Control of Convergence in a Computational Fluid Dynamics Simulation Using ANFIS

Juntaek Ryoo, Zoran Dragojlovic, and Deborah A. Kaminski

***Abstract*—Adaptive-network-based fuzzy inference system (ANFIS) was used to control convergence of a computational fluid dynamics (CFD) algorithm. Normalized residuals were used to select the relaxation factors on each iteration of the computation. The ratio between the residual of the current iteration and of the previous iteration was used as a control input. ANFIS was designed to keep the tuning index at or near one throughout the fluid dynamics simulation. The finite volume CFD algorithm SIMPLER was used as the platform for this study. Four different benchmark cases were examined.**

***Index Terms*—Adaptive-network-based fuzzy inference system (ANFIS), computational fluid dynamics (CFD), computational fluid mechanics, convergence, relaxation factors.**

I. Introduction

IN HEAT transfer and fluid dynamic simulations, stabilizing the computation and achieving convergence can be very difficult, especially in highly nonlinear cases. In an attempt to promote convergence, user-specified parameters such as relaxation factor have often been used. However, specifying these values can be difficult. The user chooses them from experience and adjusts them when the simulation diverges or fails to converge in a reasonable amount of time. In this work, adaptive-network-based fuzzy inference system (ANFIS) [1] was applied to control the relaxation factors of a common computational fluid dynamics (CFD) algorithm. Relaxation factors were adjusted automatically during code execution based on the behavior of residuals.

Little work has been reported in using soft computing methods such as fuzzy logic or neural networks to aid CFD simulations. Cort *et al.* [2] performed a one-dimensional finite element heat transfer simulation with radiative boundary conditions. They used a simple feedback control method to adjust the relaxation factors. Iida *et al.* [3] published a study in which wobbling adaptive control was applied to a CFD simulation of the Benard problem.

More recently, Ryoo *et al.* [4] developed a fuzzy algorithm to control an under-relaxed CFD computation. The algorithm used the behavior of the dependent variables and their iterative errors and inputs. However, the values of the dependent variables are not necessarily the best indicators of convergence. The algorithm was later improved by studying the behavior of residuals [5]. Recently, Dragojlovic *et al.* [6] developed a fuzzy algorithm based on a Fourier analysis of the iteration history of a CFD computation. Studies to improve the convergence of genetic algorithms using fuzzy control have been reported in the literature [7]–[9]. In these studies, genetic parameters such as crossover rate and mutation rate were adaptively regulated during the genetic search.

The well-known SIMPLER algorithm [10] was used in the present work. Four benchmark problems were tested for various conditions. ANFIS was developed to adjust relaxation factors dynamically during code execution. In contrast to the work previously done by the authors, which used rule-based fuzzy logic, ANFIS was designed to make a control surface based on training patterns. The results were compared to uncontrolled simulations and rule-based fuzzy logic control simulations [5].

II. Analysis

In all cases, the flow was assumed to be Newtonian, incompressible, laminar, two-dimensional and steady. Viscous dissipation was neglected. All thermophysical properties were assumed constant, except for the density, which was treated using the Boussinesq approximation. The buoyancy force is in the y-direction. The conservation equations for mass, momentum, and energy are given in Patankar [10] as

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0$$

$$\rho\left(u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right)=-\frac{\partial P^*}{\partial x}+\mu\nabla^2 u$$

$$\rho\left(u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}\right)=-\frac{\partial P^*}{\partial y}+g\beta\rho(T-T_r)+\mu\nabla^2 v$$

$$\rho c_p\left(u\frac{\partial}{\partial x}(T-T_r)+v\frac{\partial}{\partial y}(T-T_r)\right)=k\nabla^2(T-T_r)$$

where P^* is an effective pressure given by

$$P^*=P+g\rho_c y.$$

The conservation equation were discretized by a finite volume approach as

$$a_p\phi_p=\sum a_{nb}\phi_{nb}+b$$

where a_p is a coefficient for the point P under consideration, a_{nb}'s are the coefficients of neighboring grid points, ϕ_p is the

Manuscript received May 3, 2001; revised August 14, 2003 and April 19, 2004.

J. Ryoo is with the Institute of Automotive Technology, Yonsei University, Seoul, Korea.

Z. Dragojlovic and D. A. Kaminski are with the Department of Mechanical, Aerospace and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: kamind@rpi.edu).

