PERGAMON

International Journal of Heat and Mass Transfer 44 (2001) 3811–3822

International Journal of HEAT and MASS TRANSFER

www.elsevier.com/locate/ijhmt

Tuning of a fuzzy rule set for controlling convergence of a CFD solver in turbulent flow

Zoran Dragojlovic, Deborah A. Kaminski *, Juntaek Ryoo

Department of Mechanical Engineering, Aeronautical Engineering, and Mechanics, Rensselaer Polytechnic Institute, JEC 3018, Troy, NY 12180-3590, USA

Received 6 June 2000; received in revised form 7 January 2001

Abstract

Under-relaxation in an iterative CFD solver is guided by fuzzy logic to achieve automatic convergence with minimum CPU time. The fuzzy rule set uses information from a Fourier transform of a set of characteristic values. The control algorithm adjusts the relaxation factors for the system variables on each iteration and restarts the solver if divergence occurs. Two turbulent problems based on a k–ε model are solved. They include buoyancy driven flow in a rectangular cavity and mixed convection over a backward facing step. The incompressible Newtonian conservation equations are solved by the SIMPLER algorithm with simple substitution. In order to achieve the best performance of the fuzzy controller, the membership functions were tuned by using a gradient method. The fuzzy control algorithm with the optimal membership functions significantly reduced the CPU time needed for solving the problem, compared to the highest set of constant relaxation factors which do not cause divergence. For turbulent flow over a backward facing step, the CPU time was more than five times shorter with the fuzzy controller than with the constant relaxation factors. For turbulent buoyancy driven flow in a rectangular cavity, the CPU time required for convergence with a fuzzy controller was reduced by a factor of two, compared to the constant relaxation factors.

1. Introduction

One of the main challenges of computational fluid dynamics is having to deal with highly nonlinear differential equations. In order to solve continuity, momentum, energy and other scalar transport equations, one has to rely on iterative methods. Some of the most commonly used methods to solve discretized equations iteratively are Newton Raphson, Quasi-Newton and simple substitution. The SIMPLER algorithm, [1], which is a platform for our study, uses simple substitution in order to solve discretized governing equations of fluid motion, energy and scalar transport. However, the success of this iterative method in most CFD problems relies on under-relaxation of state variables. The value of the variable to be used for obtaining the solution in the next iteration is the value in the current iteration plus a fraction of the difference between the current value and the predicted value.

The under-relaxation method enables and improves convergence by slowing down the update rate of the system matrix coefficients. The value of the variable φ_p to be substituted into the system of discretized governing equations is obtained as a linear interpolation between the value from the previous iteration and the one obtained in the current iteration, according to the following scheme:

$$\varphi_p = \varphi_p^* + \alpha\left(\frac{\sum a_{nb}\varphi_{nb} + b}{a_p} - \varphi_p^*\right) \tag{1}$$

where $0 < \alpha < 1$ is the relaxation factor, φ_p is the value of the state variable at node P to be used for the next iteration, φ_p^* is the value of the state variable at node P in the previous iteration, φ_{nb} are the values of the variables at the surrounding nodes and a_p, a_{nb} and b are the constants from the discretized equation.

* Corresponding author. Tel.: 1-518-276-6626; fax: 1-518-276-4860.

E-mail address: kamind@rpi.edu (D.A. Kaminski).

0017-9310/01/$ - see front matter
PII: S0017-9310(01)00029-1

In a conventional method, the relaxation factor is selected manually by the user. It can be prescribed as a constant during the iterations or it can be altered from iteration to iteration, node to node or variable to variable. The values of relaxation factors have a strong influence on convergence of the system of discretized equations being solved, especially in turbulent flow. Low values typically insure convergence but result in a large number of iterations to obtain the solution. High values of the relaxation factor, on the other hand, tend to lead to divergence. If a CFD problem can be solved by a conventional under-relaxation method, there is typically an optimum value of relaxation factor which produces the solution in the smallest number of iterations. The optimum relaxation factor depends on the nature of the problem, the number of grid points used for discretization, grid spacing, iterative procedure used and other parameters [1]. In general, no method has been found for determining the optimum relaxation factor analytically in non-linear problems. The user can either attempt to guess it based on experience with similar problems or find it by exploratory computations.

In the current work, the iterative solver of the discretized governing equations is treated as if it were a plant to be controlled. Since an analytical model for this plant does not exist, using a rule based system [2] to decide on relaxation actors emerges as a suitable approach. The set of rules uses fuzzy logic [3] as the method of making decisions. Fuzzy logic was chosen as a robust technique which enables qualitative judgements applied to parameters quantitative in nature and highly problem dependent. Through guided under-relaxation, the fuzzy logic algorithm controls another algorithm, which is the CFD solver [4–7]. Algorithms for controlling algorithms have been recently introduced as a new discipline in soft computing [8,9].

The fuzzy rule set is based on the observed relationship between the oscillations in magnitude of the solution vector during the iterations and the speed of convergence. The decision making system considers a large number of characteristic values taken from previous iterations, which compose the solution history curve. A standard for fast convergence or "set point" is a mildly wavy shape of the solution history curve. Any deviations from this standard are recognized either as slow convergence or divergent behavior depending on the value of a generalized error. The deviations are minimized by implementing a table of fuzzy rules which utilizes a proportional-integral type of controller.

Two numerical models of turbulent flow and heat transfer are used for tuning the fuzzy membership functions applied in the control algorithm. The calculations of turbulent flow are based on a k–ε model of turbulence proposed by Ince and Launder [10], which involves two additional scalar transport equations, coupled with the momentum, continuity, and energy equations. The turbulent flow models are viewed as a particular challenge for fuzzy control algorithms since their computational difficulty is increased by the complexity of the governing equations.

In order to optimize the defining parameters of the fuzzy membership functions, many evaluations of the final numerical solution need to be done. This process can take a very long time, especially in the case of solving turbulent flow and heat transfer. For that reason, a gradient method was used as an optimization tool. This method was easy to implement and requires only the evaluation of the first order partial derivatives, which saves time compared to the quasi-Newton methods, for example.

2. Analysis

To examine the potential of fuzzy control algorithms applied to complex and difficult problems, this study is focused on turbulent flow. The time-averaged k–ε model of the governing equations reported in [10–12] is solved. An in-house CFD code with the SIMPLER algorithm was modified in order to simulate low Reynolds number turbulence. The problems considered here are steady and incompressible. The Boussinesq assumption of proportionality between the turbulent sharing stress and the mean strain tensor holds.

