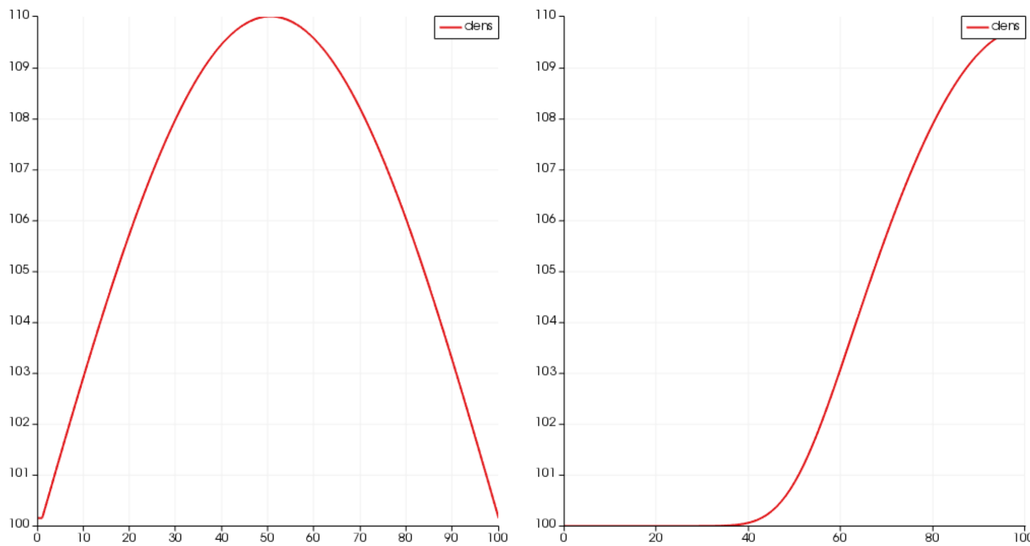
1 Advection

1.1 advec1-t

- *advec1-t.i*
- 1D generated mesh with libmesh
- Uses DG Kernels
- InflowBC and OutflowBC
- Transient problem

Figure 1 shows the results. Advects BC. It seems like the variable has to be a CONSTANT MONOMIAL.

$$\frac{\partial}{\partial t}\rho + v\frac{\partial}{\partial x}\rho = 0 \tag{1}$$



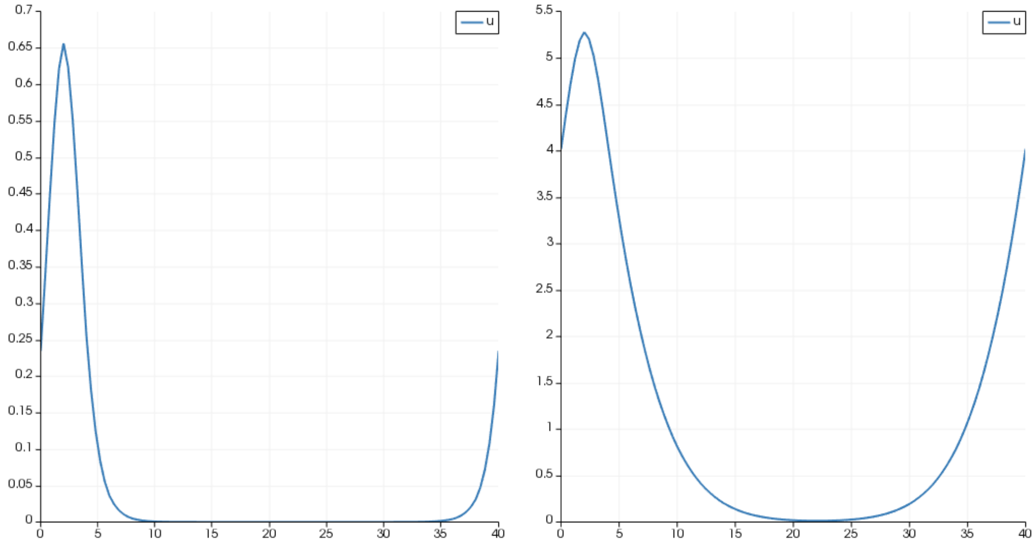
(a) t=0. (b) t=100.

