PERGAMON

Computers & Fluids 32 (2003) 97–108

computers & fluids

www.elsevier.com/locate/compfluid

Neural networks based subgrid scale modeling in large eddy simulations

F. Sarghini [a,*], G. de Felice [a], S. Santini [b]

[a] *Department of Energetics, Applied Thermofluid Dynamics and Environmental Controls, University of Naples Federico II, P.le Tecchio 80, 80125 Naples, Italy*

[b] *Department of Computer and Systems Engineering, University of Naples Federico II, P.le Tecchio 80, 80125 Naples, Italy*

Accepted 26 June 2001

Abstract

In this paper a multilayer feed-forward neural network (NN) is used as subgrid scale (SGS) model in a large eddy simulation (LES). The NN was previously off-line trained using numerical data generated by a LES of a channel flow at $Re_\tau = 180$ with Bardina's scale similar (BFR) SGS model. Results show the ability of NNs to identify and reproduce the highly nonlinear behavior of the turbulent flows, and therefore the possibility of using NN techniques in numerical simulations of turbulent flows.

1. Introduction

Numerical simulation of turbulent flows is still one of the most challenging task in computational fluid dynamics, for difficulties involved in modeling and for computational requirements. A promising technique which became popular in the last years is the large eddy simulation (LES). While in direct numerical simulation (DNS) all length and time scales must be resolved, reducing dramatically the possibility of using this approach in flows of industrial interest, in LES only the large energy-carrying scales of motion are resolved, and the subgrid unresolved scales are modeled.

Nonetheless, in the near wall region the computational cost is still very high, due to the necessity of capturing small structures behavior in the viscous sublayer. One of the most important

* Corresponding author.
E-mail address: sarghini@unina.it (F. Sarghini).

PII: S0045-7930(01)00098-6

features of LES is no requirements of tuning coefficients like for example in Reynolds averaged Navier Stokes (RANS) equations. This is obtained through the dynamic procedure based on the filtering approach [7].

However the subgrid scale (SGS) modeling itself is quite computationally expensive, and the use of sophisticated models like the mixed dynamic scale similar Lagrangian model can increase the computational cost up to 30% [14] compared with the Smagorinsky [15] one.

The term of artificial neural networks (NNs) usually refers to any architecture that has massively parallel interconnections of simple processors. In other words, a neural net can be considered as a conveniently parameterized class of nonlinear maps.

In these years, multilayer NNs have been used for modeling static nonlinear maps with satisfactory results. Proofs were given that NNs can be used, with very good results, as a generic curve fitting and modeling technique when a relationship among different variables is not so obvious (see, for example [4–6,10,16]). Only recently, novel interconnections of NNs have been tested within dynamic contexts: namely for the identification of unknown dynamical systems described by a set of ordinary differential equations [17], or as a part of lumped parameter models.

In the computational fluid dynamics context, although NNs had been used in the past in turbulence control for drag reduction [13], and, very recently [9], in pattern recognition and turbulence interpretation in a 2D artificial numerical simulation in order to reproduce the highly nonlinear dynamics of turbulent flows, their use for SGS modeling is very innovative.

Here we present this novel hybrid approach that considers the NN as embedded in a complex Navier Stokes PDE model structure. More precisely, the NN submodel, deeply enclosed in the model equations, compute the turbulent viscosity coefficient using information on both derivatives of instantaneous velocities of the turbulent flux, and Reynolds tensor components.

A goal of the proposed modeling technique is to preserve the accuracy of the complex LESs, while saving computational time. By this way we are also able to test the validity of an hybrid modeling technique in capturing the almost chaotic behavior of a turbulent flow.

The promising results obtained by this approach open new possibilities in generating approximate boundary conditions for coupling RANS or BL equations in the near wall region and LES outside.

2. Problem formulation

In the LES approach all flow variables are decomposed into a computationally resolved, large-scale, component, and an unresolved part due to small scales, which is modeled. Separation between resolved and unresolved scales is obtained by applying a spatial filtering operation

$$\bar{f} = \int_D f(x')G(x - x'; \overline{\Delta})\,\mathrm{d}x'; \quad f' = f - \bar{f}, \tag{1}$$

where $G(x - x')$ is an appropriate filtering kernel whose characteristic length is $\overline{\Delta}$ and the integral is computed all over the domain D.

