Steady-State Thermal-Hydraulic Analysis of a Multi-Channel PWR Core*

Jeongjin Han
School of Computing & Nuclear and Quantum Engineering
Korea Advanced Institute of Science and Technology (KAIST)
hjj22@kaist.ac.kr

Contents

1 Solving PDE with Finite Difference Method **2**
1.1 Governing equation and finite-volume balance . . . 2
1.2 Mesh, faces, and special interface treatment . . . 2
1.3 Face conductances G_i (exact code formulas) . . . 3
1.4 Face heat flux and wall closure (outward-positive) . . . 4
1.5 Convergence criteria (exactly as in code) . . . 4

2 Second-order convergence of the numerical method **6**
2.1 Theoretical verification . . . 6
2.2 Numerical verification . . . 6

3 Governing equations for flow channel analysis **7**
3.1 Energy balance and bulk temperature update . . . 7
3.2 Convective heat transfer and wall temperature . . . 7
3.3 Pressure-drop model and diagnostics . . . 7
3.4 Inner secant iteration on G . . . 8
3.5 Energy consistency check . . . 8

4 Constitutive relations (HTC, friction factor, and CHF) **9**
4.1 Heat transfer coefficient (HTC) . . . 9
4.2 Friction factor and pressure-drop model . . . 9
4.3 CHF, DNBR, and use of subcooled enthalpy/quality . . . 10

5 Calculation method for channel mass flux distribution **11**

6 Thermal margin of the reactor core **13**

7 Result **14**

8 Improvement **18**
8.1 From Prob5's TA comments . . . 18

*Submitted for NQE625: Computational Analysis in Nuclear Systems, KAIST, Fall 2025.

8.2 From Prob6's TA comments 19
8.3 Power normalization across meshes 20

1 Solving PDE with Finite Difference Method

1.1 Governing equation and finite-volume balance

Steady, axisymmetric radial conduction with an axial source is considered:

$$-\nabla\cdot\mathbf{q} + q'''(z) = 0, \qquad \mathbf{q} = -k(T)\,\nabla T \implies \nabla\cdot\big(k(T)\nabla T\big) + q'''(z) = 0. \tag{1}$$

Integrate over a control volume V_i and apply the divergence theorem:

$$\int_{\mathcal{V}_i} \nabla\cdot\big(k\nabla T\big)\,dV + \int_{\mathcal{V}_i} q'''(z)\,dV = \int_{\partial\mathcal{V}_i} k\nabla T\cdot\hat{\mathbf{n}}\,dA + q'''(z)\,V_i \;=\; 0, \tag{2}$$

$$\Rightarrow\; -\widetilde{G}_{L,i}\,(T_i - T_{i-1}) + \widetilde{G}_{R,i}\,(T_{i+1} - T_i) + q'''(z)\,V_i = 0. \tag{3}$$

Dividing by the axial thickness Δz (unit axial length in code) gives

$$-G_{L,i}\,(T_i - T_{i-1}) + G_{R,i}\,(T_{i+1} - T_i) + q'''(z)\,A_i = 0, \tag{4}$$

which is the interior stencil used to assemble the tridiagonal system.

Power scaling (exactly as in code). At each axial plane, the volumetric source is scaled by the local channel factor:

$$q'''_{\text{ch}}(z) \;=\; q'''(z)\,F, \qquad \texttt{q\_vol\_ch = q\_vol * F;} \tag{5}$$

and the right-hand side uses $q'''_{\text{ch}}(z)$.

1.2 Mesh, faces, and special interface treatment

Radial cell boundaries are denoted r_i ($i = 1, \ldots, R$) from the centerline to the wall, and axial nodes are z_k ($k = 1, \ldots, Z$). In the implementation, these grids are built as

$$r = \big[\,(0{:}N_f)\,\Delta r_f,\; (r_f + th_g) + (1{:}N_c)\,\Delta r_c\,\big], \tag{6}$$

$$z = \text{linspace}\big(0,\, H_f,\, N_z{+}1\big). \tag{7}$$

Thus the fuel region uses uniform spacing Δr_f up to r_f, the cladding region uses Δr_c, and a finite gap th_g separates fuel and cladding nodes.

Faces are located at midpoints, $r_{i+\frac{1}{2}} = \frac{1}{2}(r_i + r_{i+1})$, except for the fuel–gap interface face which is placed exactly at (8).

$$r_{N_f+1+\frac{1}{2}} = r_f + th_g, \tag{8}$$

to match the geometry used by the flow/BC closures. Because of this explicit placement, the interface control volume on the fuel side uses a one-cell source area correction:

$$A^{(\text{fuel part})}_{N_f+1} = \pi\left[r_f^2 - \left(r_f - \tfrac{\Delta r_f}{2}\right)^2\right], \qquad \text{(implemented as } \texttt{Ai(Nf+1)}\text{)}. \tag{9}$$

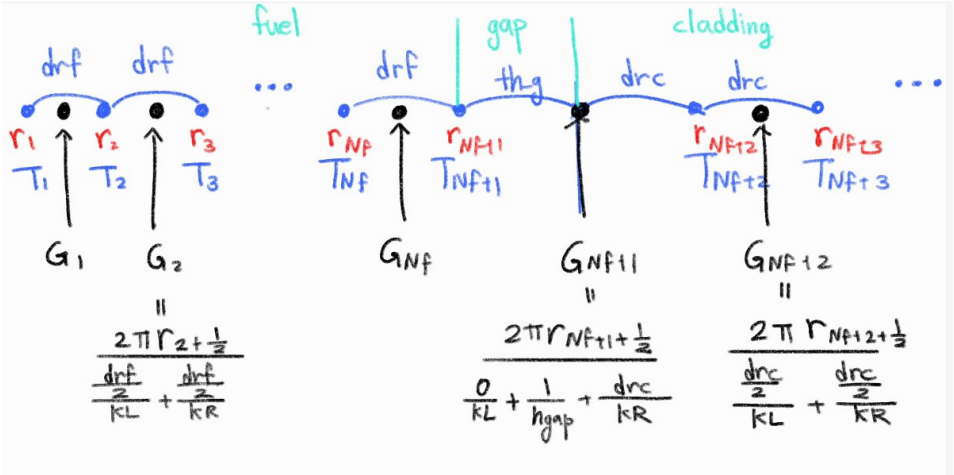


Figure 1: Radial mesh and face placement used in the FV assembly: fuel (left), explicit interface face at $r_f + th_g$ with film resistance $1/h_g$ (middle), and cladding (right). Under each face, the exact code formulas for G_i are shown.

The sketch (fuel / interface / cladding) corresponds one-to-one to Eqs. (10)–(12): - fuel faces use two half-thicknesses $(\mathrm{d}r_f/2,\ \mathrm{d}r_f/2)$ and $k_{\mathrm{UO_2}}(T)$; - the interface face uses $r_{N_f+1+\frac{1}{2}} = r_f + th_g$ and a series addition $0 + 1/h_g + dR_{\mathrm{gap}}/k_{\mathrm{Zircaloy}}$; - cladding faces use two $\mathrm{d}r_c/2$ half-paths and $k_{\mathrm{Zircaloy}}(T)$. Finally, each interior row uses its adjacent conductances ($G_L = G_i$, $G_R = G_{i+1}$) together with the source built from $q'''_{\mathrm{ch}} = q'''\,F$.

1.3 Face conductances G_i (exact code formulas)

The conductive face coefficient (W/K per unit axial length) is the harmonic sum of left/right half-paths, possibly plus a special interfacial resistance:

$$G_i \;=\; \frac{2\pi\, r_{i+\frac{1}{2}}}{\dfrac{d_L}{k_L} \;+\; R_{\mathrm{spec}} \;+\; \dfrac{d_R}{k_R}}.$$

With temperature-dependent conductivities sampled at nodes ($k_L = k(T_i)$, $k_R = k(T_{i+1})$), the implementation uses the following three cases:

Fuel region ($1 \le i \le N_f$)**:**

$$G_i \;=\; \frac{2\pi\, r_{i+\frac{1}{2}}}{\dfrac{\mathrm{d}r_f}{2}\dfrac{1}{k_{\mathrm{UO_2}}(T_i)} + \dfrac{\mathrm{d}r_f}{2}\dfrac{1}{k_{\mathrm{UO_2}}(T_{i+1})}}. \tag{10}$$

Interface face ($i = N_f + 1$)**:**

$$G_{N_f+1} \;=\; \frac{2\pi\, r_{N_f+1+\frac{1}{2}}}{\underbrace{0}_{\text{left fuel term set to 0}} + \dfrac{1}{h_g} + \dfrac{d_R}{k_{\mathrm{Zircaloy}}(T_{N_f+2})}}, \qquad d_R = dR_{\mathrm{gap}}\ (\approx \mathrm{d}r_c/2). \tag{11}$$

This matches the implementation, where `Rspec=1/h_g`, $d_L = 0$, $d_R = dR\_\mathrm{gap}$.

