An Underflow-Induced Graphics Failure Solved by SLI Arithmetic

Daniel W. Lozier
Applied and Computational Mathematics Division
Computing and Applied Mathematics Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

Abstract

Floating-point underflow is often regarded as either harmless or as an indication that the computational algorithm is in need of scaling. A counterexample to this view is given of a function for which contour plotting is difficult due to floating-point underflow. The function arose as an asymptotic solution to a model problem in turbulent combustion in which two chemical species (fuel and oxidizer) mix and react in a vortex field. Scaling is not a viable option because of extreme sensitivity to a small physical parameter. Standard graphics software packages produce erroneous contours without any indication of difficulty. This example provides support for considering symmetric level-index arithmetic, a new form of computer arithmetic which is immune to underflow and overflow.

1 Introduction

In 1989, Rehm *et. al.* [24] derived an approximate solution to a model problem in the theory of turbulent combustion. The problem is known as the Marble problem after its originator, F. E. Marble [18]. Analyses, applications and generalizations are given in [2, 12, 13, 14, 15, 20, 21, 24, 25].

The problem involves two reacting chemical species (fuel and oxidizer) which occupy two half-spaces initially. These are brought together at time $t = 0$ and a line vortex is imposed in the plane of the interface. With the center line of the vortex identified as the z-axis, cylindrical symmetry reduces the number of independent spatial variables to two. Using polar coordinates, the tangential velocity of the vortex is prescribed as a function $v_\theta(r, t)$. The rotation of the vortex stretches the interface by winding it around the origin. This promotes combustion by increasing the number of fuel and oxidizer molecules in contact with each other.

Among the simplifying assumptions made in [24] are that the kinematic viscosity is constant, the diffusion coefficients are constant and the same for both reactants, and the reaction rate is infinite. Two key model parameters are the Reynolds and Schmidt numbers. The former, defined as the ratio of the circulation of the vortex to the kinematic viscosity, increases with the speed of rotation. The latter is defined as the ratio of the kinematic viscosity to the species diffusion coefficient. The assumption of an infinite reaction rate allows a flame sheet approximation to be made. The location and shape of the flame sheet is of particular interest in applications. For example, reactant consumption and heat production are obtained by integrating the normal derivative of the solution along the flame sheet.

The flame sheet is a level curve, or contour, of the solution. Therefore, it can be computed (in principle, at least) by inverse interpolation from data on a grid. The data are the computed values of the solution on the grid. Graphics packages such as [3, 10, 19, 26] provide subroutine calls to compute and display contours of data on two-dimensional grids. These subroutines are quite powerful. They produce excellent graphics output with a minimum of programming effort. They hide the messy programming details needed to implement algorithms and drive display devices. But unfortunately, as we shall see in the case of the flame sheet, they are susceptible to failure due to underflow.

Underflow and overflow are characteristics of finite-wordlength floating-point arithmetic (FLP). The purpose of this paper is to show that, if underflow is avoided scrupulously, correct contours are generated for the Marble problem. It is believed that other problems exist for which underflow causes similar difficulties. It is important to emphasize that the avoidance of underflow is not done by scaling or any other modification of the algorithms. Rather, it is done by using a computer arithmetic that is immune to underflow (and overflow). Symmetric level-index arith-

metic (SLI) was introduced in 1984 by Clenshaw and Olver [5]. A good general reference is the introductory survey by Clenshaw, Olver and Turner [7]. Proof of closure, i.e. immunity to underflow and overflow, and a precision comparison against FLP is given by Lozier and Olver [16]. Other properties, algorithms for arithmetic operations, and applications are given in [4, 6, 8, 9, 17, 22, 23, 27, 28, 29].

In section 2 we start with a Fourier representation of the model solution and state some of its mathematical properties. The solution exhibits rather complicated behavior, especially in a neighborhood of the origin, but it also exhibits radial symmetry and this symmetry is very helpful in assessing the contours that are produced by graphics packages.

