Advanced Mathematics: Computations and Applications

Proceedings of the International Conference
AMCA–95
Novosibirsk, Russia, 20–24 June, 1995

edited by

Anatoly S. ALEKSEEV

Computing Center
Russian Academy of Sciences
Siberian Branch
Novosibirsk, Russia

Nicolay S. BAKHVALOV

Institute of Numerical Mathematics
Russian Academy of Sciences
Moscow, Russia

NCC PUBLISHER, NOVOSIBIRSK, 1995

Yu.I. Kuznetsov
Complementary Jacobi matrix 422

M. Larin
Quadri–storage scheme for sparse Cholesky factor and tree–based implementation of Cholesky method 428

Y. Notay and Z. Ould Amar
Incomplete factorization preconditioning may lead to multigrid like speed of convergence 435

Garret Sobczyk
A universal geometric algebra 447

Workshop: Approximation and Applications **457**

V.N. Belykh
Overconvergence of numerical algorithms without saturation (on an example of elliptic problems) 458

V.N. Bojkov, S.R. Lyust, B.M. Shumilov
Use of parametrical spline smoothing in highways reconstruction 463

V.B. Cherepennikov
Estimates of accuracy of approximate solutions of some linear systems of differential equations by the method of functional parameters 467

R.Z. Dautov
$O(h^2 \ln^{\alpha} 1/h)$ accuracy in distance for free boundaries in finite element approximation to the obstacle problems 477

V.I. Lebedev
Extremal polynomials with restrictions and optimal algorithms 491

P.M. Lima and A.C. Lemos
Finite element solution of degenerate boundary-value problems for ordinary differential equations 503

V.D. Liseikin
Survey of grid generation technology 511

J.J.H. Miller, E. O'Riordan, and G.I. Shishkin
On the use of fitted operator methods for singularly perturbed partial differential equations 518

V.V. Smelov
On Demography Hypothesis and a Simple Mathematical Model of Population Development 532

V.S. Synakh
An approach to the spline collocation method 539

O.V. Ushakova
An efficient algorithm and program of generation of two-dimensional curvilinear optimal adaptive grids 542

Advanced Mathematics: Computations and Applications
A.S. Alekseev and N.S. Bakhvalov (Editors), pp. 503–510
© 1995 NCC Publisher

Finite element solution of degenerate boundary-value problems for ordinary differential equations

P.M. Lima and A.C. Lemos*

In the present paper, we consider some boundary-value problems for second order differential equations which are particular cases of the *generalized Emden-Fowler equation*. The solution of these problems has a singularity at the boundary. Each problem is reduced to a sequence of linear boundary-value problems, using the methods proposed by J. Mooney [5, 6]. Then we introduce an adequate substitution of the independent variable, which leads to a new equation, with a regular solution in the new variable. This equation is then solved numerically using a finite difference scheme or a finite element method with a cubic-spline basis. The accuracy of the numerical solution is compared with the results obtained by other methods [3, 5, 6].

1. Introduction

In recent papers of J. Mooney [5, 6] and P. Lima [3] the authors proposed some methods for the numerical solution of a boundary value problem (BVP) of the following form:

$$y'' = x^p y^q, \tag{1}$$

$$y(0) = 1,\ y(1) = 0; \tag{2}$$

where c, p and q are real constants, $p > -2$, $q > 0$. Problems of this form arise in many applications in Physics and Mechanics (see [9]). In the particular case, where $p = -1/2$, $q = 3/2$, the problem (1), (2) is the well-known Thomas–Fermi equation for the ionized atom [6].

The numerical methods proposed in [5] and [6] are based on the reduction of the problem (1), (2) to a sequence of linear BVP. The monotonicity and convergence of this sequence was studied in [4] According to the method, developped in [5] and [6], each linear problem is numerically solved by a three-point finite difference scheme. Due to the existence of a singularity of the solution at $x = 0$, the numerical solution of the problem becomes difficult in the case $p < 0$, the convergence of the usual finite difference schemes being very slow. In the cited papers, the author overcomes this problem using a special finite difference scheme, which takes account of the behaviour of the solution in the neighborhood of the origin. This scheme has second order convergence in spite of the singularity, and the accuracy of the results was improved by means of the Richardson extrapolation [8], giving 9 correct decimal digits.