Governing equations for the large-scale variables can be derived by the application of the previous filtering operation (denoted by an overbar on the filtered variables) of characteristic width $\overline{\Delta}$ to the continuity and Navier Stokes equations. In the case of incompressible flow, they appear as

$$\begin{aligned} &\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\bar{u}_i \bar{u}_j) = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_i} + v\nabla^2 \bar{u}_i, \\ &\frac{\partial \bar{u}_i}{\partial x_i} = 0, \end{aligned} \tag{2}$$

where p is the pressure, v the kinematic viscosity, and u_1, u_2, u_3 (or u, w and v), are the streamwise, wall-normal and spanwise components of velocity, respectively.

These equations govern the dynamics of the large, energy-carrying scales of motion. The effect of the small scales upon the resolved part of turbulence appears in the SGS stress term, $\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j$, which must be modeled.

In LESs, the dissipative scales of motion are resolved poorly, sometimes not at all. The main role of the SGS model is, therefore, to remove energy from the resolved scales, mimicking the drain that is usually associated with the energy cascade. Thus, it is not necessary for a model to represent the "exact" SGS stresses accurately at each point in space and time, but only to account for their global effect.

Scale-similar models are based on the assumption that the most active SGSs are those closer to the cutoff, and that the scales with which they interact most are those right above the cutoff [1]. The "largest SGSs" can be obtained by filtering the SGS velocity $u'_i = u_i - \bar{u}_i$ to obtain

$$\overline{u'_i} = \bar{u}_i - \bar{\bar{u}}_i. \tag{3}$$

The SGS stresses can then be modeled as

$$\tau_{ij} = \overline{\bar{u}_i \bar{u}_j} - \bar{\bar{u}}_i \bar{\bar{u}}_j, \tag{4}$$

adding a Smagorinsky model to represent the dissipative effect of the small scales, gives the mixed model

$$\tau_{ij} - \frac{\delta_{ij}}{3}\tau_{kk} = \overline{\bar{u}_i \bar{u}_j} - \bar{\bar{u}}_i \bar{\bar{u}}_j - 2c_S \overline{\Delta}^2 |\overline{S}| \overline{S}_{ij}, \tag{5}$$

where $\overline{S}_{ij}$ is

$$\overline{S_{ij}} = \int_D \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) G(x - x'; \overline{\Delta})\, \mathrm{d}x', \tag{6}$$

and

$$|\overline{S}| = \sqrt{(2\overline{S}_{ij}\overline{S}_{ij})}. \tag{7}$$

The first part of the model is the scale-similar model; the Smagorinsky contribution provides the dissipation that is underestimated by the scale-similar part, and turbulent viscosity coefficient c_S is computed pointwise with a dynamic procedure.

3. The neural network model

Computing the turbulent viscosity coefficient is the core of a numerical simulation of turbulent flow and it is strictly related to the local conditions of the flow; this step of the computation in a LES can be computationally expensive, and here we exploit the possibility to reduce this computational weight by means of a NN approach.

There are two basic ideas behind the use of NNs in shaping input–output nonlinear mappings. The first is the concept of *neuron* or *node* and how neurons are connected to each other in order to work together. The second concept is the *training*, where a neural net is taught to model the relation between two sets of data.

The basic *neuron* is simply a sum of linear trasmittances followed by a univariate sigmoid function (see Fig. 1). In other words, the node function is usually a composite of the weighted sum and a differentiable nonlinear activation function, also known as the *transfer function*.

It is important to note that although NNs are commonly used as universal approximators with satisfying results, obviously the same could be achieved using other conventional technique as polynomials, trigonometric series, splines and orthogonal functions [11].

Extensive computer studies showed that NNs have more practical advantages. Their architecture results more fault tolerant, less sensitive to noise [12] and easily implementable in hardware because of the parameterization used. A partial theoretical justification for using NNs over its competitors can be also found in Barron [2,3]. A further advantage of NNs is the capability of modeling nonfully deterministic systems in which the relations between the involved quantities are complex and partially unknown, such as in turbulence.

