



UDC 519.63, 536.2

PACS 44.10.+i, 07.05.Tp, 02.70.Bf

DOI: 10.22363/2658-4670-2026-34-3-408-429

EDN: QXTTKT

Heat-conduction simulation of a wing-like thulium foil for an ion source

Alexander S. Ayriyan^{1,2}, Edik A. Ayryan¹, Eugeny E. Donets¹, Milena P. Veneva³

¹Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, 141980, Russian Federation

²Alikhanyan National Science Laboratory (YerPhI), 2 Alikhanyan Brothers St., Yerevan, 0036, Armenia

³RIKEN Center for Computational Science, 7-1-26 Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan

(received: July 28, 2026; revised: August 10, 2026; accepted: August 12, 2026)

Abstract. *Background.* Metal foils heated by an electric current are commonly used as evaporation elements in ion sources. Because the evaporation rate is highly sensitive to temperature, predicting the temperature distribution accurately and efficiently is essential for choosing the foil geometry and operating conditions. *Purpose.* This study develops a two-dimensional electric-thermal model of a wing-shaped thulium foil and evaluates numerical strategies for repeated simulations. *Methods.* Owing to symmetry, only one quarter of the foil is included in the computational domain. The current paths are determined by solving a Laplace problem with constant electrical conductivity. The resulting field is then scaled to the prescribed total current and used to calculate the spatially varying Joule heat source. The transient heat equation accounts for the temperature dependence of density, heat capacity, thermal conductivity, and electrical resistivity, as well as radiative and evaporative heat losses from both faces of the foil. The equation is discretized on a masked Cartesian grid and integrated using an alternating-direction implicit scheme, with Picard iterations applied at each half-step. *Results.* At a current of 2.5 A and a contact temperature of 78 K, the stationary maximum temperature is 829.803 K. All tested tridiagonal solvers produce the same converged temperature field. The fastest partition solver, combined with a CUDA task graph, completes the simulation in 36.3 s, giving a 24.1-fold speedup over the serial central-processing-unit implementation. *Conclusions.* The model accounts for nonuniform electric heating and nonlinear heat losses in the foil. Its graphics-processing-unit implementation is sufficiently efficient for repeated optimization of the foil geometry and operating current.

Key words and phrases: nonlinear heat conduction, Joule heating, alternating-direction implicit method, tridiagonal solver, partition methods, CUDA

For citation: A. S. Ayriyan, E. A. Ayryan, E. E. Donets, M. P. Veneva “Heat-conduction simulation of a wing-like thulium foil for an ion source,” *Discrete and Continuous Models and Applied Computational Science*, vol. 34, no. 3, pp. 408–429, 2026. DOI: 10.22363/2658-4670-2026-34-3-408-429. EDN: QXTTKT.

1. Introduction

Current-heated metallic foils are commonly used as evaporation elements in ion sources [1]. Their operating temperature is set by the competition between Joule heating, conduction to the cooled contacts, radiation, and evaporation. Because the evaporation rate is strongly temperature dependent, even a moderate error in the predicted temperature field can lead to a large error in the estimated material flux.

We consider a thulium foil with rectangular side wings. The wings enlarge the heated area but also disturb the current paths, particularly at the abrupt changes in width. The resulting problem is nonlinear: density, heat capacity, thermal conductivity, and electrical resistivity all vary with temperature, while radiation and evaporation introduce additional temperature-dependent losses.

An earlier formulation ranked candidate geometries by the area lying within a specified temperature interval [2]. For an evaporation source, the surface flux is a more direct performance measure. We therefore compute the stationary temperature field and evaluate the local evaporation rate from a vapor-pressure correlation and the Hertz–Knudsen formula [3–5].

© 2026 A. S. Ayriyan, E. A. Ayryan, E. E. Donets, M. P. Veneva



This work is licensed under a Creative Commons “Attribution-NonCommercial 4.0 International” license.

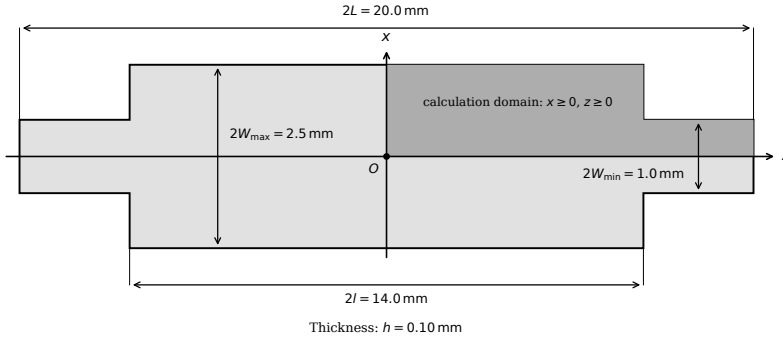


Figure 1. Geometry of the wing-shaped thulium foil in the (x, z) plane. The plate is symmetric about both the x - and z -axes: the wide central region of full width $2W_{\max} = 2.5$ mm and length $2l = 14.0$ mm narrows to the terminal contacts of width $2W_{\min} = 1.0$ mm, giving a total length $2L = 20.0$ mm and a uniform thickness $h = 0.10$ mm. Due to the symmetry, the computations are restricted to one quarter of the plate ($x \geq 0, z \geq 0$; shaded region) only, with the origin O at the centre

The optimization itself requires many solutions for different wing lengths and currents. This paper addresses the direct problem that underlies those repeated evaluations. We discretize one quarter of the symmetric foil on a masked Cartesian grid and use an alternating-direction implicit (ADI) scheme. Each half-step produces independent tridiagonal systems along rows or columns, which we solve in batches on a CUDA-capable GPU. The calculation returns the stationary field, the maximum foil temperature, and the total evaporation flux; these quantities will serve as the objective and constraint data in the subsequent optimization.

2. Mathematical model and direct heat-conduction problem

The model couples electric conduction to transient heat conduction in a thin thulium foil of prescribed geometry. Radiation and evaporation enter the thermal balance as losses from the two broad faces of the foil.

2.1. Rectangular wing-shaped foil

We use the one-parameter foil geometry introduced in Ref. [2]. The foil has total length $2L$ and minimum width $2W_{\min}$. Two symmetric rectangular wings increase the width to $2W_{\max}$ over a half-length l , with $0 \leq l \leq L$. The limits $l = 0$ and $l = L$ give uniform rectangles of widths $2W_{\min}$ and $2W_{\max}$, respectively.

The local half-width of the foil is defined as

$$W_l(z) = \begin{cases} W_{\max}, & |z| \leq l, \\ W_{\min}, & l < |z| \leq L \end{cases}.$$

The corresponding two-dimensional domain is

$$\Omega_l = \{(x, z) : -L \leq z \leq L, -W_l(z) \leq x \leq W_l(z)\}.$$

All four geometrical parameters are fixed for a single direct calculation (see Figure 1).

2.2. Heat-conduction problem

For a prescribed shape $W_l(z)$ and current I , the temperature satisfies

$$\rho_m(T)c_p(T)\frac{\partial T}{\partial t} = \nabla \cdot (\lambda(T)\nabla T) + \rho_e(T)|\mathbf{j}(x, z; I, l)|^2 - Q_{\text{loss}}(T). \quad (1)$$

Here ρ_m , c_p , λ , and ρ_e denote the mass density, specific heat capacity, thermal conductivity, and electrical resistivity of thulium, and $\mathbf{j}$ is the in-plane current-density field.

The foil thickness h is small relative to its in-plane dimensions, so the temperature is taken to be uniform through the thickness. We nevertheless retain both in-plane coordinates. A one-dimensional approximation, $j(z) = I/[2hW_i(z)]$, cannot resolve current crowding at the steps in foil width; the current field is therefore obtained from a separate two-dimensional potential problem.

2.3. Electric-current distribution

The current distribution is computed once for each geometry, before thermal integration begins. This auxiliary problem uses constant conductivity because only the geometry of the current paths is required. The normalized potential $\phi(x, z)$ satisfies

$$\nabla \cdot (\sigma_0 \nabla \phi) = 0, \quad \sigma_0 = 1,$$

on the same quarter-domain mask as the heat problem.

Because the equation is homogeneous, σ_0 does not affect the potential shape. Setting $\sigma_0 = 1$ simply fixes an auxiliary normalization; the field is later rescaled to I in Eq. (2). Temperature dependence enters only through $\rho_e(T)$ in the Joule source.

In the quarter-domain calculation, $z = 0$ is the mid-plane of the full foil and is assigned $\phi = 0.5$; the right terminal at $z = L$ is assigned $\phi = 0$ for $x \leq W_{\min}$.

All remaining free boundaries are insulating. This boundary-value problem can be effectively solved by using the over-relaxation method [6, 7]. Numerically, fluxes through faces without an active neighboring node are omitted.

The unscaled current-density field is

$$\mathbf{j}_0 = -\sigma_0 \nabla \phi.$$

The total current carried by the computed quarter-domain field through the terminal section is evaluated as

$$I_0^{(1/4)} = h \int_0^{W_{\min}} |(\mathbf{j}_0)_z(x, L)| dx.$$

The physical field used in the Joule source is then obtained by scaling $\mathbf{j}_0$ to the prescribed total current I :

$$\mathbf{j}(x, z; I, L) = \frac{I/2}{I_0^{(1/4)}} \mathbf{j}_0(x, z). \quad (2)$$

The Joule source is then

$$q_J(x, z, T) = \rho_e(T) |\mathbf{j}(x, z; I, L)|^2.$$

The loss term combines radiation with the energy carried by evaporated atoms. Surface fluxes from the two broad faces are converted to a volumetric rate by dividing by the foil thickness:

$$Q_{\text{loss}}(T) = \frac{2}{h} [\varphi_{\text{rad}}(T) + \varphi_{\text{evap}}(T)].$$

The radiative heat flux from one surface [8] is written as

$$\varphi_{\text{rad}}(T) = \varepsilon \sigma_{\text{SB}} (T^4 - T_{\text{env}}^4),$$

where ε is the effective hemispherical emissivity, σ_{SB} is the Stefan–Boltzmann constant, and T_{env} is the effective radiation temperature of the chamber.

