A PARALLEL CODE FOR MULTIPRECISION COMPUTATIONS OF THE LANE–EMDEN DIFFERENTIAL EQUATION

Vassilis S. Geroyannis\textsuperscript{1}, Vasileios G. Karageorgopoulos\textsuperscript{2}
\textsuperscript{1,2}Department of Physics, University of Patras, Patras, Greece
\textsuperscript{1}vgeroyan@upatras.gr, \textsuperscript{2}vkarageo@upatras.gr

April 28, 2016

Abstract

We compute multiprecision solutions of the Lane–Emden equation. This differential equation arises when introducing the well-known polytropic model into the equation of hydrostatic equilibrium for a nondistorted star. Since such multiprecision computations are time-consuming, we apply to this problem parallel programming techniques and thus the execution time of the computations is drastically reduced.

Keywords: Lane–Emden equation; multiprecision computations; numerical methods; parallel programming; stellar polytropic models

1 Introduction

The aim of this study is to compute multiprecision solutions of the Lane–Emden equation of stellar hydrodynamics by a code implementing the Runge–Kutta–Fehlberg method of fourth and fifth order (see e.g. [1], Sec. 17.3.4.4; see also [2], Sec. 2), working in the multiprecision environment of the “Fortran–90 Multiprecision System”, abbreviated MPFUN90, written by D. H. Bailey ([3, 4] and references therein) — available in http://crd-legacy.lbl.gov/~dhbailey/mpdist/, and licensed under the Berkeley Software Distribution License found in that site. Such highly accurate solutions can be used for checking other numerical codes and prescribing a measure of their accuracy.

Since however multiprecision computations are time-consuming, we proceed with applying parallel programming techniques, appropriate for multicore machines. The Open Multi-Processing (OpenMP; http://openmp.org/) is an Application Program Interface (API) supporting shared-memory parallel programming in C/C++ and Fortran. Several compilers implement the OpenMP API. The GNU Compiler Collection (GCC; http://gcc.gnu.org/) includes a Fortran–95 compiler, so-called “gfortran” (http://gcc.gnu.org/gfortran/), licensed under the GNU General Public License (GPL; http://www.gnu.org/licenses/gpl.html);
2 Stellar Polytropic Models

Polytropic models have been widely applied to the study of nondistorted (hence spherical) stars. Such stars obey the equations of hydrostatic equilibrium,

\[ \frac{dP}{dr} = -\frac{Gm\rho}{r^2}, \quad \frac{dm}{dr} = 4\pi r^2 \rho, \tag{1} \]

where \( P(r) \) is the pressure, \( m(r) \) the mass inside a sphere of radius \( r \), and \( \rho(r) \) the density. As "equation of state" (EOS) in the polytropic models of stars is used the well-known "polytropic" EOS ([7], Chapter IV, Eq. (1))

\[ P = K \rho^{1+(1/n)}. \tag{2} \]

We can introduce the so-called "normalization equations" ([7], Chapter IV, Eqs. (8a) and (10a), respectively)

\[ \rho = \lambda \theta^n, \quad r = \alpha \xi, \tag{3} \]

where \( \lambda \) is the central density of the star, \( \lambda = \rho_c \), and \( \alpha \) a length defined by ([7], Chapter IV, Eq. (10b))

\[ \alpha = \left[ \frac{(1 + n)K\lambda^{1/(1-n)-1}}{4\pi G} \right]^{1/2}. \tag{4} \]

Accordingly, using \( \lambda \) as the "polytropic unit of density" and \( \alpha \) as the "polytropic unit of length", we verify that, in such "polytropic units", \( \theta^n \) is the measure of density and \( \xi \) the measure of length.

We then insert Eqs. (2) and (3) into Eqs. (1a, b) and, after some algebra, we obtain the so-called "Lane–Emden equation" (cf. [7], Chapter IV, Eqs. (11) and (12))

\[ \theta'' + \frac{2}{\xi} \theta' = -\theta^n, \tag{5} \]

which, when integrated along a prescribed integration interval,

\[ \xi \in [\xi_{\text{start}} = 0, \xi_{\text{end}}] = \mathbb{I}_{\xi} \subset \mathbb{R}, \tag{6} \]

with initial conditions

\[ \theta(\xi_{\text{start}}) = 1, \quad \theta'(\xi_{\text{start}}) = 0 \tag{7} \]
where primes denote derivatives with respect to $\xi$, gives as solution the so-called “Lane–Emden function” $\theta[\xi_0] \subset \mathbb{R}$.