Figure 1: Advected density.

1.2 periodic_bc2

- *moose/examples/ex04_bcs/periodic_bc2.i*
- 1D generated mesh with libmesh
- Periodic BCs
- Transient problem

In *advec1-t-bc.i* I tried to add periodicBCs to the previous problem and it does not work. Here I tried to isolate the problem. Figure 2 shows the results. It does not work if the valiable is a CONSTANT MONOMIAL. It works if the variable is FIRST order (either MONOMIAL or LAGRANGE).



(a) t=0.

(b) t=20.

Figure 2: Periodic BCs.

1.3 advec2-t

- *advec2-t.i*
- 1D generated mesh with libmesh
- Uses DG Kernels
- TemperatureInflowBC and TemperatureOutflowBC
- Transient problem

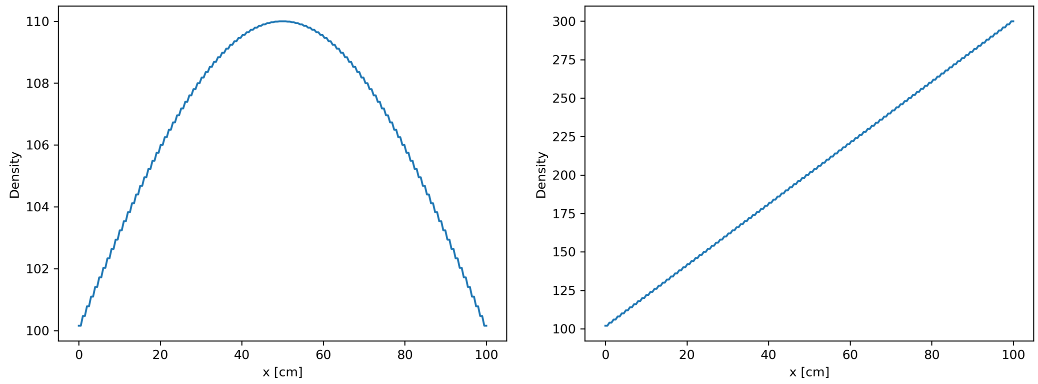
Very similar to *advec1-t.i*. Adds volumetric source. Figure 3 shows the results. It is correct. $\rho(L, t \to \infty) - \rho(0, t) = q/v * L = 200$

$$\frac{\partial}{\partial t}\rho + v\frac{\partial}{\partial x}\rho = \dot{q} \tag{2}$$

- IC: $\rho(x, 0) = 100 + 10sin(\frac{\pi}{L}x)$
- BC: $\rho(0, t) = 100$
- $v = 0.5$
- $L = 100$
- $q = 1$

1.4 advec2-ss

- *advec2-ss.i*
- 1D generated mesh with libmesh



(a) t=0.

(b) t=250.

Figure 3: Advected density with volumentric source.

- Uses DG Kernels
- Inflow and OutflowBC
- Steady problem

Same as *advec2-t* but steady state. Figure 4 shows the results. It is correct. $\rho(L) - \rho(0) = q/v * L = 200$

$$v \frac{\partial}{\partial x} \rho = \dot{q} \tag{3}$$

- BC: $\rho(0, t) = 100$
- $v = 0.5$
- $L = 100$
- $q = 1$

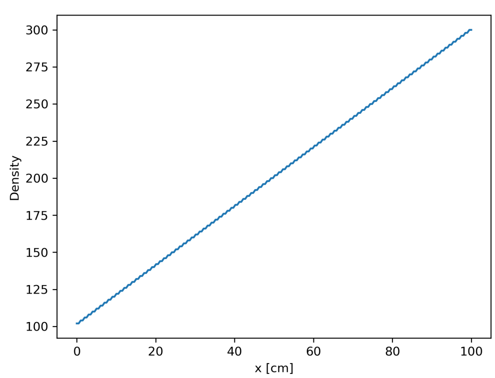


Figure 4: Steady state solution.