Each of the state variables φ (velocity components and temperature) was relaxed separately. The algorithm used in this study does not include the relaxation of pressure. The control algorithm updates the relaxation factors α^{φ} at every iteration based on the information stored from a finite number of previous iterations. In this study, the number of previous iterations considered is N where

$$N = n \quad \text{if } n < 50, \tag{2}$$

$$N = 50 \quad \text{if } n \geqslant 50 \tag{3}$$

and n is the current number of iterations. The algorithm has two principal parts: one evaluates the nature of the "solution history curve" and the other controls the features of this curve during iteration in order to produce and preserve those features that bring the fastest convergence. The solution history curve consists of a characteristic quantity extracted from the solution and observed over the last N iterations. For each of the state variables, a separate history curve is considered. The characteristic quantity that represents the solution at the nth iteration is the natural norm of the solution, also known as the magnitude of the solution vector

$$S^{\varphi}(n) = \sqrt{\sum_{i=1}^{l}\sum_{j=1}^{m}[\varphi_n(i,j)]^2}, \tag{4}$$

where i and j are the node numbers in x- and y-direction, respectively, φ_n is the nodal value of the state variable φ at the iteration n, l is the total number of nodes in the x-direction and m is the total number of nodes in the y-direction.

The magnitude of the solution vector $S^{\varphi}(n)$ at a particular iteration does not have any significant meaning if considered isolated from other iterations. The history of this quantity during code execution gives information on how stable and fast the convergence is and whether divergence is likely to occur. Fig. 1 shows some typical situations which can occur depending on the numerical difficulty of the problem and the values of constant relaxation factors used. For a given problem, low values of relaxation factor typically produce a slow rate of change of the solution from iteration to iteration. This results in stable but slow convergence, as shown in Fig. 1(a). For large systems of equations, this could mean a big expenditure of CPU time. As the relaxation factor increases, the rate of change of the solution becomes larger and the system approaches convergence faster. The solution history curve is smooth, with a mildly wavy shape due to a few overshoots or undershoots, as in Fig. 1(b). After a certain value of relaxation factor, the rate of change of the solution vector from iteration to iteration becomes too fast for the given system of algebraic equations. This manifests itself as a rough appearance of the history curve and is followed by delayed, unstable convergence, as in Fig. 1(c). Further increase in relaxation factor makes the guess of the solution during iterations almost random and the system of equations either never converges or diverges.

Spectral analysis was used in order to identify these observed trends in the solution history and distinguish them from one another. By saving the magnitude of the solution vector S^{φ} over a finite number of iterations from $n-N$ to n, a set of data is obtained which can be treated as a discrete time representation of a signal. At every iteration, a discrete Fourier transform is performed on this data, according to the following formula:

$$H_f = \sum_{k=0}^{N-1} S_k^{\varphi} \mathrm{e}^{2\pi \mathrm{i} k f}, \tag{5}$$

where f is the frequency of the periodic components of the signal

$$-\frac{1}{2} \leqslant f \leqslant \frac{1}{2}, \tag{6}$$

k is the summation index and i is the imaginary unit. The solution vector magnitude S^{φ} was summed over the last N saved values. After finding the Fourier transform of S^{φ}, only its peak values and corresponding frequencies are considered since they identify the main harmonics of the function S^{φ} on the given interval.

The Fourier transform can be used to identify how dominant the harmonics of particular frequencies are in the solution history curve. Therefore, only the frequencies that have physical meaning are considered:

$$0 \leqslant f \leqslant \frac{1}{2}. \tag{7}$$

The negative frequency domain is dropped, which is also supported by the fact that the Fourier transform is symmetric around the ordinate. The lowest value at $f = 0$ means that the corresponding harmonic does not oscillate at all, while the highest value at $f = 0.5$ means that the harmonic "zig-zags" on every iteration, which results in one cycle per two iterations or 25 cycles per interval, for $N = 50$. Dominance of the harmonics with high frequencies means that the convergence is unstable and the corresponding relaxation factor needs to be decreased, while their absence implies smooth slow convergence. The influence of the particular harmonic with frequency f on the solution history curve is estimated using the absolute value of the amplitude of that harmonic normalized by the average value of S^{φ} on the considered interval of N points. Based on the properties of Fourier transform, it can be shown that the amplitude of the harmonic with frequency f can be expressed as

$$A_f = \frac{2}{N} \sum_{k=0}^{N-1} S_k^{\varphi} \mathrm{e}^{2\pi \mathrm{i} k f} = \frac{2}{N} H_f \equiv a_f + \mathrm{i} b_f, \tag{8}$$

where a_f and b_f are the amplitudes of cosine and sine function, respectively. The average value of the magnitude of the solution vector on the considered interval is proportional to the Fourier transform that corresponds to frequency $f = 0$

$$\bar{S}^{\varphi} = \frac{1}{N} \sum_{k=0}^{N-1} S_k^{\varphi} = \frac{1}{N} H_0. \tag{9}$$

According to (8) and (9), the absolute normalized amplitude of the harmonic with frequency f is

$$A_f^* = \frac{|A_f|}{\overline{S^{\varphi}}} = 2 \frac{\sqrt{\mathrm{Re}^2(H_f) + \mathrm{Im}^2(H_f)}}{H_0}, \tag{10}$$

where the operators Re and Im denote real and imaginary part of the complex number, respectively.

As many observations show, optimal convergence occurs near the edge of instability which is characterized by the appearance of low amplitude, high frequency harmonics. The controller presented in this study is designed to keep the amplitudes of high frequency harmonics at very small values, while at the same time allowing and encouraging oscillations with low frequency.

The dimensionless amplitudes of the main harmonics A_f^* are multiplied by a weighting function designed to highlight the harmonics with high frequencies

