Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

On the use of Numerical Methods for Analysis and Control of Nonlinear Convective Systems

Edward Allen *, John A. Burns †, D. S. Gilliam ‡
* Mathematics and Statistics at Texas Tech University, Lubbock, TX edward.allen@ttu.edu
† Virginia Tech, Blacksburg, VA burns@math.vt.edu
‡ Mathematics and Statistics at Texas Tech University, Lubbock, TX david.gilliam@ttu.edu

Abstract— **A common approach to designing feedback controllers for nonlinear partial differential equations (PDEs) is to linearize the system about an equilibrium and use the linearized model as a starting point in the design process. In many practical applications (fluid flow control, thermal fluids, etc.) the equilibrium of interest is not the trivial zero state and this equilibrium must be computed numerically. This can be a complex task, especially if the equilibrium is unstable. In an earlier paper the authors showed that even for the 1D Burgers' equation, standard time marching numerical schemes can produce false steady state solutions. Consequently, in this case the first step in the design and analysis process is based on an erroneous linearization and even robust controllers may fail. It has been suggested that replacing the time marching scheme by a Newton algorithm would eliminate this false equilibrium. In this short paper we illustrate that even Newton's method can produce a numerically false equilibrium in problems that are highly sensitivity to boundary conditions. This result has important consequences when one uses numerical tools for design and control of non-linear partial differential equations that govern typical fluid flows.**

I. Introduction

In recent years it has become increasingly popular for technical papers in applied mathematics and engineering to contain simulations validating the authors work. Facilitated by the recent development of user friendly software, even with very limited knowledge of numerical methods, one can generate "approximate solutions" and impressive graphical images for extremely complex problems. For the vast majority of practical problems governed by partial differential equations (fluid dynamics, porous media, flexible structures, etc.) even the most experienced scientist is constrained to study properties of solutions obtained numerically. However, relying solely on numerical methods, not supported by a detailed theoretical study of the problem may lead to erroneous results. In [1] and [6] the authors encountered a remarkable anomaly associated with a standard hydrodynamic model – the one-dimensional viscous Burgers' equation. In [1] the authors considered a "time marching" numerical scheme for approximating steady state solutions of a particular boundary value problem for Burgers' equation. Several examples were given to illustrate that even theoretically convergent numerical schemes can produce numerical steady state solutions that do not correspond to steady state solutions of the boundary value problem. Burgers' equation has been used as a model problem for many recent studies in both optimal and feedback control (see, [2], [3], [4], [7], [8], [11], [12], [13], [14]). In 1993 Marrekchi, while working on a Neumann boundary control problem for Burgers' equation, observed that a finite element scheme used to design feedback controllers produced non-constant steady state solutions (see [17]). Moreover, recent work strongly suggests that a similar anomaly takes place for a broad class of nonlinear n-dimensional ($n \geq 1$) parabolic equations containing convective type terms. In particular, there is a strong numerical evidence that the same type of anomaly may occur in standard hydrodynamic models such as Euler and Navier Stokes equations (see [9], [10], [16]).

In this paper we consider the stationary problem associated with the dynamic problem considered in [1]. Using Newton's method we show that, due to finite precision arithmetic, this theoretically convergent numerical algorithm can produce false (purely numerical) steady state solutions.

The Model and Problem Formulation

Consider the the initial boundary value problem defined bu Burgers' equation in $L^2_{\text{AS}}(0,1)$

$$\begin{aligned} &z_t(x,t) = R^{-1} z_{xx}(x,t) - z(x,t) z_x(x,t),\ 0 \leq x \leq 1, \\ &z(x,0) = \varphi(x) \in L^2_{\text{AS}}(0,1) \\ &z_x(0,t) = 0, \\ &z_x(1,t) = 0. \end{aligned} \tag{I.1}$$

It is known (see [1]) that this equation with Neumann boundary conditions is invariant for Anti-Symmetric (AS) functions (i.e., odd about $x = 1/2$ in $(0,1)$). An AS function z satisfies $z(x) = -z(1-x)$ and we denote by $L^2_{\text{AS}}(0,1)$ the Hilbert subspace of $L^2(0,1)$ consisting of all AS functions.