The heat flux carried away by evaporation from one surface is estimated as

$$\varphi_{\text{evap}}(T) = \Gamma(T) \frac{\Delta H_{\text{sub}}}{N_A}.$$

Here $\Gamma(T)$ is the evaporation flux, ΔH_{sub} is the molar enthalpy of sublimation, and N_A is Avogadro's constant. We use a constant reference value for ΔH_{sub} .

Table 1

Saturated vapor pressure and estimated evaporation flux of metallic thulium. The flux was calculated using the Hertz–Knudsen relation with the evaporation coefficient taken equal to unity. The vapor-pressure relation is recommended for 298 K to 1400 K; values below 298 K are shown only as a formal extrapolation.

T (K)	P_{sat} (Pa)	Γ (atom cm $^{-2}$ s $^{-1}$)
100	1.88×10^{-111}	3.82×10^{-93}
200	2.17×10^{-50}	3.12×10^{-32}
298	2.22×10^{-30}	2.61×10^{-12}
300	4.16×10^{-30}	4.87×10^{-12}
400	5.30×10^{-20}	5.37×10^{-2}
500	5.84×10^{-14}	5.30×10^4
600	6.04×10^{-10}	5.00×10^8
678	1.21×10^{-7}	9.42×10^{10}
700	4.35×10^{-7}	3.33×10^{11}
778	2.25×10^{-5}	1.63×10^{13}
800	5.94×10^{-5}	4.26×10^{13}
900	2.69×10^{-3}	1.82×10^{15}
1000	5.61×10^{-2}	3.59×10^{16}

At steady state, Eq. (1) reduces to

$$\nabla \cdot (\lambda(T) \nabla T) + \rho_e(T) |\mathbf{j}(x, z; I, l)|^2 - Q_{\text{loss}}(T) = 0.$$

The in-plane free boundaries are thermally insulated. Radiation and evaporation act through Q_{loss} , not through those edge conditions.

2.4. Evaporation flux from the temperature field

Once the stationary temperature is known, the thulium flux follows directly from the local surface temperature.

For solid thulium, the saturated vapor pressure is estimated using the CRC vapor-pressure relation [9]

$$\log_{10} \left(\frac{P_{\text{sat}}}{\text{Pa}} \right) = 13.888 - \frac{12270}{T} - 0.9564 \log_{10} T, \quad 298 \text{ K} \leq T \leq 1400 \text{ K}. \quad (3)$$

We calculate the equilibrium flux from the Hertz–Knudsen relation

$$\Gamma(T) = \frac{P_{\text{sat}}(T)}{\sqrt{2\pi m_{\text{Tm}} k_B T}},$$

where k_B is Boltzmann's constant and $m_{\text{Tm}} = 168.934 \text{ 21 u}$ is the atomic mass of thulium. We set the evaporation coefficient to unity, so $\Gamma(T)$ is an equilibrium upper estimate.

Table 1 shows the strong temperature dependence. Values below 298 K are formal extrapolations of Eq. (3).

For a prescribed wing length l and electric current I , the total evaporation flux from one large surface of the foil is defined as

$$\Phi(l; I) = \int_{\Omega_l} \Gamma(T_{l,I}(x, z)) \, dx \, dz.$$

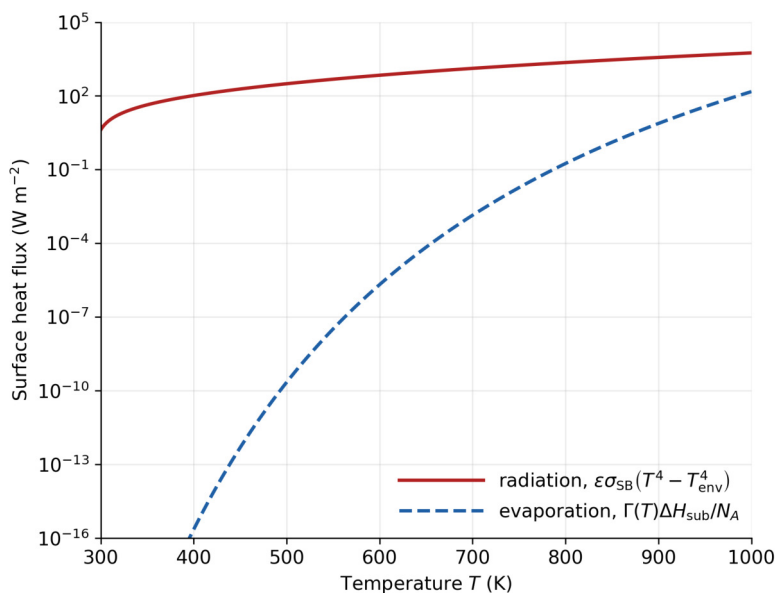


Figure 2. Radiative and evaporative surface heat fluxes of the thulium foil versus temperature. Solid line: radiation, $\varphi_{\text{rad}} = \varepsilon \sigma_{\text{SB}}(T^4 - T_{\text{env}}^4)$ with $\varepsilon = 0.10$; dashed line: evaporation, $\varphi_{\text{evap}} = \Gamma(T) \Delta H_{\text{sub}}/N_A$, with $\Gamma(T)$ from the Hertz-Knudsen relation. Below 1000 K radiation exceeds evaporation by more than three orders of magnitude and dominates the surface energy balance

This definition refers to one face of the foil.

Since the numerical calculation is performed only in one quarter of the domain, the full one-side evaporation flux is reconstructed as

$$\Phi(l; I) = 4 \int_{\Omega_l^{(1/4)}} \Gamma(T_{l,I}(x, z)) dx dz. \quad (4)$$

The total from both faces is 2Φ .

The number of atoms that ultimately reach the ionization region will be lower than this equilibrium estimate. It also depends on source geometry, transport conductance, background pressure, and deposition on surrounding surfaces.

2.5. Relative importance of radiation and evaporation

Figure 2 compares the two surface-loss channels. Below 1000 K the radiative flux $\varphi_{\text{rad}} = \varepsilon \sigma_{\text{SB}}(T^4 - T_{\text{env}}^4)$ exceeds the evaporative flux $\varphi_{\text{evap}} = \Gamma(T) \Delta H_{\text{sub}}/N_A$ by at least three orders of magnitude and is therefore the principal loss term. The approximate balance $q_J \simeq \varepsilon \sigma_{\text{SB}} T^4$ gives $T_{\text{max}} \propto \varepsilon^{-1/4}$. In the numerical model, T_{max} changes from 985 K at $\varepsilon = 0.05$ to 700 K at $\varepsilon = 0.20$, compared with 830 K at the baseline value $\varepsilon = 0.10$. This relatively weak fourth-root dependence supports the use of a single effective emissivity.

Although evaporation contributes little to cooling below 1000 K, we retain it because $\Gamma(T)$ is the quantity needed in the subsequent optimization over wing length and operating current. In the regime studied here it uses for diagnostic rather than for cooling.

2.6. Direct problem

For each prescribed pair (l, I) , we compute the transient field $T(x, z, t)$ and its stationary limit $T_{l,I}(x, z)$.

The prescribed wing length is aligned with the computational grid:

$$l = k_l \Delta z, \quad k_l = 0, 1, \dots, N_z,$$

so that the wing edge coincides with a grid line. The current remains fixed during each run.

For the computed stationary temperature field, the following output quantities are evaluated:

$$T_{\max}(l, I) = \max_{(x,z) \in \Omega_l} T_{l,I}(x, z),$$

and the total evaporation rate of the surface (4). Optimization over l and selection of a current subject to a maximum-temperature constraint are left to an outer procedure and are not considered here.

3. Thermophysical coefficients used in the numerical calculations

The model requires $\rho_m(T)$, $\lambda(T)$, $c_p(T)$, and $\rho_e(T)$ over the operating range of the foil. Earlier calculations used room-temperature constants [2]; here we construct temperature-dependent functions from experimental and reference data over approximately 60 K to 800 K.

3.1. Mass density

We reconstructed the density from the linear-expansion data in the TPRC compilation [10], which for thulium draws on the measurements of Singh, Khanduri and Tiang between 90 K and 300 K [11].

Taking

$$\rho_m(293 \text{ K}) = 9.33 \text{ g cm}^{-3},$$

as the reference density, the density at temperature T was calculated as

$$\rho_m(T) = \frac{\rho_m(293 \text{ K})}{[1 + \Delta L(T)/L_0]^3},$$

where $\Delta L(T)/L_0$ is measured relative to room temperature. We assume isotropic expansion, as no crystallographic texture is specified for the polycrystalline foil. The resulting correction is small but is retained in the calculations (Table 2).

Mass density of metallic thulium reconstructed from thermal expansion data [10, 11].

