ATLab Documentation

Turbulence and Boundary Layers Group | University of Hamburg

August 21, 2025

Contents

Preface

ATLab—an acronym for Atmospheric Turbulence Laboratory—is a set of tools whose aim is **to efficiently solve and analyze a particular set of governing equations with a controlled accuracy**. The accuracy can be controlled in different ways: comparing with analytical solutions, including linear stability analysis; grid convergence studies; balance of transport equations, like integral turbulent kinetic energy or local values at specific relevant locations (e.g., at the wall). Resolution can be measured by the ratio between the grid spacing Δx and the relevant small scales, like the Kolmogorov scale η or the thickness of the diffusion sub-layers next to the wall. For the compact schemes used here, typical values are $\Delta x/\eta \simeq 1-2$; larger values can lead to numerical instability because of the aliasing generated by the non-linear terms. Note that these schemes are non-monotone, but typical out-of-bounds deviations of conserved scalars are below $10^{-6} - 10^{-8}$ relative to the mean variations, and this error is therefore negligibly small compared to the typical error associated with the statistical convergence, of the order of $1-5\%$. The statistical convergence can be estimated by varying the sample size of the data set, e.g. varying the domain size along the statistically homogeneous directions.

The efficiency can be measured in different ways but, ultimate, it should be related with the computational time needed to understand a particular problem with a given accuracy, and so the importance of the controlled accuracy.

Making the code user-friendly comes after the previous two main priorities: controlled accuracy and efficiency.

Last, the main documentation is the code itself and the examples. This document is only a short introduction to the equations and the tools. The code is continuously under development, so this document is continuously incomplete.

Part I

User Guide

Chapter 1

Governing equations

The code is build to solve the sets of equations described in this section as efficiently as possible. These sets of equations aim to be general enough to cover several problems. The consequence is that one needs to map the particular problem to one of these generic cases, and identify the appropriate values of the parameters and defining functions (for instance, Re or b^e). Certain knowledge of the equations is therefore advantageous.

1.1 Evolution equations

1.1.1 Boussinesq

Dynamics and thermodynamics are at most coupled by the buoyancy field, and thermodynamics is not necessary (but can still be used). We consider the momentum equation and the evolution equations for an arbitrary number of scalar fields in the following form:

$$\partial_t v_i = -v_k \partial_k v_i + \partial_k \left(\mathrm{Re}^{-1} \nu \partial_k v_i\right) - \partial_i p' + \mathrm{Fr}^{-1} g_i \, b + \mathrm{Ro}^{-1} \, \epsilon_{ijk} f_k \, v_k \ , \qquad i = 1,2,3 \tag{1.1a}$$

$$\partial_t s_i = -v_k \partial_k s_i + \partial_k \left[(\mathrm{Re}\mathrm{Sc}_i)^{-1} \nu \partial_k s_i - \partial_k (\mathbf{j}_i)_k\right] + \omega_i \ , \qquad i = 1, \dots n \tag{1.1b}$$

The dynamic variables in these equations are nondimensionalized by ρ_0, U_0 and L_0. The parameters Re, Sc, Fr and Ro need to be provided.

- The scalar field ν represents the kinematic viscosity. In the simplest case, it is equal to 1.
- The scalar field b represents the buoyancy, if any. The vector (g_i) gives the direction of the buoyancy force and it is constant.
- The vector fields $\mathbf{j}_i$ represent flux terms, if any.
- The scalar fields ω_i represent source terms, if any.

These fields are defined in terms of the scalars s_k by functions $\nu^e(s_k)$, $b^e(s_k)$, $\mathbf{j}_i^e(s_k)$ and $\omega_i^e(s_k)$, to be given.

- The vector (f_i) gives the direction of the angular velocity of the frame of reference, if any. It is assumed constant.

Mass conservation,

$$\partial_k v_k = 0 \ , \tag{1.2}$$

is imposed in terms of the pressure-Poisson equation

$$\nabla^2 p' = \partial_k(\ldots) \tag{1.3}$$

with the boundary conditions that result from particularizing the momentum equation at the top and bottom boundaries. Details can be found in Mellado and Ansorge [2012].

Dimensional Formulation

A dimensional formulation can be considered by substituting the parameters `Reynolds`, `Froude` and `Rossby` the input file (by default, `tlab.ini`) with `Viscosity`, `Gravity` and `Coriolis`. The boundary and initial conditions defined in the input file should then be given in dimensional form.

