43rd AIAA Aerospace Sciences Meeting
10–13 January 2005, Reno, NV

AIAA #2005–0092

Hybrid Symbolic-Numeric Methodology: Views and Visions

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 ‡
Lockheed Martin Aeronautics Company
Fort Worth, TX 76101

We used computer aided symbolic computation in our previous papers[1,2] to explore solutions of partial differential equations in a way that involves a synergistic application of symbolic and numeric methodologies. Here we review the results thus far and present a few examples of this emerging methodology applied to the nonlinear Burgers equation. The continued development of the hybrid methodology is presented with the hope of inspiring new thought in utilizing hybrid symbolic and numeric manipulation for solving the equations of mathematical physics from a fundamental perspective: A new kind of thinking beyond what we call computer science today. We see an opportunity to revisit the very foundations of scientific computing armed with new symbolic computational tools that complement the numerical power of modern machines.

Introduction

We presented a brief overview of the use of computers for symbolic computation in Camberos, Lambe, and Luczak[1,2] called Paper I and Paper II in this note. In these two papers, we introduced hybrid symbolic-numeric (HSN) schemes using symbolic manipulation of expressions based on *ExprLib*[3–7] which is an ANSI C library designed to meet some stringent needs of application developers and problem solvers in the area of scientific computing. Specifically, several methods were developed as inspired by the Finite–Difference Method (FDM), such as the Heun, MacCormak, and Lax-Wendroff schemes. We also adapted the Runga-Kutta method in this context and looked at a method given by Derickson and Pielke[8] and applied some of these methods to several model equations including pure advection, heat, and Burgers' equation with encouraging results. For linear equations, we introduced a one–step HSN method that represented a unifying perspective for finite–difference methods. The semi–discrete form of the equation is in fact illustrated by this one–step HSN method. In this note, we want to review what we have learned so far and take a look towards the future.

The core of our effort is really an attempt to better integrate mathematics with the computer science needed to solve the differential equations of interest. This may perhaps inspire new thought and new capabilities for analyzing, developing, and optimizing numerical algorithms for best results in modelling and simulation.

The fact that others have achieved some level of success in applying related ideas gives hope that an extended approach will prove feasible for more general equations as well.[8–11] We first highlight and review the significant points of our first two papers and then summarize some emerging results from extending the approach to solve nonlinear problems, using the Burgers equation as the standard example.

*Aerospace Engineer and AIAA Associate Fellow (jose.camberos@wpafb.af.mil)
†Chief Scientist, MSSRC (llambe@mssrc.com)
‡Principal Systems Engineer (richard.luczak@lmco)

The One-Step HSN Method

A standard approach to the numerical approximation of partial differential equations like the advection equation

$$u_t + au_x = 0, \tag{1}$$

(the subscripts denote differentiation with respect to the indicated variable) begins with the Taylor series

$$u(x, t + \Delta t) = u(x, t) + \Delta t \frac{\partial u}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2} + \ldots + \frac{\Delta t^m}{m!} \frac{\partial^m u}{\partial t^m}. \tag{2}$$

Assuming we are within the radius of convergence, a solution method can be obtained by replacing all time derivatives using Equation (1). This gives

$$u(x, t + \Delta t) = u(x, t) - a\Delta t \frac{\partial u}{\partial x} + \frac{a^2 \Delta t^2}{2} \frac{\partial^2 u}{\partial x^2} + \cdots + \frac{(-a\Delta t)^m}{m!} \frac{\partial^m u}{\partial x^m}. \tag{3}$$

In developing the one-step HSN method, the prescribed initial condition is assumed to be sufficiently differentiable. We can extend the series to any given order. Using symbolic computation to evaluate the spatial derivative analytically, truncated at the second-order derivative, gives

$$u^{n+1}(x) \approx u^n(x) - a\Delta t \frac{\partial u^n}{\partial x} + \frac{a^2 \Delta t^2}{2} \frac{\partial^2 u^n}{\partial x^2}. \tag{4}$$

By calculating the derivative symbolically and using this explicit, one–step prediction formula (4), a semi-analytic approximation to the advection equation is obtained. A variety of other methods were studied from this perspective. For example, the predictor-corrector, or Heun's method, is

$$\begin{aligned} u^* &= u^n - a\Delta t \frac{\partial u^n}{\partial x} \\ u^{n+1} &= u^n - \frac{a\Delta t}{2} \left(\frac{\partial u^*}{\partial x} + \frac{\partial u^n}{\partial x} \right) \end{aligned} \tag{5}$$