In section 3 we present flame sheet contours produced by the graphics package GCONTR [26] when executed in FLP and SLI. We shall see that in FLP the contours are falsified near the origin because all function values sufficiently close to the origin underflow and are replaced by zero. Although GCONTR produces a continuous contour that is plausible at first sight, it lacks the radial symmetry that it must have from its mathematical definition and from physical considerations. This same behavior is exhibited by all the graphics packages cited. None of them issued any warning or other indication that a region of non-uniqueness had been encountered. In contrast, we shall see that, up to limits imposed by the resolution of the data grid, the contours computed in SLI are correct. In particular, they exhibit radial symmetry.

In section 4 we examine the algorithmic requirements of contour graphing. We identify three processes — search, continuation and inverse interpolation — and observe that a heuristic component of each is present. The heuristic component of inverse interpolation arises from non-uniqueness in the data and we ask whether this can be prevented. Non-uniqueness can arise from either finite precision or finite range with FLP but, with SLI, it can arise only due to finite precision. Also in section 4, we see why scaling is only very weakly effective in attacking underflow in the Marble problem.

Finally, in section 5 we state our conclusions from this graphics example.

2 The Marble Solution $Z(r, \phi)$

The solution depends on the generalized exponential integrals

$$E_n(\eta) = \int_1^\infty s^{-n} e^{-\eta s} ds \quad (n = 0, 1, 2, \ldots) \tag{1}$$

where

$$\eta = r^2 \tag{2}$$

and also on the Reynolds and Schmidt numbers

$$R_e \geq 0, \; S_c > 0. \tag{3}$$

The auxiliary functions

$$a(\eta) = \frac{1 + R_e^2 f(\eta)}{4 S_c \eta} \tag{4}$$

and

$$b(\eta) = \frac{R_e g(\eta)}{2 S_c} \tag{5}$$

are defined in terms of

$$f(\eta) = \eta^{-2}[1/3 - 2E_4(\eta) + E_4(2\eta)] \tag{6}$$

and

$$g(\eta) = \eta^{-1} E_2(\eta) + \eta^{-2}[E_3(\eta) - 1/2]. \tag{7}$$

Then

$$Z(r, \phi) = \frac{1}{2} - \frac{2}{\pi} S(\eta, \phi). \tag{8}$$

where

$$S(\eta, \phi) = \sum_{m=0}^{\infty} (-1)^m T_m(\eta, \phi) \tag{9}$$

and

$$T_m(\eta, \phi) = \frac{e^{-(2m+1)^2 a(\eta)}}{2m+1} cos(2m+1)[\phi - b(\eta)]. \tag{10}$$

Equation (8) is an asymptotic approximation to the solution of the Navier-Stokes equations, valid for large Schmidt number and uniform in any annulus $0 < r_1 \leq r \leq r_2 < \infty$. Its derivation and physical interpretation is discussed in detail in [24].

The behavior of the auxiliary functions can be obtained from the properties of the generalized exponential functions [1, Chapter 5]. It can be shown that

$$a(\eta) \sim \tfrac{1+R_e^2}{4 S_c \eta} \quad (\eta \to 0), \tag{11}$$

$$a(\eta) \sim \tfrac{1}{4 S_c \eta} \quad (\eta \to \infty), \tag{12}$$

$$b(\eta) \sim \tfrac{R_e}{4 S_c} (1 + \gamma + \ln \eta) \quad (\eta \to 0, R_e > 0), \tag{13}$$

$$b(\eta) \sim -\tfrac{R_e}{4 S_c \eta} \quad (\eta \to \infty, R_e > 0) \tag{14}$$

where $\gamma = 0.57721\ldots$ is Euler's constant. Furthermore, it can be shown that $a(\eta)$ is monotonically decreasing and that $b(\eta)$ is monotonically increasing (when $R_e > 0$).

We conclude this section with the statement of several properties of $Z(r, \phi)$.