Cladding region ($N_f + 2 \le i \le N_f + 1 + N_c$)**:**

$$G_i \;=\; \frac{2\pi\, r_{i+\frac{1}{2}}}{\dfrac{\mathrm{d}r_c}{2}\dfrac{1}{k_{\mathrm{Zircaloy}}(T_i)} + \dfrac{\mathrm{d}r_c}{2}\dfrac{1}{k_{\mathrm{Zircaloy}}(T_{i+1})}}. \tag{12}$$

Boundary conditions (as implemented in code). Two different boundary conditions are used depending on the physical location and the function being called.

- **Fuel centerline (Neumann BC).** Symmetry at the centerline implies $\partial T/\partial r = 0$, giving a homogeneous Neumann condition. In the discrete system this becomes

$$G_1\,T_1 - G_1\,T_2 = Q_1, \qquad Q_1 = q'''_{\text{ch}}(z)\,A_1, \tag{13}$$

  which in the code corresponds to:

```
GR = Gface(1,k);
A(1,1) = GR;
A(1,2) = -GR;
rhs(1) = q_vol(k) * Ai(1);
```

- **Cladding outer surface.** The treatment of the wall boundary differs between `cal_Qnode_avg` and `cal_Tfield`:
  - *Mixed (Robin) BC — in `cal_Qnode_avg`:* The wall heat flux equals the convective flux to the coolant, $-k\partial_r T = h_w(T_R - T_b)$, producing

$$-G_{R-1}\,T_{R-1} + (G_{R-1} + G_w)\,T_R = G_w\,T_b, \qquad G_w = 2\pi r_R h_w. \tag{14}$$

    Implemented as:

```
GL = Gface(R-1,k);
Gw = Gface(i_wall,k);
A(R,R-1) = -GL;
A(R,R)   = GL + Gw;
rhs(R)   = Gw * Tb(k);
```

  - *Dirichlet BC — in `cal_Tfield`:* During the coupled TH solve, the wall temperature is prescribed by the flow solution, $T_R = T_w(z_k)$:

```
A(R,R) = 1;
rhs(R) = Tw(k);
```

These two BCs correspond respectively to the stand-alone conduction solve for averaged fields (`cal_Qnode_avg`) and to the channel-wise coupled iteration (`cal_Tfield`). Both share the same Neumann symmetry at the fuel center and the same conductance stencil for interior nodes.

1.4 Face heat flux and wall closure (outward-positive)

With the converged T and G_i, the interior face flux density is

$$q''_{i+\frac{1}{2}} \;=\; -\frac{G_i}{2\pi r_{i+\frac{1}{2}}}\,(T_{i+1} - T_i), \qquad i = 1, \ldots, R-1, \tag{15}$$

and at the wall face you use the convective definition from the flow solver:

$$q''_R \;=\; h(z_k)\,\big(T_R - T_b(z_k)\big), \tag{16}$$

which corresponds to the following assembly in the code: `qface(k,i_wall) = HTCs(k) * (T(R,k) - Tb(k));`. The node heat-flux array `Qnode` is then built by simple averaging of adjacent face fluxes, with the wall column taken from the wall face.

Note on property update and under-relaxation. At each nonlinear iteration, you recompute $k_{\text{UO}_2}(T)$ and $k_{\text{Zircaloy}}(T)$ at nodes, rebuild all G_i, solve the tridiagonal system for T, then under-relax: $T \leftarrow \omega T_{\text{new}} + (1-\omega)T_{\text{old}}$.

1.5 Convergence criteria (exactly as in code)

Let T and q''_{face} be the current fields and $(\cdot)_{\text{old}}$ the previous. Denoting axial mid-face averages by

$$q''_{\text{face}}(z_{k+\frac{1}{2}}) = \tfrac{1}{2}\big(q''_{\text{face}}(z_k) + q''_{\text{face}}(z_{k+1})\big), \quad q'''_{\text{ch}}(z_{k+\frac{1}{2}}) = \tfrac{1}{2}\big(q'''_{\text{ch}}(z_k) + q'''_{\text{ch}}(z_{k+1})\big),$$

the three stopping checks are

$$r_T = \frac{\|T - T_{\text{old}}\|_\infty}{\max(1, \|T_{\text{old}}\|_\infty)} \; < \; \texttt{tolT}, \tag{17}$$

$$r_Q = \frac{\|q''_{\text{face}} - q''_{\text{face,old}}\|_\infty}{\max(1, \|q''_{\text{face,old}}\|_\infty)} \; < \; \texttt{tolQ}, \tag{18}$$

$$E_{\text{rel}} = \frac{\max\left(|P_{\text{int}} - P_{\text{gen}}|,\; |P_{\text{wall}} - P_{\text{gen}}|\right)}{\max(P_{\text{gen}}, 10^{-12})} \; < \; \texttt{tolE}, \tag{19}$$

with

$$P_{\text{gen}} = \sum_k q'''_{\text{ch}}(z_{k+\frac{1}{2}})\, A_f\, \Delta z_k, \quad P_{\text{int}} = \sum_k q''_{i_{\text{interface}}}(z_{k+\frac{1}{2}}) \left(2\pi r_{i_{\text{interface}}}\right) \Delta z_k, \quad P_{\text{wall}} = \sum_k q''_R(z_{k+\frac{1}{2}}) \left(2\pi r_R\right) \Delta z_k.$$

In `cal_Qnode_avg`: `tolT=tolQ=tolE=`10^{-10}. In `cal_Tfield` the same structure is used with the channel-scaled source from (5). Here, $A_f = \pi r_f^2$ denotes the fuel cross-sectional area at the interface radius.

2 Second-order convergence of the numerical method

The finite-difference formulation derived in Section 1 is expected to achieve *second-order spatial accuracy*. This section verifies that property both analytically and numerically.

2.1 Theoretical verification

The discrete energy balance used throughout this work is

$$-G_{L,i}\,(T_i - T_{i-1}) + G_{R,i}\,(T_{i+1} - T_i) + q'''(z)\,A_i = 0. \tag{20}$$

Here the two flux terms are differences across the left/right faces and are built symmetrically about node i:

$$\text{left flux} \propto (T_i - T_{i-1}), \qquad \text{right flux} \propto (T_{i+1} - T_i),$$

with face conductances $G_{L,i}, G_{R,i}$ constructed from **symmetric half-distances** (fuel: $\Delta r_f/2$, cladding: $\Delta r_c/2$) and harmonic averaging of $k(T)$. Consequently, Eq. (20) is a **central (symmetric) three-point stencil** for the divergence of the radial heat flux. Central stencils cancel the odd-order Taylor terms, leaving a leading truncation error of $\mathcal{O}((\Delta r)^2)$.

The source term $q'''(z)\,A_i$ is a **cell average** (mid-point rule) and is likewise second-order in the axial direction. Therefore, the discrete formulation (20) is **second-order accurate** in space:

$$\text{error} = \mathcal{O}\big((\Delta r)^2 + (\Delta z)^2\big).$$

2.2 Numerical verification

To confirm the second-order accuracy, the temperature field was computed for progressively refined meshes while keeping the same physical parameters as in Section 1. The reference solution $T_{\text{ref}}(r, z)$ corresponds to the finest grid, obtained with

$$N_c = 7, \qquad N_f = 30, \qquad N_z = 61 \quad \Rightarrow \quad N_r = N_c + N_f = 37.$$

The coarser test meshes used for comparison were

$$(N_c, N_f, N_z) \in \{(3, 10, 21),\ (4, 15, 31),\ (5, 20, 41),\ (6, 25, 51)\}.$$

For each mesh, the global relative error is defined as

$$\varepsilon(N_r, N_z) = \frac{\| T_{N_r,N_z} - T_{\text{ref}} \|_2}{\| T_{\text{ref}} \|_2}.$$

As seen in Fig. 2, the fitted slope closely matches the theoretical value of -2, confirming that the implemented finite-volume discretization achieves second-order convergence in both the fuel and cladding regions.