It is also known that $z = 0$ is the unique asymptotically stable equilibrium (see the short discussion below) and in [15] it was shown that for every initial function φ the solution of (I.1) satisfies

$$\sup_{x\in[0,1]} |z(x,t)| \to 0 \quad t \to \infty.$$

Here we are interested in the numerical solution of the stationary problem for this dynamical system. The motivation for this analysis is that, even though zero is the only AS stationary solution, even convergent numerical time-marching algorithms for (I.1) can produce false (purely numerical) steady state solutions (see [1]). The corresponding stationary Burgers' system is given by

$$R^{-1}v_{xx}(x) - v(x)v_x(x) = 0,\ 0 \le x \le 1, \qquad \text{(I.2)}$$
$$v_x(0) = 0, \quad v_x(1) = 0,$$

in the space $L^2_{\text{AS}}(0,1)$.

Since our main interest is in the case of AS solutions on the interval $[0,1]$ and since all such functions must vanish at $x = 1/2$ we can study the steady state problem on the interval $[0,1/2]$ with Dirichlet conditions at the right end. Namely, we can consider the equivalent problem for Burger's equation on $0 \le x \le 1/2$ defined by

$$z_t(x,t) = R^{-1}z_{xx}(x,t) - z(x,t)z_x(x,t), \qquad \text{(I.3)}$$
$$z(x,0) = \varphi(x) \in L^2_{\text{AS}}(0,1)$$
$$z_x(0,t) = 0,$$
$$z(1/2,t) = 0.$$

Similarly, we can replace the steady problem (I.2) by the problem

$$R^{-1}v_{xx}(x) - v(x)v_x(x) = 0,\ 0 \le x \le 1/2, \qquad \text{(I.4)}$$
$$v_x(0) = 0, \quad v(1/2) = 0.$$

We now make one further simplification which makes all the formulas a bit simpler. In particular, we formulate the above problems as problems on the interval $[0,1]$. Hence we replace the problems (I.3) and (I.4) with corresponding problems defined on the standard interval $[0,1]$ defined by

$$z_t(x,t) = R^{-1}z_{xx}(x,t) - z(x,t)z_x(x,t),$$
$$z(x,0) = \varphi(x) \qquad \text{(I.5)}$$
$$z_x(0,t) = 0,$$
$$z(1,t) = 0,$$

and

$$R^{-1}v_{xx}(x) - v(x)v_x(x) = 0,\ 0 \le x \le 1, \qquad \text{(I.6)}$$
$$v_x(0) = 0, \quad v(1) = 0,$$

respectively.

It is simple exercise to show that the general solution to (I.6) is

$$v(x) = \sqrt{2c_0}\tanh\left(\frac{R\sqrt{2c_0}}{2}(c_1 - x)\right), \qquad \text{(I.7)}$$

where c_0 and c_1 are arbitrary constants. A straightforward calculation yields

$$v_x(x) = -Rc_0\,\text{sech}^2\left(\frac{R\sqrt{2c_0}}{2}(c_1 - x)\right), \qquad \text{(I.8)}$$

and these functions cannot vanish at $x = 0$ (unless $c_0 = 0$). Thus, as we have already mentioned (see [15]), the only stationary solutions to Burgers' equation satisfying homogeneous Neumann boundary condition at $x = 0$ and Dirichlet condition at $x = 1$ is the zero function.

In order that v satisfy the Dirichlet condition at $x = 1$ we need $c_1 = 1$. Thus, we consider functions $h(\cdot)$ defined by

$$h(x) = \sqrt{2c_0}\tanh\left(\frac{R\sqrt{2c_0}}{2}(1 - x)\right), \qquad \text{(I.9)}$$

which for large R and/or large c_0 gives

$$h'(0) = -Rc_0\,\text{sech}^2\left(\frac{R\sqrt{2c_0}}{2}\right) = -\alpha, \qquad \text{(I.10)}$$

where α is, for modest values of R and c_0, an exponentially small positive number. For α sufficiently small, on a computer with finite precision arithmetic, α would be set equal to zero. In effect, on a digital computer, the function $h(x)$ would appear to satisfy the steady Burgers' problem (I.4).

