with $N = 10$, $k = 0, 2, \infty$

Table VI. Average relative error

k	$N = 5$	$N = 10$
0	0.13×10^{-2}	0.95×10^{-3}
1	0.36×10^{-3}	0.18×10^{-3}
2	0.78×10^{-4}	0.40×10^{-4}
3	0.61×10^{-4}	0.18×10^{-4}
4	0.21×10^{-4}	0.50×10^{-5}
5	0.47×10^{-5}	0.87×10^{-6}

The required computing time (CPU in seconds) for one iteration is measured. The most time-consuming procedures are the assembling process of the Jacobian and solving (*LU*-decomposition for a banded system) the linearized set of equations, cf. equation (18).

For $N = 5$ and $N = 10$ ($k = 0, 1, \dots, 5$) the total relative computing time needed for both procedures together is shown in Table VII. The computing time in Table VII is relative to the required computing time for $k = 0$ and $N = 5$.

Table VII. Relative computing time

k	$N = 5$	$N = 10$
0	1.00	1.11
1	1.18	1.59
2	1.59	2.24
3	2.06	3.24
4	2.71	4.41
5	3.53	6.00

As one might expect, more computing time is required for a larger number of elements N and for a higher order k .

8. CONCLUSIONS

From Figures 1 and 2 we see that higher accuracy can be achieved by increasing the number of elements and by increasing the order of approximation, at the cost of more computing time needed. This is not very surprising, since higher accuracy and less computing time are contradictory demands.

More meaningful is the fact that an optimum between these two demands exists. An important result is that this optimum may be achieved, not by increasing the number of elements only, but by simultaneously increasing the order of approximation.

In comparison with low-order methods ($k = 1, 2$), a reduction of required computing time or a higher accuracy is achieved. Especially for high accuracy demands ($\varepsilon \ll 1$) in high-order DE ($m = 4$) these advantages are spectacular, as can be seen in Tables I and II.

This is an important feature, since the more conventional numerical methods are based on approximations of a fixed order and sufficient accuracy is achieved by increasing the number of elements only. For a proper base on which to make a more general conclusion in this direction one should make a thorough comparison of this power-series method with the more conventional methods. It is not the author's purpose to claim that the proposed method overrules other methods, but to attract the reader's attention to the possibility that it is worthwhile to increase the order of approximation also, rather than only increasing the number of elements more or less automatically.

An other important advantage is the possibility to estimate the error in the solution, not only by order but also by magnitude. This is rather unique since the more conventional numerical methods estimate the accuracy by recalculating the problem with a different number of elements or different order of approximation.

The analytical work which has to be done preceding each different type of DE may be regarded as a disadvantage of the method. However, within the field of engineering, the same type of DE underlies a wide class of problems. Consequently, it is necessary to perform the analytical part of the problem only once. This, in combination with the possibility to program some fundamental operators which occur frequently in the DE, weakens this disadvantage substantially.

In the last example we have seen how a realistic problem involving regions of extremely different activity (shock wave) can be handled using the Taylor series method. The hardly 'shocking' key to solve this problem is scaling in combination with element redistribution.

Increasing k the order of approximation will increase the accuracy of the solution, similar to the results of the previous three, more academic, examples.

Leaving the equidistant element distribution, we might as well choose an order of approximation for each element separately, which is a rather unusual combination. Non-equidistant elements in combination with non-constant order of approximation complicates the optimization problem extremely, as it will become a kind of 'travelling salesman' problem. Finding the global optimum will need special algorithms like the 'simulated annealing' technique described in Press.¹³ Alternatives such as local (i.e. elementwise) optimization as primary research have our preference.

We hope that we can take advantage of the Taylor series method in future, especially within the areas of heat-conduction and fluid-flow problems formulated in Body-Fitted-Co-ordinates, the object of our recent study.

ACKNOWLEDGEMENT

We gratefully acknowledge the comparative numerical calculations of H. C. de Lange.

REFERENCES

1. A. Gibbons, *Comp. J.*, **3**, 108 (1960).
2. D. Barton, *Comp. J.*, **14**, 243 (1972).
3. D. Barton, I. M. Willers and R. V. Zahar, in John Rice (ed.), *Mathematical Software*, Academic Press, New York, 1971.
4. L. Collatz, *The Numerical Treatment of Differential Equations*, Springer-Verlag, Berlin, 1966.
5. R. D. Small, *Int. j. numer. methods eng.*, **19**, 739-755 (1983).
6. R. D. Small, *MRC Technical Summary Report 1923*, 1979.
7. R. D. Small, *MRC Technical Summary Report 1924*, 1979.
8. R. D. Small, *MRC Technical Summary Report 1925*, 1979.
9. L. Kuipers and R. Timman, *Handboek der Wiskunde*, Scheltema & Holkema, Amsterdam, 1963.
10. G. H. Golub and C. F. van Loan, *Matrix Computations*, North Oxford Academic Publishing, Oxford, 1983.
11. L. P. H. de Goey and H. C. de Lange, Quasi one-dimensional flames, Internal Report, Eindhoven University of Technology, FST 89 GOE, 1989.
12. R. B. Bird, W. E. Stewart and E. N. Lightfoot, *Transport Phenomena*, Wiley, New York, 1960.
13. W. H. Press, *Numerical Recipes*, Cambridge University Press, Cambridge, 1989.