HERMIS '94

E.M.E. 1918

ΑΣΟΕΕ 1920

HELLENIC EUROPEAN RESEARCH ON MATHEMATICS AND INFORMATICS '94

Volume 2

Proceedings of the Second
Hellenic European Conference
on Mathematics and Informatics

22 – 24 September 1994, Athens, Hellas

Elias A. Lipitakis
Editor

HELLENIC MATHEMATICAL SOCIETY

HERMIS '94, Volume 2
E. A. Lipitakis (Editor)
1994, Hellenic Mathematical Society
Printed in Hellas

Numerical Solution of Boundary Value Problems for the Emden-Fowler Equations Using Extrapolation Methods

Pedro M. Lima

Departamento de Matemática, Instituto Superior Técnico

Av. Rovisco Pais ,1096 LISBOA CODEX (PORTUGAL)

Abstract

In the present paper we discuss a numerical method to solve boundary value problems for a class of ordinary second-order differential equations. The original nonlinear problem is reduced to a sequence of linear problems and these are solved by the finite-difference method. The convergence of the numerical results is accelerated by using extrapolation methods.

1.Introduction

Recently, we have analysed in [6] the use of convergence acceleration techniques to improve the accuracy of finite-difference schemes. There we considered a boundary-value problem (BVP) for a second-order linear differential equation on the interval [0,1]:

$$\frac{d}{dx}\left(k_1 \frac{du}{dx} \right) - k_2 u = f \qquad (1.1)$$

$$u(0)=u(1)=0, \qquad (1.2)$$

where the coefficients k_1 and k_2 are piecewise smooth functions in [0,1] and the right-hand side f is a function that is smooth in]0,1[but may be unbounded or have unbounded derivatives near the boundary. For the case when $f(x) = x^{\alpha}(1- x)^{\beta}$, α and β being real numbers, greater than -2, we have derived asymptotic expansions for the error of the usual finite-difference method and used the E-algorithm of Brezinski [1] to accelerate the convergence of the method. The performance of this scheme was illustrated by several numerical examples.

Although this method was not designed for any specific physical problem, it turns out that similar problems with singularities arise in many applications. For example, in [10] and [11] numerical methods for the Thomas-Fermi problem and other related BVP's have been considered. As it is pointed out in these papers, the main trouble when solving numerically that equation results from the existence of a degeneracy at the endpoint 0. Although the scheme we developped in [6] for linear problems with singularities is not directly applicable to that problem (the Thomas-Fermi equation is non-linear) , it is clear that the nature of the mathematical problem is essentially the same.

The aim of the present work is to generalize the scheme, developped in [6], to the case of non-linear degenerate BVP. As an important particular case, we shall consider a generalization of the Emden-Fowler equation with the form:

$$\frac{d^2y}{dt^2} + c\ t^p y^q = 0 \quad , t \in]\,0,\,1]\,, \tag{1.3}$$

where p, q and c are real numbers, p>-2, q>1 . We shall be concerned about the solution of (1.3) which satisfies the boundary conditions:

$$y(0)=1;\ \ y(1)= 0. \tag{1.4}$$

We are specially interested in the case when -2<p<0. In this case the equation is said to be degenerate. When q=3/2, p= − 1/2, c= − 1, (1.3) becomes the Thomas-Fermi equation, studied in [10] and [11]. Generally, the order of the degeneracy is (2,p), if -1<p<0, and (1,p) , if -2<p<-1 (according to the definitions given in [11]). The numerical scheme, presented in [11], is not applicable to this last case, when the convergence of the usual numerical methods is particularly slow. Therefore, the use of convergence acceleration techniques seems to be the best way to obtain accurate numerical solutions in this case.

As discussed in [11], the solution of the problem (1.3)-(1.4) may be numerically approximated by iterative processes, where each iterate is the solution of a linear BVP. This solution may be approximated by means of a finite-difference scheme. In Sec. 2, we describe the method used to solve (1.3)-(1.4) and present asymptotic expansions for the discretization error. In Sec.3, we use the E-algorithm of Brezinski to accelerate the convergence of the obtained numerical solutions, in some particular cases. The use of the the E-algorithm is based on the asymptotic expansions, presented in the previous section. The results for the case $p= -\frac{1}{2}$ are compared with the ones, obtained in [11].

