

Implementation of MHD model equations in CFX 4.3

Ola Widlund
Faxén Laboratory
Royal Institute of Technology
Stockholm, Sweden

TRITA-MEK 2000:10

November, 2000

Abstract

This report documents an implementation of magnetohydrodynamic (MHD) model equations in the commercial flow solver *CFX* 4.3. The implementation assumes a DC external magnetic field, which can be specified in user fortran, or read from a file. The code solves the Poisson equation for mean electrostatic potential. Great care has been taken to make the calculation of total electric current densities, and sources for the potential equations, accurate even on arbitrary non-orthogonal body-fitted multiblock grids. For modeling the effect of the magnetic field on turbulence, the implementation uses the MHD turbulence model proposed by Widlund *et al.* Both a full Reynolds stress model, and a three-equation model of $K-\varepsilon$ type are available. Commercial symbolic mathematics software was used to automate linearization, optimization and code generation for the often complex source terms in the turbulence model equations. The properties of the model implementation are demonstrated in a simulation of a 3D channel flow with “M”-shaped velocity profiles. The implementation of the new MHD turbulence models appear to be well-behaved and numerically robust. The MHD model implementation can be used together with standard *CFX* modeling options, such as heat and mass transfer, multiphase and multi-component flows.

Contents

1	Introduction	2
2	Theory and model equations	3
2.1	Governing equations	3
2.2	Reynolds averaged equations	4
2.3	Mean electromagnetic equations	4
2.4	Conventional turbulence models	5
2.4.1	Second moment RST model	5
2.4.2	Eddy viscosity models	6
2.5	Models for MHD turbulence	7
3	Model implementation	9
3.1	Electric potential equation	9
3.2	Total current density and Lorentz force	10
3.3	Dimensionality anisotropy	13
3.4	Turbulence equations	14
4	Results	14
4.1	Calculation of currents on non-orthogonal grids	14
4.2	Channel flow with “M” profiles	15
5	Discussion and outlook	22
A	Using the MHD model in <i>CFX</i>	26
A.1	<i>CFX</i> options in the command file	26
A.2	MHD model options	29
B	Documentation of fortran source code	30
B.1	MHDINT	31
B.2	MHDBXT	31
B.3	USRDIF	34
B.4	USRWTM	34
B.5	MHDSRC	34
B.6	MHDJSF	35
B.7	MHDSEP	35
B.8	MHDBF	35
B.9	MHDJSC	36
B.10	MHDGEP and MHDGE2	36
C	MAPLE script for generating source terms	37
D	Listing of sample command file	40

1 Introduction

This report documents the implementation of mean electromagnetic equations and magnetohydrodynamic (MHD) turbulence models in the commercial flow solver *CFX* 4.3.

The featured MHD turbulence models were developed by Widlund *et al.* [8, 7, 9] for modeling of Joule dissipation in simulations of continuous steel casting applications. The modeling approach is based on the formulation of a transport equation for a length scale anisotropy variable, α . The novel model can be formulated either as a full Reynolds stress transport (RST) model, or as a three-equation K - ε - α model.

As for the mean equations, the external magnetic field is assumed to be steady, so that we need to solve only for the mean scalar electrostatic potential. We further assume the magnetic Reynolds number is low, so that the so-called quasistatic, or inductionless approximation is valid.

There are several motives for this *CFX* implementation. First of all, we want to demonstrate that the new MHD turbulence models can be implemented in a commercial flow solver, and that the models are numerically robust and efficient enough to be useful for engineering MHD applications. Second, a complete 3D solver for complex geometries will be very useful in the work with validation of the new turbulence models. Finally, this work shows how existing in-house implementations of the mean equations in older versions of the code can be updated to work with complex multi-block non-orthogonal grids in *CFX* 4.3.

The following list summarizes the features of the MHD model implementation:

- User scalar equations for mean electrostatic potential EPOT (Φ), and turbulence dimensionality anisotropy ALPHA (α).
- Joule dissipation terms in both Reynolds stress, K , and ε equations, so that either the full RST MHD closure or the reduced three-equation K - ε - α closure can be used. Automatic detection of model type in user fortran.
- The mean strain terms in the α equation can be disabled.
- Mean electrostatic potential and total electric current densities are computed accurately even on multiblock non-orthogonal grids.
- The external magnetic field can be specified as constant and uniform in the command file, read from a separate text file, or specified in user fortran.
- Total current density and Lorentz force components are stored in user scalars for post-processing.
- Under-relaxation of Lorentz forces in the mean momentum equations (supported by standard command file options in *CFX* 4.3).
- Walls can be either non-conducting, or given a fixed electric potential.
- The electrical boundary conditions in flow inlets and outlets can be modified with options in the command file.
- The location of a zero reference point for electric potential can be specified in the command file.

- A number of MHD model options and model parameters are read from the end of the *CFX* command file when a simulation is started. This reduces the need for recompilation of the fortran source code.

The report is structured as follows: Section 2 introduces the governing equations, and the model equations. Section 3 describes how the model equations are discretized and implemented in the code. Some of the results obtained with the code so far are described in Section 4. Section 5 contains discussion and conclusions. Appendix A describes the various *CFX* and specific MHD model options specified in the *CFX* command file, while Appendix B provides details about the fortran routines used in the implementation. Appendix C lists a MAPLE script for linearization and coding of source terms, and App. D is a listing of a sample command file.

2 Theory and model equations

2.1 Governing equations

With spatial coordinates \mathbf{x} and time t , the Navier–Stokes equations for the instantaneous velocity field $u_i(\mathbf{x}, t)$ of an incompressible fluid are

$$\frac{\partial \rho u_i}{\partial t} + u_j \frac{\partial \rho u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \frac{\partial u_i}{\partial x_j} \right) + f_i, \quad (1)$$

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (2)$$

where $p(\mathbf{x}, t)$ is the pressure field, μ is the (molecular) dynamic viscosity¹, and f_i is a body force (per unit volume). In the MHD problem at hand, the body force f_i in (1) corresponds to the magnetic Lorentz force.

The fluid is subjected to an external magnetic field, with magnetic flux density \mathbf{B} (here in vector notation). The fluid flow will induce electrical currents in the fluid, and these in turn give rise to an induced magnetic field, which can be regarded as a perturbation of the external field. The relative size of this perturbation is governed by the dimensionless magnetic Reynolds number

$$\text{Rm}_L \equiv \eta \sigma u L, \quad (3)$$

where η is magnetic permeability, σ electrical conductivity, and u and L are characteristic velocity and length scales, respectively. In most industrial and laboratory flows, the magnetic Reynolds number is very low, $\text{Rm}_L \ll 1$. The flow-induced magnetic field perturbations are then negligible beside the external field. Using this so-called quasistatic approximation, the non-relativistic Maxwell equations yield

$$\mathbf{f} \approx \mathbf{j} \times \mathbf{B}. \quad (4)$$

The instantaneous electric current \mathbf{j} is given by Ohm's law,

$$\mathbf{j} \approx \sigma(\mathbf{e} + \mathbf{u} \times \mathbf{B}), \quad (5)$$

where \mathbf{e} is the electric field. The scalar electrostatic potential ϕ is defined by

$$\mathbf{e} = -\nabla \phi. \quad (6)$$

¹Not to be confused with the Joule dissipation μ introduced later

In the absence of free charges, the current field is divergence free. If the conductivity σ is assumed constant, the divergence of Ohm's law (5), together with (6), therefore yields

$$\nabla^2\phi \approx \nabla \cdot (\mathbf{u} \times \mathbf{B}). \quad (7)$$

2.2 Reynolds averaged equations

A Reynolds decomposition is applied to split the instantaneous variables into mean and fluctuating parts. The mean of a quantity v is here defined as the ensemble average over an infinite number of realizations v_n ,

$$\langle v \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N v_n. \quad (8)$$

We use capital letters for mean values, and primes to denote fluctuating parts. The velocity is, for example, expressed as the sum $U_i + u'_i$, where U_i is the mean velocity, and u'_i is the velocity fluctuations. By definition the mean value of the fluctuation is zero.

Ensemble averaging of (1) and (2) gives the Reynolds averaged Navier–Stokes (RANS) equations, or the mean momentum equations,

$$\frac{\partial \rho U_i}{\partial t} + U_j \frac{\partial \rho U_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \frac{\partial U_i}{\partial x_j} + \rho \langle u'_i u'_j \rangle \right) + F_i, \quad (9)$$

$$\frac{\partial U_i}{\partial x_i} = 0. \quad (10)$$

Here F_i is the mean Lorentz force; see Sec. 2.3 below.

After Reynolds averaging, the transport equation for the mean Q of a passive scalar $q = Q + q'$ is

$$\frac{\partial \rho Q}{\partial t} + U_j \frac{\partial \rho Q}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\Gamma_Q \frac{\partial Q}{\partial x_j} + \rho \langle u'_i q' \rangle \right) + S_Q, \quad (11)$$

where S_Q is a volume source of the scalar, and Γ_Q is the molecular diffusivity of the scalar.

Turbulent velocity fluctuations contribute to the transport of the mean quantities through the turbulent Reynolds stresses $\rho \langle u'_i u'_j \rangle$ in (9), and the turbulent Reynolds flux $\rho \langle u'_i q' \rangle$ in (11), respectively. These turbulent correlations must be provided by a turbulence model of some kind (see Sec. 2.4).

2.3 Mean electromagnetic equations

We perform a Reynolds decomposition also of the Maxwell equations (4) through (7). The mean Lorentz force F_i in (9) is thus given by

$$\mathbf{F} = \mathbf{J} \times \mathbf{B}. \quad (12)$$

The mean electric current density is

$$\mathbf{J} = \sigma(-\nabla\Phi + \mathbf{U} \times \mathbf{B}). \quad (13)$$

The divergence of (13), together with the continuity condition $\nabla \cdot \mathbf{J} = 0$, yields

$$\nabla \cdot (\sigma \nabla \Phi) = \nabla \cdot (\sigma \mathbf{U} \times \mathbf{B}). \quad (14)$$

For constant conductivity σ , this reduces to the familiar Poisson equation

$$\nabla^2 \Phi = \nabla \cdot (\mathbf{U} \times \mathbf{B}). \quad (15)$$

The Lorentz force \mathbf{F} can be computed, if we first solve (14) with appropriate boundary conditions. If \mathbf{n} is an inward unit normal at the boundary, (13) yields the Neumann boundary condition

$$\frac{\partial \Phi}{\partial n} = -\frac{J_n}{\sigma} + \mathbf{n} \cdot (\mathbf{U} \times \mathbf{B}), \quad (16)$$

where $J_n = \mathbf{J} \cdot \mathbf{n}$ is the current into the domain. A non-conducting wall has zero normal current at the wall, $J_n = 0$. For a stationary wall with no-slip velocity boundary conditions, the boundary condition for Φ is then simply

$$\frac{\partial \Phi}{\partial n} = 0. \quad (17)$$

The formal solution to (14) with only Neumann boundary conditions contains an undetermined constant term. The electric potential must therefore be specified explicitly in at least one point in the domain. Alternatively, a fixed electric potential can be specified at one or more walls.

2.4 Conventional turbulence models

The complete set of transport equations must be closed by providing models for the Reynolds stress and flux terms $\rho \langle u'_i u'_j \rangle$ and $\rho \langle u'_i q' \rangle$ appearing in the mean momentum and scalar equations (9) and (11). Common models can be put into two broad classes; eddy viscosity models, and Reynolds stress transport (RST) models. The most common eddy viscosity model is the K - ε model.

2.4.1 Second moment RST model

A full Reynolds stress closure features transport equations for each of the six independent components of the Reynolds stress tensor $R_{ij} \equiv \langle u'_i u'_j \rangle$, complemented by a transport equation for scalar viscous dissipation, ε . The turbulent transport in the mean momentum equation (9) is thus accounted for without further modeling.

