Transient Heat Conduction: Derivation, Cylindrical (r, z) Form, Lumped-Parameter Reduction, and Finite-Difference Discretization

Roberts and the Oracle

1 Purpose and Notation

These notes review the derivation of the transient heat conduction (diffusion) equation from an energy balance, specialize it to axisymmetric cylindrical coordinates (r, z), reduce it to a lumped-parameter (zero-dimensional) model for a body embedded in an environment, and develop a conservative finite-difference (finite-volume style) discretization in (r, z) with explicit rules for (i, j) indexing and matrix assembly.

Throughout we assume *spatially varying* material properties:

$$k = k(\mathbf{x}), \qquad \rho = \rho(\mathbf{x}), \qquad c_p = c_p(\mathbf{x}),$$

where $\mathbf{x} = (x, y, z)$ or (r, θ, z) as appropriate. We allow a volumetric heat source $\dot{q}''' = \dot{q}'''(\mathbf{x}, t)$. We *assume isotropic* conductivity for simplicity (scalar k). If any property is also a function of temperature (e.g., $k = k(\mathbf{x}, T)$ or $c_p = c_p(\mathbf{x}, T)$), then the governing equations and/or discrete system become *nonlinear in* T; this is noted where it affects formulation and numerics.

2 Derivation of the Transient Conduction Equation

Consider a differential control volume dV with boundary ∂V and outward unit normal $\mathbf{n}$. The first law of thermodynamics (energy balance) gives

$$\frac{d}{dt}\int_V \rho c_p T \, dV = -\int_{\partial V} \mathbf{q}\cdot\mathbf{n}\, dA + \int_V \dot{q}''' \, dV, \tag{1}$$

where $\mathbf{q}$ is the conductive heat flux. With Fourier's law $\mathbf{q} = -\,k\,\nabla T$ and the divergence theorem,

$$\frac{d}{dt}\int_V \rho c_p T \, dV = \int_V \nabla\cdot\big(k\nabla T\big)\, dV + \int_V \dot{q}''' \, dV. \tag{2}$$

Assuming sufficient smoothness and that V is arbitrary, the pointwise PDE follows:

$$\rho(\mathbf{x})\, c_p(\mathbf{x})\, \frac{\partial T}{\partial t} = \nabla\cdot\big(k(\mathbf{x})\,\nabla T\big) + \dot{q}'''(\mathbf{x}, t), \qquad \mathbf{x}\in\Omega. \tag{3}$$

This is in conservative (divergence) form, which is essential for accurate discretization with spatially varying k.

Initial and boundary data close the problem:

$$T(\mathbf{x}, 0) = T_0(\mathbf{x}), \tag{4}$$

$$T|_{\Gamma_D} = T_b(\mathbf{x}, t) \quad \text{(Dirichlet)}, \tag{5}$$

$$-\,k\,\nabla T\cdot\mathbf{n}\big|_{\Gamma_N} = q_b''(\mathbf{x}, t) \quad \text{(Neumann)}, \tag{6}$$

$$-\,k\,\nabla T\cdot\mathbf{n}\big|_{\Gamma_R} = h(\mathbf{x}, t)\,\big(T - T_\infty(\mathbf{x}, t)\big) \quad \text{(Robin/convective)}. \tag{7}$$

If k, ρ, or c_p also depend on T, eq. (3) remains valid but the operator depends on the solution, leading to a nonlinear PDE.

3 Axisymmetric Cylindrical (r, z) Form

Assume axisymmetry (no θ-dependence). In cylindrical coordinates, with $T = T(r, z, t)$ and $k = k(r, z)$,

$$\rho(r, z)\, c_p(r, z)\, \frac{\partial T}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\, k(r, z)\, \frac{\partial T}{\partial r}\right) + \frac{\partial}{\partial z}\left(k(r, z)\, \frac{\partial T}{\partial z}\right) + \dot{q}'''(r, z, t). \tag{8}$$

The axis condition at $r = 0$ for a smooth axisymmetric field is the natural symmetry (no-flux) condition

$$\left.\frac{\partial T}{\partial r}\right|_{r=0} = 0, \tag{9}$$

while outer radial and axial boundaries take standard Dirichlet/Neumann/Robin forms. Using the divergence (conservative) form (8) automatically accounts for spatially varying k via face-based fluxes when discretized.