1.1.2 Anelastic

Dynamics and thermodynamics are coupled by the density. The momentum equation is formulated in terms of the dynamic pressure and the density:

$$\partial_t v_i = -v_k \partial_k v_i + \rho_{\mathrm{bg}}^{-1} \partial_k \left(\mathrm{Re}^{-1} \mu \partial_k v_i\right) - \rho_{\mathrm{bg}}^{-1} \partial_i p' + \mathrm{Fr}^{-1} g_i (1 - \rho/\rho_{\mathrm{bg}}) + \mathrm{Ro}^{-1}\, \epsilon_{ijk} f_k \, v_k \ . \tag{1.4}$$

The evolution equations for the scalars read:

$$\partial_t s_i = -v_k \partial_k s_i + \rho_{\mathrm{bg}}^{-1} \partial_k \left[(\mathrm{ReSc}_i)^{-1} \mu \partial_k s_i - \partial_k (\mathbf{j}_i)_k\right] + \omega_i \ , \qquad i = 1, \ldots n \tag{1.5}$$

Symbols are as explained in the Boussinesq case. The scalar field μ represents the dynamic viscosity. In the simplest case, it is equal to 1.

Mass conservation,

$$\partial_k (\rho_{\mathrm{bg}} v_k) = 0 \ , \tag{1.6}$$

is imposed in terms of the pressure-Poisson equation

$$\nabla^2 p' = \partial_k (\rho_{\mathrm{bg}} \ldots) \tag{1.7}$$

with the boundary conditions that result from particularizing the momentum equation at the top and bottom boundaries

1.1.3 Compressible

Dynamics and thermodynamics are fully coupled. The pressure in the momentum equation is the thermodynamic pressure. Mass conservation is expressed in terms of the evolution equation

$$\partial_t \rho = -\partial_k (\rho v_k) \tag{1.8}$$

instead of a solenoidal constraint.

The variables in these equations are normalized by the reference scales L_0, U_0, ρ_0 and T_0, which represent a length, a velocity, a density, and a temperature, respectively. The pressure is normalized by $\rho_0 U_0^2$.

TBD

1.2 Thermodynamics

We consider n_c components (or species) with mass fractions ζ_i.

1.2.1 Compressible

The thermal equation of state is implemented as

$$p = \rho R T \; . \tag{1.9}$$

The scalar field

$$R = \sum_1^{n_\mathrm{c}} R_i \zeta_i \tag{1.10}$$

is the specific gas constant of the mixture.

The caloric equation of state is implemented as

$$h = \sum_1^{n_\mathrm{c}} h_i \zeta_i \; , \qquad h_i = \Delta h_i^0 + \int_{T_0}^{T} c_{pi}(T) dT \; . \tag{1.11}$$

Each species has a specific heat capacity c_{pi} and a specific gas constant $R_i = \mathcal{R}/W_i$, where $\mathcal{R}$ is the universal gas constant and W_i the molar mass of the species. They are nondimensionalized by the reference values c_{p0} and $R_0 = \mathcal{R}/W_0$, respectively, which are the values of one of the species.

The thermodynamic variables are nondimensionalized as in evolution equations, i.e., the reference scales L_0, U_0, ρ_0 and T_0, which represent a length, a velocity, a density, and a temperature, respectively. The pressure is normalized by $\rho_0 U_0^2$. Thermal energy variables are normalized with $c_{p0} T_0$, where c_{p0} is a reference specific heat capacity at constant pressure.

TBD

Dimensional Formulation

In this case of a multi-species, a dimensional formulation can be considered by setting the input parameter `nondimensional` equal to `.false.` in the block `[Thermodynamics]` of the input file (by default, `tlab.ini`).

1.2.2 Anelastic

The thermal equation of state is implemented as

$$p_\mathrm{bg} = \rho R T \; . \tag{1.12}$$

The thermodynamic variables are nondimensionalized by ρ_0, T_0 and R_0, such that $p_0 = \rho_0 R_0 T_0$. The default reference values are $p_0 = 10^5$ Pa and $T_0 = 298$ K. Thermal energy variables are normalized with $c_{p0} T_0$, where c_{p0} is a reference specific heat capacity at constant pressure.