The Reynolds flux term in the scalar equation is usually modeled as

$$-\langle u'_i q' \rangle = \frac{C_s K}{\sigma_Q \varepsilon} R_{ij} \frac{\partial Q}{\partial x_j}, \quad (18)$$

where C_s is a model constant, and σ_Q is the turbulent Prandtl number for the scalar [1]. The use of the Reynolds stresses in this model makes the turbulent diffusion anisotropic. The transport equation (11) for the mean Q of a passive scalar becomes

$$\frac{\partial \rho Q}{\partial t} + U_j \frac{\partial \rho Q}{\partial x_j} - \frac{\partial}{\partial x_i} \left(\left[\Gamma_Q \delta_{ij} + \rho \frac{C_s K}{\sigma_Q \varepsilon} R_{ij} \right] \frac{\partial Q}{\partial x_j} \right) = S_Q. \quad (19)$$

If we neglect molecular diffusion, the turbulence model equations are

$$\frac{\partial \rho R_{ij}}{\partial t} + U_k \frac{\partial \rho R_{ij}}{\partial x_k} - \frac{\partial}{\partial x_k} \left(\rho \frac{C_s}{\sigma_{RS}} \frac{K}{\varepsilon} R_{kl} \frac{\partial R_{ij}}{\partial x_l} \right) = \rho \left(P_{ij} + \Pi_{ij}^{(s)} + \Pi_{ij}^{(r)} - \varepsilon_{ij} \right) + S_{ij}^{RS}, \quad (20)$$

$$\frac{\partial \rho \varepsilon}{\partial t} + U_j \frac{\partial \rho \varepsilon}{\partial x_j} - \frac{\partial}{\partial x_j} \left(\rho \frac{C_s}{\sigma_\varepsilon} \frac{K}{\varepsilon} R_{jk} \frac{\partial \varepsilon}{\partial x_k} \right) = \rho \left(C_{\varepsilon 1} \frac{\varepsilon}{K} P_{ii} / 2 - C_{\varepsilon 2} \frac{\varepsilon^2}{K} \right) + S_\varepsilon. \quad (21)$$

In the Reynolds stress equation (20), only the production term P_{ij} is exact, and all other terms require modeling. With the most common model (Launder, Reece & Rodi [3]), we have

$$P_{ij} \equiv -R_{ik} \frac{\partial U_j}{\partial x_k} - R_{jk} \frac{\partial U_i}{\partial x_k}, \quad (22)$$

$$\Pi_{ij}^{(s)} = -C_1 \frac{\varepsilon}{K} \left(R_{ij} - \frac{2}{3} K \delta_{ij} \right), \quad (23)$$

$$\Pi_{ij}^{(r)} = -\gamma \left(P_{ij} - \frac{2}{3} P_{kk} \delta_{ij} \right), \quad (24)$$

$$\varepsilon_{ij} = \frac{2}{3} \varepsilon \delta_{ij}. \quad (25)$$

Here (23) and (24) represent slow and rapid pressure-strain, while (25) is viscous dissipation. S_{ij}^{RS} is a source term due to the body force in the Navier-Stokes equations. The first two model terms in the right-hand side of the ε equation (21) represent production and viscous dissipation, respectively, while S_ε is a source term. The Reynolds stress model above has model constants $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, C_1 , γ , C_s , σ_{RS} , and σ_ε .

Wall boundary conditions for the discretized equations are generally applied using wall functions. The mean velocity and the value of ε are then prescribed some distance from the wall, within the logarithmic region of the boundary layer. The Reynolds stress equations are solved assuming the wall-normal gradient is zero, and the source terms are evaluated in a way consistent with the wall law.

2.4.2 Eddy viscosity models

In eddy viscosity models, the Reynolds stresses are modeled using the so-called Boussinesq hypothesis, relating Reynolds stress anisotropy to the mean strain tensor:

$$-\rho \langle u'_i u'_j \rangle = -\frac{2}{3} \rho K \delta_{ij} + \mu_T S_{ij} \quad (26)$$

where μ_T is the turbulent viscosity, $K \equiv \langle u'_i u'_i \rangle / 2$ is the turbulent kinetic energy, and

$$S_{ij} \equiv \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad (27)$$

is the mean strain tensor.

The two-equation K - ε model contains transport equations for the turbulent kinetic energy K , and the viscous dissipation rate ε . The turbulent viscosity in (26) is given by a constitutive relationship,

$$\mu_T = C_\mu \rho \frac{K^2}{\varepsilon}, \quad (28)$$

where C_μ is a model constant.

If (26) is inserted in (9), the mean momentum equation becomes

$$\frac{\partial \rho U_i}{\partial t} + U_j \frac{\partial \rho U_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left([\mu + \mu_T] \frac{\partial U_i}{\partial x_j} \right) + F_i. \quad (29)$$

In analogy with (26), we have the eddy diffusivity hypothesis which expresses the Reynolds flux as

$$-\rho \langle u_i' q' \rangle = \frac{\mu_T}{\sigma_Q} \frac{\partial Q}{\partial x_i}, \quad (30)$$

where σ_Q is a turbulent Prandtl number. The mean scalar transport equation then takes the form

$$\frac{\partial \rho Q}{\partial t} + U_j \frac{\partial \rho Q}{\partial x_j} - \frac{\partial}{\partial x_j} \left(\left[\Gamma_Q + \frac{\mu_T}{\sigma_Q} \right] \frac{\partial Q}{\partial x_j} \right) = S_Q. \quad (31)$$

Neglecting molecular diffusion, the model transport equations for K and ε are (Launder & Spalding, [4])

$$\frac{\partial \rho K}{\partial t} + U_j \frac{\partial \rho K}{\partial x_j} - \frac{\partial}{\partial x_j} \left(\frac{\mu_T}{\sigma_K} \frac{\partial K}{\partial x_j} \right) = \rho (P_K - \varepsilon) + S_K, \quad (32)$$

$$\frac{\partial \rho \varepsilon}{\partial t} + U_j \frac{\partial \rho \varepsilon}{\partial x_j} - \frac{\partial}{\partial x_j} \left(\frac{\mu_T}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_j} \right) = \rho \left(C_{\varepsilon 1} \frac{\varepsilon}{K} P_K - C_{\varepsilon 2} \frac{\varepsilon^2}{K} \right) + S_\varepsilon. \quad (33)$$

Here P_K is the turbulence production, which is modeled using the eddy viscosity hypothesis (26),

$$P_K \equiv \frac{P_{ii}}{2} \approx \frac{\mu_T}{\rho} S_{ij} S_{ji}. \quad (34)$$

S_K and S_ε are source terms due to the body force in the Navier–Stokes equations (1). C_μ , $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, σ_K and σ_ε are model constants.

In the standard high-Reynolds number version of the K - ε model, the treatment of wall boundary conditions is the same as for the Reynolds stress model (see above).

2.5 Models for MHD turbulence

The MHD turbulence models proposed by Widlund *et al.* [8, 7, 9] extend the conventional turbulence closures with a transport equation for a dimensionality anisotropy variable, α . The new scalar is then used to model Joule dissipation source (sink) terms for the K , ε and Reynolds stress equations. The model concept can be used either as a three equation K - ε - α model, or as a full Reynolds stress model.

The dimensionality anisotropy variable is defined in terms of the component of the Reynolds dimensionality tensor, Y_{ij} , which is parallel with the magnetic field. If n_i is a unit vector parallel with the magnetic field,

$$\alpha \equiv \frac{n_i n_j Y_{ji}}{2K}. \quad (35)$$

With this definition, the expression for scalar Joule dissipation of turbulent kinetic energy is exact for homogeneous turbulence,

$$\mu \equiv 2 \frac{\sigma B^2}{\rho} \alpha K. \quad (36)$$

The dimensionality tensor \mathbf{Y} was first introduced by Reynolds and co-workers to improve modeling of rapidly rotating turbulence [5]. Like the Reynolds stress tensor \mathbf{R} , \mathbf{Y} is a symmetric tensor with trace $2K$, so that $0 \leq \alpha \leq 1$. For isotropic turbulence, $\alpha = 1/3$. The effect of a magnetic field is to decrease α towards the limit of 2D turbulence, where $\alpha = 0$. A low value of α means that the turbulent structures are very long in the direction of the magnetic field. In contrast, a value close to unity means that the structures are flat, and short in the direction of the field.

An exact equation for α can be derived from the Navier–Stokes equations [9], but all terms in the equation require modeling. The model equation for α implemented in this work can be written on the generic form (11), with a source term

$$S_\alpha / \rho = P_\alpha + \Pi_\alpha^{(r)} + \pi_\alpha - \mu_\alpha, \quad (37)$$

where

$$P_\alpha = -2G \frac{R_{ij} S_{ji}}{K} - 2H \frac{n_i n_j R_{jk} S_{ki}}{K} + \alpha \frac{R_{ij} S_{ji}}{K}, \quad (38)$$

$$\Pi_\alpha^{(r)} = 2\alpha \left(-\frac{3}{2}\alpha^2 + \frac{29}{10}\alpha - \frac{7}{5} \right) n_i n_j S_{ji}, \quad (39)$$

$$\pi_\alpha = C_{\alpha 2} \frac{\varepsilon}{K} \left(\frac{1}{3} - \alpha \right), \quad (40)$$

$$\mu_\alpha = 2C_{\alpha 1} \frac{\sigma B^2}{\rho} \alpha^2. \quad (41)$$

The first two terms, (38) and (39), represent the effect of mean strain (production and rapid pressure–strain, respectively). π_α is a return-to-isotropy term, due to non-linear energy transfer, while μ_α is the Joule destruction term. S_{ij} is the mean strain tensor, and G and H are model functions defined below.

The modeling of the mean strain terms P_α and $\Pi_\alpha^{(r)}$ relies on the assumption that the full dimensionality tensor \mathbf{Y} is axisymmetric about the magnetic field (see [9]). This is probably a rather poor assumption in regions where the magnetic field is weak, but there the value of α is not important. Wherever magnetic effects are strong, the strain terms are likely to play little role; considering their algebraic complexity they can probably be neglected in most cases, to save computation time.

The current version of the MHD model assumes the underlying conventional turbulence model is of high-Reynolds number type, with wall function boundary conditions. The wall boundary conditions for the discretized α equation are

defined in analogy with those normally used for K and the Reynolds stresses. The equation is dominated by a balance of its source terms near walls, and transport is negligible. We therefore assume zero normal gradient in the first cell ($d\alpha/dn = 0$), and evaluate the source terms as usual. There is currently no low-Re version of the MHD model.

Magnetic destruction terms for K , ε and the Reynolds stresses are put in the source terms of the respective equations (32), (33)/(21), and (20). We have (see [8, 9])

$$S_K/\rho = -\mu = -2\frac{\sigma B^2}{\rho}\alpha K, \quad (42)$$

$$S_\varepsilon/\rho = -\mu_\varepsilon = -2C_{\alpha 2}\frac{\sigma B^2}{\rho}\alpha\varepsilon, \quad (43)$$

$$S_{ij}^{(RS)}/\rho = -\mu_{ij} = -2\frac{\sigma B^2}{\rho}\left(GR_{ij} + \frac{H}{2}[n_i n_k R_{kj} + n_j n_k R_{ki}]\right), \quad (44)$$

where

$$G(\alpha, I_{na}) = \alpha + \frac{27}{20}\alpha^2(1-\alpha)\left(I_{na} + \frac{2}{3}\right), \quad (45)$$

$$H(\alpha) = -\frac{27}{10}\alpha^2(1-\alpha), \quad (46)$$

$$I_{na} \equiv \frac{n_i n_k R_{ki}}{K} - \frac{2}{3}. \quad (47)$$

3 Model implementation

3.1 Electric potential equation

The generic user scalar transport equation available in *CFX* can be reduced to a Poisson equation by defining the scalar as quasistatic (to disable the transient term) and disabling the convection term (one option in selecting differencing scheme). With the eddy diffusivity hypothesis, the resulting model equation for mean electric potential then takes the form

$$-\frac{\partial}{\partial x_j}\left(\left[\Gamma_\Phi + \frac{\mu_T}{\sigma_\Phi}\right]\frac{\partial\Phi}{\partial x_j}\right) = S_\Phi. \quad (48)$$

To obtain the desired results, we must cancel the turbulent contribution to the total diffusivity in (48). We do this by setting the total diffusivity explicitly in the user subroutine USRDIF. The actual value used is unimportant, but the same value must of course be used when the source term S_Φ is computed. Here we use the value for the electric conductivity specified in the command file, $\Gamma_\Phi = \sigma$. For consistency with (15), the source term S_Φ in (48) must be

$$S_\Phi = -\nabla \cdot (\Gamma_\Phi \mathbf{U} \times \mathbf{B}). \quad (49)$$

When (48) is integrated over a control volume, we use Gauss' theorem to rewrite the volume integral of the source as a surface integral,

$$\begin{aligned} \int S_\Phi dV &= -\int \nabla \cdot (\Gamma_\Phi \mathbf{U} \times \mathbf{B}) dV \\ &= -\int \Gamma_\Phi (\mathbf{U} \times \mathbf{B}) \cdot d\mathbf{A}. \end{aligned}$$

In its discretized form, this is a sum over the faces f of a grid cell,

$$\begin{aligned} \int S_\Phi dV &= S_\Phi V_{cell} \\ &= - \sum_f \Gamma_\Phi(\mathbf{U}_f \times \mathbf{B}_f) \cdot \mathbf{A}_f, \end{aligned}$$

so that

$$S_\Phi = - \frac{1}{V_{cell}} \sum_f \Gamma_\Phi(\mathbf{U}_f \times \mathbf{B}_f) \cdot \mathbf{A}_f. \quad (50)$$

Here \mathbf{U}_f and \mathbf{B}_f are interpolated values of \mathbf{U} and \mathbf{B} at the faces, and \mathbf{A}_f is the outward normal area vector of the face. This approach guarantees conservation of source current, and it is consistent with the calculation of total current density described in Sec. 3.2.

Walls are generally assumed to be non-conducting, in which case zero normal flux boundary conditions should be specified for electrostatic potential. Alternatively, a fixed potential is specified at the wall. If only zero flux boundary conditions are used, the electrostatic potential is not uniquely defined. The potential must be known in some point in the domain. An MHD model option can be read from the command file to define Cartesian coordinates for a reference point in the domain where the electrostatic potential is forced to zero (see App. A.2).

