Heat Transfer Boundary Condition Test

Objective

This test problem tests all exterior boundary conditions that are available in the heat transfer module, except viewfactor radiation.

Definition

We consider the 3D conductive heat transfer problem that is defined as follows. The problem domain is the cube $[0, 11]^3$, which is discretized using a logically rectangular hex mesh of size $11 \times 11 \times 11$. The conductivity is represented by the polynomial

$$K(T) = 1 + \frac{T}{10} + \frac{T^2}{1000}.$$

The boundary conditions are a constant Dirichlet boundary conditions on the boundaries $x = 0$, $y = 0$, and $z = 0$

$$\begin{aligned} T(x, y, z) &= 10, \quad x = 0 \\ T(x, y, z) &= 20, \quad y = 0 \\ T(x, y, z) &= 30, \quad z = 0. \end{aligned}$$

On the boundary at $x = 11$ we prescribe a radiative heat loss boundary condition where the reference temperature is $T_0 = 0$ and emissivity $\epsilon = 1$, such that

$$q = -\epsilon K(T^4 - T_0^4),$$

where K is the conductivity and T the temperature on the boundary. The boundary at $y = 11$, has a homogeneous Neuman boundary condition

$$\mathbf{n} \cdot \nabla T(x, y, z) = 0, \quad y = 11,$$

and the boundary $z = 11$ has a convective heat loss boundary condition that is implemented using the HTC boundary condition type. The parameters are the reference temperature $T_0 = 50$, and the heat transfer coefficient $\beta = 50$, such that

$$q = -\beta(T - T_0).$$

Metrics

We compare the temperature at the final time step to the temperature that was obtained in a reference run (golden output). We use the ℓ^∞-norm for this comparison.

Truchas Model

To run this example in truchas, we select the heat conduction model exclusively. The boundary conditions are selected as described above. The heat transfer problem is solved using Newton Krylov with flexible GMRES as a preconditioner, which is itself preconditioned by SSOR.

We this problem for ten time steps on two different meshes. The first mesh is an orthogonal hex mesh, while the second mesh is only logically rectangular. It was generated from the orthogonal mesh by randomizing the interior vertices by at most 20% of their original position (see the cubit journal file for the randomized mesh).

On the orthogonal mesh, we use the ortho operator and on the randomized mesh, we use the support operator discretization.

Results

The results of these runs are compared to a reference run (golden output) as there is no analytic solution to this problem.