Table 2

T (K)	$\Delta L/L_0$ (%)	ρ_m (g cm ⁻³)
60	-0.163	9.376
78	-0.153	9.373
100	-0.140	9.369
200	-0.083	9.353
293	0.000	9.330
400	0.142	9.290
500	0.266	9.256
600	0.386	9.223
700	0.505	9.190
778	0.599	9.164
800	0.625	9.157

3.2. Thermal conductivity

Single-crystal thulium is anisotropic because of its hexagonal close-packed structure. Ho, Powell and Liley report recommended conductivities parallel and perpendicular to the c axis, together with a polycrystalline value [12]. We use the latter for the foil.

The source values, originally in $\text{W cm}^{-1} \text{K}^{-1}$, were converted using $1 \text{ W cm}^{-1} \text{K}^{-1} = 100 \text{ W m}^{-1} \text{K}^{-1}$. No additional Voigt–Reuss–Hill averaging was applied.

Because the recommended data end near room temperature, the code uses the following smooth continuation rather than a piecewise-constant extension:

$$\lambda(T) = \lambda_0 + A \frac{T_{\text{eff}}^p}{\rho_e(T_{\text{eff}})}, \quad (5)$$

where

$$\lambda_0 = -0.821\,188\,775 \text{ W m}^{-1} \text{K}^{-1}, \quad A = 1.07573654 \times 10^{-7}, \quad p = 0.829022226.$$

The effective temperature is limited by

$$T_{\text{eff}} = \min(T, T_{\lambda}^{\max}), \quad T_{\lambda}^{\max} = \frac{p\rho_0}{b(1-p)},$$

where ρ_0 and b are defined in the resistivity model below. The cap prevents an unphysical decrease beyond the maximum of the fitted curve. This continuation is a modelling assumption introduced to keep the nonlinear coefficient smooth.

3.3. Specific heat capacity

Below 360 K, the heat capacity is taken from the measurements of Jennings, Hill and Spedding [13], which cover 15 K to 360 K.

The original data are given as molar heat capacity $C_{p,m}(T)$ in $\text{J mol}^{-1} \text{K}^{-1}$. They were converted to mass specific heat by

$$c_p(T) = \frac{C_{p,m}(T)}{M_{\text{Tm}}}, \quad M_{\text{Tm}} = 168.94 \text{ g mol}^{-1}.$$

The difference between c_p and c_v is small relative to other model uncertainties, so Eq. (1) uses $c_p(T)$.

Above 360 K, we extend the table using the trend reported in the TPRC compilation [14]. This continuation is less direct than the low-temperature measurements and is used only to cover the foil's operating range. A constant value $c_p = 160 \text{ J kg}^{-1} \text{K}^{-1}$ is retained for comparisons with the earlier constant-coefficient model.

3.4. Electrical resistivity

Electrical resistivity directly controls the Joule source:

$$q_J(x, z, T) = \rho_e(T) |\mathbf{j}(x, z; I, l)|^2.$$

For 60 K to 300 K we use the polycrystalline measurements of Colvin, Legvold and Spedding [15]; these data are more appropriate for a foil of unspecified texture than single-crystal values.

Above 300 K, we extrapolate from the approximately linear trend in the single-crystal measurements of Edwards and Legvold [16], which provide $\rho_a(T)$, $\rho_b(T)$, and $\rho_c(T)$ up to room temperature.

The single-crystal data distinguish the basal-plane and c -axis directions. To obtain a scalar polycrystalline estimate, the conductivity was averaged as

$$\sigma_i(T) = \frac{1}{\rho_i(T)},$$

Table 3
Recommended thermal conductivity of metallic thulium from Ho, Powell and Liley [12]. Original values are given in $\text{W cm}^{-1} \text{K}^{-1}$ and converted here to $\text{W m}^{-1} \text{K}^{-1}$. The column k_{poly} was used in the numerical calculations.

$T \text{ (K)}$	$k_{\parallel c} \text{ (W m}^{-1} \text{K}^{-1}\text{)}$	$k_{\perp c} \text{ (W m}^{-1} \text{K}^{-1}\text{)}$	$k_{\text{poly}} \text{ (W m}^{-1} \text{K}^{-1}\text{)}$
50	14.2	9.18	10.2
58	17.0	10.4	11.0
60	17.9	10.4	11.2
70	21.4	10.9	11.8
80	25.4	10.7	12.7
90	24.8	10.5	13.1
100	24.6	11.1	13.5
123.2	24.4	11.7	14.4
150	24.3	12.6	15.3
173.2	24.8	13.0	15.8
200	23.2	13.4	16.2
223.2	22.1	13.8	16.4
250	20.2	13.9	16.7
273.2	16.8	14.0	16.8
298.2	13.8	14.0	16.9
300	12.3	14.1	16.9
400	—	—	16.9
500	—	—	16.9
600	—	—	16.9
700	—	—	16.9
778	—	—	16.9
800	—	—	16.9

$$\sigma_{\text{eff}}(T) = \frac{2\sigma_{a/b}(T) + \sigma_c(T)}{3},$$

and then inverted:

$$\rho_{\text{eff}}(T) = \frac{1}{\sigma_{\text{eff}}(T)}.$$

The values appear in Table 5; the code represents them by the smooth linear fit

$$\rho_e(T) = \rho_0 + bT,$$

with

$$\rho_0 = 1.662\,774\,21 \times 10^{-7} \, \Omega \, \text{m}, \qquad b = 1.730\,384\,89 \times 10^{-9} \, \Omega \, \text{m} \, \text{K}^{-1}.$$

The same differentiable fit is used in the Joule source and in Eq. (5).

Table 4

Specific heat capacity of metallic thulium used in the numerical calculations. Values below 360 K are based on the measurements of Jennings, Hill and Spedding [13]; the high-temperature continuation is based on reference data [14].

<i>T</i> (K)	<i>C</i> _{<i>p,m</i>} (J mol ^{−1} K ^{−1})	<i>c</i> _{<i>p</i>} (J g ^{−1} K ^{−1})	<i>c</i> _{<i>p</i>} (J kg ^{−1} K ^{−1})
60	24.22	0.143	143
70	24.51	0.145	145
75	24.67	0.146	146
78	24.77	0.147	147
80	24.83	0.147	147
90	25.12	0.149	149
100	25.25	0.150	150
150	25.91	0.153	153
200	26.52	0.157	157
250	26.75	0.158	158
273.15	26.88	0.159	159
298.15	27.02	0.160	160
300	27.03	0.160	160
360	27.37	0.162	162
400	—	0.160–0.165	160–165
500	—	0.165	165
600	—	0.171	171
700	—	0.177	177
778	—	0.180	180
800	—	0.182	182

4. ADI finite-difference scheme on the quarter domain

Symmetry about the longitudinal and transverse axes reduces the calculation to the quarter-domain

$$\Omega_i^{(1/4)} = \{(x,z) : 0 \leq z \leq L, \quad 0 \leq x \leq W_l(z)\}.$$

Zero normal heat flux is imposed at $x = 0$ and $z = 0$. At the terminal,

$$T_{i,N_z} = T_0, \quad 0 \leq i \leq i_{\min},$$

with $T_0 = 78$ K. The other in-plane boundaries are insulated.

A uniform Cartesian grid is introduced in the enclosing rectangle $[0, W_{\max}] \times [0, L]$:

$$\begin{aligned} x_i &= i\Delta x, \quad i = 0, \dots, N_x, \\ z_k &= k\Delta z, \quad k = 0, \dots, N_z. \end{aligned}$$

The grid is chosen so that

$$W_{\min} = i_{\min}\Delta x, \quad W_{\max} = i_{\max}\Delta x, \quad l = k_l\Delta z.$$

Table 5

Electrical resistivity of metallic thulium used in the numerical calculations. Values up to 300 K are based on polycrystalline data [15]; values above 300 K are model extrapolations based on single-crystal data [16].

T (K)	ρ_e ($\mu\Omega$ cm)	ρ_e (Ω m)	Origin
60	26.5	2.65×10^{-7}	polycrystalline data
70	28.0	2.80×10^{-7}	polycrystalline data
78	29.5	2.95×10^{-7}	polycrystalline data
80	30.0	3.00×10^{-7}	polycrystalline data
100	34.0	3.40×10^{-7}	polycrystalline data
120	37.5	3.75×10^{-7}	polycrystalline data
160	44.5	4.45×10^{-7}	polycrystalline data
200	52.0	5.20×10^{-7}	polycrystalline data
240	59.0	5.90×10^{-7}	polycrystalline data
280	65.5	6.55×10^{-7}	polycrystalline data
300	69.0	6.90×10^{-7}	polycrystalline data / joining point
400	86.1	8.61×10^{-7}	extrapolation
500	103.2	1.032×10^{-6}	extrapolation
600	120.4	1.204×10^{-6}	extrapolation
700	137.6	1.376×10^{-6}	extrapolation
778	151.0	1.510×10^{-6}	extrapolation
800	154.7	1.547×10^{-6}	extrapolation

We normally set $i_{\max} = N_x$. Active rows contain

$$i = 0, \dots, i_{\text{end}}(k), \text{ where } i_{\text{end}}(k) = \begin{cases} i_{\max}, & 0 \leq k \leq k_l, \\ i_{\min}, & k_l < k \leq N_z \end{cases},$$

and the active part of a column is

$$k = 0, \dots, k_{\text{end}}(i), \text{ where } k_{\text{end}}(i) = \begin{cases} N_z, & 0 \leq i \leq i_{\min}, \\ k_l, & i_{\min} < i \leq i_{\max} \end{cases}.$$