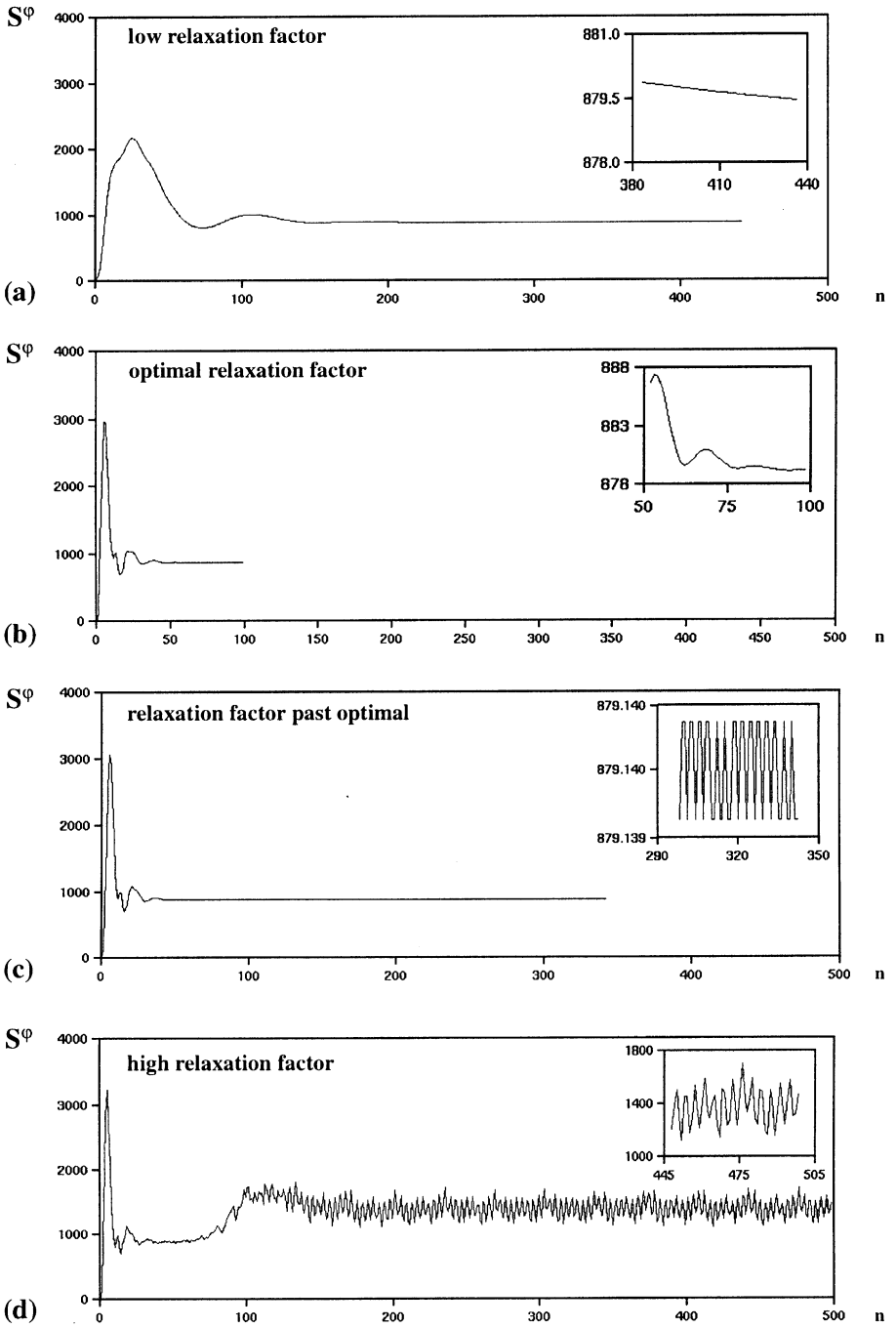

Fig. 1. Magnitude of the solution vector versus number of iterations for different relaxation factors. The smaller window represents the S_φ vector at the last 50 iterations.

$$W_f(f) = \exp\left(\frac{p_z f}{f_{\max}}\right), \tag{11}$$

where $f_{\max} = 0.5$ is the maximum frequency of one cycle per two iterations. Initially, the value of the constant $p_z = 23$ is chosen so that the weighting function satisfies the condition $1 \leqslant W_f(f) \leqslant 10^{10}$. The choice of 10^{10} as the maximum value of $W_f(f)$ is somewhat arbitrary and based on exploratory computations. For that reason, the constant p_z was one of the parameters which were op-

timized in this study. The weighting function takes on low frequencies and vice versa. As a result, if the A_f^* at $f = f_{\max}$ is on the order of magnitude of 10^{-10}, the product $W_f(f_{\max})A_f^*(f_{\max})$ will be on the order of magnitude of one. At the same time, if A_f^* at $f = 0.02$ is on the order of magnitude of one, the product $W_f(0.02)A_f^*(0.02)$ will have a magnitude on the order of unity as well. This mean that if $W_f(f)A_f^*(f)$ is to be kept near unity, the harmonics with high frequencies need to have low amplitudes and vice versa. Since unstable solution history curves are characterized by the presence of harmonics with high amplitudes and frequencies, for that case the following is satisfied:

$$W_f(f)A_f^*(f) > 1. \tag{12}$$

In order to insure fast and stable convergence, the control algorithm needs to enforce

$$W_f(f)A_f^*(f) \leqslant 1, \tag{13}$$

but the weighted amplitudes need to be as close to one as possible. This is satisfied for

$$\max[W_f(f)A_f^*(f)] \approx 1, \tag{14}$$

where the left-hand side represents the maximum of all the values of $W_f(f)A_f^*(f)$ taken at peaks of the Fourier transform with frequencies f.

The control problem was formulated so that the set point is unity, and the error is defined as

$$e(n) = \ln(\max[W_f(f)A_f^*(f)]). \tag{15}$$

The error difference is

$$\Delta e(n) = e(n) - e(n-1). \tag{16}$$

The fuzzy rule set was designed with the error and the error difference as inputs and the relative increment in relaxation factor $\Delta\alpha^\varphi/\alpha^\varphi$ as the output. The rules are given in Table 1. As an example, one of the rules states the following: if the error is negative small (NS) and the error difference is negative medium (NM) the relative increment in relaxation factor is positive medium (PM). This means that if the weighted amplitudes of the main harmonics are lower then one and decreasing, the relaxation factor needs to be increased since the rate of convergence is slowing down. Another rule states: if the error is positive medium (PM) and the error difference is positive big (PB), the change in relaxation factor is negative big (NB). The meaning of this rule is that the relaxation factor needs to be decreased if any of the weighted amplitudes of the main harmonics are higher than one and increasing in size, since this behavior indicates probable instability. The membership functions are shown in Fig. 2. The initial values of the defining parameters are $p_x = 0.001$, $p_y = 0.5$ and $p_z = 23$. The

Table 1
Fuzzy rule set[a]

PB		NS	NM	NM	NB	NB
PM	PS		NS	NM	NM	NB
PS	PM	PS		NS	NM	NM
NS	PM	PM	PS		NS	NM
NM	PB	PM	PM	PS		NS
NB	PB	PB	PM	PM	PS	
	NB	NM	NS	PS	PM	PB

[a] The output variable is the increment in relaxation factor, while the inputs are errors and error differences, as shown. PB = positive big; PM = positive medium; PS = positive small; NS = negative small; NM = negative medium; NB = negative big.