In [3], the linear problems are solved by a usual three-point finite difference scheme. The numerical results obtained by this method are then extrapolated using the E-algorithm [2], which is a generalization of the Richardson extrapolation. Unlike the Richardson extrapolation, this algorithm enables convergence acceleration when the solution has singularities, provided that the form of the asymptotic error expansion is known. The results obtained in

*Departamento de Matemática Instituto Superior Técnico Av.Rovisco Pais, 1096 Lisboa, Codex, Portugal.

[3] using this method have 10 correct digits in the case $p = -1/2$, $q = 3/2$. Besides, this method also gave accurate results (up to 8 correct digits) in the cases $p = -1$, $q = 2$ and $p = -5/4$, $q = 9/4$, which were not considered in [5].

In the present work we analyse numerical methods for the considered BVP, based on the use of a variable substitution which reduces the original equation to a BVP without singularities. This approach enables us to approximate the solution of the considered BVP by two different methods:

- second-order finite difference scheme;
- finite element method, using a cubic spline basis.

In Section 2, we review the iterative schemes used to reduce (1), (2) to a sequence of linear problems and introduce in each linear problem the variable substitution.

In Section 3, we shortly describe the numerical methods, used to approximate the problem in the new variable, and present the results obtained by these methods.

Finally, in Section 4 we analyse the numerical results and compare their accuracy with the one of the results obtained previously. This analysis leads us to some conclusions about the efficacy of the considered methods.

2. Iterative scheme and variable substitution

One of the methods used to solve the problem (1), (2), which is discussed in [5], is based on the reduction of that problem to a sequence of linear problems, using the Picard or the Newton iterative scheme. When the Picard scheme is used, we construct a sequence of functions $\{y_\nu(x)\}_{\nu \in N}$, each of them is called a Picard iterate and is the solution of the following BVP:

$$y_\nu''(x) - qx^p\, y_\nu(x) = x^p\,[y_{\nu-1}(x)^q - q\, y_{\nu-1}(x)]\,, \tag{3}$$

$$y_\nu(0) = 1, \quad y_\nu(1) = 0, \qquad \nu = 1, 2, \ldots\,. \tag{4}$$

It is a well-known result [4] that the sequence $\{y_\nu(x)\}_{\nu \in N}$, defined by the iterative scheme (3), (4) converges upwards to the exact solution of the non-linear problem if the initial approximation is $y_0(x) = 0$; and it converges downwards, if $y_0(x) = 1 - x$.

When the Newton method is used, we obtain the follwing iterative scheme:

$$y_\nu''(x) - qx^p\, y_{\nu-1}(x)^{q-1} y_\nu(x) = (1-q)x^p\, y_{\nu-1}(x)^q, \qquad \nu = 1, 2, \ldots, \tag{5}$$

with the boundary conditions (4). In [4] it is also proved that the sequence, defined by (5), converges downwards to the exact solution of (1), (2), if $y_0(x) = 1 - x$. Hence the original problem may be reduced to a sequence of linear BVP, either by the scheme (3), (4) or (5), (4). The numerical solution of a problem of this kind may be obtained by a finite difference scheme. Such method is easy to program and may be implemented in a personal computer. However, due to the singularity of the solution at $x = 0$, in order to obtain accurate results, in [5] and [6], the author had to use a special finite difference scheme, which takes account of the behaviour of the solution near the boundary. This scheme provides second-order convergence, when $-1 < p < 0$. In [3] a different method was applied to obtain the numerical solution of (3), (4) and (5), (4). In that work the author used an ordinary three-point finite difference scheme, whose order of convergence is only $2 + p$, for the given problem. The accuracy of the numerical results obtained by this scheme was then improved by means of the E-algorithm. After deriving asymptotic error expansions, the author used the E-algorithm to extrapolate

the numerical results and obtained 10 correct digits, in the case $p = -1/2$, $q = 3/2$, and 7 correct digits, in the case $p = -5/4$, $q = 9/4$.

In many cases, however, it may be difficult to obtain an asymptotic error expansion. Besides, the form of the expansion may be very complicated, and in this case the E-algorithm may be not very efficient.