2. Asymptotic Expansions for the Discretization Error

In order to approximate the solution of the non-linear BVP (1.3)-(1.4), we shall use well-known iterative schemes, based on the Piccard and the Newton methods [11]. This means that the solution of this problem is considered as the limit of a functions sequence, each of them being the solution of a linear BVP. In the case of the Piccard sequence, we start with an initial function , $y_0(x) \equiv 0$ or $y_0(x) = 1 - x$, and define each subsequent iterate as the solution of the following BVP:

$$y_\nu''(x) + c\,q\,x^p\,y_\nu(x) = x^p\left(y_{\nu\text{-}1}(x)^q - q\,y_{\nu\text{-}1}(x)\right), \qquad 0<x<1, \tag{2.1}$$

$$y_\nu(0)= 1\ ,\ y_\nu(1) =0\ ;\ \ \nu=1,2\ldots\,. \tag{2.2}$$

In order to approximate the solution of (2.1)-(2.2) , we select a set of n equispaced points, denoted $X_h=\{x_1, x_2, \ldots, x_n\}$, on the interval [0,1], and define $h=\frac{1}{n}$. Using central difference approximation we replace (3.1) and (3.2) by

$$\frac{1}{h^2}\,\delta_h^{\,2}\,\tilde{y}_\nu\,(x_i, h) + c\,q\;\tilde{y}_\nu(x_i, h)\,x_i^{\,p} = f\,(x_i\,,\,\tilde{y}_{\nu-1}(x_i\,,\,h)\,)\,,\quad i=1,2,\ldots,n\text{-}1 \tag{2.3}$$

$$\tilde{y}_\nu(0,h) = 1\;;\;\; \tilde{y}_\nu(0,1) = 0\,, \tag{2.4}$$

where

$$f(x\,,\,y) = x^p\,(\,y^q - q\,y); \tag{2.5}$$

and $\delta_h^{\,2}$ denotes the finite-difference operator defined by

$$\delta_h^{\,2}\,v(x_i,h) = v\,(x_{i+1}\,,\,h\,) - 2\,v(x_i\,,\,h) + v(x_{i-1}\,,\,h\,); \tag{2.6}$$

here and throughout the text, v(x,h) will denote a grid function, i.e., a function defined at points of the grid X_h. In particular, if v(x) is a function, defined on [0,1], then v(x,h) is the grid function resulting from the evaluation of v(x) at the points of the grid X_h. Let

$$\Phi_\nu(x,h) = \tilde{y}_\nu(x,h) - y_\nu(x)$$

be the error of the ν-th Piccard iterate, obtained by our method. In order to accelerate the convergence of the numerical results, obtained from the solution of (2.3)-(2.4), we must know an symptotic expansion of $\Phi_\nu(x,h)$, valid as $h\to 0$. In the regular case, when the solution of (2.1)-(2.2) is a sufficiently smooth function, the discretization error may be expanded in a series of even powers of the stepsize h. This problem is discussed in detail by Marchuk and Shaidurov [8], in the case of linear differential equations, and by Stetter [14], in a more general case. In the present case, however, due to the singularity of the solution at x=0, the discretization error cannot be represented as a power series in h. Therefore the simplest extrapolation algorithms, such as Richardson extrapolation [13], are not applicable to the solution of (2.3)-(2.4). In [7], we have analysed this problem in detail and obtained asymptotic expansions of $\Phi_\nu(x,h)$, in the following particular cases:

a) $p = -\frac{1}{2}$, $q = \frac{3}{2}$ (Thomas-Fermi equation);
b) $p = -1$, $q = 2$;
c) $p = -\frac{5}{4}$, $q = \frac{9}{4}$.

We shall now recall the main results obtained in that work, for these three cases. Note that if $p=\frac{m}{k}$, with $m,k \in N$, equation (2.1) may be reduced, by the substitution $x= t^k$, to the form

$$\frac{1}{k^2}\left(t^2\,\frac{d^2y}{dt^2} + (1-k)\,t\,\frac{dy}{dt}\right) + c\,q\,t^{2k-m}\,y = 0,\quad 0 < t < 1. \tag{2.7}$$

The general solution of this last equation may be found in the form of a series that converges on the interval [0,1] , by the Frobenius's method (see [4], [12]). Based on this fact, the form of the needed asymptotic error expansion may be obtained, using the same method folllowed by Mayers in [9], where he derived an asymptotic error expansion for a singular differential equation. This method was used by us for the derivation of all the formulae, presented below.