Again, the spatial derivatives can be evaluated symbolically, given the initial condition $u_0(x)$. If we substitute for u^*, the one-step formula (4) results. The same holds true for the HSN version of the two-step Runge-Kutta method and of course the workhorse of ODE numerical methods, the four-stage Runge-Kutta method:

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

Substituting for $u^*, u^{**}, \ldots$ in turn, reduces the RK-4 method to the one-step HSN formula (4). The one-step HSN formula appears to unify methods used for integrating ODE's into a single fundamental formula for solving linear PDE's using symbolic computation.

Extension to Other Methods

Other well–known classical CFD methods like MacCormack's predictor-corrector method,[12] the two–step version of Lax–Wendroff, and the Warming–Beam method can be shown to reduce to (4) when written in hybrid form. So, for a linear equation with constant coefficient, all these methods reduce to the one-step HSN method, to some order m indicating the truncation term in the Taylor sequence (3). For nonlinear equations, these formulas hint at a variety of possible solution techniques, some of which we explore below.

In standard numerical methods, the differential equation is often first discretized in one variable (like space, requiring mesh generation) and then each point subjected to time integration, either point-by-point as in explicit methods or for a collection of points as in implicit methods. In the examples given above, the spatial continuity is retained and time integration applied to the function itself, not a collection of numerical values at given points. The accuracy and cost of the technique can be compared to standard methods by example. In addition, the HSN method allows computations involving symbolic parameters: The advection speed a in (1) can be kept symbolic throughout the computation.

Paper II

The initial success encountered in our first efforts to develop the HSN method inspired our efforts to press forward and revisit the topic of series solutions with some new and powerful tools for symbolic computation. A summary of what we learned in the second paper[2] follows.

Series Methods

Series methods have been around for a long time. They form a basic mathematical approach for solving differential equations in standard texts. See for example Knopp.[13] It turns out that many of the model equations for CFD have exact solutions that can be derived using series methods which present classical solutions in a different light. The manner in which these solutions were obtained through symbolic computation may be of some pedagogical interest. Among the equations whose analytic solutions were found using symbolic computation and series methods were the convection–diffusion problem

$$u' = \frac{1}{p}u'', u(0) = a,\ u(1) = b \tag{7}$$

with solution

$$u(x) = a + \frac{b-a}{e^{pb}-1}\left(e^{px}-1\right), \tag{8}$$

and the complementary initial value problem: $u'' = pu'$, $u(0) = a$, $u'(0) = b$ with solution

$$u(x) = a + \frac{b}{p}\left(e^{px}-1\right). \tag{9}$$

We also considered the advection equation (1) with initial condition

$$u(x,0) = u_0(x) \tag{10}$$

from this perspective. It led directly to the classic solution $u_0(x-at)$. Analysis of the second-order wave equation in one space dimension

$$u_{t,t} = c^2 u_{x,x} \tag{11}$$

with initial conditions

$$u(x,0) = f(x) \tag{12}$$

$$u_t(x,0) = g(x). \tag{13}$$

led directly to the famous D'Alembert formula

$$u(x,t) = \tfrac{1}{2}\left(f(x+ct) + f(x-ct)\right) + \tfrac{1}{2c}\int_{x-ct}^{x+ct} g(y)dy. \tag{14}$$

The procedure relied entirely on manipulating the series anzatz using symbolic computation.

We also looked at Laplace's equation with initial conditions

$$u(x,0) = f(x) \tag{15}$$

$$u_y(x,0) = g(x). \tag{16}$$

and curiously found that its solution was given by a complex form of the D'Alembert formula, viz. letting $G(x) = \int^x g(z)dz$ then

$$u(x,y) = \tfrac{1}{2}\left(f(x+kyi) + f(x-kyi)\right) + \tfrac{1}{ki}\left(G(x+kyi) - G(x-kyi)\right). \tag{17}$$

Another simple series anzatz led to the following solution to the linear dispersive (KdV) equation

$$u_t + au_x + du_{x,x,x} = 0. \tag{18}$$

The series solution led directly to the analytic solution

$$u(x,t) = \frac{12db^2}{a}\frac{\exp(2\beta)}{(1+\exp(2\beta))^2} = \frac{3db^2}{a}\mathrm{sech}^2(\beta) \tag{19}$$

where

$$\beta = \tfrac{1}{2}\left(bx - b^3t + \ln\left(\frac{aa_1}{12db^2}\right)\right). \tag{20}$$

We also compared some HSN methods including the one-step, Heun, and Runge-Kutta to a Taylor series solution and noted the advantage of these methods, as well as finite-difference methods, to endow the numerical calculation with a built-in form of analytic continuation. The interested reader can consult Paper II[2] for graphs and other computational results.