II. Newton's Method

We now turn to the numerical solution of the problem (I.4) using an iterative scheme based on Newton's method. To this end we consider the nonlinear map F from $H^2(0,1)$ to $L^2(0,1)$ given by

$$F(\varphi) = R^{-1}\varphi_{xx} - \varphi_x\varphi. \qquad \text{(II.1)}$$

Newton's method for solving $F(\varphi) = 0$ becomes

$$\varphi^k = \varphi^{k-1} - [F'(\varphi^{k-1})]^{-1}(F(\varphi^{k-1})), \qquad \text{(II.2)}$$

for $k = 1, 2, \cdots$ and $\varphi^0 = \varphi_0 \in H^2(0,1)$. Here we use the notation F' to denote the Gateaux derivative given by

$$F'(\varphi)(\eta) = \lim_{\epsilon\to 0}\frac{F(\varphi + \epsilon\eta) - F(\varphi)}{\epsilon} \qquad \text{(II.3)}$$
$$= R^{-1}\eta_{xx} - (\varphi\eta)_x.$$

We claim that depending on the choice of starting value φ_0 the iterates, which we have already argued must converge to zero, in fact converges to a nonzero function.

In order to compute the Newton iterate in (II.2) we rewrite the equation as

$$F'(\varphi^{k-1})(\varphi^k - \varphi^{k-1}) = -F(\varphi^{k-1}) \tag{II.4}$$

and appeal to (II.3) to obtain

$$\begin{aligned} R^{-1}(\varphi^k - \varphi^{k-1})_{xx} - (\varphi^k(\varphi^k - \varphi^{k-1}))_x \qquad \text{(II.5)} \\ = -(R^{-1}\varphi^{k-1}_{xx} - \varphi^{k-1}_x\varphi^{k-1}). \end{aligned}$$

After some simplification, we can write (II.5) (with boundary conditions) as

$$R^{-1}\varphi^k_{xx} - (\varphi^{k-1}\varphi^k)_x = -\varphi^{k-1}\varphi^{k-1}_x \tag{II.6}$$

$$\varphi^k_x(0) = 0, \quad \varphi^k(1) = 0. \tag{II.7}$$

In order to numerically compute the Newton iterate, one must introduce some type of numerical discretization for solving this two point boundary value problem. Here we consider two methods. The first numerical scheme is based on a shooting method and the second scheme is a finite difference approximation.

III. Direct Numerical Solution

In this section we consider the corresponding stationary Burgers' problem with a small non-homogeneous Neumann boundary condition at $x = 0$ and homogeneous Dirichlet condition at $x = 1$. Thus, we consider the problem

$$\left(\frac{1}{R}\right) v_{xx}(x) - v(x)v_x(x) = 0, \tag{III.1}$$

$$v_x(0) = -\alpha, \quad v(1) = 0, \quad \alpha > 0.$$

As noted above, the solution to the stationary Burgers' problem (I.6) is given by (I.9) with derivative at $x = 0$ given by (I.10).

For computational solutions of (III.1), we use a shooting method for the corresponding initial-value problem

$$R^{-1}v_{xx}(x) - v(x)v_x(x) = 0, \ 0 < x < 1 \tag{III.2}$$

$$v(1) = 0, \quad v'(1) = -\beta, \quad \beta > 0,$$

where $-\beta$ is to be computed so that $v_x(0) = v_x(0;\beta) = 0$. Specifically, $-\beta$ is initially guessed for the slope and (III.2) is solved from $x = 1$ to $x = 0$ using an initial-value solver such as a Runge-Kutta procedure. After solving (III.2) from $x = 1$ to $x = 0$, the slope $-\beta$ is adjusted so that $F(\beta) = v_x(0;\beta)$ decreases to zero.

Assuming that initial value problem (III.2) is solved exactly by the initial-value solver, then $v_x(x) = v_x(x,\beta)$ satisfies

$$v_x(x;\beta) = h'(x) = -Rc_0 \operatorname{sech}^2\left(\frac{R\sqrt{2c_0}}{2}(1-x)\right),$$

where c_0 is determined by the condition $h'(1) = -\beta$. But $h'(1) = -\beta$ implies that $c_0 = \beta/R$. Thus, (with $v(x;\beta) = h(x;\beta)$) we have

$$v(x;\beta) = \sqrt{2\beta/R}\tanh\left(\frac{R\sqrt{2\beta/R}}{2}(1-x)\right) \tag{III.3}$$

solves (III.2) with an initial guess of slope $-\beta$.