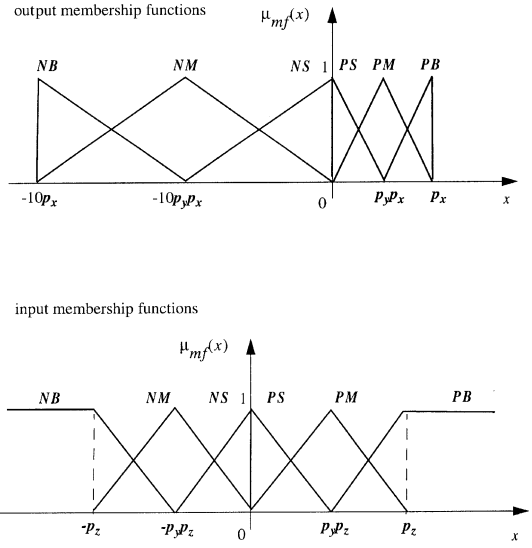

Fig. 2. Fuzzy membership functions expressed in terms of the parameters p_x, p_y and p_z for the purpose of tuning.

parameters are selected so that the values of the increments in relaxation factor are small, compared to those for errors and error differences. This is because large increments in α^φ cause instability for highly nonlinear problems.

The program starts execution with the default values of relaxation factor $\alpha^\varphi = 1$. The rule set is activated on every iteration. When the magnitudes of the residual vectors are increasing, positive increments in relaxation factors evaluated by fuzzy rules are ignored. If divergence has occurred, which is recognized when the magnitude of the solution vector hits the machine limit, the code is restarted with relaxation factors reduced by 10%. After the increment in relaxation factor is defuzzified, the relaxation factor is updated by

$$\alpha(n+1) = \alpha(n) + \Delta\alpha(n). \tag{17}$$

3. Benchmark problems

The fuzzy controller was tested on two problems of turbulent flow and heat transfer: mixed convection over a backward facing step, and buoyancy driven flow in a square cavity. The flow is two-dimensional, governed by the differential equations of fluid motion which utilize a two-equation model of turbulence proposed by Ince and Launder [10]. In each case, the convergence criterion was given by

$$\frac{\max|\varphi(n)-\varphi(n-1)|}{\max|\varphi(n)|\alpha^{\varphi}(n)} \leqslant 10^{-5}, \tag{18}$$

where the maximum iterative error in the domain was normalized by the maximum value in the domain and the relaxation factor. Using the relaxation factor in the denominator avoids false convergence, which can occur for low relaxation factors [1].

4. Turbulent mixed convection over a backward facing step

Turbulent, mixed convection in a vertical channel with a backward-facing step was chosen as a benchmark problem. The geometry of the channel is given in Fig. 3. Fluid is entering the channel with a uniform velocity $v_i = 0.3$ m/s. The values of $k_i = 5 \times 10^{-4}$ m^2/s^2 and $\varepsilon_i = 1.0 \times 10^{-4}$ m^2/s^3 at the inlet were determined by experiment as reported in [11]. These induce a realistic level of turbulence in the channel. The inlet temperature is uniform and equal to the temperature of the isothermal left wall T_C. The vertical surfaces of the right wall are kept at a constant temperature T_H, which is 30°C higher than the temperature of the left wall. The horizontal part of the step is insulated. At the wall, the no-slip condition is assumed and the values of k and ε are prescribed to be zero. At the outlet section at the top of the channel, fully-developed flow is assumed. Diffusion flux of the mean velocities, temperature, kinetic energy of turbulence and dissipation of turbulence kinetic energy is prescribed to be zero.

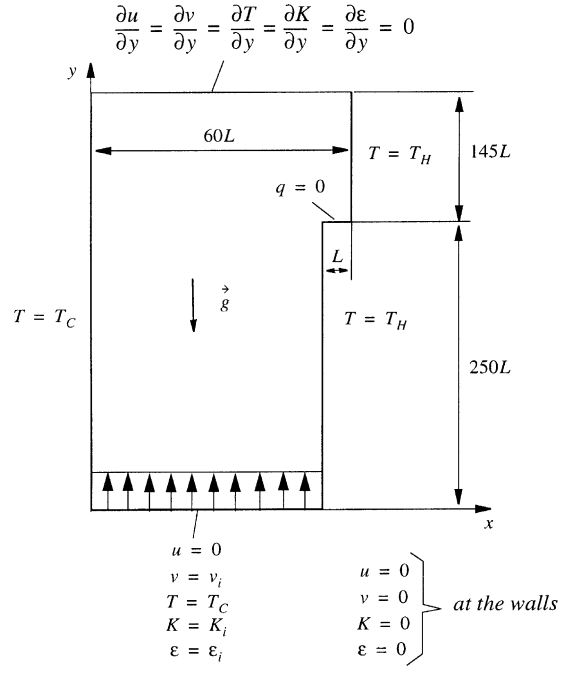

Fig. 3. Turbulent mixed convection over a backward facing step.

The channel is wide, compared to the size of the step. The small expansion ratio of the channel insures that the assumption of two-dimensionality of the flow is valid. The flow is entirely driven by the pressure drop across the channel and the buoyancy force caused by the temperature difference between the opposite walls. The Grashof number based on the height of the channel is $Gr = 3.66 \times 10^{11}$. The Reynolds number based on the inlet velocity and the width of the channel upstream from the step is $Re = 1.3 \times 10^4$. The fluid is air, with a Prandtl number of $Pr = 0.71$.

The total of 68 control volumes was used in the horizontal direction and 66 volumes were used to model the flow in the vertical direction. In the fluid region upstream from the step, 50×30 control volumes were used. For the region downstream from the step, 68×36 control volumes were used. The step was modeled as a rectangular region with viscosity prescribed as $v = 10^{30}$ cm^2/s. This measure insured that the region behaves as a solid. The mesh behind the step was densely spaced in order to capture the recirculating flow correctly. A high density of control volumes near the walls was used as well, in order to insure that the velocity and temperature boundary layers are correctly approximated.

5. Turbulent buoyancy driven flow in a rectangular cavity

A second benchmark case features turbulent flow in a tall rectangular cavity with an aspect ratio of 10. The flow is entirely driven by the buoyancy force, which is caused by the temperature difference between the vertical isothermal walls. The problem is two-dimensional, with governing equations given in [10].

The flow geometry and boundary conditions are given in Fig. 4. The temperature difference between the isothermal walls is 25°C. The horizontal walls are adiabatic. The no-slip condition is assumed at the solid surfaces, with the values of $k = 0$ and $\varepsilon = 0$ prescribed. The Grashof number based on the height of the channel is $Gr = 2.81 \times 10^{10}$.

