Structural and Multidisciplinary Optimization (2025) 68:167
https://doi.org/10.1007/s00158-025-04088-7

RESEARCH PAPER

A subspace method for 3D multiscale heat sink modelling and optimization

Dilaksan Thillaithevan[1] · Robert Hewson[1] · Ryan Murphy[1] · Matthew Santer[1] · Alex Carver[2] · Giannis Nikiteas[2] · Nicholas Raske[2]

Received: 1 April 2025 / Revised: 25 June 2025 / Accepted: 14 July 2025

Abstract

The increasing computational demands of modern microprocessors require efficient thermal management solutions. To address this design challenge, a novel 3D multiscale heat sink modelling and optimization framework is presented. The approach combines a multiscale momentum model with an iterative temperature-flux projection scheme that accurately resolves heat transport across the entire macroscale domain without relying on traditional homogenization methods. Unlike conventional fixed-grid topology optimization approaches, the framework maintains solution accuracy near solid/fluid interfaces whilst achieving superior computational efficiency, reducing memory requirements and computation time relative to equivalent explicit single-scale simulations. Bayesian optimization is utilized to demonstrate the framework's practical utility by designing multiscale heat sinks with hundreds of unit cells subject to homogeneous, and more accurate in-homogeneous surface heat flux, achieving significantly larger heat transfer in both cases, whilst maintaining pressure drop constraints. This framework enables the practical optimization of complex 3D heat sink designs at multiscale resolutions previously intractable with traditional explicit modelling approaches.

Keywords Heat sink modelling · Multiscale modelling · Heat transfer · Heat sink optimization · Multiscale optimization · Topology optimization · Bayesian optimization

1 Introduction

Heat sinks are essential thermal management devices that remove excess heat from critical components across various industries. In computing systems, they are particularly vital for cooling microprocessors, where efficient heat dissipation maintains optimal operating temperatures, enhancing both performance and long-term reliability (Rao and Vrudhula 2007). Additionally, increased heat sink efficiency can reduce energy consumption through lower pump power requirements, leading to lower operational costs and a smaller environmental impact. Generating simulation informed efficient heat sink design is non-trivial, and accurate heat sink modelling requires numerical simulations of complex fluid–structure interactions involving momentum and heat transfer. In addition, the coupling between the fluid and heat flows results in highly non-linear heat sink design spaces, making it challenging to obtain globally optimal solutions. However, this complexity also presents an opportunity to tailor heat sink designs to outperform conventional heat sink designs that commonly rely on fixed, repeated, elements like pin-fins or microchannels.

The heat sink domain, $\Omega \in \mathbb{R}^3$, is defined as, $\Omega = \Omega_f \cup \Omega_s$, with the interface, Γ_I, defining the boundary between the fluid (Ω_f) and solid (Ω_s) domains. Three primary approaches are commonly used for parameterizing Γ_I for modelling and optimization, listed in order of decreasing modelling accuracy (and inversely, decreasing computational cost) these are *explicit*, *level-set* (LS) and *density-based topology optimization* (TO). As the name suggests, the explicit representation directly defines, Ω_s, Ω_f and, most importantly, Γ_I. In a finite element (FE) setting, Ω_f and Ω_s are discretized in a conformal manner to exactly represent the geometry, with Γ_I defined on the facets at the intersection of the two domains.

Responsible Editor: Vikrant Aute.

✉ Dilaksan Thillaithevan
d.thillaithevan18@imperial.ac.uk

[1] Department of Aeronautics, Imperial College London, London, United Kingdom

[2] ToffeeX, London, United Kingdom

Published online: 30 August 2025

This enables an accurate description of the fluid and thermal behaviour across and near Γ_{I}. However, whilst this representation enhances the solution accuracy, it does so at a significant computational cost due to the resolution needed to represent the geometry and adequately model boundary effects (near Γ_{I}). As a result, the explicit representation is seldom used in heat sink optimization workflows. From an optimization perspective, another added difficulty with the explicit representation is the difficulty in obtaining design derivatives, which hinders the use of gradient-based optimizers. For instance, Shimoyama *et al.* optimized a lattice-based 3D heat sink design using an explicit representation where the design variables (lattice edges) were parameterized as a node-edge system (Shimoyama and Komiya 2022). Heat sink performance was evaluated using an explicit Computational Fluid Dynamics model (modelling momentum and heat transfer), coupled to a Bayesian optimization workflow, which does not rely on derivative information. However, the optimization is performed on domain containing only 68 lattice edges, likely due to the computational demands of the explicit modelling. Bacellar *et al.* introduced a framework for 2D shape optimization of aerofoil structures for air-to-fluid cross-flow heat exchangers (Bacellar et al. 2017), with a similar approach subsequently developed in Tancabel et al. (2022). The authors use 2D shape optimization to tailor the profile of the internal tubes in the heat exchanger and demonstrate, both numerically and experimentally, significant performance improvements over baseline designs. These results underscore the potential for heat exchanger/heat sink optimization whilst also highlighting the limitation of current explicit modelling techniques as it forces researchers to restrict their analysis to 2D problems and or simple domains with a limited set of geometrical features.

In principle, LS methods also provide a clear description of the fluid–solid interface(s). These methods rely on a binary field, $\phi(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$, whose zero level-set, $\phi = 0$, (typically) defines Γ_{I}, enabling a well-defined, differentiable, description of Γ_{I}. For instance in Troya et al. (2021), the authors perform LS-based TO in 3D of two-fluid heat exchangers. They employ an Ersatz approach to model complimentary hot and cold fluid domains using Brinkman penalization to solve the momentum transfer on Ω and validate their framework through the optimization of several heat exchangers with different pressure drop constraints and Reynolds number configurations. However, the interface between the two fluids is modelled as an infinitesimally thin membrane, introducing some limitations in capturing the full physical behaviour of the heat exchangers. The work by Feppon *et al.* demonstrated design optimization of 2D and 3D fluid-to-fluid heat exchangers through an explicit mesh presentation, enabled by a level-set description of the fluid–sold or fluid–fluid domains (Feppon et al. 2021). This approach builds on the framework introduced in Allaire et al. (2014), in which the level-set function is iteratively updated using shape derivatives, which in turn enables explicit, body-fitted, mesh representations. The optimization framework incorporates constraints on the minimum wall thickness by penalizing regions where thickness drops below a prescribed threshold. The authors conclude that the optimized designs often exhibit multiscale features characterized by periodic structures, providing further confirmation of the relevance and necessity of the present study. LS methods are also generally susceptible to several numerical challenges (Van Dijk et al. 2013), including strong sensitivity to initial conditions and convergence to local minima as observed in Feppon et al. (2021). In addition, careful implementation of regularization techniques, such as diffusion (Liu and Korvink 2008), sensitivity smoothing (Wang and Wang 2004), and signed-distance reinitialization (Wang et al. 2003), is necessary to ensure optimization robustness and stability.

Lastly, density-based TO methods parameterise Ω using a fixed grid and a fictitious, binary, design variable field $\rho(\mathbf{x}) : \Omega \rightarrow \{0, 1\}$, which describes the fluid domain as $\rho = 0 \in \Omega_f$ and $\rho = 1 \in \Omega_s$ for the solid domain. To form a well-posed optimization formulation, the discrete design variable field, $\rho(\mathbf{x})$, is relaxed to a continuous $\rho(\mathbf{x}) : \Omega \rightarrow [0, 1]$ representation. Although this relaxation is beneficial from an optimization perspective, it is detrimental for modelling interface problems (such as heat sinks). Specifically, Γ_{I} is no longer clearly defined, since Γ_{I} exists on $0 < \rho < 1$, accurately capturing the fluid and thermal behaviour at the interface is an issue and requires additional numerical procedures, such as Heaviside projection (Wang et al. 2011) to obtain crisp boundaries. To solve the momentum transfer problem a Brinkman (or Darcy) term, of the form $-\alpha(\rho)\boldsymbol{u}$ (Borrvall and Petersson 2003; Olesen et al. 2006), is typically included in the Navier–Stokes (NS) formulation to penalize fluid flow in Ω_s. For instance, Høghøj et al. (2020) formulated a TO framework for the design of 2D and 3D two-fluid heat exchangers. Using the RAMP interpolation (Borrvall and Petersson 2003), a erosion-dilation (Wang et al. 2011; Schevenels et al. 2011) process was used guarantee minimum thickness walls between the two fluids. The optimization problem was formulated to maximize heat transfer, subject to a maximum pressure drop constraint. As the authors note, whilst the TO designs were found to display improved performance relative to baseline designs, converting the density-based designs into the explicit representations needed for performance validation and CAD (manufacturing) is an on-going challenge. Similarly, Kobayashi et al. (2021) formulated a TO problem to find the Ersatz material distribution to maximize heat transfer in a heat exchanger whilst limiting the pressure drop. In their formulation, they define $0 < \rho < 1 \in \Omega_s$, $\rho = 1 \in \Omega_{f,1}$ and $\rho = 0 \in \Omega_{f,2}$, where $\Omega_{f,1}$ and $\Omega_{f,2}$ represent the two-fluid domains. Whilst this formulation guarantees a solid wall between the two fluids, unlike (Høghøj et al. 2020), their formulation does not ensure a minimum wall thickness. In Yaji et al. (2022), a density-based method is integrated into a multi-fidelity optimization framework for heat sink design. Darcy flow is used for solving multiple low-fidelity optimization problems, and these solutions are subsequently validated through high-fidelity Reynolds-Averaged Navier Stokes (RANS) simulations. The simulation results feed into an evolutionary algorithm combined with a Variational Autoencoder (VAE) to generate improved designs. Whilst reference modelling was performed using a 3D heat sink model, the optimization is performed on a 2D domain. Collectively, these examples highlight the challenges of modelling heat sink problems using density-based TO. Namely, whilst fixed-grid TO lends itself to optimization, modelling accuracy is impaired and generating manufacturing friendly explicit representations from fixed-grid designs remains a challenge.

In this paper, a novel 3D explicit representation multiscale heat sink model for optimizing cross-flow heat sinks is presented. This model enables accurate modelling of fluid/thermal behaviour near boundaries, whilst enabling designers to choose manufacturable microstructure designs. The main contributions are as follows:

- Development and validation of a non-homogenization method for solving advection–diffusion in multiscale heat sinks;
- A concurrent multiscale advection–diffusion solver that efficiently models heat sinks with hundreds of microstructures, whilst maintaining single-scale accuracy, enabling better scaling and reduced computational loads;
- Demonstration of multiscale 3D heat sink design using Bayesian optimization containing 400 unit cells for both homogeneous and in-homogeneous surface fluxes.

2 Methodology

This work considers the modelling and optimization of cross-flow heat sinks. Heat sinks dissipate heat from external sources, e.g. a CPU, which imposes a heat flux on an exterior surface of the heat sink. Coolant fluid enters at one end of the plate, traverses the internal structure and exits the domain at the other end, as shown in Fig. 1.

The efficiency of an heat sink is determined by the level of heat transfer across the structure and the pressure loss experienced by the fluid (which is representative of the pump power needed to drive the fluid). Therefore, the objective is to efficiently model the heat transfer and pressure loss for a given heat sink design, $\chi(\Omega)$, enabling tailoring of χ through optimization. This involves solving a momentum transfer problem in Ω_{f}, and heat transfer on Ω. Both of these tasks are computationally demanding and typically limit the design optimization of heat sinks. To overcome this, a novel multiscale framework for the modelling of heat sinks is presented.

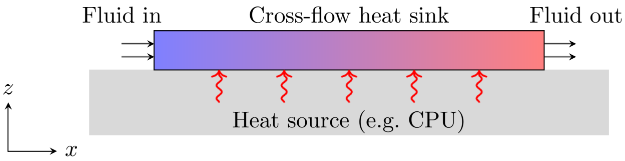


Fig. 1 Passive cross-flow heat sink domain

2.1 Multiscale notations and setting

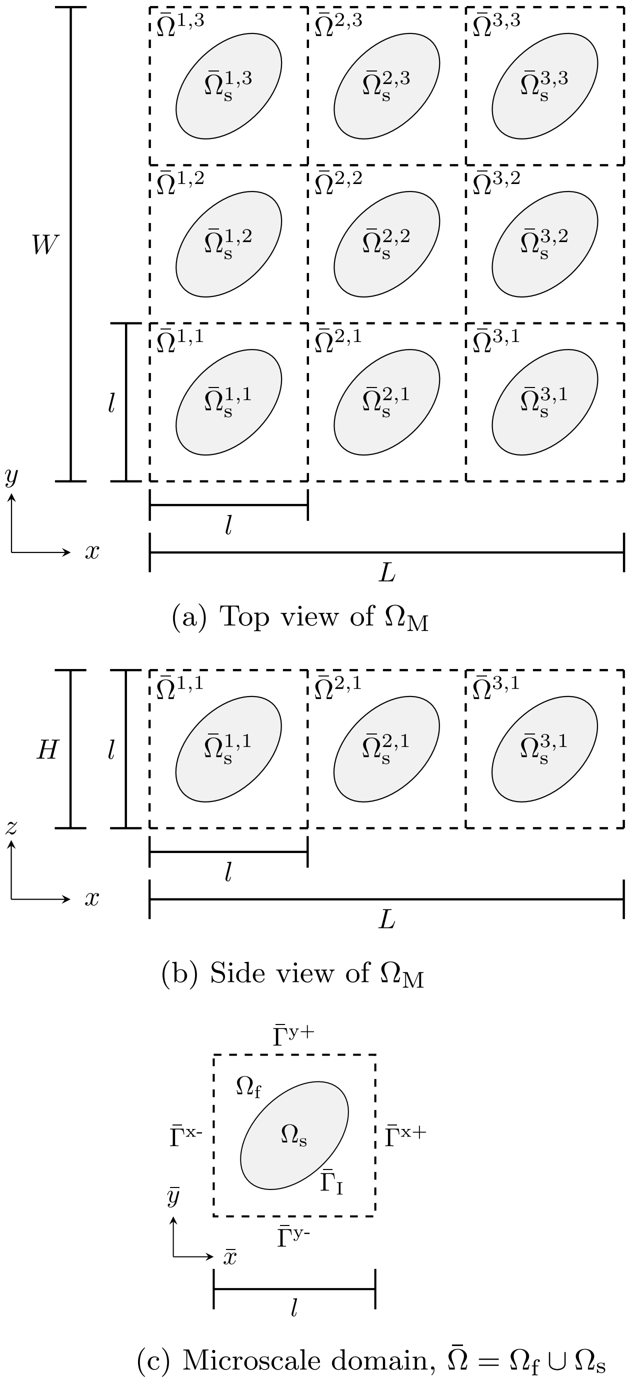
In a multiscale setting, the macroscale domain is defined as $\Omega = \Omega_{\mathrm{M}} \in \mathbb{R}^3$, using the composition of $N_x \times N_y \times N_z$ unit cells, $\bar{\Omega}$

$$\Omega_{\mathrm{M}}(\mathbf{x}_{\mathrm{M}}) = \bigcup_{i,j=0}^{N_x-1,N_y-1} \bar{\Omega}^{i,j}$$
$$\text{where } \bar{\Omega}^{i,j} := l(ie_1 + je_2) + \bar{\Omega}, \qquad \forall i \in [0, N_x - 1], \forall j \in [0, N_y - 1], \tag{1}$$

where $|\bar{\cdot}|$ and $|\cdot|_{\mathrm{M}}$ are used throughout this work to refer to micro- and macroscale quantities, respectively, $N_z = 1$ and e_i $(1 \le i \le d)$ are the regular bases of $\mathbb{R}^d$. N_x and N_y are the number of unit cells in the x and y directions, respectively, as shown in Fig. 2. Each unit cell is defined as $\bar{\Omega} = \bar{\Omega}_{\mathrm{f}} \cup \bar{\Omega}_{\mathrm{s}}$, where the subscripts $|\cdot|_{\mathrm{f}}$ and $|\cdot|_{\mathrm{s}}$ refer to the fluid and solid domains, respectively, separated by the boundary, $\bar{\Gamma}_{\mathrm{I}}$. The geometric control parameter, χ, controls the topology of $\bar{\Gamma}_{\mathrm{I}}$, which directly influences the momentum and heat transfer within Ω , and $\chi \in \mathbb{R}^{n_d \times N_x \times N_y}$ is a design variable matrix which controls $\bar{\Gamma}_{\mathrm{I}}$ in the (i, j)-th unit cell, $\bar{\Omega}^{i,j}$, where n_d is the number of design variables per unit cell. To reduce the computational burden, Ω is considered to be symmetric about the $\bar{z} = 0.5\,(z = H/2)$ plane, assuming a heat source is placed on $z = 0$ and $z = H$ as shown in Fig. 4. Lastly, $\bar{\Omega} \in \mathbb{R}^3$ is parameterized using a pin-fin microstructure with $n_d = 1$, as shown in Fig. 3, with variable pin radius such that $\chi \equiv r^* = r/l$ and $n_d = 1$. The fixed variables $t = t^* l$ and $h^*_t = 1 - 2t^*$ control the plate thickness at the top and bottom of the pin-fin, where $t^* = 0.1$ is used for the remainder of this work.