In the proposed approach a fully connected multilayer feed-forward NN is used to model the turbulent viscosity coefficient and it is embedded in the LES code; the choice of this architecture was a first step dictated by the necessity of conjugating accuracy with the minimum computational

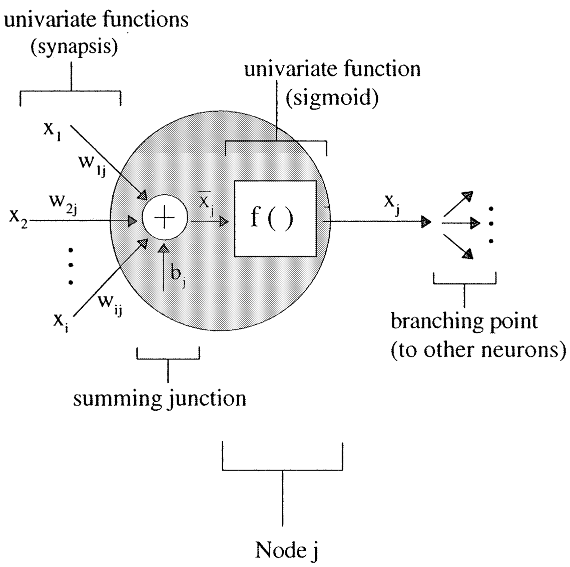

Fig. 1. Schematics of a neuron.

weight. The mathematical model of such a NN is completely defined by the number of layers, the specification of the output function for the neurons in each layer, and the weight matrices for each layer.

Different topologies as well as training techniques were tested, and a satisfactory starting point for this hybrid modeling technique is the three-layers NN, used to mimic the turbulent viscosity coefficient dynamic behavior, specified as follows (see Fig. 2):

$$x_1 = \text{logistic}(W_1 u + b_1),$$
$$x_2 = \text{logistic}(W_2 x_1 + b_2),$$
$$c_S = \text{logistic}(W_3 x_2 + b_3),$$

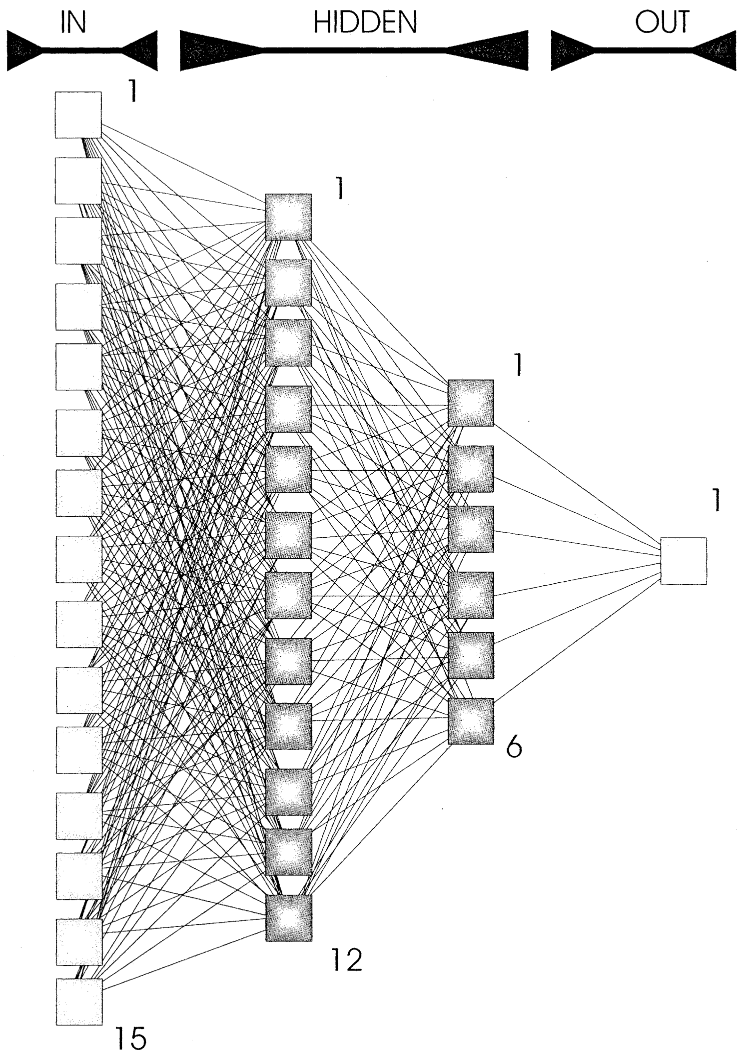

Fig. 2. Schematics of the multilayer NN.