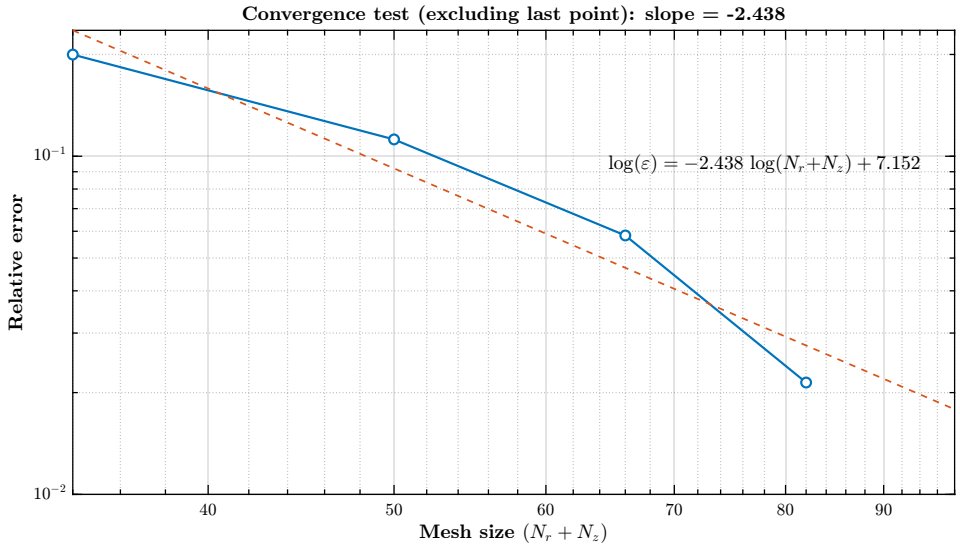


Figure 2: Second-order convergence verification. The log–log plot of mesh size $(N_r + N_z)$ versus relative error shows a slope of approximately -2, confirming second-order spatial accuracy.

3 Governing equations for flow channel analysis

Each representative subchannel is analyzed independently through the function `cal_each_fuel_pin`, which internally solves for the local mass flux G_i that satisfies the pressure-drop balance against the target $\Delta P_{\mathrm{tot,avg}}$ from the outer iteration. The inner solver determines the mass flux by repeatedly calling `cal_dPtot_and_Tb`, which computes the thermohydraulic field and the total pressure drop for a given trial G_i.

3.1 Energy balance and bulk temperature update

For each axial node k along the fuel rod, the bulk temperature $T_{b,k}$ is obtained using an enthalpy-based secant update to satisfy

$$h_{\text{subcooled}}(T_{b,k}) = h_{\text{subcooled}}(T_{b,k-1}) + \frac{q''_{k-1}\,\Delta z_{k-1}\,(2\pi r_o)}{\dot{m}}, \qquad \dot{m} = A_{\text{ch}}\,G, \tag{21}$$

where r_o is the cladding outer radius, A_{ch} is the channel flow area, and $h_{\text{subcooled}}(T)$ is the liquid enthalpy function. The bulk temperature integration proceeds axially from the inlet temperature $T_{b,\text{in}}$ until the outlet node $T_{b,\text{out}}$.

3.2 Convective heat transfer and wall temperature

Once the bulk temperature distribution is obtained, the local thermophysical properties—-viscosity $\mu(T_b)$, density $\rho(T_b)$, conductivity $k(T_b)$, and specific heat $c_p(T_b)$-— are evaluated at each node.

$$\mathrm{Re} = \frac{GD_e}{\mu}, \qquad \mathrm{Pr} = \frac{c_p\mu}{k},$$

and the convective heat transfer coefficient is first computed using the Dittus–Boelter correlation:

$$h_{\mathrm{DB}} = 0.023\,\mathrm{Re}^{0.8}\mathrm{Pr}^{0.4}\,\frac{k}{D_e}.$$

The preliminary wall temperature is therefore

$$T_{w,k}^{(\mathrm{DB})} = T_{b,k} + \frac{q''_k}{h_{\mathrm{DB},k}}. \tag{22}$$

When the wall temperature exceeds the saturation temperature T_{sat} (at $P = 15$ MPa), a secondary iteration is applied to include subcooled boiling effects using the Jens–Lottes form:

$$r(T_w) = h_{\mathrm{DB}}(T_w - T_b) + C_{\mathrm{nb}}\max(T_w - T_{\mathrm{sat}}, 0)^4 - q'' = 0, \tag{23}$$

which is solved by a secant method for each affected node. The updated wall temperature and the corresponding effective HTC are then

$$h_k = \frac{q''_k}{T_{w,k} - T_{b,k}}.$$

3.3 Pressure-drop model and diagnostics

The total pressure drop in the channel is computed as the sum of contraction, frictional, expansion, and gravitational components:

$$\Delta P_{\mathrm{tot}} = \Delta P_{\mathrm{contr}} + \Delta P_{\mathrm{fric}} + \Delta P_{\mathrm{exp}} + \Delta P_g. \tag{24}$$

Contraction and expansion losses. Based on the inlet and outlet hydraulic diameters, the local loss coefficients are

$$K_{\mathrm{SC}} = 1 - \left(\frac{d_{\text{fuel}}}{D_{\text{contr}}}\right)^2, \quad K_{\mathrm{SE}} = 0.42\left[1 - \left(\frac{d_{\text{fuel}}}{D_{\text{enter}}}\right)^2\right]^2,$$

and the associated pressure losses are evaluated using $\Delta P = K\,G^2/(2\rho)$.

Frictional pressure loss. The wall roughness $E = 3.6 \times 10^{-5}$ m and the Colebrook correlation determine the local friction factor f from the face-averaged Reynolds number $\mathrm{Re}_{\mathrm{face}} = G D_e / \bar{\mu}$. The incremental frictional pressure drop per cell is

$$\Delta P_{\mathrm{fric},k} = f_k \, \frac{\Delta z_k}{D_e} \, \frac{G^2}{2 \rho_{\mathrm{face},k}},$$

and the total ΔP_{fric} is the sum over all segments.

Gravitational head. The gravitational contribution is given by

$$\Delta P_g = g \sum_k \rho_{\mathrm{face},k} \, \Delta z_k.$$

Effective loss coefficient. For diagnostic purposes, an effective coefficient

$$K_{\mathrm{eff}} = \frac{\Delta P_{\mathrm{tot}} - \Delta P_g}{G^2} \tag{25}$$

is computed, but it is not used for updating G itself.

3.4 Inner secant iteration on G

The function `cal_each_fuel_pin` performs a secant iteration on

$$\phi(G) = \Delta P_{\mathrm{tot}}(G) - \Delta P_{\mathrm{tot,avg}},$$

starting from $G_0 = G_{\mathrm{init}}$ and $G_1 = 1.1\, G_0$. At each iteration, $\Delta P_{\mathrm{tot}}(G)$ is recomputed through `cal_dPtot_and_Tb`, and convergence is declared when either $|\phi(G)| < 10^{-3}$ Pa or $|G_{n+1} - G_n| < 10^{-6}$. If the slope $|\phi_1 - \phi_0| < 10^{-12}$, the iteration is safely terminated and the latest G is retained. This inner solver is nested inside the outer loop on $\Delta P_{\mathrm{tot,avg}}$ described in Sec. 5.

3.5 Energy consistency check

Finally, a power balance is verified between the total surface heat flux and the enthalpy rise of the coolant:

$$P_{\mathrm{flux}} = \sum_{k=1}^{Z-1} q''_k (2\pi r_o) \, \Delta z_k, \qquad P_{\mathrm{bulk}} = \dot{m} \, [\, h_{\mathrm{subcooled}}(T_{b,\mathrm{out}}) - h_{\mathrm{subcooled}}(T_{b,\mathrm{in}}) \,],$$

and their relative error, $\varepsilon_P = (P_{\mathrm{bulk}} - P_{\mathrm{flux}})/P_{\mathrm{flux}}$, is printed for debugging to confirm the energy conservation within each channel.