2.2 Multiscale momentum transfer

Assuming a sufficient scale of separation exists between the macro- and microscale, i.e. $\frac{l}{L} << 1$ and smoothly varying microscale geometry, the flow within $\Omega_{\mathrm{M,f}} = \cup\bar{\Omega}_{\mathrm{f}}$ can be characterized by periodic velocity and pressure gradients (Raske



(a) Top view of Ω_M

(b) Side view of Ω_M

(c) Microscale domain, $\bar{\Omega} = \Omega_f \cup \Omega_s$

Fig. 2 Partition of multiscale domain, $\Omega_M = \cup\bar{\Omega}$ and definition of $\bar{\Omega}$

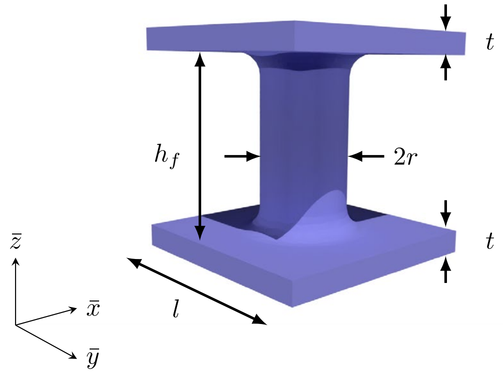


Fig. 3 Parameterized pin-fin microstructure

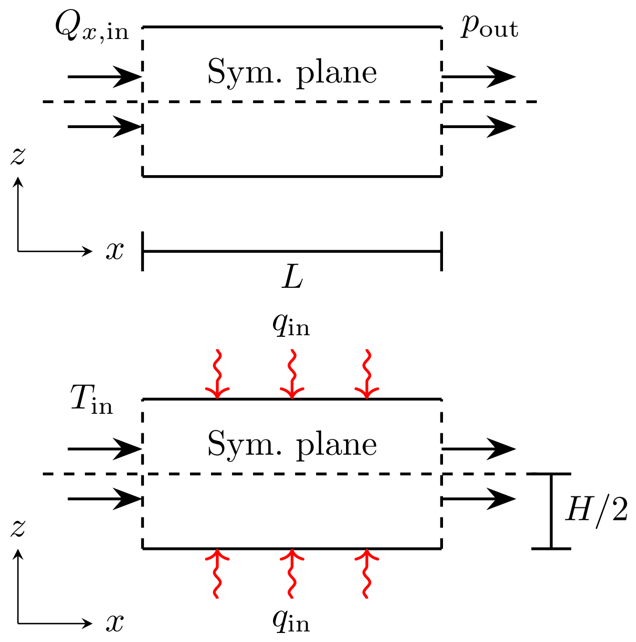


Fig. 4 Heat sink boundary conditions for fluid flow (left) and heat flow (right)

et al. 2017; Hewson et al. 2011; Sparrow and Prata 1983). This characterization facilitates the use of homogenization theory to model the flow behaviour in $\Omega_{M,f}$ by considering the homogenized properties (e.g. permeability) of each $\bar{\Omega}$, rather than directly solving the momentum transfer problem on Ω_f. In addition, the flow is assumed to be incompressible and steady, with the momentum transfer influencing the heat transfer, but not vice versa, resulting in a weak, one-way, coupled system.

2.2.1 Microscale momentum transfer

On the microscale, the aim is to find the periodic, dimensional, velocity-pressure pair $\boldsymbol{u} : \bar{\Omega}_f^{i,j} \to \mathbb{R}^3$ and $p : \bar{\Omega}_f^{i,j} \to \mathbb{R}$ such that

$$
\begin{aligned}
\mathcal{R}(\boldsymbol{u},p) \equiv \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nu\nabla^2\boldsymbol{u} + \frac{1}{\rho}\nabla p - \overline{\nabla p} &= 0 \text{ in } \bar{\Omega}_{\mathrm{f}} \\
\nabla\cdot\boldsymbol{u} &= 0 \text{ in } \bar{\Omega}_{\mathrm{f}} \\
\boldsymbol{u} &= 0 \text{ on } \bar{\Gamma}_{\mathrm{I}} \\
\boldsymbol{u}|_{\bar{\Gamma}^{\mathrm{x-}}} &= \boldsymbol{u}|_{\bar{\Gamma}^{\mathrm{x+}}} \\
\boldsymbol{u}|_{\bar{\Gamma}^{\mathrm{z-}}} &= \boldsymbol{u}|_{\bar{\Gamma}^{\mathrm{z+}}}
\end{aligned}
\tag{2}
$$

where $\overline{\nabla p}$ is the pressure gradient forcing term arising from the macroscale, the derivation of which will be explained in the following section. Equation 2 is solved using the finite element method (FEM), using first-order Lagrange elements for both velocity and pressure, the FE space, $W = \boldsymbol{V}_h \times Q_h$, is defined as

$$
\begin{aligned}
\boldsymbol{V}_h =& \{\boldsymbol{v}_h \in [H_0^1(\bar{\Omega})]^d \,:\, \boldsymbol{v}_h|_K \in [\mathbb{P}_1(K)]^d \ \forall K \in \mathcal{T}_h \mid \\
& \boldsymbol{v}_h^{\mathrm{x-}} = \boldsymbol{v}_h^{\mathrm{x+}}, \boldsymbol{v}_h^{\mathrm{z-}} = \boldsymbol{v}_h^{\mathrm{z+}} \text{ and } \boldsymbol{v}_h = 0 \text{ on } \bar{\Gamma}_{\mathrm{I}}\} \\
Q_h =& \{q_h \in [H_0^1(\bar{\Omega})]^d \,:\, q_h|_K \in \mathbb{P}_1(K) \ \forall K \in \mathcal{T}_h\},
\end{aligned}
\tag{3}
$$

where $\mathcal{T}_h$ is a tetrahedral partition of $\bar{\Omega}$ and $d = 3$ is the spatial dimension. The periodic boundary conditions (PBC) are imposed on Q_h, and the pressure component is only defined up to a constant. As a result, the constant pressure null space is eliminated in the underlying `PETSc` solver. In addition, a Galerkin Least Squares (GLS) stabilization (Hughes and Franca 1987) term is included in the weak form of Equation 2 since first-order Lagrange elements for velocity and pressure are used. The resulting discretized problem is to find $(\boldsymbol{u},p) \in W$ such that $\forall(\boldsymbol{v}_h, q_h) \in (\boldsymbol{V}_h, Q_h)$

$$A(\boldsymbol{u},\boldsymbol{v}_h) + b(p,\boldsymbol{v}_h) + A_{\mathrm{GLS}}((\boldsymbol{u},p),(\boldsymbol{v}_h,q_h)) = L(\boldsymbol{v}_h) \tag{4a}$$

$$b(q_h,\boldsymbol{u}) = 0 \tag{4b}$$

$$A(\boldsymbol{u},\boldsymbol{v}_h) = \int_{\bar{\Omega}} (\boldsymbol{u}\cdot\nabla\boldsymbol{u})\cdot\boldsymbol{v}_h + \nu\nabla\boldsymbol{u}\cdot\nabla\boldsymbol{v}_h \, dV \tag{4c}$$

$$b(p,\boldsymbol{v}_h) = \int_{\bar{\Omega}} \frac{p}{\rho}\cdot(\nabla\cdot\boldsymbol{v}_h)\, dV \tag{4d}$$

$$L(\boldsymbol{v}_h) = \int_{\bar{\Omega}} \overline{\nabla p}\cdot\boldsymbol{v}_h \, dV \tag{4e}$$

$$A_{\mathrm{GLS}}((\boldsymbol{u},p),(\boldsymbol{v}_h,q_h)) = \int_{\bar{\Omega}_{\mathrm{f}}} \tau_{\mathrm{NS}}\mathcal{R}(\boldsymbol{u},p)\cdot\mathcal{R}(\boldsymbol{v}_h,q_h)\, dV \tag{4f}$$

where Equation 4f represents the GLS term, with

$$\tau_{\mathrm{NS}} = \beta_{\mathrm{NS}}\left[\frac{4||\boldsymbol{u}||}{h^2} + \left(\frac{4}{h^2 Re}\right)^2\right]^{-1/2} , \tag{5}$$

where h is the local element size and $\beta_{\mathrm{NS}} = 0.75$ for all problems solved in this work, and the parameters used to calculate Re are given in Table 1.

2.2.2 Macroscale momentum transfer

On the macroscale, the homogenized properties of each unit cell are used to solve the global momentum transfer problem. In particular, the aim is to find each $\overline{\nabla p}^{i,j}\ \forall i \in [1,N_x], \forall j \in [1,N_y]$ such that

$$\bar{\epsilon}_q = \bar{q}^{\mathrm{x-}} + \bar{q}^{\mathrm{y-}} - \bar{q}^{\mathrm{x+}} - \bar{q}^{\mathrm{y+}} = 0 \tag{6a}$$

$$\bar{\boldsymbol{q}}^{i,j} = -\boldsymbol{k}^{i,j}(\chi)\overline{\nabla p}^{i,j} \tag{6b}$$

$$\bar{q}_x^{1,j} = Q^{\mathrm{in}} \tag{6c}$$

$$p^{i+1,j} = \bar{l}(\nabla_x p)^{i,j} + p^{i,j} \tag{6d}$$

$$p^{i,j+1} = \bar{l}(\nabla_y p)^{i,j} + p^{i,j} \tag{6e}$$

$$p = 0 \text{ on } \Gamma^{\mathrm{x+}}, \tag{6f}$$

where Equation 6b couples the micro- and macroscales through the homogenized permeability matrix $\boldsymbol{k} : \chi \rightarrow \mathbb{R}^{2\times2} = \left(\begin{smallmatrix} k_{xx} & k_{xy} \\ k_{xy} & k_{yy}\end{smallmatrix}\right)$ and $\bar{\boldsymbol{q}}^{i,j}$ represents the flux vector across $\bar{\Omega}^{i,j}$. The optimization framework that will be presented in Section 3 controls χ to modify each $\bar{\Gamma}_{\mathrm{I}}$ and by extension, $\boldsymbol{k}$, thereby influencing the global momentum and heat transfer. To solve Equations 6a-6f, an iterative method is employed to incrementally update the cell-wise pressures, $p^{i,j}$, until $\bar{\epsilon}_q^{i,j} \rightarrow \delta^{\mathrm{f}}$ is achieved, where $\delta^{\mathrm{f}} = 10^{-8}$ is the chosen numerical tolerance. Lastly, the local coupling matrix, $\boldsymbol{k}^{i,j}$ needed in Equation 6b, is derived using a finite difference scheme, requiring two solutions of Equation 2 for xy-symmetric geometries and six solutions for non-symmetric geometries.

Table 1 Definitions of non-dimensional parameters

Parameter	Definition
l	L/N_x
$\bar{L}$	l
$A_{\mathrm{in,M}}$	$N_y l h_{\mathrm{f}}$
$u_{\mathrm{in,M}}$	$Q_{\mathrm{in}} A_{\mathrm{in,M}}$
$\bar{U}$	$u/u_{\mathrm{in,M}}$
Re	$\rho\bar{U}\bar{L}/\mu$

2.2.3 Permeability surrogate model

The efficiency of solving the momentum transfer problem in a multiscale setting is derived from the fact that Equation 2 can be solved independently on each $\bar{\Omega}^{i,j}_{\mathrm{f}}$, enabling ensemble parallelism. However, the evaluation of the function $\boldsymbol{k}$ presents a computational bottleneck, as two or six simulations are needed for each evaluation of χ, depending on the level of symmetry in the microstructure. To improve the efficiency of the multiscale momentum transfer solver, a surrogate model is leveraged to represent the permeability function using local simplex interpolation, $\tilde{\boldsymbol{k}} : \chi \rightarrow \mathbb{R}^{2\times 2}$ which approximates $\boldsymbol{k}(\chi)$. To perform the local interpolation, each design variable dimension is discretized into $n^i_\chi, \forall i \in n_d$ levels, which forms the vertices of the simplex space. To approximate $\boldsymbol{k}(\chi)$, the set $\mathcal{I} = \{\boldsymbol{k}(\tilde{\chi}^1), \dots, k(\tilde{\chi}^{n_d+1})\}$ is instead evaluated, where $\tilde{\chi}^i$ are the vertices of the (n_d+1)-dimensional simplex containing χ, the first time the evaluation of $\boldsymbol{k}(\chi)$ is required in a previously unexamined simplex, and $\mathcal{I}$ is stored. Any subsequent evaluations of $\boldsymbol{k}$ that are required with a previously addressed χ simplex are then locally interpolated using the stored values of $\mathcal{I}$. For further details regarding the implementation of the simplex method, readers are referred to Murphy et al. (2021).

2.2.4 Validation

The multiscale momentum transfer model is validated using explicit NS solutions solved on the single scale. Four validation cases (heterogeneous distributions of χ), $B^{\mathbf{A}}$, $B^{\mathbf{B}}$, $B^{\mathbf{C}}$ and $B^{\mathbf{D}}$ are created, as shown in Fig. 11. A detailed discussion of the validation setup and results is provided in Appendix B. Excellent agreement is found between the single and multiscale solutions with less than 10% error across all four test cases. Furthermore, the multiscale solutions are obtained at significantly lower computational cost (both in terms of time and memory requirements), which lends itself to design optimization.