In the present paper, we use a different method to solve the linear problems (3), (4) and (5), (4). Suppose that p is a rational number, that is $p = -m/n$, where m, n are natural numbers. Then if we introduce in (3) the variable substitution

$$x = t^n, \tag{6}$$

and divide both sides by t^{2n}, we obtain the following equation:

$$\frac{1}{n^2}\left[t^2\frac{d^2y_\nu}{dt^2} + (1-n)t\frac{dy_\nu}{dt}\right] - qt^{2n-m}y_\nu = t^{2n-m}\left(y_{\nu-1}^q - q\,y_{\nu-1}\right), \quad \nu = 1,2,\ldots . \tag{7}$$

In the same way, if we introduce the variable substitution (6) in (5), we obtain

$$\frac{1}{n^2}\left[t^2\frac{d^2y_\nu}{dt^2} + (1-n)t\frac{dy_\nu}{dt}\right] - qt^{2n-m}y_{\nu-1}^{q-1}y_\nu = (1-q)t^{2n-m}y_{\nu-1}^q, \quad \nu = 1,2,\ldots . \tag{8}$$

The boundary conditions (4) remain unchanged in the case of the substitution (6). Therefore, the solutions of (7) and (8) that satisfy the boundary conditions (4) are, respectively, the Picard and Newton iterates for the problem (1), (2) in the new variable. Note that in the cases $p = -1/2$ and $p = -5/4$ these solutions are regular functions of t, that may be expanded in power series in t, when $t \in [0,1]$. Hence the solution of these problems may be accurately approximated by the usual methods. In the present paper, we consider a finite difference and a finite element method that are described in the next section.

3. Numerical methods and results

3.1. Finite difference method

In order to apply the finite difference method, let us consider an uniform grid with stepsize $h = 1/N$ on the interval $[0;1]$. Let us denote t_i the knots of this grid; then we have

$$t_i = i\,h, \qquad i = 1,2,\ldots,N.$$

Let us denote by $y_\nu(x,h)$ the approximmate solution of (7). As usual, we shall approximmate the derivatives of the solution by the central differences $\delta\,y_\nu$ and $\delta^2\,y_\nu$, where

$$\delta y_\nu(t_i,h) = \frac{y_\nu(t_{i+1},h) - y_\nu(t_{i-1},h)}{2h},$$

$$\delta^2 y_\nu(t_i,h) = \frac{y_\nu(t_{i+1},h) - 2y_\nu(t_i,h) + 2y_\nu(t_{i-1},h)}{h^2}.$$

Then the discretization of (7) leads us to the following system:

$$\frac{1}{n^2}t_i^2\delta^2 y_\nu(t_i,h) + (1-n)t_i\,\delta y_\nu(t_i,h) - qt_i^{2n-m}y_\nu(t_i,h)$$

$$= t_i^{2n-m}\left(y_{\nu-1}(t_i,h)^q - q\,y_{\nu-1}(t_i,h)\right), \qquad i = 1,2,\ldots,N-1, \quad \nu = 1,2,\ldots . \tag{9}$$

Analogously, the equations of the Newton iterative scheme may be approximated by the following linear systems

$$\frac{1}{n^2}\left[t_i^2\delta^2 y_\nu(t_i,h)+(1-n)t_i\delta y_\nu(t_i,h)\right]-qt_i^{2n-m}y_{\nu-1}(t_i,h)y_\nu(t_i,h)$$
$$=(1-q)t_i^{2n-m}y_{\nu-1}(t_i,h)^q,\qquad i=1,2,\ldots,N-1,\quad \nu=1,2,\ldots. \tag{10}$$

The boundary conditions for the finite difference schemes (9) and (10) are obtained directly from (4):

$$y_\nu(0,h)=1,\qquad y_\nu(1,h)=0. \tag{11}$$

The schemes (9), (11) and (10), (11) have *second order of convergence*, provided that the exact solutions of the corresponding BVP are of the class $C^4([0,1])$.

In the present work, we have carried out computations for the following cases:

$$m=1,\quad n=2,\quad q=3/2\quad \text{(Thomas-Fermi problem)}; \tag{12}$$
$$m=5,\quad n=4,\quad q=9/4\quad (p=-5/4). \tag{13}$$

In both these cases, we have verified that the method has convergence of the second order. Some numerical results are displayed in Table 1 and Table 2. We present the values of the approximate solution with respect to the old variable x,so that we can compare these results with the ones, presented in other works. In order to compute the value of $y_\nu(t,h)$ when $t=x_i$, we had to use the values of $y_\nu(t_j,h)$ and $y_\nu(t_{j+1},h)$, where t_j and t_{j+1} are such that $t_j\le x_i^{1/n}\le t_{j+1}$. Then $y_\nu(x_i,h)$ is evaluated by linear interpolation.

