Manufactured Solution for the 2D Compressible Transient Navier–Stokes Equations with Sutherland Viscosity Model using Maple$^{\text{TM}}$

Kemelli C. Estacio-Hiroms

August 8, 2013

Abstract

The Method of Manufactured Solutions is a valuable approach for code verification, providing means to verify how accurately the numerical method solves the partial differential equations of interest. This document presents the source terms generated by the application of the Method of Manufactured Solutions (MMS) on the 2D transient Navier–Stokes equations with Sutherland Viscosity Model using the analytical manufactured solutions for density, velocity and pressure presented by Roy et al. (2002).

1   Mathematical Model

The conservation of mass, momentum, and total energy for a compressible transient viscous fluid may be written as:

$$\frac{\partial(\rho)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{1}$$

$$\frac{\partial(\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\nabla p + \nabla \cdot (\boldsymbol{\tau}), \tag{2}$$

$$\frac{\partial(\rho e_t)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} H) = -\nabla \cdot (p\mathbf{u}) - \nabla \cdot \mathbf{q} + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{u}). \tag{3}$$

Equations (1)–(3) are known as Navier–Stokes equations and, ρ is the density, $\boldsymbol{u} = (u, v)$ is the velocity in x or y-direction, respectively, and p is the pressure. The total enthalpy, H, may be expressed in terms of the total energy per unit mass e_t, density, and pressure:

$$H = e_t + \frac{p}{\rho}.$$

For a calorically perfect gas, the Navier–Stokes equations are closed with two auxiliary relations for energy:

$$e_t = e + \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2}, \quad \text{and} \quad e = \frac{1}{\gamma - 1}RT, \tag{4}$$

and with the ideal gas equation of state:

$$p = \rho RT. \tag{5}$$

The shear stress tensor is:

$$\tau_{xx} = \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right), \quad \tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \quad \tau_{yy} = \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right), \tag{6}$$

where μ is the absolute viscosity. The heat flux vector $\mathbf{q} = (q_x, q_y)$ is given by:

$$q_x = -\kappa\frac{\partial T}{\partial x}, \quad q_y = -\kappa\frac{\partial T}{\partial y} \tag{7}$$

where κ is the thermal conductivity, which can be determined by choosing the Prandtl number:

$$\kappa = \frac{\gamma R\mu}{(\gamma - 1)\text{Pr}}. \tag{8}$$

1.1 Sutherland viscosity model

Sutherland (1893) published a relationship between the absolute temperature of an ideal gas, T, and its dynamic (absolute) viscosity, μ. The model is based on the kinetic theory of ideal gases and an idealized intermolecular-force potential. The general equation is given as:

$$\mu = \frac{A_\mu T^{\frac{3}{2}}}{T + B_\mu} \tag{9}$$

with

$$A_\mu = \frac{\mu_{\text{ref}}}{T_{\text{ref}}^{\frac{3}{2}}}(T_{\text{ref}} + B_\mu), \tag{10}$$

where B_μ is the Sutherland temperature, T_{ref} is a reference temperature, and μ_{ref} is the viscosity at the reference temperature T_{ref}.

2 Manufactured Solution

The Method of Manufactured Solutions (MMS) applied to Navier–Stokes equations consists in modifying Equations (1) – (3) by adding a source term to the right-hand side of each equation, so the modified set of equations conveniently has the analytical solution chosen *a priori*.

Roy et al. (2002) introduce the general form of the primitive manufactured solution variables to be a function of sines and cosines in x, y and z only. In this work, Roy et al. (2002)'s manufactured solutions are modified in order to address temporal accuracy as well:

$$\phi(x,y,t) = \phi_0 + \phi_x\, f_s\left(\frac{a_{\phi x}\pi x}{L}\right) + \phi_y\, f_s\left(\frac{a_{\phi y}\pi y}{L}\right) + \phi_t\, f_s\left(\frac{a_{\phi t}\pi t}{L}\right), \tag{11}$$

where $\phi = \rho, u, v$ or p, and $f_s(\cdot)$ functions denote either sine or cosine function. Note that in this case, ϕ_x, ϕ_y, and ϕ_t are constants and the subscripts do not denote differentiation.

Although Roy et al. (2002) provide the constants used in the manufactured solutions for the 2D supersonic and subsonic cases for Euler and Navier-Stokes equations, only the source term for the 2D mass conservation equation (1) is presented.

Source terms for mass conservation (Q_ρ), momentum (Q_u, Q_v) and total energy (Q_{e_t}) equations are obtained by symbolic manipulations of compressible transient Navier–Stokes equations above using Maple 15 (Maplesoft, 2011) and are presented in the following sections for the one, two and three-dimensional cases.

The manufactured analytical solution for each one of the variables in the 2D transient Navier–Stokes equations are:

$$
\begin{aligned}
\rho\left(x,y,t\right) &= \rho_0 + \rho_x \sin\left(\frac{a_{\rho x}\pi x}{L}\right) + \rho_y \cos\left(\frac{a_{\rho y}\pi y}{L}\right) + \rho_t \sin\left(\frac{a_{\rho t}\pi t}{L_t}\right), \\
u\left(x,y,t\right) &= u_0 + u_x \sin\left(\frac{a_{ux}\pi x}{L}\right) + u_y \cos\left(\frac{a_{uy}\pi y}{L}\right) + u_t \cos\left(\frac{a_{ut}\pi t}{L_t}\right), \\
v\left(x,y,t\right) &= v_0 + v_x \cos\left(\frac{a_{vx}\pi x}{L}\right) + v_y \sin\left(\frac{a_{vy}\pi y}{L}\right) + v_t \sin\left(\frac{a_{vt}\pi t}{L_t}\right), \\
p\left(x,y,t\right) &= p_0 + p_x \cos\left(\frac{a_{px}\pi x}{L}\right) + p_y \sin\left(\frac{a_{py}\pi y}{L}\right) + p_t \cos\left(\frac{a_{pt}\pi t}{L_t}\right).
\end{aligned} \tag{12}
$$

The MMS applied to 2D Navier–Stokes equations with Sutherland viscosity model consists in modifying Equations (1)–(3) by adding a source term to the right-hand side of each equation:

$$
\begin{aligned}
&\frac{\partial(\rho)}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = Q_\rho \\
&\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} + \frac{\partial(p)}{\partial x} - \frac{\partial(\tau_{xx})}{\partial x} - \frac{\partial(\tau_{xy})}{\partial y} = Q_u, \\
&\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y} + \frac{\partial(p)}{\partial y} - \frac{\partial(\tau_{yx})}{\partial x} - \frac{\partial(\tau_{yy})}{\partial y} = Q_v, \\
&\frac{\partial(\rho e_t)}{\partial t} + \frac{\partial(\rho u e_t)}{\partial x} + \frac{\partial(\rho v e_t)}{\partial y} + \frac{\partial(pu)}{\partial x} + \frac{\partial(pv)}{\partial y} + \frac{\partial(q_x)}{\partial x} + \frac{\partial(q_y)}{\partial y} - \frac{\partial(u\tau_{xx} + v\tau_{xy})}{\partial x} - \frac{\partial(u\tau_{yx} + v\tau_{yy})}{\partial y} = Q_{e_t},
\end{aligned} \tag{13}
$$

2

so this modified set of equations has for analytical solution Equation (12).