2.3 Multiscale heat transfer

Analogous to the incompatibility of homogenizing stress in elasticity, in heat transfer problems, $T \in \bar{\Omega}$ is not well suited to homogenization due to its highly local nature. To overcome this limitation, a subspace multiscale advection–diffusion (AD) model, that does not rely on homogenization, is developed.

2.3.1 Microscale heat transfer

On the microscale, the heat transport is modelled using an AD system, where the aim is to find $T : \bar{\Omega} \rightarrow \mathbb{R}$ such that

$$\mathcal{L}_{\mathrm{s}} \equiv -\kappa_{\mathrm{s}} \nabla^2 T = 0 \text{ in } \bar{\Omega}_{\mathrm{s}} \tag{7a}$$

$$\mathcal{L}_{\mathrm{f}} \equiv -\kappa_{\mathrm{f}} \nabla^2 T + \rho c_p \boldsymbol{u} \cdot \nabla T = 0 \text{ in } \bar{\Omega}_{\mathrm{f}} \tag{7b}$$

$$T = T_D \text{ on } \bar{\Gamma}_D \tag{7c}$$

$$\kappa(\bar{\mathbf{x}}) \nabla T \cdot \boldsymbol{n} = g \text{ on } \bar{\Gamma}_N \quad , \tag{7d}$$

where $\bar{\Gamma}_D$ and $\bar{\Gamma}_N$ represent Dirichlet and Neumann boundaries, respectively, with $\bar{\Gamma}_D \subset \partial\bar{\Omega}$, $\bar{\Gamma}_N \subset \partial\bar{\Omega}$ and $\bar{\Gamma}_D \cap \bar{\Gamma}_N = \emptyset$. The specific values of T_D and g are derived from the macroscale and will be defined in Section 2.3.2. Equation 7 is discretized using second-order Lagrange elements and define the FE space, Q_h, as

$$\begin{aligned} Q_h = &\{q_h \in [H^1_0(\bar{\Omega})]^d : q_h|_K \in [\mathbb{P}_2(K)]^d \ \forall K \in \mathcal{T}_h \mid \\ & q_h = T_D \text{ on } \bar{\Gamma}_{\mathrm{I}}\} \end{aligned} \tag{8}$$

where $\forall\, q_h \in Q_h$ the aim is to find T such that

$$\begin{aligned} A(T, q_h) + A_{\mathrm{GLS}}(T, q_h) = &\ 0 \\ A(T, q) = &\int_{\bar{\Omega}_{\mathrm{f}}} \rho c_p (\boldsymbol{u} \cdot \nabla T) q_h + \kappa_{\mathrm{f}} \nabla T \cdot \nabla q_h \, dV \\ &+ \int_{\bar{\Omega}_{\mathrm{s}}} \kappa_{\mathrm{s}} \nabla T \cdot \nabla q_h \, dV \\ &- \int_{\bar{\Gamma}_N} g \, ds \\ A_{\mathrm{GLS}}(T, q) = &\int_{\bar{\Omega}_{\mathrm{f}}} \tau_{\mathrm{ADC}} \mathcal{L}_{\mathrm{f}}(T) \cdot \mathcal{L}_{\mathrm{f}}(q_h) \, dV \end{aligned} \quad , \tag{9a}$$

where it should be noted that no periodicity constraints are imposed in Q_h, since, unlike the microscale momentum problem, in general it is expected that $T(x, y, z) \neq T(x + l, y, z)$ and $T(x, y, z) \neq T(x, y + l, z)$. In addition, a GLS term, A_{GLS}, is included to address the numerical instabilities in convection-dominated solutions of Equation 7 found when employing traditional continuous Galerkin formulations. The GLS scaling parameter, τ_{ADC} , is defined as

$$\tau_{\mathrm{ADC}} = \alpha_{\mathrm{GLS}} \left[\frac{4||\boldsymbol{u}||}{h^2} + \left(\frac{4}{\bar{Pe} h^2} \right)^2 \right]^{-1/2} , \tag{10}$$

where $\alpha_{\mathrm{GLS}} = 0.1$ is used throughout this work and $\bar{Pe} = \frac{l||\boldsymbol{u}||_2}{\alpha}$ is the homogenized, cell-wise, Péclet number and $\boldsymbol{u} \in \bar{\Omega}$. Cell-wise Péclet numbers are utilized for scaling the

GLS term to account for large variations in cell-wise fluid velocities which would not be accounted for with a single, global, Peclet number.

2.3.2 Macroscale heat transfer

Unlike the momentum transfer problem outlined in Section 2.2.2, classical homogenization theory is not applied to tackle the micro-macroscale heat transfer problem. Instead an iterative downwind flux matching scheme is developed, circumventing the use of homogenized thermal properties, enabling accurate thermal modelling across $\bar{\Gamma}_{\mathrm{I}}$.

The flux across a pair of adjacent cells is denoted $\bar{\Omega}^{+}$ and $\bar{\Omega}^{-}$, as shown in Fig. 5. The temperature on the outlet, $\bar{\Gamma}^{+} \in \partial\bar{\Omega}^{-}$, and inlet, $\bar{\Gamma}^{-} \in \partial\bar{\Omega}^{+}$, should, by definition, be identical. In addition, the flux across these boundaries must be identical in magnitude, i.e. $\bar{q}^{\text{out}} = \bar{q}^{\text{in}}$, where $\bar{q}^{\text{out}} = \kappa \boldsymbol{n} \cdot \nabla T(\bar{\mathbf{x}})$, $\bar{\mathbf{x}} \in \bar{\Gamma}^{+}$ and $\bar{q}^{\text{in}} = -\kappa \boldsymbol{n} \cdot \nabla T(\bar{\mathbf{x}})$, $\bar{\mathbf{x}} \in \bar{\Gamma}^{-}$. The macroscale heat transfer problem is defined as a global boundary value problem, where the aim is to find $T \in \bar{\Omega}^{i,j}\ \forall i \in [1, N_x], \forall j \in [1, N_y]$ such that

$$\begin{aligned}
&T^{i,j}|_{\bar{\Gamma}^{\text{x+}}} = T^{i+1,j}|_{\bar{\Gamma}^{\text{x-}}}, \forall i \in [1, N_x - 1], \forall j \in [1, N_y]\\
&T^{i,j}|_{\bar{\Gamma}^{\text{z+}}} = T^{i,j+1}|_{\bar{\Gamma}^{\text{z-}}}, \forall i \in [1, N_x], \forall j \in [1, N_y - 1]\\
&\kappa(\bar{\mathbf{x}})\boldsymbol{n} \cdot \nabla T|_{\bar{\Gamma}^{i,j,\text{x-}}} = -\kappa(\bar{\mathbf{x}})\boldsymbol{n} \cdot \nabla T|_{\bar{\Gamma}^{i+1,j,\text{x+}}},\\
&\qquad\qquad\qquad \forall i \in [1, N_x - 1], \forall j \in [1, N_y]\\
&\kappa(\bar{\mathbf{x}})\boldsymbol{n} \cdot \nabla T|_{\bar{\Gamma}^{i,j,\text{z-}}} = -\kappa(\bar{\mathbf{x}})\boldsymbol{n} \cdot \nabla T|_{\bar{\Gamma}^{i,j+1,\text{z+}}},\\
&\qquad\qquad\qquad \forall i \in [1, N_x], \forall j \in [1, N_y - 1]\\
&T = T^{\text{in}} \text{ on } \Gamma_{\text{f}}^{\text{x-}}\\
&\kappa \boldsymbol{n} \cdot \nabla T = q_{\text{in}} \text{ on } \Gamma^{\text{z-}} \cup \Gamma^{\text{z+}}\\
&\boldsymbol{n} \cdot \nabla T = 0 \text{ on } \partial\Omega \setminus (\Gamma^{\text{z-}} \cup \Gamma^{\text{z+}} \cup \Gamma_{\text{f}}^{\text{x-}})
\end{aligned} \tag{11}$$

where q_{in} is the external heat flux applied to the heat sink and T^{in} is the temperature at the fluid inlet boundary, $\Gamma_{\text{f}}^{\text{x-}}$. To solve this system, an iterative downwind solver is developed which sequentially updates $T \in \partial\bar{\Omega}$ using temperature and flux information from adjacent cells. The scheme is outlined for a single row multiscale system (i.e. $N_y = 1$) in Fig. 6. In the first iteration, the solver starts with the most upstream cell, $\bar{\Omega}^{0}$, and imposes a Dirichlet temperature boundary on the global inlet, $\bar{\Gamma}^{0,\text{x-}}$, this represents the fixed inlet fluid temperature. On $\bar{\Gamma}^{0,\text{x+}}$, a Neumann boundary, $\bar{q}_0^{0,+} = 0$, is weakly imposed, where the subscript refers to the iteration number. The input heat flux q^{in} is applied on $\bar{\Gamma}^{0,\text{ y-}}$ and all other surfaces are assumed to be adiabatic walls (i.e. $\bar{q} = 0$). Equation 7 is solved using the aforementioned BCs to find $T \in \bar{\Omega}^{0}$. The outlet temperature $T_0^{0,\text{x+}}$ is then projected as a Dirichlet boundary onto the inlet of the adjacent cell, $T^{1,\text{x-}} \in \bar{\Omega}^{1}$, ensuring consistent temperatures on the coincident boundary, with identical (Neumann) boundary conditions on all the other boundaries. Equation 7 is then used to solve $T^{1} \in \bar{\Omega}^{1}$. This process is repeated for the remaining unit cells. In subsequent iterations, the same procedure is repeated, but rather than imposing $\bar{q}^{0,\text{x+}} = 0$ on each $\Gamma^{\text{x+}}$, in the k-th iteration $\bar{q}_k^{\text{i,x+}} \leftarrow -\bar{q}_{k-1}^{\text{i+1,x-}}\ \forall i \in [1, N_x - 1]$ is set, with $T^{i,\text{x+}} \rightarrow T^{i+1,\text{x-}}$. Equation 7 is then repeatedly solved with updated boundary conditions for each $\bar{\Omega}$, until macroscale convergence is reached.

For the '2D' case ($N_x > 1, N_y > 1$), a similar procedure is employed; however, since the flow direction is not known a priori, the direction of the flux/temperature projections based on the local pressure gradient. In particular, flux is projected upstream and temperature is projected downstream, with the intuition that the thermal wake is advected and can be tracked using the direction of $\overline{\nabla p}$. As a result, the distribution of $\overline{\nabla p}$ determines the order in which the cells are solved. To do so, a cell ordering matrix, $D \in \mathbb{N}^{Nx \times Ny}$, is constructed. During the iterative solution produce, Equation 7 is iteratively solved with updated boundary conditions, as outlined in Algorithm 1. Unlike the multiscale system outline in Section 2.2.2, the thermal multiscale system described here no longer involves fully independent solutions on each $\bar{\Omega}$ since cells rely on T or q values from adjacent cells. However, since $\bar{\Omega}^{i,j}$, in the worst-case, requires information from the adjacent cell set $\mathcal{C}^{i,j} = \{\bar{\Omega}^{i-1,j}, \bar{\Omega}^{i+1,j}, \bar{\Omega}^{i,j-1}, \bar{\Omega}^{i+1,j+1}\}$, it is possible to solve Equation 7 on $\bar{\Omega}^{i,j}$ in the k-th iteration, provided cells in $\mathcal{C}_T^{i,j}$ have been solved up to iteration k and $\mathcal{C}_q^{i,j}$ contain solutions up to iteration $k - 1$, where $\mathcal{C}_T^{i,j}$ and $\mathcal{C}_q^{i,j}$ are the adjacent cell sets which provide temperature and flux boundaries, respectively, and $\mathcal{C}^{i,j} = \mathcal{C}_T^{i,j} \cup \mathcal{S}_q^{i,j}$, $\mathcal{C}_T^{i,j} \cap \mathcal{C}_q^{i,j} = \emptyset$. As a result, it is possible to concurrently solve Equation 7 on multiple $\bar{\Omega}$'s provided their respective adjacent cells, $\mathcal{C}$ contain solutions up to k and $k - 1$ for temperature and flux propagating adjacent cells, respectively. Relaxation to the q updates, such that $q_{k+1} = \eta q^{\pm} + (1 - \eta)q_{k-1}$, where $q^{\pm}$ is the flux from the adjacent cell and $\eta = 0.75$ is a relaxation parameter, chosen to avoid spurious flux oscillations in the solver since q_{k+1} is weakly imposed.

The iterative solution procedure is repeated until convergence is achieved on the macroscale, which is based on the mean outlet and plate ($\Gamma^{\text{y}-}$) temperatures

$$\begin{aligned}
\epsilon^{\text{out}} &= \max\left\{(|\bar{T}_k^{\text{out}} - \bar{T}_n^{\text{out}}|)/\bar{T}_n^{\text{out}} :\right.\\
&\qquad\qquad \left.\forall k \in [n-1, n-n_c], n > n_c\right\}\\
\epsilon^{\text{plate}} &= \max\left\{(|\bar{T}_k^{\text{plate}} - \bar{T}_n^{\text{plate}}|)/\bar{T}_n^{\text{plate}} :\right.\\
&\qquad\qquad \left.\forall k \in [n-1, n-n_c], n > n_c\right\}
\end{aligned}, \tag{12}$$

where n is the current iteration number, $n_c = 3$ is the number of iterations over which the convergence check is performed,

$\bar{T}_n^{\text{out}}$ and $\bar{T}_n^{\text{plate}}$ are mean outlet and plate temperatures, defined as

$$\begin{aligned}\bar{T}^{\text{out}} &= \frac{1}{N_y}\sum_{j=1}^{N_y} T|_{\bar{\Gamma}^{x+}},\; T \in \bar{\Omega}^{N_x,j}\\ \bar{T}^{\text{plate}} &= \frac{1}{N}\sum_{i=1,j=1}^{N_x,N_y} T|_{\bar{\Gamma}^{z-}},\; T \in \bar{\Omega}^{i,j}\end{aligned} \tag{13}$$

For all examples presented in this work, the tolerances, $\epsilon^{\text{TOL}} = 0.05$, is applied for ϵ^{out} and ϵ^{plate}.

In Fig. 7, a heat sink array containing 1600 3D pin-fin microstructures, graded to form two channels, is presented. A homogeneous plate heat flux is applied on $\bar{z} = 0$, as shown in Fig. 9(a). The output of the AD solver on this domain is shown in Fig. 8. The example was solved a single node of a cluster containing AMD EPYC 7742 processors, utilizing a total of 127 cores. This example was solved in 92 min, which includes meshing and solving for momentum transfer and the AD system with ≈50 million degrees of freedom (DoF). In the figure, it is possible to clearly observe the influence of the microstructure grading on the temperature distribution as the two channels with $\chi = 0$ creates pressure gradients which drives the flow towards the centre of the domain, which directs the thermal wakes formed by the pin-fins. This example highlights the efficacy and efficiency of the solver as it is able to capture complex momentum and thermal transport induced by variations in local geometry without the need for prohibitively expensive single-scale meshing and simulations. In Section 2.4, a quantitative analysis is provided to validate the accuracy and performance of the multiscale AD solver relative to single-scale simulations.

