34th AIAA Fluid Dynamics Conference
28 June–01 July 2004, Portland, OR

AIAA #2004–2330

Hybrid Symbolic-Numeric Methodology for Fluid Dynamic Simulations

José A. Camberos*
U.S. Air Force Research Laboratory
Wright–Patterson Air Force Base, Dayton, OH 45433

Larry A. Lambe†
Multidisciplinary Software Systems Research Corporation, Bloomingdale, IL 60108
and
Department of Informatics, Division of Mathematics
University of Wales, Dean St., Bangor, Gwynedd LL57 1UT, United Kingdom

Richard Luczak ‡
Science Applications International Corporation
Beavercreek, OH 45431

With all of the experience and knowledge gained from symbolic and numeric computing in the last 20 years along with advances in software engineering and the advances in terms of the availability of powerful computers, it seems that the time is at hand to revisit the original problem of solving partial differential equations using a combination of numerical calculation and symbolic manipulation. We believe that hardware technology today has become powerful enough to realize the original dream of implementing mathematics at a high level on machines in robust and efficient ways. We used computer aided symbolic computation in our previous paper[1] to explore the problem of solving partial differential equations in a more general way. Here we revisit some classic techniques in light of new technology and extend the new methodology emerging here. Continued development of the hybrid methodology is presented with the hope of inspiring new thought in the creation of hybrid symbolic–numeric algorithms for solving the equations of mathematical physics from a fundamental perspective.

Introduction

We presented a brief overview of the use of computers for symbolic computation in Camberos, Lambe, and Luczak.[1] In that work, we introduced a methodology for developing the capability for symbolic manipulation of expressions using an approach based on *ExprLib*[2,3,4,5,6] that was designed to address some stringent needs of application developers and problem solvers in the area of scientific computing. Specifically, we developed several methods as inspired by the Finite-Difference Method (FDM), such as the Heun, MacCormack, and Lax-Wendroff schemes. We also adapted the Runge-Kutta method in this context and studied a method given by Derickson and Pielke[7] then applied these methods to analyze several model equations including pure advection, heat, and Burgers' equation with encouraging results. The computations were carried out using *ExprLib* and standard ANSI C programming. In this note, we continue and extend the study of such algorithms beginning with a comparison of the simple hybrid method called the degree-k one-step Hybrid Symbolic-Numeric (HSN) in our first paper[1] and series methods. In developing this work, we plan to also have results for solving systems of equations, including the Euler equations of gas dynamics, in a follow-on technical paper. A preliminary development of this is presented here. Our goal is to develop the fundamental details that will facilitate the construction of new methods and the enhancement of existing ones using the power of computers for mathematics.

*Aerospace Engineer and AIAA Senior Member
†Chief Scientist, MSSRC (llambe@mssrc.com)
‡Principal Systems Engineer, SAIC (luczakr@saic.com)

Power Series Methods Redux

Series methods have been around for some time. See for example the classic text by Knopp.[8] The interested reader with a background in fluid dynamics may also be familiar with Blasius' original solution to Prandtl's boundary layer equations[9] which provides a good example of (pre-computer aided) use of symbolic computation involving series. Additionally, since the 1950's M. D. van Dyke noted the feasibility of using electronic digital computers to deal with the arithmetic labor of extending regular perturbation series and continued his efforts through the dawn of the personal computer in the 1980's.[10] We hope to show what could be done in this area in light of new and powerful computer based tools for symbolic computation. This Section reports some preliminary results using the *ExprLib* family of ANSI C libraries.

There are two simple formulas that are often used as the starting point for finite difference methods and are based on a Taylor series expansion of the form

$$f(x+a) = f(x) + f'(x)a + f''(x)\frac{a^2}{2!} + f'''(x)\frac{a^3}{3!} + \dots \tag{1}$$

These are the sum yielding the even terms

$$f(x,t+\Delta t) + f(x,t-\Delta t) = 2\left(f(x,t) + \frac{\partial^2 f}{\partial t^2}\frac{(\Delta t)^2}{2!} + \frac{\partial^4 f}{\partial t^4}\frac{(\Delta t)^4}{4!} + \dots\right) \tag{2}$$

and the difference which gives the odd terms

$$f(x,t+\Delta t) - f(x,t-\Delta t) = 2\left(\frac{\partial f}{\partial t}\Delta t + \frac{\partial^3 f}{\partial t^3}\frac{(\Delta t)^3}{3!} + \frac{\partial^5 f}{\partial t^5}\frac{(\Delta t)^5}{5!} + \dots\right). \tag{3}$$

These formulas will be used later.

Although it is seldom (if ever) presented as such, it turns out that many of the model equations for CFD have exact solutions that can be derived using series methods. The following development is preliminary to our (developing) understanding of how best to utilize symbolic computation in analyzing and constructing new algorithms for solving partial differential equations of interest. It also serves a valuable pedagogical purpose on its own since it presents classical solutions in a different light. We begin with the ansatz

$$u(x,t) = \sum_{n=0}^{\infty} u_n(x)t^n \tag{4}$$

where the coefficients are kept symbolic. It is slightly more convenient to work in the basis $\{\gamma_0(t), \gamma_1(t), \dots, \gamma_n(t), \dots\}$ where

$$\gamma_n(t) = \frac{t^n}{n!} \tag{5}$$

rather than the usual basis $\{1, t, \dots, t^n, \dots\}$. Note that $\gamma_0(t) = 1$ and $\gamma_1(t) = t$. Notice also that we have

$$\frac{d}{dt}(\gamma_n(t)) = \gamma_{n-1}(t). \tag{6}$$

The alternate basis will simplify some calculations a bit; thus, we assume that

$$u(x,t) = \sum_{n=0}^{\infty} u_n(x)\gamma_n(t). \tag{7}$$

Using (7) gives identical end results as those obtained with (4).