All inlet boundaries in *CFX* use Dirichlet boundary conditions for user scalars. The same is true if we have inflow through pressure and mass flow boundaries. Unless we want these boundaries to have fixed potentials, the behavior can be modified with MHD model options read from the command file, causing the flow boundaries to behave as if they were electrically insulating (see App. A.2). This is achieved by evaluating the source term (50) with $\mathbf{U} \times \mathbf{B} = 0$ on cell faces at the flow boundaries, and setting the corresponding diffusion coefficient to zero when the potential equation is solved.

3.2 Total current density and Lorentz force

The total current density is

$$\mathbf{J} = -\sigma \nabla \Phi + \sigma \mathbf{U} \times \mathbf{B}. \quad (51)$$

The two terms on the right-hand side are of the same order, and the total current density is often a small difference between them. Near an isolated wall, for example, the two terms must balance almost completely for the wall-normal current to be zero. A second-order accurate calculation of the potential gradient in a cell P can be interpreted as the average of the first-order gradients evaluated at the cell faces. To achieve the required accuracy of the total current density, it is essential that we compute the $\mathbf{U} \times \mathbf{B}$ term in a similar manner. It should therefor be computed as the average of $(\mathbf{U} \times \mathbf{B})_f$ at the cell faces.

To describe the calculation of the $\mathbf{U} \times \mathbf{B}$ term for general body-fitted grids, we need to discuss briefly the general non-orthogonal coordinate system used in *CFX*.

We first introduce the Cartesian physical space coordinates x^i , and the general non-orthogonal computational space coordinates ξ^i . The computation space

system is such that grid cells have unit volume and edges of unit length in terms of computational coordinates. The two systems are related through the Jacobian matrix J_j^i , and the inverse Jacobian matrix \bar{J}_j^i ,

$$J_j^i \equiv \frac{\partial x^i}{\partial \xi^j}, \quad (52)$$

$$\bar{J}_j^i \equiv \frac{\partial \xi^i}{\partial x^j}. \quad (53)$$

We also define the Jacobian determinant

$$|J| \equiv \det(J_j^i). \quad (54)$$

In any point in the curvilinear coordinate system (ξ^i) , we can define two frames of basis vectors:

$$\mathbf{e}_{(i)} = \frac{\partial \mathbf{x}}{\partial \xi^i} \quad (\text{tangent to coordinate curves}), \quad (55)$$

$$\mathbf{e}^{(i)} = \frac{\partial \xi^i}{\partial \mathbf{x}} \quad (\text{normal to coordinate surfaces}). \quad (56)$$

These are the *covariant* and *contravariant* basis vectors, respectively. Any Cartesian vector $\mathbf{V} = (V_i)$ has two sets of components with respect to these basis vectors:

$$\mathbf{V} = V'^i \mathbf{e}_{(i)} = V'_i \mathbf{e}^{(i)}, \quad (57)$$

where V'^i and V'_i are the contravariant and covariant components of \mathbf{V} , respectively. The two sets of basis vectors are dual to each other, which means that

$$\mathbf{e}^{(i)} \cdot \mathbf{e}_{(j)} = \delta_j^i. \quad (58)$$

From (57) it then follows that the scalar product of a basis vector with the Cartesian vector produces the corresponding components,

$$V'^i = \mathbf{V} \cdot \mathbf{e}^{(i)}, \quad (59)$$

$$V'_i = \mathbf{V} \cdot \mathbf{e}_{(i)}. \quad (60)$$

The two sets of basis vectors can be given a geometric interpretation. Let Ω be denote a parallelepiped generated by the covariant basis vectors $\mathbf{e}_{(i)}$. Let $\mathbf{A}^{(i)}$ be area vectors, i.e., vectors normal to the faces of Ω with magnitude equal to the area of the faces. See Fig. 1. Area vectors are given by

$$\mathbf{A}^{(1)} = \mathbf{e}_{(2)} \times \mathbf{e}_{(3)}, \quad \mathbf{A}^{(2)} = \mathbf{e}_{(3)} \times \mathbf{e}_{(1)}, \quad \mathbf{A}^{(3)} = \mathbf{e}_{(1)} \times \mathbf{e}_{(2)}, \quad (61)$$

and the volume of Ω is

$$vol(\Omega) = \mathbf{e}_{(1)} \cdot \mathbf{e}_{(2)} \times \mathbf{e}_{(3)} = |J|. \quad (62)$$

This is often called the infinitesimal volume element; in *CFX* the notation $\sqrt{g} = vol(\Omega) = |J|$ is used.

From the above we can express the contravariant basis vectors in terms of the area vectors and the infinitesimal volume element,

$$\mathbf{e}^{(i)} = \frac{\mathbf{A}^{(i)}}{|J|}. \quad (63)$$

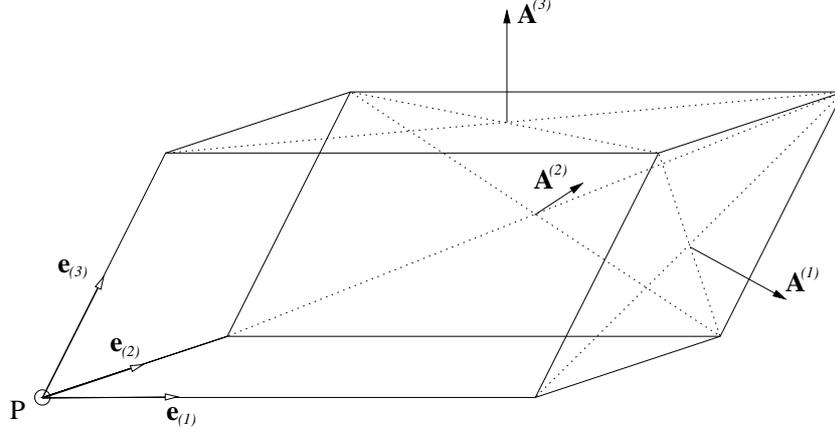


Figure 1: The covariant basis vectors in a point P generates a parallelepiped. Area vector are normal to the faces, with magnitude equal to the face area.

In *CFX*, both face area vector $\mathbf{A}^{(f)}$ (positively oriented in computational space) and the facial volume element \sqrt{g}_f is stored, so the contravariant basis vector normal to each face f is given by (63),

$$\mathbf{e}^{(f)} = \frac{\mathbf{A}^{(f)}}{\sqrt{g}_f}. \quad (64)$$

The covariant basis vector at a cell face is given by the vector between the cell centers P and Q on either side,

$$\mathbf{e}_{(f)} = d\mathbf{x}_f = \mathbf{x}_Q - \mathbf{x}_P \quad (65)$$

(assuming Q is located towards positive computational coordinates, relative to P).

Let us now return to the calculation of the $\mathbf{U} \times \mathbf{B}$ term in (51). For a more compact notation, let

$$\mathbf{Z} = \mathbf{U} \times \mathbf{B}. \quad (66)$$

Using the weight factors stored in *CFX*, we interpolate \mathbf{U} and \mathbf{B} to cell faces, and compute \mathbf{Z}_f at cell faces. The contravariant component in the face normal direction is

$$Z'_{(f)} = \mathbf{Z} \cdot \mathbf{e}_{(f)} = \mathbf{Z} \cdot d\mathbf{x}_f. \quad (67)$$

The vector \mathbf{Z}_P in the center of cell P is now redefined as the average of the contravariant components at the cell faces,

$$\mathbf{Z}_P = \sum_f W_P^{(f)} Z'_{(f)} \mathbf{e}^{(f)} \quad (68)$$

$$= \sum_f W_P^{(f)} Z'_{(f)} \mathbf{A}^{(f)} / \sqrt{g}_f. \quad (69)$$

The weight factor in the averaging is one half, $W_P^{(f)} = 1/2$.

In the special case where the grid is orthogonal, the contravariant and covariant basis vectors are parallel, $\mathbf{e}_{(i)} \parallel \mathbf{e}^{(i)}$. From (63) and (58) then follows that

$$\mathbf{e}_{(f)} = \frac{\sqrt{g_f}}{A_f^2} \mathbf{A}^{(f)}, \quad (70)$$

where $A_f = |\mathbf{A}^{(f)}|$ is the face area. If this inserted in (67) and (68), we get

$$\mathbf{z}_P = \sum_f W_P^{(f)} \hat{Z}^{(f)} \mathbf{A}^{(f)} / A_f^2, \quad (71)$$

where $\hat{Z}^{(f)} = \mathbf{z}_f \cdot \mathbf{A}^{(f)}$ is the normal flux component we used for computing the source term for the electrostatic potential equation. This was stored in memory, and can be reused for computing the total current in the orthogonal case.

To compute the total current density, we also need to compute the gradient of electrostatic potential. We use the built-in *CFX* subroutine GRADS for this. This routine however does not compute the correct gradient in cells adjacent to wall boundaries. The gradients in cells next to walls, inlets, pressure boundaries and massflow boundaries are therefor recomputed after the call to GRADS (see App. B.10). If the MHD model options for insulated inlets/outlets are invoked, this affects both the calculation of potential gradient, and the $\mathbf{U} \times \mathbf{B}$ term at the boundaries, in a way consistent with zero total current through the flow boundary (see App. B).

With total current density computed with (51), the mean Lorentz force is given by $\mathbf{F}_L = \mathbf{J} \times \mathbf{B}$. *CFX* supports under-relaxation of body forces in the command file. The Lorentz forces can be switched off completely with a MHD model option read from the command file.

3.3 Dimensionality anisotropy

In the *CFX* implementation, α is assigned a user scalar equation of the generic form (11). Source terms are linearized so that, for the source in a cell P , the source term for a scalar Q can be written

$$S = S_U + S_P Q_P. \quad (72)$$

Source terms proportional to the variable itself should be put in S_P if they are negative, since they will then enhance the diagonal dominance of the solution matrix and improve convergence and robustness.

The symbolic mathematics engine MAPLE V was used to manipulate the source terms, and translate the model expressions presented above into fortran source code (see App. C). This procedure reduces the risk for programming errors, and simplifies debugging. The procedure was as follows:

1. The source term model expression is expanded into component form and simplified.
2. The resulting expressions are evaluated term by term, sorting them into S_U or S_P . A proportional term is put into S_P if it is guaranteed to be negative, and into S_U otherwise. The sign of the term is determined by MAPLE, assuming that K and ε are positive, α is between zero and one, and all other variables and constants are real.

3. Names of variables and parameters are substituted for those used in *CFX*, and the expressions for S_U and S_P are automatically optimized and translated into fortran source code.
4. The automatically generated source code is included in the appropriate user fortran routine.

The mean velocity gradients used in P_α and $\Pi_\alpha^{(r)}$ are computed using the built-in *CFX* subroutine GRADV. As far as we can tell, the gradients obtained next to the walls are consistent with those used to compute the turbulence production and pressure-strain terms in the standard turbulence equations. If the user has selected the $K-\varepsilon$ model, the Reynolds stresses in P_α and $\Pi_\alpha^{(r)}$ are substituted by the Boussinesq hypothesis (26). The two strain terms contain the unit direction vector of the magnetic field, \mathbf{n} , which is undefined if $\mathbf{B} = 0$. In this case, the code uses $\mathbf{n} = \mathbf{e}_z$ to avoid numerical errors. Note however, that if there is no preferred direction of length scale anisotropy, the only physically relevant value for α would be the isotropic value of $1/3$. The mean strain terms P_α and $\Pi_\alpha^{(r)}$ can be switched off with an MHD model option in the command file, to reduce the computational effort.

3.4 Turbulence equations

The fortran code for the source terms in the turbulence equations is automatically generated, with the same procedure described above for the source terms of the α equation.

All magnetic source terms in the standard turbulence equations can be switched off with an MHD model option in the command file.

4 Results

4.1 Calculation of currents on non-orthogonal grids

The calculation of total current on non-orthogonal grids is demonstrated in the following simple 2D case. The electric potential and total current are computed in a box with side L , with fixed zero potential boundary conditions on all sides. We assume a uniform velocity field through the domain, from left to right, and solve only the electric potential equation. The magnetic field distribution is cone shaped, localized inside a circle of radius $0.14L$, and the field orientation is perpendicular to the plane. This induces currents $\sigma \mathbf{U} \times \mathbf{B}$ directed downwards inside the magnetic field, and a counteracting electric potential which causes the currents to close outside the magnetic field.

In Fig. 2 we compare results computed on one uniform rectangular grid, and one non-orthogonal grid with badly distorted grid cells. Both grids have 50×50 grid cells. The distortion and the uneven distribution of grid cells cause a slight asymmetry of both electric potential and current distribution on the non-orthogonal grid, but the results are otherwise consistent. In contrast, the lower right plot shows what happens if we compute the total current on the skewed grid as if it was rectangular. The accuracy of the term $\mathbf{U} \times \mathbf{B}$ is then very poor, and we get a large false horizontal current component inside the magnetic field.

Table 1: Description of channel flow simulations.

Case	Model	Description
A	$K\text{-}\varepsilon\text{-}\alpha$	Three-equation MHD model (no strain terms in α equation).
B	RST- α	MHD Reynolds stress model (no strain terms in α equation).
C	RST- α (S)	Complete MHD Reynolds stress model, including strain terms in α equation.
D	Std $K\text{-}\varepsilon$	Standard model (no MHD terms).