2.4 Model validation

Explicit 3D single-scale AD simulations are utilized to validate the multiscale heat transfer model. The same four $\chi \in \mathcal{R}^{1\times8\times8}$ distributions used for the multiscale momentum solver validation are utilized again. The validation and associated discussions are presented in C.1. From the validation study, it is clear that the multiscale AD solver is able to accurately capture thermal behaviour in the macroscale domain, with an average heat flux error of 10% relative to the single-scale model. The multiscale framework also demonstrates significant computational advantages, solving problems 2.5 times faster, with lower memory usage compared to the single-scale model. Additionally, weak scaling tests reveal that, as expected, the single-scale model performance degrades with increasing unit cells due to solver limitations and growing overheads (e.g. meshing), and the multiscale AD solver's semi-decoupled nature enables efficient parallel processing of unit cells. This parallel capability yields faster computations even for modest domains of 64 unit cells. The improved scaling is particularly valuable for optimization, as incorporating more unit cells enhances momentum transport predictions and enables finer local control over fluid and thermal behaviour. As shown in Section 4, the multiscale model is leveraged to successfully optimize domains with 400 unit cells, a task that would be impractical with the single-scale approach, given its computational demands due to poor scaling performance.

Table 2 Physical parameters used for heat sink modelling

Parameter	Value
L, H, W	60 mm, 7.5 mm, 60 mm
$Q_{x,\text{in}}$	$1.2 \times 10^3\ \text{mm}^3/\text{s}$
p_{out}	0 Pa
Re	4.19
q_{in}	$17\ \text{kW}/\text{m}^2$
T^{in}	0°C

3 Multiscale heat sink optimization

Design optimization is utilized to tailor the distribution of microscale variables, $\chi^{i,j}\ \forall\, i,j \in [N_x, N_y]$. The optimization design variables are defined as the vector of normalized radii $\chi \in \mathcal{R}^{\tilde{N}_x\times\tilde{N}_y}$, where $\tilde{N}_x$ and $\tilde{N}_y$ are defined in Section 3.2.

3.1 Bayesian optimization

The optimization problem is formulated as a heat flux maximization subject to a maximum pressure drop across the heat sink

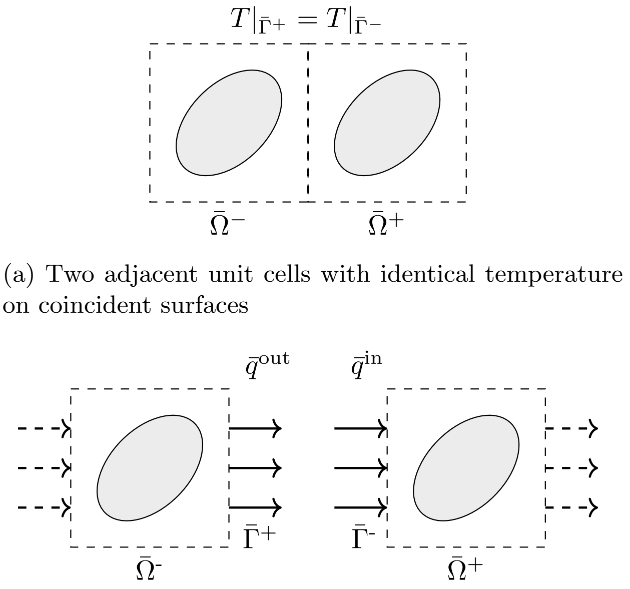
$$\max_{\chi\in\mathcal{S}}\; J = \alpha_J \mathcal{H}(\chi) \tag{14a}$$

$$\text{s.t.}\;\; G = \frac{dp}{dx}(\chi) \leq \Delta P_{\max} \tag{14b}$$

$$\mathcal{S} = \{\chi \in \mathbb{R}^{n_{\text{DV}}\times N_x\times N_y} \,:\, \chi_{\min} \leq \chi \leq \chi_{\max}\} \tag{14c}$$

Table 3 Material properties used in Ω_{f} and Ω_{s}

Parameter	Ω_{f}	Ω_{s}
Material	Ethylene Glycol 50/50	Aluminium 3003
ρ	$1090.0\ \text{kg}/\text{m}^3$	$2730.0\ \text{kg}/\text{m}^3$
c_p	3214.0 J/kgK	943.13 J/kgK
κ	0.258 W/mK	177.3 W/mK
μ	0.0065 Pa s	-



(a) Two adjacent unit cells with identical temperature on coincident surfaces

(b) Heat flux across adjacent unit cells

Fig. 5 Heat flux across adjacent $\bar{\Omega}$

where Δp_{max} is the maximum allowable pressure drop across the heat sink, with the pressure loss

$$\frac{dp}{dx}(\chi) = \frac{\sum_{j=1}^{N_y} p^{1,j} - \sum_{j=1}^{N_y} p^{Nx,j}}{N_y L} \tag{15}$$

and $\mathcal{S}$ is the feasible set of designs, χ_{min} and χ_{max} are design constraints and α_J is used to scale the heat flux values such that $\mathcal{O}(J) \approx \mathcal{O}(G)$ to better condition the optimization. Due to cross cell dependencies introduced during the temperature and flux projections in the multiscale solver, design derivatives $(\frac{dJ}{d\chi})$ are not easily defined. As a result, Bayesian optimization is employed to tailor χ. Bayesian optimization (Jones et al. 1998) has been utilized for a wide range of problems from lattice-based heat sink optimization (Shimoyama and Komiya 2022) to hyper-parameter tuning for machine learning (Snoek et al. 2012) and is particularly well suited for problems involving costly objective functions, as is the case here, and does not require gradient information.

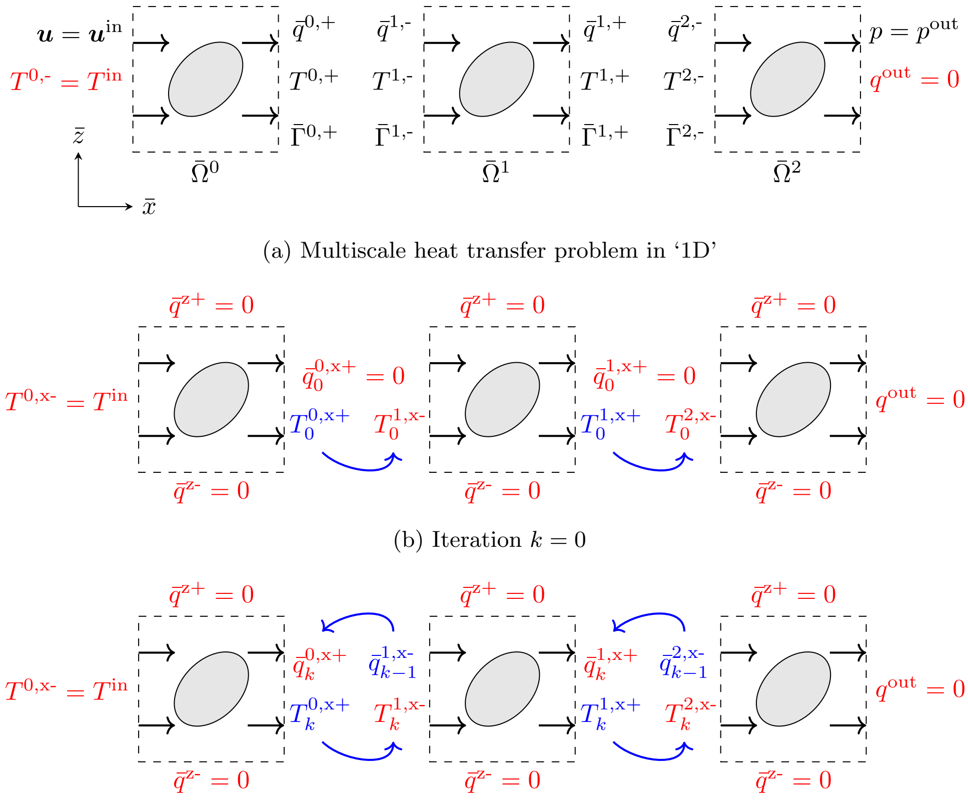
For a given objective, $J(\chi)$, during the n-th optimization iteration, Bayesian optimization uses the set of all past objective evaluations, $\mathcal{F} = \{J(\chi^1), \ldots, J(\chi^n)\}$, to construct a posterior, $\tilde{J}$, typically using Krige (1951). The posterior represents a cheap surrogate of J, and importantly, is equipped with a mean and variance, conditioned on $\mathcal{F}$. The selection of a new design point, χ^{n+1}, is governed by a chosen utility function (also referred to as the acquisition function) and represents a trade-off between exploration and exploitation. In this work, the expected improvement (EI) (Jones et al. 1998) is utilized, which as the name suggests, selects new design points to maximize the expected improvement relative to the current optimum χ^* and a hyper-parameter, ξ, which controls the level of exploration (increasing ξ leads to increased exploration). To encourage exploration and design convergence during later iterations, a continuation scheme is applied to decay ξ over the course of the optimization according to $\xi^{n+1} \leftarrow \eta\xi^n$, where $\eta = (\xi_{n_{\text{BO}}}/\xi_0)^{1/n_{\text{BO}}}$ is the decay parameter. The derivations of the posterior and acquisition functions are outside of the scope of this work, for discussions and derivations of these methods, the reader is referred to Jones et al. (1998). Lastly, to perform Bayesian optimization, the Python implementation provided in Nogueira (2014) is utilized.

3.2 Problem setup

The design space is defined as $\mathcal{S} \in \mathbb{R}^{1\times20\times20}$ with the physical parameters given in Table 2 and material properties shown in Table 3. Bayesian optimization is known to struggle with large design spaces. To overcome this limitation, several modifications are applied to the optimization formulation. First, symmetry about the $y = W/2$ plane is imposed, which provides the added benefit of reducing the computation cost in solving $\mathcal{H}(\chi)$ and $G(\chi)$. Secondly, a coarse design space $\tilde{\mathcal{S}} \in \mathbb{R}^{\tilde{N}_x\times\tilde{N}_y}$ is utilized, where $\tilde{N}_x < N_x$ and $\tilde{N}_y < N_y$, and the interpolator $\mathcal{I} : \tilde{\mathcal{S}} \to \mathcal{S}$ utilizes C1 continuous bi-variate splines to map the coarse design onto the original design space. The total number of design variables used in the optimization is given as $N_{\text{opt}} = \tilde{N}x \times \tilde{N}y$. The design space is defined using $N_x = 20$, $N_y = 10$, $\tilde{N}_x = 8$, $\tilde{N}_y = 5$ and $N_{\text{opt}} = 40$ for all examples presented in this work. As well as reducing the number of design variables, the use of $\mathcal{I}$ improves the accuracy of the multiscale momentum solver since it enables smoother variations in χ. To further reduce the computational expense of evaluations of J, symmetry conditions are applied about the $z = W/2$ plane. The optimization problems are initialized with $n_r = 30$ pseudo-random initialization of χ to explore $\tilde{\mathcal{S}}$. To enable a fair comparison between different optimizations, the same pseudo-random points and random seed are used for all the problems, followed by n_{BO} iterations of Bayesian optimization. The parameters used for the results shown in the proceeding section is outlined in Table 4.

The inclusion of inequality constraints in Bayesian optimization, such as the pressure loss constraint imposed in Equation 14, is an on-going research topic (Bhunia et al. 2019). However, since $G(\chi)$ is inexpensive to evaluate in

Fig. 6 Iterative multiscale downwind heat transfer scheme for a row of cells. Parameters in red indicate input boundary conditions and parameters in blue are outputs of Equation 7



(a) Multiscale heat transfer problem in '1D'

(b) Iteration $k = 0$

(c) Iteration $k > 0$

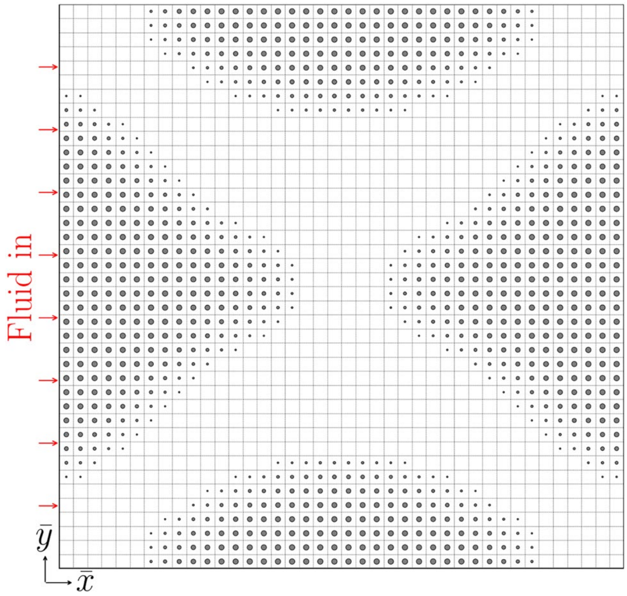


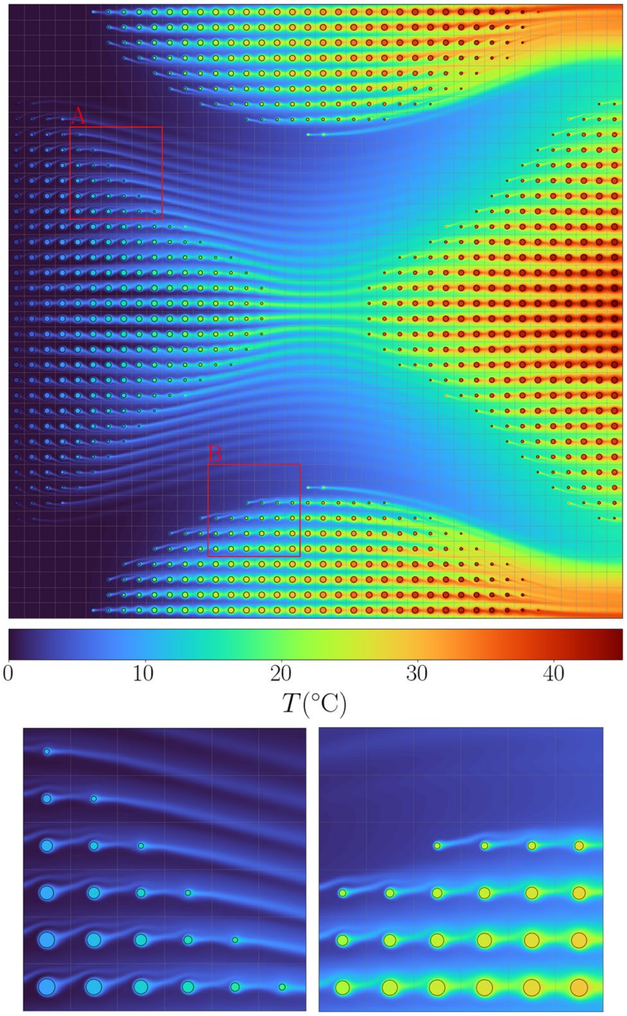
Fig. 7 Cross-section of a 40 × 40 graded multiscale heat sink array of microstructures with two channels

this work, the optimization formulation is augmented as an un-constrained problem

$$\max_{\chi \in \tilde{S}} \; \mathcal{L} = \begin{cases} J(\chi), & \text{if } G(\chi) \leq \Delta p_{\max} \\ \delta_J, & \text{otherwise} \end{cases} \tag{16a}$$

where $\delta_J = \alpha^*(\chi^* - \chi_{\min}) + \chi_{\min}$ with $\alpha^* = 0.75$ and $\chi^*$and $\chi_{\min}$ are the current maximum and minimum evaluations of J, respectively. This modification not only removes the inequality constraint, but it also lowers the computational expense of the optimization, since $G(\chi)$ is inexpensive to evaluate due to the interpolation scheme outlined in Section 2.2.3 and ensures J is only evaluated for designs that satisfy the pressure drop constraint. Whilst it is possible to set δ_J to an arbitrary small value, this form is adopted since it avoids large discontinuities in the search space, which were found to hinder the GP regression in the Bayesian optimization. It is common to construct a weighted multi-objective function, $\mathcal{L} = \alpha J + (1 - \alpha)G$. This form is not utilized since it requires evaluations of J and G in each iteration, regardless of the feasibility of the current design, requires tuning of the weighting parameter (α) and does not offer explicit control of the maximum pressure loss.