Convection/Diffusion Equation

Consider the equation (a simplified representation of steady-state advection/diffusion),

$$u' = \frac{1}{p}u'' \tag{8}$$

for a real valued function u of a real variable x and nonzero real parameter p. Assume also that u is known at boundary points:

$$u(0) = a, \ u(1) = b.$$

Assuming that u is smooth, and writing $u^{(2)} = pu'$, it is easy to see that

$$u^{(n)} = p^{n-1}u^{(1)} \tag{9}$$

for all $n \geq 2$. Assuming that u is analytic, we can apply the ansatz (7) to get

$$u(x) = \sum_{n=0}^{\infty} u^{(n)}(0)\gamma_n(x) = a + a_1\gamma_1(x) + \ldots + a_1 p^{n-1}\gamma_n(x) + \ldots \tag{10}$$

where $a_1 = u'(0)$. Now we can perform a bit of algebra on this to obtain

$$\begin{aligned}
u &= a + a_1(x + \frac{px^2}{2!} + \ldots + \frac{p^{n-1}x^n}{n!} + \ldots) \\
&= a + a_1\left(x + \frac{1}{p}\left(\frac{(px)^2}{2!} + \ldots + \frac{(px)^n}{n!} + \ldots\right)\right) \\
&= a + a_1\left(x + \frac{1}{p}\left(\left(\sum_{n=0}^{\infty}\frac{(px)^n}{n!}\right) - \frac{1}{p} - x\right)\right) \\
&= a + a_1\left(\frac{1}{p}e^{px} - \frac{1}{p}\right) \\
&= a + \frac{a_1}{p}(e^{px} - 1).
\end{aligned} \tag{11}$$

Now using $u(1) = b$, we get

$$b = a + \frac{a_1}{p}(e^p - 1)$$

so that the complete solution is

$$u(x) = a + \frac{b-a}{e^p - 1}\left(e^{px} - 1\right). \tag{12}$$

It is interesting to compare this boundary value problem with an initial value problem: $u'' = pu'$, $u(0) = a$, $u'(0) = b$. With the same assumptions as above, one immediately has

$$u(x) = a + bx + \frac{px^2}{2!} + \ldots + \frac{p^{n-1}x^n}{n!} + \ldots \tag{13}$$

and manipulations similar to those above lead to

$$u(x) = a + \frac{b}{p}\left(e^{px} - 1\right). \tag{14}$$

We expect that symbolic computation will be useful in allowing such expression of initial or boundary value problems to be encoded in this way for general solutions.

Advection Equation

A very basic equation typically used to introduce the student to the world of computational fluid dynamics models pure advection. We revisit this simple PDE to obtain solutions in terms of series. Consider

$$u_t + au_x = 0 \tag{15}$$

where we assume $a \neq 0$ (non–zero constant wave speed) with initial condition

$$u(x, 0) = u_0(x). \tag{16}$$

Using Equations (7), assuming that u_0 is smooth, and substituting into Equation (15) (and shifting an index), we have

$$u_t = \sum_{n=1}^{\infty} u_n(x)\gamma_{n-1}(t) = \sum_{n=0}^{\infty} u_{n+1}(x)\gamma_n(t) = -a\sum_{n=0}^{\infty} u_n^{(1)}(x)\gamma_n(t) \tag{17}$$

from which it immediately follows that we must have

$$u_{n+1}(x) = -au_n^{(1)}(x) \tag{18}$$

beginning with $u_1(x) = -au_0^{(1)}(x)$. It is easy to see that this recursive formula amounts to

$$u_n(x) = (-1)^n a^n u_0^{(n)}(x) \tag{19}$$

and since $a^n \gamma_n(t) = \frac{(at)^n}{n!}$, we have

$$u(x,t) = u_0(x) - u_0^{(1)}(x)at + u_0^{(2)}(x)\frac{(at)^2}{2!} - \dots \tag{20}$$

Note that (20) is just the Taylor series expansion of $u_0(x-at)$, so we have derived the classic solution to(15). For comparison, the standard (textbook) approach develops the solution using the method of characteristics. As is usually noted in most textbooks, it can be easily verified directly that $u(x,t) = u_0(x-at)$ is a solution to (15) for any C^1 function u_0. It is interesting that this result was derived using an assumption of smoothness.

Wave Equation in 1D

Consider now the second-order wave equation in one space dimension

$$u_{t,t} = c^2 u_{x,x} \tag{21}$$

with initial conditions

$$u(x,0) = f(x) \quad \text{and} \quad u_t(x,0) = g(x). \tag{22}$$

After a few index shifts we have

$$u_t = \sum_{n=0}^{\infty} u_{n+1}(x)\gamma_n(t) \tag{23}$$

$$u_{t,t} = \sum_{n=0}^{\infty} u_{n+2}(x)\gamma_n(t) \tag{24}$$

$$u_{x,x} = \sum_{n=0}^{\infty} u_n^{(2)}(x)\gamma_n(t). \tag{25}$$

With this result one may see that the two given initial functions are precisely what are needed to solve the resulting recurrence relations

$$u_{n+2}(x) = c^2 u_n^{(2)}(x) \tag{26}$$

and we have

$$u_{2n}(x) = c^{2n} f^{(2n)}(x) \tag{27}$$

$$u_{2n+1}(x) = c^{2n} g^{(2n)}(x). \tag{28}$$

Writing out the series, we separate the general solution such that $u(x,t) = e(x,t) + h(x,t)$ where

$$e(x,t) = f(x) + \frac{f^{(2)}(x)}{2!} + \frac{f^{(4)}(x)}{4!} + \dots \tag{29}$$

and

$$h(x) = \frac{1}{c}\left(g(x)ct + \frac{g^{(2)}(x)}{2!}(ct)^3 + \frac{g^{(4)}(x)}{4!}(ct)^5 + \dots\right). \tag{30}$$