It follows that the shooting method procedure, with exact initial-value solver, reduces to finding β so that $F(\beta) = v_x(0;\beta) = -\beta\operatorname{sech}^2\left(\sqrt{R\beta/2}\right) = 0$. Then, the solution to (III.2) is equal to $v(x;\beta^*)$ where β^* is a value such that $F(\beta^*) = 0$. We now consider the problem of finding β so that

$$F(\beta) = -\beta\operatorname{sech}^2\left(\sqrt{R\beta/2}\right) = 0. \tag{III.4}$$

Clearly the only solution to this equation is $\beta = 0$.

So, let us proceed to solve the problem $F(\beta) = 0$ using Newton's method. Note that (III.4) implies

$$F'(\beta) = -\operatorname{sech}^2\left(\widetilde{\beta}\right)\left(1 + \left(\widetilde{\beta}\right)\tanh\left(\widetilde{\beta}\right)\right) \tag{III.5}$$

where we have set

$$\widetilde{\beta} = \sqrt{R\beta/2}, \tag{III.6}$$

and therefore Newton's method gives

$$\beta_{j+1} = \beta_j - F'(\beta_j)^{-1}F(\beta_j)$$

which, using (III.4) and (III.5) and doing some simplifying, can be written as

$$\beta_{j+1} = \beta_j - \frac{\beta_j}{\left(1 + \left(\widetilde{\beta}\right)\tanh\left(\widetilde{\beta}\right)\right)}.$$

After a bit more simplifying we obtain

$$\beta_{j+1} = \beta_j\,\frac{\gamma_j}{1+\gamma_j}, \quad \gamma_j = \left(\widetilde{\beta}\right)\tanh\left(\widetilde{\beta}\right). \tag{III.7}$$

Notice that the function

$$g(\gamma) = \frac{\gamma}{1+\gamma}$$

satisfies $0 \le g(\gamma) < 1$ and is only zero when $\gamma = 0$. Furthermore, g is a monotone decreasing function of γ. From this, we conclude that the iterates β_j are monotone decreasing with increasing j, as they should be. Indeed, for infinite precision arithmetic the numerical method produces a sequence of iterates that satisfies $\beta_j \to 0$ as $j \to \infty$.

However, on a finite precision digital computer this limit may not be achieved and it becomes important to understand what the algorithm actually produces.

Let ξ denote the base for a computer system and t the number of digits. On the interval $[\xi^{m-1}, \xi^m]$, the floating point numbers are evenly spaced with separation ξ^{m-t}. In addition, for prescribed exponent limit L for underflow, any number between $-\xi^{L-1}$ and ξ^{L-1} is set equal to zero in the computer. For example, in MATLAB, $\xi = 2$ and $L = -1074$. Indeed, in the IEEE Standard 754, for denormalized floating point numbers with base $\xi = 2$, $L = -149$ for single precision and $L = -1074$ for double precision.

With this in mind we consider starting values β_0 for the Newton iterations. Recall that to compute β_1 one must compute

$$F(\beta_0) = -\beta_0 \operatorname{sech}^2\left(\sqrt{R\beta_0/2}\right).$$

If we choose β_0 large enough so that

$$\beta_0 \operatorname{sech}^2\left(\sqrt{\frac{R\beta_0}{2}}\right) < \xi^{L-1},$$

then, for the arithmetic described above we have

$$F(\beta_0) = 0.$$

For example, suppose that $\xi = 2$, $L = -149$ and $R = 2.2210^4$. If β_0 is chosen greater than unity, then $F(\beta_0) < 2^{-150}$ and $F(\beta_0)$ would be set equal to zero on the computer. The Newton iterations would terminate with the computed solution $v(x;\beta_0)$ given by (III.3). This can be easily verified in MATLAB version 7 using the command *single*. In particular, with $\beta_0 = .999$ the command

$$\text{single}\left(-\beta_0 \operatorname{sech}^2\left(\sqrt{R\beta_0/2}\right)\right)$$

produces $-1.4013 \times 10^{-45} = -1.9637 \times 2^{-149}$ but with $\beta_0 = 1.0001$ the same command gives $F(\beta_0) = 0$.

IV. A Finite Difference Method

In this section we employ a finite difference method for approximating solutions of the steady state Burgers' boundary value problem

$$R^{-1}v_{xx}(x) - v(x)v_x(x) = 0, \qquad \text{(IV.1)}$$

$$v_x(0) = -\alpha, \quad v(1) = 0,$$

for $\alpha > 0$ or $\alpha = 0$. We note that for $\alpha > 0$ there are two non-zero solutions. One is a "small" stable equilibrium and the other equilibrium is given by (I.9) - (I.10) and is known to be unstable.