4 Lumped-Parameter (Zero-D) Reduction

If internal temperature gradients are negligible (classically, a small Biot number based on the *smallest* k and a representative length), we approximate $T(\mathbf{x}, t) \approx \bar{T}(t)$. Define the thermal capacitance (heat capacity)

$$C \;\equiv\; \int_V \rho(\mathbf{x})\, c_p(\mathbf{x})\, dV, \tag{10}$$

which correctly aggregates spatially varying ρc_p. Let $U(t)$ be the overall heat transfer coefficient times area (possibly time- or state-dependent), and $\dot{Q}_{\text{gen}}(t) = \int_V \dot{q}'''(\mathbf{x}, t)\, dV$ the total volumetric generation. An energy balance for the body gives the ODE

$$C\,\frac{d\bar{T}}{dt} = -\,U(t)\,\big(\bar{T} - T_\infty(t)\big) + \dot{Q}_{\text{gen}}(t), \tag{11}$$

with $\bar{T}(0) = \bar{T}_0$. For constant U, T_∞, and no generation, the solution is

$$\bar{T}(t) = T_\infty + \big(\bar{T}_0 - T_\infty\big)\, \exp\!\left(-\,\frac{t}{\tau}\right), \qquad \tau = \frac{C}{U}. \tag{12}$$

If h, $\varepsilon\sigma$, or properties depend on temperature, then $U = U(\bar{T})$ and/or $C = C(\bar{T})$, and (11) becomes nonlinear in $\bar{T}(t)$.

5 Finite-Difference Discretization in (r, z)

We discretize (8) in a conservative, cell-centered fashion on a structured grid. Let $i = 1, \ldots, N_r$ index radial cell centers and $j = 1, \ldots, N_z$ index axial cell centers. For clarity, assume uniform spacings Δr and Δz, and place centers at

$$r_i = \big(i - \tfrac{1}{2}\big)\,\Delta r, \qquad z_j = \big(j - \tfrac{1}{2}\big)\,\Delta z,$$

so that the west face of cell $(1, j)$ lies at $r_{1-1/2} = 0$ (the axis). Define face radii and axial positions

$$r_{i\pm\frac{1}{2}} = r_i \pm \frac{\Delta r}{2}, \qquad z_{j\pm\frac{1}{2}} = z_j \pm \frac{\Delta z}{2}.$$

Let $T^n_{i,j}$ denote the cell-centered temperature at time level t^n, with step Δt. Define the volumetric heat capacity $C_{i,j} = \rho_{i,j}\, c_{p,i,j}$, with a cell average of ρc_p.

5.1 Face Conductivities and Conservative Stencil

To handle spatially varying k, evaluate conductivity on faces by harmonic averaging of neighboring cell values (second-order and interface-consistent):

$$k_{i+\frac{1}{2},j} = \left(\frac{1}{2}\frac{1}{k_{i,j}} + \frac{1}{2}\frac{1}{k_{i+1,j}}\right)^{-1}, \tag{13}$$

$$k_{i-\frac{1}{2},j} = \left(\frac{1}{2}\frac{1}{k_{i-1,j}} + \frac{1}{2}\frac{1}{k_{i,j}}\right)^{-1}, \tag{14}$$

$$k_{i,j+\frac{1}{2}} = \left(\frac{1}{2}\frac{1}{k_{i,j}} + \frac{1}{2}\frac{1}{k_{i,j+1}}\right)^{-1}, \tag{15}$$

$$k_{i,j-\frac{1}{2}} = \left(\frac{1}{2}\frac{1}{k_{i,j-1}} + \frac{1}{2}\frac{1}{k_{i,j}}\right)^{-1}. \tag{16}$$

For interior cells, approximating the divergence form in (8) by central differences in flux gives the conservative operator

$$\begin{aligned}\big[\nabla\cdot(k\nabla T)\big]_{i,j} \approx \frac{1}{r_i\,\Delta r}&\left[r_{i+\frac{1}{2}}\,k_{i+\frac{1}{2},j}\,\frac{T_{i+1,j}-T_{i,j}}{\Delta r} - r_{i-\frac{1}{2}}\,k_{i-\frac{1}{2},j}\,\frac{T_{i,j}-T_{i-1,j}}{\Delta r}\right]\\ &+\frac{1}{\Delta z}\left[k_{i,j+\frac{1}{2}}\,\frac{T_{i,j+1}-T_{i,j}}{\Delta z} - k_{i,j-\frac{1}{2}}\,\frac{T_{i,j}-T_{i,j-1}}{\Delta z}\right].\end{aligned} \tag{17}$$

Note that at the axis ($i = 1$), $r_{1-\frac{1}{2}} = 0$, so the west flux vanishes and (9) is naturally satisfied without special ghost cells.