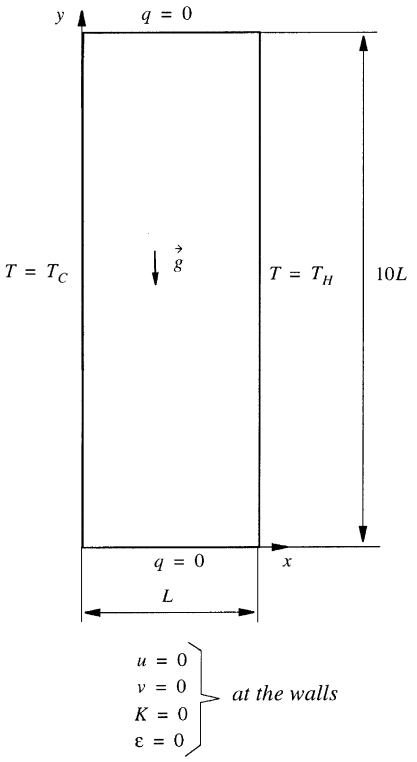

Fig. 4. Turbulent buoyancy driven flow in a rectangular cavity.

The grid used for modeling the rectangular region is nonuniform. The number of control volumes used in both directions in 68. A high density of control volumes in the wall region was necessary in order to resolve the boundary layer velocity and temperature profiles. The grid distribution across the computational half-domain is given as

$$x_i = \frac{L}{2}\left(2\frac{i-2}{l_{cv}}\right)^{2.5} \quad \text{for } 2 \leqslant i \leqslant \frac{l_{cv}}{2}+2, \tag{19}$$

$$y_i = L\frac{j-2}{m_{cv}} \quad \text{for } 2 \leqslant j \leqslant \frac{m_{cv}}{2}+2, \tag{20}$$

where x_i and y_j are the node coordinates, i and j and the counters in the horizontal and vertical direction, respectively, l_{cv} is the total number of control volumes in the x-direction and m_{cv} is the total number of control volumes in the y-direction.

6. Optimization of fuzzy membership functions

Since many evaluations of the numerical solution during the optimization of fuzzy membership functions can require an excessive amount of CPU time, only three parameters are selected to be optimized. One of the chosen parameters is the positive domain of definition of the output membership functions p_x, as shown in Fig. 2. Previous numerical investigations indicated that this parameter has a large influence on the speed and stability of convergence. The negative part of the interval is maintained ten times wider then the positive one, in order to insure a fast drop of the relaxation factor if the iterations become unstable. The second parameter to optimize is the relative position of the peak which defines the "positive medium" and "negative medium" membership functions. This peak coincides with the point where the "positive (or negative) small" membership function stops and "positive (or negative) big" membership function begins. The parameter is denoted as p_y in Fig. 2. Its significance comes from the fact that it determines the centers of areas of all the membership functions, on a given interval of definition. The third parameter to optimize is the half-domain of the input membership functions, denoted as p_z in Fig. 2. This parameter is directly linked to the tolerated frequency range of the solution vector oscillations during the iterative process.

The gradient method is employed in order to find the set of parameters p_x, p_y, and p_z, which minimizes the number of iterations needed for the solution to converge. In order to avoid divergence or minimize the number of oscillations needed for convergence, the defining parameters for the membership functions need to varied in such a way that they stay within an order of magnitude of their original values. In addition, there is a large difference between the orders of magnitude of p_x, p_y and p_z. A reasonable range of p_z is 10,000 times wider then the corresponding interval of p_x. This causes the formation of a long, narrow "valley" in the objective function distribution around the optimal point. In order to avoid slow convergence associated with this phenomenon, the parameters are normalized by their initial values. At the beginning of the optimization, the normalized parameters have values $p_x^* = p_y^* = p_z^* = 1$. All the gradient evaluations and updates of the parameters are performed in the normalized space. Before the fuzzy membership functions are updated, the normalized parameters are converted back to p_x, p_y and p_z.

The optimization algorithm starts by obtaining the necessary gradient information. This information is stored in a "cell" which consists of four points in the 3-D space of parameters, as shown in Fig. 5. Point 1 represents the initial set of normalized parameters p_x^*, p_y^* and p_z^*. This set is accompanied by the number of iterations needed for the solution to converge, with the fuzzy membership functions defined by that particular set:

$$n_i = F(p_x, p_y, p_z). \tag{21}$$

Point 2 is obtained by perturbing the parameter p_x^* by 5% of its value and finding the new number of iterations needed for convergence. Similarly, points 3 and 4 are

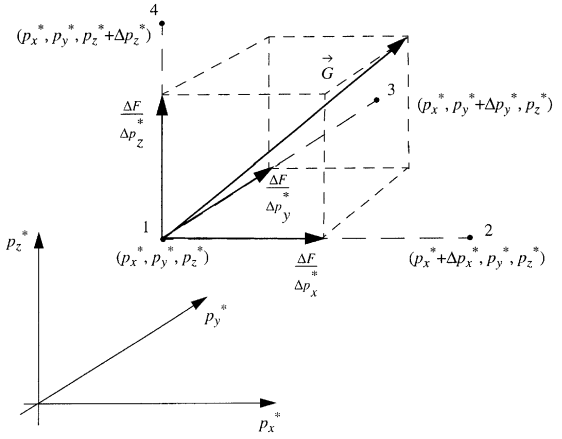

Fig. 5. Approximate estimation of gradient during the optimization. Function F represents the number of iterations needed for a solution to converge, with a given set of parameters p_x, p_y and p_z.

obtained by the equivalent perturbations of the parameters p_y and p_z, respectively. Each time the perturbations were made, the code was run in order to obtain the number of iterations for convergence. The approximate partial derivatives in three perpendicular directions are estimated as

$$\frac{\Delta F}{\Delta p_x^*} = \frac{F(p_x^* + \Delta p_x^*, p_y^*, p_z^*) - F(p_x^*, p_y^*, p_z^*)}{\Delta p_x^*}, \tag{22}$$

$$\frac{\Delta F}{\Delta p_y^*} = \frac{F(p_x^*, p_y^* + \Delta p_y^*, p_z^*) - F(p_x^*, p_y^*, p_z^*)}{\Delta p_y^*}, \tag{23}$$

$$\frac{\Delta F}{\Delta p_z^*} = \frac{F(p_x^*, p_y^*, p_z^* + \Delta p_z^*) - F(p_x^*, p_y^*, p_z^*)}{\Delta p_z^*}. \tag{24}$$

The ratios given by Eqs. (22)–(24) form a gradient vector G, which points into the direction of the steepest increase in the value of the objective function F.