Before time integration, the current field is stored as

$$j_{i,k}^2 = |\mathbf{j}(x_i, z_k; I, l)|^2.$$

At a Picard iterate $\tilde{T}$, the thermal coefficients are

$$C_{i,k}(\tilde{T}) = \rho_m(\tilde{T}_{i,k})c_p(\tilde{T}_{i,k}),$$

$$\lambda_{i,k}(\tilde{T}) = \lambda(\tilde{T}_{i,k}),$$

and

$$q_{\text{eff},i,k}(\tilde{T}) = j_{i,k}^2 \rho_e(\tilde{T}_{i,k}) - Q_{\text{loss}}(\tilde{T}_{i,k}).$$

Here

$$Q_{\text{loss}}(T_{i,k}^n) = \frac{2}{h} \left[\varepsilon \sigma_{\text{SB}} ((T_{i,k}^n)^4 - T_{\text{env}}^4) + \Gamma(T_{i,k}^n) \frac{\Delta H_{\text{sub}}}{N_A} \right].$$

We discretize diffusion in conservative flux form. At an active node (i, k) , the contributions through the vertical faces are

$$F_{i+1/2,k}^x = \begin{cases} \lambda_{i+1/2,k}^n (T_{i+1,k} - T_{i,k}), & (i+1, k) \in \Omega_l^{(1/4)}, \\ 0, & (i+1, k) \notin \Omega_l^{(1/4)}, \end{cases}$$

and

$$F_{i-1/2,k}^x = \begin{cases} \lambda_{i-1/2,k}^n (T_{i,k} - T_{i-1,k}), & (i-1, k) \in \Omega_l^{(1/4)}, \\ 0, & (i-1, k) \notin \Omega_l^{(1/4)}. \end{cases}$$

Similarly, the fluxes through the horizontal faces are

$$F_{i,k+1/2}^z = \begin{cases} \lambda_{i,k+1/2}^n (T_{i,k+1} - T_{i,k}), & (i, k+1) \in \Omega_l^{(1/4)}, \\ 0, & (i, k+1) \notin \Omega_l^{(1/4)}, \end{cases}$$

and

$$F_{i,k-1/2}^z = \begin{cases} \lambda_{i,k-1/2}^n (T_{i,k} - T_{i,k-1}), & (i, k-1) \in \Omega_l^{(1/4)}, \\ 0, & (i, k-1) \notin \Omega_l^{(1/4)}. \end{cases}$$

When both neighboring nodes are active, face conductivities are arithmetic means:

$$\lambda_{i+1/2,k}^n = \frac{\lambda(T_{i,k}^n) + \lambda(T_{i+1,k}^n)}{2},$$

$$\lambda_{i,k+1/2}^n = \frac{\lambda(T_{i,k}^n) + \lambda(T_{i,k+1}^n)}{2}.$$

If the neighboring node lies outside the foil, the corresponding flux is zero; no ghost temperature is introduced. The corner $(i_{\min}, k_l)$ requires no special condition.

The discrete operators are then written as

$$L_x^{(n)} T_{i,k} = \frac{F_{i+1/2,k}^x - F_{i-1/2,k}^x}{\Delta x^2},$$

and

$$L_z^{(n)} T_{i,k} = \frac{F_{i,k+1/2}^z - F_{i,k-1/2}^z}{\Delta z^2}.$$

Time integration uses a Peaceman–Rachford ADI scheme [17–19], with Picard iteration at each half-step. The first half-step is implicit in x and explicit in z :

$$C_{i,k}^{n+1/2,(s)} \frac{T_{i,k}^{n+1/2,(s+1)} - T_{i,k}^n}{\Delta t/2} = L_x^{n+1/2,(s)} T_{i,k}^{n+1/2,(s+1)} + L_z^n T_{i,k}^n + q_{\text{eff},i,k}^{n+1/2,(s)}. \quad (6)$$

For each fixed k , Eq. (6) gives a tridiagonal system along the active row $i = 0, \dots, i_{\text{end}}(k)$:

$$a_{i,k}^x T_{i-1,k}^{n+1/2,(s+1)} + b_{i,k}^x T_{i,k}^{n+1/2,(s+1)} + c_{i,k}^x T_{i+1,k}^{n+1/2,(s+1)} = r_{i,k}^x. \quad (7)$$

For ordinary active nodes,

$$a_{i,k}^x = -\frac{\lambda_{i-1/2,k}^{n+1/2,(s)}}{\Delta x^2}, \quad c_{i,k}^x = -\frac{\lambda_{i+1/2,k}^{n+1/2,(s)}}{\Delta x^2},$$

$$b_{i,k}^x = \frac{2C_{i,k}^{n+1/2,(s)}}{\Delta t} + \frac{\lambda_{i-1/2,k}^{n+1/2,(s)} + \lambda_{i+1/2,k}^{n+1/2,(s)}}{\Delta x^2},$$

and

$$r_{i,k}^x = \frac{2C_{i,k}^{n+1/2,(s)}}{\Delta t} T_{i,k}^n + L_x^n T_{i,k}^n + q_{\text{eff},i,k}^{n+1/2,(s)}. \quad (8)$$

If a neighboring node in the x direction is absent, the corresponding coefficient $a_{i,k}^x$ or $c_{i,k}^x$ is set to zero. In particular, at $x = 0$ the symmetry condition gives $a_{0,k}^x = 0$, while at the right end of an active row $c_{i_{\text{end}}(k),k}^x = 0$.

The row $k = N_z$ is not advanced by the x -sweep. It belongs to the current terminal for $0 \leq i \leq i_{\text{min}}$ and is fixed as

$$T_{i,N_z}^{n+1/2} = T_0, \quad 0 \leq i \leq i_{\text{min}}.$$

The second half-step is implicit in z . The accepted intermediate field provides the explicit x contribution, while Picard iteration updates the implicit coefficients:

$$C_{i,k}^{n+1,(s)} \frac{T_{i,k}^{n+1,(s+1)} - T_{i,k}^{n+1/2}}{\Delta z/2} = L_z^{n+1,(s)} T_{i,k}^{n+1,(s+1)} + L_x^{n+1/2} T_{i,k}^{n+1/2} + q_{\text{eff},i,k}^{n+1,(s)}. \quad (9)$$

For each fixed i , Eq. (9) gives a tridiagonal system along the active column $k = 0, \dots, k_{\text{end}}(i)$:

$$a_{i,k}^z T_{i,k-1}^{n+1,(s+1)} + b_{i,k}^z T_{i,k}^{n+1,(s+1)} + c_{i,k}^z T_{i,k+1}^{n+1,(s+1)} = r_{i,k}^z. \quad (10)$$

For ordinary active nodes,

$$a_{i,k}^z = -\frac{\lambda_{i,k-1/2}^{n+1,(s)}}{\Delta z^2}, \quad c_{i,k}^z = -\frac{\lambda_{i,k+1/2}^{n+1,(s)}}{\Delta z^2},$$

$$b_{i,k}^z = \frac{2C_{i,k}^{n+1,(s)}}{\Delta t} + \frac{\lambda_{i,k-1/2}^{n+1,(s)} + \lambda_{i,k+1/2}^{n+1,(s)}}{\Delta z^2},$$

and

$$r_{i,k}^z = \frac{2C_{i,k}^{n+1,(s)}}{\Delta t} T_{i,k}^{n+1/2} + L_x^{n+1/2} T_{i,k}^{n+1/2} + q_{\text{eff},i,k}^{n+1,(s)}.$$

At $z = 0$, the symmetry condition gives $a_{i,0}^z = 0$. For columns inside the side wing, $i > i_{\text{min}}$, the active segment ends at $k = k_l$, and the upper face is insulated; therefore $c_{i,k_l}^z = 0$. For columns inside the central part, $0 \leq i \leq i_{\text{min}}$, the active segment extends to $k = N_z$, where the Dirichlet condition is imposed by replacing the last equation with

$$T_{i,N_z}^{n+1} = T_0, \quad 0 \leq i \leq i_{\text{min}}.$$

Equivalently,

$$a_{i,N_z}^z = 0, \quad b_{i,N_z}^z = 1, \quad c_{i,N_z}^z = 0, \quad r_{i,N_z}^z = T_0.$$

The Picard iteration in each half-step is stopped when both an absolute and a relative update criterion are satisfied,

$$\max_{(i,k) \in \Omega_l^{(1/4)}} |T_{i,k}^{(s+1)} - T_{i,k}^{(s)}| < \varepsilon_{\text{nl,abs}},$$

$$\frac{\max_{(i,k) \in \Omega_l^{(1/4)}} |T_{i,k}^{(s+1)} - T_{i,k}^{(s)}|}{\max \left(1, \max_{(i,k) \in \Omega_l^{(1/4)}} |T_{i,k}^{(s+1)}| \right)} < \varepsilon_{\text{nl,rel}}. \quad (11)$$

After each full step, stationarity is tested with the maximum temperature rate,

$$R_T^n = \max_{(i,k) \in \Omega_l^{(1/4)}} \left| \frac{T_{i,k}^{n+1} - T_{i,k}^n}{\Delta t} \right|. \quad (12)$$

The loop ends when $R_T^n < \varepsilon_{\text{rate}}$ or $t_n \geq t_{\text{max}}$. The default $t_{\text{max}} = \text{DBL_MAX} - 10\Delta t$ effectively disables the time limit, so production runs normally stop on the rate criterion. The elapsed physical time then estimates the foil's approach time to steady operation.

5. GPU implementation of the ADI solver

We implemented the ADI scheme in CUDA C for a single GPU. The row systems in Eq. (7) and the column systems in Eq. (10) are independent within their respective half-steps. We compare several ways of solving these systems and dispatching the associated kernels; all produce the same discrete solution. Figure 3 summarizes the implementation.