Some Further Experiments

Burgers' Equation

Several possible methods, based on a nonlinear version of the HSN method and others based on series solutions were explored for solving the viscous Burgers' equation

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} \tag{21}$$

where Re is the Reynolds number. We assume the initial condition $u(x,0) = u_0(x)$, where u_0 is a given function. A pioneering attempt to solve (21) using symbolic computation was reported by Derickson and Pielke,[8] where the authors presented a two-step method for integrating in time. In Paper I, we reviewed this two-step method and noted that it was essentially Heun's method for solving ODEs. In subsequent conversations with the first author of that paper, we were informed[14] that considerable time and effort was used in pruning, by hand, the output of a calculation of the algorithm using a standard computer algebra system in order to control the expression swell while at the same time maintaining relevant information. Using the pruning facilities available in *ExprLib* we were able to write a C program which executed the algorithm on many examples using the automatic pruning techniques developed. By keeping the Reynolds number symbolic and substituting various values one can see the great potential for implementing parametric studies that do not require re-executing the program each time the parameter is changed. Instead, a single solution may be obtained with subsequent substitution of numerical values for the relevant parameter. We can report that our results agree nicely with those found by Derickson and Pielke.[8] See Figure 1. The sequence of numbers indicate time steps and the upper half of the initial sine wave shifts to the right, the lower half shifts to the left. In the limit $Re \to \infty$ a discontinuous shock solution emerges, creating a saw-tooth shape that decays in time.

Linear Advection with Systems of Equations

The power of the methodology we have explored can be demonstrated by the linear advection equation generalized in vector form,

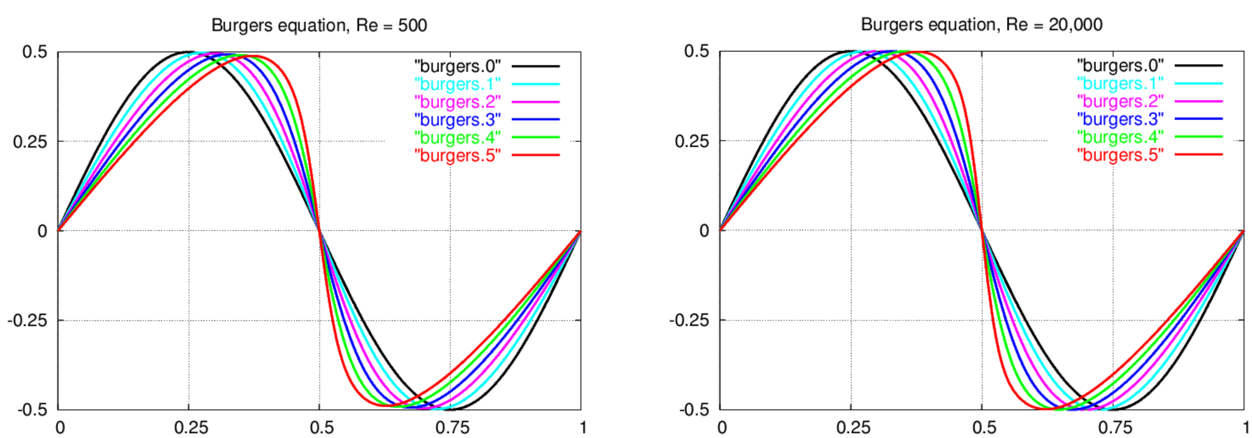
$$[u]_t + \boldsymbol{A} \cdot [u]_x = 0 \tag{22}$$