Two optimization cases are investigated with different heat flux distributions on $\bar{z} = 0$: a homogeneous distribution



(a) Close-up of highlighted region A (b) Close-up of highlighted region B

Fig. 8 Temperature distribution along $\bar{z} = 0.5$ plane for a 40×40 multiscale heat sink array of microstructures with two channels

Table 4 Parameters used in Bayesian optimization

Parameter	Value
N_x, N_y	20, 10
$\tilde{N}_x$, $\tilde{N}_y$	8, 5
N_{opt}	40
χ_{min}, χ_{max}	0, 0.2
n_r, n_{BO}	30, 100
ξ_0, ξ_K	0.15, 0.005
α_J	10^6
Δp_{max}	7.89

(Case **A**) and an in-homogeneous heat flux distribution (Case **B**) to better represent a localized heat source, such as a chip, as shown in Fig. 9(b). Prior to design optimization benchmark, designs with homogeneous pin-fin radii and identical heat flux distributions to optimization cases **A** and **B** are established to serve as a comparison for the performance of the optimized designs. The benchmarks use homogeneous $\chi = 0.075$ and $\chi = 0.2$ pin-fin radii, which are denoted B^{A} and B^{B}, respectively, where the pressure loss from the $\chi = 0.075$ design is used as the target maximum pressure loss, $\Delta p_{\text{max}} = 7.89$ (i.e. $G(\chi = 0.075) = 7.89$).

4 Results

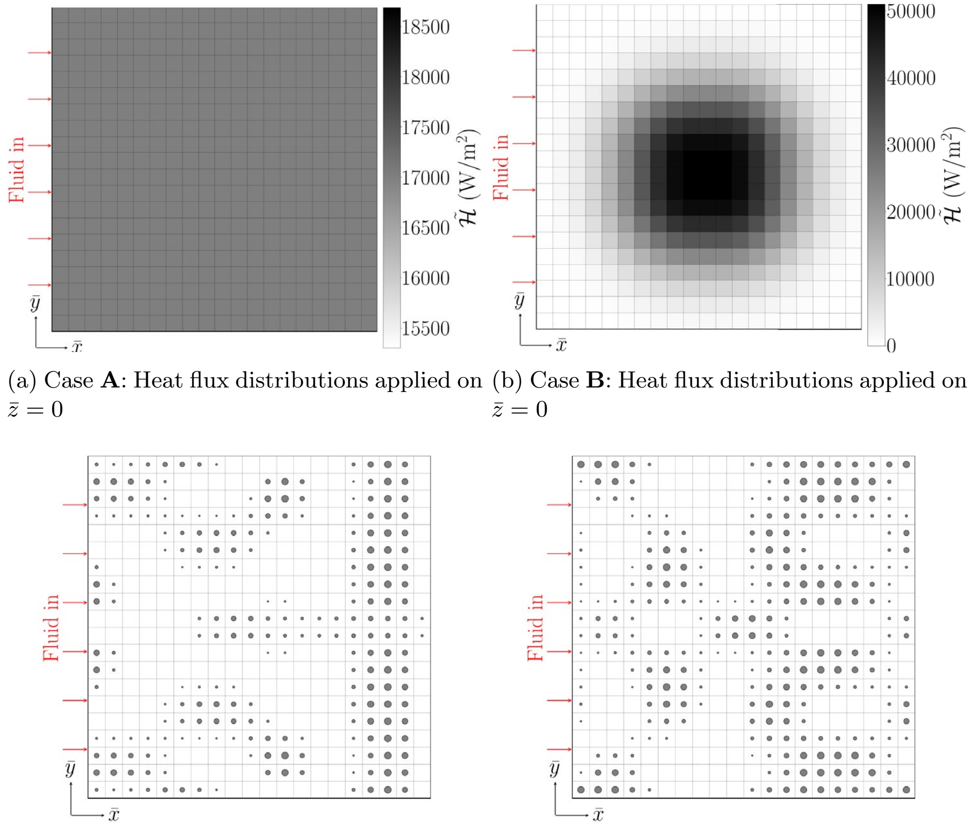
Each optimization case was completed in less than 80 h, with the average evaluation of $J(\chi)$ taking $t_{\text{iter}} = 47$ min using 127 concurrent processors to solve the microscale momentum transfer problems and 20 parallel processors to solve the microscale AD systems, where the macroscale AD system contains ≈ 24 million DoFs, depending on the exact pin-fin configuration. It should also be noted the total optimization time is less than $(n_{\text{BO}} + n_{\text{r}}) \times t_{\text{iter}} =$ since J is not evaluated for infeasible designs.

The performance metrics for the optimized designs and corresponding benchmarks are shown in Table 5. The optimized pin-fin designs significantly outperform the homogeneous benchmark configurations in both cases. For Case **A**, a 75% increase in heat transfer is achieved whilst respecting the pressure loss constraint set by $B^{\text{A}}_{\chi=0.075}$. In Case **B,** a significantly larger 558% increase in heat transfer is achieved. Since the pins located in regions of low applied heat flux provide negligible contributions to the overall heat transfer, design optimization results in a drastic improvements in the total heat transfer in comparison to the Case **A**. The optimized pin-fin distributions are shown in Fig. 9, and the resulting temperature profiles are shown in Fig. 10 for Case **A** and **B**, respectively. The optimization balances two competing factors: pressure loss management and heat extraction, leading to heterogeneous pin-fin distributions. These create locally varying pressure gradients which are visible through the changing directions of the thermal wakes formed behind the pins as shown in Fig. 10.

The optimized design in Case **A** (with homogeneous heat flux), shown in Fig. 9(a), contains several design features where the influence of these competing factors are apparent. For instance, the upstream (left) region features, on average, thinner pins to manage pressure losses and reduce the formation of thermal wakes as shown in the temperature plots in Fig. 10(a). Near the outlet (right), a band of larger-radius pins maximize heat extraction. Between the inlet and outlet, four distinct flow channels (two in each symmetric domain) are formed using thinner or removed pins, creating high advection regions. These channels guide cooler

Table 5 Performance metrics for benchmark and optimized pin-fin distributions

Parameter	Case **A**			Case **B**		
	$B^{\mathrm{A}}_{\chi=0.075}$	$B^{\mathrm{A}}_{\chi=0.2}$	Optimization	$B^{\mathrm{B}}_{\chi=0.075}$	$B^{\mathrm{B}}_{\chi=0.2}$	Optimization
J	3.21	3.36	**5.64**	0.17	0.32	**1.10**
G	7.89	15.55	**7.22**	7.89	15.55	**7.81**



(a) Case **A**: Heat flux distributions applied on $\bar{z} = 0$

(b) Case **B**: Heat flux distributions applied on $\bar{z} = 0$

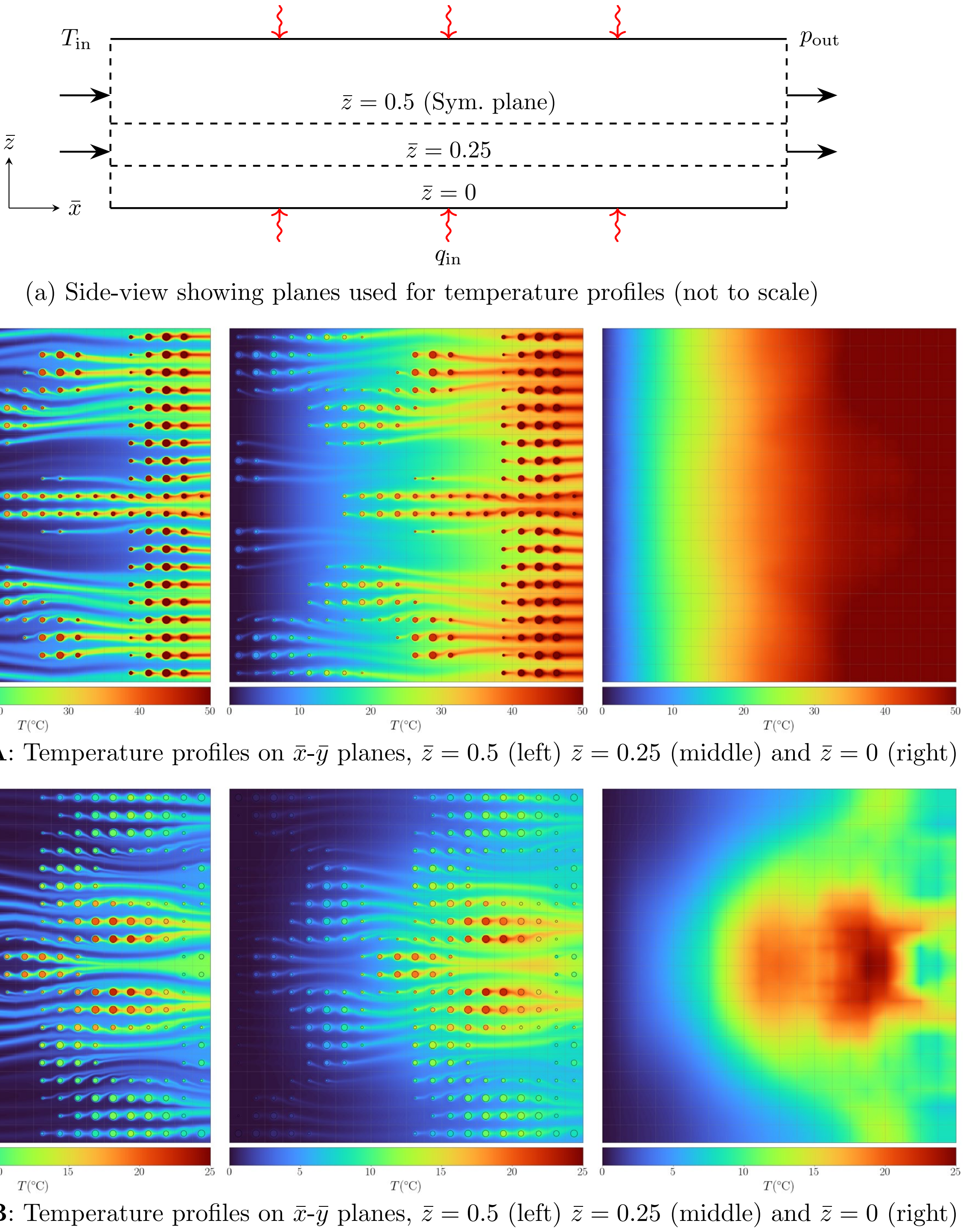
(c) Case **A**: Optimized pin-fin distribution

(d) Case **B**: Optimized pin-fin distribution

Fig. 9 Applied heat flux distributions (top row) and resulting optimized pin-fin distributions (bottom row)

fluid onto the band of thicker pins at the outlet, minimizing the impact of thermal wakes formed upstream. This is particularly evidenced by the arrangement of pins above and below the centre of the domain, which is arranged to diffuse the cooler upstream fluid before it reaches the outlet, improving the heat sink efficiency. The optimized design in Case **A** respects the pressure loss constraint; however, the slack between the constraint ($\Delta p_{\max} = 7.89$) and the design (7.22) indicates a limitation in the optimization approach. The Bayesian optimization framework is overly conservative near constraint boundaries due to the GP regression resolution which underestimates the potential of designs near the constraint boundary due to the applied augmented objective function, shown in Equation 16. To overcome this limitation, it may be possible to link the penalty factor, δ_{J}, applied to infeasible designs, to the GP length scale. Alternatively, the acquisition function can be directly constrained forming a constrained improvement function. Future work will explore improved constraint handling techniques to enable better search of the constrained design space.

For Case **B**, an in-homogeneous, localized, heat flux distribution was applied on $\bar{z} = 0$ with a concentration of heat flux near the centre of the domain, radially decaying to the minimum value near the boundaries of the domain. The optimized pin-fin distribution is shown in Fig. 9(b) and contains distinct design features that were not present in Case **A**. As expected, pin-fins are concentrated in the central region to maximize heat extraction where the largest external heat flux is applied. Unlike Case **A**, there are no distinct channels in the centre of the domain. Instead, the design directs cooler fluid from the exterior (top and bottom) of the domain towards the centre. The pins near



(a) Side-view showing planes used for temperature profiles (not to scale)

(a) Case **A**: Temperature profiles on $\bar{x}$-$\bar{y}$ planes, $\bar{z} = 0.5$ (left) $\bar{z} = 0.25$ (middle) and $\bar{z} = 0$ (right)

(b) Case **B**: Temperature profiles on $\bar{x}$-$\bar{y}$ planes, $\bar{z} = 0.5$ (left) $\bar{z} = 0.25$ (middle) and $\bar{z} = 0$ (right)

Fig. 10 Heat sink side view showing definition of $\bar{z} = 0.5$, $\bar{z} = 0.25$ and $\bar{z} = 0$ (top row). Temperature profiles on $\bar{x}$- $\bar{y}$ plane along $\bar{z} = 0.5$ (left), $\bar{z} = 0.25$ (middle) and $\bar{z} = 0$ (right) planes for optimized designs Case **A** (middle row) and **B** (bottom row), with flow going from left to right

the outlet are arranged in a checker board pattern, alternating between larger pins and void, where it is interesting to note that empty spaces have been placed upstream of thicker pins to provide cleaner flows to these pins. However, the temperature profile at the outlet indicates untapped performance due to the presence of relatively cold fluid at the outlet, suggesting room for further design improvements. This limitation likely arises due to two factors; the design variable interpolation, $\mathcal{I}$, and the multiscale discretization (i.e. $\mathcal{S}$) utilized. It is conceivable that the optimizer is limited by the choice of $\tilde{\mathcal{S}}$ when attempting to finely control local fluid and thermal flows to better extract heat with a localized heat source. To overcome this, it may be possible to increase the multiscale discretization, i.e. increase the dimensionality of $\mathcal{S}$ or reduce the coarseness of the interpolation grid, i.e. increase the dimensionality of $\tilde{\mathcal{S}}$, with the optimal choice(s) likely driven by the spatial distribution of heat flux applied on $\bar{z} = 0$ and choice of microstructure parametrisation.