Table 1. Numerical results obtained by the finite difference method for the Thomas–Fermi equation (case (12)). $y^1(x_i)$ denotes the corresponding values given in [6]

$x_i=t_i^2$	$y^1(x_i)$	$y_\nu(x_i,h)$ $N=50$	$y_\nu(x_i,h)$ $N=100$	$y_\nu(x_i,h)$ $N=200$	$y_\nu(x_i,h)$ $N=400$
0.1	.849474382	.849329385	.849431448	.849464696	.849471623
0.2	.727231852	.727081847	.727195974	.727222226	.727229975
0.3	.619294515	.619159178	.619263660	.619285914	.619292880
0.4	.520414506	.520292372	.520386340	.520406690	.520413249
0.5	.427550017	.427439149	.427523544	.427542905	.427548595
0.6	.338686150	.338591308	.338659461	.338681958	.338684884
0.7	.252398194	.252326339	.252374840	.252392354	.252396730
0.8	.167649022	.167568435	.167625496	.167645763	.167647868
0.9	.083686767	.083598342	.083675130	.083682279	.083685371

3.2. Finite element method

An approximate solution of (1), (2) may be also obtained by combining the Piccard iterative scheme with the finite element method. In this case, the solution of (7) is approximated by a function $u_\nu^N(t)$ with the form:

$$u_\nu^N(t)=\sum_{i=0}^{N}C_i^\nu B_i^N(t)+\bar{B}_0(t), \tag{14}$$

where B_i are given basis functions, $i=0,1,\ldots,N$; $\bar{B}_0(t)$ is a known function that satisfies the boundary condition at $t=0$; and C_i are unknown coefficients. In this work we use as basis functions the so called *cubic B-splines* which are piecewise polynomial functions of degree 3,

Table 2. Numerical results obtained by the finite difference method for the Emden–Fowler equation in the case (13)). $y^1(x_i)$ denotes the corresponding values given in [6]

$x_i = t_i^4$	$y^1(x_i)$	$y_\nu(x_i,h)$ $N=50$	$y_\nu(x_i,h)$ $N=100$	$y_\nu(x_i,h)$ $N=200$	$y_\nu(x_i,h)$ $N=400$
0.1	.704396	.704405851	.704394191	.704394799	.704396749
0.2	.5901638	.590114613	.590165055	.590162246	.590163079
0.3	.5016888	.501745943	.501702591	.501692007	.501689508
0.4	.4233797	.423310412	.423348527	.423381504	.423379620
0.5	.3498278	.349856967	.349825791	.349823383	.349825486
0.6	.2786059	.278668813	.278620032	.278608624	.278606173
0.7	.20849740	.208304619	.208430710	.208496491	.208496145
0.8	.13887276	.138618351	.138795483	.138864135	.138868741
0.9	.06941817	.069115716	.069330067	.069404143	.069412686

such that $B_i^N(t) = 0$, if $t \notin [t_{i-2}, t_{i+2}]$; $B_i^N(t) \in C^2([0,1])$; $i = 0, 1, \ldots, N$. For the properties of the B-splines, see [7].

According to the Galerkin method (see [1] or [7]), the coefficients C_i that define the approximate solution $u_\nu^N(t)$ may be computed by solving the linear system:

$$\sum_{i=0}^{N} C_i^\nu \langle LB_i^N(t), B_j^N(t)\rangle + \langle \bar{B}_0, B_j^N\rangle = \langle f(u_{\nu-1}^N, t), B_j^N(t)\rangle, \qquad j = 0, 1, \ldots, N, \tag{15}$$

where L denotes a linear operator such that

$$Lu(t) = \frac{1}{n^2}\left[t^2 \frac{d^2u}{dt^2} + (1-n)\frac{du}{dt}\right] - qt^{2n-m}u(t),$$

the angular parenthesis denote the scalar product in the usual sense

$$\langle u, v\rangle = \int_0^1 u(t)\, v(t)\, dt,$$

and

$$f(u,t) = t^{2n-m}(u(t)^q - q\, u(t)).$$

The linear system of equations, given by (15), may be written under the matricial form $A^{(N)}u_\nu^N x = b^{(N)}$ where the entries of the matrix $A^{(N)}$ are

$$a_{ij}^{(N)} = \langle LB_i^n, B_j^N\rangle, \qquad i, j = 1, \ldots, N+1,$$

and the entries of $b^{(N)}$ are

$$b_j^{(N)} = \langle f(u_{\nu-1}^N, t), B_j^N\rangle - \langle \bar{B}_0, B_j^N\rangle, \qquad j = 1, \ldots, N+1.$$