The source terms Q_ρ, Q_u, Q_v and Q_{e_t} are presented with the use of the following auxiliary variables:

$$
\begin{aligned}
\mathtt{Rho} &= \rho_0 + \rho_x \sin\left(\frac{a_{\rho x}\pi x}{L}\right) + \rho_y \cos\left(\frac{a_{\rho y}\pi y}{L}\right) + \rho_t \sin\left(\frac{a_{\rho t}\pi t}{L}\right), \\
\mathtt{U} &= u_0 + u_x \sin\left(\frac{a_{ux}\pi x}{L}\right) + u_y \cos\left(\frac{a_{uy}\pi y}{L}\right) + u_t \cos\left(\frac{a_{ut}\pi t}{L}\right), \\
\mathtt{V} &= v_0 + v_x \cos\left(\frac{a_{vx}\pi x}{L}\right) + v_y \sin\left(\frac{a_{vy}\pi y}{L}\right) + v_t \sin\left(\frac{a_{vt}\pi t}{L}\right), \\
\mathtt{P} &= p_0 + p_x \cos\left(\frac{a_{px}\pi x}{L}\right) + p_y \sin\left(\frac{a_{py}\pi y}{L}\right) + p_t \cos\left(\frac{a_{pt}\pi t}{L}\right), \\
\mathtt{T} &= \frac{\mathtt{P}}{R\,\mathtt{Rho}}, \\
\mathtt{Mu} &= \frac{A_\mu\,\mathtt{T}^{\frac{3}{2}}}{\mathtt{T} + B_\mu},
\end{aligned}
\tag{14}
$$

which simply are the manufactured solutions for ρ, u, v and p, the temperature (5), and the fluid viscosity according to Sutherland model (9), respectively. The following expressions for the derivatives of the viscosity are also used:

$$
\begin{aligned}
\frac{\partial\,\mathtt{Mu}}{\partial x} &= \frac{a_{\rho x}\pi\rho_x\,\mathtt{Mu}^2}{A_\mu L\,\mathtt{Rho}\sqrt{\mathtt{T}}}\cos\left(\frac{a_{\rho x}\pi x}{L}\right) - \frac{3}{2}\frac{a_{\rho x}\pi\rho_x\,\mathtt{Mu}}{L\,\mathtt{Rho}}\cos\left(\frac{a_{\rho x}\pi x}{L}\right) + \\
&\quad + \frac{a_{px}\pi p_x\,\mathtt{Mu}^2}{A_\mu L R\,\mathtt{Rho}\,\mathtt{T}^{3/2}}\sin\left(\frac{a_{px}\pi x}{L}\right) - \frac{3}{2}\frac{a_{px}\pi p_x\,\mathtt{Mu}}{LR\,\mathtt{Rho}\,\mathtt{T}}\sin\left(\frac{a_{px}\pi x}{L}\right) \\
\frac{\partial\,\mathtt{Mu}}{\partial y} &= -\frac{a_{\rho y}\pi\rho_y\,\mathtt{Mu}^2}{A_\mu L\,\mathtt{Rho}\sqrt{\mathtt{T}}}\sin\left(\frac{a_{\rho y}\pi y}{L}\right) + \frac{3}{2}\frac{a_{\rho y}\pi\rho_y\,\mathtt{Mu}}{L\,\mathtt{Rho}}\sin\left(\frac{a_{\rho y}\pi y}{L}\right) + \\
&\quad - \frac{a_{py}\pi p_y\,\mathtt{Mu}^2}{A_\mu L R\,\mathtt{Rho}\,\mathtt{T}^{3/2}}\cos\left(\frac{a_{py}\pi y}{L}\right) + \frac{3}{2}\frac{a_{py}\pi p_y\,\mathtt{Mu}}{LR\,\mathtt{Rho}\,\mathtt{T}}\cos\left(\frac{a_{py}\pi y}{L}\right).
\end{aligned}
\tag{15}
$$

2.1   2D Mass Conservation

The mass conservation equation may be written as an operator $\mathcal{L}_\rho$:

$$
\mathcal{L}_\rho = \mathcal{L}_{\rho\,\mathrm{time}} + \mathcal{L}_{\rho\,\mathrm{convection}}
$$

where:

$$
\begin{aligned}
\mathcal{L}_{\rho\,\mathrm{time}} &= \frac{\partial(\rho)}{\partial t} \\
\mathcal{L}_{\rho\,\mathrm{convection}} &= \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y}.
\end{aligned}
\tag{16}
$$

The operators defined in Eq. (16) are applied into Equation (12), providing respective source terms that will compound source term Q_ρ:

$$
Q_\rho = Q_{\rho\,\mathrm{time}} + Q_{\rho\,\mathrm{convection}}.
$$

They are:

$$
\begin{aligned}
Q_{\rho\,\mathrm{convection}} &= \cos\left(\frac{a_{\rho x}\pi x}{L}\right)\frac{a_{\rho x}\pi\rho_x\,\mathtt{U}}{L} - \sin\left(\frac{a_{\rho y}\pi y}{L}\right)\frac{a_{\rho y}\pi\rho_y\,\mathtt{V}}{L} + \\
&\quad + \frac{\pi\,\mathtt{Rho}}{L}\left(a_{ux}u_x\cos\left(\frac{a_{ux}\pi x}{L}\right) + a_{vy}v_y\cos\left(\frac{a_{vy}\pi y}{L}\right)\right) \\
Q_{\rho\,\mathrm{time}} &= \frac{a_{\rho t}\pi\rho_t}{L_t}\cos\left(\frac{a_{\rho t}\pi t}{L_t}\right),
\end{aligned}
$$

where $\mathtt{Rho}$, $\mathtt{U}$, $\mathtt{V}$ and $\mathtt{W}$ are defined in Equation (14).