1.5 advec3-t

- *advec3-t.i*
- 1D generated mesh with libmesh
- Uses DG Kernels
- TemperatureInflowBC and TemperatureOutflowBC
- Transient problem

Very similar to *advec1-t.i*. Solves for the temperature advection equation. Advects BC. Figure 5 shows the results.

$$\rho c_p \frac{\partial}{\partial t} T + \rho c_p v \frac{\partial}{\partial x} T = 0 \tag{4}$$

- BC: $T(0,t) = 930$
- IC: $T(x,0) = 930$
- $\rho = 1e-2$
- $c_p = 2e3$
- $v = 0.5$
- $L = 100$

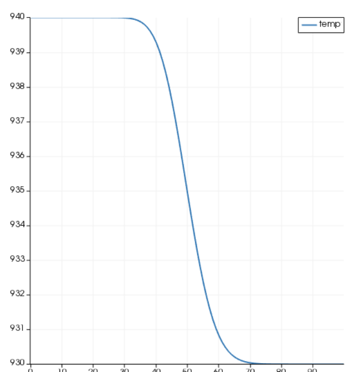


Figure 5: Advects BC.

1.6 advec4-t

- *advec4-t.i*
- 1D generated mesh with libmesh
- Uses DG Kernels
- TemperatureInflowBC and TemperatureOutflowBC
- Transient problem

Similar to *advec4-t.i* Adds a point source and solves for temperature. Figure 6 shows the results. It is correct. $T(L) - T(0) = q/(\rho c_p v) * L = 10$

$$\rho c_p \frac{\partial}{\partial t} T + \rho c_p v \frac{\partial}{\partial x} T = \dot{q} \tag{5}$$

- BC: $T(0, t) = 930$
- IC: $T(x, 0) = 930$
- $\rho = 1e - 2$
- $c_p = 2e3$
- $v = 0.5$
- $L = 100$
- $q = 1$

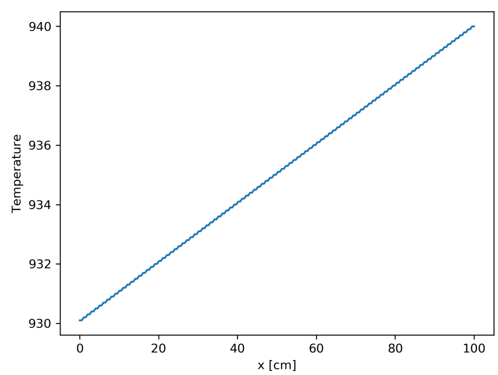


Figure 6: t=250.

1.7 advec5-t

- *advec5-t.i*
- pseudo-1D: GeneratedMesh
- Uses DG Kernels
- TemperatureInflowBC and TemperatureOutflowBC
- Transient problem

Similar to *advec4-t.i* but has a q'' on the wall. Figure 7 shows the results.

$$\rho c_p \frac{\partial}{\partial t} T + \rho c_p v \frac{\partial}{\partial x} T = 0 \tag{6}$$

This is not the real equation. When using the Galerkin method, a new term appears due to the neumann BC.