The non-orthogonal grid above was constructed of two 50×25 blocks. This enabled us to use the same simple test case to verify that the algorithms for electrostatic potential and total current work for multiblock grids with orientation changes at block boundaries.

4.2 Channel flow with “M” profiles

The following simulation of a channel flow considers the same geometry as the mercury experiment by Tananaev [6] (also studied numerically by Kenjereš and Hanjalić [2]). The flow is subjected to a localized transverse magnetic field, as illustrated in Fig. 3 (the region of the magnetic field is shaded). In terms of the channel half-width $h = 0.020$ m, the channel Reynolds number is $\text{Re} = Uh/\nu = 2 \cdot 10^5$, the Hartmann number $\text{Ha} = \sqrt{\sigma/(\rho\nu)}Bh = 700$, and the interaction number $N = \text{Ha}^2/\text{Re} = 2.45$.

The most striking effect of the magnetic field is the development of an “M”-shaped velocity profile. This is caused by the distribution of mean Lorentz forces in the entry to the magnet, and it is enhanced further in the magnet exit.

The Joule dissipation of turbulence in the magnetic field reduces the turbulent viscosity, but the reduction of turbulence is visible only as a secondary effect on the mean velocity profiles. Unless we have measured turbulent statistics to compare with, the dominating effect of the mean Lorentz forces makes this experimental configuration a less than ideal test case for MHD turbulence models. Nevertheless, it serves here as an illustration of how the MHD model implementation works, and a chance to look at the behavior of the MHD turbulence models in a wall-bounded 3D flow.

The number of grid cells used in the channel cross-section was 59×39 (largest in the y direction), and 65 in the streamwise direction. The grid was stretched to give the highest resolution close to walls, and in the entry and exit of the magnet.

All walls were assumed non-conducting, and the inlet and downstream pressure boundary defined as electrically insulated. We used standard turbulent wall functions, with y^+ in the range 50 to 80 inside the magnet.

Four different simulations were performed, with different turbulence models. See Table 1. In all cases, the model constants were $\{C_{\alpha 1}, C_{\alpha 2}, C_{\varepsilon \alpha}\} = \{0.866, 0.2, 0.5\}$.

Figure 4 shows the velocity profile in the center of the channel, and in the center of the magnet. The width of the wall jet predicted by the MHD models is

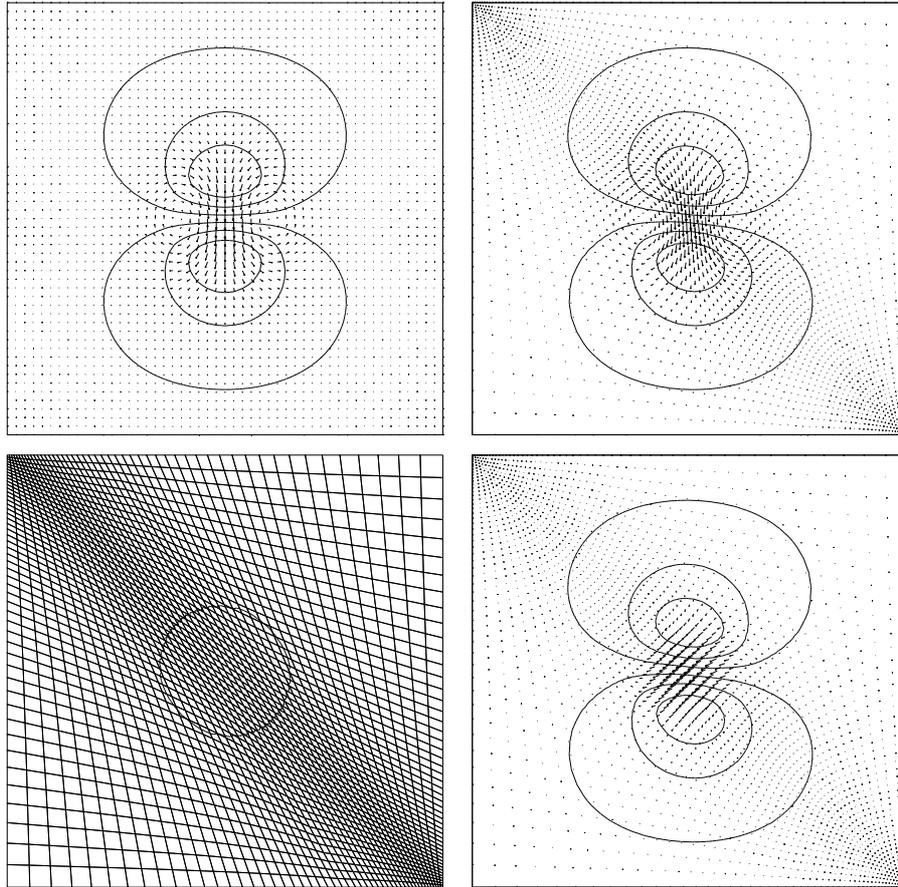


Figure 2: The top two figures show contours of electrostatic potential and vectors of total current for a uniform rectangular grid (left), and a non-orthogonal grid (right). The non-orthogonal grid and the region of the magnetic field is shown in the lower left plot. The lower right plot, finally, shows what happens if we try to compute the total current as if the grid was rectangular (assuming contravariant and covariant basis vectors are parallel). Note that the current here has a false horizontal component inside the magnetic field.

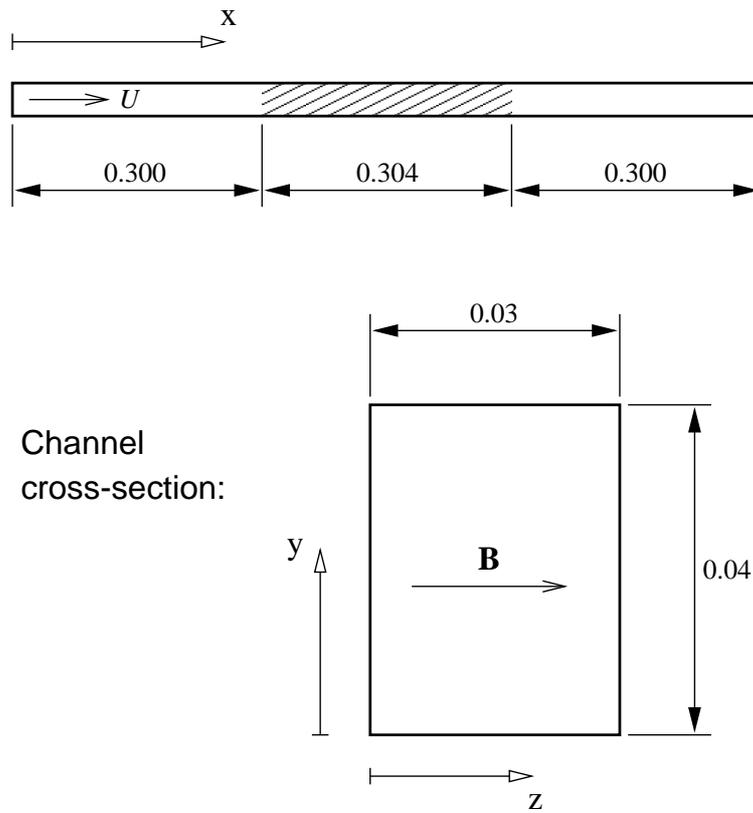


Figure 3: Geometry of the channel flow.

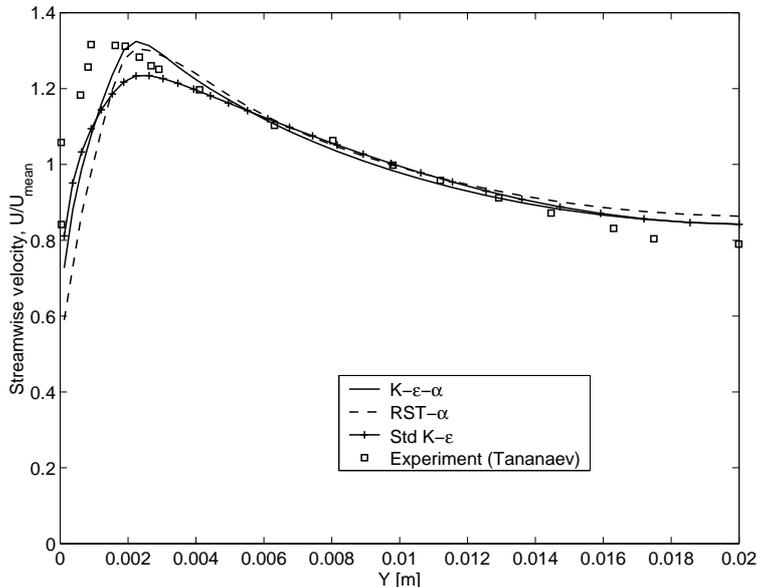


Figure 4: Mean velocity profiles in the center of the magnet. Lines show model predictions, and square symbols show experimental data (Tananaev [6]) for comparison.

in good agreement with experiments. With the standard $K-\varepsilon$ model, turbulent diffusion is over predicted, and the velocity peak is “smeared out”. All simulations locate the velocity peak too far from the wall. Assuming the experimental data is reliable, there is currently no obvious explanation for this. The trend is the same for both MHD models and the standard $K-\varepsilon$. It may be a result of the standard wall functions we used in the simulation. We are also aware that certain inhomogeneous contributions to the Joule dissipation near walls are currently not accounted for in the MHD models. Figure 5 shows vectors of total current, and contours of electric potential, in the entry of the magnet. Figure 6 shows contours of the x -component of the Lorentz force in the same region, together with the mesh. The grid resolution in the streamwise direction is rather poor, and one might suspect that the strong gradients of total current and Lorentz forces in the entry of the magnet are not resolved properly, and that this could be a source of error. However, a test with three times more grid cells in the streamwise direction in the entry of the magnet did not affect the location of the velocity maximum.

Figure 7 shows K , ε and μ in the center of the magnet, as predicted by the $K-\varepsilon-\alpha$ model. The results with the standard $K-\varepsilon$ model are shown for comparison. In the core flow, the MHD model predicts K and ε values about two orders of magnitude smaller than the standard model, but close to the wall the difference is small. This is because the turbulence production and viscous dissipation dominates over Joule dissipation close to the wall. A local interaction number $N_{loc} = \sigma B^2 K / (\rho \varepsilon)$ is of order 10 or larger outside the velocity peak, but drops off to below unity when we approach the wall (not shown here).

Figure 8 compares predictions of turbulent kinetic energy in the center of

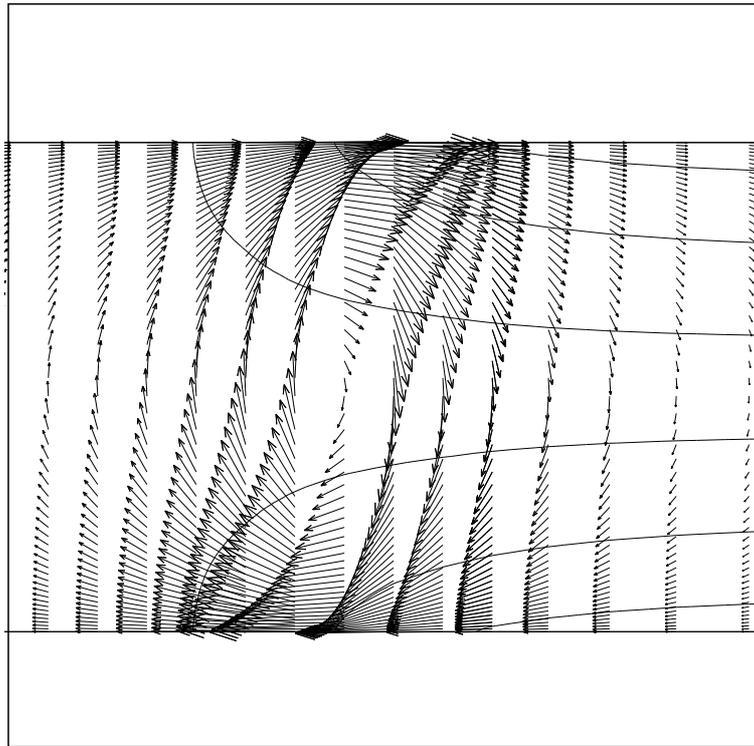


Figure 5: Vectors of total current, and contours of electric potential in the entry of the magnet (K - ε - α model).