1) $p = -\frac{1}{2}$, $q=\frac{3}{2}$.

In this case, using the Frobenius's method, we have shown that the solution of the boundary value problem (2.1)-(2.2) may be represented in the form

$$y_\nu(x) = 1 + y_{2,\nu}x + \sum_{k=3}^{\infty} y_{k,\nu}\, x^{\frac{k}{2}} \quad , \quad \forall\, x \in [0,1], \quad \nu=1,2,... \tag{2.8}$$

Using this result and the properties of the finite-difference scheme (2.3)-(2.4), we have concluded that the discretization error, in this case, allows an asymptotic expansion with the form

$$\Phi_\nu(x,h) = C_{1,\nu}(x)\, h^{\frac{3}{2}} + C_{2,\nu}(x)\, h^2 + C_{3,\nu}(x)\, h^{\frac{5}{2}} + C_{4,\nu}(x)\, h^3 \ln h + O(h^3) \quad (\text{as } h\to 0) \tag{2.9}$$

2) $p = -1$, $q = 2$.

In this case, using the same method to obtain the solution of (2.1)-(2.2) in the form of a series, we have obtained

$$y_\nu(x) = 1 + y_{1,\nu}\, x \ln x + \tilde{y}_{1,\nu}\, x + P_{1,\nu}(\ln x)\, x^2 + \dots + P_{k-1,\nu}(\ln x)\, x^k + \dots, \tag{2.10}$$

where $P_k(y)$ is a polynome of degree not greater than k , with constant coefficients. From this result, we have derived the following asymptotic error expansion:

$$\Phi_\nu(x,h) = C_{1,\nu}(x)\, h \ln h + C_{2,\nu}(x)\, h + C_{3,\nu}(x)\, h^2 (\ln h)^2 + C_{4,\nu}(x)\, h^2 \ln h + O(h^2) \tag{2.11}$$

3) $p = -\frac{5}{4}$, $q=\frac{9}{4}$.

In this last case, usin g the same method as in a) and b), we came to the following representation of the Piccard iterates:

$$y_\nu(x) = \sum_{k=0}^{\infty} y_{3k,\nu}\, x^{\frac{3k}{4}} + x \sum_{k=0}^{\infty} y_{3k+4,\nu}\, x^{\frac{3k}{4}} = 1 + y_{3,\nu}\, x^{\frac{3}{4}} + y_{4,\nu}\, x + \dots \tag{2.12}$$

Further, we used (2.12) to show that the discretization error, in this case, may be expanded in the following form:

$$\Phi_\nu(x,h) = C_{1,\nu}(x)\, h^{\frac{3}{4}} + C_{2,\nu}(x)\, h + C_{3,\nu}(x)\, h^{\frac{3}{2}} + C_{5,\nu} h^{\frac{7}{4}} + O(h^2) \quad (\text{as } h\to 0) . \tag{2.13}$$

These are the main results that we use in the next section, where we shall apply the E-algorithm to accelerate the convergence of the numerical results.

3. Convergence Acceleration and Numerical Results

In this section we shall present some numerical results, obtained by applying a known extrapolation method to the solutions of the the finite-difference scheme (2.3)-(2.4). The iterates $\tilde{y}_\nu(x,h)$ were computed for a a series of different stepsizes $h_k = h_0/ 2^k$, $k=1,2,...,7$. For each value of h, the terms of the sequences $\{\tilde{y}_\nu(x,h)\}$ were computed untill the condition

$$\left\| \tilde{y}_{\nu+1}(x,h) - \tilde{y}_\nu(x,h) \right\|_2 = \left(\sum_{i=0}^{n} \left(\tilde{y}_{\nu+1}(x_i,h) - \tilde{y}_\nu(x_i,h)\right)^2 \right)^{\frac{1}{2}} \leq \epsilon \qquad (3.1)$$

was satisfied, for a given ϵ. In our computations, since we used double precision arithmetics, we set $\epsilon=10^{-14}$.