The second-order differential equation (5) together with the initial conditions (7) establish an “initial value problem” (IVP). Our aim in this study is to compute multiprecision solutions $\theta[\xi] \subset \mathbb{R}$ of this IVP. There are, however, two problems regarding Eq. (5). First, to remove the indeterminate form $\theta'/\xi$ at the origin, appearing in the left-hand side, we modify the denominator by adding a tiny quantity, $\xi_0$, to it, i.e. $\theta'/(\xi + \xi_0)$. Since $\xi_0$ is small, the initial conditions (7) are valid at the starting point $\xi_{start} + \xi_0 = \xi_0$ as well. Accordingly, the integration interval becomes

$$\xi \in [\xi_0, \xi_{end}] = \mathbb{R} \subset \mathbb{R}. \quad (8)$$

The second problem is that for values $\xi$ greater than the first root $\xi_1$ of $\theta(\xi)$, $\xi > \xi_1$, the Lane–Emden function changes sign, $\theta(\xi) < 0$, and thus the term $\theta^n$ in Eq. (5) becomes undefined — raising a negative real number to a real power, e.g. $(-2.7)^{1.5}$, is not defined in $\mathbb{R}$. This undefined issue can be removed by taking instead the real power of the absolute value of $\theta$, $|\theta|^n$. Note however that this “numerical trick” is appropriate only when interested in finding the first root $\xi_1$ of the function $\theta$; while it becomes unreliable when searching for higher roots of $\theta$ (i.e. $\xi_2, \xi_3, \xi_4, \ldots$). To compute such higher roots — involved, e.g., in models of planetary systems — we need to apply the so-called “complex–plane strategy”; full details of this method can be found in [8].

As it is usual in numerical analysis, we proceed by transforming Eq. (5) into a system of two first-order differential equations, with the IVP under consideration having then the form

$$\theta' = \eta, \quad \eta' = -\frac{2}{(\xi + \xi_0)} \eta - |\theta|^n, \quad \xi \in \mathbb{R}, \quad \theta(0) = 1, \quad \eta(0) = 0. \quad (9)$$

By solving this IVP, we can compute several significant physical characteristics of a stellar model with finite radius (i.e. $n \in [0, 5)$, since the model with polytropic index $n = 5$ has an infinite radius). If the radius $R$, the mass $M$, and an appropriate polytropic index $n$ are known for a stellar model, then the polytropic units $\alpha$ and $\lambda$ are given by

$$\alpha = \frac{R}{\xi_1(n)}, \quad \lambda = -\frac{\xi_1(n)}{4\pi \theta'(|\xi_1(n)|)} \frac{M}{R^3}, \quad (10)$$

where the symbol $\xi_1(n)$ shows that the root $\xi_1$ varies with $n$.

The ratio of the central density, $\lambda = \rho_c$, to the mean density, $\bar{\rho}$, is given by (cf. [2], Chapter IV, Eq. (78))

$$\frac{\lambda}{\bar{\rho}} = -\frac{\xi_1(n)}{3 \theta'(|\xi_1(n)|)} \quad (11)$$

Furthermore, it can be proved that the coefficient $N_n$ appearing in the mass-radius relation ([8], Chapter IV, Eqs. (72) and (75)) is equal to

$$N_n = \frac{1}{n+1} \left[ \frac{4 \pi}{\omega_n^{n+1}} \right]^{1/n}, \quad (12)$$
where the coefficient $\omega_n$ is defined by ([7], Chapter IV, Eq. (73))

\[ \omega_n = -\xi_1(n)^{(n+1)/(n-1)} \theta'(_1(n)). \] (13)

Finally, the coefficient $W_n$ appearing in the central-pressure relation ([7], Chapter IV, Eq. (80)) is defined by ([7], Chapter IV, Eq. (81))

\[ W_n = \frac{1}{4\pi(n+1)\left[\theta'(_1(n))\right]^2}. \] (14)

### 3 Parallel Programming

Our computer comprises an Intel® Core™ i7–950 processor with four physical cores. This processor possesses the Intel® Hyper-Threading Technology, which delivers two processing threads per physical core. gfortran has been installed in this computer by the TDM-GCC “Compiler Suite for Windows” ([http://tdm-gcc.tdragon.net/](http://tdm-gcc.tdragon.net/)) free software distributed under the terms of the GPL.