5.1. Overall organization

All field arrays, including the two temperature layers, squared current density $j_{i,k}^2$, and the temperature-dependent coefficients $C_{i,k}$, $\lambda_{i,k}$ and $q_{\text{eff},i,k}$, are stored on the device in row-major order over the enclosing rectangle, $\text{IDX}(i, k) = k(N_x + 1) + i$. The auxiliary electric-current problem is linear and is solved once on the host before the time loop; only the resulting field $j_{i,k}^2$ is copied to the device. The transient integration then runs entirely on the GPU. Host-device transfers during integration are limited to the maximum temperature change and maximum temperature, and occur only when a convergence test is evaluated.

Each time step is carried out by a small set of kernels: a coefficient kernel that evaluates $C_{i,k}$, $\lambda_{i,k}$ and $q_{\text{eff},i,k}$ from the current temperature estimate with one thread per node; the two sweep kernels that assemble and solve the tridiagonal systems of the first and second half-steps; and reduction kernels that return rates for the Picard test (11) and for the stationary-rate test (12). The two temperature layers are kept in two device buffers that are exchanged by pointer swap after each accepted step, so that no full-field copy is performed inside the time loop.

5.2. Mapping of the tridiagonal systems onto the device

Each tridiagonal system is assigned to *one* CUDA block, and the threads of that block cooperate on that single system rather than each thread owning a complete system. This keeps the three diagonals and the right-hand side of a given system in a contiguous span of device memory, so that the parallel assembly and scatter phases issue coalesced accesses, and it allows the block to use `__syncthreads()` to separate the assembly, the solve and the scatter phases. The block dimension is fixed at 256 threads.

Because the quarter domain is L-shaped, the systems do not all have the same length. The rows split into long rows in the wide central part ($0 \leq k \leq k_l$, length $i_{\text{max}} + 1$) and short rows in the side wing ($k_l < k < N_z$, length $i_{\text{min}} + 1$); the columns split analogously into long columns in the central part ($0 \leq i \leq i_{\text{min}}$, length $N_z + 1$) and short columns in the side wing ($i_{\text{min}} < i \leq i_{\text{max}}$, length $k_l + 1$). Each group is handled by a separate kernel launch, so that a launch processes a batch of equal-length systems and the storage for that batch is sized to the exact system length rather than to the largest system in the domain. For the grid used here ($N_x = 50$, $N_z = 400$, $i_{\text{max}} = 50$, $i_{\text{min}} = 20$, $k_l = 280$) the four batches contain 281, 119, 21 and 30 systems of length 51, 21, 401 and 281, respectively. For a batch of N_{sys} systems of length n_{max} , four device arrays of size $N_{\text{sys}} n_{\text{max}}$ hold the sub-, main- and super-diagonals and the right-hand side; system s occupies the contiguous span starting at $s n_{\text{max}}$, and its solution overwrites the right-hand side in place.

5.3. Assembly, solve and scatter within a block

Within a block the work proceeds in three barrier-separated phases. In the *assembly* phase the threads stride over the active nodes of the system, $\text{count} = \text{tid}, \text{tid} + \text{blockDim}, \dots$, and each thread writes the coefficients a, b, c and the right-hand side r of its nodes according to Eqs. (7)–(8) for the x -sweep and the analogous relations for the z -sweep. The face conductivities are evaluated by arithmetic averaging of the neighbouring nodal values, and a neighbour outside the quarter domain contributes a zero flux, so that the corresponding off-diagonal coefficient vanishes. The symmetry conditions at $x = 0$ and $z = 0$ enter as a vanishing lower diagonal at $\text{count} = 0$, and the end of each active segment as a vanishing upper diagonal at $\text{count} = n - 1$; the Dirichlet terminal node (i, N_z) with $i \leq i_{\text{min}}$ is

assembled directly as $a^z = 0$, $b^z = 1$, $c^z = 0$, $r^z = T_0$, i.e. the last equation of the affected columns is replaced rather than eliminated.

After a barrier the *solve* phase reduces each system to its solution, and after a second barrier the *scatter* phase writes the computed line back into the two-dimensional temperature array, again with the threads striding over the nodes of the system. The same kernel structure is used for both half-steps; only the index running along the system (the row index i for the x -sweep, the column index k for the z -sweep) and the direction of the implicit and explicit operators differ. The assembly and scatter phases are identical across all variants considered below; the variants differ only in the algorithm that implements the solve phase.

5.4. Tridiagonal-solve algorithms

We compared four solvers that differ in how elimination is distributed across the block [20]. Their solutions agree to round-off.

(i) Serial Thomas sweep (baseline). Thread 0 executes the classical Thomas algorithm [21]: forward elimination followed by backward substitution. A pivot smaller than 10^{-300} marks the solution as NaN. The guard was never triggered because the transient term $2C_{i,k}/\Delta t$ makes these systems strongly diagonally dominant. This implementation matches the serial CPU sweep bit-for-bit, but the device is under-utilized during this phase because the other block threads remain idle.

(ii) Two-sided Thomas algorithm. This variant applies Thomas elimination simultaneously from both ends of the system [22, 23]. The two elimination fronts advance towards a central index, after which substitution proceeds outwards. Two threads per block carry out the sweeps, halving the length of the inherently sequential dependency chain.

(iii) Austin-Berndt-Moulton partition method. Production runs use the memory-efficient partitioning scheme of Austin, Berndt, and Moulton (ABM) [24]. A system of N rows is divided into N_p contiguous partitions, each of length $N_k \approx N/N_p$. Each thread eliminates the interior unknowns of one partition and expresses them in terms of two interface unknowns. These interface unknowns form a reduced tridiagonal system of order $2N_p$. After this system is solved by thread 0 using the serial Thomas algorithm, the threads recover the interior unknowns of their respective partitions in parallel. The ABM method stores only the boundary-coupling coefficients rather than the full fill-in vectors, requiring approximately 6 kB of shared memory in our kernels. The partition method is applied to systems with $N > 100$. Smaller systems containing 21 and 51 unknowns, are solved using two-sided elimination.

(iv) Vendor batched solver (cuSPARSE). For an external comparison, the batched solve was delegated to the NVIDIA cuSPARSE library [25] routine `cusparseDgtsv2StridedBatch`, which solves a batch of equal-length tridiagonal systems stored in the strided layout already used by the assembly phase. This variant reuses the same assembly and scatter kernels and replaces only the solve, so it measures the cost of a production-quality library solver on the identical data layout.

5.5. Kernel launch strategy

We also compared two launch strategies. In *plain* mode every kernel of a step – the coefficient kernel, the four assembly/solve batches of each sweep, and the reductions – is launched individually through the ordinary runtime API. Because a single step of the present problem issues of the order of a few tens of very short kernels (each system is at most 401 unknowns long, so a kernel runs only for microseconds), the fixed CPU-side cost of launching each kernel becomes comparable to the kernel's own execution time, and over the $\sim 8 \times 10^4$ steps of a full run this launch overhead accumulates into a noticeable fraction of the wall time.