Note that from the Taylor series even terms, (2), it follows immediately that

$$e(x,t) = \tfrac{1}{2}\left(f(x+ct) + f(x-ct)\right) \tag{31}$$

and noting the degree shift in the expansion of $h(x,t)$ and using the difference instead of the sum, i.e. the Taylor series odd terms (3), we note that, letting

$$G(x) = \int g(y)dy, \tag{32}$$

then

$$h(x,t) = G(x+ct) - G(x-ct) \tag{33}$$

and we have derived the D'Alembert formula

$$u(x,t) = \tfrac{1}{2}\left(f(x+ct) + f(x-ct)\right) + \frac{1}{2c}\left(\int_{x-ct}^{x+ct} g(y)dy\right). \tag{34}$$

As is well known, the above formula of D'Alembert is valid for any C^2 function, so again it is interesting that this result was derived assuming smoothness. For comparison, D'Alembert's formula can also be obtained by casting the wave equation into canonical form and solving.

Laplace's Equation

Consider the wave equation (21) once more. If we assume that the solution is analytic, it is reasonable to allow $c = ki$ where k is real and $i = \sqrt{-1}$. Thus consider the equation

$$u_{t,t} = -k^2 u_{x,x} \tag{35}$$

with initial conditions

$$u(x,0) = f(x) \quad \text{and} \quad u_t(x,0) = g(x). \tag{36}$$

Using this value of c gives the solution

$$u(x,t) = \tfrac{1}{2}\left(f(x+kti) + f(x-kti)\right) + \tfrac{1}{ki}\left(G(x+kti) - G(x-kti)\right) \tag{37}$$

where $G(x) = \int^x g(z)dz$ as before. The series approach above should yield this formula directly in a straightforward manner. A simple change in notation demonstrates that the unique solution to

$$u_{y,y} + k^2 u_{x,x} = 0, \tag{38}$$

which is the Laplace equation, with initial conditions

$$u(x,0) = f(x) \quad \text{and} \quad u_y(x,0) = g(x). \tag{39}$$

can be developed as follows.

Proposition 1 *Suppose that $u(x,y)$ is an analytic solution of Equation (38) with initial values*

$$u(x,0) = 0, \qquad \frac{\partial u(x,y)}{\partial y}\mid_{y=0} = g(x).$$

Let

$$G(x) = \int^x g(z)dz$$

then

$$u(x,y) = \tfrac{1}{2}\left(f(x+kyi) + f(x-kyi)\right) + \frac{1}{ki}\left(G(x+kyi) - G(x-kyi)\right). \tag{40}$$

We understand that such an initial value problem for the Laplace equation is ill posed;[11] however, such problems do arise in certain physical and engineering problems (e.g., subsonic flow behind a bow shock wave, the design of Pierce-type electron guns) and a numerical method for solving them was given by Sugai in 1959.[12] To test new schemes for solving the Laplace equation, the following formulas, which are not too difficult to obtain, can be used to generate examples.

Proposition 2 *Let x and y be real variables and n be a non negative integer. Then*

$$\tfrac{1}{2i}\left\{(x+iy)^n - (x-yi)^n\right\} = \sum_{r=0}^{4r+1\le n}\binom{n}{4r+1} y^{4r+1}x^{n-4r-1} - \sum_{r=0}^{4r+3\le n}\binom{n}{4r+3} y^{4r+3}x^{n-4r-3} \tag{41}$$

and

$$\tfrac{1}{2}\left\{(x+iy)^n + (x-yi)^n\right\} = \sum_{r=0}^{4r\le n}\binom{n}{4r} y^{4r}x^{n-4r} - \sum_{r=0}^{4r+2\le n}\binom{n}{4r+2} y^{4r+2}x^{n-4r-2}. \tag{42}$$

Nonlinear Dispersive (KdV) Equation

The formula given by

$$u(x,t) = \operatorname{sech}^2(x-4t) \tag{43}$$

is well known to represent a soliton solution to the Korteweg-deVries (KdV) equation

$$u_t + 12u_x + u_{x,x,x} = 0 \tag{44}$$

with initial function

$$u_0(x) = u(x,0) = \frac{1}{\cosh^2(x)}. \tag{45}$$

Again we seek a series ansatz, but first note a well known identity, viz.,

$$\frac{1}{\cosh^2(x)} = \frac{4}{(\exp(x)+\exp(-x))^2} = \frac{4\exp(2x)}{(1+\exp(2x))^2}$$

along with the series representation

$$\frac{1}{(1+z)^2} = \sum_{n=0}^{\infty}(-1)^n(n+1)z^n.$$

So it seems reasonable to look for solutions of the form

$$u(x,t) = \sum_{n=1}^{\infty} a_n \exp(bx+ct)^n.$$

Let us aim at the more general equation

$$u_t + au_x + du_{x,x,x} = 0.$$

Letting $\mathcal{E} = \exp(bx+ct)$, we note that

$$\frac{\partial \mathcal{E}^n}{\partial t} = nc\mathcal{E}^n, \qquad \frac{\partial \mathcal{E}^n}{\partial x} = nb\mathcal{E}^n, \qquad \frac{\partial^3 \mathcal{E}^n}{\partial x^3} = (nb)^3\mathcal{E}^n$$