4 Constitutive relations (HTC, friction factor, and CHF)

4.1 Heat transfer coefficient (HTC)

Step 1: Bulk temperature via a secant solve. For each channel and axial segment, we first advance the bulk liquid temperature $T_b(z)$ by enforcing the single–phase energy balance

$$\dot{m}\,\Delta h \;=\; q''(z)\,P_w\,\Delta z, \qquad \dot{m} = G\,A_{\mathrm{ch}}, \tag{26}$$

where $q''(z)$ is taken on the wall side of $\mathbf{Q}_{\text{node}}$ (i.e. `Qnode_ch(:,R)`), and $P_w = \pi d_o$ (wetted perimeter based on equivalent outer diameter). With the liquid enthalpy function $h_\ell(T)$, we define $h_{\mathrm{targ}} = h_\ell\big(T_{b,k-1}\big) + q''_{k-1}\,P_w\,\Delta z/\dot{m}$ and solve $h_\ell(T_{b,k}) - h_{\mathrm{targ}} = 0$ by a two–point secant iteration (two temperature guesses per node) exactly as implemented in `cal_dPtot_and_Tb`.

Step 2: Dittus–Boelter baseline and wall temperature. After $T_b(z)$ is known, thermophysical properties are evaluated at $T_b(z)$, and the Reynolds/Prandtl numbers are formed as

$$\mathrm{Re}(z) = \frac{G\,D_e}{\mu\big(T_b(z)\big)}, \qquad \mathrm{Pr}(z) = \frac{c_p\big(T_b(z)\big)\,\mu\big(T_b(z)\big)}{k\big(T_b(z)\big)}.$$

The single–phase HTC then follows from the Dittus–Boelter correlation

$$h_{\mathrm{DB}}(z) = 0.023\,\frac{k(T_b)}{D_e}\,\mathrm{Re}(z)^{0.8}\,\mathrm{Pr}(z)^{0.4}, \tag{27}$$

and the corresponding preliminary wall temperature is

$$T_{w,\mathrm{DB}}(z) = \frac{q''(z)}{h_{\mathrm{DB}}(z)} + T_b(z). \tag{28}$$

Step 3: Jens–Lottes switch and effective HTC. If nucleate boiling is indicated by $T_{w,\mathrm{DB}}(z) \geq T_{\mathrm{sat}}(P)$, we solve for the actual wall temperature $T_w(z)$ using the nonlinear balance

$$r(T_w) \;=\; h_{\mathrm{DB}}\big(T_w - T_b\big) + C_{\mathrm{nb}}\max\big(T_w - T_{\mathrm{sat}}, 0\big)^4 - q'' \;=\; 0, \tag{29}$$

with a secant update initialized at $T_{w,\mathrm{DB}}$ (as in the code). Here T_{sat} is evaluated at $P = 15$ MPa and the implemented coefficient is $C_{\mathrm{nb}} = \exp\big(4P[\mathrm{MPa}]/6.2\big)/25^4 \times 10^6$. Finally, we define the *effective* local HTC actually used in the field update as

$$h(z) = \frac{q''(z)}{T_w(z) - T_b(z)}. \tag{30}$$

4.2 Friction factor and pressure-drop model

The total pressure drop used in `cal_dPtot_and_Tb` is

$$\Delta P_{\mathrm{tot}} = \Delta P_{\mathrm{contr}} + \Delta P_{\mathrm{fric}} + \Delta P_{\mathrm{exp}} + \Delta P_g, \tag{31}$$

with contraction/expansion losses $\Delta P_{\mathrm{contr}} = K_{\mathrm{SC}}\,G^2/(2\rho_{\mathrm{in}})$, $\Delta P_{\mathrm{exp}} = K_{\mathrm{SE}}\,G^2/(2\rho_{\mathrm{out}})$ and a gravity head $\Delta P_g = g\sum \rho_{\text{face}}\,\Delta z$ using face–averaged densities. The distributed friction term is integrated axially as

$$\Delta P_{\mathrm{fric}} = \sum_{\text{faces}} \left(\frac{f}{D_e}\,\Delta z\right) \frac{G^2}{2\,\rho_{\text{face}}}, \tag{32}$$

where the Darcy friction factor follows the composite law used in the code:

$$f = \begin{cases} \dfrac{64}{\mathrm{Re}}, & \mathrm{Re} < 2100, \\[2ex] \text{Colebrook: } \dfrac{1}{\sqrt{f}} = -2\log_{10}\!\left(\dfrac{\varepsilon/D_e}{3.7} + \dfrac{2.51}{\mathrm{Re}\sqrt{f}}\right), & \mathrm{Re} \geq 2100, \end{cases}$$

solved by a scalar root finder initialized with the Haaland estimate.

4.3 CHF, DNBR, and use of subcooled enthalpy/quality

The subcooled (compressed–liquid) enthalpy used to advance T_b and to parameterize CHF is

$$h_{\mathrm{sub}}(T_b) = h_f\big(T_b[^\circ\mathrm{C}]\big) \times 10^3 + v_f\big(T_b[^\circ\mathrm{C}]\big)\,\big(15\ \mathrm{MPa} - P_{\mathrm{sat}}(T_b)\big), \tag{33}$$

i.e. the standard approximation $h \approx h_f(T) + v_f(T)\,[P - P_{\mathrm{sat}}(T)]$ at $P = 15$ MPa.

Quality mapping used for CHF. Following the code, the local equilibrium quality used to query the CHF LUT is not obtained by a two–phase energy balance along the channel; instead, it is a *pointwise mapping* from the subcooled enthalpy to the equilibrium two–phase scale at the same pressure $P = 15$ MPa:

$$T_{\mathrm{sat},15} \;=\; T_{\mathrm{sat}}(15\ \mathrm{MPa}), \tag{34}$$

$$h_{f,15} \;=\; h_f\big(T_{\mathrm{sat},15}\big) \times 10^3, \qquad h_{g,15} \;=\; h_g\big(T_{\mathrm{sat},15}\big) \times 10^3, \tag{35}$$

$$x(z) \;=\; \frac{h_{\mathrm{sub}}(z) - h_{f,15}}{h_{g,15} - h_{f,15}}. \tag{36}$$

This corresponds exactly to the implemented routine `quality(h_sub)`, which evaluates $T_{\mathrm{sat},15}$, forms $\{h_{f,15}, h_{g,15}\}$, and computes x by (36).

CHF lookup and DNBR. With $x(z)$ from (36), we query the LUT $\mathrm{CHF}_{\mathrm{LUT}} = \texttt{CHFLUT2006}(15, G, x)$. The code then applies the diameter correction and unit conversion

$$\mathrm{CHF}(z) = \mathrm{CHF}_{\mathrm{LUT}}(z) \left(\frac{D_e}{0.008}\right)^{-1/2} \times 10^3 \quad [\mathrm{W/m^2}], \tag{37}$$

and forms

$$\mathrm{DNBR}(z) = \frac{\mathrm{CHF}(z)}{q''(z)}, \qquad \mathrm{MDNBR} = \min_z \mathrm{DNBR}(z). \tag{38}$$

5 Calculation method for channel mass flux distribution

We distribute the channel-wise mass fluxes $\{G_i\}_{i=1}^{M}$ by enforcing a single, common total pressure drop $\Delta P_{\mathrm{tot,avg}}$ across all representative channels while satisfying global mass conservation. Thanks to core symmetry, only $M = 11$ channel groups are considered:

$$(5, E), (5, D), (5, C), (5, B), (5, A), (6, D), (6, C), (6, B), (6, A), (7, C), (7, B),$$

with the number of pins and *normalized* power/peaking factors

$$\mathbf{N}_s = \{236,\ 944,\ 944,\ 960,\ 960,\ 944,\ 1888,\ 1920,\ 1792,\ 960,\ 1920\},$$

$$\mathbf{F}_s = \{0.6246,\ 0.7304,\ 1.1819,\ 1.3147,\ 0.8301,\ 0.9164,\ 1.2103,\ 1.2090,\ 0.5906,\ 1.2617,\ 0.8934\}.$$

Normalization of peaking factors. Since the raw peaking factors do not satisfy $\sum_i N_{s,i} F_{s,i} = N_{\mathrm{rod}}$, we scale them by

$$F_{s,i}^{(\mathrm{norm})} \;=\; F_{s,i}\,\frac{N_{\mathrm{rod}}}{\sum_{j=1}^{M} N_{s,j} F_{s,j}}, \tag{39}$$

which is implemented as:

```
function normalize_Fs = normalize_Fs(Fs, Ns)
    global N_rod verbose
    scale = N_rod / sum(Fs .* Ns);
    if verbose
        fprintf("[DEBUG] [normalize_Fs] scale %.4f \n", scale);
    end
    normalize_Fs = Fs * scale;
end
```