2.2 2D Momentum

2.2.1 Velocity u

For the generation of the analytical source term Q_u, x-momentum equation (2) is written as an operator $\mathcal{L}_u$:

$$\mathcal{L}_u = \mathcal{L}_{u\,\text{time}} + \mathcal{L}_{u\,\text{convection}} + \mathcal{L}_{u\,\text{pressure}} + \mathcal{L}_{u\,\text{viscous}}$$

with each one of the sub-operators defined as follows:

$$\mathcal{L}_{u\,\text{time}} = \frac{\partial(\rho u)}{\partial t}$$

$$\mathcal{L}_{u\,\text{convection}} = \frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y}$$

$$\mathcal{L}_{u\,\text{pressure}} = \frac{\partial(p)}{\partial x}$$

$$\mathcal{L}_{u\,\text{viscous}} = -\frac{\partial(\tau_{xx})}{\partial x} - \frac{\partial(\tau_{xy})}{\partial y}$$

Source term Q_u is obtained by operating $\mathcal{L}_u$ on Equations (12) together with the use of the auxiliary relations given in Equations (14) and (15). It yields:

$$Q_u = Q_{u\,\text{time}} + Q_{u\,\text{convection}} + Q_{u\,\text{pressure}} + Q_{u\,\text{viscous}}$$

with

$$Q_{u\,\text{convection}} = \frac{a_{\rho x}\pi\rho_x\,\texttt{U}^2}{L}\cos\left(\frac{a_{\rho x}\pi x}{L}\right) - \frac{a_{\rho y}\pi\rho_y\,\texttt{U}\,\texttt{V}}{L}\sin\left(\frac{a_{\rho y}\pi y}{L}\right) - \frac{a_{uy}\pi u_y\,\texttt{Rho}\,\texttt{V}}{L}\sin\left(\frac{a_{uy}\pi y}{L}\right) +$$
$$+ \frac{\pi\,\texttt{Rho}\,\texttt{U}}{L}\left(2a_{ux}u_x\cos\left(\frac{a_{ux}\pi x}{L}\right) + a_{vy}v_y\cos\left(\frac{a_{vy}\pi y}{L}\right)\right)$$

$$Q_{u\,\text{pressure}} = -a_{px}\pi p_x\sin\left(\frac{a_{px}\pi x}{L}\right)/L$$

$$Q_{u\,\text{viscous}} = \frac{1}{3}\frac{\texttt{Mu}\pi^2}{L^2}\left(4a_{ux}^2 u_x\sin\left(\frac{a_{ux}\pi x}{L}\right) + 3a_{uy}^2 u_y\cos\left(\frac{a_{uy}\pi y}{L}\right)\right) +$$
$$- \frac{2}{3}\frac{\partial\,\texttt{Mu}}{\partial x}\frac{\pi}{L}\left(2a_{ux}u_x\cos\left(\frac{a_{ux}\pi x}{L}\right) - a_{vy}v_y\cos\left(\frac{a_{vy}\pi y}{L}\right)\right) +$$
$$+ \frac{\partial\,\texttt{Mu}}{\partial y}\frac{\pi}{L}\left(a_{uy}u_y\sin\left(\frac{a_{uy}\pi y}{L}\right) + a_{vx}v_x\sin\left(\frac{a_{vx}\pi x}{L}\right)\right)$$

$$Q_{u\,\text{time}} = -\frac{a_{ut}u_t\pi\,\texttt{Rho}}{L_t}\sin\left(\frac{a_{ut}t\pi}{L_t}\right) + \frac{a_{\rho t}\rho_t\pi\,\texttt{U}}{L_t}\cos\left(\frac{a_{\rho t}t\pi}{L_t}\right),$$

with $\texttt{Mu}$, $\texttt{Rho}$, $\texttt{U}$, and $\texttt{V}$ defined in Equation (14) and the derivatives $\dfrac{\partial\,\texttt{Mu}}{\partial x}$ and $\dfrac{\partial\,\texttt{Mu}}{\partial y}$ defined in Equation (15).

2.2.2 Velocity v

Analogously to the velocity-u case, for the generation of the analytical source term Q_v, y-momentum equation (2) is written as an operator $\mathcal{L}_v$:

$$\mathcal{L}_v = \mathcal{L}_{v\,\text{time}} + \mathcal{L}_{v\,\text{convection}} + \mathcal{L}_{v\,\text{pressure}} + \mathcal{L}_{v\,\text{viscous}}$$

with each one of the sub-operators defined as follows:

$$\mathcal{L}_{v\,\text{time}} = \frac{\partial(\rho v)}{\partial t}$$

$$\mathcal{L}_{v\,\text{convection}} = \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y}$$

$$\mathcal{L}_{v\,\text{pressure}} = \frac{\partial(p)}{\partial y}$$

$$\mathcal{L}_{v\,\text{viscous}} = -\frac{\partial(\tau_{yx})}{\partial x} - \frac{\partial(\tau_{yy})}{\partial y}$$

Source term Q_v is obtained by operating $\mathcal{L}_v$ on Equations (12) together with the use of the auxiliary relations given in Equations (14) and (15). It yields:

$$Q_v = Q_{v\,\text{time}} + Q_{v\,\text{convection}} + Q_{v\,\text{pressure}} + Q_{v\,\text{viscous}}$$

with

$$Q_{v\,\text{convection}} = \frac{a_{\rho x}\pi\rho_x\,\mathtt{U}\,\mathtt{V}}{L}\cos\left(\frac{a_{\rho x}\pi x}{L}\right) - \frac{a_{\rho y}\pi\rho_y\,\mathtt{V}^2}{L}\sin\left(\frac{a_{\rho y}\pi y}{L}\right) - \frac{a_{vx}\pi v_x\,\mathtt{Rho}\,\mathtt{U}}{L}\sin\left(\frac{a_{vx}\pi x}{L}\right) +$$

$$+\frac{\pi\,\mathtt{Rho}\,\mathtt{V}}{L}\left(a_{ux}u_x\cos\left(\frac{a_{ux}\pi x}{L}\right) + 2a_{vy}v_y\cos\left(\frac{a_{vy}\pi y}{L}\right)\right)$$

$$Q_{v\,\text{pressure}} = \frac{a_{py}\pi p_y}{L}\cos\left(\frac{a_{py}\pi y}{L}\right)$$

$$Q_{v\,\text{viscous}} = \frac{1}{3}\frac{\mathtt{Mu}\pi^2}{L^2}\left(3a_{vx}^2 v_x\cos\left(\frac{a_{vx}\pi x}{L}\right) + 4a_{vy}^2 v_y\sin\left(\frac{a_{vy}\pi y}{L}\right)\right) +$$

$$+\frac{\partial\,\mathtt{Mu}}{\partial x}\frac{\pi}{L}\left(a_{uy}u_y\sin\left(\frac{a_{uy}\pi y}{L}\right) + a_{vx}v_x\sin\left(\frac{a_{vx}\pi x}{L}\right)\right) +$$

$$-\frac{2}{3}\frac{\partial\,\mathtt{Mu}}{\partial y}\frac{\pi}{L}\left(-a_{ux}u_x\cos\left(\frac{a_{ux}\pi x}{L}\right) + 2a_{vy}v_y\cos\left(\frac{a_{vy}\pi y}{L}\right)\right)$$

$$Q_{v\,\text{time}} = \frac{a_{vt}v_t\pi\,\mathtt{Rho}}{L_t}\cos\left(\frac{a_{vt}\pi t}{L_t}\right) + \frac{a_{\rho t}\rho_t\pi\,\mathtt{V}}{L_t}\cos\left(\frac{a_{\rho t}\pi t}{L_t}\right),$$

where $\mathtt{Mu}$, $\mathtt{Rho}$, $\mathtt{U}$ and $\mathtt{V}$ are defined in Equation (14) and the derivatives $\dfrac{\partial\,\mathtt{Mu}}{\partial x}$ and $\dfrac{\partial\,\mathtt{Mu}}{\partial y}$ are given in Equation (15).