Digital Object Identifier 10.1109/TFUZZ.2004.839656

value of the dependent variable for the equation under consideration, ϕ_{nb}'s are the values of the neighboring grid points and b is the source term. The generic variable ϕ is used to represent u, v, and T.

On each iteration, the assumed values of the solution vector were updated with under-relaxed values according to

$$\phi_n^* = \phi_{n-1} + \alpha(\phi_n + \phi_{n-1})$$

where α is a relaxation factor which varies between 0 and 1. The relaxation factor can be varied from iteration to iteration and from point to point in the solution domain without affecting the final converged solution. The goal of the current work is to design a controller that will adjust values of α to enable or accelerate convergence.

III. Adaptation of ANFIS to CFD Simulation Control

Most control systems are designed to reach target values of the control variables. However, in this computational simulation, iterative error cannot be chosen as a control input because iterative error can be artificially reduced by reducing the relaxation factor without actually solving the system of equations of interest. Instead, the tuning index and its error, as described later, were used as inputs.The discretized equation for the residual is given by

$$r = \sum a_{nb}\phi_{nb} + b - a_p\phi_p.$$

The residual was divided by the value of the coefficient a_p in the discretized equation to obtain the displacement, d, which is defined as

$$d = \frac{\sum a_{nb}\phi_{nb} + b - a_p\phi_p}{a_p}.$$

The displacement d can be considered as a normalized residual error of the discretized equation. The 2-norm of the displacement was used as a representative value for the spatial domain

$$||d|| = \sqrt{\sum d^2}.$$

In the current work, a tuning index similar to one used by Bare [11] was defined as

$$\gamma_n = \frac{||d||_n}{||d||_{n-1}}$$

where $||d||_n$ and $||d||_{n-1}$ are 2-norms of displacements on two consecutive iterations. The tuning index was used to indicate whether the convergence is linear and to moderate the rate of convergence. When the tuning index is near one, the computation can be considered behaving linear. Therefore, maintaining the tuning index near one can be set as a control goal and the error in the simulation and the change in error were defined, respectively, as

$$e_n = 1 - \gamma_n$$
$$\Delta e_n = e_n - e_{n-1}.$$

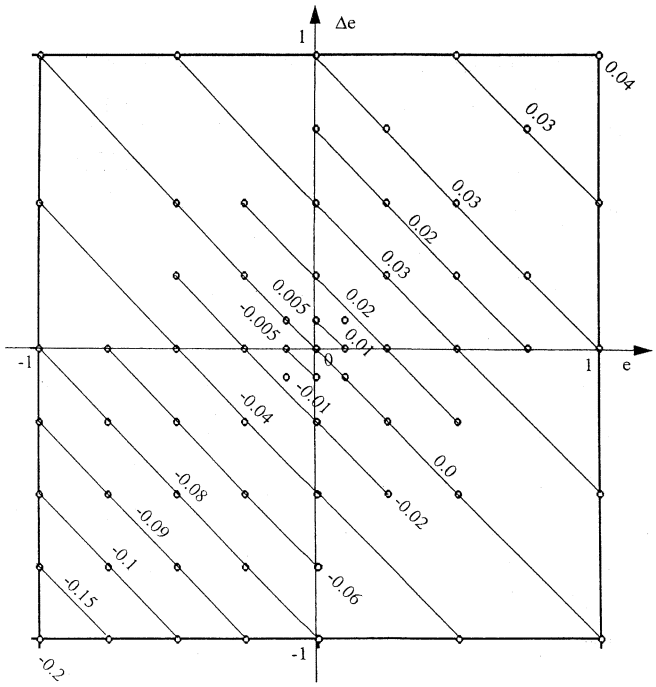

Fig. 1. Change in relaxation factor $\Delta\alpha$, as a factor of error e, and change in error Δe.

Note that the error cannot be bigger than 1. Furthermore, any changes in error bigger than 1 or smaller than -1 were treated as 1 and -1, respectively. The control was focused on driving both e_n and Δe_n to zero.

