User Guide for Global RANS (GRANS) solver

Matteo Puccioni;
mxp174930@utdallas.edu
05/14/2018

1. Introduction

The present code is designed to evaluate velocity and pressure fields around one or more wind turbines rotating with known angular velocity. To do that, a Reynolds Averaged Navier Stokes (RANS) approach is used to obtain the mean value of the flow quantities once that some conditions on the domain boundaries are provided.
The inputs of the code are solely geometric and they must be selected by the user according to which situation must be simulated; beside that, the turbulence model (necessary in a RANS approach to close the problem) must be decided *a priori* and it may be chosen between an Eddy Viscosity (EV) model, a Mixing Length (ML) model or a more complex Turbulent Kinetic Energy (TKE) model.
Whatever are the initial settlings chosen for a specific simulation, the code will return the time-averaged velocity and pressure fields over the selected domain, eventually along with the development of the TKE. Beyond the validation of the mean quantities (mass and axial momentum balance), the streamwise velocity deficit and the power and thrust coefficients represent the main global outputs of the program, evaluated in a dedicated part at the end of the procedure.
From the numerical point of view, final solution is approached in an iterative way starting from a divergence-free velocity field and solving for the updated solution through a pseudo-spectral Collocation method. The typical time required to obtain the converged field is 15 minutes on a commercial PC.

2. Solving strategy

Let us build a cylindrical domain around the turbine having the streamwise direction coincident with the turbine axis; even though it is customary to select the inlet and outlet positions, the former is usually placed upstream the rotor and the latter in the wake of the turbine. From here to the following, the reference frame is indicated by: (r, θ, x), where r stands for the spanwise direction (parallel to the turbine blade), θ is the azimuthal position (along which the turbine is rotating) and x is the direction of the incoming flow, assumed parallel to the rotor axis.
The flow around the turbine is assumed to be incompressible, viscid and axial-symmetric (i.e. no dependence on θ is carried on). With the same notation as

before, mean velocity components are organized as follows: $\boldsymbol{U}\,(r,x) = (U_r, U_\theta, U_x)$ and made dimensionless with the incoming speed, U_∞. In the same way, mean pressure, $p\,(r,x)$ is made non-dimensional via ρU_∞^2 where ρ is the (constant) fluid density. Under these assumptions, governing equations are:

$$\nabla \cdot \boldsymbol{U} = 0;$$

$$\boldsymbol{U} \cdot \nabla \boldsymbol{U} = -\nabla p + \left(\frac{1}{Re} + \nu_T\right) \nabla^2 \boldsymbol{U} + \nabla \nu_T \cdot \nabla \boldsymbol{U} + \boldsymbol{F}(\boldsymbol{U}, r, x).$$

In previous equations, $Re = \frac{DU_\infty}{\nu} \simeq 10^5$ is the Reynolds number (D is the turbine diameter), $F\,(U,r,x)$ is the force caused by the turbine and $\nu_T(U,r,x)$ is the Eddy Viscosity (EV). This quantity can be evaluated in one of the following three ways:

- **Imposed by the user**: if the EV is known, it might be kept fixed throughout all the iterative procedure. In this case, it is evaluated as a function of the sole x coordinate through its mean value and a constant slope.

- **Mixing Length** (ML) model: in this case, EV is related to the mean flow through an algebraic model: $\nu_T(x,r) = l_m(x) \cdot (2S{:}S)^{0.5}$. In this equation, $l_m(x)$ stands for the ML (assumed as a function only of the streamwise location) and S is the mean velocity stress tensor. Specifically, ML is modeled through a 4P-logistic shape function:

  $$l_m(x) = \frac{l_\infty}{\{1 + \exp[-k(x - x_{tr})]\}}, x \geq x_{tr}$$

  $$l_m(x) = 0 \text{ otherwise.}$$

  For the present work, $l_\infty = 0.03$ (asymptotic value), $k = 0.3142$ (initial slope) and $x_{tr} = x_T$ where x_T is the turbine location.