Now we program each of the corresponding series for u_t, u_x, and $u_{x,x,x}$ symbolically and compute the coefficients of the series $u_t + au_x + du_{x,x,x}$ where a and d are symbolic constants. It is however convenient to see that the n^{th} coefficient of uu_x is just

$$\sum_{r=1}^{n-1} b(n-r)a_r a_{n-r}$$

for $n \geq 2$ (the series starts with the $\mathcal{E}^2$ term). Thus the recurrence relation is

$$(nc + (nb)^3 d)a_n = -ab\sum_{r=1}^{n-1}(n-r)a_r a_{n-r}, \qquad \text{for } n \geq 2. \tag{46}$$

The degree-one term only involves u_t and $u_{x,x,x}$ and amounts to

$$a_1 c + a_1 b^3 = 0$$

so assuming that $a_1 \neq 0$, we have $c = -b^3$ and the recurrence relation becomes

$$a_n = \frac{-a}{b^2(n^3-n)d}\sum_{r=1}^{n-1}(n-r)a_r a_{n-r}$$

for $n \geq 2$. Using symbolic computation to generate the first ten or so terms one quickly sees that a_n is of the form

$$\frac{(-1)^n a^{n-1} a_1^n}{c_n b^{2(n-1)d^{n-1}}}.$$

but the sequence c_n remained a puzzle. Finally after observing some interesting patterns in the prime factorizations of the c_n, we listed out nc_n instead. It then becomes clear now that

$$nc_n = 2^{2(n-1)}3^{n-1} \tag{47}$$

so that the n^{th} term had been revealed as

$$a_n = \frac{(-1)^{n-1}na^{n-1}a_1^n}{2^{2(n-1)}3^{n-1}b^{2(n-1)}d^{n-1}}. \tag{48}$$

Now a bit of rearranging gives

$$a_n = \frac{(-1)^{n-1}na^{n-1}a_1^n}{4^{n-1}3^{n-1}b^{2(n-1)}d^{n-1}} = \frac{(-1)^{n-1}na^{n-1}a_1a_1^{n-1}}{12^{n-1}b^{2(n-1)}d^{n-1}} = (-1)^{n-1}na_1\left(\frac{aa_1}{12b^2d}\right)^{n-1}. \tag{49}$$

Thus the series is

$$u = a_1\left(\mathcal{E} - 2\xi\mathcal{E}^2 + 3\xi^2\mathcal{E}^3 - \dots\right) = a_1\mathcal{E}\left(1 - 2(\xi\mathcal{E}) + 3(\xi\mathcal{E})^2 - \dots\right) = \frac{a_1\mathcal{E}}{(1+(\xi\mathcal{E}))^2} \tag{50}$$

where $\mathcal{E} = \exp(bx - b^3t)$ and $\xi = aa_1/12db^2$.

Remark: Note that using the identities above, we can write

$$u(x,t) = \frac{12db^2}{a}\frac{\exp(2\beta)}{(1+\exp(2\beta))^2} = \frac{3db^2}{a}\text{sech}^2(\beta) \tag{51}$$

where

$$\beta = \tfrac{1}{2}\left(bx - b^3t + \ln\left(\frac{aa_1}{12db^2}\right)\right) \tag{52}$$

and if we choose $a_1 = \frac{12db^2}{a}$, the log term vanishes. If we then take $b = 2$, $a = 12$, and $d = 1$, we arrive at the example at the beginning of this Section.

Taylor Series and One-Step HSN Methods

Recall the (degree-k) one-step HSN from the authors' first paper[1] for approximating the solution $u^n = u(x, nt)$ to the advection equation

$$u_t + au_x = 0$$

is given as

$$u^{n+1} = u^n - \frac{\partial u^n}{\partial x}at + \frac{1}{2}\frac{\partial u^n}{\partial x^2}a^2t^2 + \dots + \frac{(-1)^k}{k!}\frac{\partial^k u^n}{\partial x^k}a^kt^k \tag{53}$$

starting with

$$u^0 = u(x,0).$$

The function $u^m(x)$ is an approximation to $u(x, m\Delta t)$ for a given time step Δt. Thus, the degree k one-step HSN for $n = 1$ agrees with a degree k Taylor series approximation. This is the first step in a straightforward induction that gives the following.

Proposition 3 *The first k terms in the iterative formula for u^{n+1} in the degree k one-step HSN agrees with the first k terms of the Taylor series expansion of the solution $u(x,(n+1)\Delta t)$ centered at $t = 0$.*

In light of this Proposition, it is natural to wonder about the significance of the higher order terms in the one-step HSN method. We can illustrate this through an example after which we will give an analysis.

Consider the case such that

$$u_0(x) = \frac{1}{1+x^2}. \tag{54}$$

The exact solution to the advection equation with this initical condition is

$$\frac{1}{1+(x-at)^2}.$$

Take $a = -1$, $x = 1$ and consider three iterations of the degree-three one-step method $h(t) = u^3(1, 3t)$, the Taylor series $f(t)$ up to degree three (evaluated at $3t$), and the exact solution $g(t)$ (also evaluated at $3t$). We find that

$$h(t) = \frac{\alpha+\beta}{8}, \qquad f(t) = \frac{9t^2-6t+2}{4}, \qquad g(t) = \frac{1}{9t^2+6t+2}, \tag{55}$$

where $\alpha = -420t^9 + 420t^8 - 255t^6$ and $\beta = 210t^5 - 78t^4 + 18t^2 - 12t + 4$. We see in Figure 1 on the following page that the HSN method follows the exact solution over a larger range than the Taylor series.