where c_S is the turbulent viscosity coefficient, output of the net; $x_1 \in R^{12}$ and $x_2 \in R^6$ are the output vectors of the first and second hidden layer, respectively; $W_1 \in R^{15\times 12}$, $W_2 \in R^{12\times 6}$ and $W_3 \in R^{6\times 1}$ are the weight matrices; $b_1 \in R^{12}$, $b_2 \in R^6$ and $b_3 \in R^1$ are the bias column vectors; the logistic $(\cdot)$ operator is intended componentwise and u is the input vector of the neural net defined as $u = (u_{,x}, u_{,y}, u_{,z}, v_{,x}, v_{,y}, v_{,z}, w_{,x}, w_{,y}, w_{,z}, u'u', v'v', w'w', u'v', u'w', v'w')^{\mathrm{T}} \in R^{15}$; $(\ )_{,x}, (\ )_{,y}, (\ )_{,z}$ are the derivatives of streamwise, wall-normal and spanwise components of velocity, and u', v', w' are the instantaneous fluctuations respect the global means.

The choice of input vector was dictated by the idea of introducing all the variables involved in the calculation of the original SGS model, and to let the NN to perform a sensitivity analysis and a proper pruning of the unnecessary inputs; in some way, this choice is linked to the idea that the NN should be able to self organize the inputs recognizing a common pattern when it is present.

4. Identification and validation

As already mentioned, in LES the NN is a nonlinear mapping between the input and output quantities of the subgrid turbulent coefficient. The NN contains a certain number of parameters (weights W and biases b), whose final values have to be identified during the training period. In our context a learning rule is used to adjust the weights and the biases of the NN so as to minimize the summed-squared-error (SSE) between the outputs of the neural model and the measured data

$$\min_{W,b} \int_{t_{\mathrm{in}}}^{t_{\mathrm{fin}}} (y - \hat{y})^2 \, \mathrm{d}t, \tag{8}$$

where y is an *experimental* output which, in our case, is obtained by running the complete LES code with Bardina's scale similar model inside; $\hat{y}$ is the corresponding NN output; t_{in} and t_{fin} are respectively the initial and final time instant so that $(t_{\mathrm{fin}} - t_{\mathrm{in}})$ defines a time duration corresponding to 2 large eddy *turnover times*.

Assuming as a characteristic parameter of the turbulent activity for this type of flow the dimensionless distance from the wall y^+, defined as $y^+ = (u_\tau y)/\nu$, where $u_\tau = (\tau_{\mathrm{w}}/\rho)^{1/2}$, is the friction velocity and $\tau_{\mathrm{w}} = \mu \, \mathrm{d}\overline{U}/\mathrm{d}y\big|_{\mathrm{w}}$ is the shear stress at the wall, the NN training is performed on a predefined set of points which are function of the wall distance itself. Sampling points, one for each computational layer, are chosen in the middle of the channel in order to avoid boundary effects, and for each point about 5500 samples are used for training and validation, according to a windowing procedure: the first 3000 and the last 1500 are used for training, while the remaining intermediate 1000 are used for validation.

The Bardina's SGS model generating the training and the validation sets was chosen because of its feature of providing a localized characterization of the turbulent flow field. Although this model does not provide enough subgrid dissipation, in this investigation it is a useful candidate in order to explore the capacity of NNs to detect dynamics nonlinearities in turbulent flows. In this case, the multilayer feed-forward network could be used as a simplified instantaneous nonlinear approximator.

In general, scale similar models employ multiple filtering operations to identify the smallest resolved scales, which have been shown to be the most active in the interaction with the unresolved SGSs. They do not assume that the principal axes of the strain-rate tensor are aligned with those of the SGS stress tensor, and allow the explicit calculation of the SGS energy. They can provide backscatter in a numerically stable and physically realistic manner, and predict SGS stresses that are well correlated with the locations where large Reynolds stresses occur.

Notice that the Bardina's model, which is substituted by the NN submodel in our 'hybrid' LES, is a complex, highly nonlinear dynamical model introducing in the 'dynamics procedure' [8] a relative large number of matrix contractions and filtering operations on different scales in order to extract information on the SGS components close to the 'cut-off frequency region'.