Defining the value of ν from (3.1), we may now consider, for each gridpoint x_i, a sequence

$$\tilde{y}_\nu(x_i,h_0), \quad \tilde{y}_\nu(x_i, h_1), \; ..., \quad \tilde{y}_\nu(x_i,h_7) \; . \qquad (3.2)$$

In our computations, we have used the following stepsizes

$$h_0=1/30, \quad h_k= h_{k-1}/2 \; , \; k=1,2,...,7 \; . \qquad (3.3)$$

The sequences (3.2) were used as the basis of the extrapolation process. When we apply an extrapolation algorithm we obtain, from the sequence (3.2), sucessive transforms. Each transform is a sequence that must converge to the same limit $y_\nu(x)$. If each transform converges with an order higher than the previous one, we say that the extrapolation algorithm accelerates the convergence of the original sequence.

When an expansion of the error is known, such as the ones obtained in Sec. 3, a natural way to accelerate the convergence of the sequence (3.2) is to use the E-algrithm of Brezinski [1],[2] and Havie [5]. This is a very general algorithm designed under the assumption that we know the asymptotic error expansion for a given sequence (S_n) :

$$S_n = S + a_1 g_1(n) + a_2 g_2 (n) + ... + a_k g_k(n) \; , \; n=0,1,2, \; ... \; , \qquad (3.4)$$

where $g_i(n)$ are predefined sequences, which satisfy the condition $\mathbf{g_{i+1}(n) = o(g_i(n))}$, **when** $\mathbf{n \to \infty}$. If r+1 terms of the sequence S_n are known, we can compute S by solving the linear system

$$S_{n+i} = S + a_1 g_1(n+i) + a_2 g_2(n+i) + ... + a_k \; g_k(n+i) \; , \; i=0,1,...,r. \qquad (3.5)$$

Usually the terms of S_n do not satisfy (3.4) exactly (to obtain (3.4) we must ignore the remainder of the asymptotic error expansion). Therefore,the solution of (3.5) is only an approximmation of S and depends on n and k. We shall denote this value by $E_k^{(n)}$ .The E-algorithm is a recursive way to compute $E_k^{(n)}$. Note that the usual algorithms to solve linear systems (such as gaussian elimination or LU factorization) are not recommendable to solve (3.5), because they are numerically unstable in this case. The computation of $E_k^{(n)}$, using the E-algorithm, starts with

$$E_0^{(n)} = S_n \ , \ n=0,1,2,\ldots,n_{max} \tag{3.6}$$

$$g_{0,i}(n) = g_i(n) \ , \ i=1,2,\ldots,n_{max} \ , \ n=1,2,\ldots,n_{max}-1 \ . \tag{3.7}$$

For $k=1,2,\ldots,n_{max}$ and $n=0,1,\ldots,n_{max}-k$ the recursive formulae are (see [1])

$$E_k^{(n)} = E_{k-1}^{(n)} + g_{k-1,k}^{(n)} \frac{E_{k-1}^{(n)} - E_{k-1}^{(n+1)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}} \ , \tag{3.8}$$

$$g_{k,i}^{(n)} = g_{k-1,i}^{(n)} + g_{k-1,k}^{(n)} \frac{g_{k-1,i}^{(n)} - g_{k-1,i}^{(n+1)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}} \ , \ i=k+1,\ k+2,\ldots,n_{max} \ . \tag{3.9}$$

Here $g_{k,i}^{(n)}$ are auxiliary sequences, which depend only on the terms $g_i(n)$ of the asymptotic expansion (3.4). To represent the extrapolation process, the values of $E_k^{(n)}$ are usually displayed in a double-entry array, with the form

$$\begin{array}{llll} E_o^{(0)} = S_0 \ ; & & & \\ E_o^{(1)} = S_1 \ , & E_1^{(0)} \ ; & & \\ E_o^{(2)} = S_2 \ , & E_1^{(1)} \ , & E_2^{(0)} \ ; & \\ E_o^{(3)} = S_3 \ , & E_1^{(2)} \ , & E_2^{(1)} \ ; & E_3^{(0)} \ ; \\ & \ldots & & \end{array} \tag{3.10}$$

The first column of this array is the initial sequence (3.2), whose convergence we want to accelerate (in our case, the sequence $\tilde{y}_\nu(x_i, h_k)$, $k=0,1,2,\ldots,7$. Each subsequent column contains a new transformed sequence. The performance of the E-algorithm as a convergence accelerator is evaluated by analysing the behaviour of the sucessive columns of the E-array.