ANFIS [1] was applied to control the relaxation factors. The network used in this study has two inputs and one output, and three layer-1 membership functions for each input were used. The inputs were the errors and changes in errors, and the output was the change in relaxation factor, $\Delta\alpha$. The change in relaxation factor, $\Delta\alpha$, was used to update relaxation factor as follows:

$$\alpha_{n+1} = (1 + \Delta\alpha)\alpha_n.$$

The relaxation factors was restricted to take on values between zero and one. A set of input–output patterns was carefully design as given in Fig. 1. The dots in this figure represent the actual data points used for the training and the lines connect dots that have the same change in relaxation factor. When the error and the change in error are both positive, the tuning index is smaller than one (converging slower than linear) and also smaller than the previous tuning index (decreasing in convergence rate). Therefore, the relaxation was increased in order to speed up the convergence. On the other hand, when the error and the change in error are both negative, the tuning index is bigger than one (converging faster than linear) and also bigger than the previous tuning index (increasing convergence rate aggressively). In such a case, the relaxation factor was decreased in order to stablize the computation. In cases that the error and the change in error have different signs, the adjustment of the relaxation factor was kept relatively small. In highly nonlinear cases, a constant relaxation factor can promote round-off computational error even though the error and the change in error approach zero. In order

to prevent such a situation, the input–output pattern was design to induce small fluctuations in the value of relaxation factor.

IV. Application to Benchmark Problems

Four benchmark problems were examined to test the utility of the controller. Fluid properties were assumed constant and the nonslip condition was applied to all surfaces. The convergence criterion was given by

$$\frac{\max|\phi_n - \phi_{n-1}|}{\max|\phi_n| \cdot \alpha_n} \leq 10^{-5}.$$

Here, the maximum error in the domain was divided by the maximum function value in the domain. The maximum value was multiplied by the relaxation factor to avoid the illusion of convergence from heavy relaxation factors.

The controller was used to relax the two components of velocity and the temperature field. The relaxation factor for each variable was independently controlled. The pressure and pressure correction were not under-relaxed, since little benefit resulted for the cases examined. By default, each relaxation factor was initialized at 1. Then, the controller adjusted the relaxation factor as needed to achieve convergence.

A. Buoyancy-Driven Square Cavity Case

The geometry for a two-dimensional steady state buoyancy driven flow in a square cavity is presented in Fig. 2. The cavity is closed and filled with a constant property fluid. The left and right sides are isothermal at two different temperatures and the horizontal sides are insulated. Because of heat transfer from the hot wall, a buoyancy-driven recirculation pattern appears in the cavity. The problem is nondimensionalized and characterized by a Rayleigh number, $\mathrm{Ra} = \rho^2 g \beta \Delta T L^3 \mathrm{Pr}/\mu^2$. From $\mathrm{Ra} = 10^2$ to $\mathrm{Ra} = 10^6$, five cases were simulated. A practical flow will probably be turbulent after $\mathrm{Ra} = 10^6$.

B. Lid-Driven Square Cavity Case

As a second benchmark problem, a cavity with a lid which moves horizontally with constant velocity U as shown in Fig. 3 was simulated. The energy equation was not solved since the flow is assumed isothermal. The problem was nondimensionalized and characterized by a Reynolds number, Re, which is based on the lid speed. Two cases were tested at $\mathrm{Re} = 10^2$ and $\mathrm{Re} = 10^3$.

C. Backward-Facing Step

Fig. 4 shows a two-dimensional flow in a channel with a sudden expansion. The inlet flow temperature is the same as the temperature of the left straight wall. The step in the right part of the channel is insulated and the straight part of the right wall is at a higher temperature than the inlet. The inlet flow enters the domain in parabolic profile [12]. The Reynolds number Re was set at 100 and the Peclet number Pe was set at 70. The problem was nondimensionalized and simulated as a mixed convection flow as well as a forced convection flow with Grashof number $\mathrm{Gr} = 0$ and $\mathrm{Gr} = 1000$.

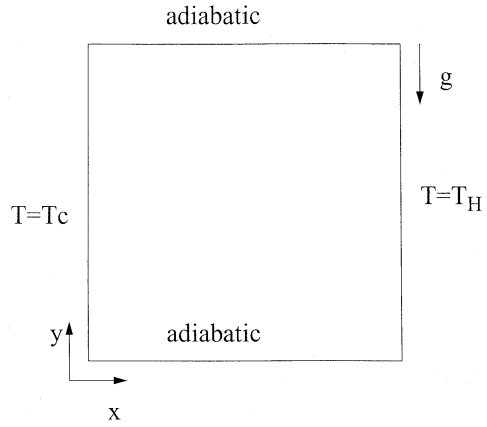

Fig. 2. Buoyancy-driven cavity.