5.2 Time Discretization: θ-Method

Define a time-weighted scheme ($\theta = 0$ explicit, $\theta = \frac{1}{2}$ Crank–Nicolson, $\theta = 1$ implicit):

$$C_{i,j}\,\frac{T^{n+1}_{i,j} - T^n_{i,j}}{\Delta t} = \theta\,\mathcal{L}[T^{n+1}]_{i,j} + (1-\theta)\,\mathcal{L}[T^n]_{i,j} + \dot{q}'''_{i,j}(t^{n+\theta}), \tag{18}$$

where $\mathcal{L}[\cdot]$ denotes the spatial operator in (17). For linear, T-independent properties, $\mathcal{L}$ is linear and the fully implicit and Crank–Nicolson schemes are unconditionally stable; for $\theta = 0$ (explicit), classical stability restrictions apply and become more severe near the axis due to the $1/r$ factor. If k or C depend on T, then (18) is nonlinear in T^{n+1} and must be solved iteratively (see the remark on nonlinearity below).

Expanding (18) for interior nodes yields the five-point stencil

$$a_P\,T^{n+1}_{i,j} - a_W\,T^{n+1}_{i-1,j} - a_E\,T^{n+1}_{i+1,j} - a_S\,T^{n+1}_{i,j-1} - a_N\,T^{n+1}_{i,j+1} = b_{i,j}, \tag{19}$$

with coefficients

$$D_W = \frac{r_{i-\frac{1}{2}}\, k_{i-\frac{1}{2},j}}{r_i\, \Delta r^2}, \quad D_E = \frac{r_{i+\frac{1}{2}}\, k_{i+\frac{1}{2},j}}{r_i\, \Delta r^2}, \quad D_S = \frac{k_{i,j-\frac{1}{2}}}{\Delta z^2}, \quad D_N = \frac{k_{i,j+\frac{1}{2}}}{\Delta z^2}, \tag{20}$$

$$a_W = \theta\, D_W, \qquad a_E = \theta\, D_E, \qquad a_S = \theta\, D_S, \qquad a_N = \theta\, D_N, \qquad a_P = \frac{C_{i,j}}{\Delta t} + a_W + a_E + a_S + a_N. \tag{21}$$

The right-hand side collects previous-time contributions and sources:

$$b_{i,j} = \frac{C_{i,j}}{\Delta t}\, T^n_{i,j} + (1-\theta)\Big[D_W\, T^n_{i-1,j} + D_E\, T^n_{i+1,j} + D_S\, T^n_{i,j-1} + D_N\, T^n_{i,j+1} - (D_W + D_E + D_S + D_N)\, T^n_{i,j}\Big] + \dot{q}'''_{i,j}\big(t^{n+\theta}\big). \tag{22}$$

For axis nodes ($i = 1$), $D_W = 0$ and the above formulas remain valid.

5.3 Boundary Conditions in the Discrete System

Dirichlet data at a boundary cell $(i,j) \in \Gamma_D$ is imposed by overwriting the stencil with

$$a_P \leftarrow 1, \qquad a_W = a_E = a_S = a_N \leftarrow 0, \qquad b_{i,j} \leftarrow T_b(t^{n+1}), \tag{23}$$

or by eliminating the unknown and modifying neighbors if preferred.

Neumann data $-k\,\partial T/\partial n = q''_b$ at an exposed face is enforced by replacing the face flux with the specified value. For example, at the outer radial face $r = r_{N_r+\frac{1}{2}}$ (east face of cell (N_r, j)), eliminate $T_{N_r+1,j}$ and set

$$\text{east flux} \;\approx\; r_{N_r+\frac{1}{2}}\, q''_b, \tag{24}$$

which contributes to $b_{i,j}$ with the appropriate sign via (17) and (22), while $a_E \leftarrow 0$.