5 Conclusion

Heat sinks are fundamental components in modern electronic devices, particularly in CPUs and GPUs. Their design directly impacts system reliability, energy efficiency and overall performance. Whilst structural optimization can enhance heat sink designs, traditional modelling approaches using grid-based, density parametrizations face significant limitations. These include inaccurate wall boundary definitions and optimization results that require substantial post-processing for manufacturing, often compromising performance. Explicit modelling methods, though more accurate, are rarely used due to their prohibitive computational costs, especially for detailed heat sink designs, this is especially true for multiscale designs which often contain hundreds if not thousands of unit cells.

This work introduces a novel 3D multiscale heat sink modelling framework that addresses these challenges. The proposed approach uses an explicit 3D representation without relying on classical homogenization, ensuring accurate temperature profiles near interfaces. Numerical examples were used to highlight both the framework's practicality and its superior performance, compared to traditional explicit single-scale modelling. The results show up to 90% reduction in memory requirements and 70% decrease in wall time, achieved through speed-ups in meshing and more efficiently solving momentum and heat transport on the decomposed domains. The framework is validated through two optimization problems, using Bayesian optimization on a multiscale domain with 400 unit cells, a scale previously intractable for optimization with an explicit 3D representation. The numerical optimization results demonstrate the effectiveness of the framework for modelling optimizing complex multiscale heat sink using 3D explicit representations for the fluid and thermal behaviour at the microscale. The optimization results highlight that the models presented in this work are able to efficiently model and optimize multiscale heat sinks containing 100's of microstructures. Lastly, the use of both homogeneous and in-homogeneous applied heat flux distributions in the optimization setup further highlights the performance gains that can be made via design optimization of heat sinks in comparison to traditionally used homogeneous heat sink designs as heat flux distributions generated by electronic devices are seldom homogeneous.

The numerical examples presented reveal several promising research directions that could further enhance the framework's capabilities and performance:

- Optimization: Whilst the current Bayesian optimization implementation has proven effective, it constrains the number of design variables that can practically be considered. Exploring alternative optimization frameworks or developing new dimensionality reduction methods could enable the optimization of more complex designs with a larger number of unit cells, opening up possibilities for more sophisticated multiscale heat sink designs;
- Framework Scalability: The framework already handles hundreds of unit cells effectively for optimization, implementing data reuse strategies and multi-resolution techniques could make optimization of systems with thousands of unit cells more practical;
- Optimal Multiscale Discretization: The results from the non-uniform heat flux distribution has raised important questions about optimal discretization levels. The ideal level of discretization likely depends on both the microstructure parameterization and the specific heat flux distribution. Preliminary studies are presented in Appendix D to demonstrate the influence of the discretization on the multiscale solution and design optimization. However, further studies are needed to better understand this relationship, which could lead to more efficient and effective designs;
- Microstructure Parameterization: The proposed framework demonstrates potential for introducing additional microstructure design variables or entirely different, and potentially complex, parametrizations. This is enabled by the microstructure-agnostic nature of the framework;
- Refined Constraint Handling: Developing specialized GP regression techniques or new methods for pressure loss constraint integration could enhance the optimization process, leading to more efficient designs.

6 Replication of results

The authors wish to withhold the data for commercial purposes. This includes the multiscale model and a Python code implementing the optimization.

Governing equations on the single scale

Single-scale momentum transfer

On the single scale, the momentum transfer is described by the NS equations, where the aim is to find the velocity-pressure pair $\boldsymbol{u} : \Omega_{\mathrm{f}} \rightarrow \mathbb{R}^3$ and $p : \Omega_{\mathrm{f}} \rightarrow \mathbb{R}$ such that

$$\begin{aligned}\boldsymbol{u}\cdot\nabla\boldsymbol{u}-\nu\nabla^2\boldsymbol{u}+\frac{1}{\rho}\nabla p &= 0 \text{ in } \Omega_\mathrm{f}\\ \nabla\cdot\boldsymbol{u} &= 0 \text{ in } \Omega_\mathrm{f}\\ \boldsymbol{u} &= 0 \text{ on } \Gamma_\mathrm{I}\\ p &= 0 \text{ on } \Gamma_{\mathrm{x}+}\\ \boldsymbol{u} &= \boldsymbol{u}_\mathrm{in} \text{ on } \Gamma_{\mathrm{x}\text{-}}\end{aligned} \tag{A1}$$

where Γ_I is the solid–fluid interface, $\Gamma_{\mathrm{x}\text{-}}$ and $\Gamma_{\mathrm{x}+}$ are the inlet and outlet. Similar to the microscale NS problem outlined in Equation 4a, GLS stabilization is utilized for the single-scale discretization.

Single-scale heat transfer

On the single scale, the heat transport is described by an AD system, where the aim is to find $T : \Omega \rightarrow \mathbb{R}$ such that

$$\begin{aligned}-\kappa_\mathrm{s}\nabla^2 T &= 0 \text{ in } \Omega_\mathrm{s}\\ -\kappa_\mathrm{f}\nabla^2 T+\rho c_p\boldsymbol{u}\cdot\nabla T &= 0 \text{ in } \Omega_\mathrm{f}\\ T &= T_\mathrm{in} \text{ on } \Gamma_{\mathrm{x}-}\\ \kappa\nabla T\cdot\boldsymbol{n} &= q_\mathrm{in} \text{ on } \Gamma_{z-}\end{aligned} \tag{A2}$$

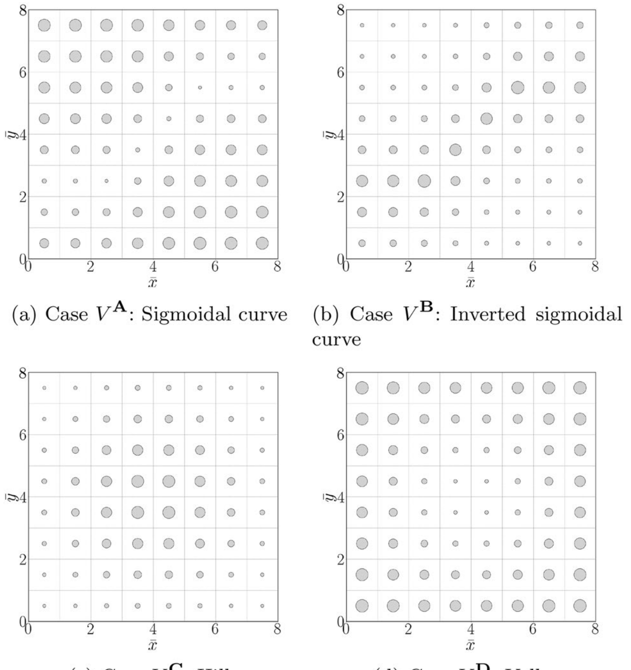
where q_in is the heat flux applied on the surface of the heat sink, T_in is the inlet temperature of the coolant fluid. Similar to the microscale AD system, system of equations is discretized using second-order Lagrange elements and apply GLS stabilization to improve the solver stability.

Multiscale momentum validation

Validation of the multiscale momentum framework has been conducted using explicit single-scale NS solutions. Four in-homogeneous distributions of $\chi \in \mathbb{R}^{1\times8\times8}$, which are shown in Fig. 11, were utilized. The domain and boundary conditions used for all validation cases are described in Table 2, with the material properties of Ω_s and Ω_f given in Table 3. To solve the multiscale momentum transfer problem, the macroscale momentum transfer model is first evaluated to compute $\overline{\nabla p}^{ij}$ $\forall i \in [1, N_x], \forall j \in [1, N_y]$. Each $\overline{\nabla p}^{ij}$ is then used to solve Equation 2 on each $\bar{\Omega}^{ij}$. Finally, the single-scale problem is solved using Equation A1. In the validation, the total pressure gradients and microscale velocities predicted by both methods are compared. Both systems are discretized and solved in `FEniCS` (Logg and Wells 2010; Alnæs et al. 2015) using a direct LU solver. The single-scale problem is solved using 64 `MPI` cores, whilst the microscale solutions of Equation 2 are computed across the $N_x \times N_y$ domain using 3 `MPI` cores per $\bar{\Omega}$. In addition, 12 parallel processes are utilized to solve 12 microscale problems simultaneously, requiring 36 processors in total. Both the single-scale and microscale meshes are created using `GMSH` (Geuzaine and Remacle 2009), where the microscale meshes are generated in parallel on 12 processors.

Table 6 Pressure error validation results for multiscale momentum transfer model

Case	ΔP error (% relative to SS)
$V^\mathbf{A}$	− 5.0
$V^\mathbf{B}$	4.8
$V^\mathbf{C}$	− 2.4
$V^\mathbf{D}$	− 8.5

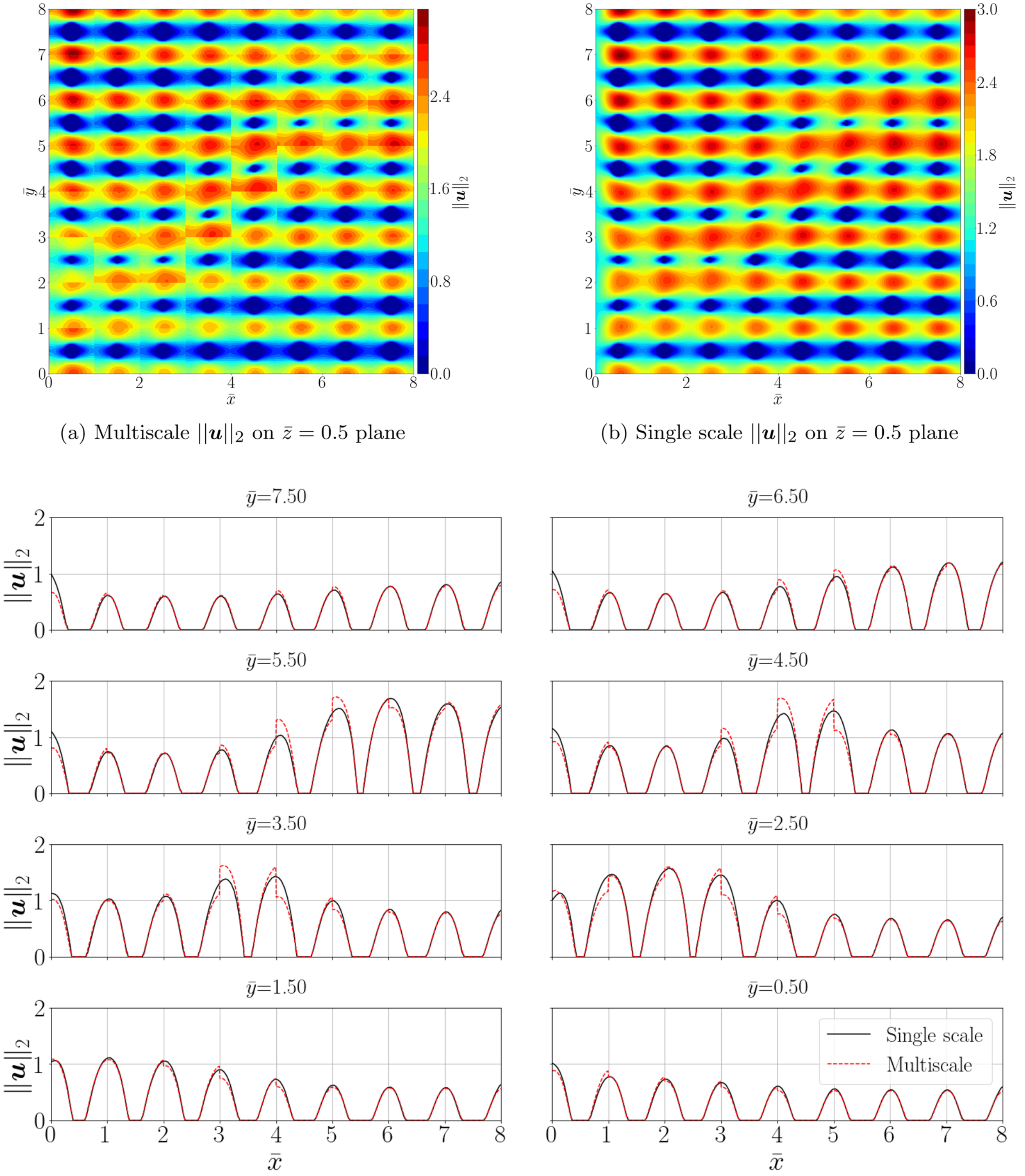


(a) Case $V^\mathbf{A}$: Sigmoidal curve (b) Case $V^\mathbf{B}$: Inverted sigmoidal curve

(c) Case $V^\mathbf{C}$: Hill (d) Case $V^\mathbf{D}$: Valley

Fig. 11 Distributions of χ used for single-scale validation

The numerical results of the validation study are presented in Table 6, where excellent agreement between the multiscale and single-scale Δp values is found, with less than 10% error across all test cases. Comparisons of $||\boldsymbol{u}||_2$ along the $\bar{z} = 0.5$ surface and line plots along varying $\bar{y}$ for case $V^\mathbf{A}$ are shown in Fig. 12, 13. From these plots, it should be noted that the multiscale solution closely matches the single-scale solution. Differences appear near $\partial\bar{\Omega}$'s, as expected, since periodic solutions in $\boldsymbol{u}$ are expected, whereas in-homogeneous χ leads to discontinuous changes in the homogenized permeability of each $\bar{\Omega}$, which are visible in the line plots across cell boundaries. In addition, performance figures for both methods are presented in Table 7. The multiscale method is shown to be 74% faster on average (for meshing + solve) compared to the conventional, explicit, single-scale method, for the `MPI` and ensemble parallel processing setup used here. The memory requirements are also significantly reduced, although it



(a) Multiscale $||\boldsymbol{u}||_2$ on $\bar{z} = 0.5$ plane

(b) Single scale $||\boldsymbol{u}||_2$ on $\bar{z} = 0.5$ plane

(c) Line plots comparing $||\boldsymbol{u}||_2$ for single and multiscale solutions at various $\bar{y}$ on the $\bar{z} = 0.5$ plane

Fig. 12 Comparison of multiscale and single-scale momentum transfer solutions for case $V^{\mathbf{A}}$