The next task of the optimization algorithm is to change the parameters p_x^*, p_y^* and p_z^* along the direction opposite from the gradient G, until the minimum of the objective function is found. The direction of steepest descent is only local to the current point and varies in the parametric space. For this reason, the vector of parameters needs to be changed in small increments and the gradient needs to be recalculated as frequently as possible. The advancement of the parameters in the direction of steepest descent of the function F is given by the following expressions:

$$p_{x(k+1)}^* = p_{x(k)}^* - \lambda \left.\frac{\Delta F}{\Delta p_x^*}\right|_k, \tag{25}$$

$$p_{y(k+1)}^* = p_{y(k)}^* - \lambda \left.\frac{\Delta F}{\Delta p_y^*}\right|_k, \tag{26}$$

$$p_{z(k+1)}^* = p_{z(k)}^* - \lambda \left.\frac{\Delta F}{\Delta p_z^*}\right|_k, \tag{27}$$

where k is the number of the current step in the optimization process. The parameter λ determines the size of the increment in the vector of parameters $\{p_x^*, p_y^*, p_z^*\}$ along the direction opposite from the gradient G. At the beginning of the optimization process, the value of λ is estimated as

$$\lambda = \frac{0.1\sqrt{(p_x^*)^2 + (p_y^*)^2 + (p_z^*)^2}}{\sqrt{\left(\frac{\Delta F}{\Delta p_x^*}\right)^2 + \left(\frac{\Delta F}{\Delta p_y^*}\right)^2 + \left(\frac{\Delta F}{\Delta p_z^*}\right)^2}}. \tag{28}$$

This limits the initial increment in the vector of parameters to 10% of its magnitude. If the gradient is decreasing during further steps, the value of λ is kept constant and the change in vector $\{p_x^*, p_y^*, p_z^*\}$ becomes smaller, as it approaches the minimum. If the gradient is increasing, the λ needs to be recalculated by the application of the Eq. (28). This insures that the change in the vector of parameters stays small.

Every new set of values of p_x^*, p_y^* and p_z^* is followed by finding the solution of the CFD problem for which the fuzzy membership functions are being optimized. This makes the optimization process very time-consuming. In order to reduce the number of runs of SIMPLER, the gradient vector is updated in a cycle manner, one direction at a time.Whenever a new guess in the values of p_x, p_y and p_z is made by the application of Eqs. (25)–(27), a different parameter is perturbed and the corresponding component of the gradient vector is updated. The optimization continues until the set of parameters p_x^*, p_y^*, and p_z^* starts changing insignificantly and the magnitude of the gradient vector gets close to zero. This indicates that a local minimum of the objective function F has been found.

7. Results and discussion

The solutions obtained by using SIMPLER with the control algorithm were verified by comparison with the results published in [11,12]. The details of verification are available in [7]. All the solutions satisfy the convergence criterion given by Eq. (18). In the case of turbulent flow over a backward facing step, a grid independent solution was obtained by using 68 control volumes in the horizontal and 66 control volumes in the vertical direction. The convergence of the solution was strongly dependent on the grid density near the walls, grid density in the recirculating flow region immediately behind the

step, and the variation of grid density in the computational domain. A similar influence of the mesh density distribution on convergence was noticed for turbulent buoyancy driven flow in a rectangular cavity. For this numerical model, a grid-independent solution was obtained by using 68 control volumes in each direction.

A comparison between the performance of the fuzzy controller and the case using constant relaxation factors was done for the initial set of parameters which define fuzzy membership functions: $p_x = 0.001, p_y = 0.5$ and $p_z = 23$. The application of the fuzzy controller in solving turbulent flow over a backward facing step resulted in convergence after 8168 iterations. The iterative process was initialized with the values of the relaxation factors for u, v, T, K and ε equal to unity. During the iterations, the relaxation factors were altered by the fuzzy control algorithm, as shown in Fig. 6. If the set of relaxation factors was too high, the solution diverged after only a few iterations and the code needed to be restarted. At every restart, the relaxation factors were reduced by 10% until their values dropped low enough so that restarting was no longer needed. Between restarts, the relaxation factors are also modified by the fuzzy control algorithm. The last restart of the solver marks the end of this initial, unstable zone, as shown in Fig. 6. In the case of turbulent flow over a backward facing step, the total number of restarts was 38 and the last one occurred after 179 iterations. The values of the relaxation factors immediately after the last restart were

$$\alpha^u = 0.0211, \tag{29}$$

$$\alpha^v = 0.0218, \tag{30}$$

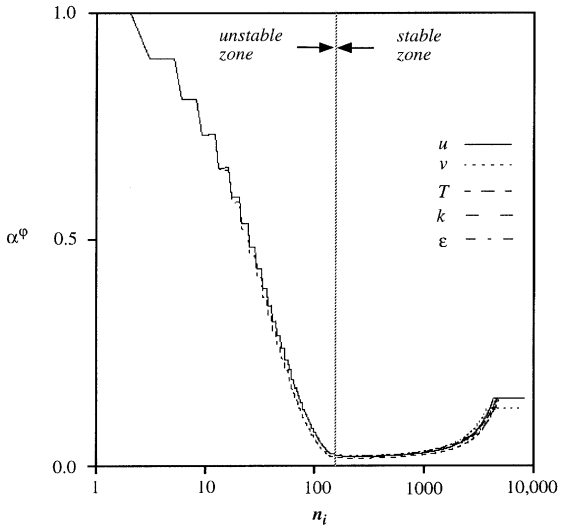

Fig. 6. Relaxation factors applied on state variables versus number of iterations for turbulent flow over a backward facing step.

$$\alpha^T = 0.0165, \tag{31}$$

$$\alpha^K = 0.0211, \tag{32}$$

$$\alpha^\varepsilon = 0.021. \tag{33}$$

During the stable portion of the iterative process, the relaxation factors were continuously increased by the fuzzy rules. For turbulent flow over a backward facing step, the final values of the relaxation factors were one order of magnitude larger then those obtained after the last restart of the solver.

In order to demonstrate how the fuzzy control of the relaxation factors influenced the speed of convergence, the same problem was solved by using constant relaxation factors, with the values given by Eqs. (29)–(33). These are considered to be the largest constant values of the relaxation factor, which don't cause the solver to diverge. The number of iterations needed to obtain the solution was 42,707, which is more then five times larger than in the case when the fuzzy controller was used. The total saving in CPU time obtained by the fuzzy control algorithm was 7 h 52 min and 39 s. All the computations were performed on a cluster of IBM RS/6000 workstations with a 580 MHz processor.

To illustrate the convergence properties of the iterative procedure with the fuzzy controller and with constant relaxation factors, the behavior of the iterative errors during the computation is shown in Fig. 7. The natural norm of the vector of iterative errors on the computational domain was divided by the natural norm of the corresponding variable and used as an indicator of the speed of convergence. Comparison between Figs. 7(a) and (b) reveals a similarity between the curves obtained by the controller and those obtained by the constant relaxation factors. In both cases, there is the presence of an initial transient interval, characterized by oscillatory behavior of the iterative errors. The length of this interval is about 2000 iterations, regardless of whether the fuzzy controller or the constant relaxation factors were used. It is followed by a smooth, monotonic decrease of the iterative errors, until the limit which marks convergence was reached. The speed of drop of the iterative errors was more then 5 times faster with the fuzzy controller than with the constant relaxation factors. This implies that the gradual increase of relaxation factors by the fuzzy controller during the iterations significantly reduced the overall computational effort required to solve this problem.