where $[u] : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n$ and $\boldsymbol{A}$ is an $n \times n$ matrix with real entries. The Cauchy problem requires the solution $[u]$ given the initial function(s) $[u]_0(x) = [u](x,0)$. Of course for $n = 1$, we have the scalar advection equation with solution $u(x,t) = u_0(x - at)$. A general solution can be constructed for arbitrary n if $\boldsymbol{A}$ has real eigenvalues $\lambda_1, ..., \lambda_n$ with corresponding eigenvectors $\nu_1, ..., \nu_n$ that are linearly independent. Thus the matrix $\boldsymbol{P}$ with columns equal to the ν_i is invertible, giving the diagonal matrix $\boldsymbol{D} = \boldsymbol{P}^{-1} \cdot \boldsymbol{A}\boldsymbol{P}$, with diagonal entries $\lambda_1, ..., \lambda_n$. We know that setting $[w] = \boldsymbol{P}^{-1} \cdot [u]$, we get

$$\boldsymbol{P}^{-1} \cdot [u]_t + \boldsymbol{P}^{-1} \cdot \boldsymbol{A} \cdot [u]_x = 0 \quad \text{or} \quad \longrightarrow \quad [w]_t + \boldsymbol{D} \cdot [w]_x = 0. \tag{23}$$

The decoupled system consists of n one-dimensional advection equations with solution

$$[y](x,t) = \begin{bmatrix} w_0^1(x - \lambda_1 t) \\ \vdots \\ w_0^n(x - \lambda_n t) \end{bmatrix} \tag{24}$$



a) Re = 500. **b)** Re = 20,000.

Figure 1 Time evolution of smooth initial function for Burgers' equation.

where $[w] = \boldsymbol{P}^{-1} \cdot [u](x,0)$. The desired solution is therefore $[u] = \boldsymbol{P} \cdot [y]$. For example, choosing $u_0^1 = \tanh(x^2 - 1)$, $u_0^2 = \sin(x)$ and keeping a and b symbolic the solution for

$$\frac{\partial u^1}{\partial t} = a \frac{\partial u^1}{\partial x} + b \frac{\partial u^2}{\partial x} \tag{25}$$

$$\frac{\partial u^2}{\partial t} = b \frac{\partial u^1}{\partial x} + a \frac{\partial u^2}{\partial x} \tag{26}$$

can be computed symbolically as[15]

$$\begin{aligned} u^1 \;=\; & 0.5 \tanh\left[x^2 + 2(b-a)xt + (a^2 - 2ab + b^2)t^2 - 1\right] \\ & + 0.5 \tanh\left[x^2 - 2(b+a)xt + (a^2 + 2ab + b^2)t^2 - 1\right] - 0.5 \sin[x - (a-b)t] + 0.5 \sin\left[x - (a+b)t\right] \end{aligned} \tag{27}$$

$$\begin{aligned} u^2 \;=\; & 0.5 \tanh\left[x^2 - 2(a+b)xt + (a^2 + 2ab + b^2)t^2 - 1\right] \\ & - 0.5 \tanh\left[x^2 + 2(b-a)xt + (a^2 - 2ab + b^2)t^2 - 1\right] + 0.5 \sin[x - (a-b)t] + 0.5 \sin[x - (a+b)t)] \end{aligned} \tag{28}$$

Burgers' Equation and Pole Dynamics

In exploring the considerable quantity of literature on Burgers' equation we came across an interesting series of papers[16–20] and selected references therein. These papers analyze a connection between the behavior of poles in a complexification of the solution to the viscous Burgers' equation with a cubic initial function. The analysis is facilitated through the use of the exact solution as obtained by the Hopf-Cole transformation. We were curious to see what could be done in the context of HSN and *ExprLib* so we carried out another experiment with the one-dimensional Burgers' equation using a polynomial initial function. Here, we used the HSN methods for the viscous equation and extended the domain of our solution approximation beyond the range normally obtained with finite difference methods by using a sequence of high precision Padé rational approximations of the multivariate polynomial expressions. This approach allowed the "capturing" of poles for the approximate analytic solution. By taking a high enough degree in the numerator, we obtained symbolic expressions and used them to analyze the behavior of the poles in time. This was an effective implementation of the algorithm from Derickson and Pielke[8] with *ZExprLib*, the exact coefficient version of the *ExprLib* library. Here is an actual code segment from the program followed by the results:

```
nTimeStep = 10;

u = parseStrToZExpr ("x^3-5*x");      /* breaking time t=1/5 */
dt = parseStrToZExpr ("dt");
dt2 = parseStrToZExpr ("dt / 2");

for (i = 0; i < nTimeStep; i++)
  {
    /* an HSN method for Burgers' equation */
    U = capU (u);
    uu = zExprPlus (u, zExprTimes (dt, U)); /* u^* = u^n + dt U^n */
    UU = capU (uu);
    a = zExprTimes (dt2, zExprPlus (U, UU));
    u = zExprPlus (u, a);          /* u^{n+1} + dt/2 (U^n + UU^n) */
```