The multilayer feed-forward NN was off-line trained through a backpropagation with weight decay technique [18]. This algorithm allows to decrease the weights of the links while training them with backpropagation. Moreover, in addition to each update of a weight, the weight is decreased by a part of its old value. The effect is similar to the pruning algorithms (which try to make NNs smaller by pruning unnecessary links or units, thus giving evidence of the relevance of input values) so that weights are driven to zero unless reinforced by backpropagation. The NN was initialized by a randomized weights technique which sets all weights and the bias with distributed random values.

5. Training and validation results

In Fig. 3, as an example, normalized c_S coefficient is shown at different y^+, highlighting the good results obtained in both training and validation data set. The solid and the dashed line refer respectively to behavior of the Bardina's model (BFR) and the NN output.

We want to highlight that the NN converge to the prefixed error threshold within a very small number of algorithm iterations (<500) for set points near the channel wall, while it takes even two

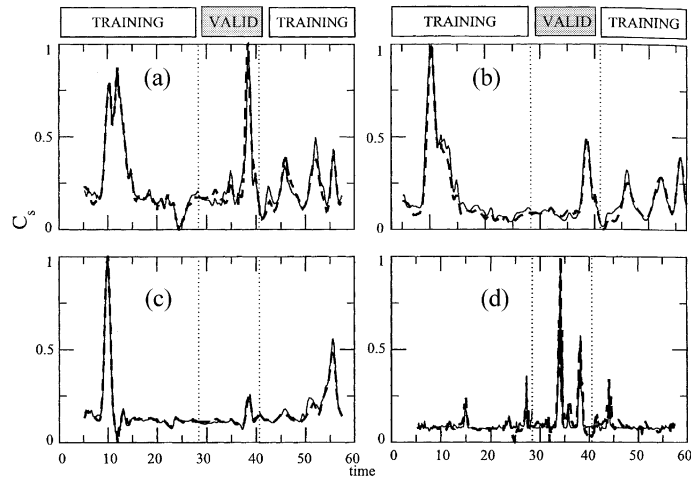

Fig. 3. Training and validation results; normalized c_S coefficient at $y^+ = 0.4$ (a), 4 (b), 13 (c), 56 (d). (- - - -) NN, (—) BFR.

orders of magnitude more to give the results in points near the middle of the channel. An accurate analysis of these results showed that this behavior was caused probably by the presence of spurious numerical peaks computed in the validation/training signals by the Bardina's model in the region positioned in the middle of the channel. A further reduction of the training cycles could be obtained with a more complete and accurate investigation in order to optimize the network configuration and the training algorithm, but it is out of the purpose of the present investigations.

6. Results and discussion

A spectral Chebyshev–Chebyshev–Fourier code was used for the numerical test, with $96 \times 64 \times 64$ points in the streamwise, spanwise and wall-normal directions; the NN-hybrid LES was performed on a statistically 2D turbulent channel flow, and the Reynolds number based on the wall friction velocità was $Re_\tau = 180$. This resolution was already tested for the same type of flow and Reynolds number in other papers. A *priori* and *a posteriori* test were performed in order to investigate the behavior on NNs in a hybrid LES.