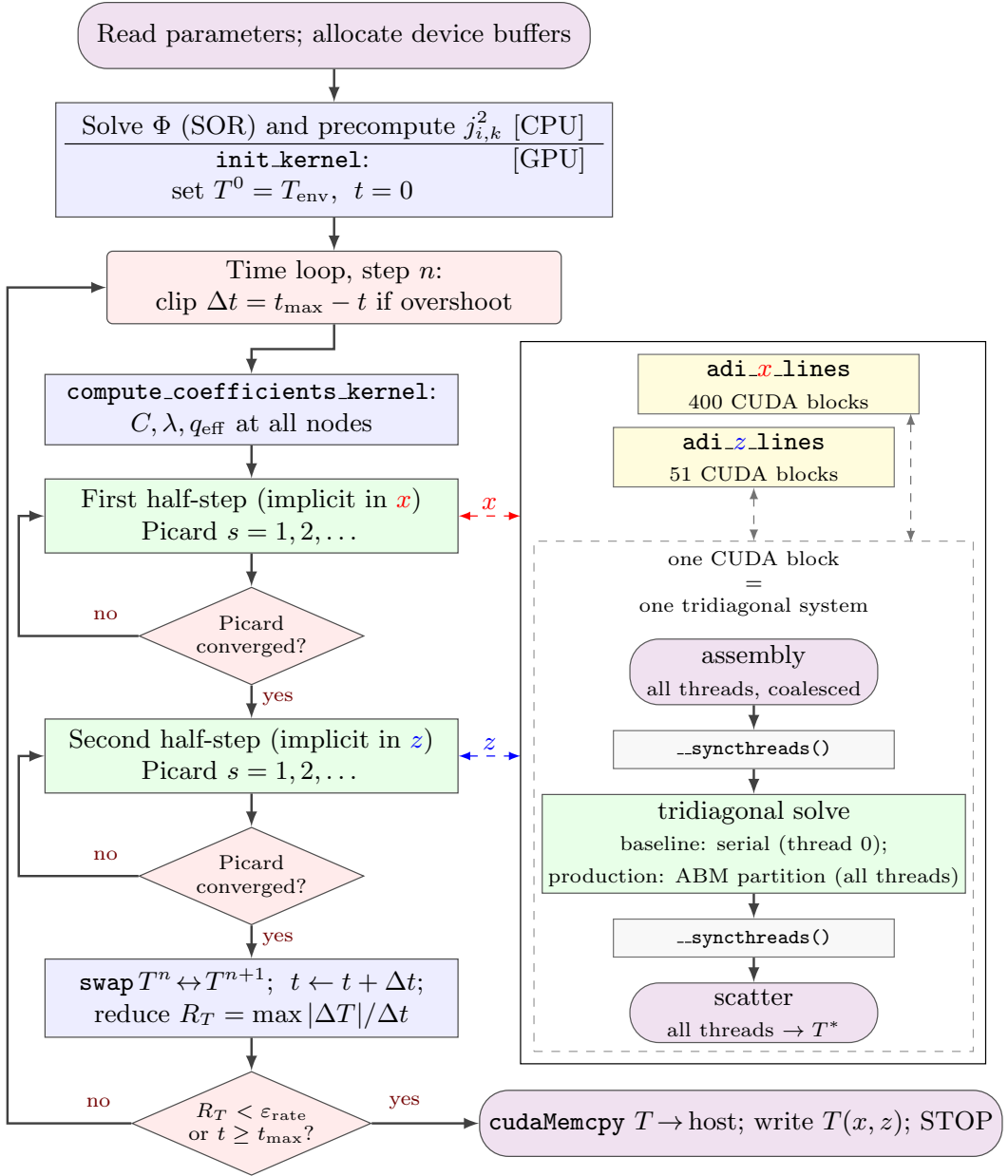


Figure 3. Structure of the GPU implementation of the ADI solver. The host solves the electric problem once; the transient loop runs entirely on the device. Each ADI half-step launches one kernel per batch of equal-length tridiagonal systems (the x -sweep is shown exploded on the right: 281 long rows of length 51 and 119 short rows of length 21; the z -sweep is organized analogously into 21 long and 30 short columns). Every tridiagonal system is mapped to a single CUDA block, whose threads cooperate in three barrier-separated phases - parallel assembly, the tridiagonal solve, and parallel scatter of the solution. The solve phase is interchangeable: the baseline runs a serial Thomas sweep in thread 0, whereas the production configuration distributes it across the threads of the block through the ABM partition method (Section 5)

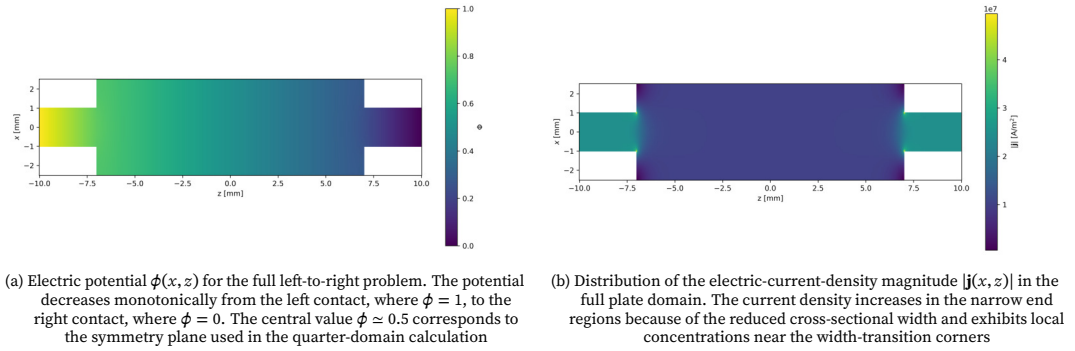


Figure 4. Electric properties of the full plate geometry: (a) electric potential and (b) magnitude of the electric current density

The second mode uses *CUDA task graphs* [26]. The kernel-call flow within one time step is represented as a task graph and submitted as a single GPU operation. We construct the graph implicitly by capturing the stream workflow with `cudaStreamBeginCapture` and `cudaStreamEndCapture`, instantiate it with `cudaGraphInstantiate`, and execute it with `cudaGraphLaunch`. This approach is particularly effective for short kernels, whose execution time can be comparable to their launch latency, because it reduces the gaps between successive kernel launches [27]. Since Δt , the grid, and the batch structure remain fixed, the same graph is used throughout a run. A shortened final step, if needed to reach $t_{\max}$ exactly, is launched separately.

5.6. Time loop and stopping

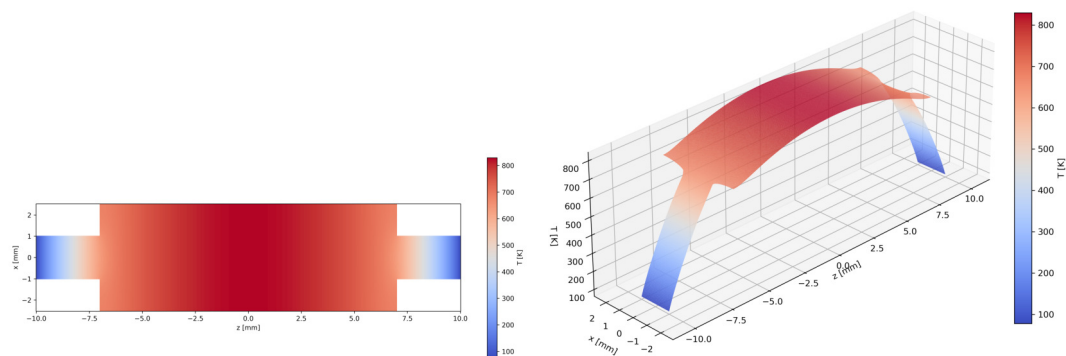
The Picard iterations of each half-step reuse the coefficient and sweep kernels: the coefficients are re-evaluated from the latest temperature estimate, the tridiagonal systems are re-assembled and re-solved, and the maximum update is reduced on the device. To limit host-device synchronization, the update norm is copied back and tested only on every third iteration and on the last allowed iteration. After a full step the maximum temperature change is reduced on the device and divided by Δt to form the stationary rate R_T^n of Eq. (12). The time loop terminates when $R_T^n < \epsilon_{\text{rate}}$ or when the accumulated physical time reaches the prescribed bound $t_{\max}$; when the next step overshoot $t_{\max}$, the step size is reduced to $\Delta t = t_{\max} - t_n$ for that final step in order to land exactly on $t_{\max}$.

6. Numerical results

We solved the coupled problem for a total current $I = 2.5$ A, with the cold contacts held at $T_0 = 78$ K, the surrounding medium at $T_{\text{env}} = 293$ K, and a surface emissivity $\epsilon = 0.10$. The grid used $N_x = 50$, $N_z = 400$ (`grid_scale` = 1), with the wing transition at $k_l = 280$ and $i_{\min} = 20$. The time step was $\Delta t = 1 \times 10^{-3}$ s, and each Picard update used at most 20 iterations per half-step with tolerances $\epsilon_{nl, \text{abs}} = 10^{-6}$ and $\epsilon_{nl, \text{rel}} = 10^{-10}$.

The potential decreases monotonically between the contacts (Figure 4a). As expected, the current density is largest in the narrow end regions and shows additional crowding near the re-entrant corners (Figure 4b). This nonuniform $|j|^2$ field is used in the Joule source.

Starting from $T_0 = 78$ K, the calculation reached the criterion $R_T = \max_{i,k} |T_{i,k}^{n+1} - T_{i,k}^n| / \Delta t < \epsilon_{\text{rate}}$ with $\epsilon_{\text{rate}} = 1 \times 10^{-2}$ K s⁻¹ after 77 397 steps, or 77.397 s of simulated time. The maximum temperature was 829.803 K. Figures 5a and 5b show the stationary field. The cold contacts keep the narrow ends near 78 K, while the maximum lies in the broad central region. Changes in the field near the width transitions reflect the redistribution of current there.



(a) Two-dimensional temperature map of the stationary regime in the full symmetric thulium plate domain. The narrow end regions are maintained close to the prescribed contact temperature $T_0 = 78$ K, whereas the central wide part reaches the highest temperature

(b) Three-dimensional representation of the stationary temperature field $T(x, z)$ in the full symmetric thulium plate domain. Local distortions near the width-transition zones reflect current redistribution and the associated spatial variation of Joule heating

Figure 5. Stationary temperature distribution in the full symmetric thulium plate domain: a two-dimensional temperature map and b three-dimensional representation of the temperature field. The temperature decreases toward the narrow cold-contact regions, while the maximum occurs in the central part of the wider section

All tridiagonal solvers produced the same converged field and maximum temperature, allowing their execution times to be compared directly.

7. Performance of the tridiagonal-solver strategies

We benchmarked the batched solvers on one NVIDIA V100 against a serial CPU implementation. Every variant used $N_x = 50$, $N_z = 400$, $k_l = 280$, and $\Delta t = 1 \times 10^{-3}$ s, stopped at step 77 397, and produced a bit-for-bit identical $T_{\max} = 829.803$ K. Wall-clock differences therefore arise from the solver and launch strategy rather than from different stopping histories.

The tested configurations comprise the one-thread Thomas baseline, the two-sided Thomas algorithm, the ABM partition solver, and `cusparseDgtsv2StridedBatch`. Each in-house solver was measured with ordinary kernel launches and with CUDA task graphs. We also tested device intrinsics in the partition sweep. Each configuration was run five times; Table 6 reports the mean and standard deviation; Figure 6 and 7 show the same data graphically.

The serial CPU run took 876 s. ABM partitioning achieved 21–24 $\times$ speedup (Figure 7), roughly twice that of the serial and two-sided Thomas algorithms (11–12 $\times$), because the elimination work is shared across the block. CUDA task graphs reduced runtime by about ten percent over the 77 397 time steps. Explicit use of FMA intrinsics produced no speedup because `nvcc` enables contraction of eligible multiply-add operations by default (`-fmad=true`). The fastest configuration, ABM partitioning with a CUDA task graph, completed in 36.3 s (24.1 $\times$; Figure 6), about ten percent faster than `cuSPARSE` at 40.2 s (21.8 $\times$).