According to GF-M (Sec. 6.1.16), to enable the processing of the OpenMP directive sentinel `!$OMP` (OMP-M, Sec. 2.1), gfortran is invoked by the `-fopenmp` option. Then all lines beginning with the sentinels `!$OMP` and `!$` (OMP-M, Sec. 2.2) are processed by gfortran.

We develop here a parallel programming code for solving the polytropic IVP (9). The code consists of two parts. The task of the first part is to provide all the available computer cores with the required variables. The second part performs numerical computations in parallel for all the polytropic indices given. We use the work-sharing constructs of OpenMP package (OMP-M, Sec. 2.5) in order to share the numerical work and to activate the available computer cores.

A critical step in parallel programming is the demarcation of the shared memory by using data-sharing attribute clauses (OMP-M, Sec. 2.9.3) like `SHARED`, `PRIVATE` and `FIRSTPRIVATE` (for an indicative list of variables categorized this way, see Table 1). Variables with values to be shared among all threads are declared as `SHARED`. On the other hand, variables are declared as `PRIVATE` when each thread has to work with its own copy of values of such variables. Furthermore, variables initialized according to values assigned prior to entry into parallel processing are declared as `FIRSTPRIVATE`. The computations are distributed over all the threads by the `SCHEDULE(DYNAMIC)` clause (OMP-M, Sec. 2.5.1, Table 2-1). Thus, once a particular thread finishes its allocated iteration, it returns to get a next one from the iterations that are left to be processed. The work-sharing construct `DO` (OMP-M, Sec. 2.5.1) is used in order for the computer to share among its threads the work of calculating the first root for each polytropic index. The user has to provide a value to an integer variable `NMODEL`, equal to the number of threads available in his/her computer; thus the number of the models, which are to be processed in parallel, must be less or equal to `NMODEL`. 

4
4 Multiprecision Environment and Modified Runge–Kutta–Fehlberg 5(4) Code

The computations are performed in the high precision environment of the package MPFUN90 [3, 4]. This package supports a flexible, arbitrarily high level of numeric precision — with hundreds or thousands of decimal digits. MPFUN90 is written in Fortran–90. An intermediate translating program is not required, since translation of the code in multiprecision is accomplished by merely utilizing advanced features of Fortran–90. In this study, our computations are carried out with a precision of 64 decimal digits.

The code DCRKF54, developed and used in [2], can solve complex IVPs. We have modified properly this code in order for the modified one, so-called DDRKF54, to solve real IVPs in the multiprecision environment of MPFUN90. The header of the subroutine DDRKF54 has the form

\begin{verbatim}
SUBROUTINE DDRKF54(IMDL,A,B,N,Y,DEQS,H,HMIN,HMAX,
& ATOL,RTOL,QLBD,NFLAG)
\end{verbatim}

We assign to these arguments the values IMDL = n (integer), A = r_{start} (real of proper kind), B = r_{end} (real of proper kind), N = 2 (integer), Y(1) = \theta(\xi_{start}) (real of proper kind), Y(2) = \theta'(\xi_{start}) (real of proper kind). The remaining arguments are discussed in [2] (Appendix).

We compute the first root of the Lane–Emden function \theta by combining DDRKF54 with a code that mimics the well-known bisection algorithm. The basic idea is that, when the density changes sign from plus to minus, then the code returns to the last computed point, reduces the stepsize to its half, and repeats the computation at the new point. If the density is again negative, then the process is repeated.

5 Numerical Results

Integration takes place along two successive intervals. The first interval is a very short one, \( I_1 = [10^{-26}, 30 \times 10^{-26}] \), in order for the code DDRKF54 to be accurately initiated. The stepsize along this interval is kept constant and very small, of order \( 10^{-29} \) (Table 2).

The second integration interval, \( I_2 \), extends from the end of \( I_1 \) up to a value \( \sim \frac{3}{2} \xi_1 \). In this study, we resolve 9 polytropic models with indices \( n = 0.00, 1.00, 1.50, 2.00, 2.45, 2.50, 3.00, 3.23, 3.25, \) and \( 3.50 \).