In the next lines we present some numerical results, obtained for the boundary-value problem, considered in the previous sections, and comment these results. As done in the previous sections, we shall present these results separately for different values of p: $p=-\frac{1}{2}$, $p=-1$, $p=-\frac{5}{4}$.

a) $p=-\frac{1}{2}$, $q=\frac{3}{2}$.

The first numerical results we present here for the problem (1.3)-(1.4) were obtained in the case $p=-\frac{1}{2}$, $q=\frac{3}{2}$, c=-1 . In this case the BVP (1.3)-(1.4) is known as the Thomas-Fermi problem. The Piccard approximation of the solution of this problem which satisfies (3.1) was obtained with $\nu=15$ (in the case $y_0(x) \equiv 0$) or $\nu=14$ (if $y_0(x)=1-x$). In table 1 we present the values of $\tilde{y}_{15}(x_i,h)$,

$x_i = i/10$, $i=1,2,...,9$, obtained by the present method with different stepsizes . In this table we present also the best approximations obtained by the extrapolation method, in each gridpoint. In this case , the extrapolation process was based on the asymptotic error expansion (2.9).This means that the $g_i(n)$ sequences, in this case, have the form:

$$g_1(n) = (h_n)^{\frac{3}{2}} , g_2(n) = (h_n)^2 , g_3(n)=(h_n)^{\frac{5}{2}} , \quad ... \qquad (3.11)$$

Comparing the columns of the E-array for each x_i, we observe that the values of the first column ($E_o^{(n)}$) have 3 common decimal digits ; the terms of the column $E_2^{(n)}$ have 6 common digits ; the number of common digits in the column $E_4^{(n)}$ and in the remaining columns is 10. Based on this fact, we have taken the common digits of $E_4^{(n)}$ as the best approximations, obtained by extrapolation. The values in the last column of table 1 agree with the corresponding values, presented in [11]). The accuracy of the results obtained by the E-algorithm was compared with the accuracy, given by the ϵ-algorithm [3] and Richardson extrapolation. Since the exact solution is not known, we considered as a measure of the accuracy the minimal difference between successive terms of a given column:

$$\delta_k = \min_{i=0,7-k} \left| E_k^{(i+1)} - E_k^{(i)} \right| \qquad (3.12)$$

The results obtained by the E-algorithm, for each column, are the most accurate. In this case, we have, for the second column, $\delta_2 \approx 9 \times 10^{-11}$; for the fourth column, $\delta_4 \approx 1 \times 10^{-11}$; there is no improvement in the results of the remaining columns. Using the ϵ-algorithm, we can also obtain high accuracy. The corresponding values are $\delta_2 \approx 2 \times 10^{-9}$ and $\delta_4 \approx 3 \times 10^{-12}$. As it could be expected, the results given by the Richardson extrapolation are considerably less accurate. In this case, we have $\delta_2 \approx 3 \times 10^{-7}$, $\delta_4 \approx 1 \times 10^{-7}$.

x_i	N=60	N=480	N=3840	EXTRAP.
0.1	0.8497892482	0.8494898911	0.8494750906	0.8494743810
0.2	0.7275141383	0.7272451707	0.7272324534	0.7272318523
0.3	0.6195388560	0.6193058204	0.6192950221	0.6192945151
0.4	0.5206216119	0.5204239737	0.5204149288	0.5204145060
0.5	0.4277212100	0.4275577758	0.4275503624	0.4275500169
0.6	0.3388225043	0.3386922878	0.3386864222	0.3386861495
0.7	0.2525003851	0.2524027674	0.2523983962	0.2523981933
0.8	0.1677173342	0.1676520630	0.1676491563	0.1676490216
0.9	0.0837211353	0.0836882891	0.0836868348	0.0836867675

Table 1. Approximate values of the solution in the case $p=-\frac{1}{2}$, $q=\frac{3}{2}$ (Thomas-Fermi problem) at the gridpoints x_i=0.1, 0.2, ...,0.9 . Here and in the next tables **N** is the total number of gridpoints used for each approximation; the last column contains the best approximation of the solution, obtained by extrapolation using the E-algorithm.

b) p= − 1, q=2.