The GPU computations were performed on an NVIDIA Tesla V100-SXM2 (16 GB) accelerator of the “Govorun” supercomputer at the HybriLIT heterogeneous computing platform [28], using code built with the CUDA 11.2 toolkit and GCC 9.5.0 as the host compiler (`-arch=sm_70 -O3`). For comparison, the CPU version was run on the same platform, on Intel Xeon Platinum 8268 processors (24 cores, 2.9 GHz, 192 GB DDR4) and built with the same compiler.

8. Discussion

The calculated temperature field reflects the combined effects of the nonuniform current distribution and heat removal through the cold contacts. The maximum temperature occurs in the broad central

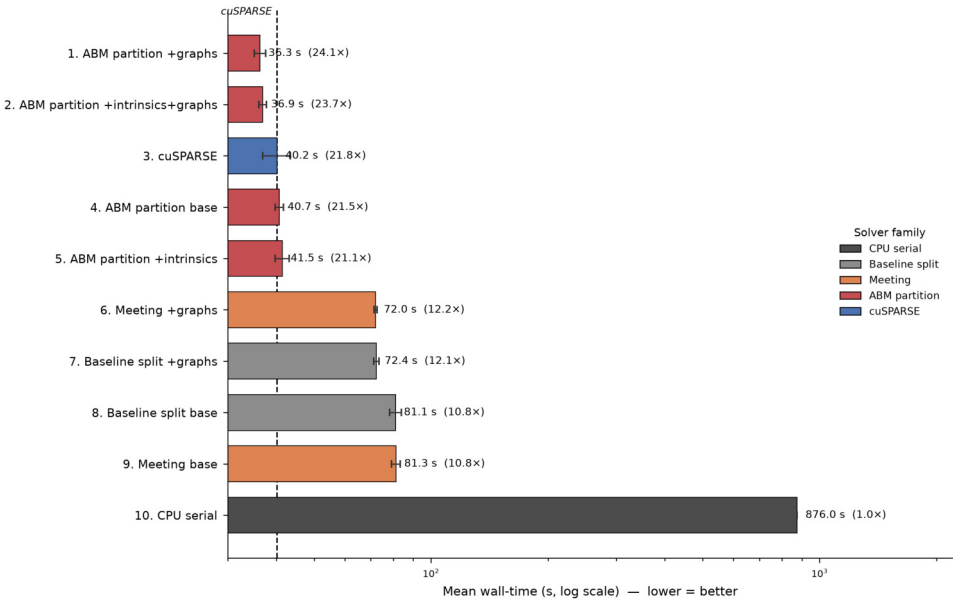


Figure 6. Mean wall-clock time of the tridiagonal-solver configurations on an NVIDIA V100 (logarithmic scale; lower is better; error bars show the standard deviation over five runs). The dashed line shows the cuSPARSE timing. Colours distinguish the tridiagonal algorithms. All configurations converge to the same $T_{\max} = 829.803$ K at step 77 397

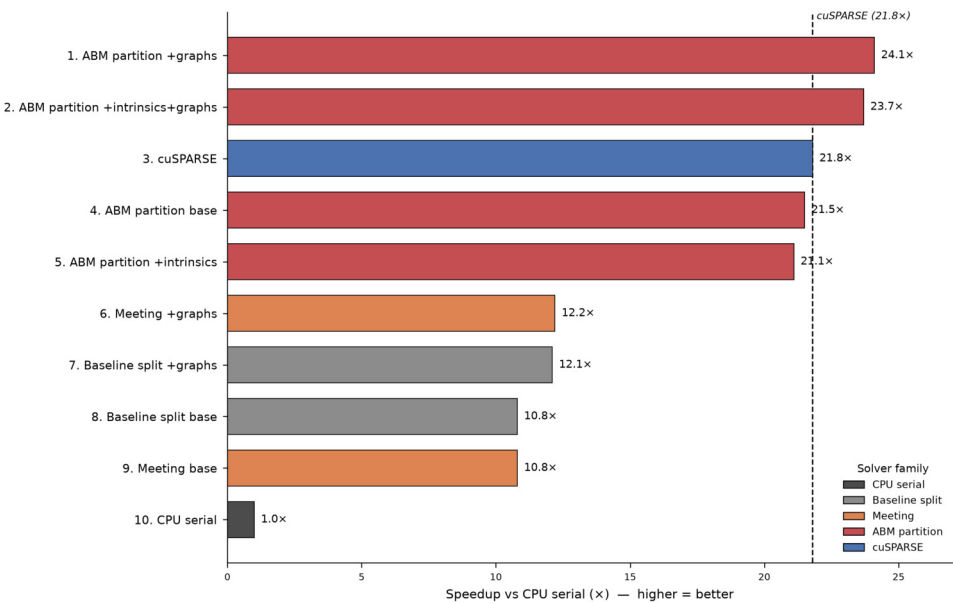


Figure 7. Speedup of the GPU configurations relative to the single-thread CPU run (linear scale; higher is better). The dashed line shows the cuSPARSE result. The ABM partition configurations reach 21-24x speedup; the two configurations that use CUDA task graphs outperform the vendor routine

Table 6

Wall-clock time (mean $\pm$ standard deviation over five runs) and speedup relative to the single-thread CPU run for the tridiagonal-solver strategies on an NVIDIA V100. All configurations converge at step 77 397 to $T_{\text{max}} = 829.803$ K.

Configuration	Wall time (s)	Speedup
ABM partition, + task graph	36.3 $\pm$ 1.2	24.1 $\times$
ABM partition, + intrinsics + task graph	36.9 $\pm$ 0.8	23.7 $\times$
cuSPARSE	40.2 $\pm$ 3.3	21.8 $\times$
ABM partition, plain	40.7 $\pm$ 1.0	21.5 $\times$
ABM partition, + intrinsics	41.5 $\pm$ 1.7	21.1 $\times$
Two-sided Thomas, + task graph	72.0 $\pm$ 0.6	12.2 $\times$
Serial Thomas, + task graph	72.4 $\pm$ 1.1	12.1 $\times$
Serial Thomas, plain	81.1 $\pm$ 2.8	10.8 $\times$
Two-sided Thomas, plain	81.3 $\pm$ 2.1	10.8 $\times$
CPU serial	876.0	1.0 $\times$

region, whereas the narrow end regions remain close to the prescribed contact temperature. The local changes in the temperature field near the width transitions are associated with current crowding at the re-entrant corners. This behaviour confirms the need for a two-dimensional electric model, since the one-dimensional approximation cannot resolve these local variations in the current density.

The present results should be considered in the context of the model assumptions. The electric-current distribution is calculated using constant conductivity and is subsequently scaled to the prescribed total current, while the temperature dependence of the electrical resistivity is included in the Joule heat source. In addition, a constant effective emissivity is used, and some thermophysical properties are extrapolated beyond the available experimental ranges. The numerical results therefore characterize the selected foil geometry and operating conditions rather than providing a universal optimum.

The ADI discretization produces batches of independent tridiagonal systems. Mapping each system to a CUDA block makes it possible to change the solution algorithm without modifying the assembly procedure or data layout. For the grid considered here, ABM partitioning approximately doubled the performance relative to the serial and two-sided Thomas algorithms. CUDA task graphs provided a further improvement of about ten percent, whereas device intrinsics changed the execution time by less than the run-to-run variation. The fastest configuration, ABM partitioning combined with a CUDA task graph, completed the calculation in 36.3 s and achieved a 24.1 $\times$ speedup over the single-thread CPU implementation. It was also approximately ten percent faster than cuSPARSE.

Compared with the earlier formulation [2], which ranked candidate geometries according to the area within a prescribed temperature interval, the present model provides the evaporation flux as a more direct measure of ion-source performance. The computational efficiency of the GPU implementation makes the direct solver suitable for repeated calculations in an optimization procedure in which the foil geometry and operating current are varied. Future development of the model will couple the electric field to the evolving temperature-dependent conductivity.

9. Conclusions

A two-dimensional electric-thermal model of a wing-shaped thulium foil was developed. The model accounts for temperature-dependent material properties, nonuniform Joule heating, radiation, and evaporation. The numerical results show that the proposed approach can determine the stationary temperature field and provide the quantities needed to evaluate the foil performance. All tested

tridiagonal solvers produced the same converged solution, while the GPU implementation substantially reduced the computation time. The resulting solver can therefore be used for repeated calculations when optimizing the foil geometry and operating current.

Author Contributions: A. Ayriyan: Conceptualization (physical and mathematical model, formulation of mathematical problem), Methodology, Software, Investigation (numerical computations), Writing, Supervision. E. Ayryan: Methodology (numerical solution of the elliptic equation). E. E. Donets: Conceptualization (physical problem). M. Veneva: Software (parallel tridiagonal solver and CUDA implementation), Methodology (GPU optimization). All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: All data and parameters required to reproduce the results of this study are provided within the article. No additional datasets are required.

Acknowledgments: The authors thank Hovik Grigorian and Jan Buša Jr. for helpful discussions and technical assistance. Computations were held on the basis of the HybriLIT heterogeneous computing platform (MLIT, JINR).

Conflicts of Interest: The authors declare no conflict of interest.

Declaration on Generative AI: Claude (Anthropic) was used solely to improve the language and readability of text originally written by the authors and to assist in preparing Figures 6 and 7 from the numerical data reported in Table 6. The authors reviewed and approved all AI-assisted changes and take full responsibility for the content of the manuscript.