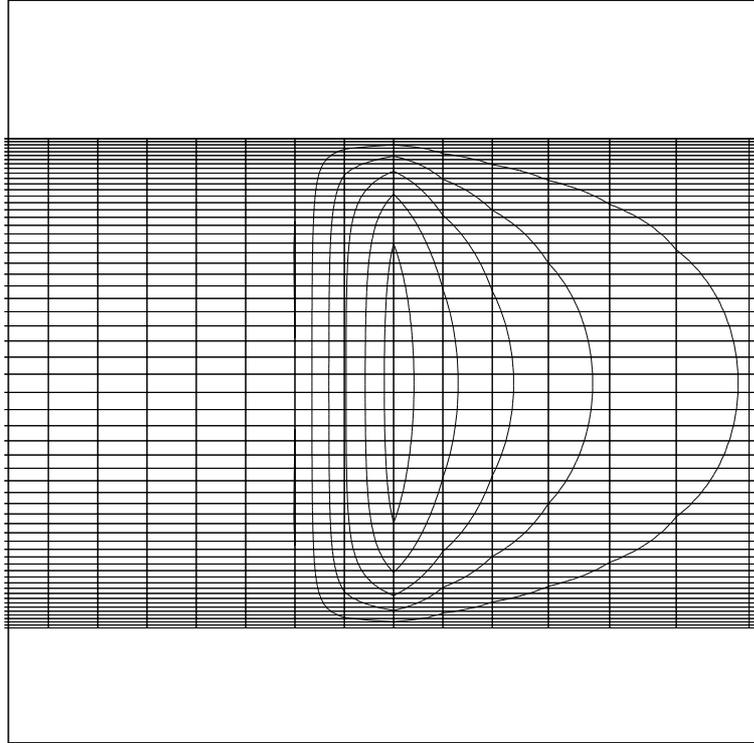


Figure 6: Contours of Lorentz force, together with the mesh in the same plane, in the entry of the magnet (K - ε - α model).

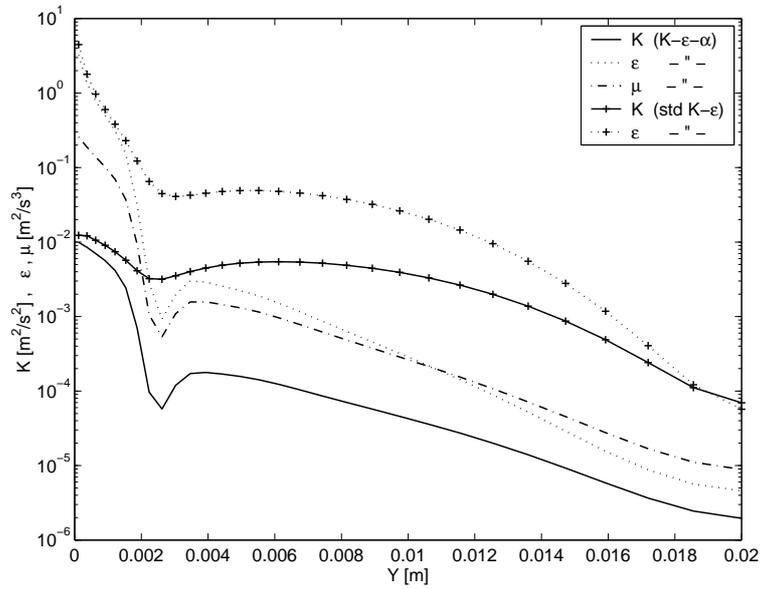


Figure 7: Turbulent kinetic energy, viscous and Joule dissipation in the center of the magnet, as predicted with the K - ε - α model. Predictions with the standard K - ε model are shown for comparison.

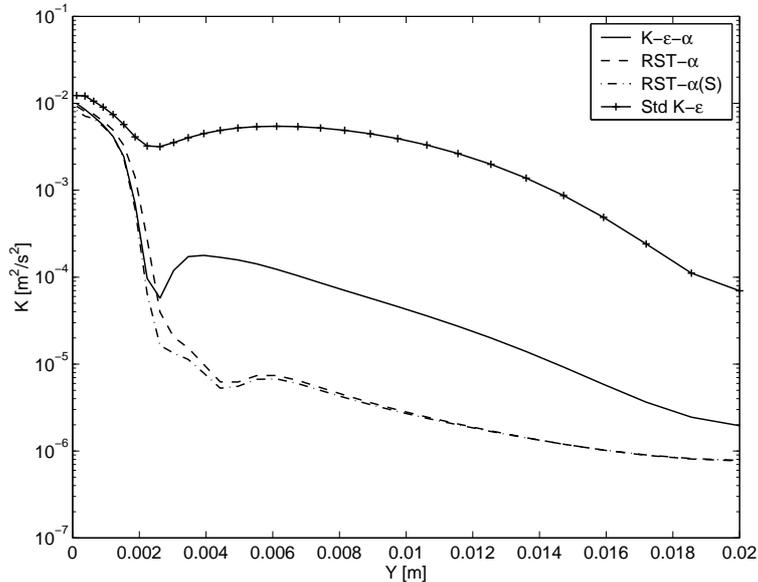


Figure 8: Predictions of turbulent kinetic energy with different turbulence models (center of magnet).

the magnet, as predicted by the different models (Tab. 1). We see it makes little difference whether we include the strain terms in the α equation, or neglect them (models RST- α (S) and RST- α , respectively). As expected, all MHD models predict lower energy than the standard K - ε model, but there is also a significant difference between the three equation K - ε - α and the RST- α models. The difference between the two is that the production term in a Reynolds stress model is exact, but it is modeled using the Boussinesq hypothesis in the K - ε model. The Boussinesq hypothesis relates the Reynolds stress anisotropy to the mean strain tensor. This is often a good approximation in ordinary shear flows, but in MHD flows it neglects the effect of Joule dissipation on the stress anisotropy.

The Reynolds stresses predicted by the RST- α model in the center of the magnet are plotted in Fig. 9 (normalized with the turbulent kinetic energy). The plotted cross-section is in the mid plane of the channel ($z = 0.015$), so the only non-zero shear stress is $\langle u'v' \rangle$. The effect of the anisotropic Joule dissipation dominates in the center of the channel, which causes the component $\langle w'w' \rangle$ parallel with the magnetic field to be larger than the other normal stresses. This behavior is typical for Joule dissipation of homogeneous turbulence, for which the current model works well. In wall-bounded flows, however, the model neglects certain inhomogeneous contributions to the Joule dissipation tensor, and to the pressure-strain terms in the Reynolds stress equations. The effect of these contribution would presumably be to redistribute energy from the field-parallel component into the other components, to force the turbulence field towards a 2D and two-component (2C) state.

The left pane in Fig. 10 shows the profile of α predicted by the different models in the center of the magnet, while the right pane shows the budget of α

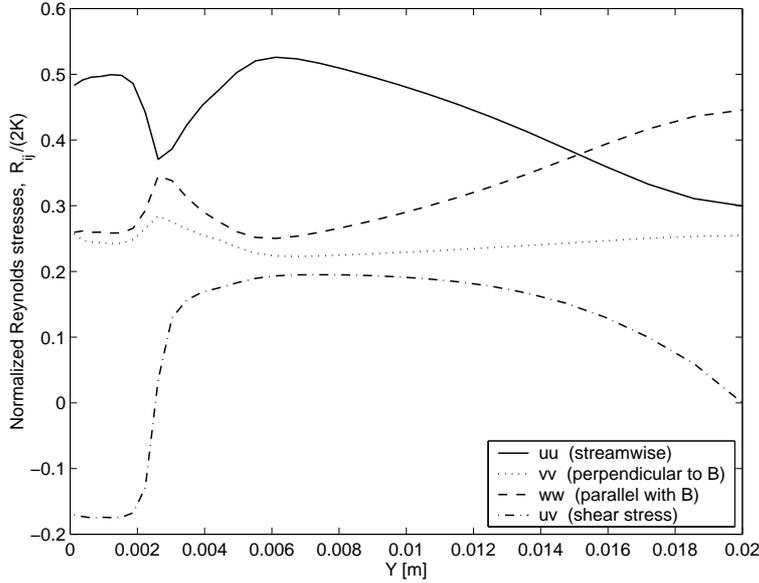


Figure 9: Profiles of normalized Reynolds stresses in the center of the magnet.

production and dissipation in the case of the RST- α model. The strain terms in the α equation give a significant positive contribution to α only near the wall. The right plot shows that the source and sink terms dominate over transport near the wall. This behavior is similar to that of the turbulent kinetic energy and the Reynolds stresses, and thus supports our choice of zero normal flux boundary conditions in the first node near walls.

Figure 11, finally, shows profiles of effective viscosity, $\mu + \mu_T$, in the exit of the magnet, as predicted with the different models. Note that the baseline of the figure represents the molecular viscosity. The difference between predictions with the RST models and the K - ε - α was discussed earlier (see Fig. 8).

5 Discussion and outlook

This report documents an implementation of mean electromagnetic equations, Lorentz forces, and MHD turbulence models in the commercial code *CFX* 4.3.

From a numerical point of view, the most difficult part of this effort is not, as one might think, the implementation of the new turbulence models. Instead the challenge is to solve the equation for electrostatic potential, and compute the total current density with sufficient accuracy, especially on general non-orthogonal grids.

We used the symbolic mathematics tool MAPLE V to manipulate, simplify and linearize all source terms for the turbulence equations, and to translate them directly into fortran source code for the *CFX* subroutines. This minimizes the risk for programming errors, and simplifies validation and debugging. The procedure will also simplify future modifications, and evaluation of alternative models.

Both the Reynolds stress and eddy viscosity versions of the implemented

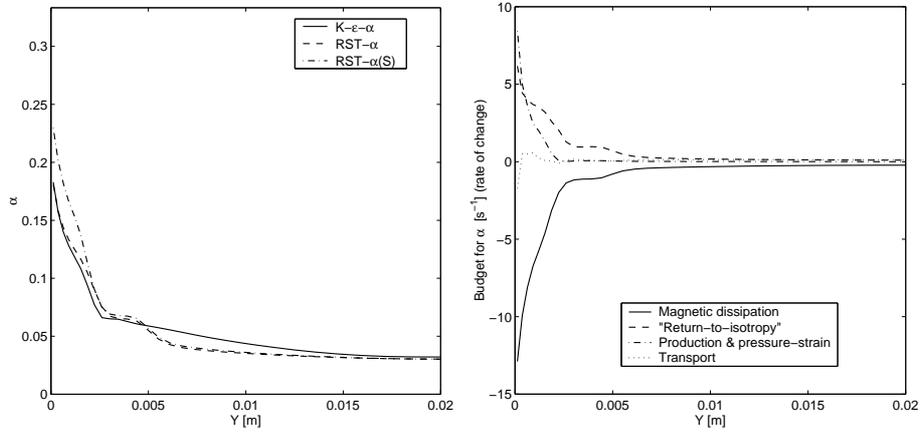


Figure 10: Left: Profiles of α in the center of the magnet. Right: Budget of α in the same cross-section (RST- α (S) model).

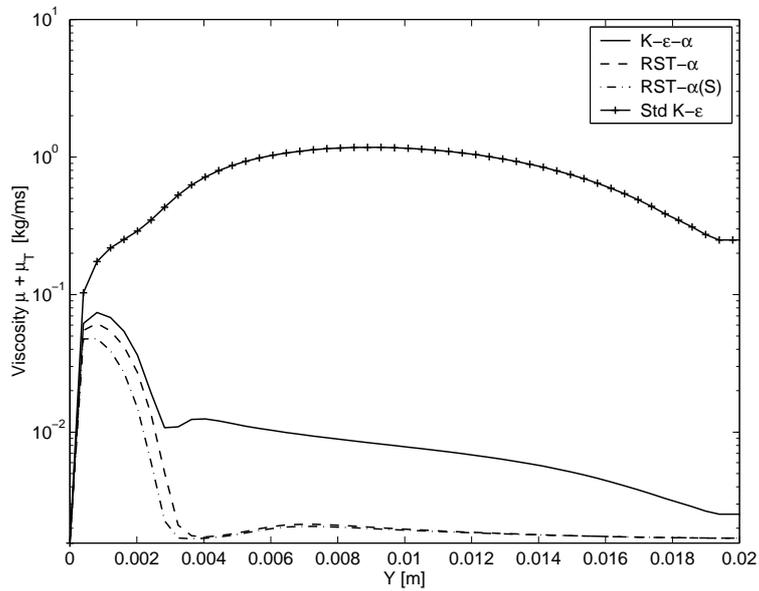


Figure 11: Predicted effective viscosities, $\mu + \mu_T$, in the exit of the magnet. The baseline of the plot represents the value of the molecular viscosity.

MHD turbulence models appear to be very robust and well-behaved. Apart from the extra work required for solving one more equation, the rate of convergence seem to be no worse for the MHD models, than for the standard models.

The current versions of the MHD turbulence models were developed to perform well for homogeneous turbulence, in the absence of walls. In order to simplify the model expressions, some inhomogeneous contributions to fluctuating pressure and fluctuating electric potential were neglected. Some of the missing terms are analogous to the wall reflection terms in the rapid pressure-strain model by Launder, Reece and Rodi [3]. This is the Reynolds stress model used in *CFX*, but the wall reflection terms are not implemented. For ordinary shear flows, the neglected terms are usually important only near walls. In MHD flows, however, the turbulence tends to be two-dimensional, so that turbulent structures grow very large in one direction. When the length of these structures becomes the same order as the distance to the nearest wall, we can expect the neglected inhomogeneous terms to be important. They will hence be important even in the core of an MHD channel flow, if the length scale anisotropy is sufficiently developed.