Outer iteration (on $\Delta P_{\mathrm{tot,avg}}$). Let G_{avg} be the target core-average mass flux from the single-channel reference. For a trial $\Delta P_{\mathrm{tot,avg}}$, each channel solver returns $\Delta P_{\mathrm{grav},i}$ and $K_{\mathrm{eff},i}$ (defined by $\Delta P_{\mathrm{tot},i} \approx \Delta P_{\mathrm{grav},i} + K_{\mathrm{eff},i} G_i^2$). The global mass-flow residual is

$$f(\Delta P_{\mathrm{tot,avg}}) = \sum_{i=1}^{M} N_{s,i} \sqrt{\frac{\max(\Delta P_{\mathrm{tot,avg}} - \Delta P_{\mathrm{grav},i}, 0)}{\max(K_{\mathrm{eff},i}, 10^{-12})}} \;-\; \Big(\sum_{i=1}^{M} N_{s,i}\Big) G_{\mathrm{avg}}. \tag{40}$$

If $|f(\Delta P_{\mathrm{tot,avg}})| < 10^{-6}\,(\sum_i N_{s,i} G_{\mathrm{avg}})$ at the beginning of an outer iteration, we stop immediately (no further outer or inner secant steps are performed); otherwise $\Delta P_{\mathrm{tot,avg}}$ is updated by a secant step. [2]

Per-channel solve (inside `cal_each_fuel_pin`). Crucially, we do *not* update G_i from a closed form after the outer step. Instead, for each channel i we solve

$$\phi_i(G) := \Delta P_{\mathrm{tot},i}(G) - \Delta P_{\mathrm{tot,avg}} = 0 \tag{41}$$

by a dedicated secant iteration *inside* `cal_each_fuel_pin`, using two initial guesses $G_0 = G_i$ and $G_1 = 1.1\,G_0$, and stopping when either $|\phi_i(G_1)| < \mathrm{tol}_f$ (with $\mathrm{tol}_f = 10^{-3}$ Pa) or $|G_1 - G_0| < \mathrm{tol}_G$ (with $\mathrm{tol}_G = 10^{-6}$). A small-slope guard is used if $|\,\phi_i(G_1) - \phi_i(G_0)\,| < 10^{-12}$, and G is clamped to $G \geq 10^{-6}$.

[2] In our implementation, once the outer stopping tolerance is satisfied at the beginning of an iteration, we terminate without taking another secant step on $\Delta P_{\mathrm{tot,avg}}$ and without re-invoking the per-channel secant solves for G_i. The latest $\{G_i\}$ obtained in the previous outer step are accepted as the final solution.

Algorithm 1 Coupled mass-flux distribution (outer on $\Delta P_{\mathrm{tot,avg}}$; inner per-channel on G_i)

Given $\{(N_{s,i}, F_{s,i})\}_{i=1}^{M}$, normalize $F_{s,i}$ by (39).
Initialize $\Delta P_{\mathrm{tot,avg}}$, G_i (per channel), and tolerances.
for it $= 1 :$ `itmax` **do**
    Compute residual $f(\Delta P_{\mathrm{tot,avg}})$ by (40).
    if $|f| < 10^{-6} \times (\sum_i N_{s,i} G_{\mathrm{avg}})$ **then**
        break ▷ Stop; do not re-enter secant once within tolerance.
    end if
    for $i = 1 : M$ **do** ▷ Channel analyses
        $Q_{\mathrm{node},i} \leftarrow Q_{\mathrm{node,avg}} \cdot F_{s,i}$
        Solve (41) by secant inside `cal_each_fuel_pin` to obtain G_i
        Extract $\Delta P_{\mathrm{grav},i}$ and $K_{\mathrm{eff},i}$ for diagnostics
    end for
    Update $\Delta P_{\mathrm{tot,avg}}$ by a secant step on $f(\cdot)$
    Clamp $\Delta P_{\mathrm{tot,avg}} \geq \max_i(\Delta P_{\mathrm{grav},i})$
end for

Remarks. (i) The outer loop uses $\Delta P_{\mathrm{tot,avg}}$ as the only global unknown; per-channel mass fluxes G_i are found by the *inner* secant solves and are *not* overwritten by any explicit update formula. (ii) Diagnostic scans of $f(\Delta P)$ are optionally printed to ensure monotonic bracketing and robust secant progress. (iii) All tolerances and guards (`epsK=` 10^{-12}, $G \geq 10^{-6}$, etc.) match the implementation.

6 Thermal margin of the reactor core

This section defines how the thermal margins were evaluated from the detailed subchannel results. Two metrics are used: the *minimum departure from nucleate boiling ratio* (MDNBR) and the *peak cladding temperature* (PCT). To account for design conservatism and possible overpower conditions, the calculated values are normalized by two multiplicative factors:

- E_F : Engineering factor (accounts for model/measurement uncertainties), here $E_F = 1.1$.
- O_F : Overpower factor (accounts for power peaking excursions), here $O_F = 1.1$.

The acceptance reference values are:

- MDNBR criterion: 1.3 (dimensionless).
- PCT criterion: $1204\,^{\circ}\mathrm{C}$; for consistency with the code temperature unit, this is converted to Kelvin as $T_{\mathrm{lim}} = 1204 + 273.15 = 1477.15$ K.

The non-dimensional margins are then defined as

$$\mathrm{MDNBR}_{\mathrm{margin}} \;=\; \frac{\mathrm{MDNBR}_{\mathrm{calc}}}{1.3 \times E_F \times O_F}, \tag{42}$$

$$\mathrm{PCT}_{\mathrm{margin}} \;=\; \frac{T_{\mathrm{lim}}}{\mathrm{PCT}_{\mathrm{calc}} \times E_F \times O_F} \;=\; \frac{1204 + 273.15}{\mathrm{PCT}_{\mathrm{calc}} \times E_F \times O_F}. \tag{43}$$

With the factors used in this work ($E_F = 1.1$, $O_F = 1.1$), the combined conservatism is $E_F \times O_F = 1.21$. A margin greater than unity (> 1.0) in Eqs. (42)–(43) indicates the design meets the corresponding limit with the specified engineering and overpower allowances applied.