- IC: $T(x, y, 0) = 930$

- BC: $T(x, 0, t) = 930$
- BC: $q''(0, y, t) = 10 sin(\pi/Ly)$
- $\rho = 1e - 2$
- $c_p = 2e3$
- $v = 0.5$
- $L = 100$
- $\Delta_x = 2$

$$T(L) - T(0) = \frac{1}{\rho c_p v} \frac{1}{\Delta_x \Delta_z} \int_0^L q'' dy \Delta_z = \frac{1}{10} \frac{1}{2} \frac{10 \times 2 \times L}{\pi} \tag{7}$$

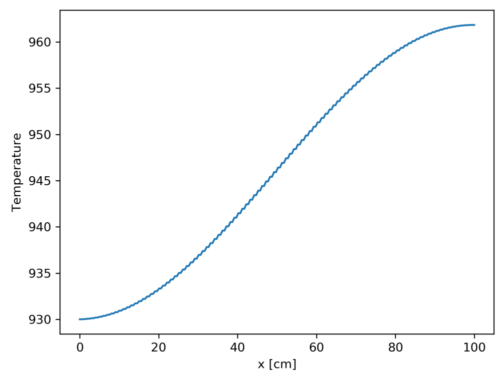


Figure 7: Advects temperature while wall is been heated.

1.8 advec5-ss

- *advec5-ss.i*
- pseudo-1D: GeneratedMesh
- Uses DG Kernels
- TemperatureInflowBC and TemperatureOutflowBC
- Steady state problem

Steady state version of *advec5-t.i*. Figure 8 shows the results.

$$\rho c_p v \frac{\partial}{\partial x} T = 0 \tag{8}$$

This is not the real equation. When using the Galerkin method, a new term appears due to the neumann BC.

- BC: $T(x, 0) = 930$
- BC: $q''(0, y) = 10 sin(\pi/Ly)$

- $\rho = 1e-2$
- $c_p = 2e3$
- $v = 0.5$
- $L = 100$
- $\Delta_x = 2$

$$T(L) = T(0) + \frac{1}{\rho c_p v}\frac{1}{\Delta_x \Delta_z}\int_0^L q'' dy \Delta_z = T(0) + \frac{1}{10}\frac{1}{2}\frac{10 \times 2 \times L}{\pi} = 961 \tag{9}$$

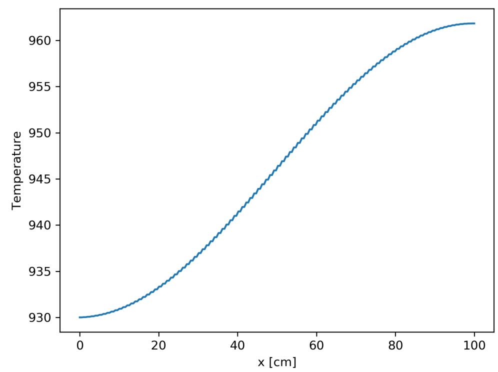


Figure 8: Advects temperature while wall is been heated.

1.9 advec6-t

- *advec6-t.i*
- Mesh: 2D-coolant.msh
- Uses DG Kernels
- TemperatureInflowBC and TemperatureOutflowBC
- Transient problem

Like *advec5-t.i* but uses the values of the PMR600 (or close values). Figure 9 shows the results. Constanst came from [2] and [1].

$$\rho c_p \frac{\partial}{\partial t} T + \rho c_p v \frac{\partial}{\partial x} T = 0 \tag{10}$$

This is not the real equation. When using the Galerkin method, a new term appears due to the neumann BC.

- BC: $T(x, 0) = 490°C$
- BC: $q''(0, y) = 22.24 sin(\pi/Ly)$
- $\rho(7MPa, 490°C) = 4.368 kg/m^3 = 4.368e-6 kg/cm^3$

- $c_p(7MPa, 490°C) = 5.188J/g/K = 5.188e3J/kg/K$
- $v = 26.57m/s = 2657cm/s$
- $L = 793cm$
- $R = 0.794cm$

$$T(L) - T(0) = \frac{1}{\rho c_p v} \frac{1}{\pi R^2} \int_0^L q'' dy 2\pi R = \frac{1}{60.2} \frac{2}{R} \frac{20 \times 2 \times L}{\pi} \tag{11}$$