The multiprecision results of this study are given in Tables 3–5. A polytropic index appropriate for verifying the accuracy of our code is \( n = 1 \), since, as it is well-known, this case has an analytical solution and the corresponding first root of the Lane–Emden function is \( \xi_1 = \pi \) (5, Chapter IV).
Acknowledgments

The authors acknowledge the use of the Fortran-90 Multiprecision System [3, 4].

References

[1] G. Engeln-Mullges and F. Uhlig, *Numerical Algorithms with Fortran* (Springer-Verlag, Berlin, 1996).

[2] V. S. Geroyannis and F. N. Valvi, *Int. J. Mod. Phys. C* 23, 5 (2012).

[3] D. H. Bailey, *ACM Transactions on Mathematical Software*, 21, No 4, 288 (1995).

[4] D. H. Bailey, *RNR Technical Report*, RNR-94-013 (1995).

[5] *Using GNU Fortran*, [http://gcc.gnu.org/onlinedocs/gcc-4.7.0/gfortran.pdf](http://gcc.gnu.org/onlinedocs/gcc-4.7.0/gfortran.pdf) (Free Software Foundation, Boston, 2012; a copy of the GNU Free Documentation License is included in Sec. “GNU Free Documentation License”) (GF-M).

[6] *OpenMP Application Program Interface*, [http://www.openmp.org/mp-documents/OpenMP3.1.pdf](http://www.openmp.org/mp-documents/OpenMP3.1.pdf) (OpenMP Architecture Review Board, 2011) (OMP-M).

[7] S. Chandrasekhar, *Stellar Structure* (Dover, New York, 1939).

[8] V. S. Geroyannis and V. G. Karageorgopoulos, *New Astronomy* 28, 9 (2014).
Table 1: Indicative list of variables involved in the computations.

| Name        | Description                        | Category    |
|-------------|------------------------------------|-------------|
| POL_INDEX   | polytropic index                   | FIRSTPRIVATE|
| T_END       | final end point                    | FIRSTPRIVATE|
| T_IN        | local start point                  | FIRSTPRIVATE|
| T_OUT       | local end point                    | FIRSTPRIVATE|
| X           | independent variable of integration| FIRSTPRIVATE|
| Y           | dependent variable(s) of integration| FIRSTPRIVATE|
| H           | initial stepsize                   | FIRSTPRIVATE|
| TOLROOT     | error tolerance for computing root(s)| FIRSTPRIVATE|
| YPRIME      | first derivative(s) of Y           | PRIVATE     |
| FIRST_ROOT  | first root $\xi_1$                 | PRIVATE     |
| OMEGA_ON    | $\omega_n$, Eq. (13)              | PRIVATE     |
| CDEFF_NN    | $n_n$, Eq. (12)                    | PRIVATE     |
| CDEFF_WN    | $W_n$, Eq. (14)                    | PRIVATE     |
| NEQ         | number of first-order DEQs         | SHARED      |
| HMIN        | minimum allowed stepsize           | SHARED      |
| HMAX        | maximum allowed stepsize           | SHARED      |
| ATOL        | absolute error tolerance           | SHARED      |
| RTOL        | relative error tolerance           | SHARED      |
| QLBD        | error–tolerance coefficient of DDRKF54 | SHARED      |

Table 2: Initial values of variables and values of parameters.

| Name    | Value in the first interval | Value in the second interval |
|---------|----------------------------|------------------------------|
| T_IN    | 0                          | $30 \times 10^{-26}$         |
| T_END   | $30 \times 10^{-26}$       | model dependent              |
| EPS     | $10^{-42}$                 | $10^{-42}$                   |
| H       | $10^{-29}$                 | $10^{-4}$                    |
| HMIN    | $10^{-10}$                 | $10^{-5}$                    |
| HMAX    | $10^{-1}$                  | $10^{-1}$                    |
| RTOL    | $10^{-42}$                 | $10^{-32}$                   |
| ATOL    | $10^{-42}$                 | $10^{-46}$                   |
| QLBD    | $7.5 \times 10^{-1}$       | $7.5 \times 10^{-1}$         |
| TOLROOT | inactive                   | $10^{-46}$                   |
Table 3: Characteristics of the Lane–Emden function, I.