2.3 Total Energy Conservation Equation

The total energy equation is written as an operator:

$$\mathcal{L}_{e_t} = \mathcal{L}_{e_t\text{time}} + \mathcal{L}_{e_t\,\text{convection}} + \mathcal{L}_{e_t\,\text{pressure work}} + \mathcal{L}_{e_t\,\text{viscous work}} + \mathcal{L}_{e_t\,\text{conduction}}$$

with

$$\mathcal{L}_{e_t\text{time}} = \frac{\partial(\rho e_t)}{\partial t},$$

$$\mathcal{L}_{e_t\,\text{convection}} = \frac{\partial(\rho u e_t)}{\partial x} + \frac{\partial(\rho v e_t)}{\partial y},$$

$$\mathcal{L}_{e_t\,\text{pressure work}} = +\frac{\partial(pu)}{\partial x} + \frac{\partial(pv)}{\partial y},$$

$$\mathcal{L}_{e_t\,\text{conduction}} = +\frac{\partial(q_x)}{\partial x} + \frac{\partial(q_y)}{\partial y},$$

$$\mathcal{L}_{e_t\,\text{viscous work}} = -\frac{\partial(u\tau_{xx} + v\tau_{xy})}{\partial x} - \frac{\partial(u\tau_{yx} + v\tau_{yy})}{\partial y},$$

Therefore, source term Q_{e_t} is given by

$$Q_{e_t} = Q_{e_t\text{time}} + Q_{e_t\,\text{convection}} + Q_{e_t\,\text{pressure work}} + Q_{e_t\,\text{viscous work}} + Q_{e_t\,\text{conduction}},$$

where:

$$Q_{e\,\text{convection}} = \frac{1}{2}\frac{a_{\rho x}\pi\rho_x\,\texttt{U}^3}{L}\cos\left(\frac{a_{\rho x}\pi x}{L}\right) + \frac{1}{2}\frac{a_{\rho x}\pi\rho_x\,\texttt{U}\,\texttt{V}^2}{L}\cos\left(\frac{a_{\rho x}\pi x}{L}\right) - \frac{\frac{1}{2}a_{\rho y}\pi\rho_y\,\texttt{U}^2\,\texttt{V}}{L}\sin\left(\frac{a_{\rho y}\pi y}{L}\right) +$$

$$-\frac{1}{2}\frac{a_{\rho y}\pi\rho_y\,\texttt{V}^3}{L}\sin\left(\frac{a_{\rho y}\pi y}{L}\right) - \frac{a_{px}\pi p_x\,\texttt{U}}{(\gamma-1)L}\sin\left(\frac{a_{px}\pi x}{L}\right) + \frac{a_{py}\pi p_y\,\texttt{V}}{(\gamma-1)L}\cos\left(\frac{a_{py}\pi y}{L}\right) +$$

$$+\frac{1}{2}\frac{\pi\,\texttt{Rho}\,\texttt{U}^2}{L}\left(3a_{ux}u_x\cos\left(\frac{a_{ux}\pi x}{L}\right) + a_{vy}v_y\cos\left(\frac{a_{vy}\pi y}{L}\right)\right) +$$

$$-\frac{\pi\,\texttt{Rho}\,\texttt{U}\,\texttt{V}}{L}\left(a_{uy}u_y\sin\left(\frac{a_{uy}\pi y}{L}\right) + a_{vx}v_x\sin\left(\frac{a_{vx}\pi x}{L}\right)\right) +$$

$$+\frac{1}{2}\frac{\pi\,\texttt{Rho}\,\texttt{V}^2}{L}\left(a_{ux}u_x\cos\left(\frac{a_{ux}\pi x}{L}\right) + 3a_{vy}v_y\cos\left(\frac{a_{vy}\pi y}{L}\right)\right) +$$

$$+\frac{\pi\,\texttt{P}}{(\gamma-1)L}\left(a_{ux}u_x\cos\left(\frac{a_{ux}\pi x}{L}\right) + a_{vy}v_y\cos\left(\frac{a_{vy}\pi y}{L}\right)\right)$$

$$Q_{e\,\text{pressure work}} = -\frac{a_{px}\pi p_x\,\texttt{U}}{L}\sin\left(\frac{a_{px}\pi x}{L}\right) + \frac{a_{py}\pi p_y\,\texttt{V}}{L}\cos\left(\frac{a_{py}\pi y}{L}\right) +$$

$$+\frac{\pi\,\texttt{P}}{L}\left(a_{ux}u_x\cos\left(\frac{a_{ux}\pi x}{L}\right) + a_{vy}v_y\cos\left(\frac{a_{vy}\pi y}{L}\right)\right)$$

$$Q_{e\,\text{conduction}} = \texttt{k}\frac{\partial\,\texttt{Mu}}{\partial x}\frac{a_{\rho x}\pi\rho_x\,\texttt{P}}{LR\,\texttt{Mu}\,\texttt{Rho}^2}\cos\left(\frac{a_{\rho x}\pi x}{L}\right) + \texttt{k}\frac{\partial\,\texttt{Mu}}{\partial x}\frac{a_{px}\pi p_x}{LR\,\texttt{Mu}\,\texttt{Rho}}\sin\left(\frac{a_{px}\pi x}{L}\right) +$$

$$-\texttt{k}\frac{\partial\,\texttt{Mu}}{\partial y}\frac{a_{\rho y}\pi\rho_y\,\texttt{P}}{LR\,\texttt{Mu}\,\texttt{Rho}^2}\sin\left(\frac{a_{\rho y}\pi y}{L}\right) - \texttt{k}\frac{\partial\,\texttt{Mu}}{\partial y}\frac{a_{py}\pi p_y}{LR\,\texttt{Mu}\,\texttt{Rho}}\cos\left(\frac{a_{py}\pi y}{L}\right) +$$

$$+\frac{\texttt{k}\pi^2}{L^2 R\,\texttt{Rho}}\left(a_{px}^2 p_x\cos\left(\frac{a_{px}\pi x}{L}\right) + a_{py}^2 p_y\sin\left(\frac{a_{py}\pi y}{L}\right)\right) +$$

$$-\frac{\texttt{k}\pi^2\,\texttt{P}}{L^2 R\,\texttt{Rho}^2}\left(a_{\rho x}^2\rho_x\sin\left(\frac{a_{\rho x}\pi x}{L}\right) + a_{\rho y}^2\rho_y\cos\left(\frac{a_{\rho y}\pi y}{L}\right)\right) +$$

$$-\frac{2\,\texttt{k}\pi^2}{L^2 R\,\texttt{Rho}^2}\left(a_{\rho x}a_{px}\rho_x p_x\cos\left(\frac{a_{\rho x}\pi x}{L}\right)\sin\left(\frac{a_{px}\pi x}{L}\right) + a_{\rho y}a_{py}\rho_y p_y\sin\left(\frac{a_{\rho y}\pi y}{L}\right)\cos\left(\frac{a_{py}\pi y}{L}\right)\right) +$$