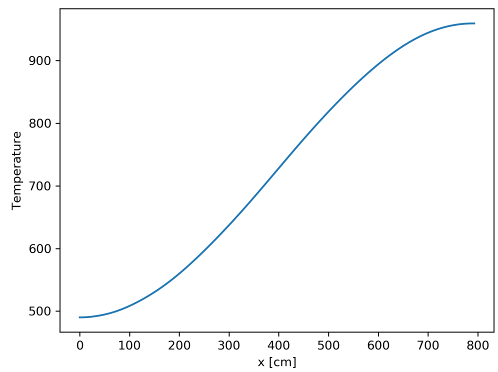


Figure 9: Advects temperature while wall is been heated.

1.10 advec6-ss

- *advec6-ss.i*
- Mesh: 2D-coolant.msh
- Uses DG Kernels
- TemperatureInflowBC and TemperatureOutflowBC
- Steady-state problem

Like *advec6-t.i* but steady state. Figure 10 shows the results.

$$\rho c_p v \frac{\partial}{\partial x} T = 0 \tag{12}$$

This is not the real equation. When using the Galerkin method, a new term appears due to the neumann BC.

1.11 diff1-ss

- *diff1-ss.i*
- GeneratedMesh
- Uses DG Kernels
- DGFunctionDiffusionDirichletBC

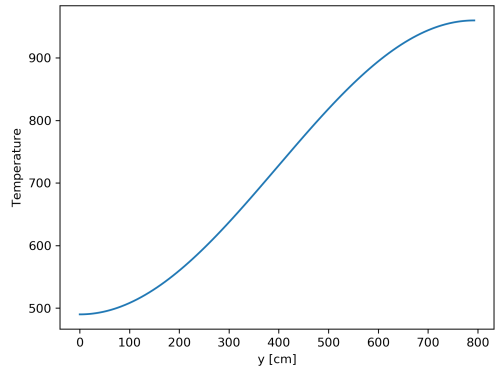


Figure 10: Advects temperature while wall is been heated.

- Steady-state problem

Figure 11 shows the results.

$$k\nabla^2 T + q = 0 \tag{13}$$

- BC: $T(2, y) = 490°C$
- $k = 1$
- $q = 1$
- $\Delta_x = 2$

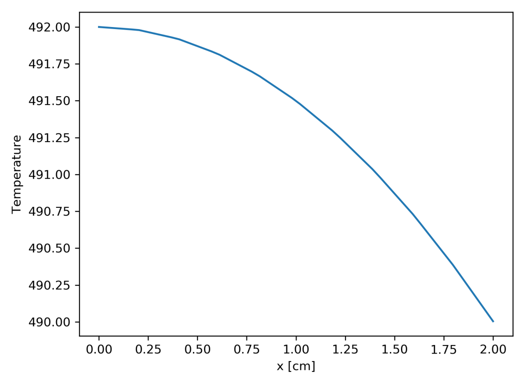


Figure 11: Diffusion using DG Kernels.

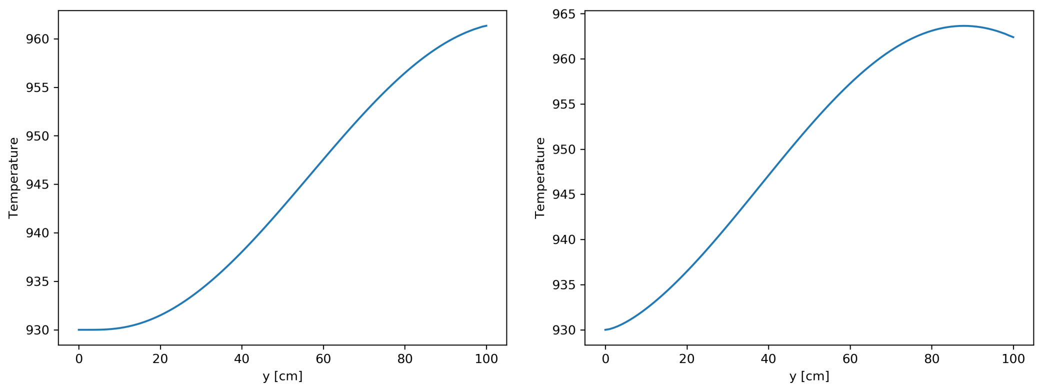
1.12 cg-advec1-ss

- *cg-advec1-ss.i*
- 2D GeneratedMesh
- Uses CG Kernels
- Steady-state problem

Figure 12 and 13 shows the results.