Robin/convective data $-k\,\partial T/\partial n = h\,(T - T_\infty)$ can be enforced at a face by eliminating the ghost temperature with a second-order relation. At the east face (outer radius) of (N_r, j), a common discrete form is

$$T_{N_r+1,j} \;=\; \alpha\, T_{N_r,j} + \gamma, \qquad \alpha = \frac{1-\beta}{1+\beta}, \quad \gamma = \frac{2\beta}{1+\beta}\, T_\infty, \quad \beta = \frac{h\,\Delta r}{k_{N_r+\frac{1}{2},j}}, \tag{25}$$

which, when inserted into the east-difference term, yields a diagonal augmentation and a known contribution:

$$a_P \leftarrow a_P + a_E\,\alpha, \qquad b_{N_r,j} \leftarrow b_{N_r,j} + a_E\,\gamma, \qquad a_E \leftarrow 0. \tag{26}$$

Analogous formulas apply at the west (if not at the axis), south, and north faces, using the appropriate spacings and conductivities.

5.4 Indexing and Matrix Assembly

Map (i,j) to a single vector index p by

$$p = i + (j-1)\, N_r, \qquad 1 \le i \le N_r, \quad 1 \le j \le N_z, \tag{27}$$

so that east/west neighbors are at $p \pm 1$ and north/south neighbors at $p \pm N_r$ (when they exist). The linear system

$$\mathbf{A}\, \mathbf{T}^{n+1} = \mathbf{b} \tag{28}$$

has, for each interior node, five nonzeros in row p:

$$\begin{aligned} A_{p,p} &= a_P, & A_{p,p-1} &= -a_W \quad (i > 1), & A_{p,p+1} &= -a_E \quad (i < N_r), \\ A_{p,p-N_r} &= -a_S \quad (j > 1), & A_{p,p+N_r} &= -a_N \quad (j < N_z), \end{aligned} \tag{29}$$

with $\mathbf{b}_p = b_{i,j}$ from (22) plus any boundary contributions. Dirichlet rows contain a single 1 on the diagonal and the prescribed value in $\mathbf{b}$.

It is often convenient to write (18) in operator form

$$\big(\mathbf{M} + \theta\,\Delta t\,\mathbf{K}\big)\,\mathbf{T}^{n+1} = \big(\mathbf{M} - (1-\theta)\,\Delta t\,\mathbf{K}\big)\,\mathbf{T}^{n} + \Delta t\,\mathbf{s}^{n+\theta}, \tag{30}$$

where $\mathbf{M}$ is diagonal with entries $C_{i,j}$, $\mathbf{K}$ encodes the conservative diffusion stencil (and Robin augmentations), and $\mathbf{s}$ collects sources and boundary data. For spatially varying k and ρc_p independent of T, $\mathbf{K}$ and $\mathbf{M}$ are fixed; if $k = k(T)$ or $c_p = c_p(T)$, then $\mathbf{K}$ and/or $\mathbf{M}$ depend on $\mathbf{T}$ and (30) must be solved by nonlinear iterations (e.g., Picard with under-relaxation or Newton's method).

6 Remarks on Consistency, Stability, and Nonlinearity

The discretization above is *conservative* by construction and second-order accurate in space on uniform grids. Using harmonic face conductivities ensures correct flux continuity across material interfaces when k is discontinuous in space.

For the explicit scheme ($\theta = 0$), the time step Δt must satisfy a stability bound depending on the local diffusivity $\alpha = k/(\rho c_p)$ and spacings. On uniform grids with constant properties in a rectangular geometry, a typical bound is

$$\Delta t \lesssim \frac{1}{2\,\alpha}\,\frac{1}{\frac{1}{\Delta r^2} + \frac{1}{\Delta z^2}}, \tag{31}$$

while the (r,z) operator introduces additional geometric stiffness near $r = 0$. Fully implicit ($\theta = 1$) or Crank–Nicolson ($\theta = \frac{1}{2}$) schemes are unconditionally stable for *linear* problems.