In the code above, the routine `capU (u)` simply returns $-uu_x + eu_{x,x}$. The code continues:

```
    /* evaluate ans at dt = 1/50, e = 0 */
    var[0] = "dt";
    var[1] = "e";
    val[0] = parseStrToZExpr ("1/50");
    val[1] = parseStrToZExpr ("0");
    ans = zExprEval (ans, var, val, 2);
    /* pick out the numerator and denominator */
    num = zExprNum (ans);
    den = zExprDen (ans); k = 2;
```

```
    /* this is a rational function of degree <= two
       in both the numerator and denominator */
    r = zExprPade (k, 2, num, den);

    printf ("step %d: toDouble pade (%d,2) = ", i+1, k);
    /* convert exact coefficients to doubles */
    printf ("%s\n\n", zExprToDoubStr (r));
  }
```

Here are the results for 10 steps.

```
step 1: toDouble pade (2,2) = (-20.4445 * x + 1) / (x^2 + 3.68203)

step 2: toDouble pade (2,2) = (-16.3652 * x + 1) / (x^2 + 2.62205)

step 3: toDouble pade (2,2) = (-12.7611 * x + 1) / (x^2 + 1.79133)

step 4: toDouble pade (2,2) = (-9.6304 * x + 1) / (x^2 + 1.16117)

step 5: toDouble pade (2,2) = (-6.97062 * x + 1) / (x^2 + 0.702931)

step 6: toDouble pade (2,2) = (-4.77983 * x + 1) / (x^2 + 0.388227)

step 7: toDouble pade (2,2) = (-3.07217 * x + 1) / (x^2 + 0.18988)

step 8: toDouble pade (2,2) = (-1.90468 * x + 1) / (x^2 + 0.0816807)

step 9: toDouble pade (2,2) = (-1.31816 * x + 1) / (x^2 + 0.0336045)

step 10: toDouble pade (2,2) = (-1.29784 * x + 1) / (x^2 + 0.0156453)
```

These results agree with those in the cited papers. Note that the poles are initially purely complex and placed symmetrically along the imaginary axis. As time t approached the breaking time, the poles coalesce to the real x-axis. It appears that there exists a relationship between the dynamics of pole locations in the solution to Burgers equation and the occurrence of shocks in the corresponding, physically relevant, entropy-solution. In a references cited above, which we found after the fact, the authors used the Hopf-Cole transformation to study an exact solution to Burgers equation with cubic initial function and show this same result. As opposed to the methods used therein, we only used analytic Padé approximations to study poles; the exact solution was not needed. We expect that these methods will generalize to higher dimensions and to systems of equations so that we can continue to develop a technique to capture, observe, and track singularities.

Visions

Discontinuous Initial Data

Engineers have been dealing with the problem of modelling discontinuous initial conditions for some time now and it was natural for us to wonder about how to use HSN in that context. We thus began exploring the general theory of existence of solutions for hyperbolic and slightly viscous equations. In the case of one-dimensional conservation laws, global existence of weak solutions when the initial function has small total variation goes back to the 1960's. In this case, it is known that if one adds a viscous term of the form $\epsilon u_{x,x}$ to the inviscid equation (and for each ϵ, one considers the same initial function as the inviscid case), a physically–relevant inviscid weak solution is obtained by taking the limit as $\epsilon \to 0$ for the viscous solution. In practice, this correct weak solution is realized in finite-difference methods either implicitly as in filtering, limiting, etc. or explicitly as in upwind methods. For the hybrid symbolic-numeric (HSN) methods developed, the degree of differentiability required in the initial functions would seem to limit the scope of such methods since it seems to rule out discontinuous initial conditions and the solution to the well-known Riemann problem. However, we used *ExprLib* to conduct the following experiment for the 1D advection equation (1) and corresponding viscous equation

$$u_t + au_x = \epsilon u_{x,x} \tag{29}$$

where we allow the initial function to vary with ϵ in a way that in the limit as $\epsilon \to 0$ we get the initial conditions for the Riemann problem (this can be done with various algebraic combinations of the hyperbolic tangent). The experiments were quite successful and we subsequently were able to verify them theoretically (in fact, the viscous equation can be transformed to the heat equation via a straightforward transformation and one proceeds from there). In higher dimensions, the theory is lacking and we expect that our HSN approach will lead to new insights. The *ExprLib* library makes such experiments that require symbolic computation not only possible but convenient.