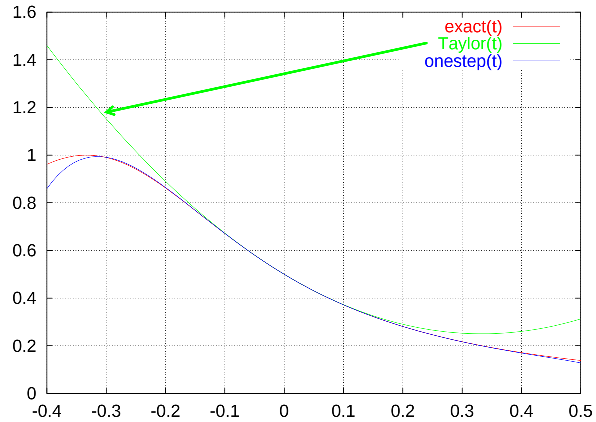

Figure 1 Comparison of 1-step HSN, Taylor, and exact solutions for the 1D advection equation.

To understand why this should be, note that the one-step HSN, while based on a Taylor series approximation, moves its center with each iteration. While the first iteration $u^1(x, \Delta t)$ is centered at $t = 0$, the second iteration $u(x, \Delta t + \Delta t)$ is centered at $t = \Delta t$, and so on. This is the same kind of "built–in analytic continuation" that makes finite difference methods so effective as well.

Further HSN Methods

We mentioned a variety of HSN methods in the debut paper earlier this year.[1] We will take a closer look at them in this Section. We'll continue to use the advection equation as our model to build a solid understanding. Consider first the predictor–corrector method for ordinary differential equations (ODEs) known as Heun's method. In semi-symbolic form this becomes

$$\begin{aligned} u^* &= u^n - a\Delta t \frac{\partial u^n}{\partial x} \\ u^{n+1} &= u^n - \frac{a\Delta t}{2}\left(\frac{\partial u^*}{\partial x} + \frac{\partial u^n}{\partial x}\right). \end{aligned} \tag{56}$$

With the same initial condition tested above, Equation (54), three iterations of Heun's method for $a = -1$ and $x = 1$ gives $h(t) = (\alpha + \beta)/8$, with $\alpha = -45t^6 + 90t^5 - 54t^4$ and $\beta = 18t^2 - 12t + 4$. Now we compare this solution with the Taylor series and the exact solution in Figure 2 and note the Heun HSN method is slightly less accurate than the one–step HSN. Formally, the Heun HSN is a second–order method while the one–step HSN of degree-three is third order accurate.

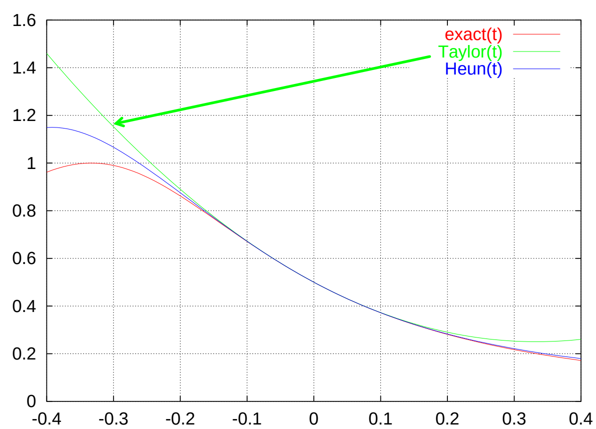

Figure 2 Comparison of Heun HSN, Taylor series, and the exact solution of the advection equation.

Another hybrid symbolic-numeric (HSN) method is a revised version of the two-step Runge-Kutta method

$$\begin{aligned} u^* &= u^n - \frac{a\Delta t}{2}\frac{\partial u^n}{\partial x} \\ u^{n+1} &= u^n - a\Delta t \frac{\partial u^*}{\partial x}. \end{aligned} \tag{57}$$

Also, the HSN version of the four-stage Runge-Kutta (RK) method becomes

$$\begin{aligned} u^* &= u^n - \frac{a\Delta t}{2}\frac{\partial u^n}{\partial x} \\ u^{**} &= u^n - \frac{a\Delta t}{2}\frac{\partial u^*}{\partial x} \\ u^{***} &= u^n - a\Delta t \frac{\partial u^{**}}{\partial x} \\ u^{n+1} &= u^n - \frac{a\Delta t}{6}\left(\frac{\partial u^n}{\partial x} + 2\frac{\partial u^*}{\partial x} + 2\frac{\partial u^{**}}{\partial x} + \frac{\partial u^{***}}{\partial x}\right). \end{aligned} \tag{58}$$

Again with the same initial condition tested above, Equation (54), three iterations of the four-stage RK-HSN method evaluated at $a = -1$ and $x = 1$ gives

$$\frac{\alpha + \beta + \gamma}{1536} \tag{59}$$

where $\alpha = -51975x^{12} + 132300x^{10} - 170100x^9$, $\beta = 91560x^8 - 37760x^6 + 30720x^5$, and $\gamma = -11904x^4 + 3264x^2 - 2304x + 768$.