Property 1 *The series (9) is convergent everywhere. Furthermore, its convergence is increasingly rapid as* $\eta \to 0$ *and its sum vanishes at the origin.*

Property 2 *When* $R_e = 0$, *the Marble solution is* $Z(r,\phi) = \frac{1}{2}\mathrm{erfc}(\sqrt{S_c}\, r\cos\phi)$ *exactly.*

Although the applications of scientific interest are for $R_e > 0$, this property is useful in confirming the correctness of computer programs.

Property 3 *If* $R_e > 0$, *the series (9) oscillates with unboundedly growing frequency within an exponentially vanishing envelope as* $\eta \to 0$.

Property 4 *The numerical range of the Marble solution is the unit interval and*

$$\lim_{r\to\infty} Z(r,\phi) = \begin{cases} 0 & (0 \leq \mid \phi \mid < \pi/2), \\ 1/2 & (\phi = \pm\pi/2), \\ 1 & (\pi/2 < \mid \phi \mid \leq \pi). \end{cases} \tag{15}$$

Property 5 *The Marble solution exhibits radial symmetry, i.e.*

$$Z(r,\phi \pm \pi) = 1 - Z(r,\phi). \tag{16}$$

Property 6 *(i) The level curve* $Z = 1/2 + \epsilon, 0 < \epsilon < 1/2$, *is the radially symmetric image of* $Z = 1/2 - \epsilon$. *(ii) The level curve* $Z = 1/2$ *may be regarded as having two branches, joined at the origin, of which one exhibits the behavior:*

1. *As* $r \to 0$, *the branch spirals into the origin, encircling it infinitely often*
2. *As* $r \to \infty$ *the branch approaches the positive y-axis*

and the other is its radially symmetric image.

Part (i) is a restatement of Property 5. In part (ii), the level curve $Z = 1/2$ corresponds to the contour on which the series (9) vanishes. This occurs when

$$\phi = b(\eta) \pm \frac{\pi}{2}. \tag{17}$$

Taking the positive sign and using (13) and (14),

$$\phi(r) \sim \begin{cases} \frac{R_e}{2S_c}\ln(r) & (r \to 0), \\ \frac{\pi}{2} - \frac{R_e}{4S_c r^2} & (r \to \infty). \end{cases} \tag{18}$$

The two conclusions follow. Taking the opposite sign in (17) produces the opposite branch.

3 Contours of the Marble Solution

One of the main purposes of the papers [24] and [25] is to present graphical depictions of the Marble solution, particularly contour plots showing the possible positions of the flame sheets. The contour $Z = 1/2$ (equivalently, $S = 0$; cf. Eq. (8)) is particularly important because it corresponds to the stoichiometric ratio of fuel and oxidizer in which exactly the right concentration of each species is present for 100% consumption. The contours $Z = 1/2 \pm \epsilon$ (equivalently, $S = \pm\epsilon$) for small ϵ are equally important because perfect stoichiometry is an ideal state unlikely to be realized in nature. This application of computer graphics is typical of many attempted by scientists who seek to increase their understanding of the passage of a physical system into some ideal but singular state.

3.1 Contours in FLP Arithmetic

The complicated behavior described by Property 6 suggests that generating the contours of the Marble solution may be a difficult challenge for a graphics algorithm. This suspicion is justified by the graphical results shown in Fig. 1. Nine plots are shown, three each for $R_e = 10, 100, 400$ (top row, middle row and bottom row, respectively) and three each for $\epsilon = 10^{-2}, 10^{-4}, 0$ (left column, middle column and right column, respectively). The square region of xy-space is the same for all plots. The square is centered at the origin and the coordinate axes are drawn. All calculations were done with 32-bit IEEE standard arithmetic [11] on a square grid of $(151)^2$ points with $S_c = 10$.

The contours divide the squares in the left column into three regions, a left region in which $Z \leq 0.49$, a central region in which $0.49 \leq Z \leq 0.51$, and a right region in which $Z \geq 0.51$. Similarly, the squares in the middle column are divided into a left region in which $Z \leq 0.4449$, a central region in which $0.4449 \leq Z \leq 0.5001$, and a right region in which $Z \geq 0.5001$. The contours themselves in the left and middle columns are seen to exhibit the radial symmetry that is required by part (i) of Property 6. But the subdivision of the squares in the right column is very different. First, there are only two regions although it appears as if there should be a central region which is not fully delineated. Second, the radially symmetric spiral behavior required by part (ii) of Property 6 is absent.