Further experimentation led us to the following solution to Burgers' equation (parametrized in ϵ)

$$u_t^\epsilon + u^\epsilon u_x^\epsilon = \epsilon u_{x,x}^\epsilon \qquad \longrightarrow \qquad u(x,t)^\epsilon = 1 - \tfrac{1}{2}\left\{1 - \tanh\left[\left(\tfrac{1}{2}t - x\right)/(4\epsilon)\right]\right\} \tag{30}$$

which may or may not be well known. The important point is that the limit as $\epsilon \to 0$ of the initial function is a step function, i.e. the limit as $\epsilon \to 0$, u^ϵ is apparently a solution to a Riemann problem. Since u^ϵ is smooth, one could hope to develop an HSN method for approximating it starting from the initial function

$$u(x,0)^\epsilon = 1 - \tfrac{1}{2}\left\{1 - \tanh[(-x)/(4\epsilon)]\right\} \tag{31}$$

which is also smooth. In fact, it is not hard to see that any step function can be given by a hyperbolic tangent function (suitably adorned with some parameters) and furthermore the hyperbolic tangent is not the only choice. While we have made some initial computations, we do not yet see any practical applications of this fact. It may however be of some theoretical use in understanding state-of-the-art schemes which require the (approximate) solution of Riemann problems. It is possible that the two experiments mentioned above may be related or combined in a profitable way. Further experiments are needed and may shed more light on a challenging area.

Singularities

As we continued to understand the dynamics of solutions to Burgers equations and began to turn towards more sophisticated models of fluid motion, we found many thought-provoking articles. On interesting title that caught our attention: "Singularities out of Euler Flow? Not out of the Blue!", where the authors ask the question[21]

> Does three-dimensional incompressible Euler flow with smooth initial conditions develop a singularity with infinite vorticity in finite time?

Apparently, this question is yet unsolved at the time of this writing. The authors of that paper also noted the case of Burgers' equation as we examined in the last section and they gave a numerical study of singularities for 2D Euler flow. It is worth pointing out again that in the last section, we found that knowledge of the exact solution was not needed in order to carry out the analysis. The approximate analytic answer obtained via HSN (along with a singularity study involving the Padé approximation) was sufficient. We think that an appropriate generalization of hybrid symbolic-numeric methods (of which we have only scratched the surface in these three papers) is not only possible for the Euler, Navier-Stokes and other equations, but can lead to insights which in turn can lead to new robust and efficient algorithms for approximating solutions to these equations. With some long hard work and proper funding support, we believe that such methods could indeed be realized.

The View Back – The View Forward

Curiously, after carrying out the study of pole dynamics for Burgers' equation, we found a very interesting paper[22] by F. C. Gey and M. B. Lesser from 1969 which also studies Burgers' equation using computer aided symbolic computation in the form of a library. In their case, they use a FORTRAN extension that was called ALTRAN which was a system for rational function manipulation.[23–26] While they used a different HSN method than we do in the previous section, they did polish off the solution with a Padé approximation and were very close to obtaining our results!

The references cited above amply demonstrate that people were interested in using symbolic computation on computers to solve scientific and engineering problems early on. The computational power needed to do this properly however did not exist at the time and we believe that this is the main reason that the ALTRAN paradigm did not evolve further, extinguishing its early and promising potential. Instead, efficient purely numerical methods became dominant. Numerical techniques, in the early days, were limited by available computing resources that forced the research direction towards fully discretized, purely numerical calculations. These techniques were quite successful in making the best use of available resources, requiring not only mathematics, but a new kind of thinking, a thinking that evolved into what we call computer science today. As computers continue to gain in computational and storage space, we see an opportunity to revisit the very foundations of scientific computing armed with new symbolic computational tools. If the foundations are explored with current and future technologies in hand, who knows what improvements and better ideas may be found!