The missing inhomogeneous terms should certainly contribute to the Joule dissipation terms in the scalar K and ε equations, but their effect is perhaps more obvious in the Reynolds stress equations. Here they should contribute to the Joule dissipation tensor, and to the rapid and slow pressure-strain terms. For a wall perpendicular to the magnetic field, their net effect would be to redistribute energy from the wall-normal component (parallel with the field), to the other components. This means that the Reynolds stress component parallel with magnetic field will be damped, so that we approach a turbulent state which is two-component (2C) as well as 2D. This a feature typical for wall-bounded MHD flows, and different from the behavior of homogeneous turbulence. The channel flow simulation discussed in Sec. 4.2 (especially Fig. 9) illustrated the inability of the current MHD Reynolds stress model to predicted the damping of the field-parallel stress component.

Including the inhomogeneous contributions in the scalar K and ε (and α) equations is probably relatively straightforward. Including them in the Reynolds stress equations is much more difficult. The simple Launder, Reece and Rodi model only works well for relatively small stress anisotropies. Since all terms are linear, they will produce unphysical results even for rather moderate anisotropies. To deal with the situation, completely new and algebraically complex higher-order model terms would have to be developed, including also dimensionality information. The main reason we want accurate predictions of Reynolds stresses, is to model anisotropic diffusion, using Eq. (18). Considering the difficulties with a full Reynolds stress model, a more effective approach is probably to use a three-equation K - ε - α , with an algebraic model for modeling of Reynolds stresses and anisotropic diffusion.

In future work, the present model implementation will be used to study the effect of the different MHD model constants, and to compare model predictions with experiments reported in literature. Unfortunately turbulence statistics are very difficult to measure, and not often reported in the literature. This makes validation of MHD turbulence models very difficult. Direct numerical simulations (DNS), and large eddy simulations (LES) may play an important role here.

References

- [1] B. J. Daly and F. H. Harlow. Transport equations in turbulence. *Physics of Fluids*, 13(11):2634–2649, 1970.
- [2] S. Kenjereš and K. Hanjalić. On the implementation of effects of Lorentz force in turbulence closure models. *Int. Journal of Heat and Fluid Flow*, 21(3):329–337, June 2000.
- [3] B. E. Launder, G. J. Reece, and W. Rodi. Progress in the development of a Reynolds-stress turbulence closure. *Journal of Fluid Mechanics*, 68:537–566, 1975.
- [4] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. *Computer Methods in Appl. Mech. and Eng.*, 3:269–289, 1974.
- [5] William C. Reynolds. Effects of rotation on homogeneous turbulence. In *Proc. 10th Australasian Fluid Mechanics Conference*, Melbourne, Australia, 1989. University of Melbourne.
- [6] A. B. Tananaev. *MHD Duct Flows*. Atomizdat, 1979.
- [7] O. Widlund and G. Tallbäck. Modeling of anisotropic turbulent transport in simulations of liquid metal flows in magnetic fields. In *Proceedings of the Third Int. Symposium on Electromagnetic Processing of Materials, EPM2000, April 3-6, 2000*, pages 97–102, Nagoya, Japan, 2000. ISIJ.
- [8] O. Widlund, S. Zahrai, and F. H. Bark. Development of a Reynolds stress closure for modeling of homogeneous MHD turbulence. *Physics of Fluids*, 10(8):1987–1996, 1998.
- [9] O. Widlund, S. Zahrai, and F. H. Bark. Structure information in rapid distortion analysis and one-point modeling of axisymmetric magnetohydrodynamic turbulence. *Physics of Fluids*, 12(10):2609–2620, 2000.

A Using the MHD model in *CFX*

In order to use this MHD model implementation, the first step is of course to include all the necessary MHD fortran routines, and the *CFX* user fortran routines, and make sure that latter contain the proper calls to the MHD routines. If the magnetic field should be defined in user fortran, the routine MHDBXT should be updated accordingly. Even if MHDBXT is not used, it should be included in the compilation.

Most of the modeling can then be controlled from the *CFX* command file. Section A.1 describes the necessary *CFX* options in the command file, while Sec. A.2 describes the special MHD model options read from the end of the command file.

A.1 CFX options in the command file

CFX user subroutines must be specified, as follows:

```
>>USER FORTRAN
  USRSRC
  USRBF
  USRINT
  USRWTM
  USRDIF
```

The first three perform calls to specific MHD routines, which do the actual work. The user can add his/her own stuff in any of the routines.

The following user scalar variable names must be defined. The names are important, but the order is not.

```
>>VARIABLE NAMES
  USER SCALAR1 'EPOT'
  USER SCALAR2 'ALPHA'
  USER SCALAR3 'USRD B_X'
  USER SCALAR4 'USRD B_Y'
  USER SCALAR5 'USRD B_Z'
  USER SCALAR6 'USRDCC J_X'
  USER SCALAR7 'USRDCC J_Y'
  USER SCALAR8 'USRDCC J_Z'
  USER SCALAR9 'USRDCC S_MAG'
  USER SCALAR10 'USRDCC S_RET'
  USER SCALAR11 'USRDCC S_PRD'
  USER SCALAR12 'USRDCC S_PST'
```

The last seven are stored only for post-processing, and the code could easily be rewritten to save storage space.

The differencing scheme for EPOT must be set to “NO CONVECTION”, to disable the convection term. If the simulation is transient, also the transient term must be disabled by defining EPOT as “quasistatic”. Example:

```
>>MODEL DATA
  >>DIFFERENCING SCHEME
    EPOT          'NO CONVECTION'
```

```

>>PHYSICAL PROPERTIES
  >>TRANSIENT PARAMETERS
    >>FIXED TIME STEPPING
      TIME STEPS 30*0.010
    >>QUASISTATIC VARIABLES
      EPOT

```

Fluid viscosity and density should be specified as usual. Note however that exactly the same values must be specified for the corresponding MHD model options (see next section)! It is recommended to set the scalar diffusivity of ALPHA equal to the viscosity. The value specified for EPOT is unimportant, as it is overridden by the value of the MHD model option COND read from the end of the command file (see next section).

```

/* ----- */
#CALC
  DENS = 13550.0;
  VISC = 0.00156;
  COND = 1.05E+6;
#ENDCALC
/* ----- */
>>FLUID PARAMETERS
  VISCOSITY #VISC
  DENSITY #DENS
>>SCALAR PARAMETERS
>>DIFFUSIVITIES
  EPOT #COND
  ALPHA #VISC

```

The MHD model implementation supports both the RST model, and the high-Re K - ε model. The code automatically recognizes which one is used. The turbulent Prandtl number for ALPHA is set to 1.0, same as for the Reynolds stresses.

```

>>TURBULENCE PARAMETERS
  >>TURBULENCE MODEL
    TURBULENCE MODEL 'DIFFERENTIAL STRESS'
/*   TURBULENCE MODEL 'K-EPSILON'   */
>>TURBULENT PRANDTL NUMBER
  ALPHA 1.0

```

The body forces can be underrelaxed from within the command file, if it helps convergence. If turbulence equations are underrelaxed, it makes sense to use the same under-relaxation for ALPHA. Note that no under-relaxation should be used for EPOT, as it makes convergence slower than it needs to be. It is recommended to allow a fairly large number of sweeps for both pressure and EPOT equations (and a small value of the reduction factor), as this is noted to improve the rate of convergence.

```

>>SOLVER DATA
  >>PROGRAM CONTROL
    MAXIMUM NUMBER OF ITERATIONS 1200

```

```

MASS SOURCE TOLERANCE 1.0E-6
>>UNDER RELAXATION FACTORS
U VELOCITY 0.4
V VELOCITY 0.4
W VELOCITY 0.4
BFX 0.2
BFY 0.2
BFZ 0.2
K 0.3
EPSILON 0.3
ALPHA 0.3
EPOT 1.0 /* NOTE! */
>>SWEEPS INFORMATION
>>MAXIMUM NUMBER
EPOT 50
PRESSURE 200
>>REDUCTION FACTORS
EPOT 0.10
PRESSURE 0.10

```

Flow boundary conditions are set as usual. The inlet Dirichlet boundary condition for EPOT (default treatment for scalars) can be overridden using MHD model options for insulated inlets and outlets (see next section).

```

>>MODEL BOUNDARY CONDITIONS
>>SET VARIABLES
PATCH NAME 'OUT'
PRESSURE 5000.0
>>SET VARIABLES
PATCH NAME 'IN'
NORMAL VELOCITY 1.15
K 2.0E-5
EPSILON 9.9E-6
EPOT 0.0
ALPHA 0.333

```

ALPHA should have zero normal flux at walls. This is the default treatment for all scalars in *CFX*, so the specification in the example below is redundant. The same is true for EPOT and non-conducting walls, but in some situations one might want to set EPOT to a fixed potential, as for the wall “W_HIGH” in the example below.

```

>>WALL BOUNDARY CONDITIONS
PATCH NAME 'W_LOW'
EPOT FLUX 0.0
ALPHA FLUX 0.0
>>WALL BOUNDARY CONDITIONS
PATCH NAME 'W_HIGH'
EPOT 0.0
ALPHA FLUX 0.0

```

A complete sample command file is found in App. D.

Table 2: A list of scalar model parameters, set using MHD model options read from the end of the command file.

Model parameter	Description
PERM	Magnetic permeability, μ_m [Vs/Am].
COND	Electrical conductivity, σ [$\Omega^{-1}m^{-1}$].
DENS	Fluid density, ρ [kg/m^3].
VISC	Fluid dynamic viscosity, μ [$kg/m/s$].
CA1	MHD model constant, $C_{\alpha 1}$.
CA2	MHD model constant, $C_{\alpha 2}$.
CA3	MHD model constant, $C_{\alpha 3}$ (currently not used).
CA4	MHD model constant, $C_{\alpha 4}$ (currently not used).
CEA	MHD model constant, $C_{\varepsilon\alpha}$.
CAMOD	Moderator coefficient for Joule dissipation source terms in turbulence equations (CAMOD=1 by default).
CLF	Moderator coefficient for Lorentz force in mean momentum equations (CLF=1 by default).
ALPHA	Constant initial value for ALPHA (default is ALPHA=0.33).

A.2 MHD model options

The *CFX* frontend stops reading the command file when it reaches the stop command,

```
>>STOP
```

This allows us to control certain features of the MHD model implementation, by specifying a number of MHD model options at the end of the *CFX* command file.

Table 2 is a description of scalar model parameters that can be set. The model option is written on one line, and its new value on the following. The following lines, for example, sets the viscosity, electric conductivity and density to that of mercury:

```
COND
1.05E+6
DENS
13550.0
VISC
0.00156
```

Table 3 shows a list of option for defining the external magnetic field. The following lines, for example, indicate that a magnetic field is specified in user fortran (MHDBXT), but it should be multiplied by two:

```
B_FORT
BSCALE
2.0
```

Table 3: Model options for defining the external magnetic field.

Model option	Description
B0	Sets a uniform magnetic field, with all three components given on the following line.
B_FOR	The magnetic field is specified in subroutine MHD-BXT (must be supplied by the user.
B_FILE	The magnetic field should be read from file 'B.dat' in the problem directory. (See next option.)
XYZ	Causes the code to print a file 'xyz.dat' in the problem directory, containing XYZ coordinates of all cell centers. The magnetic field in 'B.dat' should be in this order.
BSCALE	The magnetic field read from file, or defined in MHD-BXT, is multiplied by this factor before use. (BSCALE=1.0 by default.)

Table 4 describes model options used to modify the solution of the EPOT and ALPHA equations. The following example prevents currents through inlets and outlets, specifies a zero reference point for the EPOT equation (in the origin), and disables the mean strain terms in the ALPHA equation:

```

INSULATED_INLETS
INSULATED_OUTLETS
EPOT_REFERENCE_POSITION
  0.0      0.0      0.0
NO_STRAIN_IN_ALPHA

```

B Documentation of fortran source code

Table 5 shows a list of fortran subroutines. The grouping and indentation is used as a rough illustration of dependencies and calling order. All routines starting with “USR” are standard *CFX* user subroutines, while those starting with “MHD” are used only for implementation of the MHD models.

The *CFX* routines USRINT, USRBF and USRSRC contain only calls to the corresponding routines MHDINT, MHDBF and MHDSRC, respectively. The remaining routines are discussed individually below.

The common block UC_MHD is used to simplify the transfer of common variables and model parameters between the subroutines of the model implementation. Most of the parameters are initialized in the subroutine MHDINT (based on MHD model options read at the end of the *CFX* command file), although some of the parameters are given useful defaults in the block data subprogram MHD_DATA. The parameters carried in the common block are listed in Table 6.

Table 4: Model options for modifying the solution of the EPOT and ALPHA equations.

Model option	Description
INSULATED_INLETS INSULATED_OUTLETS	Causes inlet and outlet patches, respectively, to be electrically insulated. (Affects the source term and solution matrix in the EPOT equation, and the calculation of total current.)
NO_STRAIN_IN_ALPHA	DISABLES the strain-related model terms in the ALPHA equation. The terms are algebraically complex, and requires calculation of mean velocity gradients. The model terms give inaccurate results outside the magnetic field, and slows down the calculation. The effect of the terms appears to be small inside a magnetic field.
EPOT_REFERENCE_POSITION	The next line should contain the XYZ coordinates for a point where EPOT will be forced to zero. This is necessary unless a fixed value of EPOT is specified on any of the boundaries (otherwise EPOT is undefined!).