The central region in the plots in the left and middle columns, within which the Marble solution does not change very much, expands as the Reynolds number increases (down the columns) and shrinks as ϵ de-

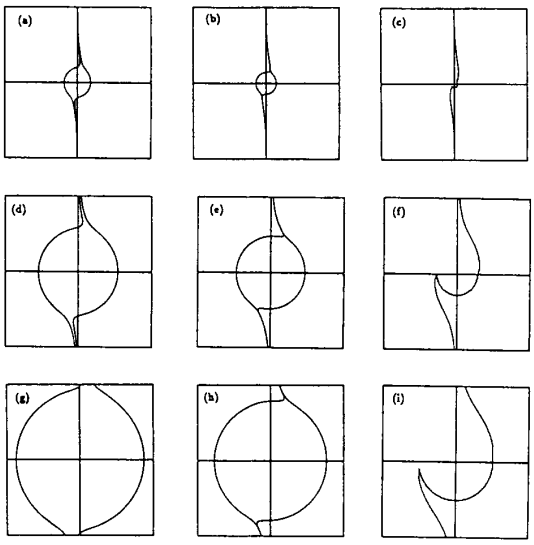

Figure 1: Contours of the function $Z(r, \phi)$ computed in 32-bit IEEE arithmetic. The approach to perfect stoichiometry for three different Reynolds numbers (row 1: $R_e = 10$; row 2: $R_e = 100$; row 3: $R_e = 400$) and three different stoichiometric ratios (col. 1: $\epsilon = 10^{-2}$; col. 2: $\epsilon = 10^{-4}$; col. 3: $\epsilon = 0$) is depicted. The asymmetric shapes in col. 3 are not consistent with the established mathematical properties or physical interpretations of the function.

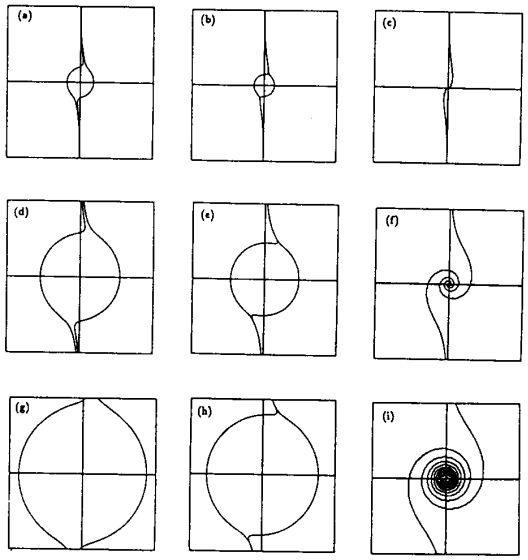

Figure 2: Contours of the function $Z(r, \phi)$ computed in 32-bit SLI arithmetic. The approach to perfect stoichiometry for three different Reynolds numbers (row 1: $R_e = 10$; row 2: $R_e = 100$; row 3: $R_e = 400$) and three different stoichiometric ratios (col. 1: $\epsilon = 10^{-2}$; col. 2: $\epsilon = 10^{-4}$; col. 3: $\epsilon = 0$) is depicted. The symmetric shapes in col. 3 are consistent with the established mathematical properties and physical interpretations of the function.

creases (across the columns). However, the central region should vanish (to the resolution of the grid, at least) when $\epsilon = 0$. Since the plots in the right column exhibit an apparent central region with a radius much larger than the mesh resolution, we conclude that the plotting software has lost critical information in the right column – due, it is reasonable to suspect, to underflow. Examination of the data on the grid showed that indeed underflow is the culprit – a central disclike area containing nothing but zeros corresponds exactly to the partly delineated central region of all three plots in the right column. In contrast, the only zero in the data for the left and middle columns is at the origin where, in fact, $Z = 1/2$.