Next we have analysed the numerical results in the case **p= − 1, q=2, c= − 1.**

If we start the Piccard iteration process with $y_0(x) \equiv 0$, the approximation of the solution which satisfies (3.1) is obtained with $\nu=22$; if we start with $y_0(x) = 1 - x$, the approximation which satisfies (3.1) is obtained with $\nu=21$. In table 2 we present the values of $\tilde{y}_{21}(x_i,h)$, $x_i= i/10$, $i=1,2,...,9$, obtained by the present method with **h=1/60, h=1/480** and **h=1/3840.** In this table we also present the results obtained by extrapolation. The best approximations obtained have apparently 9-10 digits, depending on the considered gridpoints. In this case the asymptotic error expansion of the finite-difference method is given by (2.11). According to this results, the $g_i(n)$ sequences, in this case, should have the form:

$$g_1(n) = h_n \ln (h_n) \, , \, g_2(n) = h_n \, , \, g_3(n)=h_n^2(\ln h_n)^2 \, , \, g_4(n)=h_n^2 \ln h_n, \, g_5(n)= h_n^2,$$
$$g_6(n) =h_n^3(\ln h_n)^3 \, , \, g_7(n)=h^3 (\ln h_n)^2 \, ,... \tag{3.13}$$

In spite of this, it was observed that more accurate numerical results can be obtained, if we ignore the terms $g_1(n)$, $g_3(n)$ and $g_4(n)$. This indicates that the corresponding coefficients a_1,a_3 and a_4, on the right-hand side of (3.5), must be zero or close to zero. The results obtained using the E-algorithm were compared with the ones that are given by other extrapolation algorithms. Using again the value of (3.12) as a measure of accuracy, we have, for the E-algorithm, $\delta_2 \approx 2\times 10^{-10}$, $\delta_4 \approx 10^{-10}$; in the remaining columns of the E-array the results are not improved. In the case of the ϵ-algorithm, the corresponding values are $\delta_2 \approx 6\times 10^{-8}$, $\delta_4 \approx 7\times 10^{-10}$. If the Richardson extrapolation is used, we obtain $\delta_2 \approx 2\times 10^{-10}$, $\delta_4 \approx 9\times 10^{-11}$. The success of the Richardson extrapolation in this case is explained by the fact that the first logarithmic terms of the asymptotic error expansion may be ignored, as seen above.

x_i	N=60	N=480	N=3840	EXTRAP.
0.1	0.7856713601	0.7808375568	0.7802145946	0.780125259
0.2	0.6619833203	0.6580410620	0.6575404128	0.657468748
0.3	0.5620923700	0.5588211863	0.5584077292	0.55834858
0.4	0.4732135031	0.4704997105	0.4701574951	0.4701085515
0.5	0.3901168829	0.3878995908	0.3876203645	0.3875804363
0.6	0.3101516883	0.3083978723	0.3081772098	0.3081456595
0.7	0.2318377682	0.2305301570	0.2303657404	0.2303422339
0.8	0.1543208470	0.1534512950	0.1533420117	0.1533263885
0.9	0.0771207308	0.0766862143	0.0766316259	0.0766238223

Table 2. Approximate values of the solution in the case **p= − 1.0, q=2.0** at the gridpoints x_i=0.1, 0.2, ...,0.9 .

c) $p = -5/4$, $q=9/4$.

Finally we present the numerical results obtained in the case $p= -\frac{5}{4}$, $q=\frac{9}{4}$, $c=-1$.

In this case, the Piccard iterates satisfy condition (3.1) beginning with $\nu=28$ or $\nu=27$, if we start with $y_0(x) \equiv 0$ or $y_0(x)=1-x$, respectively . In table 3 we present the values of $\tilde{y}_{27}(x_i,h)$, $x_i= i/10$, $i=1,2,...,9$, obtained with diferent stepsizes, as in the previous examples. In this table we also present the results obtained by extrapolation using the E-algorithm. In this case the asymptotic error expansion of the finite-difference method is given by (2.13). This means that the $g_i(n)$ sequences, in this case, have the form:

$$g_1(n) = (h_n)^{\frac{3}{4}} , g_2(n) = h_n , g_3(n)=(h_n)^{\frac{3}{2}} , g_4(n) = h_n^{\frac{7}{4}} , g_5(n) =(h_n)^2, \; ... \; . \qquad (3.14)$$

