2D Heat Sink Model with Forced Air, Fins, and Implicit ADI Solver

This module solves a two-dimensional transient heat conduction problem in a rectangular plate or heat sink with forced convection to air and optional fins. The solution is based on a finite-difference discretization in the plate plane (x–y) and an unsteady implicit Alternating Direction Implicit (ADI) scheme, coupled to a simplified row-wise warmed-air model and a compact fin-efficiency treatment. The interface allows the user to draw heat source regions, finned regions, and adjust airflow parameters, while viewing temperature fields, air temperature fields, and probe histories in real time.

1. Geometry, Material, and Finite-Difference Grid

The plate represents the planform of a heat sink, PCB, or base plate with constant thickness t. The user specifies:

- Plate width W (m) in the horizontal x-direction.
- Plate height H (m) in the vertical y-direction.
- Plate thickness t (m).
- Solid material: thermal conductivity k (W/m·K), density ρ (kg/m^3), and specific heat c_p (J/kg·K).

The plate is discretized into a uniform, node-centered finite-difference grid with nx cells along the width and ny cells along the height. The code sets ny so that the aspect ratio of the grid roughly matches H/W:

• dx = W / nx
• ny ≈ nx × (H / W)
• dy = H / ny

Each cell represents a small rectangular volume with dimensions dx × dy × t. Temperature unknowns T_{ij} are associated with these cells. The thermal diffusivity is computed as:

$$\alpha = k / (\rho c_p)$$

2. Heat Sources and Source Mapping

Heat sources are defined interactively by dragging rectangular regions on the canvas while in “Draw Sources” mode. Each source region S_k is characterized by:

• A rectangle in physical space (projected from canvas pixels).
• Total source power amplitude P_k (W).
• Temporal type: constant or half-wave sinusoidal.
• Frequency f_k (Hz) for sinusoidal sources.

The code maps each source rectangle to a list of affected cells via a pixel-to-grid mapping. For each source and each time step t, a cell-averaged volumetric source term Q_i is built:

• For constant sources: $Q_i = P_k / N_k$ for all cells i in source k, where N_k is the number of cells in the source region.
• For half-wave sinusoidal sources: a time factor $\max(0, \sin(2\pi f_k t))$ is applied to represent periodic pulsed heating.

In the steady-state solver, sinusoidal sources are converted to an equivalent time-averaged power using a $1/\pi$ factor, approximating the mean of the positive half-wave.

3. Fins and Effective Convective Area

Fins are also defined by drawing rectangular regions on the canvas in "Draw Fins" mode. Each fin band is described by:

• Fin thickness t_f (m).
• Fin length L_f (m) protruding into the flow.
• Fin spacing s (m) between fin centers or gaps.
• Fin material conductivity k_f (W/m·K).

The code converts each fin band into an effective augmentation of the convective area in each covered cell. For a given cell with width dx and height dy, the number of fins per unit width is approximated by:

$$N_f \approx dx / (t_f + s)$$

A simple one-dimensional rectangular-fin efficiency model is used. Defining:

• h – local convective heat-transfer coefficient (W/m²·K).
• $m = \sqrt{2 h / (k_f t_f)}$.
• L_f – fin length (m).

The fin efficiency η_f is approximated as:

$$\eta_f = \tanh(m L_f) / (m L_f)$$

The effective additional fin area $A_{f,eff}$ in a cell is then:

$$A_{f,eff} \approx \eta_f \times (2 L_f\, dy\, N_f)$$

This fin area is added to the base plate area to form an effective convective area A_{eff}, which is later used in the heat balance for each cell.

4. Convective Model and Warmed Air Treatment

The plate exchanges heat with a moving air stream. The convective heat transfer is modeled using an h–A_{eff}–ΔT term on each cell and a row-wise air-energy balance that warms the air as it flows across the plate.

The user specifies:

• Air velocity v (m/s).
• Air gap height H_gap (m).
• Air density ρ_air (kg/m^3) and specific heat c_p,air (J/kg·K).
• Air inlet temperature T_in (°C).
• Lateral mixing parameter λ (0–0.2).
• Flow direction: left→right or right→left.
• Option to expose one or both sides of the plate to the same air stream.