  In case of multiple turbines, the trend reported in Figure 1 (said l_{m1}) is used to evaluate the turbulence development for the waked turbines (see Iungo *et al*. Data-driven RANS solver for low-computational cost simulations of wind turbine wakes, *Wind Energy*, 2010):

  $$l_{m,i} = \left(k l_{m,i}^{inc} + 1\right) l_{m,1}.$$

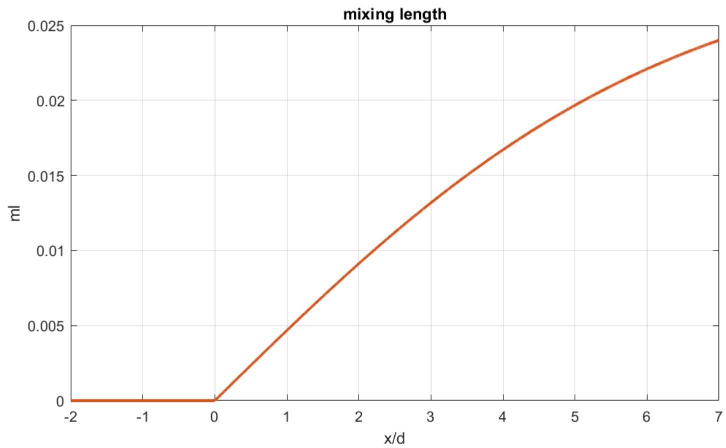

Figure 1 - 4-P Logistic Mixing Length for a single turbine located at x=0.

In the previous equation, $l_{m,i}^{inc}$ stands for the ML evaluated 0.3D upstream of the turbine and *k*=59.3 is an amplification factor calibrated against an assessment with LES data.

- **Turbulent Kinetic Energy** (TKE) model: here a differential equation governing the TKE is involved; indicating the latter by k, firstly it is related to the EV by: $\nu_T\,(r,x) = 0.5 l_m(x) \cdot k(r,x)^{0.5}$; then, k is governed by:

$$\boldsymbol{U} \cdot \nabla k = P - \varepsilon - \nabla \cdot \boldsymbol{T}'.$$

On the right-hand side, the three terms represent respectively:

  - TKE production: $P = -\overline{u_i u_j}\dfrac{\partial U_i}{\partial x_j}$;
  - TKE dissipation: $\varepsilon = \dfrac{k^{3/2}\, c_\mu^{3/4}}{l^*}$ where c_μ and $l^* = 0.03$ are constants;
  - TKE transportation: $\boldsymbol{T}' = -\dfrac{\nu_T}{\sigma_k}\nabla k$, where $\sigma_k \simeq 1$ is the turbulent Prandtl's constant.

The rotor's force per unit length on the flow, indicated by $\boldsymbol{F}\,(\boldsymbol{U}, r, x)$ in the governing equations, is assumed to be done by two components: $\boldsymbol{F} = F_\theta\,(\boldsymbol{U}, r, x)\boldsymbol{e_\theta} + F_x\,(\boldsymbol{U}, r, x)\boldsymbol{e}_x$ representing respectively the swirl and the drag imposed on the fluid. Both are modeled through the Actuator Disk hypothesis, for which, provided the angular velocity ω, components are related to the mean flow by:

$$F_x\,(x,r) = 0.5\rho U_{rel}^2 C_{TH}(U,r) c(r) f_x(U,r) \cdot e^{-\frac{0.5(x-x_T)^2}{\sigma_x^2}}$$

$$F_\theta(x,r) = 0.5\rho U_{rel}^2 C_{TG}(U,r)c(r)f_\theta(U,r) \cdot e^{-0.5(x-x_T)}.$$

In both equations, U_{rel} is the velocity at turbine location seen from a reference frame rotating with the turbine, thus: $U_{rel}^2 = U_x^2 + (U_\theta - \omega r)^2$; then, $c(r)$ stands for the chord length (known from the blade's geometry) and f_x, f_θ are Prandtl's correction coefficients introduced to account for the losses at the tip and hub. Finally, C_{TH} and C_{TG} are respectively thrust and tangential coefficients evaluated through the local angle of attack of the flow.