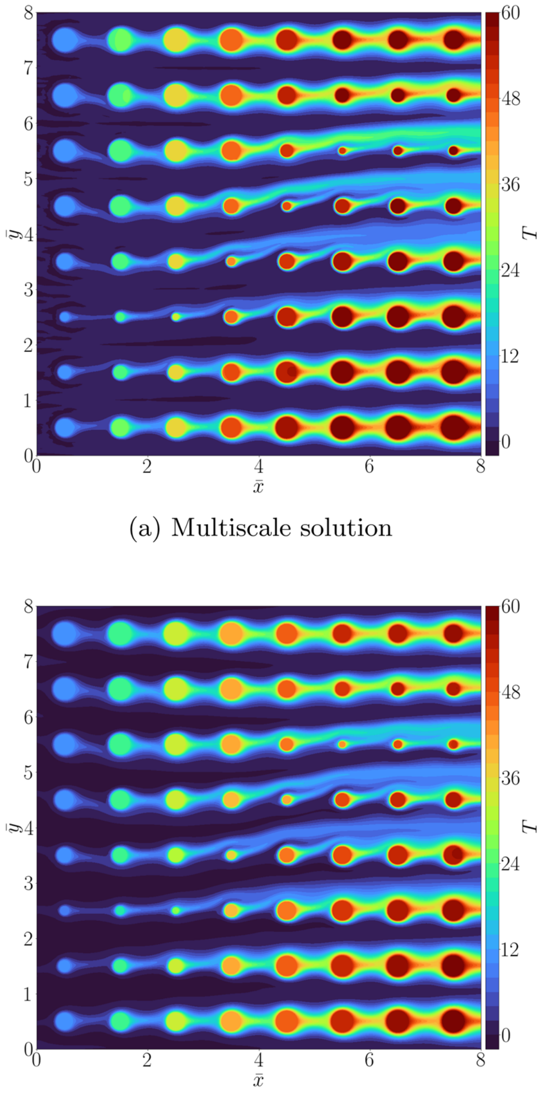
Table 7 Computational performance metrics for multiscale momentum transfer model

Case	Meshing time		Solve time		Memory usage		DoFs	
	MS	SS	MS	SS	MS	SS	MS	SS
$V^{\mathbf{A}}$	20.1 s	479.1 s	454.5 s (22.6 s†, 431.9 s*)	1971.3 s	14.5 GB	80.6 GB	6.0 m	7.4 m
$V^{\mathbf{B}}$	28.8 s	515.7 s	529.1 s (23.4 s†, 400.6 s*)	1836.0 s	14.2 GB	83.7 GB	5.8 m	7.6 m
$V^{\mathbf{C}}$	20.5 s	496.1 s	431.9 s (23.7 s†, 408.2 s*)	1896.3 s	14.5 GB	88.6 GB	5.8 m	7.5 m
$V^{\mathbf{D}}$	22.4 s	472.8 s	459.2 s (23.7 s†, 435.5 s*)	1969.3 s	13.6 GB	85.7 GB	6.0 m	7.5 m

should be noted that these figures are highly dependent on the combination of the solver, number of `MPI` cores and parallel processing cores utilized. Overall it is found that the multiscale method is able to accurately and more efficiently (in terms of compute requirements and time) solve the micro- and macroscale fluid behaviour in comparison to the single-scale method, which lends itself to design optimization.

Multiscale advection–diffusion solver algorithms

The multiscale thermal solver relies on iteratively solving the advection–diffusion equation on each $\bar{\Omega}$ with updated boundary conditions at each iteration. Since boundary conditions on each $\bar{\Omega}$ is projected from adjacent cells, each $\bar{\Omega}$ can be solved once the boundary information of its adjacent cells is available, locally advancing the iterations faster than solving all cells in a given iteration before re-solving the cells with updated boundary conditions. To start the multiscale solution procedure, a cell solver ordering matrix, $D \in \mathbb{N}^{N_x \times N_y}$, is first constructed, which dictates the order in the cells can be solved to maximize the number of cells that are solved concurrently. Using D, the multiscale solver is evaluated using the procedure shown in Algorithm 1.



(a) Multiscale solution

(b) Single scale solution

Fig. 13 Comparison of multiscale (left) and single-scale (right) temperature profiles for case $V^{\mathbf{A}}$

Algorithm 1 Concurrent Multiscale Thermal Solver

```
Require: Initial conditions, domain decomposition $\{\bar{\Omega}^{i,j}\}_{i,j=1}^{N_x,N_y}$
Ensure: Converged temperature field $T$
for $i \in [1, N_x], j \in [1, N_y]$ do
    Solve $\boldsymbol{u} \in \bar{\Omega}^{i,j}$                                        ▷ Equation 2
Compute global cell ordering matrix $D \in \mathbb{N}^{N_x \times N_y}$
Set $k \leftarrow 0$
while $\epsilon^{\text{out}}$ AND $\epsilon^{\text{plate}} > \epsilon^{\text{TOL}}$ do
    for all $\bar{\Omega}^{i,j}$ in parallel where $(i, j)$ follows ordering in $D$ do
        if $k = 0$ OR
        $(\mathcal{C}_T^{i,j}$ solved at iteration $k$ and $\mathcal{C}_q^{i,j}$ solved at iteration $k-1)$ then
            for $X \in \{\bar{x}, \bar{y}\}$ do
                if $\frac{\partial p}{\partial X} < 0$ then
                    Set $\Gamma_{X+}^{i,j}$ as Dirichlet boundary
                else
                    Set $\Gamma_{X+}^{i,j}$ as Neumann boundary
                Set $(\delta_i, \delta_j) \leftarrow \begin{cases} (1,0) & \text{if } X = \bar{x} \\ (0,1) & \text{otherwise} \end{cases}$
                Set $g_c^{X+} \in \partial\bar{\Omega}^{i,j} \leftarrow \begin{cases} 0 & \text{if } k = 0 \\ g_{k-1}^{X-} \in \partial\bar{\Omega}^{i+\delta_i, j+\delta_j} & \text{if } k > 0 \end{cases}$
                Set $T_D^{X+} \in \partial\bar{\Omega}^{i,j} \leftarrow T_k^{X-} \in \partial\bar{\Omega}^{i+\delta_i, j+\delta_j}$
            Solve $T \in \bar{\Omega}^{i,j}$ with $g = g_c$ and $T_D$                    ▷ Equation 7
            Mark $\bar{\Omega}^{i,j}$ as solved at iteration $k$
    $k \leftarrow k + 1$
```

Multiscale advection–diffusion solver validation

The multiscale AD solver was validated against explicit single-scale AD solutions. The boundary conditions used for the validation are given in Table 2, with the material properties shown in Table 3. In the validation, the total outlet heat flux on the single- ($\mathcal{H}$) and multiscale ($\bar{\mathcal{H}}$) is compared

$$\begin{aligned} \mathcal{H} &= \int_{\Gamma^{x+}} \rho c_p T(\boldsymbol{u} \cdot \boldsymbol{n}) ds \\ \bar{\mathcal{H}} &= \sum_{j=1}^{N_y} \left(\int_{\Gamma^{N_x,j,x+}} \rho c_p T(\boldsymbol{u} \cdot \boldsymbol{n}) ds \right) \\ \Delta\mathcal{H} &= (\bar{\mathcal{H}} - \mathcal{H})/\mathcal{H} \end{aligned} \tag{C3}$$

where $\Delta\mathcal{H}$ is the error parameter. On the single scale, the NS system is solved using 64 `MPI` cores. In the multiscale framework, 64 parallel cores are used to solve Equation 2 on each $\bar{\Omega}$ in an ensemble manner and 8 parallel cores for the multiscale AD solver. Both the single-scale and multiscale AD systems are solved using a direct LU solver. Whilst there are performance gains to be had with alternative solvers, e.g.

Table 8 Computational performance metrics for the multiscale AD solver

Case	DoFs		Solve time		Memory usage		$\Delta\mathcal{H}$
	SS	MS	SS	MS	SS	MS	
$V^{\mathbf{A}}$	7.4 m	6.0 m	1928.1 s	900.4 s (−53.3%)	394.3 GB	43.6 GB (−88.9%)	+12.3%
$V^{\mathbf{B}}$	7.6 m	5.8 m	2751.7 s	959.9 s (−65.1%)	435.5 GB	44.2 GB (−89.9%)	+8.9%
$V^{\mathbf{C}}$	7.5 m	5.8 m	2165.3 s	1348.0 s (−37.7%)	405.5 GB	45 GB (−88.9%)	+9.8%
$V^{\mathbf{D}}$	7.5 m	6.0 m	2227.4 s	857.6 s (−61.5%)	385.2 GB	45 GB (−88.3%)	−3.3%

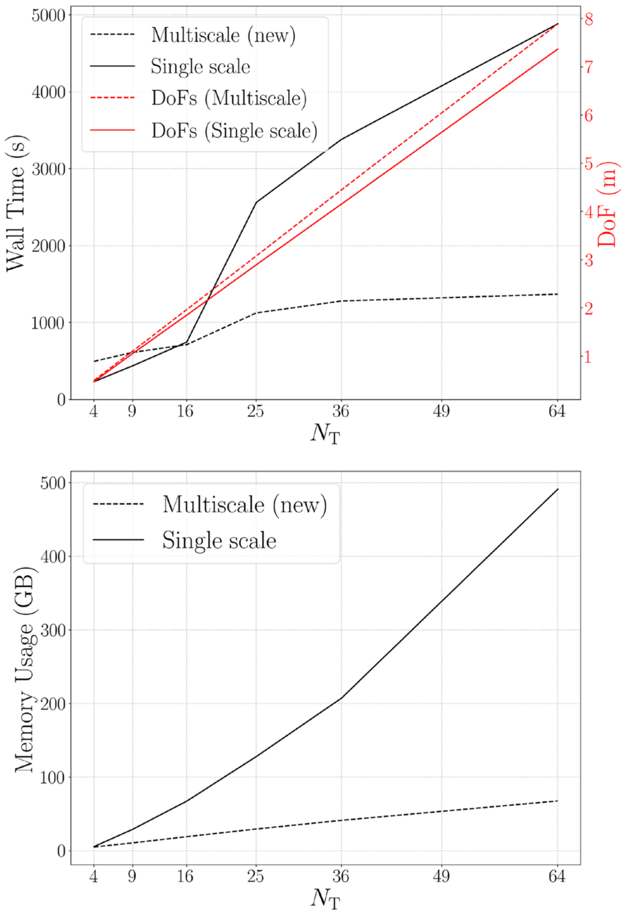


Fig. 14 Weak scaling for single-scale and multiscale (meshing + NS solve + AD solve)

iterative Multigrid, the LU solver is used for both cases to enable equivalent, baseline, comparisons. Future work will be directed towards selecting more performant solvers and pre-conditioners for both scales. Lastly, artificial diffusion is added via GLS stabilization to the multiscale and single-scale AD systems to overcome stabilization issues arising in standard Galerkin formulations and solved using the formulation presented in A.2. The multiscale stabilization term is scaled using the homogenized, cell-wise Pe number, whilst on the single scale, the local flow velocities are directly used to compute the local flow Pe number (13).

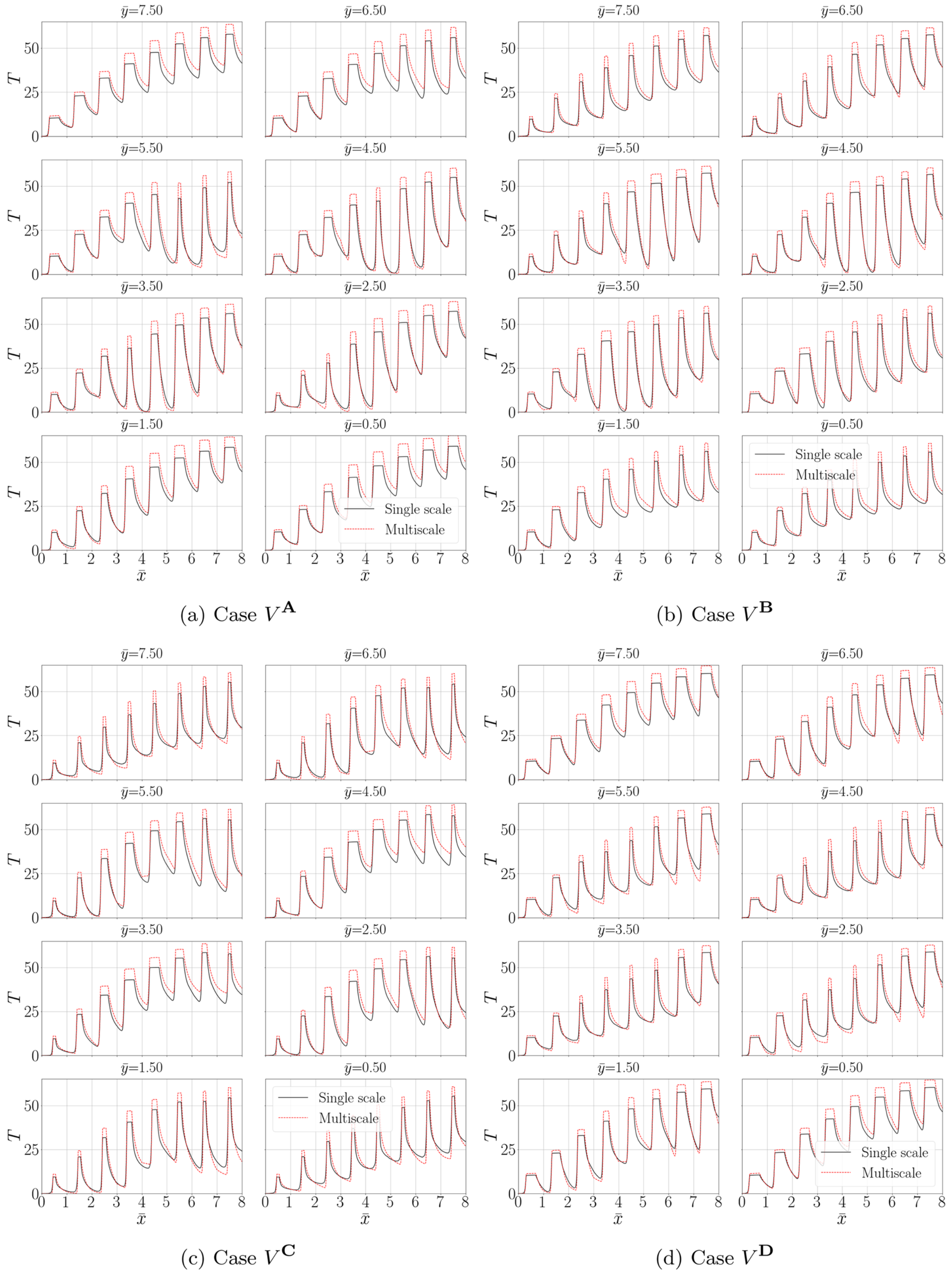
The numerical results of the validation study are presented in Table 8. The multiscale framework solves the AD system 40-60% faster compared to the single scale solve with significantly lower memory requirements. It should be noted that whilst the same solver is utilized for both the single scale and multiscale, the memory requirements and solution times are dependent on the chosen solver and level of `MPI` parallelism. The number of degrees of freedom (DoF) in the single-scale system is also greater than the multiscale system as a finer mesh discretization was needed to ensure converged results on the single scale. Overall the validation results demonstrated that the multiscale solver is able to capture the correct temperature distributions, with an average absolute error of less than 10% for the outlet heat flux. The line plots shown in Fig. 15 further highlight the efficacy of the multiscale solver as it closely matches the single-scale temperature distribution, particularly at the boundaries of the cells.