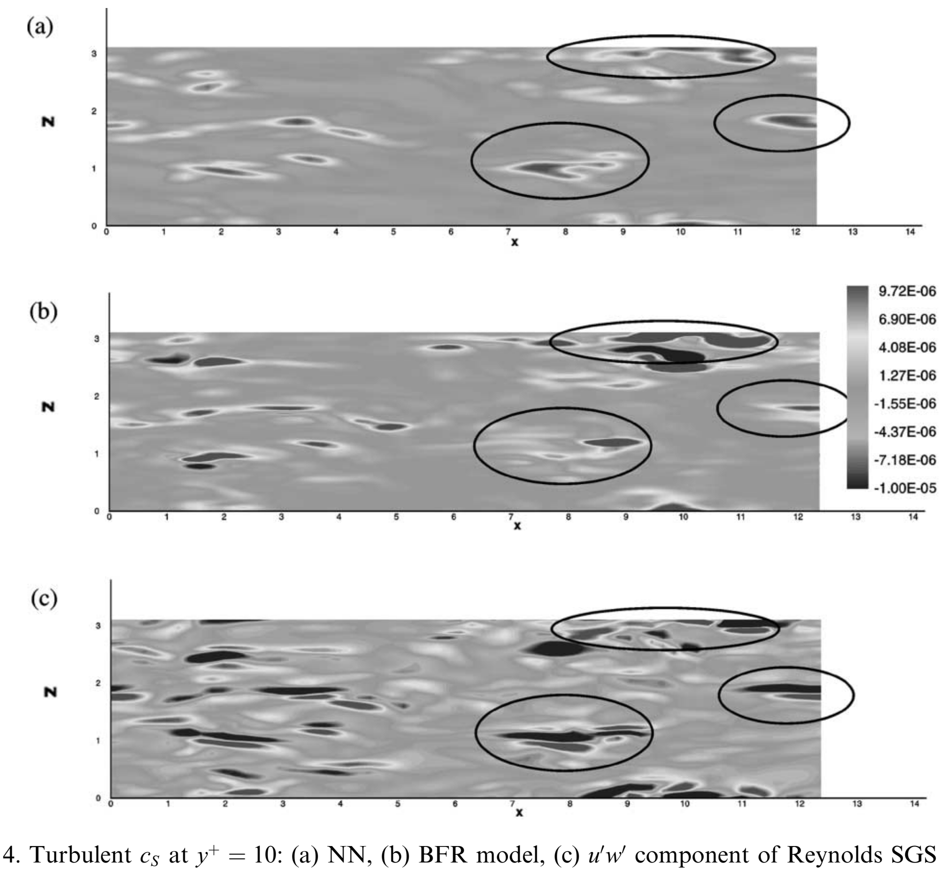

Fig. 4. Turbulent c_S at $y^+ = 10$: (a) NN, (b) BFR model, (c) $u'w'$ component of Reynolds SGS tensor.

In the *a priori* case, starting from the same velocity field, turbulent viscosity coefficient c_S was obtained both by using the complete dynamic procedure and by using NNs and results were compared. In Fig. 4(a) and (b) a comparison of instantaneous c_S coefficient at $y^+ = 10$ is shown, while in Fig. 4(c) the most active component $u'w'$ of Reynolds SGS tensor is reported as an indicator of turbulent activity. NN simulation predicts turbulence intensities in good agreement with the Bardina's model, and local features of the flow, highlighted by the circles, are preserved.

The same accuracy is preserved in the wall region, buffer region and log region, although some spurious peaks are present in the core region of the channel ($y^+ > 100$), probably due to numerical noise in the training signal, as showed by the increase of training cycles required to reduce the SSE error under the assigned level.

This spurious numerical noise is a known characteristics of many models, requiring often filtering procedures; indeed, those spurious peaks are present also in turbulent numerical simulations performed with the original BFR model, which is used in conjunction with a local or global averaging technique. On the other side, it is also possible that the different physical structure of the core region requires a different topology of the NN, and this hypothesis requires further investigation. In the *a posteriori* test these peaks are clipped, in order to stabilize the computation, and starting from $y^+ > 100$ a plane averaged $c_S(z)$ coefficient instead of $c_S(x, y, z)$ is used. Anyway, by using NNs, the computational time is reduced of about 20%.

In Fig. 5 the streamwise-wall normal component $-2c_S\overline{S_{12}}$ of the SGS eddy viscosity is computed, showing a very good agreement between the one computed by NNs and the BFR one.

In Fig. 6 the asymptotic behavior of the turbulent quantity $c_S\Delta^2$ in the near wall region is plotted, where Δ is the filtering length, showing a correct proportionality to y^{+3}.