In the present work, all the integrals arising in the entries of $A^{(N)}$ and $b^{(N)}$ were evaluated by using a Gaussian quadrature rule, with three knots in each interval $[t_i, t_{i+1}]$. As known, due to the properties of the B-splines, $A^{(N)}$ is a band matrix, whose entries satisfy

$$a_{ij}^{(N)} = 0, \qquad \text{if} \quad |i-j| > 3.$$

After computing $A^{(N)}$ and $b^{(N)}$, the linear system (15) may be solved by a factorization method, taking into acount that $A^{(N)}$ is a band matrix. Since $A^{(N)}$ does not depend on ν, in each iteration we only have to update the entries of $b^{(N)}$ and solve the linear system again.

In Tables 3–6 we present some numerical results obtained by the finite element method for the cases (12) and (13). Note that all the computations in the present work were carried out in double precision arithmetics. The algorithms were coded in FORTRAN for personal computers. In a PC with a 486sx processor, the computation by the finite element method with $N = 100$ takes about 100 sec.; for the same N, it takes about 15 sec., if we use the finite difference method.

Table 3. Numerical results obtained by the finite element method for the first Picard iterate of Thomas–Fermi equation (case (12)), with $y_0(x) = 0$. $y^1(x_i)$ denotes the corresponding values given in [6]

$x_i = t_i^2$	$y^1(x_i)$	$y_1^N(x_i)$ $N = 10$	$y_1^N(x_i)$ $N = 20$	$y_1^N(x_i)$ $N = 40$	$y_1^N(x_i)$ $N = 100$	$y_1^N(x_i)$ $N = 200$
0.1	.816341088	.816343943	.816341302	.816341097	.816341087	.816341087
0.2	.676635195	.676641250	.676635125	.676635190	.676635195	.676635195
0.3	.560264866	.560270228	.560264773	.560264859	.560264865	.560264865
0.4	.459468617	.459472246	.459468798	.459468623	.459468617	.459468617
0.5	.369682481	.369681793	.369682433	.369682485	.369682481	.369682480
0.6	.287802887	.287803964	.287803102	.287802876	.287802887	.287802887
0.7	.211536747	.211540246	.211536818	.211536760	.211536747	.211536747
0.8	.139089613	.139087375	.139089437	.139089611	.139089613	.139089613
0.9	.068993124	.068995589	.068992918	.068993108	.068993124	.068993124

Table 4. Numerical results obtained by the finite element method for the Thomas–Fermi equation (case (12)), after 14 iterations. $y^1(x_i)$ denotes the corresponding values given in [6]. The values displayed in the last column were obtained by means of the Richardson extrapolation, using all the considered grids

$x_i = t_i^2$	$y^1(x_i)$	$y^N(x_i)$ $N = 50$	$y^N(x_i)$ $N = 100$	$y^N(x_i)$ $N = 200$	$y^N(x_i)$ $N = 400$	Extrap.
0.1	.849474382	.849820425	.849646506	.849560221	.849517246	.85947438
0.2	.727231852	.727764893	.727496978	.727364070	.727297875	.73723185
0.3	.619294515	.619921433	.619606322	.619450009	.619372160	.61929451
0.4	.520414506	.521066649	.520738848	.520576248	.520495270	.52041450
0.5	.427550017	.428173479	.427860084	.427704638	.427627225	.42755002
0.6	.338686150	.339237708	.338960449	.338822932	.338754449	.33868615
0.7	.252398194	.252843415	.252619604	.252508601	.252453323	.252398193
0.8	.167649022	.167961476	.167804403	.167726503	.167687710	.167649022
0.9	.083686767	.083847966	.083766931	.083726741	.083706727	.08368676

4. Conclusions

The numercial experiments carried out in this work have shown that the use of the variable substitution $x = t^n$ has significant advantages for the discretization of the considered equations:

- When the finite difference method is applied, with a given stepsize h, we can obtain higher accuracy when we solve the equations (7) or (8), than if we solve the original

Table 5. Numerical results obtained by the finite element method for the first Picard iterate of the Emden–Fowler equation in the case (13)), with $y^0(x_i) = 0$