To help the numerical convergence, F_x and F_θ are spread over the streamwise direction through a Gaussian-shaped function based on: $\sigma_x = 0.3D$ and $\sigma_\theta = 0.1D$. Finally, the turbine hub effect is included as a bluff body force per unit length:

$$F_\theta^{hub} = 0;$$
$$F_x^{hub} = 0.5\rho U_x^2(x_T - D, 0)C_D r_{hub}.$$

It must be noted that the axial velocity is evaluated one diameter upstream the rotor disk; in particular, $C_D = 1.1$ and $r_{hub} = 0.05D$. For each radial position, the local maximum between the blade and hub axial force is evaluated and spread on the streamwise neighborhood via Gaussian function.

3. Numerical features

Governing equations are solved numerically through a pseudo-Spectral collocation method over a bi-dimensional grid of points spaced through Gauss-Lobatto-Chebyshev (GLC) criterion. Once that the user has selected N_x nodes over x and N_r nodes on the spanwise direction, the program maps them on the chosen domain according to an East-to-West and North-to-South enumeration. To avoid spurious modes, velocity and momentum balance are evaluated on the GLC grid, while mass balance and pressure are computed on a staggered Gauss-Chebyshev (GC) map, made by $(N_x - 2) \times (N_r - 2)$ nodes. The vector of the unknowns is represented by velocity and pressure values (and TKE eventually) on the nodes. In this work, they are settled in this order: $\boldsymbol{q} = [\boldsymbol{U_r}\ \boldsymbol{U_\theta}\ \boldsymbol{U_x}\ \boldsymbol{p}\ \boldsymbol{k}]$. Each of them is a mono-dimensional vector of $N_x \times N_r$ entries (except for pressure, which is $(N_x - 2) \times (N_r - 2)$).

The boundary conditions necessary to close the problem are the following:

- Inlet: imposed velocity profile: $\frac{U_x}{U_\infty} = 1; U_r = 0; U_\theta = 0;\ k/U_\infty^2 = 10^{-4}$.
- Upper boundary: two suitable sets of boundary conditions are possible:

- Stress free: useful to simulate an open environment; $p - \left(\frac{1}{Re} + \nu_T\right)\frac{\partial U_r}{\partial r} = 0; \frac{\partial U_\theta}{\partial r} = 0; \frac{\partial U_x}{\partial r} = 0.$
- Slip: useful to simulate a solid wall: $U_r = 0; U_\theta = 0; \frac{\partial U_x}{\partial r} = 0.$

- Rotor axis: symmetry conditions: $U_r = 0; U_\theta = 0; \frac{\partial U_x}{\partial r} = 0.$
- Outlet: either homogeneous or non-homogeneous Neumann boundary conditions.

Once that the problem is closed, solution is found iteratively starting from the initial field and evaluating the L_2 residual error associated to the balance equations. Velocity and pressure fields are updated through the Newton-Raphson linearization of the governing equations on the grid nodes.
As last noticeable numerical feature, under relaxation factors are introduced to help the convergence. Specifically, two distinct scalar quantities (bounded between 0 and 1) are associated respectively to the turbine forcing and, eventually, to the TKE production term. It is a good practice to do not use the lowest values to not slow down the convergence; a minimum value for both could be assessed at 0.6.

4. List of the scripts

In the following, a list of the main scripts present in the folder "Single_Turbine" is reported along with a brief explanation of their role. In "Multiple_Turbines" the name and the aim are the same. In bolded character are reported the most important scripts

- Global: All the global variables, common to all the scripts, are grouped;
- Grid_plot: It returns the plot of the Gauss-Lobatto-Chebyshev grid where the global RANS is solved;
- Jacobian_validation_TKE: Validates the computation of the Jacobian matrix against a Finite Difference method;
- **Main_ADM**: Main script that must run to obtain mean flow quantities;
- Mom_balance_loc_standalone: Verifies the axial momentum balance over a cylinder with the outer basis moving downward;
- Post_processing: It returns all the plot about mean velocity and turbulence;
- **Power_coeff_standalone**: Given a certain range of TSR, it returns power and thrust coefficients associated, along with axial and tangential forcing along the turbine blade;
- **Pre_processing**: All the geometric information can be set here;

- Tke_balance_standalone: It verifies the TKE balance over a cylinder with the outer basis moving downward;
- TKE_plot_standalone: It returns the spanwise trend of TKE at some chosen axial locations.