The caloric equation of state is formulated in terms of the static energy

$$h = \sum_1^{n_\mathrm{c}} h_i \zeta_i + \frac{\gamma_0 - 1}{\gamma_0} H^{-1} (x_3 - x_{3,0}) \; . \tag{1.13}$$

The scalar field c is the specific heat capacity, a function of the scalars s_i. The parameter

$$H = \frac{R_0 T_0}{g L_0} \tag{1.14}$$

is a nondimensional scale height, or the inverse of a nondimensional gravity, to be provided. The parameter

$$R_0/c_{p0} = (\gamma_0 - 1)/\gamma_0 \ , \tag{1.15}$$

a conversion factor between gas constants and heat capacities (thermal equation of state and caloric equation of state). We refer to it in the code as `GRATIO`.

The background profiles $\{\rho_{\mathrm{bg}},\, p_{\mathrm{bg}},\, T_{\mathrm{bg}},\, \zeta_{i,\mathrm{ref}}\}$ correspond to a state of thermodynamic and hydrostatic equilibrium. The code solves the system of equations

$$\partial_3\, p_{\mathrm{bg}} = -H^{-1}\, g_3\, \rho_{\mathrm{bg}} \ , \qquad p_{\mathrm{bg}}|_{x_3 = x_{3,0}} = p_{\mathrm{bg},0} \ , \tag{1.16a}$$

$$p_{\mathrm{bg}} = \rho_{\mathrm{bg}} R_{\mathrm{bg}} T_{\mathrm{bg}} \ . \tag{1.16b}$$

$\mathbf{g}$ is defined opposite to the gravitational acceleration (the problem is formulated in terms of the buoyancy). The two equations above relate 4 thermodynamic variables and we need two additional constraints. Typically, we impose the background profile of static energy (enthalpy plus potential energy) and the composition. If there is only one species, then $R_{\mathrm{bg}} = 1$ and we only need one additional constraint.

Currently implemented only for air-water mixtures. In this case, the first scalar is the energy variable and the remaining scalars are the composition (e.g., total water specific humidity and liquid water specific humidity).

Dimensional Formulation

A dimensional formulation can be considered by setting the parameter `nondimensional` equal to `.false.`, which sets `GRATIO` equal to 1

1.2.3 Mixtures

We consider n_{c} components (or species) with mass fractions ζ_i. These need not be equal to the prognostic scalar variables s_k, and the relationship between them $\zeta_i = \zeta_i^e(s_k)$ needs to be given.

In the generic case, we choose

$$\zeta_i = s_i \ , \quad i = 1, \ldots, n_{\mathrm{c}} - 1 \ , \tag{1.17}$$

and the last component has a mass fraction

$$\zeta_{n_{\mathrm{c}}} = 1 - \sum_{1}^{n_{\mathrm{c}}-1} \zeta_i \ . \tag{1.18}$$

The gas constant can then be written as

$$R = \sum_{1}^{n_{\mathrm{c}}} \zeta_i R_i = \sum_{1}^{n_{\mathrm{c}}-1} (R_i - R_N) \ , \tag{1.19}$$

and similarly for other thermodynamic variables.

Particular cases different from this one occur for instance when we consider chemical equilibrium or phase equilibrium, or when different combinations of mass fractions might be preferable to better represent conservation properties.

Air-water mixtures in anelastic formulations

We consider the total water specific humidity as prognostic variable. If necessary, we also consider the liquid water specific humidity, which can be diagnostic in the case of phase equilibrium, or prognostic otherwise.

Chapter 2

Post-Processing Tools

2.1 Averages

See file `dns/tools/postprocessing/averages`.

Allows for conditional analysis.

2.2 Summary of budget equations for second-order moments

Let f and g be fluid properties per unit mass. We use the Reynolds decomposition

$$f = \overline{f} + f' \;, \tag{2.1}$$

and the Favre decomposition

$$f = \widetilde{f} + f'' \;, \tag{2.2}$$

where

$$\widetilde{f} = \overline{\rho f}/\overline{\rho} \;. \tag{2.3}$$

Favre decomposition proves convenient in variable-density flows. The overbar indicates ensemble average, which is approximated by spatial average, or temporal average, or both, depending on the configuration. We will refer to it as Reynolds average, and use the term Favre average to refer to the density-weighted averages (property per unit volume). Note that $\widetilde{f''} = 0$ and $\overline{f'} = 0$, but

$$\overline{f''} = \overline{f} - \widetilde{f} \;, \tag{2.4}$$

which only zero in constant-density flows, when $\overline{f}$ and $\widetilde{f}$ coincide. The covariances satisfy