$$-\frac{2\,\texttt{k}\pi^2\,\texttt{P}}{L^2 R\,\texttt{Rho}^3}\left(a_{\rho x}^2\rho_x^2\cos\left(\frac{a_{\rho x}\pi x}{L}\right)^2 + a_{\rho y}^2\rho_y^2\sin\left(\frac{a_{\rho y}\pi y}{L}\right)^2\right)$$

$$Q_{e\,\text{time}} = -\frac{a_{ut}u_t\pi\,\texttt{Rho}\,\texttt{U}}{L_t}\sin\left(\frac{a_{ut}\pi t}{L_t}\right) + \frac{a_{vt}v_t\pi\,\texttt{Rho}\,\texttt{V}}{L_t}\cos\left(\frac{a_{vt}\pi t}{L_t}\right) + \frac{1}{2}\frac{a_{\rho t}\rho_t\pi\,\texttt{U}^2}{L_t}\cos\left(\frac{a_{\rho t}\pi t}{L_t}\right) +$$

$$+\frac{1}{2}\frac{a_{\rho t}\rho_t\pi\,\texttt{V}^2}{L_t}\cos\left(\frac{a_{\rho t}\pi t}{L_t}\right) - \frac{a_{pt}p_t\pi}{(\gamma-1)L_t}\sin\left(\frac{a_{pt}\pi t}{L_t}\right),$$

Again, $\texttt{Mu}$, $\texttt{Rho}$, $\texttt{U}$, $\texttt{V}$, and $\texttt{P}$ are defined in Equation (14) and the derivatives $\frac{\partial\,\texttt{Mu}}{\partial x}$ and $\frac{\partial\,\texttt{Mu}}{\partial y}$ are given in Equation (15). Accordingly,

$$\texttt{k} = \frac{\gamma R\,\texttt{Mu}}{(\gamma-1)\text{Pr}}.$$

3   Boundary Conditions

Additionally to verifying code capability of solving the governing equations accurately in the interior of the domain of interest, one may also verify the software capability of correctly imposing boundary conditions. Therefore, the gradients of the analytical solutions (11) have been calculated and translated into C codes. For the 2D case, they are:

$$\nabla\rho = \begin{bmatrix} \dfrac{a_{\rho x}\pi\rho_x}{L}\cos\left(\dfrac{a_{\rho x}\pi x}{L}\right) \\[2ex] -\dfrac{a_{\rho y}\pi\rho_y}{L}\sin\left(\dfrac{a_{\rho y}\pi y}{L}\right) \end{bmatrix}, \qquad \nabla p = \begin{bmatrix} -\dfrac{a_{px}\pi p_x}{L}\sin\left(\dfrac{a_{px}\pi x}{L}\right) \\[2ex] \dfrac{a_{py}\pi p_y}{L}\cos\left(\dfrac{a_{py}\pi y}{L}\right) \end{bmatrix},$$

$$\nabla u = \begin{bmatrix} \dfrac{a_{ux}\pi u_x}{L}\cos\left(\dfrac{a_{ux}\pi x}{L}\right) \\[2ex] -\dfrac{a_{uy}\pi u_y}{L}\sin\left(\dfrac{a_{uy}\pi y}{L}\right) \end{bmatrix}, \qquad \nabla v = \begin{bmatrix} -\dfrac{a_{vx}\pi v_x}{L}\sin\left(\dfrac{a_{vx}\pi x}{L}\right) \\[2ex] \dfrac{a_{vy}\pi v_y}{L}\cos\left(\dfrac{a_{vy}\pi y}{L}\right) \end{bmatrix}.$$

4 Hierarchic MMS

The complexity, and consequently length, of the source terms increase with both dimension and physics handled by the governing equations. Applying commands in order to simplify such extensive expression is challenging even with a very good machine; thus, a suitable alternative to this issue is to simplify the equation by dividing it into a combination of sub-operators handling different physical phenomena. Then, each one of the operators may be applied to the manufactured solutions individually, and the resulting sub-source terms are combined back to represent the source term for the original equation.

For instance, instead of writing the three-dimensional Navier-Stokes energy equation using one single operator $\mathcal{L}$:

$$\mathcal{L} = \frac{\partial(\rho e_t)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} e_t) + \nabla \cdot \boldsymbol{q} + \nabla \cdot (p\boldsymbol{u}) - \nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{u}), \tag{17}$$

to then be used in the MMS, let Equation (17) be written with five operators, according to their physical meaning:

$$\begin{aligned} \mathcal{L}_{\text{time}} &= \frac{\partial(\rho e_t)}{\partial t}, \\ \mathcal{L}_{\text{convection}} &= \nabla \cdot (\rho \boldsymbol{u} e_t), \\ \mathcal{L}_{\text{conduction}} &= \nabla \cdot \boldsymbol{q}, \\ \mathcal{L}_{\text{pressure}} &= \nabla \cdot (p\boldsymbol{u}), \\ \mathcal{L}_{\text{viscous}} &= -\nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{u}), \end{aligned} \tag{18}$$

where $\mathcal{L}_{\text{time}}$ denotes the rate of accumulation of inertial and kinetic energy, $\mathcal{L}_{\text{convection}}$ is the net rate of internal and kinetic energy increase by convection, $\mathcal{L}_{\text{conduction}}$ is the net rate of heat addition due to heat conduction, $\mathcal{L}_{\text{pressure}}$ is the rate of work done on the fluid by external body forces, and $\mathcal{L}_{\text{viscous}}$ is the rate of work done on the fluid by viscous forces. Naturally:

$$\mathcal{L} = \mathcal{L}_{\text{time}} + \mathcal{L}_{\text{convection}} + \mathcal{L}_{\text{conduction}} + \mathcal{L}_{\text{pressure}} + \mathcal{L}_{\text{viscous}}.$$

After the application of each sub-operator defined in (18), the corresponding sub-source terms are also simplified, factorized and sorted. Then, the final factorized version is checked against the original one, in order to assure that no human error has been introduced.

An advantage of this strategy is the possibility of inclusion and/or removal of other physical effects without the need of re-doing previous manipulations. For instance, in order to simplify this model by considering constant viscosity, changes should be made only in operator $\mathcal{L}_{\text{viscous}}$, in the total energy and momentum equations; the other terms on such equations, the continuity equation and the equation for the transport of the scalar all remain the same.

This strategy, named "Hierarchic MMS", results in less time spent in the manipulations, decreases the computational effort and occasional software crashes, and also increases the flexibility in the code verification procedure.

A List of model / manufactured solution parameters

There are a variety of parameters present in the flow described by Navier-Stokes equations with the passive transport of a generic scalar and the Sutherland viscosity model, due to both fluid properties and the constants arising from the chosen manufactured solutions.

Table 1 shows the constants arising from fluid properties and their representation in the C code and Table 2 shows the constants present in the manufactured solutions.