In the case of the turbulent buoyancy driven flow in a rectangular cavity, the solution obtained by the fuzzy control algorithm converged after 7099 iterations. During the unstable period of the iterative process, the solver had to be restarted 13 times. The last restart occurred after 43 iterations. After the last restart, the relaxation factors had been reduced to

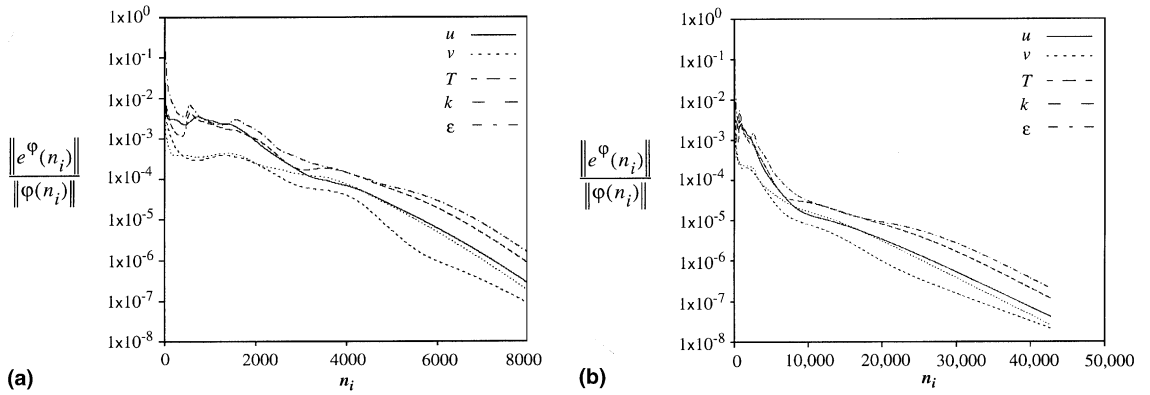

Fig. 7. Normalized iterative errors versus number of iterations for turbulent flow over backward facing step. The results given in (a) were obtained by the fuzzy control algorithm, while (b) represents the errors obtained by constant relaxation factors.

$$\alpha^u = 0.277, \tag{34}$$

$$\alpha^v = 0.279, \tag{35}$$

$$\alpha^T = 0.271, \tag{36}$$

$$\alpha^K = 0.288, \tag{37}$$

$$\alpha^\varepsilon = 0.288. \tag{38}$$

The relaxation factors were further increased by the fuzzy sets of rules, until convergence was achieved.

The performance of the fuzzy logic controller was compared to the iterative procedure obtained with the constant relaxation factors, given by Eqs. (34)–(38). The constant relaxation factors produced convergence after 16,153 iterations, which is more then two times larger compared to the number of iterations needed for convergence with the fuzzy controller. The total CPU time saved by using the fuzzy rule set in this case was 1 h 56 min and 27 s.

During the tuning procedure, new sets of values of p_x, p_y and p_z are successively produced by the gradient method, until the minimum number of iterations needed for convergence is reached. Fig. 8 shows the number of iterations needed for the solution to converge versus the step number of the optimization process. The two "bumps" in Fig. 8(a) indicate overshoot of the optimum point. The gradient method converged in 17 steps, which took more than 10 hours of CPU time. The large expenditure in CPU time can be explained by the large number of iterations needed for convergence at every step. In the case of mixed convection over a backward facing step, a slightly smaller number of steps were

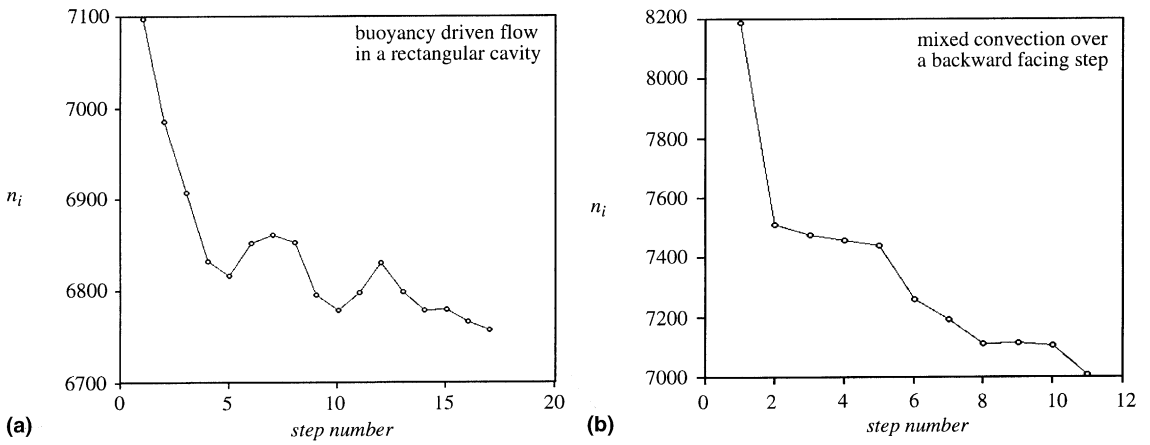

Fig. 8. Tuning of the fuzzy membership functions for two numerical models of turbulent flow and heat transfer. The total number of iterations needed for convergence for every set of fuzzy membership functions is shown versus step number.

Table 2
Defining parameters for optimal membership functions evaluated on numerical models of turbulent flow and heat transfer[a]

Numerical model	Optimal membership functions		
	p_x	p_y	p_z
Buoyancy driven flow in a rectangular cavity	0.00195	0.4121	21.272
Mixed convection over a backward facing step	0.001653	0.6856	20.3509

[a] The initial values of p_x, p_y, and p_z are 0.001, 0.5 and 23, respectively.

needed for optimization, as shown in Fig. 8(b). The optimization went smoothly without any overshoots of the extremum point. However, the execution time was longer than in the case shown in Fig. 8(a). This is due to the considerably larger number of iterations needed for this numerical model to converge.

The defining parameters for the optimal membership functions in the case of the turbulent flow models are shown in Table 2. The positive interval of definition of the output membership functions, p_x, increased for both models but stayed within the same order of magnitude as the initial value of 0.001. This indicates that the guess for p_x made by trial and error and some experience with laminar models was done correctly. The peak of the membership functions "positive medium" and "negative medium" was significantly altered from its original value of 0.5, as shown in Table 2. The values of this parameter changed by the order of 0.1 in the positive and negative direction. The lower values of p_y indicate a shifting of the centers of area of the membership functions towards the origin. This produces slower variations in relaxation factor due to the action of the fuzzy controller. Similarly, the higher values of p_y produce faster changes in relaxation factor during iterations.