$$\overline{f'g'} = \overline{fg'} = \overline{f'g} = \overline{f'g''} = \overline{f''g'} = \overline{fg} - \overline{f}\,\overline{g} \;, \tag{2.5}$$

$$\widetilde{f''g''} = \widetilde{fg''} = \widetilde{f''g} = \widetilde{f'g''} = \widetilde{f''g'} = \widetilde{fg} - \widetilde{f}\widetilde{g} \;, \tag{2.6}$$

$$\overline{fg} - \overline{f}\widetilde{g} = \overline{f''g'} - \overline{f}\,\overline{g''} \;. \tag{2.7}$$

We consider evolution equations of the following form:

$$\partial_t\,(\rho f) + \nabla\cdot\,(\rho f \mathbf{v}) = -\nabla\cdot\mathbf{F} + \rho S_f \;, \tag{2.8a}$$

$$\partial_t\,(\rho g) + \nabla\cdot\,(\rho g \mathbf{v}) = -\nabla\cdot\mathbf{G} + \rho S_g \;. \tag{2.8b}$$

Multiplying the first equation by g and the second by f and adding them, one finds the evolution equation for the product $\rho f g$:

$$\begin{aligned}\partial_t(\rho f g)+\nabla\cdot(\rho f g\mathbf{v}) =& -g\nabla\cdot\mathbf{F}-f\nabla\cdot\mathbf{G}\\ &+\rho g S_f+\rho f S_g\ ,\end{aligned} \tag{2.9}$$

having used the equation of conservation of mass

$$\partial_t\rho+\nabla\cdot(\rho\mathbf{v})=0 \tag{2.10}$$

in the left hand side.

The evolution equation for the mean properties can be written as

$$\partial_t\left(\overline{\rho}\widetilde{f}\right)+\nabla\cdot\left(\overline{\rho}\widetilde{f}\widetilde{\mathbf{v}}\right)=-\nabla\cdot\left(\overline{\mathbf{F}}+\overline{\rho}\widetilde{f''\mathbf{v}''}\right)+\overline{\rho}\widetilde{S_f}\ , \tag{2.11a}$$

$$\partial_t\left(\overline{\rho}\widetilde{g}\right)+\nabla\cdot\left(\overline{\rho}\widetilde{g}\widetilde{\mathbf{v}}\right)=-\nabla\cdot\left(\overline{\mathbf{G}}+\overline{\rho}\widetilde{g''\mathbf{v}''}\right)+\overline{\rho}\widetilde{S_g}\ . \tag{2.11b}$$

Multiplying the first equation by $\widetilde{g}$ and the second by $\widetilde{f}$ and adding them, one finds the evolution equation for the product $\overline{\rho}\widetilde{f}\widetilde{g}$:

$$\begin{aligned}\partial_t\left(\overline{\rho}\widetilde{f}\widetilde{g}\right)+\nabla\cdot\left(\overline{\rho}\widetilde{f}\widetilde{g}\widetilde{\mathbf{v}}\right) =& -\widetilde{g}\nabla\cdot\overline{\mathbf{F}}-\widetilde{f}\nabla\cdot\overline{\mathbf{G}}\\ &-\nabla\cdot\left(\overline{\rho}\widetilde{f''\mathbf{v}''}\widetilde{g}+\overline{\rho}\widetilde{g''\mathbf{v}''}\widetilde{f}\right)\\ &+\overline{\rho}\widetilde{\mathbf{v}''f''}\cdot\nabla\widetilde{g}+\overline{\rho}\widetilde{\mathbf{v}''g''}\cdot\nabla\widetilde{f}\\ &+\widetilde{g}\overline{S_f}+\widetilde{f}\overline{S_g}\ ,\end{aligned} \tag{2.12}$$

having used the equation of conservation of mass

$$\partial_t\overline{\rho}+\nabla\cdot\left(\overline{\rho}\widetilde{\mathbf{v}}\right)=0 \tag{2.13}$$

in the left hand side.