B.1 MHDINT

Called from USRINT. Reserves permanent workspace for currents at cell faces (computed in MHDJSF and MHDJSC), checks which turbulence model is used, and initializes common block pointers to user scalars.

Reads a number of MHD model options from the end of the command file, starting after the *CFX* “>>STOP” command. See App. A.2 for a description of model options.

Depending on model options, the magnetic field is either specified as constant and uniform, read from the file “B.dat” in the problem directory, or a call is made to the subroutine MHDBXT. One option causes the routine to print Cartesian coordinates for all nodes to a file “xyz.dat”, in the order B-field values should appear in “B.dat”.

In case all inlets, outlets and walls are non-conducting, there is an option for specifying the coordinates of a zero reference point for the electrostatic potential EPOT. If the given point is not in the flow domain, the center of the first block is used instead.

B.2 MHDBXT

Called from MHDINT. This routine can be used to specify the magnetic field in all nodes, e.g. as function of position. Even if the routine is not used, a dummy version should still be available at compilation time. The following example sets a uniform field $B = 1$ in the z direction.

Table 5: List of fortran subroutines, grouped and indented to show calling order.

Subroutines	Purpose
USRINT	(Call MHDINT.)
MHDINT	Read MHD model options from the end of the <i>CFX</i> command file. Initialize model parameters. Call MHDBXT, if magnetic field is defined in user fortran.
MHDBXT	Definition of magnetic field in all nodes (optional).
USRBF	(Call MHDBF.)
MHDBF	Compute total current density and Lorentz force.
MHDJSC	Compute contravariant component of $\mathbf{U} \times \mathbf{B}$ at cell faces (for body-fitted grids).
MHDGEP	Compute gradient of EPOT.
MHDGE2	Recompute EPOT gradient in a single cell next to a boundary.
USRDIF	Set diffusivity for EPOT equation explicitly.
USRWTM	Set turbulent wall multiplier for EPOT explicitly, to override turbulent wall function.
USRSRC	(Call MHDSRC.)
MHDSRC	Compute source terms for scalar and turbulence equations.
MHDJSF	Compute normal flux components of $\mathbf{U} \times \mathbf{B}$ on all cell faces.
MHDSEP	Compute source term for the EPOT equation. Modify matrix coefficients at certain boundaries.
MHD_DATA	BLOCK DATA subprogram. Assigns default values for some parameters in the UC_MHD common block.

```

SUBROUTINE MHDBXT(XP,YP,ZP,SCAL,IBX,IBY,IBZ,BSCALE,BMAX
+
,NNODE,NPHASE,NSCAL)
C
C   Routine for specification of external B-field.
C
C   DIMENSION
+   SCAL(NNODE,NPHASE,NSCAL)
+   ,XP(NNODE),YP(NNODE),ZP(NNODE)
C
C-----
C
C   B-field should be specified below for all nodes, e.g.
C   as a function of node coordinates XP, YP and ZP!
C
C   Maximum norm of B should be stored in BMAX!

```

Table 6: Contents of UC_MHD common block.

Parameter	Type	Description
PERM	real	Magnetic permeability, μ_m [Vs/Am].
COND	real	Electrical conductivity, σ [$\Omega^{-1}m^{-1}$].
DENS	real	Fluid density, ρ [kg/m^3].
VISC	real	Fluid dynamic viscosity, μ [$kg/m/s$].
CA1	real	MHD RST model constant, $C_{\alpha 1}$.
CA2	real	MHD RST model constant, $C_{\alpha 2}$.
CA3	real	MHD RST model constant, $C_{\alpha 3}$ (currently not used).
CA4	real	MHD RST model constant, $C_{\alpha 4}$ (currently not used).
CEA	real	MHD RST model constant, $C_{\varepsilon\alpha}$.
CAMOD	real	Moderator coefficient for Joule dissipation source terms in turbulence equations (CAMOD=1 by default).
CLF	real	Moderator coefficient for Lorentz force in mean momentum equations (CLF=1 by default).
ISCL	integer	Variable number for SCAL array (from GETVAR).
IEP	integer	Scalar number for EPOT variable (from GETSCA).
IALPHA	integer	Scalar number for ALPHA variable (from GETSCA).
IB(1:3)	integer	Scalar numbers for stored B -field components.
IJ(1:3)	integer	Scalar numbers for stored total current components.
IJSF	integer	Pointer to facial currents in WORK array.
INDEPR	integer	INODE number for EPOT zero reference point.
LINSIN	logical	Logical TRUE to avoid currents through INLET boundaries.
LINSPR	logical	Logical TRUE to avoid currents through PRESSURE boundaries.
LINSMF	logical	Logical TRUE to avoid currents through MASS-FLOW boundaries.
LEPREF	logical	Logical TRUE if an EPOT zero reference points is defined.
LNOSTR	logical	Logical TRUE to disable mean strain terms in ALPHA equation.
LRSM	logical	Logical TRUE indicates that the Reynolds stress model is used. If FALSE, we assume the high-Re K - ε model is specified.

```

C
C -----
      DO 100 INODE=1,NNODE
C
      BX = 0.0
      BY = 0.0
      BZ = 1.0
C
      SCAL(INODE,1,IBX)=BX
      SCAL(INODE,1,IBY)=BY
      SCAL(INODE,1,IBZ)=BZ
      BMAX=MAX(BMAX,SQRT(BX**2+BY**2+BZ**2))
C
100  CONTINUE
C-----
      RETURN
      END

```

The factor `BSCALE` can be specified in the command file, so that the magnitude of the magnetic field can be scaled without recompiling the fortran routines.

B.3 USRDIF

Explicitly sets the total diffusivity for the EPOT equation equal to the electric conductivity `COND` stored in the `UC_MHD` common block. This is necessary to override the turbulent diffusivity added by `CFX`. (Note that the diffusivity can be scalar or tensorial, depending on what turbulence model is used.)

B.4 USRWTM

For the EPOT equation, sets the turbulent wall multiplier `TMULT` for each wall boundary node to

$$TMULT = \frac{COND}{YWALL}, \quad (73)$$

where `COND` is the electric conductivity, and `YWALL` the distance to the first node. This overrides the effect of the turbulent wall function in `CFX`, and ensures natural wall boundary conditions for the EPOT equation.

B.5 MHDSRC

Called from `USRSRC`, for both `ICALL=1` and `ICALL=2`.

For the EPOT equation, and if `ICALL=1`, `MHDJSF` is called to compute normal flux components of the term $\sigma(\mathbf{U} \times \mathbf{B})$ on all cell faces. Then `MHDSEP` is called to compute the source term for EPOT. `MHDSEP` is called also if `ICALL=2`, for modification of `SU`, `SP`, and `AM` to account for boundary conditions.

`MHDSRC` further computes magnetic source terms for all turbulence equations, including `ALPHA`. The magnetic source terms in all turbulence equations are multiplied by the factor `CAMOD`. By default `CAMOD=1`, but by setting it to zero with the MHD model options in the command file, all magnetic source terms are disabled.

If the mean strain terms in the ALPHA equation are to be computed, the mean velocity gradients are computed with the *CFX* routine GRADV. The different ALPHA source term contributions are currently stored in user scalars (see Sec. 2.5):

$$\text{S_MAG} = -\mu_\alpha, \quad (74)$$

$$\text{S_RET} = \pi_\alpha, \quad (75)$$

$$\text{S_PRD} = P_\alpha, \quad (76)$$

$$\text{S_PST} = \Pi_\alpha^{(r)}. \quad (77)$$

B.6 MHDJSF

Called from MHDSRC. Computes and stores the normal flux components $\sigma(\mathbf{U} \times \mathbf{B})_f \cdot \mathbf{A}_f$ on all cell faces.

In the interior of the flow domain, facial values of \mathbf{U} and \mathbf{B} are interpolated using the *CFX* weight factors WFACT. We assume no-slip boundaries, so that $\mathbf{U} = 0$ on all walls. The relevant velocity at inlet, pressure and massflow boundaries are taken from the dummy nodes at the boundary. If the command file options for insulated boundaries are specified, however, the normal flux is set to zero at the respective boundaries.

B.7 MHDSEP

Called from MHDSRC for both ICALL=1 and ICALL=2.

If ICALL=1, MHDSEP loops over all cells and computes the source term for EPOT from the facial source currents computed in MHDJSF.

If ICALL=2, and if the relevant insulated boundary option is specified, the matrix coefficient AM in the direction of the boundary is set to zero in all active cells adjacent to the boundary.

If a zero reference point for EPOT is specified, MHDSEP sets SU=0 for this node, SP to a large negative number, and all its matrix coefficients to zero. This forces EPOT to zero in the node.

B.8 MHDBF

Called from USRBF. Computes the total current density and the Lorentz force in all cells. The total current density is stored in user scalars.

The total current density is given by

$$\mathbf{J} = -\sigma \nabla \Phi + \sigma \mathbf{U} \times \mathbf{B}. \quad (78)$$

In the first part of the routine, MHDGEP is called to compute the gradient of EPOT. For the $\mathbf{U} \times \mathbf{B}$ term to behave well together with the gradient term, we need to compute $\mathbf{U} \times \mathbf{B}$ in the cell center as the average of the contravariant components on the faces of the cell. The procedure is similar to that used for body forces in *CFX*, to make these compatible with gradients of pressure.

If the grid is rectangular, the contravariant components of $\mathbf{U} \times \mathbf{B}$ can be expressed in terms of the normal flux components already computed by MHDJSF for the EPOT source terms. Note, however, that MHDJSF is called from MHDBF on the first iteration, if it is a restart from an existing dump file. This

is because the momentum equations are the first variables in the outer iteration, and no call has yet been made to MHDJSF from MHDSRC.

If the grid is body-fitted (and presumed non-orthogonal), we call MHDJSC to compute and store the contravariant components of $\mathbf{U} \times \mathbf{B}$ on the faces.

The Lorentz force is multiplied by the factor CLF before it is added to the body force array. By default CLF=1, but by setting it to zero with the MHD model option in the command file, the Lorentz force can be disabled.

B.9 MHDJSC

Called from MHDBF. Computes and stores the contravariant components of $\mathbf{U} \times \mathbf{B}$ on the faces. (These components overwrite the normal flux components previously stored by MHDJSF.) The general approach, and the treatment of boundary conditions, is similar to that in MHDJSF (see App. B.6).

B.10 MHDGEP and MHDGE2

Called by MHDBF. Calls the *CFX* routine GRADS to compute the gradients of EPOT. The gradients must however be modified in the cells adjacent to walls, inlets, pressure and massflow boundaries. The main reason for this is that the normal gradient at walls is incomplete, as the wall gradient of a scalar usually depends on a turbulent wall function.

The modification at boundaries is made in two steps. First, the gradient in cells adjacent to wall, inlet, pressure and massflow boundaries are marked as undefined, by setting its x component to a large dummy value (larger than 10^{10}). In the next step, we loop again over cells adjacent to the different boundary types. The new gradient in the node P is defined as

$$\left(\frac{\partial\Phi}{\partial\mathbf{x}}\right)_P = \frac{1}{V_P} \sum_f \Phi_f \mathbf{A}_f, \quad (79)$$

where V_P is the cell volume, Φ_f the value of EPOT at the cell faces f , and \mathbf{A}_f is the face area vectors. This equation follows from the generalized Gauss law (see the *CFX* manual, Sec. 13.2). If the gradient is already defined (will happen if the cell is adjacent to more than one boundary), the contribution from the current boundary is modified with the appropriate value of EPOT at the current face.

The calculation of the gradient (79) in an individual cell is performed in the routine MHDGE2, which is called from MHDGEP with the appropriate EPOT value for the current boundary face.

The EPOT value for a face at a wall is taken from the active cell. This corresponds to the desired zero normal gradient at the wall. The default behavior for an inlet boundary is to take the EPOT value from the boundary node, where the boundary conditions are stored. If the insulated inlet option is active, the value is instead taken from the active cell (zero normal gradient). The default behavior for pressure and massflow boundaries is to use the EPOT value from the dummy node. The value in the dummy node is equal to the value in the active cell on outflow (Neumann boundary conditions), but equal to the Dirichlet condition specified at the patch in case of inflow. If the insulated outlet option is active, the value in the active node is used instead (zero normal gradient).