We use a second order approximation for both derivative terms. Let

$$x_j = (j-1)h, \quad h = \frac{1}{N}, \quad j = 1, 2, \cdots, (N+1),$$

and denote by φ_j the approximate values of $\varphi(x_j)$. The difference approximations are defined by

$$\varphi''(x_j) \approx \frac{\varphi_{j-1} - 2\varphi_j + \varphi_{j+1}}{h^2},$$

$$\varphi'(x_j) \approx \frac{\varphi_{j+1} - \varphi_{j-1}}{2h}.$$

The boundary conditions are approximated by

$$\varphi_{N+1} \approx \varphi(x_{N+1}) = \varphi(1) = 0$$

and

$$\frac{\varphi_2 - \varphi_0}{2h} \approx \varphi'(0) = -\alpha,$$

which implies that

$$\varphi_2 - \varphi_0 = -2\alpha h, \quad \text{and } \varphi_0 = \varphi_2 + 2\alpha h. \qquad \text{(IV.2)}$$

With this notation we can write the discretization of the boundary value problem (IV.1) as

$$\frac{\varphi_{j-1} - 2\varphi_j + \varphi_{j+1}}{h^2} - R\left(\frac{\varphi_{j+1} - \varphi_{j-1}}{2h}\right)\varphi_j = 0 \quad \text{(IV.3)}$$

which can be written as

$$\varphi_{j-1} - 2\varphi_j + \varphi_{j+1} - \frac{Rh}{2}\left(\varphi_{j+1} - \varphi_{j-1}\right)\varphi_j = 0. \quad \text{(IV.4)}$$

The special case of $j = 1$ for (IV.3) (using (IV.2)) gives

$$(\varphi_2 + 2h\alpha) - 2\varphi_1 + \varphi_2 - \frac{Rh}{2}(-2h\alpha)\varphi_1,$$

which simplifies to

$$\left(Rh^2\alpha - 2\right)\varphi_1 + 2\varphi_2 = -2h\alpha. \qquad \text{(IV.5)}$$

Also, when $j = N$ in (IV.3) it follows that

$$\varphi_{N-1} - 2\varphi_N + \varphi_{N+1} - R(\varphi_{N+1} - \varphi_{N-1})\varphi_N = 0$$

and since $\varphi_{N+1} = 0$ we have

$$\varphi_{N-1} - 2\varphi_N + R\varphi_{N-1}\varphi_N = 0.$$

Define the vectors Φ and Ψ in $\mathbb{R}^N$ (coefficient functions) by

$$\Phi = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N \end{bmatrix}, \quad \Psi = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_N \end{bmatrix}. \qquad \text{(IV.6)}$$

We also define the $N \times N$ matrix A with $A_{11} = (Rh^2\alpha - 2)$ by

$$A = \begin{bmatrix} A_{11} & 2 & 0 & \cdots & & 0 \\ 1 & -2 & 1 & \ddots & & \vdots \\ 0 & 1 & -2 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & & 1 \\ 0 & \cdots & \cdots & 0 & 1 & -2 \end{bmatrix}. \tag{IV.7}$$

Finally, define the $N \times N$ matrices M and P by

$$M = \frac{Rh}{2} \begin{bmatrix} 0 & 0 & 0 & \cdots & & 0 \\ 1 & & -1 & \ddots & & \vdots \\ 0 & 1 & & -1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & & -1 \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{bmatrix}, \tag{IV.8}$$

$$P = \begin{bmatrix} -2h\alpha \\ 0 \\ \\ \vdots \\ \\ 0. \end{bmatrix} \tag{IV.9}$$

The nonlinear function C is defined by

$$C(\Phi) = (M\Phi).*\Phi + P, \tag{IV.10}$$

where $.*$ denotes the Hadamard (component-wise) multiplication defined by

$$\Phi .* \Psi = \begin{bmatrix} \varphi_1\psi_1 \\ \varphi_2\psi_2 \\ \vdots \\ \varphi_N\psi_N \end{bmatrix}.$$

Finally, we define the nonlinear vector function F by

$$F(\Phi) = A\Phi + C(\Phi).$$

In this setting the problem (IV.1)becomes: Find Φ satisfying

$$F(\Phi) = 0.$$

To find zeros of F we apply a Newton iteration. Thus, we introduce the Newton iteration in $\mathbb{R}^N$

$$\Phi_k = \Phi_{k-1} - F'(\Phi_{k-1})(F(\Phi_{k-1})), \quad k = 1, 2, \cdots$$

where we choose the initial vector Φ_0.