Averaging Eq. (2.9) and substracting Eq. (2.12) from it, one finds

$$\begin{aligned}\partial_t\left(\overline{\rho}\widetilde{f''g''}\right)+\nabla\cdot\left(\overline{\rho}\widetilde{f''g''\widetilde{\mathbf{v}}}\right) =& -\nabla\cdot\left(\overline{\rho}\widetilde{\mathbf{v}''f''g''}+\overline{\mathbf{F}'g''}+\overline{\mathbf{G}'f''}\right)\\ &+\overline{\mathbf{F}'\cdot\nabla g''}+\overline{\mathbf{G}'\cdot\nabla f''}\\ &-\overline{\rho}\widetilde{\mathbf{v}''f''}\cdot\nabla\widetilde{g}-\overline{\rho}\widetilde{\mathbf{v}''g''}\cdot\nabla\widetilde{f}\\ &+\overline{\rho}\widetilde{S_f''g''}+\overline{\rho}\widetilde{S_g''f''}\\ &-\overline{g''}\nabla\cdot\overline{\mathbf{F}}-\overline{f''}\nabla\cdot\overline{\mathbf{G}}\ ,\end{aligned} \tag{2.14}$$

having used the relationship

$$\overline{\rho f g\mathbf{v}}=\overline{\rho}\widetilde{f}\widetilde{g}\widetilde{\mathbf{v}}+\overline{\rho}\widetilde{\mathbf{v}''f''g''}+\widetilde{f''g''}\widetilde{\mathbf{v}}+\widetilde{f''\mathbf{v}''}\widetilde{g}+\widetilde{g''\mathbf{v}''}\widetilde{f}\ , \tag{2.15}$$

together with

$$\begin{aligned}\overline{g\nabla\cdot\mathbf{F}}-\widetilde{g}\nabla\cdot\overline{\mathbf{F}} &= \overline{g''\nabla\cdot\mathbf{F}'}+\overline{g''}\nabla\cdot\overline{\mathbf{F}}\\ &=\nabla\cdot\left(\overline{\mathbf{F}'g''}\right)-\overline{\mathbf{F}'\cdot\nabla g''}+\overline{g''}\nabla\cdot\overline{\mathbf{F}}\end{aligned} \tag{2.16}$$

and similarly for the other second-order term $f\nabla\cdot\mathbf{G}$.

2.2.1 Reynolds Stresses

Consider the momentum equations:

$$f = u_i \ , \qquad F_k = p\,\delta_{ik} - \tau_{ik} \ , \qquad S_f = b\,g_i - \epsilon_{imk} f_m u_k \ , \tag{2.17}$$

$$g = u_j \ , \qquad G_k = p\,\delta_{jk} - \tau_{jk} \ , \qquad S_g = b\,g_j - \epsilon_{jmk} f_m u_k \ . \tag{2.18}$$

Substituting into Eq. (2.14), we find

$$\partial_t R_{ij} = C_{ij} + P_{ij} - E_{ij} + \overline{\rho}^{-1}(\Pi_{ij} - \partial_k T_{ijk} + M_{ij}) + B_{ij} - F_{ij} \tag{2.19}$$

where

$$\begin{aligned}
R_{ij} &= \widetilde{u_i'' u_j''} && \text{(Reynolds-stress component)}\\
C_{ij} &= -\widetilde{u}_k\,\partial_k R_{ij} && \text{(mean advection)}\\
P_{ij} &= -R_{ik}\partial_k \widetilde{u}_j - R_{jk}\partial_k \widetilde{u}_i && \text{(mean-gradient (shear) production)}\\
E_{ij} &= \overline{\rho}^{-1}(\overline{\tau_{jk}'\partial_k u_i''} + \overline{\tau_{ik}'\partial_k u_j''}) && \text{(viscous dissipation)}\\
T_{ijk} &= \overline{\rho u_i'' u_j'' u_k''} + \overline{p' u_i'}\delta_{jk} + \overline{p' u_j'}\delta_{ik} - (\overline{\tau_{jk}' u_i''} + \overline{\tau_{ik}' u_j''}) && \text{(turbulent transport)}\\
\Pi_{ij} &= \overline{p'(\partial_j u_i'' + \partial_i u_j'')} && \text{(pressure strain)}\\
M_{ij} &= \overline{u_i''}(\partial_k \overline{\tau}_{jk} - \partial_j \overline{p}) + \overline{u_j''}(\partial_k \overline{\tau}_{ik} - \partial_i \overline{p}) && \text{(mean flux)}\\
B_{ij} &= \widetilde{b'' u_j''}\,g_i + \widetilde{b'' u_i''}\,g_j && \text{(buoyancy production-destruction)}\\
F_{ij} &= \epsilon_{imk} f_m R_{jk} + \epsilon_{jmk} f_m R_{ik} && \text{(Coriolis redistribution)}
\end{aligned}$$

Depending on symmetries, several terms can be zero (within statistical convergence). Note that if $b \equiv 1$, then $B_{ij} = 0$. The mean flux term is sometimes written as $M_{ij} = D_{ij} - G_{ij}$, the first term grouping the mean viscous stress contributions and the last term the mean pressure contributions. In cases of constant density, then $\overline{u_j''} = 0$ and $M_{ij} = D_{ij} = G_{ij} = 0$.