Weak scaling tests are also performed to inspect the behaviour of the multiscale solver as the number of unit cells increases. For increasing $N = N_x \times N_y$, the single-scale system (i.e. explicit single-scale meshing, NS and AD) and multiscale system (i.e. multiscale meshing, momentum transfer and AD solves) are solved with an increasing number of processors. For both cases, the number of parallel processors are set to N for each run. In the single-scale tests, this corresponds to the number of `MPI` processors, and on the multiscale, these are simply parallel cores. The results of the weak scaling test are shown in Fig. 14 where the multiscale framework is found to have significantly better scaling performance both for the total wall time and memory requirements, due to the fully decoupled nature of the meshing and NS solutions and semi-decoupled nature of the multiscale AD solver.

Impact of multiscale discretization

The choice of multiscale discretization acts as another parametrization variable in the design optimization. Higher discretization levels could enable the optimizer to more finely control local fluid and thermal flow, which may be particularly beneficial when imposing in-homogeneous heat flux. This is demonstrated in Fig. 16, where the temperature profile along $\bar{z} = 0.5$ is shown for heat sinks with increasing multiscale discretization from 900 cells to 3600 cells with fixed $L = W = 60$ mm and decreasing H. As the discretization is increased, the influence of individual microstructures on the flow reduces, leading fluid and thermal flow features on a reduced length scale.

In a design optimization setting, this increased control can only be harnessed if the number of design variables is increased in line with the number of microstructure cells. To demonstrate this, the optimization results presented in Table 4 for Case **A** (homogeneous heat flux) are re-run with increasing multiscale discretization with fixed $\tilde{\mathcal{S}} \in \mathbb{R}^{8\times5}$ (i.e. 40 design variables). The multiscale discretization is increased from $\mathcal{S} \in \mathbb{R}^{1\times16\times16}$ to $\mathcal{S} \in \mathbb{R}^{1\times30\times30}$. The optimized heat flux values are presented in Table 9 and the resulting temperature profiles along $\bar{z} = 0.5$ are presented alongside the optimized pin-fin distributions in Fig. 17. The pin-fin distribution converges to the same design as the multiscale discretization is increased. As the



(a) Case $V^{\mathbf{A}}$

(b) Case $V^{\mathbf{B}}$

(c) Case $V^{\mathbf{C}}$

(d) Case $V^{\mathbf{D}}$

Fig. 15 Line plots showing comparison of multiscale and single-scale T for various $\bar{y}$ along the $\bar{z} = 0.5$ plane

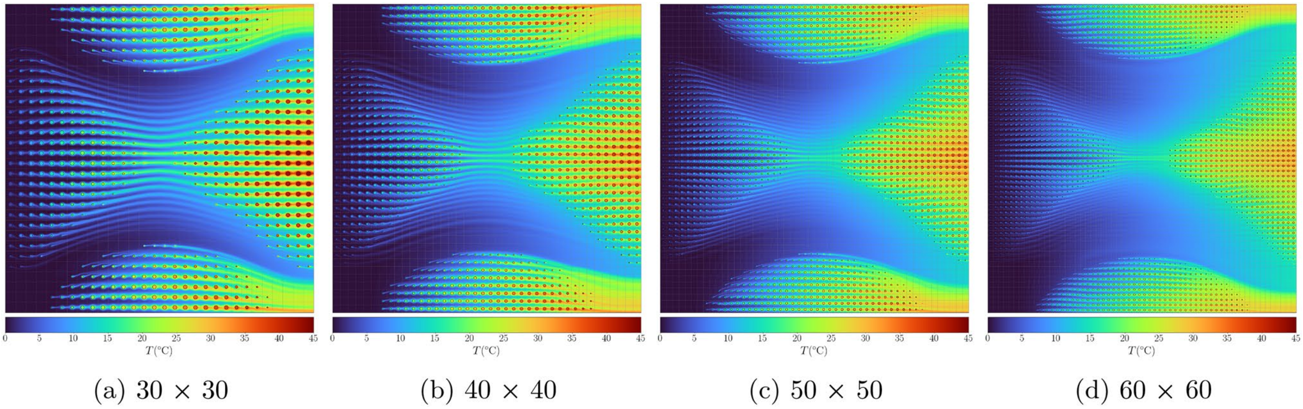


Fig. 16 Temperature distribution along $\bar{z} = 0 : 5$ plane for heat sink arrays with increasing multiscale discretization

Table 9 Influence of multiscale discretization on multiscale optimization

Multiscale Discretization	Optimized heat flux	Benchmark Heat flux
16×16	5.64	2.01
20×20	5.64	3.21
24×24	6.70	4.77
30×30	10.22	7.59

number of design variables is kept fixed, the optimizer is unable to leverage the increase in local flow control. These preliminary results highlight that whilst multiscale discretization indeed has the potential to impact the design optimization through increased fluid and thermal flow control, the increased multiscale discretization should be paired with an increased number of optimization design variables to effectively leverage this phenomenon.

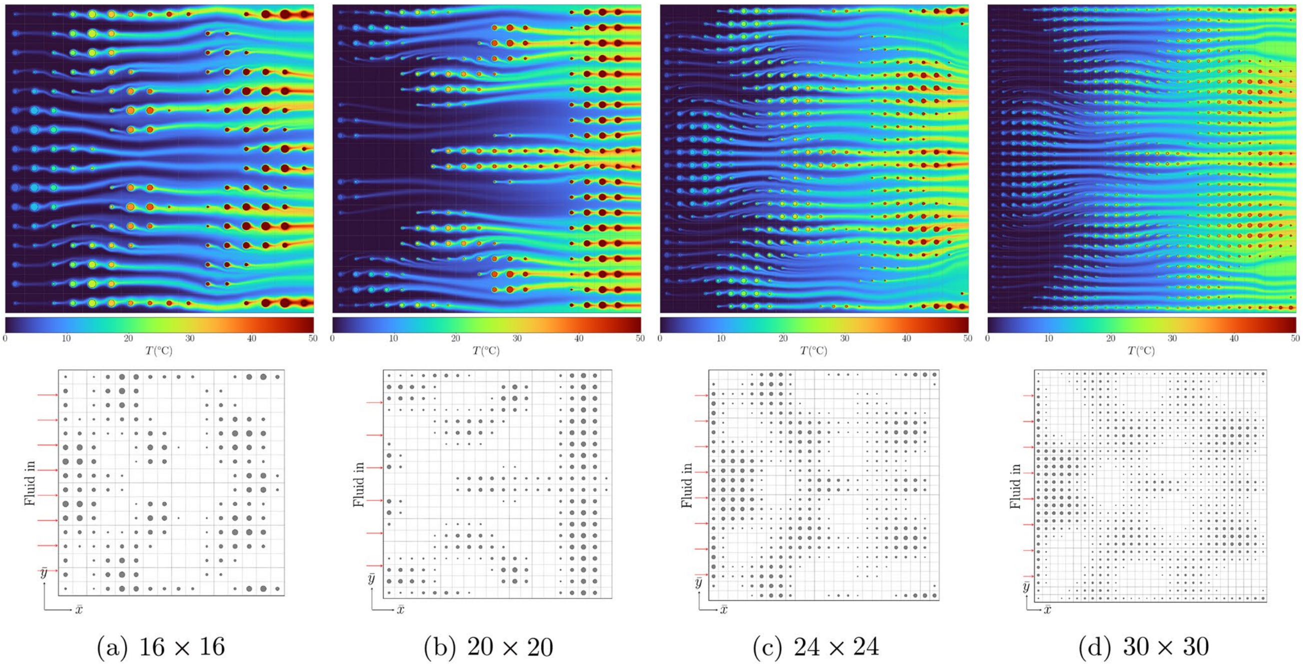


Fig. 17 Optimized designs for varying multiscale discretizations showing optimized temperature distribution (top row) and optimized pin-fin radii (bottom row) for discretizations ranging from 16×16 to 30×30

Acknowledgements This work was supported by the UK National Aerospace Technology Exploitation Programme (project number: 10061301). We gratefully acknowledge the contributions of the TofeeX team, whose guidance on the software implementation of this work is greatly appreciated.

Author Contributions **D.T** contributed towards conceptualization, methodology, formal analysis, validation, visualization, investigation, software, writing-original draft, review and editing. **R.M** contributed towards methodology, software and writing-review. **R.H** and **M.S** contributed towards funding acquisition, conceptualization and writing-review and editing. **A.C** and **N.G** contributed towards software. **N.R** contributed towards funding acquisition, conceptualization and Writing-review.

Funding This work was supported by the UK National Aerospace Technology Exploitation Programme (project number: 10061301).

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no Conflict of interest.

References

Allaire G, Dapogny C, Frey P (2014) Shape optimization with a level set based mesh evolution method. Comput Methods Appl Mech Eng 282:22–53. https://doi.org/10.1016/j.cma.2014.08.028

Alnæs MS, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, Richardson C, Ring J, Rognes ME, Wells GN (2015) The FEniCS Project Version 1.5. Archive of Numerical Software https://doi.org/10.11588/ans.2015.100.20553

Bacellar D, Aute V, Huang Z, Radermacher R (2017) Design optimization and validation of high-performance heat exchangers using approximation assisted optimization and additive manufacturing. Science and Technology for the Built Environment 23(6):896–911. https://doi.org/10.1080/23744731.2017.1333877

Bhunia AK, Sahoo L, Shaikh AA (2019) Constrained Optimization with Inequality Constraints. In Springer Optimization and Its Applications 153:249–267. https://doi.org/10.1007/978-981-32-9967-2_11

Borrvall T, Petersson J (2003) Topology optimization of fluids in Stokes flow. Int J Numer Meth Fluids 41(1):77–107. https://doi.org/10.1002/fld.426

Feppon F, Allaire G, Dapogny C, Jolivet P (2021) Body-fitted topology optimization of 2D and 3D fluid-to-fluid heat exchangers. Comput Methods Appl Mech Eng 376:113638. https://doi.org/10.1016/j.cma.2020.113638

Geuzaine C, Remacle J (2009) Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. Int J Numer Meth Eng 79(11):1309–1331. https://doi.org/10.1002/nme.2579

Hewson RW, Kapur N, Gaskell PH (2011) A two-scale model for discrete cell gravure roll coating. Chem Eng Sci 66(16):3666–3674. https://doi.org/10.1016/j.ces.2011.04.034

Høghøj LC, Nørhave DR, Alexandersen J, Sigmund O, Andreasen CS (2020) Topology optimization of two fluid heat exchangers. Int J Heat Mass Transf 163:120543. https://doi.org/10.1016/j.ijheatmasstransfer.2020.120543

Hughes TJR, Franca LP (1987) A new finite element formulation for computational fluid dynamics: VII. The stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces. Comput Methods Appl Mech Eng 65(1):85–96. https://doi.org/10.1016/0045-7825(87)90184-8

Jones D, Schonlau M, Welch W (1998) Efficient global optimization of expensive black-box functions. J Global Optim 13:455–492. https://doi.org/10.1023/A:1008306431147

Kobayashi H, Yaji K, Yamasaki S, Fujita K (2021) Topology design of two-fluid heat exchange. Struct Multidiscip Optim 63(2):821–834. https://doi.org/10.1007/s00158-020-02736-8

Krige DG (1951) Journal of the Chemical Metallu, Rgical & Mining Society of South Africa. Journal of the Chemical Metallurgical & Society of South Mining Africa 52(6):119–139

Liu Z, Korvink JG (2008) Adaptive moving mesh level set method for structure topology optimization. Eng Optim 40(6):529–558. https://doi.org/10.1080/03052150801985544

Logg A, Wells GN (2010) DOLFIN: Automated Finite Element Computing. ACM Transactions on Mathematical Software **37**(2) https://doi.org/10.1145/1731022.1731030

Murphy R, Imediegwu C, Hewson R, Santer M (2021) Multiscale structural optimization with concurrent coupling between scales. Struct Multidiscip Optim. https://doi.org/10.1007/s00158-020-02773-3

Nogueira F (2014) Bayesian Optimization: Open source constrained global optimization tool for Python. https://github.com/bayesian-optimization/BayesianOptimization

Olesen LH, Okkels F, Bruus H (2006) A high-level programming-language implementation of topology optimization applied to steady-state Navier-Stokes flow. Int J Numer Meth Eng 65(7):975–1001. https://doi.org/10.1002/nme.1468 [physics]

Rao R, Vrudhula S (2007) Performance optimal processor throttling under thermal constraints, pp. 257–266. ACM, New York, NY, USA. https://doi.org/10.1145/1289881.1289925

Raske N, Hewson RW, Kapur N, Boer GN (2017) A predictive model for discrete cell gravure roll coating. Physics of Fluids **29**(6) https://doi.org/10.1063/1.4984127

Schevenels M, Lazarov BS, Sigmund O (2011) Robust topology optimization accounting for spatially varying manufacturing errors. Comput Methods Appl Mech Eng 200(49–52):3613–3627. https://doi.org/10.1016/j.cma.2011.08.006

Shimoyama K, Komiya A (2022) Multi-objective Bayesian topology optimization of a lattice-structured heat sink in natural convection. Struct Multidiscip Optim 65(1):1–15. https://doi.org/10.1007/s00158-021-03092-x

Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian Optimization of Machine Learning Algorithms. https://arxiv.org/abs/1206.2944

Sparrow EM, Prata AT (1983) Numerical solutions for laminar flow and heat transfer in a periodically converging-diverging tube, with experimental confirmation. Numerical Heat Transfer 6(4):441–461. https://doi.org/10.1080/01495728308963099

Tancabel J, Aute V, Klein E, Lee C-Y, Hwang Y, Ling J, Muehlbauer J, Radermacher R (2022) Multi-scale and multi-physics analysis, design optimization, and experimental validation of heat exchangers utilizing high performance, non-round tubes. Appl Therm Eng 216:118965. https://doi.org/10.1016/j.applthermaleng.2022.118965

Troya MAS, Tortorelli DA, Andrej J, Beck VA (2021) Three-dimensional topology optimization of heat exchangers with the level-set method. 14th WCCM-ECCOMAS Congress 1300:1–25. DOI: https://doi.org/10.48550

Van Dijk NP, Maute K, Langelaar M, Van Keulen F (2013) Level-set methods for structural topology optimization: A review. Struct Multidiscip Optim 48(3):437–472. https://doi.org/10.1007/s00158-013-0912-y

Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192(1):227–246. https://doi.org/10.1016/S0045-7825(02)00559-5

Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. Struct Multidiscip Optim 43(6):767–784. https://doi.org/10.1007/s00158-010-0602-y

Wang M, Wang X (2004) Pde-driven level sets and shape sensitivity for structural topology optimization. In: ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 1–9. ASME, ???. https://doi.org/10.1115/detc2004-57038

Yaji K, Yamasaki S, Fujita K (2022) Data-driven multifidelity topology design using a deep generative model: Application to forced convection heat transfer problems. Comput Methods Appl Mech Eng 388:114284. https://doi.org/10.1016/j.cma.2021.114284