A simple calculation yields

$$\begin{aligned} F'(\Phi)(\Psi) &= A\Psi + \mathrm{diag}(\Phi)M\Psi + \mathrm{diag}(M\Phi)\Psi \\ &\equiv \mathcal{M}_{k-1}\Psi, \end{aligned} \tag{IV.11}$$

where

$$\mathrm{diag}(\Phi) = \begin{bmatrix} \Phi_1 & 0 & \cdots & 0 \\ 0 & \Phi_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Phi_N \end{bmatrix}.$$

Therefore, the Newton iteration can be written as

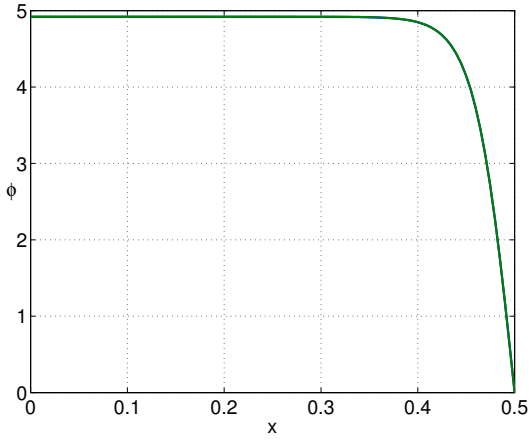
$$\Phi_k = \Phi_{k-1} - \mathcal{M}_{k-1}\big(A\Phi_{k-1} + C(\Phi_{k-1})\big). \tag{IV.12}$$

Numerical Examples and Conclusions

Example 1: Set $R = 10$, $\alpha = 10^{-8}$ and choose the initial function $\varphi(x) = M(1 - 2x)$ with $M = 5$. The corresponding Newton iterates defined in (IV.12) converge after 10 iterations with $N = 500$ to the numerical solution obtained in Section III. Namely, for this α we find $c_0 = 12.1065$ and the stationary solution is

$$h(x) = \sqrt{2c_0}\tanh\left(\frac{R\sqrt{2c_0}}{2}(1 - x)\right)$$

In the following figure we have plotted both $h(x)$ and the numerical solution. Observe that the two plots are identical and for $\alpha = 10^{-8} > 0$, Newton's method converges to $h(x)$ which is an unstable equilibrium in this case.



We note that for M small enough or negative the numerical solution converges to the zero function.

However, it is important to note that the parameter α in the Newton iteration only appears in two places in the numerical scheme. It appears in the $(1, 1)$ position in matrix A and in vector P in the nonlinear term. If we set these terms equal to zero, i.e., if we set $\alpha = 0$, then the numerical scheme still converges to a nonzero stationary function which is determined by the numerical precision of the particular

computer on which the program is run. This is again a false numerical steady state solution and, as seen above, it can be an order of magnitude away from the only true equilibrium.

Example 2: The choice of initial function certainly can influence the convergence of the algorithm. However, if $R = 10$, $\alpha = 10^{-8}$ and the initial function is selected to be $\varphi(x) = M\cos(\pi x)$ with $M = 1$, then the corresponding Newton iterates defined in (IV.12) again converge to exactly the same stationary solution as above.

Conclusions and Future Work

The examples above illustrate the need to conduct a careful analysis of the numerical methods used to compute equilibrium (steady state) solutions for nonlinear parabolic partial differential equations. Such systems are common in fluid flow control and heat transfer applications. These problems are typically highly sensitive to boundary conditions and even standard convergent numerical methods can produce false solutions. Also, we note that for problems of this type it is impossible to use numerical methods alone to make predictions about existence, uniqueness or stability of stationary solutions. This is especially true for complex Euler flows (see [16]). Finally, the results presented here merely point out an important issue that is often ignored in numerical design of controllers. Work needs to be done to provide a rigorous foundation for analyzing this approach for highly sensitive systems.