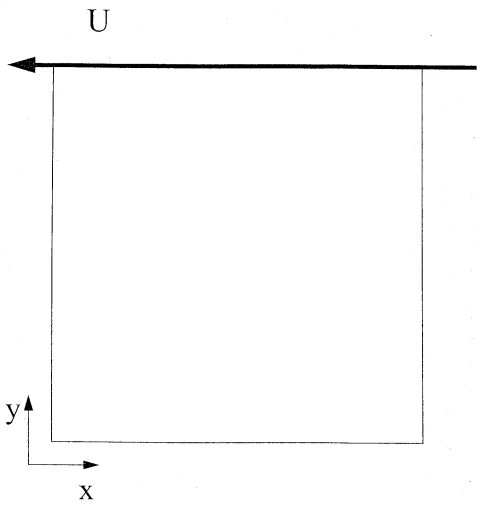

Fig. 3. Lid-driven square cavity.

D. Conjugate Buoyancy-Driven Cavity

The geometry for a buoyancy driven flow in a square cavity with a solid wall of finite thickness at one side is presented in Fig. 5. The cavity is closed and filled with a constant property fluid. The left side of the cavity is isothermal at a cold temperature and the right side exchanges heat with the wall. The right side of the wall is isothermal at a hot temperature. The horizontal sides are insulated. The problem is nondimensionalized and characterized by a Rayleigh number and conduction-convection ratio defined as Lk_w/Dk_f. Three cases were simulated: $Lk_w/Dk_f = 5$, $Lk_w/Dk_f = 25$, and $Lk_w/Dk_f = 50$ all at $\mathrm{Re} = 10^7$. For this highly nonlinear case, the code was restarted if divergence occurred because of a high initial-relaxation factor. If the computation diverged or reached the machine limit, the computation was restarted from the initial guess, here, a zero velocity field, with the low relaxation factor which had been calculated up to that point.

V. Result and Discussion

The solution from the controlled computation and the fixed relaxation factor computation were compared with published

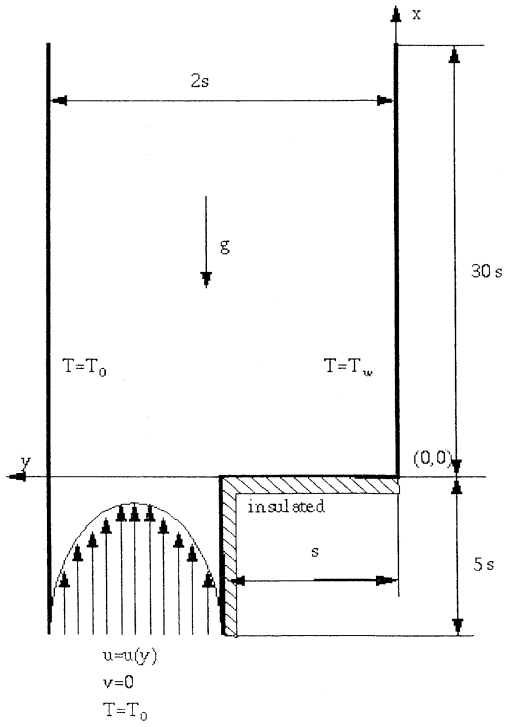

Fig. 4. Backward facing step channel.

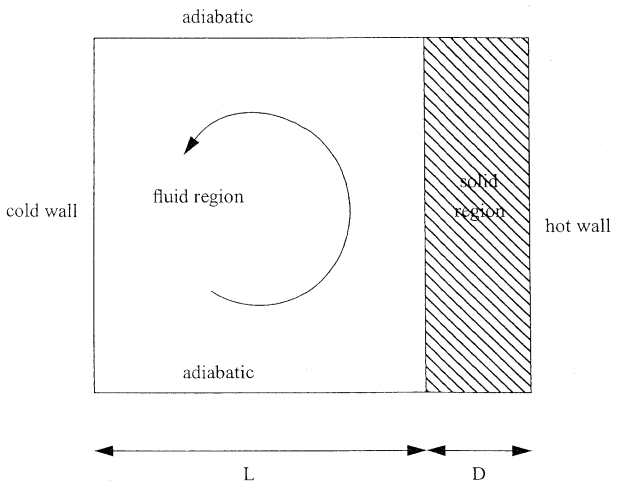

Fig. 5. Conjugate buoyancy-driven cavity.

result [12]–[15]. The result form the controlled and the uncontrolled cases are identical to at least three significant figures and match the published results well.