When $\epsilon = 0$, the boundary of the underflow region is approached by the stoichiometric contour, then the plotting algorithm rather arbitrarily chose to draw a path around part of the boundary. The plots in Fig. 1 were produced using GCONTR [26] but exactly the same behavior was obtained with the graphics packages [3, 10, 19]. In fact, [19] was used in [24] in an unsuccessful attempt to produce the stoichiometric contour and so Eq. (17) was used instead. In another depiction of the stoichiometric contour in [24], the central region is blacked in by a draftsman.

None of the plotting packages cited produced any error message or diagnostic warning. Repeating the GCONTR computation in 64-bit IEEE arithmetic produced results that were not much better. Its extra precision is not needed and its additional exponent range is not adequate to avoid the falsification of the stoichiometric contour.

3.2 Contours in SLI Arithmetic

To confirm the conclusion that the falsification of the stoichiometric contour exhibited in the right column of Fig. 1 is due solely to underflow, GCONTR and the program used to generate S were modified to execute in SLI arithmetic. The results are shown in Fig. 2. There is, as expected, no difference in the left and middle columns. However, in marked contrast

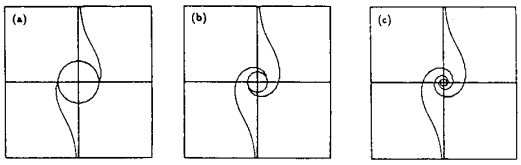

Figure 3: Contours of the function $Z(r, \phi)$ computed in 32-bit SLI arithmetic. The approach to perfect stoichiometry for Reynolds number $R_e = 100$ and three different stoichiometric ratios (col. 1: $\epsilon = 10^{-40}$; col. 2: $\epsilon = 10^{-400}$; col. 3: $\epsilon = 10^{-4000}$) is depicted; compare Fig. (3d,e,f). The development of the spiral is consistent with the established mathematical properties and physical interpretations of the function.

to Fig. 1, the right column exhibits very clearly the development of the radially symmetric stoichiometric contour as R_e increases.

The center of Fig. (2i) displays a dense pattern that is not a spiral. The reason can be seen by considering the frequency of oscillation of the cosine factor in the leading term of Eq. (9). The equation

$$b(r + h(r)) = b(r) + \pi. \tag{19}$$

is easily solved for small r by using Eq. (13) and we can write

$$\frac{r}{h(r)} = \frac{1}{exp(2\pi S_c/R_e) - 1}. \tag{20}$$

If r is the radial step size and $r/h(r) > 1$, then the leading cosine factor is poorly resolved at least up to $r/h(r)$ radial steps. We have $r/h(r) = 0.0019, 1.14, 5.9$ for $S_c/R_e = 1, 1/10, 1/40$, estimates that are in reasonable agreement with Figs. (2c,f,i).

To demonstrate the development of the spiral more fully in the case $R_e = 100$, say, it is necessary to take smaller values of ϵ. Figure 3 fills in the gap between Figs. (2e) and (2f) with $\epsilon = 10^{-40}, 10^{-400}, 10^{-4000}$. The spiral is just beginning to be apparent in Fig. (3a) and Fig. (3b) with ϵ slightly below the underflow limit of 32-bit and 64-bit IEEE arithmetic, respectively. It is quite apparent in Fig. (3c) with ϵ near the underflow limit of the widest available exponent range in a commercial FLP arithmetic. The spiral shown in Fig. (2f) results from numbers as small as 10^{-42800}, well beyond the range of any commercial FLP arithmetic.

4 Contouring Algorithms

According to Snyder [26], there are two approaches to calculating contours. In the cellular approach, each grid cell is visited once and all contours that intersect it are drawn before proceeding to the next cell. This approach is valuable when the grid is so large that the data exceeds available memory. In the contour following approach, each contour is drawn completely before the next contour is processed. Snyder states that contour following is more economical of plotter pen movements and it is the approach he takes in his GCONTR subroutine. It is also the approach taken by the graphics packages [3, 10, 19] and it is the only approach considered here.