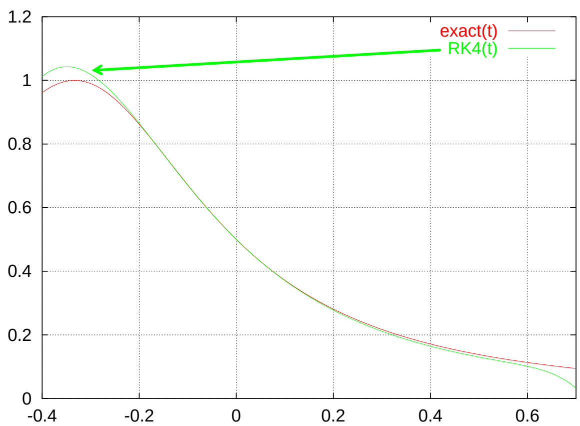

Figure 3 Comparison of Runge-Kutta HSN and the exact solution of the advection equation.

Figure 3 gives a comparison of this method with the exact solution. Note that as compared in Figures 1–3, the one-step HSN method of degree-three gives the best overall approximate solution for this case.

These methods work quite well for a linear hyperbolic equation. However, as is well known, nonlinear equations more often than not present peculiar difficulties that have challenged numerical methods for many years. We have hinted at some of these difficulties in context of the HSN method previously and intend to pursue this development further. Consider, for example, again the dimensionless form of the viscous Burgers equation

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} \tag{60}$$

where Re is the Reynolds number and we have initial condition $u(x, 0) = u_0(x)$.

A two step method was presented by Derickson and Pielke[7] and was reviewed by the present authors.[1] A derivation using an approach similar to the degree-k one-step HSN can also be applied to this equation, namely the Taylor series in time. Substituting the differential relation (60) into each time derivative yields an approximation ready for use with various schemes. Truncating at the first–order derivative, we have

$$u(x, t + \Delta t) \approx u(x, t) + \Delta t F(u) \tag{61}$$

where we define

$$F(u) = -u\frac{\partial u}{\partial x} + \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} \tag{62}$$

which is the right–hand side of Burgers' equation at a given time. A one–step HSN method could then be written as

$$u^{n+1}(x) \approx u^n(x) + \Delta t F(u^n) \tag{63}$$

Again, it can be shown that the order is the same as that of the truncated Taylor series (first–order in this case). Multi–step methods are easily developed from what we outlined above: Heun's predictor–corrector idea now yields

$$\begin{aligned} u^*(x) &= u^n(x) + \Delta t F(u^n) \\ u^{n+1}(x) &= u^n(x) + \frac{\Delta t}{2}\left[F(u^*) + F(u^n)\right]; \end{aligned} \tag{64}$$

while a Runge–Kutta two–step method would thus be

$$\begin{aligned} u^* &= u^n + \frac{\Delta t}{2}F(u^n) \\ u^{n+1} &= u^n + \Delta t F(u^*), \end{aligned} \tag{65}$$

and MacCormack's essential idea (full–step predictor followed by a half–step corrector) in hybrid symbolic–numeric form would be

$$\begin{aligned} u^* &= u^n + \Delta t F(u^n) \\ u^{n+1} &= \tfrac{1}{2}(u^n + u^*) + \frac{\Delta t}{2}F(u^*). \end{aligned} \tag{66}$$

Including higher–order terms in any of these sequences is straight–forward; however for this nonlinear case, the gain in accuracy does not compensate for the exponential increase in the number of terms that need to be carried through. Even two–step methods, which are generally second–order accurate, quickly accumulate terms that can slow down the calculation considerably, as noted by Derickson and Pielke.[7] We demonstrate how this problem can be handled by example.

Comparison with Taylor Series

Consider the method of Derickson and Pielke[7] for Burgers' equation adapted to the case of infinite Reynolds number, i.e. to the inviscid Burgers' equation

$$u_t + uu_x = 0. \tag{67}$$

The method, given $u^0(x) = u(x,0)$ as initial function, is

$$\begin{aligned} u^* &= u^n + tB(u^n) \\ u^{n+1} &= u^n + \frac{t}{2}(B(u^n) + B(u^*)) \end{aligned} \tag{68}$$

where B is the operator, a limiting form of (62),

$$B(f) = -f\frac{\partial f}{\partial x}. \tag{69}$$

The series ansatz

$$u(x,t) = \sum_{n=0}^{\infty} u_n(x)t^n$$

gives rise to the following recursion for the coefficients u_n where $u_0(x) = u(x,0)$:

$$(n+1)u_{n+1} = -\sum_{k=0}^{n} u_k u'_{n-k} \tag{70}$$

where the prime denotes derivative with respect to x.

As an example, let the initial function now be given by $u_0(x) = x^2$. Direct computation shows that the exact solution is

$$u(x,t) = \frac{1 + 2xt - \sqrt{1 + 4xt}}{2t^2}. \tag{71}$$

Three time step iterations of the algorithm (68) produces a polynomial of degree 63 in t (with coefficients that are polynomials in x) giving an approximation $v(x,3t)$ of $u(x,3t)$ and the Taylor series approximation $w(x,t)$ up to degree 63 evaluated at t replaced by $3t$ also gives an approximation to $u(x,3t)$. Consider $u(1,r)$, $v(1,r)$, and $w(1,r)$ for $r = 1/4, 1/3$. Each of these were computed using *ZExprLib* , the exact rational coefficient version of *ExprLib* . The *final* rational answers were coerced to double precision. We have

$$\begin{aligned} \text{Exact}: \quad u(1,\tfrac{1}{4}) &= 0.6862915 && (72) \\ \text{HSN}: \quad v(1,\tfrac{1}{4}) &= 0.6894361 && (73) \\ \text{Taylor}: \quad w(1,\tfrac{1}{4}) &= 0.6841509 && (74) \end{aligned}$$

while

$$\begin{aligned} \text{Exact}: \quad u(1,\tfrac{1}{3}) &= 0.6261360 && (75) \\ \text{HSN}: \quad v(1,\tfrac{1}{3}) &= 0.6314292 && (76) \\ \text{Taylor}: \quad w(1,\tfrac{1}{3}) &= -182133.402 && (77) \end{aligned}$$