ANFIS achieved convergence for all cases tested. Numbers of iterations required for convergence using various constant relaxation factors are compared to those required using fuzzy control [5] and ANFIS in Table I. In general, the number of iterations using a controller was lower than that for most of the constant relaxation factor cases. In most cases, ANFIS outperformed the rule-based fuzzy controller. A plot of the iteration required for convergence versus the relaxation factor for a Rayleigh number of 10^5 is shown in Fig. 6.

Table II shows the CPU time required to achieve convergence for the buoyancy-driven cavity case with Ra$=$ 100. For constant relaxation factor cases, the CPU time was proportional to

TABLE I
ITERATIONS REQUIRED TO CONVERGE FOR VARIOUS FIXED RELAXATION FACTORS AND CONTROLLED CASES

Cases	Constant Relaxation				Rule-based fuzzycontrol	ANFIS
	0.2	0.4	0.6	0.8		
Buoyancy driven Cavity						
$Ra = 10^2$	3024	1301	683	328	252	150
$Ra = 10^3$	5704	2278	1101	477	289	158
$Ra = 10^4$	4009	1583	754	322	136	119
$Ra = 10^5$	3407	1323	627	275	141	164
$Ra = 10^6$	2482	1034	521	238	270	194
Lid driven Cavity						
$Re = 10^2$	2346	921	437	187	143	139
$Re = 10^3$	1641	640	300	120	285	96
Backward Facing Step Channel						
$Gr = 0$	4123	1598	743	304	292	216
$Gr = 10^3$	3158	1235	582	244	253	292
Conjugate Buoyancy driven Cavity						
$Lk_w/Dk_f = 5$	Osc	Div	912	Osc	431	264
$Lk_w/Dk_f = 25$	4290	1681	791	Osc	314	282
$Lk_w/Dk_f = 50$	Div	1650	753	Osc	345	321

Div: Divergence, Osc: Oscillation

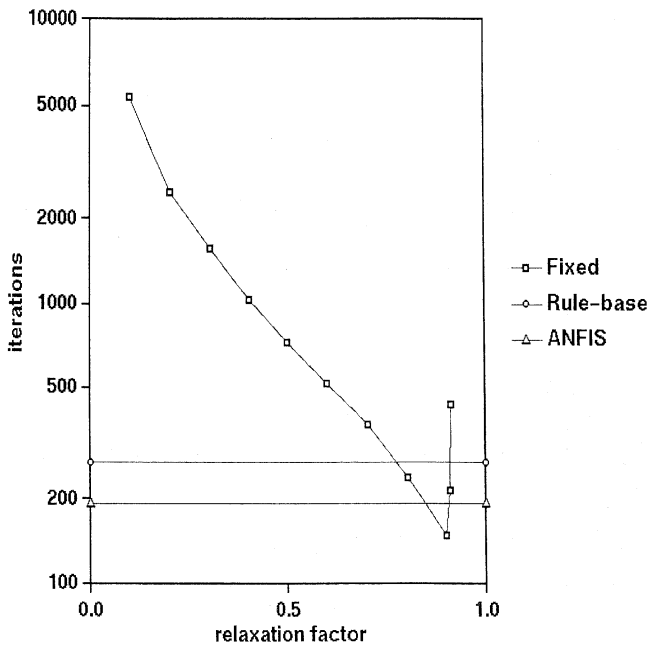

Fig. 6. Iterations required to converge in buoyancy-driven cavity Ra$= 10^6$.

the iteration required for convergence and the rule-based fuzzy control and ANFIS needed additional computational load for its own calculation. However, note that, in larger CFD simulation cases, the computational load for control algorithm will be relatively small compared to the computational load for CFD simulation.