Contracting indices, the budget equation for the turbulent kinetic energy $K = R_{ii}/2$ reads

$$\partial_t K = C + P + B - E + \overline{\rho}^{-1}\left(\Pi - \partial_k T_k + M\right) \ . \tag{2.20}$$

Note that $F_{ii} = 0$. If the flow is solenoidal, then $\Pi = 0$.

2.2.2 Scalar Fluxes

Consider the momentum and scalar equations:

$$f = u_i \ , \qquad F_k = p\,\delta_{ik} - \tau_{ik} \ , \qquad S_f = b\,g_i - \epsilon_{imk} f_m u_k \ , \tag{2.21}$$

$$g = \zeta \ , \qquad G_k = -q_k \ , \qquad S_g = w \ . \tag{2.22}$$

Substituting into Eq. (2.14), we find

$$\partial_t R_{i\zeta} = C_{i\zeta} + P_{i\zeta} - E_{i\zeta} + \overline{\rho}^{-1}(\Pi_{i\zeta} - \partial_k T_{i\zeta k} + M_{i\zeta}) + B_{i\zeta} - F_{i\zeta} + Q_{i\zeta} \tag{2.23}$$

where

$$
\begin{aligned}
R_{i\zeta} &= \widetilde{u_i''\zeta''} && \text{(scalar-flux component)} \\
C_{i\zeta} &= -\widetilde{u}_k\,\partial_k R_{i\zeta} && \text{(mean advection)} \\
P_{i\zeta} &= -R_{ik}\,\partial_k\widetilde{\zeta} - R_{\zeta k}\,\partial_k\widetilde{u}_i && \text{(mean-gradient and tilting production)} \\
E_{i\zeta} &= \overline{\rho}^{-1}(\overline{q_k'\partial_k u_i''} + \overline{\tau_{ik}'\partial_k\zeta''}) && \text{(molecular destruction)} \\
\Pi_{i\zeta} &= \overline{p'\partial_i\zeta''} && \text{(pressure-flux interaction)} \\
T_{i\zeta k} &= \overline{\rho\zeta''u_k''u_i''} + \overline{p's''}\delta_{ik} - \overline{\tau_{ik}'s''} - \overline{q_k'u_i''} && \text{(turbulent transport)} \\
M_{i\zeta} &= \overline{u_i''}\partial_k\overline{q}_k + \overline{\zeta''}(\partial_k\overline{\tau}_{ik} - \partial_i\overline{p}) && \text{(mean flux)} \\
B_{i\zeta} &= \widetilde{b''\zeta''}g_i && \text{(buoyancy iteraction)} \\
F_{i\zeta} &= \epsilon_{imk}f_m R_{k\zeta} && \text{(Coriolis interation)} \\
Q_{i\zeta} &= \widetilde{w''u_i''} && \text{(source)}
\end{aligned}
$$

The mean flux term is sometimes written as $M_{i\zeta} = D_{i\zeta} - G_{i\zeta}$, the first term grouping the mean molecular flux contributions and the last term the mean pressure contributions.

2.2.3 Scalar Variance

Consider the scalar equation twice:

$$f = \zeta\ , \qquad F_k = -q_k\ , \qquad S_f = w\ , \tag{2.24}$$

$$g = \zeta\ , \qquad G_k = -q_k\ , \qquad S_g = w\ . \tag{2.25}$$

Substituting into Eq. (2.14), we find

$$\partial_t R_{\zeta\zeta} = C_{\zeta\zeta} + P_{\zeta\zeta} - E_{\zeta\zeta} + \overline{\rho}^{-1}(-\partial_k T_{\zeta\zeta k} + M_{\zeta\zeta}) + Q_{\zeta\zeta} \tag{2.26}$$