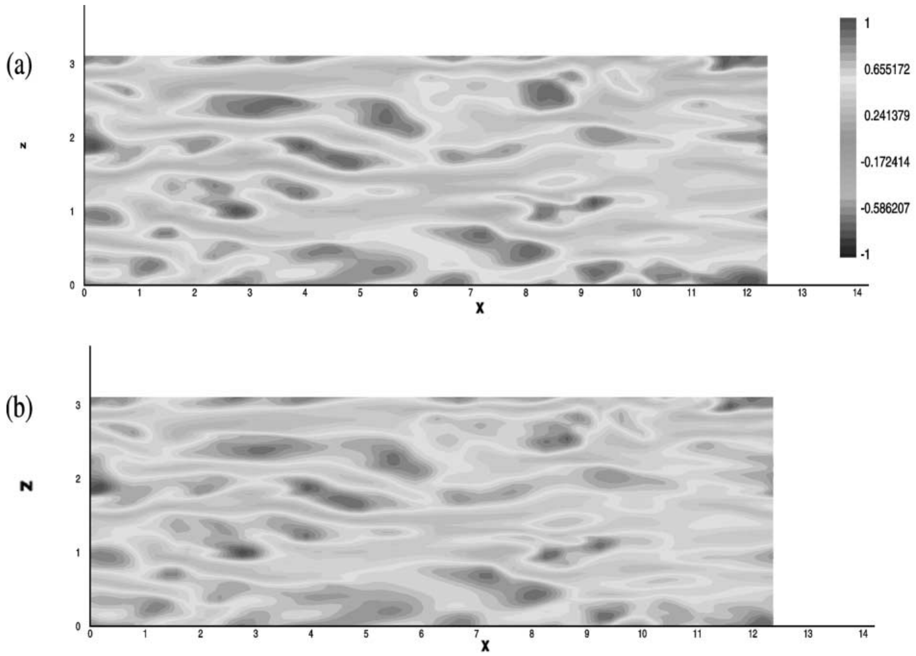

Fig. 5. $-2c_S\overline{S}_{12}$ at $y^+ = 10$; (a) NN, (b) BFR model.

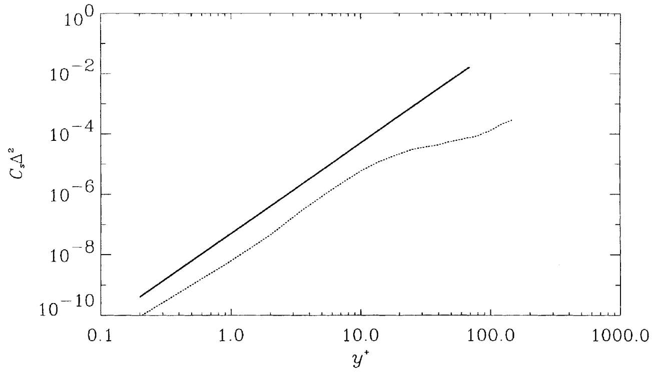

Fig. 6. Asymptotic behaviour of $c_S\Delta^2$ in the near wall region: (—) $\propto y^{+3}$; (----) NN Model.

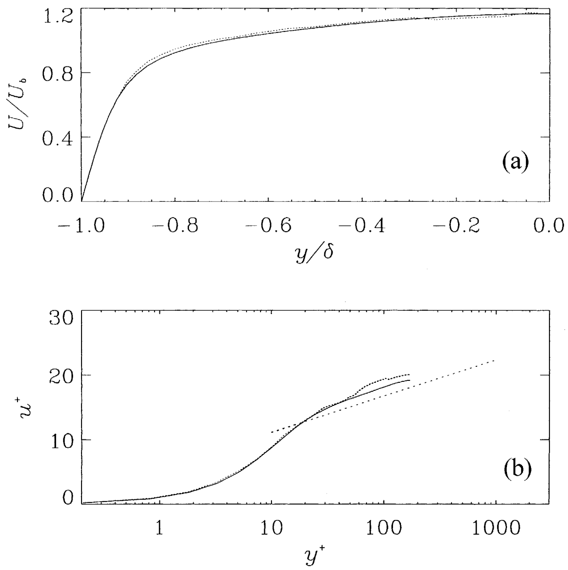

Fig. 7. Mean velocity profiles in the 2D plane channel at $Re_\tau = 180$; (—) BFR model; (----) hybrid NN model.

Mean velocity profiles in nondimensional distance from the centerline and in wall units are plotted in Fig. 7, while turbulence intensity profiles are plotted in Fig. 8.

7. Concluding remarks

In this paper the possibility of using NNs for SGS modeling in LES is shown, obtaining a computational time saving of about 20%, although the computational time saving was computed not considering the training time.