$$k\nabla^2 T + q = 0 \tag{14}$$

- BC: $T(x,0) = 930°C$
- $q = 10sin(\pi/Ly)$
- $L = 100$
- $\Delta_x = 2$
- $\rho = 1e-2$
- $c_p = 2e3$
- $k = 1$



(a) (0,0,0)-(0,100,0).　(b) (2,0,0)-(2,100,0).

Figure 12: Steady state advection diffusion.

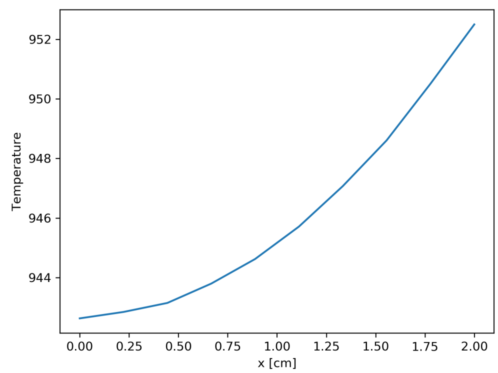


Figure 13: Steady state advection diffusion on (0,50,0)-(2,50,0).

1.13 cg-advec2-ss

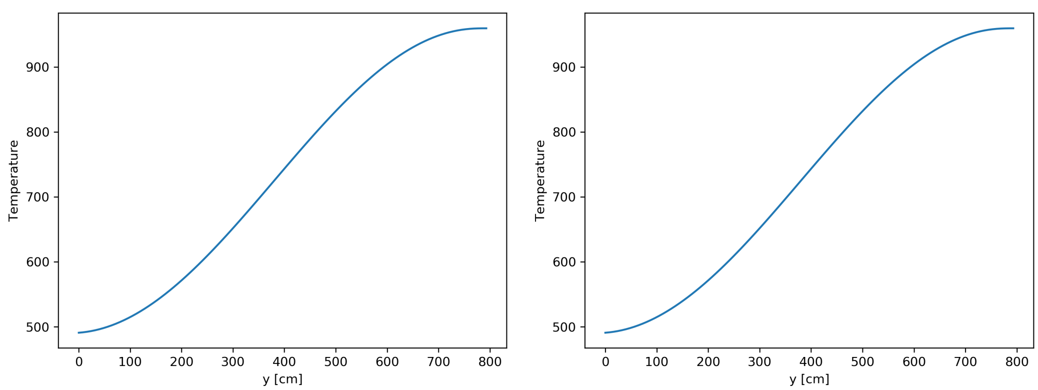
- *cg-advec2-ss.i*
- mesh: 2D-coolantB.msh
- Uses CG Kernels
- Steady-state problem

Figure 14 and 15 shows the results.

$$\rho c_p v \frac{\partial}{\partial x} T = 0 \tag{15}$$

This is not the real equation. When using the Galerkin method, a new term appears due to the neumann BC.

- BC: $T(x, 0) = 490°C$
- $q''(0, y) = 22.24 sin(\pi/Ly) W/cm^2$
- $L = 793cm$
- $R_C = 0.794cm$
- $\rho(7MPa, 490°C) = 4.368kg/m^3 = 4.368e-6kg/cm^3$
- $c_p(7MPa, 490°C) = 5.188J/g/K = 5.188e3J/kg/K$
- $v = 26.57m/s = 2657cm/s$
- $k = 10W/m/K = 1e3W/cm/K$



(a) (0,0,0)-(0,793,0). (b) (0.794,793,0)-(0.794,793,0).