where

$$
\begin{aligned}
R_{\zeta\zeta} &= \widetilde{\zeta''\zeta''} && \text{(scalar variance)} \\
C_{\zeta\zeta} &= -\widetilde{u}_k\,\partial_k R_{\zeta\zeta} && \text{(mean advection)} \\
P_{\zeta\zeta} &= -2R_{k\zeta}\,\partial_k\widetilde{\zeta} && \text{(mean-gradient production)} \\
E_{\zeta\zeta} &= 2\overline{\rho}^{-1}\overline{q_k'\partial_k\zeta''} && \text{(molecular destruction)} \\
T_{\zeta\zeta k} &= \overline{\rho\zeta''^2u_k''} - 2\overline{q_k'\zeta''} && \text{(turbulent transport)} \\
M_{\zeta\zeta} &= 2\overline{\zeta''}\partial_k\overline{q}_k && \text{(mean flux)} \\
Q_{\zeta\zeta} &= 2\widetilde{w''\zeta''} && \text{(source)}
\end{aligned}
$$

Part II

Technical Guide

Chapter 3

Memory management

See file `base/tlab_memory`.

Main arrays are allocated during initialization. Intermediate variables are to be stored in `tmp` and not in scratch arrays. Scratch arrays can always be used in low level procedures (e.g., `finitedifferences`), and it is better not to use them in high level procedures (e.g., `operators`).

array	size	content
`x`	number of points in Ox	x-coordinate information
`y`	number of points in Oy	y-coordinate information
`z`	number of points in Oz	z-coordinate information
`g`	number of points in each direction × number of arrays for numerics	finite differences information
`q`	number of points × number of flow fields	flow variables
`s`	number of points × number of scalar fields	scalar variables
`txc`	extended number of points × number of temporary fields	temporary variables
`wrk3d`	extended number of points	scratch
`wrk2d`	maximum number of points in 2D planes × number of scratch planes	scratch
`wrk1d`	maximum number of points in 1D lines × number of scratch lines	scratch

Table 3.1: Main arrays and their sizes in terms of the number of points. The first 4 are derived types.

Pointers are also defined during initialization to access these memory spaces with arrays of different shape and even different type, more specifically, complex type needed in Fourier decomposition. See corresponding procedures in `base/tlab_memory`.

Chapter 4

Profiling

The diagrams shown below have been constructed with `gprof2dot.py`[1].