Conclusions

Our goal has been to map out the development of algorithms and ideas that did not yet exist as well as reexamining others. We demonstrated the potential by realizing a few such algorithms and ideas. Now we hope to inspire others to continue with the long road ahead - one that offers a great chance of success and the possibility of developing new and interesting methods that were not feasible or evident before.

Acknowledgments

The authors gratefully acknowledge MSSRC for software support during the course of this research and beyond. We also thank the Chief Scientist Innovative Research Fund (CSIRF) of the Air Vehicles Directorate, U. S. Air Force Research Laboratory (AFRL/VA) for funding the work from our first two papers[1,2] on which the present work was based.

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, Department of Defense, or the U.S. Government.

References

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

[2]Camberos, J. A., Lambe, L., and Luczak, R., "Hybrid Symbolic Numeric Methodology for Fluid Dynamic Simulations," *AIAA Paper 2004–2330*, Portland, OR, June 2004.

[3]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.

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

[5]Lambe, L. A., Richard, L., and Nehrbass, J. W., "Symbolic computation in electromagnetic modeling," DOD UGC 2001.

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

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

[8]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.

[9]De Chant, L. A., Seidel, J. A., and Kline, T. R., "Extension of a Combined Analytical/Numerical Initial Value Problem Solver for Unsteady Periodic Flow," *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 1163–1186.

[10]Verhoff, A., "A Compact Formulation for Analytical Solution of the 2D Euler Equations," *AIAA Paper 2003–3918*, Orlando, Fl, June 2003.

[11]Boyd, J. P., *Chebyshev and Fourier spectral methods*, Dover Publications Inc., Mineola, NY, 2nd ed., 2001.

[12]MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," *AIAA Paper 69–0354*, Cincinnati, OH, 1969.

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

[14]Derickson, R., Personal communication, 2004.

[15]See <www.mssrc.com> for further examples.

[16]Weideman, J. A. C., "Computing the dynamics of complex singularities of nonlinear PDEs," *SIAM J. Appl. Dyn. Syst.*, Vol. 2, No. 2, 2003, pp. 171–186 (electronic).

[17]Senouf, D., "Dynamics and condensation of complex singularities for Burgers' equation. I," *SIAM J. Math. Anal.*, Vol. 28, No. 6, 1997, pp. 1457–1489.

[18]Senouf, D., "Dynamics and condensation of complex singularities for Burgers' equation. II," *SIAM J. Math. Anal.*, Vol. 28, No. 6, 1997, pp. 1490–1513.

[19]Senouf, D., Caflisch, R., and Ercolani, N., "Pole dynamics and oscillations for the complex Burgers equation in the small-dispersion limit," *Nonlinearity*, Vol. 9, No. 6, 1996, pp. 1671–1702.

[20]Bessis, D. and Fournier, J.-D., "Complex singularities and the Riemann surface for the Burgers equation," *Nonlinear physics (Shanghai, 1989)*, Res. Rep. Phys., Springer, Berlin, 1990, pp. 252–257.

[21]Frisch, U., Matsumoto, T., and Bec, J., "Singularities of Euler flow? Not out of the blue!" *J. Statist. Phys.*, Vol. 113, No. 5-6, 2003, pp. 761–781, Progress in statistical hydrodynamics (Santa Fe, NM, 2002).

[22]Gey, F. C. and Lesser, M. B., "Computer generation of series and rational function solutions to partial differential initial value problems," *Proceedings of the 1969 24th national conference*, ACM Press, 1969, pp. 559–572.

[23]Hall, A., "The ALTRAN System for Rational Function Manipulation - A Survey," *Communications of the ACM*, Vol. 14, No. 8, 1971, pp. 517–521.

[24]Hall, Jr., A. D., "Solving a problem in eigenvalue approximation with a symbolic algebra system," *SIAM J. Comput.*, Vol. 4, 1975, pp. 163–174.

[25]Rink, R. A. and Guru, B. P., "Analytical solutions for a class of nonlinear differential equations using ALTRAN," *Nordisk Tidskr. Informationsbehandling (BIT)*, Vol. 16, No. 2, 1976, pp. 161–171.

[26]Rink, R. A. and Guru, B. P., "Automating analytical solutions of nonlinear differential equations," *Computational methods in nonlinear mechanics (Proc. Internat. Conf., Austin, Tex., 1974)*, Texas Inst. Comput. Mech., Austin, Tex., 1974, pp. 57–66.