TABLE II
CPU TIME FOR BUOYANCY-DRIVEN CAVITY WITH Ra= 100

Constant Relaxation				Rule-based fuzzy logic	ANFIS
0.2	0.4	0.6	0.8		
29.85	12.19	6.03	3.37	6.58	3.99

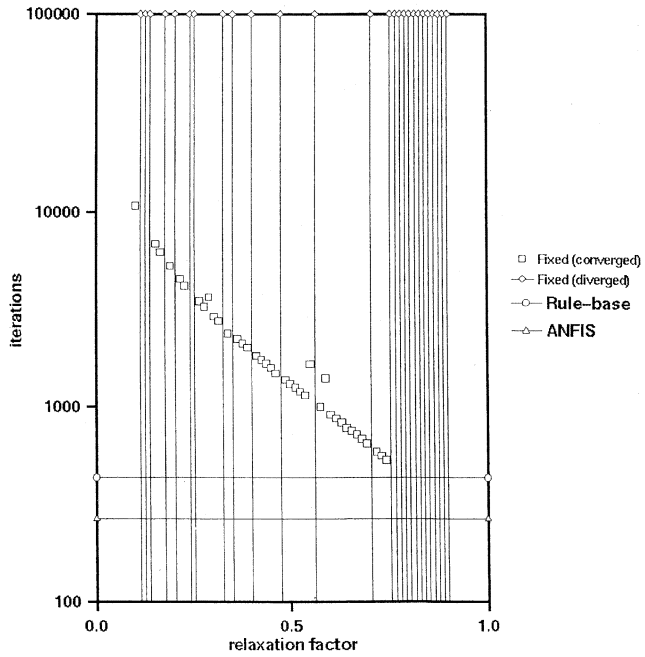

Fig. 7. Iteration required to converge in conjugate cavity $Lk_w/Dk_f = 5$. Vertical lines represent cases that diverged and horizontal lines represent controlled cases.

The difference in performance between the fuzzy controller and ANFIS results from the fact that neither are optimized for the problems they are solving. It is unproductive to completely optimize a controller for problems of this type. We want a controller that can solve new problems (e.g., new geometries and boundary conditions) as well as the limited set it was tested on. A controller which is optimized to solve one particular case or set of cases of fluid flow may not be optimal for a new case but it might give an acceptable solution for that case. The training set for the ANFIS algorithm described in Fig. 1 was sufficient to solve four very different problems in laminar flow. Thus, this training set has some general applicability. However, for problems with significantly different physical effects, such as turbulent flow, a different training set may be needed.

The conjugate buoyancy driven cavity gives interesting cases, one of which is shown in Fig. 7. In these cases, it is difficult to predict whether a given relaxation factor will produce convergence. There are no well-defined zones of convergence. If a case diverges, a slight increase or decrease in relaxation factor can produce convergence. This is in contrast to the generally accepted idea: that the lower the relaxation factor, the slower the convergence and the higher the relaxation factor, the greater the risk of divergence. Therefore, in the conjugated buoyancy driven cases, even slow convergence is not guaranteed by simply lowering relaxation factors, which makes the choice of relaxation factor extremely difficult.

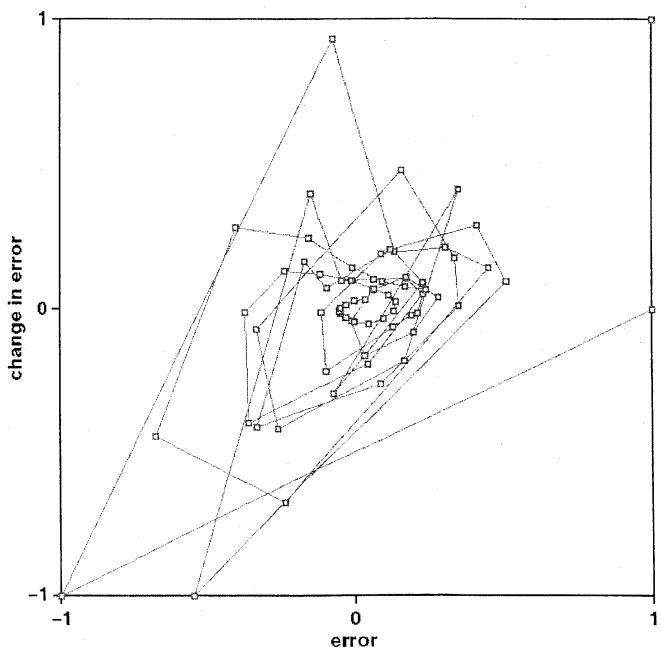

Fig. 8. Input from ANFIS in early stages of computation of buoyancy-driven cavity Ra= 10^6.