Clearly, the HSN method has produced an approximation that gives good values beyond the radius of convergence of the Taylor approximation (which happens to be $\frac{1}{4}$). The degree 63 term of the method given by (68) is

$$-215482439153543760000000t^{63}x^{65}$$

whereas the analogous term of the Taylor series approximation is

$$-3684791698758166594790090427135469 50t^{63}x^{65}.$$

Series solutions suffer from the cumbersome algebra required when doing calculations by hand. Perhaps this explains why this approach lost favor once computers were utilized for finite difference calculations. Now its possible to let the computer do the algebra and devise series-like methods that are computationally efficient. Here is the result of timing the *ZExprLib* program using (68):

```
real   0m0.031s
user   0m0.020s
sys    0m0.010s
```

Such results give one hope that the emerging approach will allow the development of new methods which are competitive with current techniques but allow much more creativity when implementing the mathematics. One is not limited to Taylor series only, which is the basis for finite-difference schemes.

Another One-Step Method

As a final note, consider the Burgers equation and a partition of the time variable

$$0 = t_0 < t_1 < \ldots < t_n < \ldots$$

once more. Integrating over the interval $[t_n, t_{n+1}]$ gives

$$\int_{t_n}^{t_{n+1}} u_t dt = \int_{t_n}^{t_{n+1}} (-uu_x + \frac{1}{Re}u_{x,x})dt. \tag{78}$$

This time, using Simpson's rule for the right hand side, we obtain

$$u^{n+1} - u^n = \tfrac{1}{3}\left(F(u)\mid_{t_n} + 4F(u)\mid_{t_{n+\frac{1}{2}}} + F(u)\mid_{t_{n+1}}\right) \tag{79}$$

where as before $F(u) = -uu_x + \frac{1}{Re}u_{x,x}$. We thus get the one-step scheme

$$u^{n+1} = u^n + \tfrac{1}{3}\left(F(u^n) + 4F(u)\mid_{t_{n+\frac{1}{2}}} + F(u)\mid_{t_{n+1}}\right) \tag{80}$$

and we can use a second order Taylor series approximation for the last two terms:

$$F(u)(x,(n+\tfrac{1}{2})\Delta t) = F(u^n) + F(u^n)_t\frac{\Delta t}{2} + F(u^n)_{t,t}(\frac{\Delta t}{2})^2 \tag{81}$$

$$F(u)(x,(n+1)\Delta t) = F(u^n) + F(u^n)_t\Delta t + F(u^n)_{t,t}(\Delta t)^2. \tag{82}$$

Assuming that u has continuous first partial-derivatives, we can evaluate $F(u^n)_t$ as

$$F(u^n)_t = -u_tu_x - uu_{t,x} + \frac{1}{Re}u_{t,x,x} \tag{83}$$

and similarly for $F(u^n)_{t,t}$. These were calculated using *ExprLib* to obtain the first derivative

$$F(u^n)_t = \frac{1}{Re^2}\frac{\partial^4 u}{\partial x^4} - \frac{2}{Re}u\frac{\partial^3 u}{\partial x^3} + \left(u^2 - \frac{4}{Re}\frac{\partial u}{\partial x}\right)\frac{\partial^2 u}{\partial x^2} + 2u\left(\frac{\partial u}{\partial x}\right)^2 \tag{84}$$

and the second derivative

$$\begin{aligned} F(u^n)_{t,t} = & -\frac{3}{Re^2}u\frac{\partial^5 u}{\partial x^5} + \left(\frac{1}{Re^3} - \frac{9}{Re^2}\frac{\partial u}{\partial x} + \frac{3}{Re}u^2\right)\frac{\partial^4 u}{\partial x^4} + \left(\frac{18}{Re}u\frac{\partial u}{\partial x} - \frac{16}{Re^2}\frac{\partial^2 u}{\partial x^2} - u^3\right)\frac{\partial^3 u}{\partial x^3} \\ & + \left(\frac{18}{Re}\left(\frac{\partial^2 u}{\partial x^2}\right)^2 - 9u^2\frac{\partial u}{\partial x}\right)\frac{\partial^2 u}{\partial x^2} - 6u\left(\frac{\partial u}{\partial x}\right)^3 + \frac{12}{Re}u\left(\frac{\partial^2 u}{\partial x^2}\right)^2. \end{aligned} \tag{85}$$

where we omitted the time index n on the right-hand side for clarity. As can be seen, the algebraic manipulations become cumbersome but present no problem for a machine using symbolic computation. We defer the presentation of results with this method to a future paper, when our understanding is more complete. Our intent here is solely to demonstrate how symbolic computation allows for the development of algorithms for solving these kinds of problems (the solution of PDEs). We have no doubt as to the effectiveness of the tools now at hand and are eager to show the full results for nonlinear scalar conservation laws as well as systems of equations. Our final section is a brief step towards this goal, which we hope to have in our grasp by the sequel.