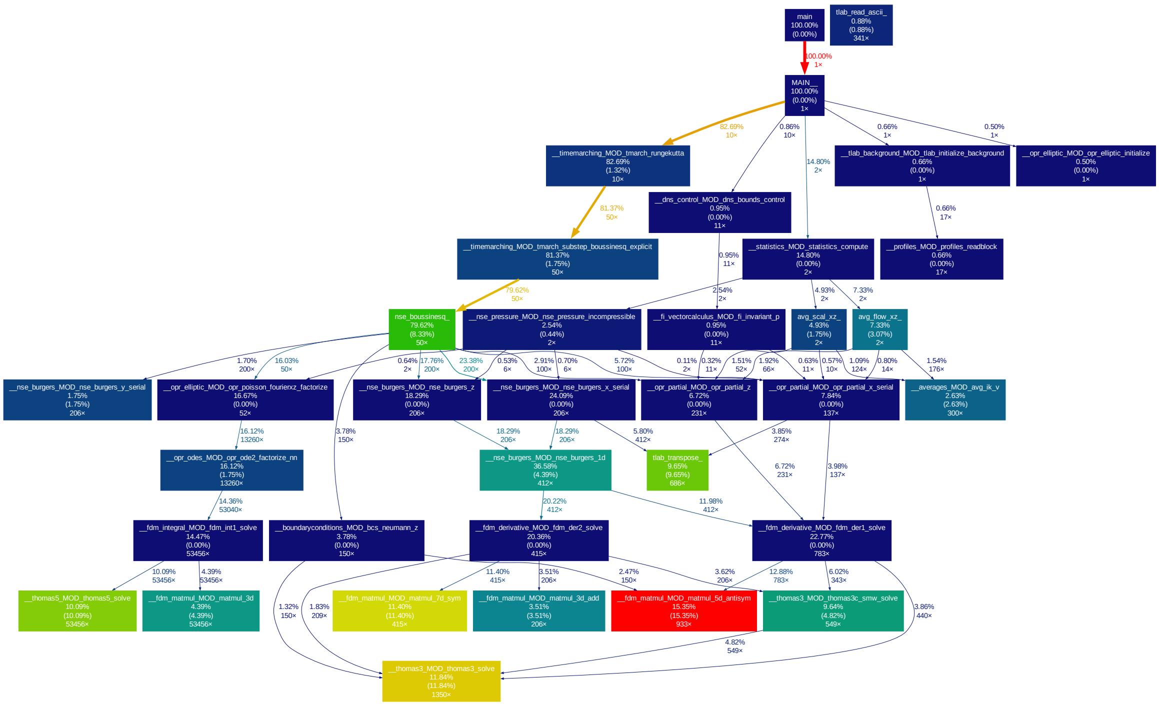

Figure 4.1: Profiling diagram of `examples/Case08` running 10 iterations in serial mode. Profiling data obtained from `gfortran -pg` and processed with `gprof`, running the command `gprof path/to/your/executable | gprof2dot --color-nodes-by-selftime | dot -Tpdf -o output.pdf`.

[1] `https://github.com/jrfonseca/gprof2dot`

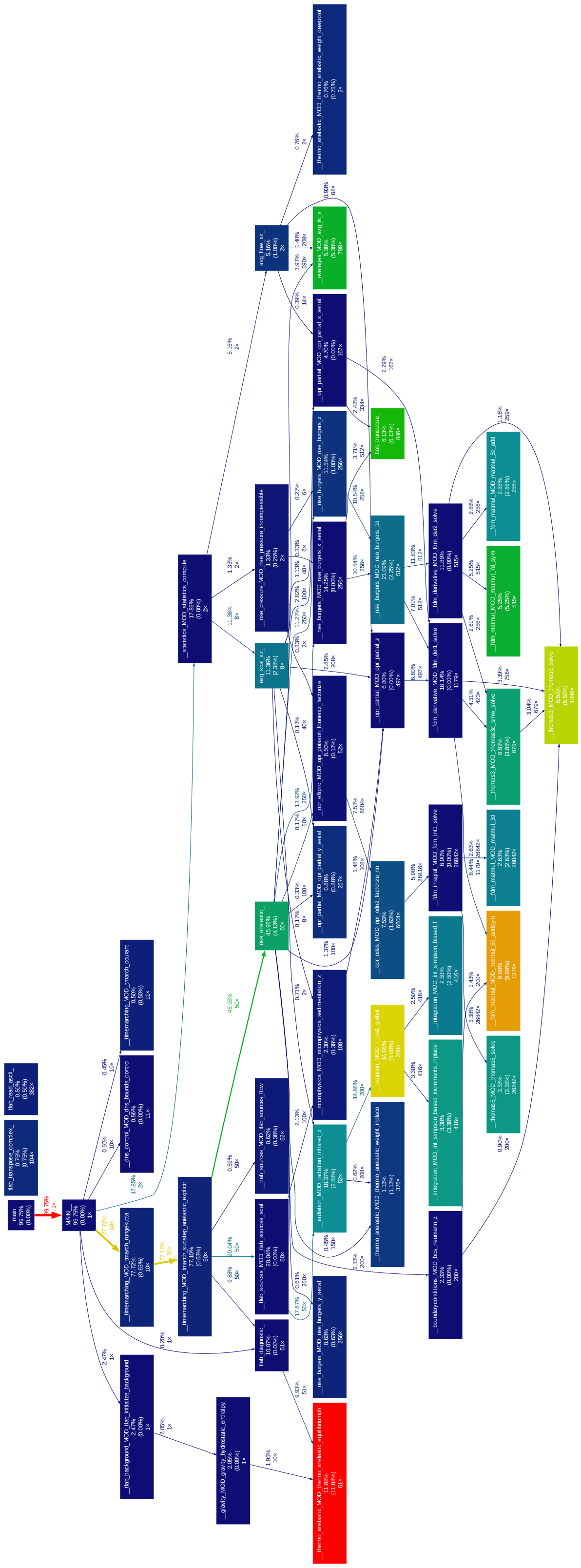

Figure 4.2: Profiling diagram of `examples/Case65` running 10 iterations in serial mode. Profiling data obtained from `gfortran -pg` and processed with `gprof`, running the command `gprof path/to/your/executable | gprof2dot --color-nodes-by-selftime | dot -Tpdf -o output.pdf`.

Bibliography

Juan Pedro Mellado and Cedrick Ansorge. Factorization of the fourier transform of the pressure-poisson equation using finite differences in colocated grids. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 92(5):380–392, 2012.