For air, the Sutherland's law coefficients are: $B_\mu = 110.4\,[K]$, $T_{ref} = 273.15\,[K]$, $\mu_{ref} = 1.716 \times 10^{-5}\,[kg/ms]$ and, therefore, $A_\mu = 1.458 \times 10^{-6}[kg/ms\sqrt{K}]$. Additionally, $\gamma = 1.4$, $R = 287\,[J/kgK]$, $Pr = 0.7$.

Table 1: Relations between fluid properties in the model documentation and in the C code.

Variable	Description	Equation	Representation in C
B_μ	Sutherland temperature	(9)	B_mu
T_{ref}	reference temperature	(9), (10)	T_ref
μ_{ref}	reference viscosity	(10)	mu_ref
γ	ratio of specific heats	(8)	Gamma
Pr	Prandtl number	(8)	Pr
R	gas constant	(5)	R
κ	thermal conductivity coefficient	(8)	kappa

Table 2: Constants in the manufactured solutions and their representation in C codes.

Constant	Representation in C	Constant	Representation in C
L	L	a_{ux}	a_ux
L_t	Lt	a_{uy}	a_uy
$a_{\rho x}$	a_rhox	a_{ut}	a_ut
$a_{\rho y}$	a_rhoy	u_0	u_0
ρ_0	rho_0	u_x	u_x
ρ_x	rho_x	u_y	u_y
ρ_y	rho_y	u_t	u_t
ρ_t	rho_t	a_{vx}	a_vx
a_{px}	a_px	a_{vy}	a_vy
a_{py}	a_py	a_{vt}	a_vt
a_{pt}	a_pt	v_0	v_0
p_0	p_0	v_x	v_x
p_x	p_x	v_y	v_y
p_y	p_y	v_t	v_t
p_t	p_t	π	PI

B C Files

The *C* files for both source terms and gradients of the manufactured solutions are:

- NavierStokes_2d_transient_Sutherland_e_code.C
- NavierStokes_2d_transient_Sutherland_rho_code.C
- NavierStokes_2d_transient_Sutherland_u_code.C
- NavierStokes_2d_transient_Sutherland_v_code.C
- NavierStokes_2d_transient_Sutherland_manuf_solutions_grad_code.C

```c
#include <math.h>

double SourceQ_rho (double x, double y, double t)
{
  double RHO;
  double U;
  double V;
  double Q_rho;
  double Q_rho_convection;
  double Q_rho_time;
  RHO = rho_0 + rho_x * sin(a_rhox * PI * x / L) + rho_y * cos(a_rhoy * PI * y / L) + rho_t * sin(
      a_rhot * PI * t / Lt);
  U = u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) + u_t * cos(a_ut * PI * t
      / Lt);
  V = v_0 + v_x * cos(a_vx * PI * x / L) + v_y * sin(a_vy * PI * y / L) + v_t * sin(a_vt * PI * t
      / Lt);
  Q_rho_convection = cos(a_rhox * PI * x / L) * a_rhox * PI * rho_x * U / L - sin(a_rhoy * PI * y
      / L) * a_rhoy * PI * rho_y * V / L + (a_ux * u_x * cos(a_ux * PI * x / L) + a_vy * v_y * cos
      (a_vy * PI * y / L)) * PI * RHO / L;
  Q_rho_time = a_rhot * PI * rho_t * cos(a_rhot * PI * t / Lt) / Lt;
  Q_rho = Q_rho_convection + Q_rho_time;
  return(Q_rho);
}
```

Listing 1: File NavierStokes_2d_transient_Sutherland_rho_code.C

```c
#include <math.h>

double SourceQ_u (double x, double y, double t)
{
  double RHO;
  double P;
  double U;
  double V;
  double T;
  double MU;
  double DMu_Dx;
  double DMu_Dy;
  double Q_u;
  double Q_u_convection;
  double Q_u_pressure;
  double Q_u_viscous;
  double Q_u_time;
  RHO = rho_0 + rho_x * sin(a_rhox * PI * x / L) + rho_y * cos(a_rhoy * PI * y / L) + rho_t * sin(
      a_rhot * PI * t / Lt);
  U = u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) + u_t * cos(a_ut * PI * t
      / Lt);
  V = v_0 + v_x * cos(a_vx * PI * x / L) + v_y * sin(a_vy * PI * y / L) + v_t * sin(a_vt * PI * t
      / Lt);
  P = p_0 + p_x * cos(a_px * PI * x / L) + p_y * sin(a_py * PI * y / L) + p_t * cos(a_pt * PI * t
      / Lt);
  T = P / RHO / R;
  MU = A_mu * pow(T, 0.3e1 / 0.2e1) / (T + B_mu);
```

```
      DMu_Dx = a_rhox * PI * rho_x * MU * MU * cos(a_rhox * PI * x / L) / A_mu / L / RHO / sqrt(T) -
          0.3e1 / 0.2e1 * a_rhox * PI * rho_x * MU * cos(a_rhox * PI * x / L) / L / RHO + a_px * PI *
          p_x * MU * MU * sin(a_px * PI * x / L) / A_mu / L / R / RHO * pow(T, -0.3e1 / 0.2e1) - 0.3e1
          / 0.2e1 * a_px * PI * p_x * MU * sin(a_px * PI * x / L) / L / R / RHO / T;
      DMu_Dy = -a_rhoy * PI * rho_y * MU * MU * sin(a_rhoy * PI * y / L) / A_mu / L / RHO / sqrt(T) +
          0.3e1 / 0.2e1 * a_rhoy * PI * rho_y * MU * sin(a_rhoy * PI * y / L) / L / RHO - a_py * PI *
          p_y * MU * MU * cos(a_py * PI * y / L) / A_mu / L / R / RHO * pow(T, -0.3e1 / 0.2e1) + 0.3e1
          / 0.2e1 * a_py * PI * p_y * MU * cos(a_py * PI * y / L) / L / R / RHO / T;
      Q_u_convection = a_rhox * PI * rho_x * U * U * cos(a_rhox * PI * x / L) / L - a_rhoy * PI *
          rho_y * U * V * sin(a_rhoy * PI * y / L) / L - a_uy * PI * u_y * RHO * V * sin(a_uy * PI * y
          / L) / L + (0.2e1 * a_ux * u_x * cos(a_ux * PI * x / L) + a_vy * v_y * cos(a_vy * PI * y /
          L)) * PI * RHO * U / L;
      Q_u_pressure = -a_px * PI * p_x * sin(a_px * PI * x / L) / L;
      Q_u_viscous = (0.4e1 * a_ux * a_ux * u_x * sin(a_ux * PI * x / L) + 0.3e1 * a_uy * a_uy * u_y *
          cos(a_uy * PI * y / L)) * MU * PI * PI * pow(L, -0.2e1) / 0.3e1 - 0.2e1 / 0.3e1 * (0.2e1 *
          a_ux * u_x * cos(a_ux * PI * x / L) - a_vy * v_y * cos(a_vy * PI * y / L)) * DMu_Dx * PI / L
          + (a_uy * u_y * sin(a_uy * PI * y / L) + a_vx * v_x * sin(a_vx * PI * x / L)) * DMu_Dy * PI
          / L;
      Q_u_time = a_rhot * PI * rho_t * U * cos(a_rhot * PI * t / Lt) / Lt - a_ut * PI * u_t * RHO *
          sin(a_ut * PI * t / Lt) / Lt;
      Q_u = Q_u_convection + Q_u_pressure + Q_u_viscous + Q_u_time;
      return(Q_u);
}
```