Euler Equations

We will show briefly that the methodology outlined above for linear and nonlinear scalar differential equations can be extended to solve systems of equations. The one-dimensional form of the Euler gas dynamic equations can be written as

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) &= 0 \\ \frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2 + p) &= 0 \\ \frac{\partial}{\partial t}(\rho e) + \frac{\partial}{\partial x}(\rho u e + p u) &= 0 \end{aligned} \tag{86}$$

where $e = \frac{1}{2}u^2 + p/(\gamma - 1)$ is the total energy and γ represents the ratio of specific heats. Depending on preference, one may use this equation of state to solve for the variables $\{\rho, u, e\}$ or $\{\rho, u, p\}$. We may assume initial conditions

$$\begin{aligned} \rho(x,0) &= \rho_0(x) \\ u(x,0) &= u_0(x) \\ p(x,0) &= p_0(x) \end{aligned} \tag{87}$$

where ξ_0 is a given function of its argument. A similar approach to the degree k one-step HSN method can also be applied to these equations, namely the Taylor series in time. First, we write the system of equations in symbolic form as the single conservation law

$$\frac{\partial \mathcal{Q}}{\partial t} + \frac{\partial \mathcal{F}}{\partial x} = 0 \longrightarrow \frac{\partial \mathcal{Q}}{\partial t} = -\frac{\partial \mathcal{F}}{\partial x} \tag{88}$$

Substituting the differential relation (88) into each time derivative yields an approximation ready for use with various schemes. Truncating at the first-order derivative, we have

$$\mathcal{Q}(x, t + \Delta t) \approx \mathcal{Q}(x,t) + \Delta t G(\mathcal{Q}) \tag{89}$$

where we define

$$G(\mathcal{Q}) = -\frac{\partial \mathcal{F}}{\partial x} \tag{90}$$

which is the right-hand side of Burgers' equation at a given time. A one-step HSN method could then be written as

$$\mathcal{Q}^{n+1}(x) \approx \mathcal{Q}^n(x) + \Delta t G(\mathcal{Q}^n) \tag{91}$$

Again, it can be shown that the order is the same as that of the truncated Taylor series (first-order in this case). Multi-step methods are easily developed from what we outlined above: Heun's predictor-corrector, Runge-Kutta two-step, and a hybrid symbolic-numeric form of MacCormack's scheme. Including higher-order terms in the sequence is again straight-forward; however, as noted for the nonlinear scalar case, the gain in accuracy does not compensate for the exponential increase in the number of terms that need to be carried through. Even two-step methods, which are generally second-order accurate, quickly accumulate terms that can slow down the calculation considerably, as noted by Derickson and Pielke.[7] We demonstrated how this problem can be handled by example for Burgers' equation. We are working on a similar approach to show that the methodology can be extended to systems of equations as our understanding matures.

Conclusions

We have continued developing a new approach to the solution of differential equations with particular attention to problems in CFD. In this paper, the basic tools were explored, developed, and implemented in a way that will facilitate the construction of new algorithms, hybrid symbolic numerical schemes, that have the potential to revitalize the computational approach to solving problems of interest in physics and engineering. This new approach is not a method in itself but simply the use of modern computing power for the application of mathematics at a very high (deep?) level. We believe that the methods emerging will change the landscape of scientific (i.e. mathematical/physical) computing in much the same way that finite–difference techniques transformed the numerical solution to the mathematical equations of interest: By enhancing the mental capabilities of its creators.

Acknowledgments

This work was sponsored by a grant from the Chief Scientist Innovative Research Fund of the Air Vehicles Directorate, U. S. Air Force Research Laboratory (AFRL/VA CSIRF). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory, Dept. of Defense, or the U.S. Government.

References

[1]Camberos, J. A., Lambe, L., and Luczak, R., "CFD with Hybrid Symbolic Numeric Computatioin," *AIAA Paper 2004–0242*, Reno, NV, January 2004.

[2]Lambe, L., Luczak, R., and Nehrbass, J., "A New Finite Difference Method for the Helmholtz Equation Using Symbolic Computation," *International Journal of Computational Engineering Science*, Vol. 4, No. 2, 2003.

[3]Lambe, L. A., Richard, L., and Nehrbass, J. W., "Symbolic computation in electromagnetic modeling," The 2001 Electromagnetic Code Consortium (EMCC) Annual Meeting (May).

[4]Lambe, L. A., Richard, L., and Nehrbass, J. W., "Symbolic computation in electromagnetic modeling," The DOD UGC 2001 Meeting (June).

[5]Lambe, L. A., "ExprLib Workshop," February 2001, Wright Patterson AFB.

[6]Lambe, L. A., "ExprLib Training Workshop," November 2003, Wright Patterson AFB.

[7]Derickson, R. G. and Pielke, Sr., R. A., "A preliminary study of the Burgers equation with symbolic computation," *Journal of Computational Physics*, Vol. 162, 2000, pp. 219–244.

[8]Knopp, K., *Theory and Applications of Infinite Series*, Blackie and Son, Glasgow, 1928, transl. by Ms. R.C. Young.

[9]Blasius, H., "The boundry layers in fluids with little friction," Technical Memorandum 1256, National Advisory Committee for Aeronautics, Washington, D.C, 1950, http://naca.larc.nasa.gov/.

[10]van Dyke, M. D., "Computer-Extended Series," *Annual Review of Fluid Mechanics*, Vol. 16, 1984, pp. 287–309.

[11]John, F., "A note on "improper" problems in partial differential equations," *Comm. Pure Appl. Math.*, Vol. 8, 1955, pp. 591–594.

[12]Sugai, I., "Numerical solution of Laplace's equation, given Cauchy conditions," *IBM J. Res. Develop.*, Vol. 3, 1959, pp. 187–188.