If properties depend on temperature, the discrete system is nonlinear. A standard approach (Picard) is to freeze k and C at an iterate, solve (30), update T, and iterate until convergence. Newton's method converges faster but requires Jacobian contributions from $k(T)$ and/or $C(T)$.

7 Checklist of Common Boundary Conditions in (r,z)

For convenience, the most common boundary conditions in axisymmetric problems are summarized:

$$\begin{aligned} \text{Axis symmetry at } r = 0: &\quad \left.\frac{\partial T}{\partial r}\right|_{r=0} = 0, && (32) \\ \text{Insulated/adiabatic:} &\quad -k\,\nabla T\cdot\mathbf{n} = 0, && (33) \\ \text{Prescribed temperature:} &\quad T = T_b(\mathbf{x},t), && (34) \\ \text{Prescribed heat flux:} &\quad -k\,\nabla T\cdot\mathbf{n} = q_b''(\mathbf{x},t), && (35) \\ \text{Convection:} &\quad -k\,\nabla T\cdot\mathbf{n} = h(\mathbf{x},t)\,(T - T_\infty(\mathbf{x},t)). && (36) \end{aligned}$$

Discretization follows the rules described above (Dirichlet row replacement, Neumann as known face flux, Robin via the elimination (25)).

8 Summary

Starting from an energy balance with Fourier conduction and allowing spatially varying properties yields the transient heat equation in conservative form (3). In (r, z) under axisymmetry, the operator is (8) with the natural axis condition (9). When internal gradients are negligible, the field reduces to the lumped ODE (11) with the correct, spatially aggregated capacitance (10). A conservative (r, z) discretization based on face fluxes (17), harmonic face conductivities, and a θ-method in time leads to the five-point stencil (19)–(22), straightforward matrix assembly via the mapping (27), and clean implementations of Dirichlet/Neumann/Robin data. If properties are also temperature-dependent, the problem (and resulting algebraic system) becomes nonlinear, requiring iteration within each time step.

Appendix

To support the implementation of at least some of the ideas explored above, the function `construct_system` is provided to set up and solve the 2-D conduction equations in (r, z) geometry for steady-state conditions. Specifically, we consider the steady, axisymmetric equation

$$-\nabla\cdot\big(k\nabla T\big) = \dot{q}'''(r,z), \qquad \text{with no } \theta\text{-dependence.} \tag{37}$$

All material properties may vary spatially, i.e. $k = k(r, z)$. If a property is also temperature dependent, e.g. $k = k(r, z, T)$, the discrete problem becomes nonlinear and must be solved iteratively.

Grid, faces, and coefficient definitions

Use a uniform, cell-centered grid:

$$r_i = \Big(i - \tfrac{1}{2}\Big)\Delta r, \quad i = 1,\dots,N_r, \qquad z_j = \Big(j - \tfrac{1}{2}\Big)\Delta z, \quad j = 1,\dots,N_z, \tag{38}$$

with radial face radii $r_{i\pm\frac{1}{2}} = r_i \pm \frac{\Delta r}{2}$ (so $r_{1-\frac{1}{2}} = 0$ at the axis). Let $k_{i,j} = k(r_i, z_j)$. Use harmonic averaging for face conductivities:

$$k_{i+\frac{1}{2},j} = \left(\frac{1}{2k_{i,j}} + \frac{1}{2k_{i+1,j}}\right)^{-1}, \qquad k_{i-\frac{1}{2},j} = \left(\frac{1}{2k_{i-1,j}} + \frac{1}{2k_{i,j}}\right)^{-1}, \tag{39}$$

$$k_{i,j+\frac{1}{2}} = \left(\frac{1}{2k_{i,j}} + \frac{1}{2k_{i,j+1}}\right)^{-1}, \qquad k_{i,j-\frac{1}{2}} = \left(\frac{1}{2k_{i,j-1}} + \frac{1}{2k_{i,j}}\right)^{-1}. \tag{40}$$

Define nonnegative diffusion coefficients

$$D_W = \frac{r_{i-\frac{1}{2}}\, k_{i-\frac{1}{2},j}}{r_i\,\Delta r^2}, \quad D_E = \frac{r_{i+\frac{1}{2}}\, k_{i+\frac{1}{2},j}}{r_i\,\Delta r^2}, \quad D_S = \frac{k_{i,j-\frac{1}{2}}}{\Delta z^2}, \quad D_N = \frac{k_{i,j+\frac{1}{2}}}{\Delta z^2}. \tag{41}$$