Three algorithms are used in contour following. These are search, continuation and inverse interpolation.

search The data matrix is searched to find a cell edge that contains a point on the contour line. This search, like all searches, is heuristic. That is, there is no way to minimize the search time according to an objective criterion that is universally applicable. Snyder uses a spiral search strategy starting in whatever cell happens to contain the pen when the search begins.

continuation In the simplest case, the contour intersects the cell boundary in exactly two points and a straight line is used to join them. All other cases are more complicated and some require heuristic choice. For example, if the contour intersects an entire edge plus a single point of the opposite edge, a unique mathematical determination of where to continue the contour from the point is impossible. Snyder states and justifies an algorithm that avoids contour crossings and produces unique contours when the contour and cell boundary intersect at exactly three or four points.

inverse interpolation This is used in search and continuation to identify points of intersection. Heuristics are introduced when the algorithm fails and so, in the interest of keeping heuristics to a minimum, we examine it in some detail.

4.1 Inverse Interpolation

Let $X = f(x)$ be a real function of a real variable, and let

$$A = f(a), B = f(b) \tag{21}$$

where $a \neq b$. If C is any point lying between A and B, we approximate $f^{-1}(C)$ by

$$c = \frac{a(B-C)+b(C-A)}{B-A}, \tag{22}$$

provided $B \neq A$.

Now suppose $\tilde{A}$ and $\tilde{B}$ are hardware (or software) FLP approximations of A and B. Let ρ be the relative error bound of the FLP system, and σ the underflow threshold. Then it can happen that $\tilde{B} = \tilde{A}$ when $B \neq A$ if

$$| B - A | < \rho \max(| A |, | B |) \tag{23}$$

or

$$\max(| A |, | B |) < \sigma. \tag{24}$$

If (23) or (24) is true, we regard it as a failure of the floating-point system; either its precision or its range is deficient.

By the mean value theorem, there exists m between a and b such that

$$B - A = (b-a)f'(m). \tag{25}$$

Thus precision failure occurs when

$$| b - a | < M_p(a,b) \tag{26}$$

where

$$M_p(a,b) = \frac{\rho \max(| A |, | B |)}{| f'(m) |}. \tag{27}$$

Now $| b - a |$ corresponds to the mesh spacing in a contouring algorithm. The inequality (26) tells us that precision failure will occur when the mesh spacing is less than $M_p(a,b)$.

But the mesh spacing has to be small enough to yield acceptable graphics quality. Suppose an acceptable mesh spacing is 10^{-2}, and suppose $\rho = 10^{-7}$ and $\sigma = 10^{-39}$. These values of the rounding unit and underflow threshold are consistent with 32-bit IEEE arithmetic. From (27) with $\max(| A |, | B |) = 1$, failure occurs in grid cells where the derivative is smaller than 10^{-5}. Extending precision so that $\rho = 10^{-16}$, which is consistent with 64-bit IEEE arithmetic, allows acceptable graphics quality with derivatives as small as 10^{-14}.

Alternatively, it may be possible to translate the data by computing $f(x) - K$ where K is chosen so that $\max(| A - K |, | B - K |) \ll \max(| A |, | B |)$. Graphically this corresponds to moving the original graph of $f(x)$ up or down so that its very flat portion (where the derivative is very small) is near the x-axis. If there exists a zero, k say, of $f'(x)$ near a and b, then $f(k)$ would be a good choice for K. Suppose that $\max(| A - K |, | B - K |) = 10^{-35}$. Then derivatives as small as 10^{-40} could be accommodated without failure.

