```
# Additional constants and definition of compressible reference state:
Ra = Constant(1e5) # Rayleigh number
Di = Constant(0.5) # Dissipation number
T0 = Constant(0.091) # Non-dimensional surface temperature
tcond = Constant(1.0) # Thermal conductivity
rho_0, alpha, cpr, cvr, gruneisen = 1.0, 1.0, 1.0, 1.0, 1.0
rhobar = Function(Q, name="CompRefDensity").interpolate(rho_0 * exp(((1.0 - X[1]) * Di) / alpha))
Tbar = Function(Q, name="CompRefTemperature").interpolate(T0 * exp((1.0 - X[1]) * Di) - T0)
alphabar = Function(Q, name="IsobaricThermalExpansivity").assign(1.0)
cpbar = Function(Q, name="IsobaricSpecificHeatCapacity").assign(1.0)
chibar = Function(Q, name="IsothermalBulkModulus").assign(1.0)
FullT = Function(Q, name="FullTemperature").assign(Tnew+Tbar)

--------------------------------------------------------------------------------------------------
# Equations in UFL:
I = Identity(2)
stress = 2 * mu * sym(grad(u)) - 2./3.*I*mu*div(u)
F_stokes = inner(grad(v), stress) * dx - div(v) * p * dx - (dot(v, k) * (Ra * Ttheta * rhobar * alphabar - (Di/gruneisen) * (cpr/cvr)*rhobar*chibar*p) * dx)
F_stokes += -w * div(rhobar*u) * dx  # Mass conservation
F_energy = q * rhobar * cpbar * ((Tnew - Told) / delta_t) * dx + q * rhobar * cpbar * dot(u, grad(Ttheta)) * dx + dot(grad(q), tcond * grad(Tbar + Ttheta)) * dx + q * (alphabar * rhobar * Di * u[1] * Ttheta) * dx  - q * ((Di/Ra) * inner(stress, grad(u)) ) * dx

--------------------------------------------------------------------------------------------------
# Temperature boundary conditions:
bctb, bctt = DirichletBC(Q, 1.0 - (T0*exp(Di) - T0), bottom), DirichletBC(Q, 0.0, top)

--------------------------------------------------------------------------------------------------
# Pressure nullspace:
stokes_solver = NonlinearVariationalSolver(stokes_problem, solver_parameters=solver_parameters, transpose_nullspace=p_nullspace)
```