| $n$  | $\xi_1$                              | $\xi_2^2 \theta' (\xi_1)$ |
|------|--------------------------------------|---------------------------|
| 0.00 | 2.44948 9127 8178 86819 72840 74705 9 | 4.89897 9485 66356 19639 45681 49411 8 |
| 1.00 | 3.14159 2653 58973 3846 26433 83279 5 | 3.14159 2653 58973 3846 26433 83279 5 |
| 1.50 | 3.65375 3762 1912 24260 90942 80459 2 | 2.71405 5120 10865 71902 45332 77696 4 |
| 2.00 | 4.35287 4595 9612 67697 35700 61526 1 | 2.41104 6012 96983 78364 84427 44671 4 |
| 2.45 | 5.23614 1404 86923 45598 24188 51147 6 | 2.20681 7979 31278 65923 1 (−1) |
| 2.50 | 5.35527 5490 93600 09273 6 | 2.18719 95655 17078 95321 92590 89736 0 |
| 3.00 | 6.86986 86193 76900 37545 45281 87123 1 | 2.01823 5950 96622 40281 28131 70057 9 |
| 3.23 | 7.91690 48976 54771 5893 80785 74290 1 | 1.95492 90412 25479 62411 38693 00245 9 |
| 3.50 | 9.3580 53442 44850 44410 47426 25789 3 | 1.89055 70934 43116 39390 85853 05853 5 |

Table 4: Characteristics of the Lane–Emden function, II.

| $n$  | $\lambda_s / \rho_s$ | $\omega / n$ |
|------|----------------------|--------------|
| 0.00 | 1.00000 00000 00000 00000 00000 00000 00000 0 (+0) | 3.33333 33333 33333 33333 33333 33333 33333 3 (+0) |
| 1.00 | 3.28986 81366 96452 87294 48303 33292 0 (+0) | 1.04949 80935 71378 17586 20320 76059 5 (+1) |
| 1.50 | 5.99070 45163 03533 56943 21224 39795 1 (+0) | 4.13518 51814 61111 26824 04912 87756 7 (+0) |
| 2.00 | 1.14025 42616 95126 16631 26684 57624 2 (+1) | 3.82662 62035 47210 20115 81302 79323 9 (+0) |
| 2.45 | 2.16843 46128 86921 36359 82587 2 (+1) | 3.56027 44648 51473 84876 18285 43047 9 (+0) |
| 2.50 | 2.34064 62267 06270 85025 51279 20224 5 (+1) | 3.82662 62035 47210 20115 81302 79323 9 (+0) |
| 3.00 | 5.41824 81076 30151 80169 27241 2 (+1) | 2.01823 5950 96622 40281 28131 70057 9 (+0) |
| 3.23 | 8.46085 06877 60592 67790 61083 54949 5 (+1) | 1.57926 45176 10004 22561 84996 73130 6 (+0) |
| 3.50 | 1.52883 66297 09408 05111 21531 83844 2 (+2) | 1.20425 46220 02747 85274 19573 06534 3 (+0) |

Table 5: Characteristics of the Lane–Emden function, III.

| $n$  | $N_n$ | $W_n$ |
|------|-------|-------|
| 0.00 | 0.0000 00000 00000 00000 00000 00000 00000 0 (+0) | 1.19366 20731 89215 01826 66282 25283 9 (+1) |
| 1.00 | 6.28318 53071 79586 47692 52867 66559 0 (+0) | 3.92699 08169 87241 54807 83042 29099 3 (+1) |
| 1.50 | 4.24216 67959 99143 05430 8804 94342 2 (+1) | 7.0140 37135 02919 86069 32395 45144 2 (+1) |
| 2.00 | 3.64747 94209 21269 20134 44373 45944 9 (+1) | 1.63818 33457 74008 02382 13552 61677 0 (+0) |
| 2.45 | 3.51539 09189 35827 95514 94515 07072 2 (+1) | 3.56027 44648 51473 84876 18285 43047 9 (+0) |
| 2.50 | 3.51500 94082 14026 96517 75801 90482 5 (+1) | 3.90906 17224 91728 32009 55170 68744 8 (+0) |
| 3.00 | 3.63936 06192 87273 76108 23069 23888 2 (+1) | 1.10506 81023 46314 79930 43159 71504 1 (+1) |
| 3.23 | 3.77536 15181 81869 22747 82268 81199 8 (+1) | 1.93379 07742 64900 15775 48009 21363 7 (+1) |
| 3.50 | 4.01433 51975 50305 19191 13723 47988 4 (+1) | 4.09098 16957 57662 25263 91197 48667 2 (+1) |