4.2 The Marble Solution

For $R_e = 100$, the minimum of the data used in producing Figs. (1d,e,f), (2d,e,f) and (3) was 10^{-42800}, approximately. The mesh spacing was 0.04 and an approximation to the derivative at this radius is 4×10^{-42802}. Therefore

$$M_p(0,\rho) \approx \frac{1}{8} \times 10^{42795}.$$

This suggests that obtaining the correct spiraling behavior down to the mesh resolution by scaling would require a tremendous effort. Let us consider the relatively simple task of scaling 32-bit IEEE just enough to obtain the contour that would be produced by 64-bit IEEE arithmetic. Since the 64-bit underflow threshold is approximately the eighth power of the 32-bit threshold, approximately eight scalings will be needed. Each scaling will be effective in a thin annulus about the origin. The radius of the 64-bit underflow region turns out to be approximately eight mesh spacings less than the corresponding 32-bit radius. If all the annuli had the same thickness — which of course they do not — then the thickness would be only one mesh spacing. In other words, scaling attacks the problem only very weakly. Added to this weakness would be the difficulties presented by the contouring software itself, which is oriented toward work in rectangular regions. Accordingly, a complicated mosaic of subregions would need to be combined to form the complete contour plot.

5 Conclusions

The Marble example leads to several general conclusions. Although stated in terms of underflow, the conclusions are equally valid for overflow.

First, there exist real-world scientific computing applications in which underflow is a serious enough problem that scaling by itself does not provide a complete remedy. The Marble example generates meaningful numbers far below the underflow threshold of all commercially available FLP hardware. The graphical depiction of computed solutions by contour plots is a popular technique in scientific computing and computer-aided design. Therefore, one might expect

to be able to find other examples of failures due to underflow in computer graphics.

Second, the need to deal with underflow leads to the use of heuristics that would be better avoided in mathematical software. In contouring software, the heuristic of circumnavigating part of the boundary of an underflow region without informing the user of a possible difficulty could have been replaced by the equally valid heuristic of blacking in the entire underflow region. The difficulty is that no single heuristic is appropriate to all cases.

Third, the twin phenomena of underflow and finite precision complicate the process of attributing suspected incorrect results to the computer arithmetic or to algorithmic limitations. If instead of applying the contouring software to the series (9) we had applied it directly to the Marble solution (8), the resulting stoichiometric contours would have been almost indistinguishable from the curves shown in Figs. (1c,f,i), no matter whether FLP or SLI arithmetic had been used. The cause would have been finite precision in both cases. The additional information that a dense non-spiral pattern is possible in the center of the stoichiometric contours, and that this pattern is caused by the algorithmic limitation of choosing a regular rectangular grid of data, is rather easily obtained from the SLI results.

The author hopes that the Marble example presented in this paper will help convince more people that computer arithmetic for scientific computing should move toward being immune, or at least highly resistant, to underflow and overflow.

6 Acknowledgement

The contribution of Ronald G. Rehm in the development of the Marble problem, the derivation of the asymptotic solution, and the interpretation of its physical meaning is gratefully acknowledged.

References

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series*. U. S. Government Printing Office, tenth printing or later edition, June 1964.

[2] H. R. Baum, D. M. Corley, and R. G. Rehm. Time-dependent simulation of small-scale turbulent mixing and reaction. In *Twenty-First International Symposium on Combustion*, pages 1263–1270. The Combustion Institute, 1986.

[3] F. Clare, D. Kennison, and B. Lackman. Ncar graphics user's guide version 2.00. Technical Report NCAR/TN-2831, National Center for Atmospheric Research, Boulder, CO, August 1987.

[4] C. W. Clenshaw, D. W. Lozier, F. W. J. Olver, and P. R. Turner. Generalized exponential and logarithmic functions. *Computers and Mathematics with Applications*, 12B(5/6):1091–1101, 1986.

[5] C. W. Clenshaw and F. W. J. Olver. Beyond floating point. *Journal of the Association for Computing Machinery*, 31(2):319–328, April 1984.

[6] C. W. Clenshaw and F. W. J. Olver. Level-index arithmetic operations. *SIAM Journal on Numerical Analysis*, 24(2):470–485, April 1987.

[7] C. W. Clenshaw, F. W. J. Olver, and P. R. Turner. *Level-Index Arithmetic: An Introductory Survey*, volume 1397 of *Lecture Notes in Mathematics*, pages 95–168. Springer-Verlag, 1989.