Internal and boundary balance equations (uniform, cell-centered grid)

Interior nodes $(1 < i < N_r,\ 1 < j < N_z)$**.**

$$\big(D_W + D_E + D_S + D_N\big)\, T_{i,j} - D_W\, T_{i-1,j} - D_E\, T_{i+1,j} - D_S\, T_{i,j-1} - D_N\, T_{i,j+1} = \dot{q}'''_{i,j}. \tag{E1}$$

Axis nodes $(i = 1,\ 1 < j < N_z)$ **(symmetry at** $r = 0$**).** Here $r_{1-\frac{1}{2}} = 0 \Rightarrow D_W = 0$:

$$\big(D_E + D_S + D_N\big)\, T_{1,j} - D_E\, T_{2,j} - D_S\, T_{1,j-1} - D_N\, T_{1,j+1} = \dot{q}'''_{1,j}. \tag{E2}$$

Outer radial boundary $(i = N_r)$ **at** $r = r_{N_r+\frac{1}{2}}$ **(outward normal** $+\hat{r}$**).** *Dirichlet:*

$$T_{N_r,j} = T_b(z_j). \tag{E3D}$$

Neumann (specified outward heat flux): $-k\,\partial T/\partial r = q''_R(z_j)$. The known east-face flux adds to the RHS:

$$\big(D_W{+}D_S{+}D_N\big)\,T_{N_r,j} - D_W\,T_{N_r-1,j} - D_S\,T_{N_r,j-1} - D_N\,T_{N_r,j+1} = \dot{q}'''_{N_r,j} \;+\; \frac{r_{N_r+\frac{1}{2}}}{r_{N_r}\,\Delta r}\,q''_R(z_j). \tag{E3N}$$

Robin (convective): $-k\,\partial T/\partial r = h\,(T - T_\infty)$. With

$$\beta_E = \frac{h\,\Delta r}{k^\star_E}, \quad \alpha_E = \frac{1-\beta_E}{1+\beta_E}, \quad \gamma_E = \frac{2\beta_E}{1+\beta_E}\,T_\infty, \quad D^\star_E = \frac{r_{N_r+\frac{1}{2}}\,k^\star_E}{r_{N_r}\,\Delta r^2}, \tag{42}$$

the boundary row becomes

$$\big(D_W{+}D_S{+}D_N{+}D^\star_E\alpha_E\big)\,T_{N_r,j} - D_W\,T_{N_r-1,j} - D_S\,T_{N_r,j-1} - D_N\,T_{N_r,j+1} = \dot{q}'''_{N_r,j} + D^\star_E\gamma_E, \tag{E3R}$$

where a common choice is $k^\star_E = k_{N_r,j}$.

Top boundary $(j = N_z)$ **at** $z = z_{N_z+\frac{1}{2}}$ **(outward normal** $+\hat{z}$**).** *Dirichlet:*

$$T_{i,N_z} = T_t(r_i). \tag{E4D}$$

Neumann: $-k\,\partial T/\partial z = q''_t(r_i)$ adds to the RHS:

$$\big(D_W{+}D_E{+}D_S\big)\,T_{i,N_z} - D_W\,T_{i-1,N_z} - D_E\,T_{i+1,N_z} - D_S\,T_{i,N_z-1} = \dot{q}'''_{i,N_z} \;+\; \frac{q''_t(r_i)}{\Delta z}. \tag{E4N}$$

Robin: $-k\,\partial T/\partial z = h\,(T - T_\infty)$. With

$$\beta_N = \frac{h\,\Delta z}{k^\star_N}, \quad \alpha_N = \frac{1-\beta_N}{1+\beta_N}, \quad \gamma_N = \frac{2\beta_N}{1+\beta_N}\,T_\infty, \quad D^\star_N = \frac{k^\star_N}{\Delta z^2}, \tag{43}$$

we obtain

$$\big(D_W{+}D_E{+}D_S{+}D^\star_N\alpha_N\big)\,T_{i,N_z} - D_W\,T_{i-1,N_z} - D_E\,T_{i+1,N_z} - D_S\,T_{i,N_z-1} = \dot{q}'''_{i,N_z} + D^\star_N\gamma_N. \tag{E4R}$$