The overall convective coefficient h is estimated from the air velocity using a simple empirical correlation of the form:

$$h \approx 1.16\ (10.45 - v + 10\sqrt{v})$$

This provides a rough forced-convection coefficient that increases with velocity. For each cell, a base plate area (one or two sides) plus any fin-contributed area yields the effective convective area A_eff,i.

To account for air heating as it passes over the plate, the code computes a two-dimensional air-temperature field Ta(x,y) via a row-wise energy balance. For a given row j, the total mass flow rate across the channel is:

$$\dot{m}\_total = \rho\_air\ v\ H\_gap$$

and the mass flow rate per row is ṁ_row = ṁ_total / ny.

The air temperature is marched from inlet to outlet. At each cell along the flow direction, the net heat transferred from the plate to the air is approximated as:

$$Q\_i = h\ A\_eff,i\ \max(0,\ T\_i - Ta)$$

The air temperature is then updated using an energy balance:

$$T\_a,new = T\_a,old + Q\_i / (\dot{m}\_row\ c_p,air)$$

This marching is performed separately for each row in the chosen flow direction. Additionally, a simple lateral mixing operator is applied to Ta(x,y) when the user specifies a non-zero mixing parameter λ, implemented as a vertical discrete diffusion step. If the warmed-air option is disabled, the air temperature is held fixed at T_in everywhere.

5. Governing Equation and Semi-Discrete Form

The plate temperature evolution is governed by the transient heat equation with in-plane conduction, volumetric heat sources, and convective cooling:

$$\rho\ c_p\ t\ \partial T/\partial t = k\ t\ (\partial^2 T/\partial x^2 + \partial^2 T/\partial y^2) + \dot{Q} - h\ A\_eff\ (T - T\_a)$$

Here, Q̇ denotes the total applied power in the control volume (W), divided by the control-volume volume V = dx dy t to obtain an equivalent volumetric source term. In the finite-difference discretization, this becomes for each cell (i,j):

$$\rho\, c_p\, V\, (dT_{ij}/dt) = k\, t\, [\, (T_{i+1,j} - 2\, T_{ij} + T_{i-1,j})/dx^2 + (T_{i,j+1} - 2\, T_{ij} + T_{i,j-1})/dy^2\,] + Q_{ij} - h\, A_{eff,ij}\, (T_{ij} - T_{a,ij}).$$

Introducing the thermal diffusivity α and dividing by $\rho\, c_p\, V$ yields the semi-discrete form suitable for time integration. Adiabatic (zero-flux) boundary conditions are implemented at the outer edges by modifying the finite-difference stencils in the tri-diagonal systems.

6. Implicit ADI Time Integration

Time integration is carried out using an implicit Alternating Direction Implicit (ADI) scheme. Each full time step Δt is split into two half-steps:

1. 1. x-implicit / y-explicit half-step.
2. 2. y-implicit / x-explicit half-step.

For the x-implicit half-step, each row j is treated independently. Using central differences in x and y and implicit weighting in x, the update at intermediate time level n+½ leads to a tri-diagonal system in T^*_{ij}. The right-hand side contains explicit contributions from conduction in y, source Q_{ij}, and convection to the air temperature $T\_a_{,ij}$.

The y-implicit half-step is analogous, sweeping each column i with a tri-diagonal solve that is implicit in y and explicit in x. In each sweep the module assembles coefficient vectors a, b, c and a right-hand side d, solving the tri-diagonal system via the Thomas algorithm.

Because both spatial directions are treated implicitly in alternating fashion, the ADI method is unconditionally stable for pure diffusion, and inclusion of the convection sink term implicitly further improves numerical robustness when h A_eff is large.

7. Transient Simulation and Auto-Stop Logic

The transient solver advances the temperature field in time starting from an initial condition (usually uniform at ambient). For each animation frame, the code performs multiple ADI substeps, logs the probe temperature, and redraws the temperature and air fields.

The time step Δt can be specified by the user or chosen automatically using a diffusion-based estimate:

$$\Delta t_{auto} \approx 0.02\, L^2 / \alpha$$

An auto-stop heuristic monitors the maximum plate temperature. If the relative change |T_max,new − T_max,old| / T_max,old remains below a small threshold for several iterations, the simulation is paused and flagged as having reached a near-steady state.