The best approximations obtained have apparently 7-8 exact digits, depending on the considered gridpoints. Comparing with other extrapolation algorithms, the E-algorithm gives the most accurate results, with $\delta_2 \approx 3 \times 10^{-7}$, $\delta_4 \approx 9 \times 10^{-8}$ and $\delta_6 \approx 10^{-8}$. When we applied the ϵ-algorithm to the same numerical results, we obtained $\delta_2 \approx 1 \times 10^{-6}$, $\delta_4 \approx 2 \times 10^{-7}$ and $\delta_6 \approx 2 \times 10^{-8}$. In the case of the Richardson extrapolation, the convergence was not accelerated, as it can be seen form the following values: $\delta_2 \approx 6 \times 10^{-5}$, $\delta_4 \approx 5 \times 10^{-5}$, $\delta_6 \approx 5 \times 10^{-5}$.

x_i	N=60	N=480	N=3840	EXTRAP.
0.1	0.7243310984	0.7086484640	0.7052930535	0.70439
0.2	0.6058343675	0.5934905863	0.5908652928	0.5901638
0.3	0.5145680324	0.5044204038	0.5022648263	0.5016888
0.4	0.4340467335	0.4256413499	0.4238566193	0.4233797
0.5	0.3585507266	0.3516769786	0.3502177754	0.3498278
0.6	0.2855147462	0.2800702786	0.2789146600	0.2786059
0.7	0.2136540855	0.2095903303	0.2087278747	0.20849740
0.8	0.1423038323	0.1395999025	0.1390260983	0.13887276
0.9	0.0711327953	0.0697815237	0.0694947898	0.06941817

Table 3. Approximate values of the solution in the case $p=-\frac{5}{4}$, $q=\frac{9}{4}$ at the gridpoints x_i=0.1, 0.2, ... 0.9.

Acknowledgements. The numerical results presented in this section were obtained using the program EXTRAP01, whose author is M. Graça [6]. This program incorporates several subroutines, created by Brezinski and Redivo-Zaglia [3], allowing to apply the most known extrapolation algorithms to a given sequence.

Bibliography:

[1] C. Brezinski, A General Extrapolation Algorithm, Numer. Math. 35 (1980), p. 175-187.

[2] C.Brezinski, A Survey of Iterative Extrapolation by the E-algorithm, Det Kong. Norske Vid. Selsk. Skr. 2 (1989), 1-26.

[3] C.Brezinski and M. Redivo-Zaglia, Extrapolation Methods – Theory and Practice; North-Holland, Amsterdam, 1991.

[4] M.V. Fedoriuk, Asymptoticheskie Metody dlia Lineinykh Obyknovennykh Differentsialnykh Uravnenii (in Russian), Nauka, Moscow, 1983, p.13-15.

[5] T.Havie, Generalized Neville Type Extrapolation Schemes, BIT 19 (1979), 204-213.

[6] P.M. Lima and M.M. Graça (submitted to J. Comp. Appl. Math).

[7] P.M.Lima, Numerical Methods and Asymptotic Error Expansions for the Emden-Fowler Equations, preprint, Departamento de Matemática, IST, Lisbon, 1994.

[8] G.I.Marchuk and V.V.Shaidurov, Difference Methods and Their Extrapolations, Springer-Verlag, New York, 1983.

[9] D.F. Mayers, The Deferred Approach to the Limit in Ordinary Differential Equations, Comp. J. 7 (1964), p.54-57.

[10] J.W. Mooney, Solution of a Thomas-Fermi Problem Using Linear Approximants. Comp. Phys. Commun. 76 (1993), p.51-57.

[11] J.W. Mooney, Numerical Schemes for Degenerate Boundary-Value Problems, J. Phys A 26 (1993), p. L413-L421.

[12] A. Naifeh, Introduction to Perturbation Techniques, John Wiley & Sons, New York, 1981, chap.13.3.

[13] L. Richardson and J.Gaunt, The Deferred Approach to the Limit, Trans. Roy. Soc. V. A226, p.299-361.

[14] H.J.Stetter, Asymptotic Expansions for the Error of Discretization Algorithms for Nonlinear Functional Equations, Numer. Mat. 7 (1965), pp 18-31.