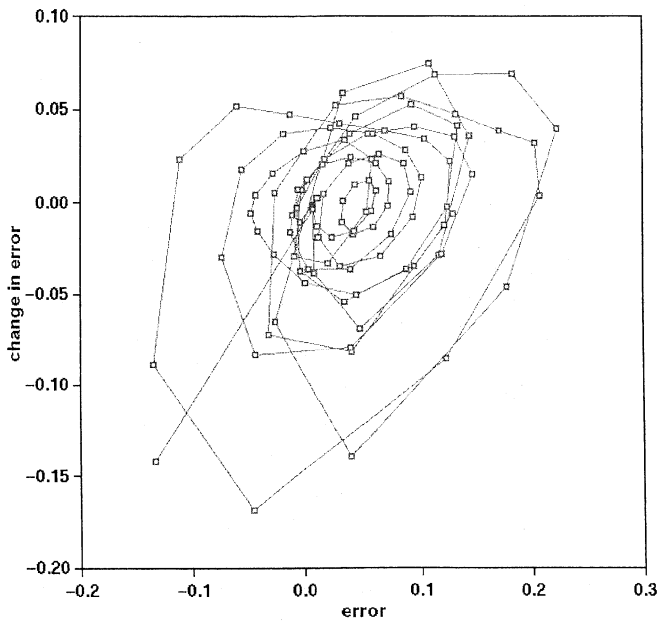

Fig. 9. Input from ANFIS in late stages of computation of buoyancy-driven cavity Ra= 10^6.

It is worth while to investigate the trajectory of the error and the change in error and their corresponding output. Two trajectories are presented in Figs. 8 and 9, for the early computational stage and the late computational stage, respectively, in the buoyancy driven Ra= 10^6 case. In Fig. 10, the changes in relaxation factor throughout the iteration are presented. Patterns for ANFIS training can be designed by studying these figures to achieve a desirable control surface.

Although it is possible for the user to select a constant relaxation factor that will outperform the controlled algorithm, it is generally not easy to do so especially in highly nonlinear cases.

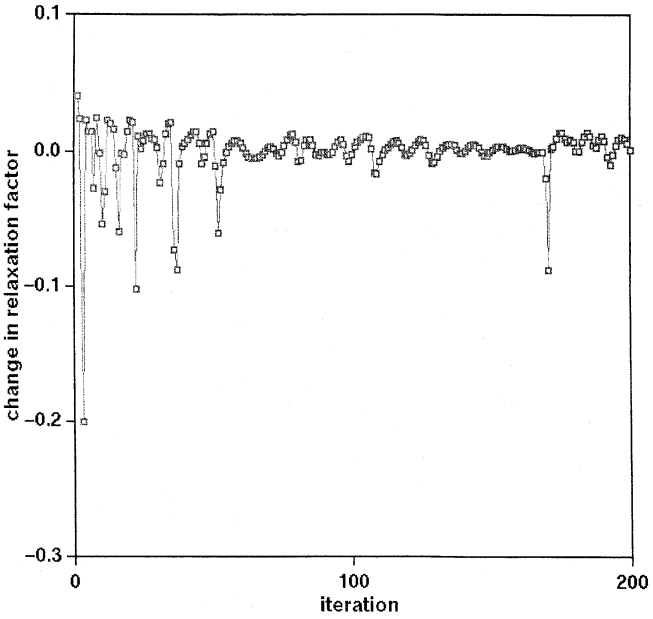

Fig. 10. Change of relaxation factors of buoyancy-driven cavity Ra= 10^6.

Furthermore, the danger of divergence is always present if relaxation factors are set too high, and the user may need to manually restart the code more than once with new lower relaxation factors. This work can be extended to other systems of nonlinear equations used in engineering and science.

REFERENCES

[1] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, Jun. 1993.

[2] G. E. Cort, A. L. Graham, and N. L. Johnson, "Comparison of methods for solving nonlinear finite-element equations in heat transfer," in *Proc. Amer. Soc. Mechnical Engineers*, 1982, 82-HT-40 ASME.

[3] S. Iida, K. Ogawara, S. Furusawa, and N. Ohata, "A fast converging method using wobbling adaptive control of SOR relaxation factor for 2D Benard convection," *J. Mech. Eng. Soc. Jpn.*, vol. 7, pp. 168–174, 1994.