Listing 2: File NavierStokes_2d_transient_Sutherland_u_code.C

```
#include <math.h>

double SourceQ_v (double x, double y, double t)
{
  double RHO;
  double P;
  double U;
  double V;
  double T;
  double MU;
  double DMu_Dx;
  double DMu_Dy;
  double Q_v;
  double Q_v_convection;
  double Q_v_pressure;
  double Q_v_viscous;
  double Q_v_time;
  RHO = rho_0 + rho_x * sin(a_rhox * PI * x / L) + rho_y * cos(a_rhoy * PI * y / L) + rho_t * sin(
      a_rhot * PI * t / Lt);
  U = u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) + u_t * cos(a_ut * PI * t
      / Lt);
  V = v_0 + v_x * cos(a_vx * PI * x / L) + v_y * sin(a_vy * PI * y / L) + v_t * sin(a_vt * PI * t
      / Lt);
  P = p_0 + p_x * cos(a_px * PI * x / L) + p_y * sin(a_py * PI * y / L) + p_t * cos(a_pt * PI * t
      / Lt);
  T = P / RHO / R;
  MU = A_mu * pow(T, 0.3e1 / 0.2e1) / (T + B_mu);
  DMu_Dx = a_rhox * PI * rho_x * MU * MU * cos(a_rhox * PI * x / L) / A_mu / L / RHO / sqrt(T) -
      0.3e1 / 0.2e1 * a_rhox * PI * rho_x * MU * cos(a_rhox * PI * x / L) / L / RHO + a_px * PI *
      p_x * MU * MU * sin(a_px * PI * x / L) / A_mu / L / R / RHO * pow(T, -0.3e1 / 0.2e1) - 0.3e1
      / 0.2e1 * a_px * PI * p_x * MU * sin(a_px * PI * x / L) / L / R / RHO / T;
  DMu_Dy = -a_rhoy * PI * rho_y * MU * MU * sin(a_rhoy * PI * y / L) / A_mu / L / RHO / sqrt(T) +
      0.3e1 / 0.2e1 * a_rhoy * PI * rho_y * MU * sin(a_rhoy * PI * y / L) / L / RHO - a_py * PI *
      p_y * MU * MU * cos(a_py * PI * y / L) / A_mu / L / R / RHO * pow(T, -0.3e1 / 0.2e1) + 0.3e1
      / 0.2e1 * a_py * PI * p_y * MU * cos(a_py * PI * y / L) / L / R / RHO / T;
  Q_v_convection = a_rhox * PI * rho_x * U * V * cos(a_rhox * PI * x / L) / L - a_rhoy * PI *
      rho_y * V * V * sin(a_rhoy * PI * y / L) / L - a_vx * PI * v_x * RHO * U * sin(a_vx * PI * x
      / L) / L + (a_ux * u_x * cos(a_ux * PI * x / L) + 0.2e1 * a_vy * v_y * cos(a_vy * PI * y /
      L)) * PI * RHO * V / L;
  Q_v_pressure = a_py * PI * p_y * cos(a_py * PI * y / L) / L;
  Q_v_viscous = (0.3e1 * a_vx * a_vx * v_x * cos(a_vx * PI * x / L) + 0.4e1 * sin(a_vy * PI * y /
      L) * a_vy * a_vy * v_y) * MU * PI * PI * pow(L, -0.2e1) / 0.3e1 + (a_uy * u_y * sin(a_uy *
```

```
      PI * y / L) + a_vx * v_x * sin(a_vx * PI * x / L)) * DMu_Dx * PI / L - 0.2e1 / 0.3e1 * (-
      a_ux * u_x * cos(a_ux * PI * x / L) + 0.2e1 * a_vy * v_y * cos(a_vy * PI * y / L)) * DMu_Dy
      * PI / L;
  Q_v_time = a_vt * v_t * PI * RHO * cos(a_vt * PI * t / Lt) / Lt + a_rhot * rho_t * PI * V * cos(
      a_rhot * PI * t / Lt) / Lt;
  Q_v = Q_v_convection + Q_v_pressure + Q_v_viscous + Q_v_time;
  return(Q_v);
}
```