C MAPLE script for generating source terms

Below is a listing of a MAPLE script for generating the fortran source for the source terms in the ALPHA equation. A similar procedure is used for the source terms in the Reynolds stress equations.

```

restart:with(linalg):with(plots):
#-----
# Load definitions for using the function 'sumt'
# to manipulate tensors in index notation
# (from Torbj\{o}rn Sj\{o}gren, Doctoral Thesis, KTH, 1997,
# TRITA-MEK 1997:5, ISSN 0348-467X)
read('/home/mech/olwi/maple/twopoint2.1olwi');
read('/home/mech/olwi/rdt/Mmod/tensorfunc2.0');
#-----
# Define model functions, and
# substitutions for CFX fortran variable names
#-----
f1:=-9/5*(1-alpha):
scalfunc:={G=alpha-3*alpha^2*f1/2/2*(Ina+2/3),
           H=3*alpha^2*f1/2,
           aN=CA3*(-alpha+4*alpha^2-3*alpha^3)
           +13/10*alpha^2-3/10*alpha^3 }:
namesubs:={R.1.1=UU,R.2.2=VV,R.3.3=WW,
           R.1.2=UV,R.2.3=VW,R.1.3=WU}:
fortsubs:={R.1.1=RS[INODE,IPH,1],R.2.2=RS[INODE,IPH,2],
           R.3.3=RS[INODE,IPH,3],R.1.2=RS[INODE,IPH,4],
           R.2.3=RS[INODE,IPH,5],R.1.3=RS[INODE,IPH,6],
           alpha=ALPHA,
           n.1=DUV[1],n.2=DUV[2],n.3=DUV[3],
           K=TE[INODE,IPH],eps=ED[INODE,IPH],ttuin=TTUINV,
           Ug11=UGRX,Ug12=UGRY,Ug13=UGRZ,
           Ug21=VGRX,Ug22=VGRY,Ug23=VGRZ,
           Ug31=WGRX,Ug32=WGRY,Ug33=WGRZ}:
EVMSubs:={seq(seq(R.i.j=2/3*K*delta[i,j]-nuT*(Ug[i,j]+Ug[j,i]),
                 j=1..3),i=1..3)}:
#-----
# Define tensors and tensorial groups
#-----
R:=array(symmetric,1..3,1..3):
n:=array(1..3,[n1,n2,n3]):
Ug:=array(1..3,1..3):
#-----
# Set variable properties for linearization:
# - Diagonal R_ij: > 0
# - Off-diagonal R_ij: REAL
# - Kinetic energy: K>0
# - Dissipation: eps>0
# - Alpha: 0<=alpha<=1
# - direction unit vector, all components: -1 <= n_k <= 1
# - CMU and CA3 >0
#-----
for i from 1 to 3 do
  assume(n.i>=-1,n.i<=1);
  for j from i to 3 do
    if (i=j) then
      assume(R.i.j>0);
    else
      assume(R.i.j,real);
    fi;
    R[i,j]:=R.i.j;
  end do;
end do;

```

```

    od;
od;
for i from 1 to 3 do
  for j from 1 to 3 do
    assume(Ug.i.j,real);
    Ug[i,j]:=Ug.i.j;
  od;
od;
assume(alpha>=0,alpha<=1);
assume(K>0);
assume(TELIM>0);
assume(eps>0);
assume(EDLIM>0);
assume(CA3>0);
assume(CMU>0);
a:=evalm(sumt(R[i1,i2]/K-2/3*delta[i1,i2],0,2)):
Ina_:=simplify(sumt(n[s1]*n[s2]*a[s2,s1],2,0)):
InaRST:=expand(subs(K=TELIM,Ina_)):
InaEVM:=simplify(subs(nuT=CMU*K^2/EDLIM,subs(EVMsubs,Ina_))):
#
#=====
# Derived tensors, groups and invariants
#-----
# Production term, P_alpha
#-----
prod1:=sumt(-2*G*Ug[s1,s2]*R[s1,s2],2,0):
prod2:=sumt(-H*Ug[s1,s2]*n[s1]*n[s3]*R[s3,s2],3,0):
prod3:=sumt(-H*Ug[s1,s2]*n[s2]*n[s3]*R[s3,s1],3,0):
prodK:=sumt(-Ug[s1,s2]*R[s1,s2],2,0):
Prod:=simplify((prod1+prod2+prod3-2*alpha*prodK)/(2*TELIM)):
#-----
# Production term in eddy viscosity model:
#   prod1 & prodK: subs K=0 to cancel K-term
#   (drops out due to continuity)...
#-----
prod1EVM:=subs(K=0,subs(EVMsubs,prod1)):
prod2EVM:=subs(EVMsubs,prod2):
prod3EVM:=subs(EVMsubs,prod3):
prodKEVM:=subs(K=0,subs(EVMsubs,prodK)):
ProdEVM:=simplify(subs(nuT=CMU*K^2/EDLIM,
  (prod1EVM+prod2EVM+prod3EVM-2*alpha*prodKEVM)/(2*K))):
#-----
# Pressure-strain term, Pi_alpha:
#-----
Pi_:=simplify(sumt(2*(aN-alpha)*Ug[s1,s2]*n[s1]*n[s2],2,0)):
#
#=====
# Compute source terms and print fortran to file
# Linear terms put in SP if always negative. Otherwise in SU.
#-----
writeto('alpha_source.f');
#-----
# Production term:
#-----
x:=expand(subs(Ina=InaRST,subs(scalfunc,Prod))):
# Prepare linearization of source term...
su0:=coeff(x,alpha,0):
sp0:=expand(simplify((x-su0)/alpha)):
# -- Only negative linear part in SP, rest in SU...
# -- Check whether polynomial, or product/nil...
if (whattype(sp0)='+') then
  sp:=select(t->is(t<=0),sp0):

```

```

    su:=su0+alpha*remove(t->is(t<=0),sp0):
else
  if (is(sp0<=0)) then
    sp:=sp0:
  else
    su:=su0+alpha*sp0:
  fi:
fi:
# -- Simplify...
su:=simplify(su):
sp:=simplify(sp):
lprint('C -----'):
lprint('C   Production (RST version)'):
lprint('C -----'):
fortran([SU[INODE,IPH]=SU[INODE,IPH]+DENS*VOL(INODE)*subs(fortsubs,su),
        SP[INODE,IPH]=SP[INODE,IPH]+DENS*VOL(INODE)*subs(fortsubs,sp)],
        optimized):
#-----
# Production term in EV model:
#-----
x:=expand(subs(Ina=InaEVM,subs(scalfunc,ProdEVM))):
# Prepare linearization of source term...
su0:=coeff(x,alpha,0):
sp0:=expand(simplify((x-su0)/alpha)):
# -- Only negative linear part in SP, rest in SU...
# -- Check whether polynomial, or product/nil...
if (whattype(sp0)='+') then
  sp:=select(t->is(t<=0),sp0):
  su:=su0+alpha*remove(t->is(t<=0),sp0):
else
  if (is(sp0<=0)) then
    sp:=sp0:
  else
    su:=su0+alpha*sp0:
  fi:
fi:
# -- Simplify...
su:=simplify(su):
sp:=simplify(sp):
lprint('C -----'):
lprint('C   Production (EVM version)'):
lprint('C -----'):
fortran([SU[INODE,IPH]=SU[INODE,IPH]+DENS*VOL(INODE)*subs(fortsubs,su),
        SP[INODE,IPH]=SP[INODE,IPH]+DENS*VOL(INODE)*subs(fortsubs,sp)],
        optimized):
#-----
# Pressure-strain term:
#-----
x:=expand(subs(scalfunc,Pi_)):
# Prepare linearization of source term...
su0:=coeff(x,alpha,0):
sp0:=expand(simplify((x-su0)/alpha)):
# -- Only negative linear part in SP, rest in SU...
# -- Check whether polynomial, or product/nil...
if (whattype(sp0)='+') then
  sp:=select(t->is(t<=0),sp0):
  su:=su0+alpha*remove(t->is(t<=0),sp0):
else
  if (is(sp0<=0)) then
    sp:=sp0:
  else
    su:=su0+alpha*sp0:

```

```

      fi:
fi:
# -- Simplify...
su:=simplify(su):
sp:=simplify(sp):
lprint('C -----'):
lprint('C   Pressure-strain interaction'):
lprint('C -----'):
fortran([SU[INODE,IPH]=SU[INODE,IPH]+DENS*VOL(INODE)*subs(fortsubs,su),
        SP[INODE,IPH]=SP[INODE,IPH]+DENS*VOL(INODE)*subs(fortsubs,sp)],
        optimized):
writeto('terminal');

```

D Listing of sample command file

```

>>CFX4
>>SET LIMITS
  TOTAL INTEGER WORK SPACE 14000000
  TOTAL CHARACTER WORK SPACE 1500
  TOTAL REAL WORK SPACE 90000000
  MAXIMUM NUMBER OF BLOCKS 10
  MAXIMUM NUMBER OF PATCHES 50
  MAXIMUM NUMBER OF INTER BLOCK BOUNDARIES 20
>>OPTIONS
  TURBULENT FLOW
  TRANSIENT FLOW
  USER SCALAR EQUATIONS 12
>>USER FORTRAN
  USRSRC
  USRBF
  USRINT
  USRWTM
  USRDIF
>>VARIABLE NAMES
  USER SCALAR1 'EPOT'
  USER SCALAR2 'ALPHA'
  USER SCALAR3 'USRD B_X'
  USER SCALAR4 'USRD B_Y'
  USER SCALAR5 'USRD B_Z'
  USER SCALAR6 'USRDCC J_X'
  USER SCALAR7 'USRDCC J_Y'
  USER SCALAR8 'USRDCC J_Z'
  USER SCALAR9 'USRDCC S_MAG'
  USER SCALAR10 'USRDCC S_RET'
  USER SCALAR11 'USRDCC S_PRD'
  USER SCALAR12 'USRDCC S_PST'
  END

/* - - - - Parameters - - - - */
#CALC
  DENS = 13550.0;
  VISC = 0.00156;
  COND = 1.05E+6;
#ENDCALC
/* - - - - - - - - - - - - - - */

>>MODEL DATA
>>DIFFERENCING SCHEME
  EPOT 'NO CONVECTION'
>>PHYSICAL PROPERTIES

```

```

>>TRANSIENT PARAMETERS
  >>FIXED TIME STEPPING
    TIME STEPS 30*0.010
  >>QUASISTATIC VARIABLES
    EPOT
  >>FLUID PARAMETERS
    VISCOSITY #VISC
    DENSITY #DENS
    END
  >>SCALAR PARAMETERS
  >>DIFFUSIVITIES
    EPOT #COND
ALPHA #VISC
  >>TURBULENCE PARAMETERS
  >>TURBULENCE MODEL
    TURBULENCE MODEL 'DIFFERENTIAL STRESS'
/* TURBULENCE MODEL 'K-EPSILON' */
  >>TURBULENT PRANDTL NUMBER
    ALPHA 1.0 /* Same as for Reynolds stresses */

>>SOLVER DATA
  >>PROGRAM CONTROL
    MAXIMUM NUMBER OF ITERATIONS 1200
    ITERATIONS OF TURBULENCE EQUATIONS 3
    PRESSURE REFERENCE POSITION 0.452 0.02 0.015
    MASS SOURCE TOLERANCE 1.0E-6
  >>UNDER RELAXATION FACTORS
    U VELOCITY 0.4
    V VELOCITY 0.4
    W VELOCITY 0.4
    BFX 0.2
    BFY 0.2
    BFZ 0.2
    K 0.3
    EPSILON 0.3
    UU 0.3
    VV 0.3
    WW 0.3
    UV 0.3
    VW 0.3
    WU 0.3
    ALPHA 0.3
    EPOT 1.0 /* NOTE! */
  >>SWEEPS INFORMATION
  >>MAXIMUM NUMBER
    EPOT 50
    PRESSURE 200
  >>REDUCTION FACTORS
    EPOT 0.10
    PRESSURE 0.10

>>MODEL BOUNDARY CONDITIONS
  >>SET VARIABLES
    PATCH NAME 'OUT'
    PRESSURE 5000.0
  >>SET VARIABLES
    PATCH NAME 'IN'
    NORMAL VELOCITY 1.15
    K 2.0E-5
    EPSILON 9.9E-6
    EPOT 0.0
    ALPHA 0.333

```

```

/* - - - - - */
>>WALL BOUNDARY CONDITIONS
  PATCH NAME 'W_LOW'
  EPOT FLUX  0.0
  ALPHA FLUX 0.0
>>WALL BOUNDARY CONDITIONS
  PATCH NAME 'W_HIGH'
  EPOT FLUX  0.0
  ALPHA FLUX 0.0
>>WALL BOUNDARY CONDITIONS
  PATCH NAME 'W_SOUTH'
  EPOT FLUX  0.0
  ALPHA FLUX 0.0
>>WALL BOUNDARY CONDITIONS
  PATCH NAME 'W_NORTH'
  EPOT FLUX  0.0
  ALPHA FLUX 0.0
/* - - - - - */

>>OUTPUT OPTIONS
>>PRINT OPTIONS
  >>WHAT
    NO VARIABLES
    WALL PRINTING
  >>WHEN
    FINAL SOLUTION

>>STOP

% - - - MHD model options follow - - - - -
%
%
% On a COMPAQ mahine, the components of vectors must lie
% within these marks:
%      |           |           |
COND
1.05E+6
PERM
1.25664E-6
DENS
13550.0
VISC
0.00156
CEA
1.5
EPOT_REFERENCE_POSITION
  0.452      0.020      0.015
INSULATED_INLETS
INSULATED_OUTLETS
NO_STRAIN_IN_ALPHA
B_FORT
BSCALE
1.35

```