Although the reduction of the computational cost could appear as a main target of the of the proposed investigation, the more important result is the possibility to use NNs techniques

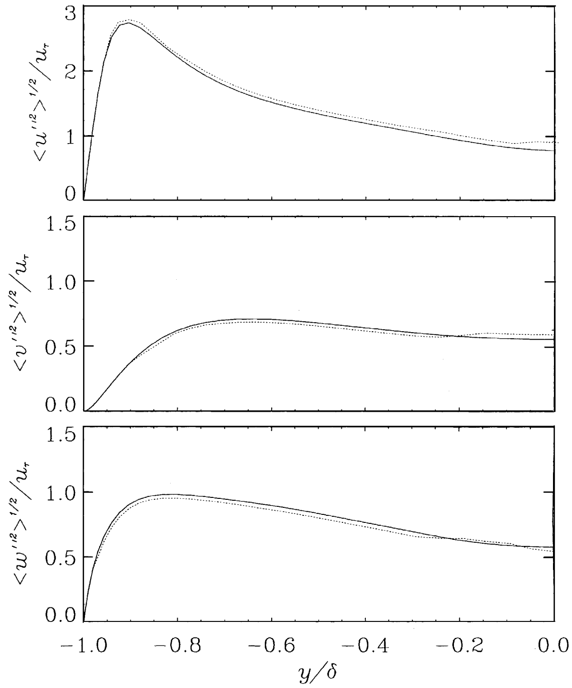

Fig. 8. Turbulence intensity profiles in the 2D plane channel at $Re_\tau = 180$; (—) BFR model; (- - - -) hybrid NN model.

properly trained by LES simulations to detect turbulence dynamics. As a consequence, it also opens new possibilities of use of the NN technology, like for example the possibility to generate approximate boundary conditions in coupling LES in the free turbulence region and RANS techniques in the near wall region to further reduce computational requirements, or to detect turbulent patterns and coherent structures in order to develop turbulence control strategies.

Generalization of the algorithm is a work in progress, as well as a better choice of the training algorithm: it requires further investigations and it is out of the purpose of this work. At the moment, the algorithm seems to work for a certain range of the Reynolds number respect to the training Reynolds number, but, for example, switching to $Re_\tau = 1050$, at the moment it requires a novel training.

Using more sophisticated SGS models for training, like the mixed scale similar model with Lagrangian averaging on pathlines will probably require different NN configurations. Although some good and promising results are already obtained, some key points must still be addressed, like global spatial scaling of the input signals, the choice of training sets of sampling points, and the memory effect due to space–time correlation of turbulent signals.

References

[1] Bardina J, Ferziger JH, Reynolds WC. AIAA Paper No. 80–1357, 1980.
[2] Barron AR. Proceedings of the Seventh Yale Workshop on Adaptative and Learning Syst. 1992. p. 68–72.

[3] Barron R. IEEE Trans Inform Theory 1993;39:930–45.
[4] Chen S, Cowan CFN, Grant PM. IEEE Trans Neur Networks 1991;2:302–9.
[5] Cybenko G. Mathematics Contr Signals and Syst 1989;2:303–14.
[6] Funahashi K. Neural Networks 1989;2:183–92.
[7] Germano M. J Fluid Mech 1992;238:235.
[8] Germano M, Piomelli U, Moin P, Cabot WH. Phys Fluids A 1991;3:1760.
[9] Giralt F, Arenas A, Ferre-Gine J, Rallo R, Kopp GA. Phys Fluids 2000;12:1826.
[10] Hornik K, Stinchcombe M, White H. Neural Networks 1989;2:359–66.
[11] Glielmo L, Santini S, Milano M. IEEE/ASME Transaction on Mechatronics 2000;2:132–41.
[12] Haykin S. Neural networks. 3rd ed. NewYork: McMillan; 1994.
[13] Lee C, Kim J, Babcock D, Goodman R. Phys Fluids 1997;9:1740.
[14] Sarghini F, Piomelli U, Balaras E. Phys Fluids 1999;6:1596.
[15] Smagorinsky. Mon Weather Rev 1963;91:99.
[16] Wang Z, Tham MT, Morris E. Int Journal Control 1992;56:655.
[17] Wang W, Lin C. IEEE Trans on Neural Networks 1998;9:294–307.
[18] Werbos P. Proceedings of the IEEE International Conference on Neural Networks, IEEE Press, 1988. p. 343–53.