Listing 3: File NavierStokes_2d_transient_Sutherland_v_code.C

```
#include <math.h>

double SourceQ_e (double x, double y, double t)
{
  double RHO;
  double P;
  double U;
  double V;
  double T;
  double MU;
  double DMu_Dx;
  double DMu_Dy;
  double kappa;
  double Q_e;
  double Q_e_convection;
  double Q_e_work_pressure;
  double Q_e_work_viscous;
  double Q_e_conduction;
  double Q_e_time;
  RHO = rho_0 + rho_x * sin(a_rhox * PI * x / L) + rho_y * cos(a_rhoy * PI * y / L) + rho_t * sin(
      a_rhot * PI * t / Lt);
  U = u_0 + u_x * sin(a_ux * PI * x / L) + u_y * cos(a_uy * PI * y / L) + u_t * cos(a_ut * PI * t
      / Lt);
  V = v_0 + v_x * cos(a_vx * PI * x / L) + v_y * sin(a_vy * PI * y / L) + v_t * sin(a_vt * PI * t
      / Lt);
  P = p_0 + p_x * cos(a_px * PI * x / L) + p_y * sin(a_py * PI * y / L) + p_t * cos(a_pt * PI * t
      / Lt);
  T = P / RHO / R;
  MU = A_mu * pow(T, 0.3e1 / 0.2e1) / (T + B_mu);
  DMu_Dx = a_rhox * PI * rho_x * MU * MU * cos(a_rhox * PI * x / L) / A_mu / L / RHO / sqrt(T) -
      0.3e1 / 0.2e1 * a_rhox * PI * rho_x * MU * cos(a_rhox * PI * x / L) / L / RHO + a_px * PI *
      p_x * MU * MU * sin(a_px * PI * x / L) / A_mu / L / R / RHO * pow(T, -0.3e1 / 0.2e1) - 0.3e1
      / 0.2e1 * a_px * PI * p_x * MU * sin(a_px * PI * x / L) / L / R / RHO / T;
  DMu_Dy = -a_rhoy * PI * rho_y * MU * MU * sin(a_rhoy * PI * y / L) / A_mu / L / RHO / sqrt(T) +
      0.3e1 / 0.2e1 * a_rhoy * PI * rho_y * MU * sin(a_rhoy * PI * y / L) / L / RHO - a_py * PI *
      p_y * MU * MU * cos(a_py * PI * y / L) / A_mu / L / R / RHO * pow(T, -0.3e1 / 0.2e1) + 0.3e1
      / 0.2e1 * a_py * PI * p_y * MU * cos(a_py * PI * y / L) / L / R / RHO / T;
  kappa = Gamma * R * MU / (Gamma - 0.1e1) / Pr;
  Q_e_convection = a_rhox * PI * rho_x * pow(U, 0.3e1) * cos(a_rhox * PI * x / L) / L / 0.2e1 +
      a_rhox * PI * rho_x * U * V * V * cos(a_rhox * PI * x / L) / L / 0.2e1 - a_rhoy * PI * rho_y
      * U * U * V * sin(a_rhoy * PI * y / L) / L / 0.2e1 - a_rhoy * PI * rho_y * pow(V, 0.3e1) *
      sin(a_rhoy * PI * y / L) / L / 0.2e1 - a_px * PI * p_x * U * sin(a_px * PI * x / L) / (Gamma
      - 0.1e1) / L + a_py * PI * p_y * V * cos(a_py * PI * y / L) / (Gamma - 0.1e1) / L + (0.3e1
      * a_ux * u_x * cos(a_ux * PI * x / L) + a_vy * v_y * cos(a_vy * PI * y / L)) * PI * RHO * U
      * U / L / 0.2e1 - (a_uy * u_y * sin(a_uy * PI * y / L) + a_vx * v_x * sin(a_vx * PI * x / L)
      ) * PI * RHO * U * V / L + (a_ux * u_x * cos(a_ux * PI * x / L) + 0.3e1 * a_vy * v_y * cos(
      a_vy * PI * y / L)) * PI * RHO * V * V / L / 0.2e1 + (a_ux * u_x * cos(a_ux * PI * x / L) +
      a_vy * v_y * cos(a_vy * PI * y / L)) * PI * P / (Gamma - 0.1e1) / L;
  Q_e_work_pressure = -a_px * PI * p_x * U * sin(a_px * PI * x / L) / L + a_py * PI * p_y * V *
      cos(a_py * PI * y / L) / L + (a_ux * u_x * cos(a_ux * PI * x / L) + a_vy * v_y * cos(a_vy *
      PI * y / L)) * PI * P / L;
  Q_e_conduction = kappa * DMu_Dx * a_rhox * PI * rho_x * P * cos(a_rhox * PI * x / L) / L / R /
      MU * pow(RHO, -0.2e1) + kappa * DMu_Dx * a_px * PI * p_x * sin(a_px * PI * x / L) / L / R /
      MU / RHO - kappa * DMu_Dy * a_rhoy * PI * rho_y * P * sin(a_rhoy * PI * y / L) / L / R / MU
      * pow(RHO, -0.2e1) - kappa * DMu_Dy * a_py * PI * p_y * cos(a_py * PI * y / L) / L / R / MU
      / RHO + (a_px * a_px * p_x * cos(a_px * PI * x / L) + a_py * a_py * p_y * sin(a_py * PI * y
      / L)) * kappa * PI * PI * pow(L, -0.2e1) / R / RHO - (a_rhox * a_rhox * rho_x * sin(a_rhox *
```

```
        PI * x / L) + a_rhoy * a_rhoy * rho_y * cos(a_rhoy * PI * y / L)) * kappa * PI * PI * P *
        pow(L, -0.2e1) / R * pow(RHO, -0.2e1) - (0.2e1 * a_rhox * a_px * rho_x * p_x * cos(a_rhox *
        PI * x / L) * sin(a_px * PI * x / L) + 0.2e1 * a_rhoy * a_py * rho_y * p_y * sin(a_rhoy * PI
        * y / L) * cos(a_py * PI * y / L)) * kappa * PI * PI * pow(L, -0.2e1) / R * pow(RHO, -0.2e1
        ) - (0.2e1 * a_rhox * a_rhox * rho_x * rho_x * pow(cos(a_rhox * PI * x / L), 0.2e1) + 0.2e1
        * a_rhoy * a_rhoy * rho_y * rho_y * pow(sin(a_rhoy * PI * y / L), 0.2e1)) * kappa * PI * PI
        * P * pow(L, -0.2e1) / R * pow(RHO, -0.3e1);
  Q_e_time = -a_ut * u_t * PI * RHO * U * sin(a_ut * PI * t / Lt) / Lt + a_vt * v_t * PI * RHO * V
        * cos(a_vt * PI * t / Lt) / Lt + a_rhot * rho_t * PI * U * U * cos(a_rhot * PI * t / Lt) /
        Lt / 0.2e1 + a_rhot * rho_t * PI * V * V * cos(a_rhot * PI * t / Lt) / Lt / 0.2e1 - sin(a_pt
        * PI * t / Lt) * a_pt * p_t * PI / (Gamma - 0.1e1) / Lt;
  Q_e = Q_e_convection + Q_e_work_pressure + Q_e_work_viscous + Q_e_conduction + Q_e_time;
  return(Q_e);
}
```

Listing 4: File `NavierStokes_2d_transient_Sutherland_e_code.C`

```
rho_an = rho_0 + rho_x * sin(a_rhox * pi * x / L) + rho_y * cos(a_rhoy * pi * y / L) + rho_t * sin
    (a_rhot * pi * t / Lt);
p_an = p_0 + p_x * cos(a_px * pi * x / L) + p_y * sin(a_py * pi * y / L) + p_t * cos(a_pt * pi * t
    / Lt);
u_an = u_0 + u_x * sin(a_ux * pi * x / L) + u_y * cos(a_uy * pi * y / L) + u_t * cos(a_ut * pi * t
    / Lt);
v_an = v_0 + v_x * cos(a_vx * pi * x / L) + v_y * sin(a_vy * pi * y / L) + v_t * sin(a_vt * pi * t
    / Lt);
grad_rho_an[0] = rho_x * cos(a_rhox * pi * x / L) * a_rhox * pi / L;
grad_rho_an[1] = -rho_y * sin(a_rhoy * pi * y / L) * a_rhoy * pi / L;
grad_p_an[0] = -p_x * sin(a_px * pi * x / L) * a_px * pi / L;
grad_p_an[1] = p_y * cos(a_py * pi * y / L) * a_py * pi / L;
grad_u_an[0] = u_x * cos(a_ux * pi * x / L) * a_ux * pi / L;
grad_u_an[1] = -u_y * sin(a_uy * pi * y / L) * a_uy * pi / L;
grad_v_an[0] = -v_x * sin(a_vx * pi * x / L) * a_vx * pi / L;
grad_v_an[1] = v_y * cos(a_vy * pi * y / L) * a_vy * pi / L;
```

Listing 5: File `NavierStokes_2d_transient_Sutherland_manuf_solutions_grad_code.C`

References

Maplesoft (2011, November). Maple$^{\text{TM}}$15: the essential tool for mathematics and modeling. http://www.maplesoft.com/Products/Maple/.

Roy, C., T. Smith, and C. Ober (2002). Verification of a Compressible CFD Code using the Method of Manufactured Solutions. In *32nd AIAA Fluid Dynamics Conference and Exhibit*, Number AIAA 2002-3110.

Sutherland, W. (1893). The viscosity of gases and molecular force. *Philosophical Magazine 5*(36), 507–531.