$x_i = t_i^4$	$y_1^N(x_i)$ $N = 50$	$y_1^N(x_i)$ $N = 100$	$y_1^N(x_i)$ $N = 200$	$y_1^N(x_i)$ $N = 400$
0.1	.453278539	.453278661	.453278668	.453278669
0.2	.314245470	.314245529	.314245534	.314245534
0.3	.234576687	.234576760	.234576765	.234576765
0.4	.180038619	.180038664	.180038666	.180038667
0.5	.138685111	.138685164	.138685167	.138685167
0.6	.104942855	.104942905	.104942908	.104942908
0.7	.0757646544	.0757646734	.0757646726	.0757646728
0.8	.0492988965	.0492988978	.0492988964	.0492988966
0.9	.0243283116	.0243282956	.0243282937	.0243282937

Table 6. Numerical results obtained by the finite element method for the Emden–Fowler equation in the case (13)), after 16 iterations. $y^1(x_i)$ denotes the corresponding values given in [3]. The values displayed in the last column were obtained by means of the Richardson extrapolation, using all the considered grids

$x_i = t_i^4$	$y^1(x_1)$	$y^N(x_i)$ $N = 50$	$y^N(x_i)$ $N = 100$	$y^N(x_i)$ $N = 200$	$y^N(x_i)$ $N = 400$	Extrap.
0.1	.704396	.707605141	.705991291	.705191496	.704793370	.7043964
0.2	.5901638	.593958816	.592049484	.591103728	.590633061	.5901638
0.3	.5016888	.505553583	.503608644	.502645636	.502166473	.5016889
0.4	.4233797	.427034503	.425194859	.424284252	.423831231	.4233797
0.5	.3498278	.353089349	.351447394	.350634846	.350230658	.3498278
0.6	.2786059	.281345147	.279965917	.279283511	.278944092	.2786058
0.7	.2084974	.210622931	.209552624	.209023148	.208759814	.20849741
0.8	.1388728	.140321712	.139592058	.139231127	.139051627	.13887277
0.9	.0694182	.070151737	.069782314	.069599588	.069508716	.0694182

equations (3) or (5). For example, with $N = 100$, we obtain obtain about 4–5 correct digits, both in case (12) and (13) (see Table 1 and Table 2). Note that the results presented in [3], obtained with the same N, without variable substitution, have only 3 correct digits, in case (12), and 2 correct digits, in case (13).

- The variable substitution enables the use of the Richardson extrapolation to accelerate the convergence of the results, obtained by the finite difference or finite element method. When the variable substitution is not applied, the convergence acceleration may be obtained by more general extrapolation algorithms (such as the E-algorithm). However, in this case, more operations are required to obtain the same acuracy.

Concerning the accuracy of the finite element method, we have remarked that when we use it to compute the first Picard iterate, i.e., to solve (7) with $\nu = 0$, we can obtain very accurate results. Using $N = 200$, we obtain about 10 correct digits in the case (12) (see Table 3), and 8 correct digits, in the case (13) (see Table 5). However, the convergence of the method is not so fast when we consider the last computed iterate. In this case, with $N = 100$, we cannot obtain more than 3 correct digits, in the case (12) (see Table 4), and 2 correct digits in the case (13) (see Table 6). The reason for this slow convergence may be the errors in the computation of the integrals arising in the right-hand side of the system. However, using the Richardson extrapolation, we can improve the accuracy up to 9 correct digits, in the case (12), and to 7 correct digits, in the case (13).

References

[1] O. Axelsson and V.A. Barker, *Finite element solution of boundary value problems – theory and computation*, Academic Press Inc, USA, 1984.

[2] C. Brezinski, *A general extrapolation algorithm*, Numer. Math., **35**, 1980, 175–187.

[3] P.M. Lima, *Numerical methods and asymptotic error expansions for the Emden–Fowler equations*, submitted to J. Comp. Appl. Math.

[4] J.W. Mooney, *Monotone methods for the Thomas-Fermi equation*, Quart. Appl. Math., **36**, 1978, 305–314.

[5] J.W. Mooney, *Numerical schemes for degenerate boundary-value problems*, J.Phys., **A26**, 1993, L413–L421.

[6] J.W. Mooney, *Solution of a Thomas-Fermi problem using linear approximants*, Comp. Phys. Commun., **76**, 1993, 51–57.

[7] P.M. Prenter, *Splines and Variational Methods*, John Wiley & Sons, New York, 1975.

[8] L. Richardson and J. Gaunt, *The deferred approach to the limit*, Trans. Roy. Soc. (London), **A226**, 1927, 299–361.

[9] J.W. Wong, *On the generalized Emden-Fowler equation*, SIAM review, **17**, 1975, 339–360.