Acknowledgements The research of the second author was supported in part by Air Force Office of Scientific Research grant FA9550-07-1-0273 and the third author was supported in part by Air Force Office of Scientific Research grant FA9550-07-0214.

References

[1] E. Allen, J. A. Burns, D. Gilliam, J. Hill, V. I. Shubov, "The impact of finite precision arithmetic and sensitivity on the numerical solution of partial differential equations," *J.Math. Computer Modeling*, (2002).

[2] J. A. Atwell, "Proper Orthogonal Decomposition for Reduced Order Control of Partial Differential Equations," PhD Thesis, Department of Mathematics, VPI & SU, April 2000.

[3] J. A. Atwell and B. B. King, "Stabilized Finite Element Methods and Feedback Control for Burgers' Equation," *Proceedings of the 2000 American Control Conference*; Virginia Tech ICAM Report 00-02-01.

[4] T. R. Bewley, P. Moin, and R. Temam, "Control of Turbulent Flows," Systems Modelling and Optimization, (Detroit, MI, 1997), 3–11, Chapman & Hall CRC, Boca Raton, FL, (1999).

[5] J. T. Borggaard and J. A. Burns, "A PDE Sensitivity Equation Method for Optimal Aerodynamic Design" *Journal of Computational Physics*, Vol. 136, pages 366-384, (1997).

[6] J. A. Burns, A. Balogh, D. Gilliam and V. Shubov, "Numerical stationary solutions for a viscous Burgers' equation," *Journal of Mathematical Systems, Estimation and Control*, (1998).

[7] J. A. Burns and S. Kang, "A control problem for Burgers' equation with bounded input/output', Nonlinear Dynamics", (1991), Vol. 2, 235-262.

[8] J. A. Burns and S. Kang, "A Stabilization problem for Burgers' equation with unbounded control and observation", International Series of Numerical Mathematics, (1991), Vol. 100, 51-72.

[9] J. A. Burns and John Singler, "Modeling Transition: New Scenarios, System Sensitivity and Feedback Control", Transition and Turbulence Control, M. Gad-el-Hak and H. M. Tsai, Eds., World Scientific Publishing, 2005, 1 - 37.

[10] J. A. Burns and John Singler, "Feedback Control of Low Dimensional Models of Transition to Turbulence", 44th IEEE Conference on Decision and Control, Seville, Spain, December 2005, pp. 3140 - 3145.

[11] C. I. Byrnes, D. S. Gilliam and V. I. Shubov, "On the Global Dynamics of a Controlled Viscous Burgers' Equation," *Journal of Dynamical and Control Systems*, Volume 4, 1998, no. 4, 457–519. .

[12] C. I. Byrnes, D. S. Gilliam and V. I. Shubov. "Boundary control, feedback stabilization and the existence of attractors for a viscous Burgers' Equation," (1994), Preprint.

[13] C. I. Byrnes, D. Gilliam, V. Shubov and Z. Xu, "Steady state response to Burgers' equation with varying viscosity," *Progress in Systems and Control: Computation and Control IV*, Eds. K.L.Bowers and J. Lund, (1995), pp.75-98.

[14] C. I. Byrnes, D. S. Gilliam and V. I. Shubov, "Boundary Control, Stabilization and Zero – Pole Dynamics for a Nonlinear Distributed Parameter", International Journal of Robust and Nonlinear Control, Vol. 9, 737-768 (1999).

[15] Chongsheng Cao and Edriss Titi, "Asymptotic Behavior of Viscous Burgers' Equations with Neumann Boundary Conditions," *Third Palestinian Mathematics Conference, Bethlehem University, West Bank, Mathematics & Mathematics Education*,. Elaydi, E. S. Titi, M. Saleh, S. K. Jain and R. Abu Saris, editors, World Scientifc, 2002.

[16] A. Jameson, "Airfoils Admitting Non-unique Solutions of the Euler Equations," *AIAA 22nd Fluid Dynamics, Plasmadynamics & Lasers Conference, Honolulu*, June, (1991).

[17] H. Marrekchi, "Dynamic Compensators for a Nonlinear Conservation Law", Ph.D. Thesis, Department of Mathematics, Virginia Polytechnic Institute and State University, September, 1993

[18] S. M. Pugh, "Finite element approximations of Burgers' Equation," *Masters of Science Dissertation, VPI & SU*, September , (1995).