[4] J. Ryoo, D. Kaminski, and Z. Dragojlovic, "Automatic convergence in a computatinal fluid dynamics algorithm using fuzzy logic," presented at the 6th Annu. Conf. Computational Fluid Dynamics Society of Canada, VIII, Quebec City, Canada, 1998.

[5] ——, "A residual-based fuzzy logic algorithm for control of convergence in a computational fluid dynamics simulation," *J. Heat Transfer*, vol. 121, pp. 1076–1078, 1999.

[6] Z. Dragojlovic, D. A. Kaminski, and J. Ryoo. Control of convergence in convective flow simulation using a fuzzy rule set that stabilizes iterative oscillations. presented at *Proc. Amer. Soc. Mechanical Engineers, 33rd National Heat Transfer Conf.*. [CD-ROM]NHTC99-229

[7] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 4, pp. 656–667, Aug. 1994.

[8] P. T. Wang, G. S. Wang, and Z. G. Hu, "Speeding up the search process of genetic algorithm by fuzzy logic," in *Proc. 5th Eur. Congr. Intelligent Techniques and Soft Computing*, 1997, pp. 665–671.

[9] R. Subbu, A. C. Sanderson, and P. P. Bonissone, "Fuzzy logic controlled genetic algorithm versus tuned genetic algorithm: An agile manufacturing application," in *Proc. 1999 IEEE Int. Symp. Intelligent Control (ISIC)*, 1998, pp. 434–440.

[10] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*. New York: McGraw-Hill, 1980.

[11] W. H. Bare, R. J. Mullholland, and S. S. Sofer, "Design of a self-tuning rule based controller for a gasoline refinery cataytic reformer," *IEEE Trans. Autom. Control*, vol. 35, no. 2, pp. 156–164, Feb. 1990.

[12] B. F. Blackwell and B. F. Armaly, "Benchmark problem definition and summary of computational results for mixed convection over a backward facing step," in *Proc. ASME*, 1993, HTD 258.

[13] J. W. Elder, "Laminar free convection in a vertical slot," *J. Fluid Mech.*, vol. 23, pp. 77–98, 1965.

[14] R. Schreiber and H. B. Keller, "Driven cavity flows by efficient numerical techniques," *J. Comput. Phys.*, vol. 49, pp. 310–333, 1983.

[15] D. A. Kaminski, "Analysis of natural convection flow in a square enclosure with complete account of conduction in one of the vertical walls," Ph.D. dissertation, Rensselaer Polytechnic Inst., Troy, NY.

Juntaek Ryoo received the B.S. degree in mechanical engineering from Yonsei University, Seoul, Korea, in 1995, and the M.S. and Ph.D. degrees from Rensselaer Polytechnic Institute, Troy, NY, in 1999 and 2002, respectively. His M.S. thesis focused on computational fluid dynamics and his Ph.D. work focused on multidisciplinary design optimization.

During a one-year postdoctoral appointment, he studied the modeling of carbon nano tubes using soft computing. Currently, he is a Researcher at the Institute of Automotive Technology, Yonsei University.

Zoran Dragojlovic received the Ph.D. degree in mechanical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2000.

Between January 2000 and November 2001, he worked on low-pressure chemical vapor deposition tools at Applied Materials, Inc., San Diego, CA, where he specialized in developing thin films for semiconductor devices. He joined the University of California, San Diego in November 2001, where he is currently pursuing his research career in the Department of Electrical and Computer Engineering. His research and academic interests include efficient algorithms in fluid mechanics and heat transfer. He is currently developing an algorithm for compressible Navier–Stokes equations with adaptive grid refinement, tailored for simulating the physics of Inertial Confinement Fusion chamber.

Deborah A. Kaminski is an Associate Professor of Mechanical Engineering at Rensselaer Polytechnic Institute. received the B.S. degree in physics, the M.S. degree in chemical engineering, and the Ph.D. degree in mechanical engineering, all from Rensselaer Polytechnic Institute, Troy, NY, in 1973, 1976, and 1985, respectively.

Her areas of research include the use of fuzzy logic, neural nets, and genetic algorithms applied to computational fluid mechanics and radiative heat transfer. She is the author of more than 60 technical publications and has served as an Associate Technical Editor of the *Journal of Heat Transfer*.