7 Result

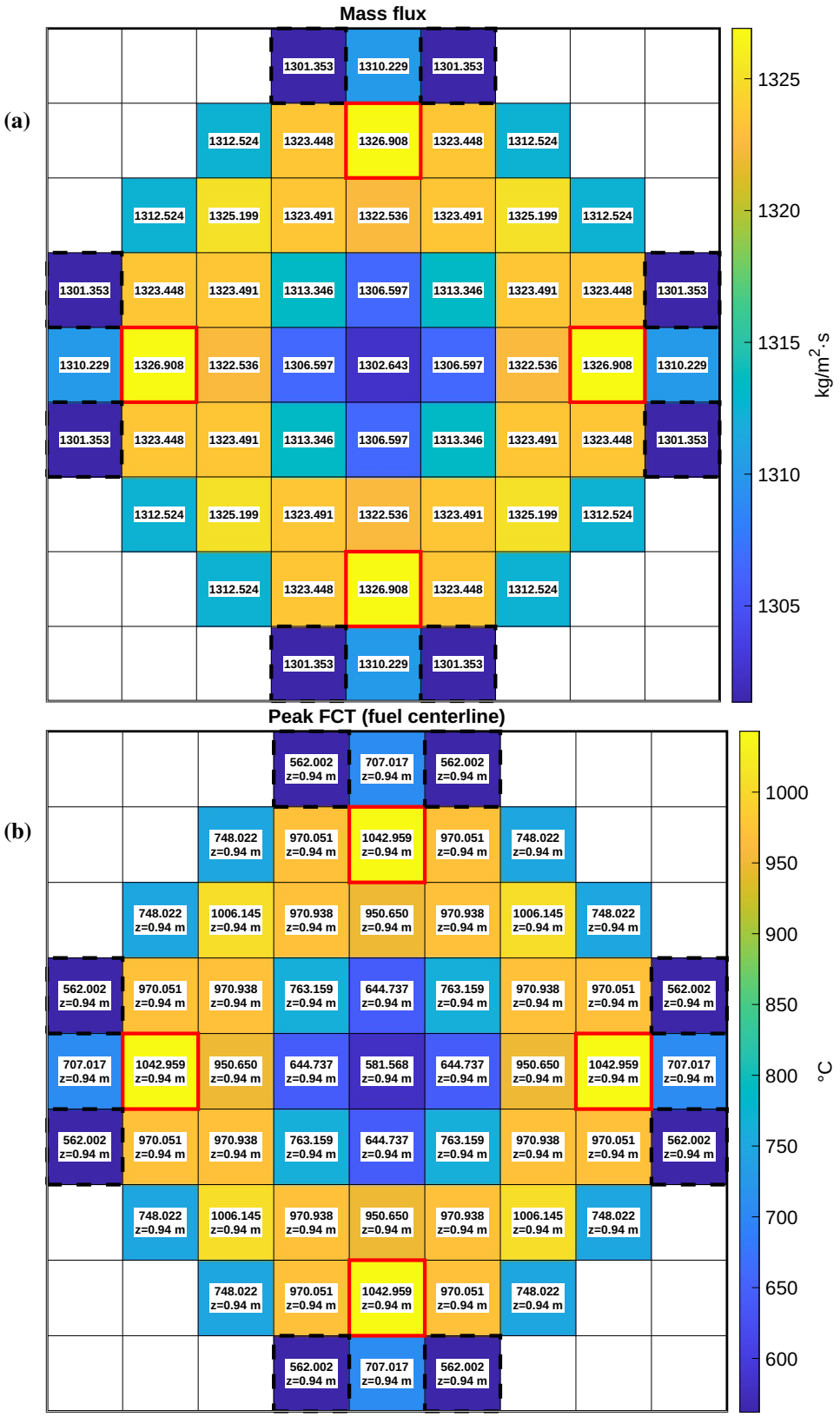


Figure 3: Comparison between (a) the 2D core-wise mass flux distribution and (b) the 2D PFCT map.

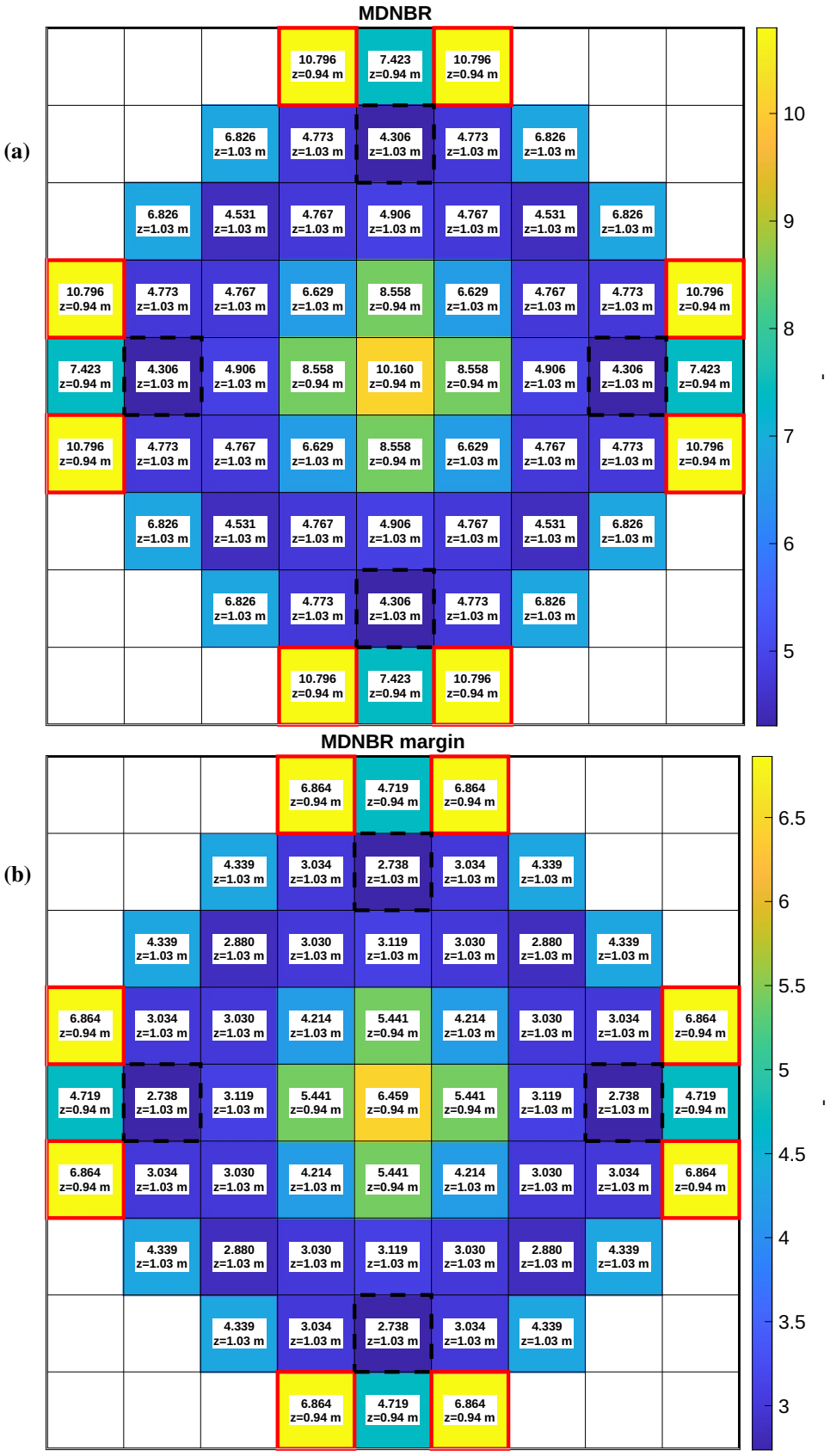


Figure 4: Comparison between (a) the 2D MDNBR distribution and (b) the 2D MDNBR margin distribution.

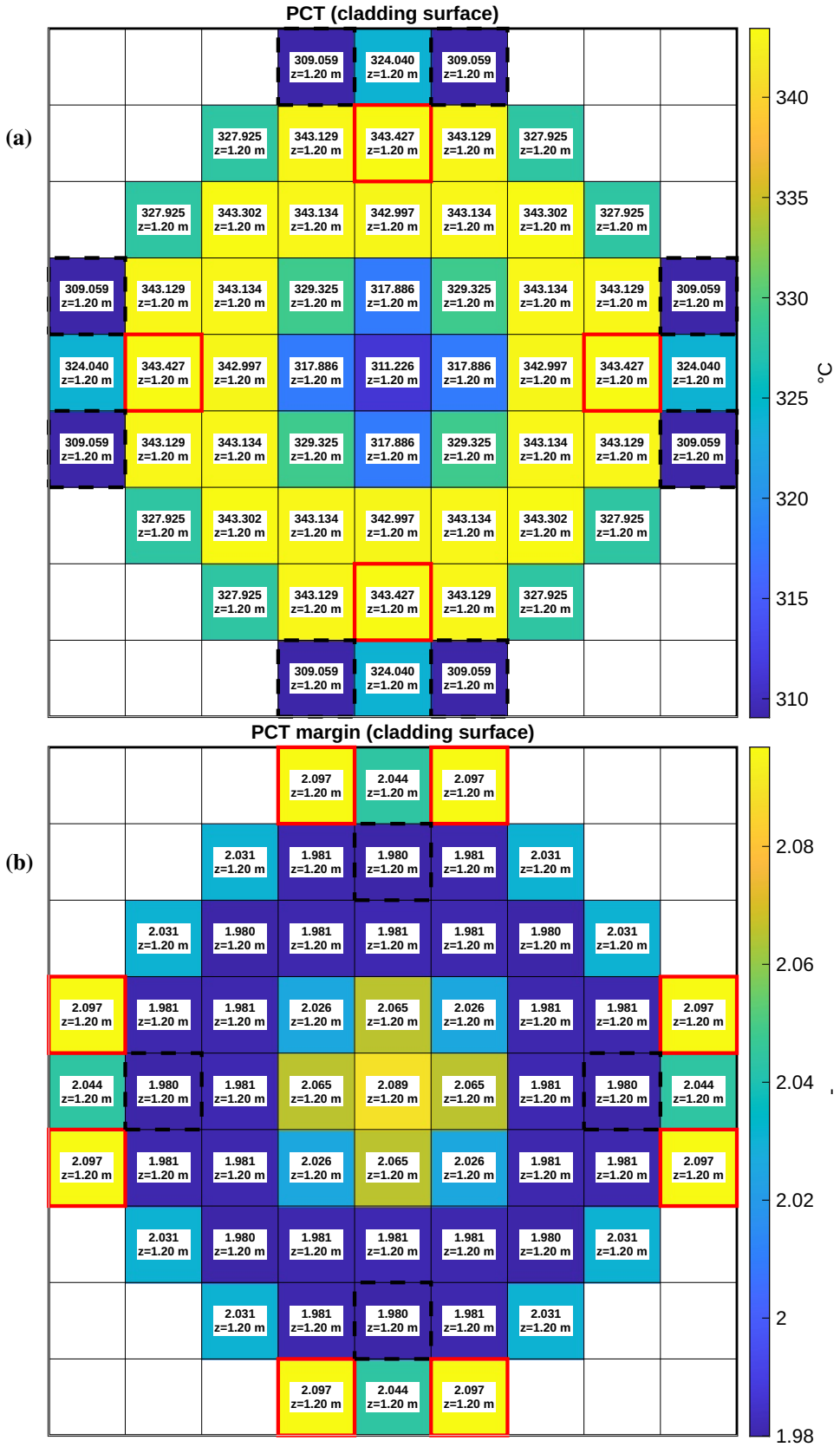


Figure 5: Comparison between (a) the 2D peak cladding temperature (PCT) distribution and (b) the 2D PCT margin distribution.