[8] C. W. Clenshaw and P. R. Turner. The symmetric level-index system. *IMA Journal of Numerical Analysis*, 8:517–526, 1988.

[9] C. W. Clenshaw and P. R. Turner. Root-squaring using level-index arithmetic. *Computing*, 43:171–185, 1989.

[10] Computer Associates International, Inc., 711 Stewart Avenue, Garden City, NY 11530. *CA-DISSPLA User Manual*, release 11.0 edition, 1989.

[11] IEEE, Inc., 345 East 47th Street, New York, NY 10017. *ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic*, 1985.

[12] A. R. Karagozian. *An Analytical Study of Diffusion Flames in Vortex Structures*. PhD thesis, California Institute of Technology, Pasadena, California, 1982.

[13] A. R. Karagozian and F. E. Marble. Study of a diffusion flame in a stretched vortex. *Combustion Science and Technology*, 1986.

[14] A. M. Laverdant and S. M. Candel. Etude de l'interaction de flammes de diffusion et de premlange avec un tourbillon. *La Recherche Aerospatiale*, 1988.

[15] A. M. Laverdant and S. M. Candel. A numerical analysis of a diffusion flame-vortex interaction. *Combustion Science and Technology*, 1988.

[16] D. W. Lozier and F. W. J. Olver. Closure and precision in level-index arithmetic. *SIAM Journal on Numerical Analysis*, 27(5):1295–1304, October 1990.

[17] D. W. Lozier and P. R. Turner. Robust parallel computation in floating-point and sli arithmetic. *Computing*, 48:239–257, 1992.

[18] F. E. Marble. Growth of a diffusion flame in the field of a vortex. In C. Casci, editor, *Recent Advances in Aerospace Sciences*, 1985.

[19] Megatek Corporation, 9645 Scranton Road, San Diego, CA 92121. *TEMPLATE V5.5 2D/3D Reference Manual*, February 1986.

[20] M. G. Mungal and P. E. Dimotakis. Mixing and combustion with low heat release in a turbulent shear layer. *Journal of Fluid Mechanics*, 1984.

[21] O. P. Norton. *The Effects of a Vortex Field on Flames with Finite Reaction Rates.* PhD thesis, California Institute of Technology, Pasadena, California, 1983.

[22] F. W. J. Olver. Rounding errors in algebraic processes — in level-index arithmetic. In M. G. Cox and S. Hammarling, editors, *Reliable Numerical Computation*, pages 197–205. Oxford University Press, 1990.

[23] F. W. J. Olver and P. R. Turner. Implementation of level-index arithmetic using partial table lookup. In M. J. Irwin and R. Stefanelli, editors, *Computer Arithmetic Symposium 8*, pages 144–147, Washington, DC, 1987. IEEE Computer Society.

[24] R. G. Rehm, H. R. Baum, D. W. Lozier, and J. Aronson. Diffusion-controlled reaction in a vortex field. *Combustion Science and Technology*, 66:293–317, 1989.

[25] R. G. Rehm, H. R. Baum, H. C. Tang, and D. W. Lozier. Finite-rate diffusion-controlled reaction in a vortex. *Combustion Science and Technology.* To appear.

[26] W. V. Snyder. Algorithm 531: Contour plotting. *ACM Transactions on Mathematical Software*, 4(3):290–294, September 1978.

[27] P. R. Turner. Towards a fast implementation of level-index arithmetic. *Bulletin of the IMA*, 22:188–191, 1986.

[28] P. R. Turner. A software implementation of sli arithmetic. In M. D. Ercegovac and E. E. Swartzlander, Jr., editors, *Computer Arithmetic Symposium 9*, pages 18–24, Washington, DC, September 1989. IEEE Computer Society.

[29] P. R. Turner. Implementation and analysis of extended sli operations. In P. Kornerup and D. W. Matula, editors, *Computer Arithmetic Symposium 10*, pages 118–126, Washington, DC, 1991. IEEE Computer Society.