Figure 14: Steady state advection diffusion.

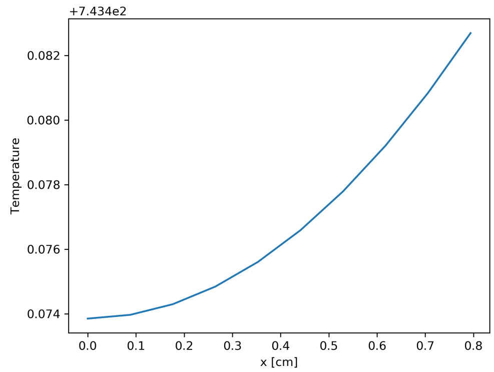


Figure 15: Steady state advection diffusion on (0,400,0)-(0.794,400,0).

1.14 cg-advec3-ss

- *cg-advec3-ss.i*
- mesh: 3D-coolant.msh
- Uses CG Kernels
- Steady-state problem

Figure 16 shows the results.

$$\rho c_p v \frac{\partial}{\partial x} T = 0 \tag{16}$$

This is not the real equation. When using the Galerkin method, a new term appears due to the neumann BC.

- BC: $T(r, 0) = 490°C$
- $q''(R, z) = 22.24 sin(\pi/Lz) W/cm^2$
- $L = 793cm$
- $R_C = 0.794cm$
- $\rho(7MPa, 490°C) = 4.368kg/m^3 = 4.368e-6kg/cm^3$
- $c_p(7MPa, 490°C) = 5.188J/g/K = 5.188e3J/kg/K$
- $v = 26.57m/s = 2657cm/s$
- $k = 10W/m/K = 1e3W/cm/K$

1.15 cg-advec4-ss

- *cg-advec4-ss.i*
- mesh: 3D-cell.msh
- Uses CG Kernels

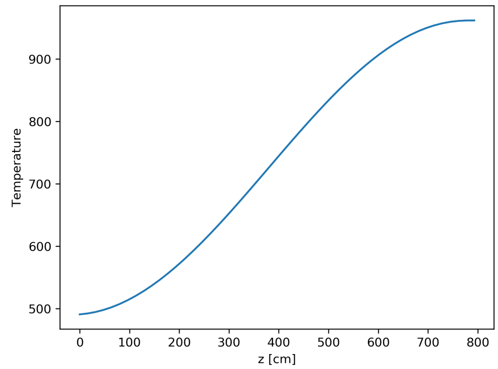


Figure 16: Steady state advection diffusion on (0,0,0)-(0,0,793).

- Steady-state problem

Figure 18 displays the geometry of the unit cell. Figure 17 shows the thermal conductivities of the fuel and moderator results. Figure 19 and 20 show the results.

$$\rho c_p v \frac{\partial}{\partial x} T = 0 \tag{17}$$

This is not the real equation. When using the Galerkin method, a new term appears due to the neumann BC.

- BC: $T(r, 0) = 490°C$
- $q''(R, z) = 43.8 sin(\pi/Lz) W/cm^3$
- $L = 793cm$
- $R_C = 0.794cm$
- $R_F = 0.635cm$
- $\rho_C(7MPa, 490°C) = 4.368kg/m^3 = 4.368e - 6kg/cm^3$
- $c_{p,C}(7MPa, 490°C) = 5.188J/g/K = 5.188e3J/kg/K$
- $v = 26.57m/s = 2657cm/s$
- $k_C = 10W/m/K = 1e3W/cm/K$
- $k_F(T)$
- $k_M(T)$

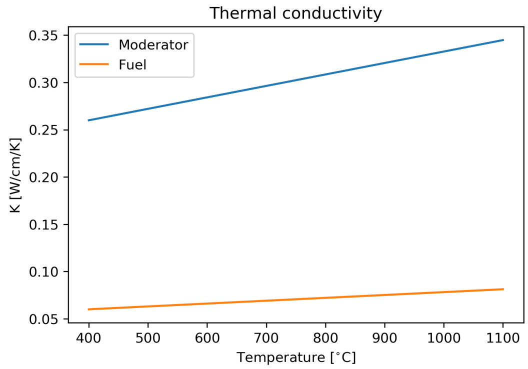


Figure 17: Thermal conductivity of the fuel and the moderator.