Interpretation of 2D core maps. The six 2D core maps visualize the axial peak or limiting values of key thermal–hydraulic parameters across representative assemblies:

- **Mass flux (G):** local coolant flow rate per channel, determined by an equal pressure-drop constraint;
- **PFCT (peak fuel centerline temperature):** axial peak fuel centerline temperature;
- **MDNBR:** minimum DNBR within each channel;
- **MDNBR margin:** safety margin relative to the DNBR limit;
- **PCT (peak cladding temperature):** axial peak cladding surface temperature;
- **PCT margin:** temperature margin from the PCT safety limit.

In all figures, colored contours indicate normalized magnitudes across the core. The **red solid outline** marks the assembly with the *highest* value , whereas the **black dashed outline** marks the *lowest* value. This consistent convention allows straightforward comparison of peak regions across different thermal parameters.

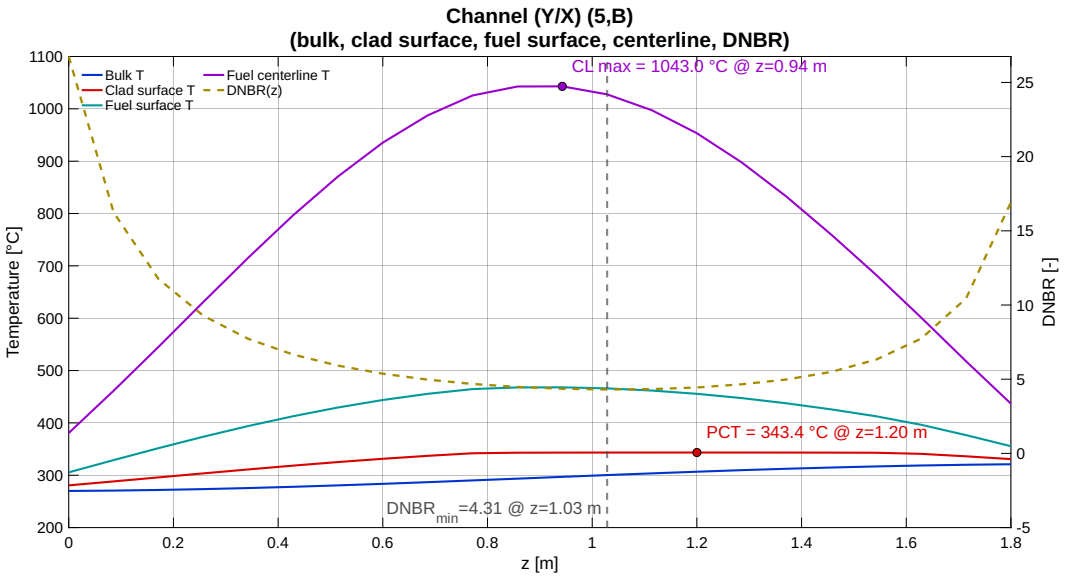


Figure 6: Axial temperature and DNBR profiles along the hot channel.

Figure interpretation. The figure presents the axial variation of four temperature components within the **hottest channel** — identified as position $(Y/X) = (5/B)$, where the local power and thermal limits are most restrictive. This channel exhibits the *highest* mass flux and fuel power (PFCT), and simultaneously the *lowest* safety margins (MDNBR and MDNBR margin). It also corresponds to the peak cladding temperature (PCT) and the minimum PCT margin observed in the 2D core maps.

The plotted quantities are: the bulk coolant temperature $T_{\mathrm{bulk}}(z)$, the cladding surface temperature $T_{\mathrm{clad}}(z)$, the fuel surface temperature $T_{\mathrm{fuel}}(z)$, and the fuel centerline temperature $T_{\mathrm{center}}(z)$, with the dashed curve showing the local $\mathrm{DNBR}(z)$ distribution.

Markers indicate the three limiting positions used for thermal-margin assessment:

- $\mathrm{DNBR}_{\mathrm{min}}$ — the axial location of the minimum departure-from-nucleate-boiling ratio;
- PCT — the peak cladding temperature;
- PFCT (or FCT) — the peak fuel centerline temperature.

These points correspond to the same limiting channel shown in the 2D core maps (MDNBR, MDNBR margin, PCT, PCT margin, and PFCT) and are used to evaluate the overall thermal safety margins of the reactor core.

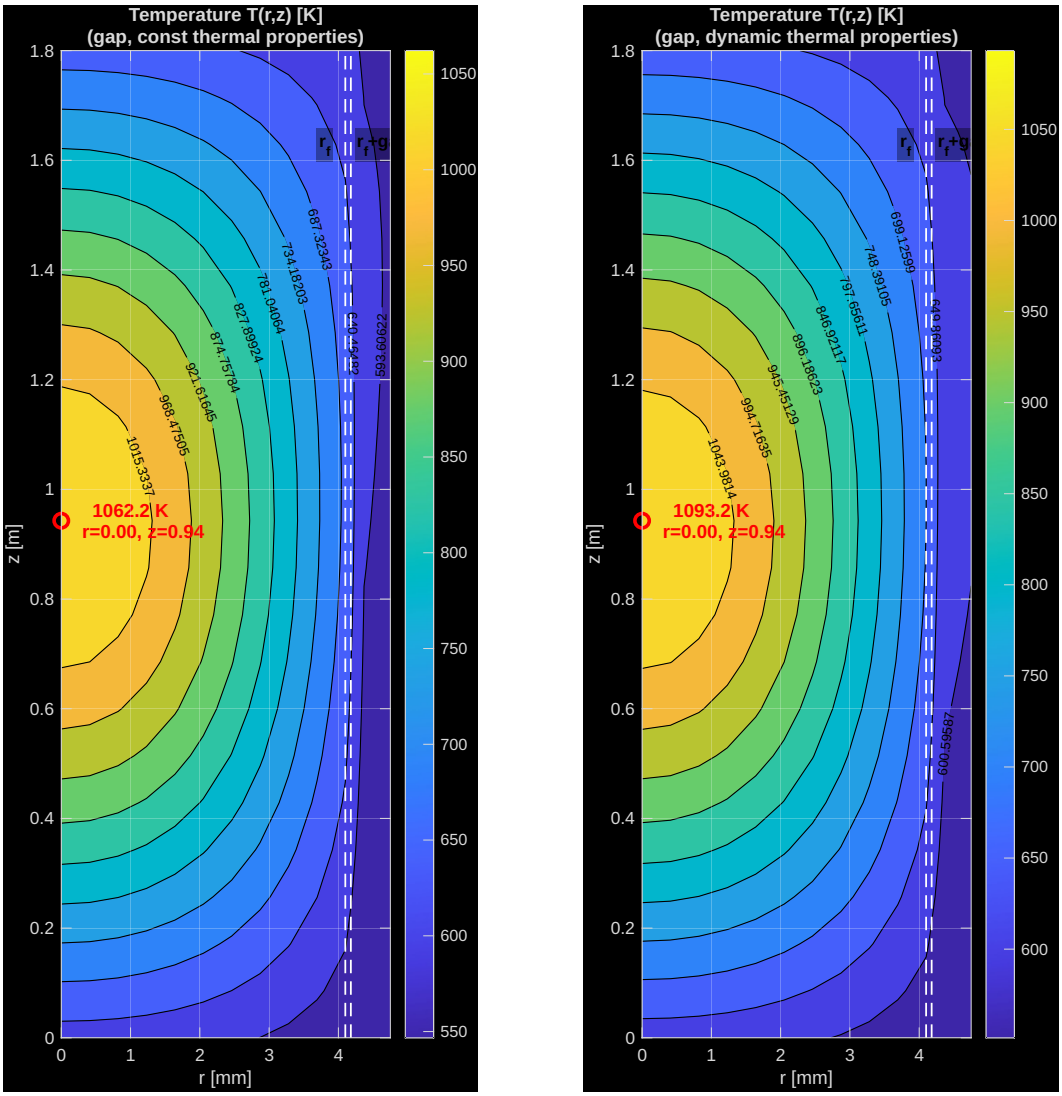
8 Improvement

This section summarizes the changes made in response to the TAs' comments for Prob. 5 and Prob. 6, and quantifies their impact on key thermal metrics.