For the two turbulent models presented in Table 2, the half-domain of the input membership functions, p_z, reduced very slightly from its original value. By comparing the values of p_x and p_z listed in the table for the two numerical models, it is interesting to note that a higher value of p_x is followed by a higher value p_z and vice versa. The parameter p_z is closely related to the desired amplitude of iterative oscillations of the solution vector magnitude, through the formula given by Eq. (11). If the output membership functions allow large positive increments in relaxation factors, then the restriction on the sizes of amplitudes A_f^* needs to be more severe in order to prevent divergence. This means that the alarming amplitude of the iterative oscillations will be lower in this case, for any particular non-zero frequency. High rates of increase in relaxation factors need to be accomplished by low accompanied by low amplitude iterative oscillations and vice versa, in order to achieve fast convergence. No apparent relationship can be found between p_y and p_x or p_y and p_z, from the data available so far and with the methodology used in this study.

The performance of the triangular membership functions before and after tuning is shown in Table 3. The tuning of the defining parameters reduced the number of iterations needed for convergence by 340 in the case of buoyancy driven flow in a rectangular cavity. This was only a 5% improvement in the speed of convergence, which means that the original set of fuzzy membership functions was nearly optimal for this problem. However, tuning of the membership functions applied in mixed convection over a backward facing step reduced the number of iterations needed for solution by more then 1000, which is nearly a 20% improvement.

It would not be practical to tune the rule set for every new problem encountered. The gain in execution time would be far outweighed by the time needed to perform the optimization. Instead the goal of the exploratory work described here is to demonstrate: (1) that a single fuzzy rule set (the non-optimized set) can be used on two very different problems in turbulent flow and (2) that it is possible to produce a better rule set on an individual problem by using optimization techniques. The hope is that once a fairly large number of individual cases are optimized, clear guidelines for a single effective rule set will emerge. The authors have previously shown [6] that a single rule set was able to solve five very different cases of laminar flow and heat transfer. There is good potential for the development of a single rule set that is able to solve a board range of problems in turbulent flow.

Table 3
Number of iterations needed for convergence for two sets of triangular membership functions[a]

Numerical model	Characteristic numbers	Control volumes used	Number of iteration	
			Optimal	Before tuning
Buoyancy driven flow in a rectangular cavity	$Gr = 2.8 \times 10^{10}$	68×68	6758	7098
Mixed convection over a backward facing step	$Gr = 3.66 \times 10^{11}$ $Re = 1.3 \times 10^4$	68×68	7009	8189

[a] The numerical models feature turbulent flow and heat transfer.

8. Summary

A control algorithm which selects relaxation factors at every iteration according to criteria that insures fast and stable convergence was developed. The algorithm uses spectral analysis and fuzzy logic in order to evaluate and control convergence during iteration.

In order to examine the potential of fuzzy control algorithms applied in difficult problems, the time averaged k–ε model was used for solving natural convection in a rectangular cavity and mixed convection over an adiabatic backward facing step. The total saving in CPU time using the fuzzy controller for the rectangular cavity problem was two hours and the savings for the backward facing step was eight hours.

The triangular membership functions were tuned by singling out three significant defining parameters and optimizing them for each numerical model separately. The optimization improved the CPU time needed for convergence by 5% in the case of turbulent buoyancy driven flow in a rectangular cavity and 20% in the case of turbulent mixed convection over the backward facing step. The implication of the study is that shape and configuration of the fuzzy membership functions can have an important influence on the efficiency of the controller.

The savings in CPU time are not the only important effects of using a fuzzy controller. There is perhaps a very significant saving in engineering time, since the user will no longer be required to manually search for a set of relaxation factors that avoid divergence. In addition, the idea of using fuzzy logic to control a non-linear simulation does not rely on any particular features of the SIMPLER algorithm. It is reasonable to suppose that a rule set could be found for many other algorithms as well. It was not particularly difficult to find a set of membership functions and rules that produced good results and the solution for the problems described here was not very sensitive to the parameters used in the fuzzy logic. Thus fuzzy controllers have high potential for useful application in numerical simulation.

References

[1] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Series in Computational Process and Thermal Sciences, Hemisphere, Washington, DC, 1980.
[2] A.A. Hopgood, Knowledge Based Systems for Engineers and Scientists, Electronic Engineering Systems Series, CRC Press, Boca Raton, 1993.
[3] L. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[4] J. Ryoo, D.A. Kaminski, Z. Dragojlovi, Automatic convergence in a computational fluid dynamics algorithm using fuzzy logic, in: The Sixth Annual Conference of the Computational Fluid Dynamics Society of Canada, Quebec, VIII, 1998, pp. 1–6.
[5] J. Ryoo, Control of under-relaxation in computational fluid dynamics using fuzzy logic, Master of Science Thesis, Rensselaer Polytechnic Institute, Troy, NY, 1998.
[6] Z. Dragojlovic, D.A. Kaminski, J. Ryoo, Control of convergence of a computational fluid dynamics algorithm using fuzzy logic, ASME IMECE, Anaheim Brea, CA, 1998.
[7] Z. Dragojlovic, Control of convergence in a computational fluid dynamics algorithm using fuzzy logic, Ph.D. Thesis, Rensselaer Polytechnic Institute, NY, 2000.
[8] P. Arabshahi, J.J. Choi, R.J. Marks, T.P. Caudelll, Fuzzy control of back propagation, in: First IEEE International Conference on Fuzzy Systems, San Diego, CA, 1992. pp. 967–972.
[9] M.A. Lee, H. Tagaki, Dynamic control of genetic algorithm using fuzzy logic techniques, in: Proceedings of the Fifth International Conference on Genetic Algorithms, 1993, pp. 76–83.
[10] N.Z. Ince, B.E. Launder, On the computation of buoyancy driven turbulent flows in rectangular enclosures, Int. J. Heat Fluid Flow 10 (1989) 110–117.
[11] J.Z. Zhao, A. Li, T.S. Chen, B.F. Armaly, Turbulent mixed convection flow over a vertical backward-facing step, in: AIAA/ASME Joint Thermophysics and Heat Transfer Conference, 2, ASME, 1998. pp. 185–191.
[12] P.W. Giel, F.W. Schmidt, A comparison of turbulence modeling predictions to experimental measurements for high Rayleigh number natural convection in enclosures, in: Proceedings of the Ninth International Heat Transfer Conference, 1, Jerusalem, Israel, 1990, pp. 175–180.