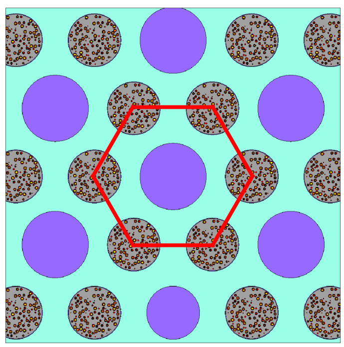
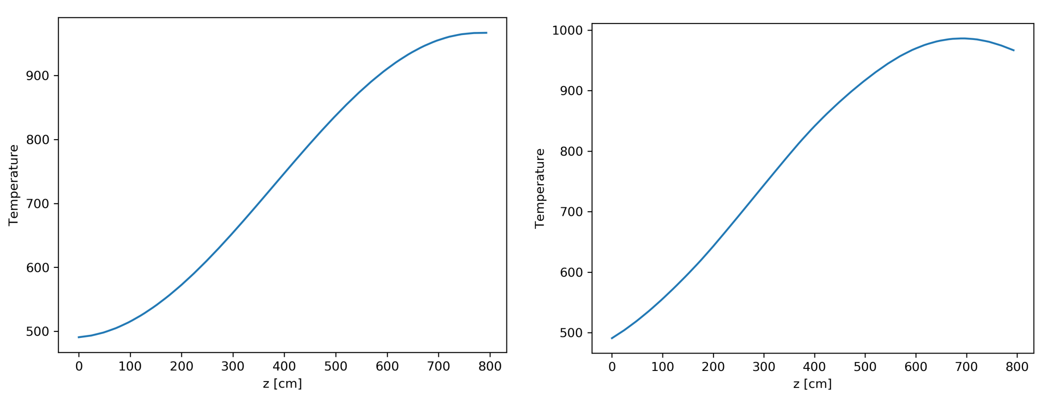


Figure 18: Unit cell geometry.



(a) Coolant centerline: (0,0,0)-(0,793,0).

(b) Fuel centerline: (1.88,0,0)-(1.88,793,0).

Figure 19: Steady state advection diffusion.

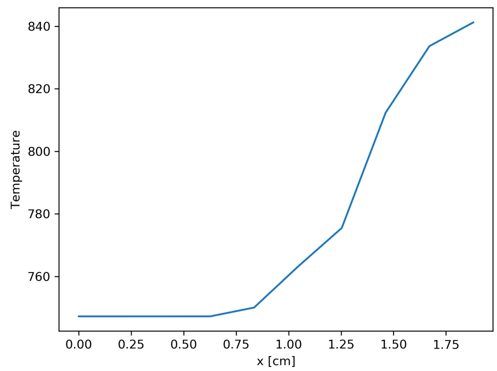


Figure 20: Across coolant, moderator, and fuel: (0,0,400)-(1.88,0,400).

References

[1] Nam-Il Tak. DEVELOPMENT OF A CORE THERMO-FLUID ANALYSIS CODE FOR PRISMATIC GAS COOLED REACTORS. *NUCLEAR ENGINEERING AND TECHNOLOGY*, page 14, 2014.

[2] Nam-il Tak, Min-Hwan Kim, and Won Jae Lee. Numerical investigation of a heat transfer within the prismatic fuel assembly of a very high temperature reactor. *Annals of Nuclear Energy*, 35(10):1892–1899, October 2008.