Bottom boundary $(j = 1)$ **at** $z = z_{\frac{1}{2}}$ **(outward normal** $-\hat{z}$**).** *Dirichlet:*

$$T_{i,1} = T_b(r_i). \tag{E5D}$$

Neumann: $-k\,\partial T/\partial n = q''_b(r_i)$ with $n = -\hat{z}$ yields the same magnitude RHS add:

$$\big(D_W{+}D_E{+}D_N\big)\,T_{i,1} - D_W\,T_{i-1,1} - D_E\,T_{i+1,1} - D_N\,T_{i,2} = \dot{q}'''_{i,1} \;+\; \frac{q''_b(r_i)}{\Delta z}. \tag{E5N}$$

Robin: With

$$\beta_S = \frac{h\,\Delta z}{k^\star_S}, \quad \alpha_S = \frac{1-\beta_S}{1+\beta_S}, \quad \gamma_S = \frac{2\beta_S}{1+\beta_S}\,T_\infty, \quad D^\star_S = \frac{k^\star_S}{\Delta z^2}, \tag{44}$$

the discrete balance is

$$\big(D_W{+}D_E{+}D_N{+}D^\star_S\alpha_S\big)\,T_{i,1} - D_W\,T_{i-1,1} - D_E\,T_{i+1,1} - D_N\,T_{i,2} = \dot{q}'''_{i,1} + D^\star_S\gamma_S. \tag{E5R}$$

Sign convention: $q'' > 0$ denotes heat *leaving* the solid (outward). Each term has units of $\mathrm{W\,m^{-3}}$.

Bare-bones Python builder for the steady system $-\nabla\cdot(k\nabla T)=\dot{q}'''$

The function below assembles a dense $\mathbf{A}\in\mathbb{R}^{(N_rN_z)\times(N_rN_z)}$ and $\mathbf{b}\in\mathbb{R}^{N_rN_z}$ on a uniform, cell-centered grid, using (E1)–(E5R). Boundary tuples:

`('dirichlet',T)`, `('neumann',q_flux)`, `('robin',h,T_inf)`.