8.1 From Prob5's TA comments

Issue. Both the constant-property and temperature-dependent (dynamic) models in the previous version predicted an average-channel fuel centerline temperature exceeding 1100 K, whereas the TA indicated that it should be around 1060 K or lower than 1100K.

Fix. The conduction solver (Section 1) was revised to correct the thermal-property handling and boundary conditions, and the updated temperature-dependent model now yields physically consistent results below 1100 K (approximately 1060 K at the peak). The results shown in Fig. 7 correspond to this corrected implementation.



(a) Constant thermal properties

(b) Temperature-dependent thermal properties

Figure 7: Fuel-centerline temperature contours (average channel): constant vs. temperature-dependent properties. The dynamic-property model yields a peak consistent with the TA guidance ($\sim$1060 K and below 1100 K).

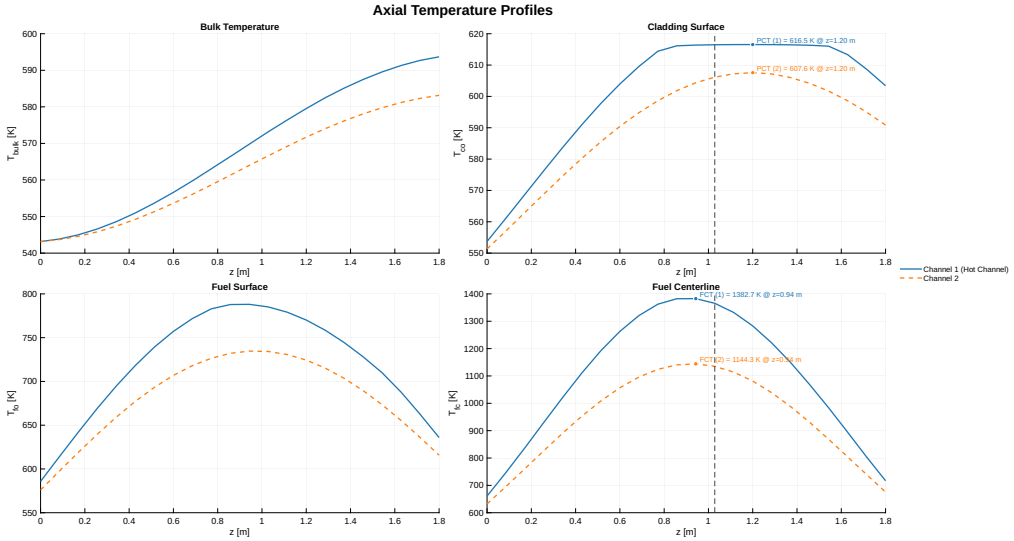
8.2 From Prob6's TA comments

Issues. The fuel-surface temperature appeared too high due to the FV interface placement.

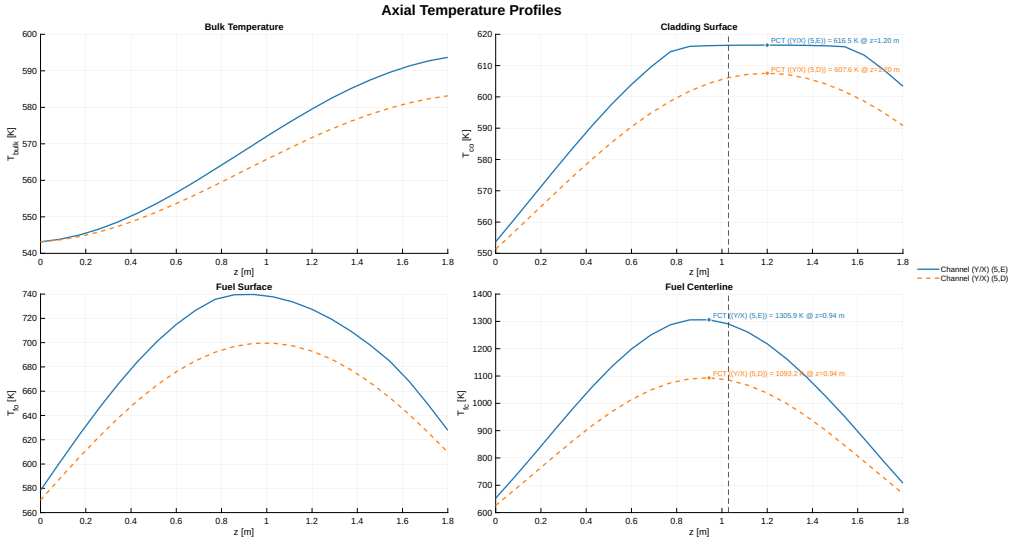
Fixes. The FV interface treatment (Section 1) was refined: the fuel–gap face is now positioned exactly at $r_f + th_g$, and the interface conductance G_{N_f+1} is evaluated with

$$d_L = 0, \qquad R_{\text{spec}} = \frac{1}{h_g}, \qquad d_R = dR_{\text{gap}}.$$

By setting $d_L = 0$, the left (fuel-side) half-cell is not redundantly added in the interfacial resistance, since the adjacent fuel node T_{N_f} already accounts for its own half-distance in the previous flux balance. This avoids double-counting the thermal resistance across the fuel–gap boundary and makes the interface temperature coupling more stable. The fuel-side interface area $A^{(\text{fuel})}_{N_f+1}$ was also corrected to ensure consistency with this definition. As a result, the previously overestimated fuel-surface temperature was lowered to physically consistent values.



(a) Previous implementation (before fixes)



(b) Revised implementation (after mesh and interface fixes)

Figure 8: Axial profiles in the hot and average channels under the Prob. 6 conditions: comparison before and after applying mesh and FV-interface fixes. The corrected model yields lower fuel-surface and centerline temperatures.

Outcome. After these modifications, the hot-channel peak temperature decreased from ~ 788 K to ~ 739 K, and the average-channel peak from ~ 734 K to ~ 699 K (values read from the updated profiles).

8.3 Power normalization across meshes

Even with identical physical parameters, slight numerical differences in the total generated power P_{gen} can occur due to mesh discretization. To eliminate this dependency, the volumetric heat source $q_{\mathrm{vol}}(z)$ was normalized so that the integrated power exactly matches the design value $\dot{q}$.

The normalization rescales q_{vol} by a factor

$$\text{scale} = \frac{P_{\mathrm{target}}}{P_{\mathrm{gen}}} = \frac{\dot{q}}{\sum_k \left[\frac{1}{2}(q_{\mathrm{vol},k} + q_{\mathrm{vol},k+1})A_f\right] \Delta z_k}, \qquad q_{\mathrm{vol}} \leftarrow \text{scale} \times q_{\mathrm{vol}}.$$

This ensures that $P_{\mathrm{gen}} = \dot{q}$ for any mesh refinement, making the temperature field and outlet temperature T_{out} independent of (N_c, N_f, N_z).

Table 1 compares T_{out} and P_{gen} across refinement cases, with and without normalization. When normalization is applied, the total generated power remains identical and the outlet temperature is consistent across all meshes, confirming mesh-independent energy balance.

Table 1: Comparison of outlet temperature and total generated power with and without `normalize_q_vol`.

Mesh setting (N_c, N_f, N_z)	With `normalize_q_vol`		Without `normalize_q_vol`	
	T_{out} [K]	P_{gen} [W]	T_{out} [K]	P_{gen} [W]
(3, 10, 21)	583.150	24,502.5245	583.096	24,467.6220
(4, 15, 31)	583.150	24,502.5245	583.125	24,486.5103
(5, 20, 41)	583.150	24,502.5245	583.136	24,493.3700
(6, 25, 51)	583.150	24,502.5245	583.141	24,496.6082
(7, 30, 61)	583.150	24,502.5245	583.144	24,498.3890