8. Steady-State Solver via Pseudo-Transient Marching

The steady-state solver constructs a time-averaged source field Q_{ij} and marches the heat equation forward in pseudo-time using larger time steps until convergence. The algorithm:

• Aggregates source powers (including sinusoidal sources via a $1/\pi$ factor).
• Sets Δt proportional to h_cell^2 / α, with h_cell = max(dx, dy).
• Periodically recomputes the warmed-air field T_a(x,y).
• Performs several ADI sweeps per iteration to accelerate convergence.
• Checks the maximum absolute temperature change per iteration against a tolerance.

Once converged, the solver reports the effective time step, maximum temperature change, and approximate total power Q_total entering the plate. The final temperature and air fields can be inspected visually or through the probe interface.

9. Post-Processing, Probes, and Plotting

The module supports several post-processing features:

• A probe at physical coordinates (x,y), mapped to the nearest cell, with instantaneous temperature readout.
• Continuous logging of probe temperature versus time during transient runs.
• A separate 2D plot of T_probe(t), with auto-scaled axes and simple grid lines.
• CSV export of the probe history for external analysis.
• Color heatmaps of the plate temperature and optional air-temperature overlay.
• Visual highlighting of source regions and finned regions on the main canvas.

10. Boundary Conditions and Numerical Details

The finite-difference operators implement approximately adiabatic (zero-flux) boundary conditions at the outer edges of the plate by modifying the tri-diagonal coefficients at the first and last nodes in each sweep. This is consistent with either a physically insulated edge or a symmetry plane when modeling a repeating pattern.

All temperature, air-temperature, source, and area fields are stored in 1D arrays of length $N = nx \times ny$ with row-major indexing ($index = j \cdot nx + i$). Rendering functions convert these to canvas pixel coordinates for display.

11. Model Assumptions and Limitations

The module is designed as a fast engineering tool for exploring heat sink and forced-convection layouts, not as a full CFD/FEA solver. Key assumptions and limitations include:

• The plate is modeled as 2D with uniform thickness; through-thickness gradients are not resolved.
• Material and air properties are taken as constant and temperature independent.
• The convective coefficient h is estimated from a simple empirical correlation that may not capture detailed boundary-layer or turbulence effects.

• The warmed-air model is one-dimensional along the flow direction in each row, with crude lateral mixing; complex 3D flow structures are neglected.
• Fins are represented through fin-efficiency and area augmentation models rather than fully resolved 2D or 3D fin conduction.
• Radiation and conduction into supports or neighboring components are ignored.

Within these assumptions, the module provides an efficient, interactive platform for comparing fin layouts, airflow conditions, and source distributions, and for building intuition about forced-air heat sink behavior.

12. Reference Correlations for Convective Heat-Transfer Coefficient h

The convective heat-transfer coefficient used in this module is based on standard forced-convection correlations and empirical relations widely used in electronics cooling. The implemented form:

$$h \approx 1.16\,(10.45 - v + 10\sqrt{v})$$

is adapted from the classical McAdams correlation for mixed natural/forced convection over small surfaces. Although originally developed for air flow across compact geometries, it is still used extensively in quick-estimate electronics thermal design.

More formal convection correlations are derived from flat-plate boundary-layer theory, expressed through Nusselt, Reynolds, and Prandtl numbers:

$$Nu_x = 0.332\, Re_x^{1/2}\, Pr^{1/3} \quad \text{(laminar)}$$

$$Nu_x = 0.0296\, Re_x^{0.8}\, Pr^{1/3} \quad \text{(turbulent)}$$

These yield an average plate coefficient of the form $h = k_{air} Nu / L$ and serve as the basis for many heat-sink design approaches.

Key references:

- McAdams, W.H., *Heat Transmission*, McGraw–Hill (1954).
- Incropera, DeWitt, Bergman, Lavine, *Fundamentals of Heat and Mass Transfer*, Wiley.
- Holman, J.P., *Heat Transfer*, McGraw–Hill.
- Ellison, G., *Thermal Computations for Electronics*, CRC Press.