If $k=k(r,z,T)$, rebuild $\mathbf{A},\mathbf{b}$ with the current iterate $T^{(m)}$ (e.g. Picard/Newton).

```python
import numpy as np

def construct_system(r_points, z_points, k_fun, qppp,
                     BC_z, BC_r_top, Bc_r_bottom):
    """
    Build dense A, b for -div(k grad T) = q''' on an axisymmetric (r,z)
       grid.

    Parameters
    ----------
    r_points : 1D array (Nr,) of radial cell centers, uniform, r[0] ~ Δr/2
    z_points : 1D array (Nz,) of axial cell centers, uniform, z[0] ~ Δz/2
    k_fun    : callable(r, z [, T]) -> k(r,z)
    qppp     : callable(r, z) -> volumetric source (W/m^3)
    BC_r_top : ('dirichlet',T) OR ('neumann',q'') OR ('robin',h,T_inf)
    BC_z     : tuple for TOP axial boundary
    Bc_r_bottom : tuple for BOTTOM axial boundary

    Returns
    -------
    A : (Nr*Nz, Nr*Nz) dense ndarray
    b : (Nr*Nz,) dense ndarray
    """
    r = np.asarray(r_points, dtype=float)
    z = np.asarray(z_points, dtype=float)
    Nr, Nz = r.size, z.size
    drs = np.diff(r); dzs = np.diff(z)
    dr, dz = drs[0], dzs[0]
    r_faces = np.hstack([0.0, r + 0.5*dr])  # r_{i-1/2}, r_{i+1/2}

    def idx(i,j): return i + j*Nr

    # cell-centered properties
    K = np.zeros((Nr,Nz)); Q = np.zeros((Nr,Nz))
    for i in range(Nr):
        for j in range(Nz):
            try:    K[i,j] = k_fun(r[i],z[j])
            except TypeError: K[i,j] = k_fun(r[i],z[j],None)
            Q[i,j] = qppp(r[i],z[j])

    def harm(a,b): return 2*a*b/(a+b) if (a>0 and b>0) else 0.0

    A = np.zeros((Nr*Nz,Nr*Nz)); b = np.zeros(Nr*Nz)

    for j in range(Nz):
        for i in range(Nr):
            p = idx(i,j); ri = r[i]; rw,re = r_faces[i],r_faces[i+1]
            DW=DE=DS=DN=0.0
            if i>0:  DW=(rw*harm(K[i-1,j],K[i,j]))/(ri*dr*dr)
            if i<Nr-1: DE=(re*harm(K[i,j],K[i+1,j]))/(ri*dr*dr)
            if j>0:  DS=harm(K[i,j-1],K[i,j])/(dz*dz)
            if j<Nz-1: DN=harm(K[i,j],K[i,j+1])/(dz*dz)

            diag=DW+DE+DS+DN
            A[p,p]+=diag; b[p]=Q[i,j]
            if i>0: A[p,idx(i-1,j)]-=DW
            if i<Nr-1: A[p,idx(i+1,j)]-=DE
            if j>0: A[p,idx(i,j-1)]-=DS
            if j<Nz-1: A[p,idx(i,j+1)]-=DN

            # outer radial boundary (i=Nr-1): Dirichlet, Neumann, Robin
            if i==Nr-1:
                kind=BC_r_top[0].lower()
                if kind=='dirichlet':
                    A[p,:]=0; A[p,p]=1; b[p]=BC_r_top[1]; continue
                elif kind=='neumann':
                    A[p,p]-=DE; qR=BC_r_top[1]
                    b[p]+=(re/(ri*dr))*qR
                elif kind=='robin':
                    h,Tinf=BC_r_top[1],BC_r_top[2]; k_face=K[i,j]
                    DEs=(re*k_face)/(ri*dr*dr)
                    beta=h*dr/max(k_face,1e-300)
                    alpha=(1-beta)/(1+beta); gamma=(2*beta/(1+beta))*Tinf
                    A[p,p]-=DE; A[p,p]+=DEs*alpha; b[p]+=DEs*gamma

            # top axial boundary (j=Nz-1)
            if j==Nz-1:
                kind=BC_z[0].lower()
                if kind=='dirichlet':
                    A[p,:]=0; A[p,p]=1; b[p]=BC_z[1]; continue
                elif kind=='neumann':
                    A[p,p]-=DN; qt=BC_z[1]; b[p]+=qt/dz
                elif kind=='robin':
                    h,Tinf=BC_z[1],BC_z[2]; k_face=K[i,j]
                    DNs=k_face/(dz*dz)
                    beta=h*dz/max(k_face,1e-300)
                    alpha=(1-beta)/(1+beta); gamma=(2*beta/(1+beta))*Tinf
                    A[p,p]-=DN; A[p,p]+=DNs*alpha; b[p]+=DNs*gamma

            # bottom axial boundary (j=0)
            if j==0:
                kind=Bc_r_bottom[0].lower()
                if kind=='dirichlet':
                    A[p,:]=0; A[p,p]=1; b[p]=Bc_r_bottom[1]; continue
                elif kind=='neumann':
                    A[p,p]-=DS; qb=Bc_r_bottom[1]; b[p]+=qb/dz
                elif kind=='robin':
                    h,Tinf=Bc_r_bottom[1],Bc_r_bottom[2]; k_face=K[i,j]
                    DSs=k_face/(dz*dz)
                    beta=h*dz/max(k_face,1e-300)
                    alpha=(1-beta)/(1+beta); gamma=(2*beta/(1+beta))*Tinf
                    A[p,p]-=DS; A[p,p]+=DSs*alpha; b[p]+=DSs*gamma
    return A,b
```

Listing 1: Bare-bones Python builder for the steady (r, z) conduction system

Notes: (i) The interior stencil maps directly to (E1); at the axis, $D_W = 0$ yields (E2) automatically. (ii) Neumann rows add the known face flux to b (see (E3N), (E4N), (E5N)). (iii) Robin rows augment the diagonal and the RHS as in (E3R), (E4R), (E5R). (iv) If $k(r, z, T)$, wrap a nonlinear loop that rebuilds $\mathbf{A}, \mathbf{b}$ at each iterate.