References

- [1] D. E. Donets, E. D. Donets, E. E. Donets, V. V. Salnikov, and V. B. Shutov, "Production of highly charged ion beams Kr32+, Xe44+, Au54+ with Electron String Ion Source (ESIS) Krion-2 and corresponding basic and applied studies," *JINST*, vol. 5, p. C09001, 2010. DOI: 10.1088/1748-0221/5/09/C09001
- [2] A. Ayriyan, E. Donets, H. Grigorian, N. Kolkovska, and A. Lebedev, "Algorithm for Solving an Optimization Problem for the Temperature Distribution on a Plate," *EPJ Web of Conferences*, vol. 108, p. 02 010, 2016. DOI: 10.1051/epjconf/201610802010
- [3] A. H. Persad and C. A. Ward, "Expressions for the Evaporation and Condensation Coefficients in the Hertz-Knudsen Relation," *Chemical Reviews*, vol. 116, no. 14, pp. 7727–7767, 2016. DOI: 10.1021/acs.chemrev.5b00511
- [4] Y. B. Zudin, *Non-equilibrium Evaporation and Condensation Processes*. Cham, Switzerland: Springer, 2017. DOI: 10.1007/978-3-319-67306-6
- [5] V. P. Carey, *Liquid-Vapor Phase-Change Phenomena*, 3rd ed. Boca Raton, FL: CRC Press, 2020.
- [6] D. M. Young, "Iterative Methods for Solving Partial Difference Equations of Elliptic Type," *Transactions of the American Mathematical Society*, vol. 76, no. 1, pp. 92–111, 1954. DOI: 10.1090/S0002-9947-1954-0059635-7
- [7] A. A. Samarskii and E. S. Nikolaev, *Numerical Methods for Grid Equations, Volume II: Iterative Methods*. Basel: Birkhäuser, 1989. DOI: 10.1007/978-3-0348-9142-4
- [8] T. L. Bergman, A. S. Lavine, F. P. Incropera, and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 8th ed. Hoboken, NJ: Wiley, 2017.
- [9] D. R. Lide, Ed., *CRC Handbook of Chemistry and Physics*, 84th ed. Boca Raton: CRC Press, 2003, Section 6, Fluid Properties; vapor-pressure data for the elements.
- [10] Y. S. Touloukian, R. K. Kirby, R. E. Taylor, and P. D. Desai, Eds., *Thermal Expansion: Metallic Elements and Alloys* (Thermophysical Properties of Matter: The TPRC Data Series). New York: IFI/Plenum, 1975, vol. 12.
- [11] S. Singh, N. C. Khanduri, and T. Tiang, "Thermal Expansions of Thulium and Lutetium, 90–300 K," *Solid State Communications*, vol. 5, no. 3, pp. 167–173, 1971.
- [12] C. Y. Ho, R. W. Powell, and P. E. Liley, "Thermal Conductivity of the Elements," *Journal of Physical and Chemical Reference Data*, vol. 1, no. 2, pp. 279–421, 1972.
- [13] L. D. Jennings, E. Hill, and F. H. Spedding, "Heat Capacity of Thulium from 15 Degrees to 360 Degrees K," *The Journal of Chemical Physics*, vol. 34, no. 6, pp. 2082–2088, 1961. DOI: 10.1063/1.1731825

- [14] Y. S. Touloukian and E. H. Buyco, Eds., *Specific Heat: Metallic Elements and Alloys* (Thermophysical Properties of Matter: The TPRC Data Series). New York: IFI/Plenum, 1970, vol. 4.
- [15] R. V. Colvin, S. Legvold, and F. H. Spedding, “Electrical Resistivity of the Heavy Rare-Earth Metals,” *Physical Review*, vol. 120, no. 3, pp. 741–745, 1960. DOI: 10.1103/PhysRev.120.741
- [16] L. R. Edwards and S. Legvold, “Transport Properties of Thulium Single Crystals,” *Physical Review*, vol. 176, no. 3, pp. 753–758, 1968. DOI: 10.1103/PhysRev.176.753
- [17] D. W. Peaceman and H. H. Rachford, “The Numerical Solution of Parabolic and Elliptic Differential Equations,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 3, no. 1, pp. 28–41, 1955. DOI: 10.1137/0103003
- [18] A. A. Samarskii and P. N. Vabishchevich, *Computational Heat Transfer*. Chichester; New York: Wiley, 1995, vol. 1–2.
- [19] A. A. Samarskii, *The Theory of Difference Schemes*. New York: Marcel Dekker, 2001, Russian original: *Teoriya raznostnykh skhem*, Nauka, Moscow, 1977.
- [20] Y. Zhang, J. Cohen, and J. D. Owens, “Fast Tridiagonal Solvers on the GPU,” *ACM SIGPLAN Notices (PPoPP '10)*, vol. 45, no. 5, pp. 127–136, 2010. DOI: 10.1145/1837853.1693472
- [21] L. H. Thomas, “Elliptic Problems in Linear Difference Equations over a Network,” *Watson Sci. Comput. Lab. Report*, Columbia University, New York, 1949.
- [22] I. Babuška, “Numerical Stability in Problems of Linear Algebra,” *SIAM Journal on Numerical Analysis*, vol. 9, no. 1, pp. 53–77, 1972. DOI: 10.1137/0709008
- [23] H. A. van der Vorst, “Large Tridiagonal and Block Tridiagonal Linear Systems on Vector and Parallel Computers,” *Parallel Computing*, vol. 5, no. 1–2, pp. 45–54, 1987. DOI: 10.1016/0167-8191(87)90005-6
- [24] T. M. Austin, M. Berndt, and J. D. Moulton, “A Memory Efficient Parallel Tridiagonal Solver,” Los Alamos National Laboratory, Tech. Rep. LA-UR-03-4149, 2004.
- [25] N. Corporation, *cuSPARSE Library*, <https://docs.nvidia.com/cuda/cusparse/>, NVIDIA CUDA Toolkit documentation.
- [26] N. Corporation, *CUDA C++ Programming Guide*, <https://docs.nvidia.com/cuda/cuda-c-programming-guide/>, Section on CUDA Graphs.
- [27] M. Veneva, “GPU-Based Optimizations for waLBerla: Porting waLBerla to CUDA Task Graphs,” in *JSC Guest Student Programme Proceedings 2021*, I. Kabadshow, Ed., ser. IAS Series, vol. 52, Forschungszentrum Jülich GmbH, Zentralbibliothek, Verlag, 2023, pp. 59–82.
- [28] A. Anikina, D. Belyakov, T. Bezhanlyan, et al., “Structure and Features of the Software and Information Environment of the HybriLIT Heterogeneous Platform,” in *Distributed Computer and Communication Networks (DCCN 2024)*, ser. Lecture Notes in Computer Science, vol. 15460, Springer, Cham, 2025, pp. 444–457. DOI: 10.1007/978-3-031-80853-1_33

Information about the authors

Ayriyan, Alexander S.—Candidate of Physical and Mathematical Sciences, Head of the Nonlinear Systems Modeling Methods Sector, Scientific Department of Computational Physics, Meshcheryakov Laboratory of Information Technologies (MLIT), Joint Institute for Nuclear Research (JINR); Research Scientist at the Division of Computational Physics and Information Technologies, A. Alikhanyan National Science Laboratory (Yerevan) (e-mail: ayriyan@jinr.ru, ORCID: 0000-0002-5464-4392)

Ayryan, Edik A.—Candidate of Physical and Mathematical Sciences, Assistant to the Director of the Meshcheryakov Laboratory of Information Technologies (MLIT), Joint Institute for Nuclear Research (JINR) (e-mail: ayryan@jinr.ru, ORCID: 0000-0001-6478-4041)

Donets, Eugeny E.—Candidate of Physical and Mathematical Sciences, Head of the Electron Beam Ion Source Sector, Scientific and Experimental Department of Injection and the Nuclotron Ring, Veksler and Baldin Laboratory of High Energy Physics (VBLHEP), Joint Institute for Nuclear Research (JINR) (e-mail: edonets@jinr.ru)

Veneva, Milena P.—Student Trainee at the RIKEN Center for Computational Science (e-mail: milena.p.veneva@gmail.com, ORCID: 0000-0002-6421-4716)

RUSSIAN METADATA

УДК 519.63, 536.2

PACS 44.10.+i, 07.05.Tr, 02.70.Bf

DOI: 10.22363/2658-4670-2026-34-3-408-429

EDN: QXTTKT

Моделирование теплопроводности в тулиевой фольге крыловидной формы для источника ионовА. С. Айриян^{1,2}, Э. А. Айрян¹, Е. Е. Донец¹, М. П. Венева³¹ Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, г. Дубна, 141980, Российская Федерация² Национальная научная лаборатория им. А. Алиханяна, ул. Братьев Алиханян, д. 2, г. Ереван, 0036, Армения³ Центр вычислительных наук RIKEN, Минатодзима-минами-мати 7-1-26, район Тьюо, г. Кобе, префектура Хёго, 650-0047, Япония

Аннотация. *Актуальность.* Металлическая фольга, нагреваемая электрическим током, широко применяется в качестве испарительных элементов источников ионов. Поскольку скорость испарения сильно зависит от температуры, точное и эффективное определение распределения температуры необходимо для выбора геометрии фольги и рабочих условий. *Цель.* Разработать двумерную электротепловую модель тулиевой фольги крыловидной формы и оценить численные методы для многократного проведения расчётов. *Методы.* Благодаря симметрии в расчётную область включается только одна четверть фольги. Траектории электрического тока определяются посредством решения задачи Лапласа при постоянной электропроводности. Полученное поле нормируется на заданный полный ток и используется для вычисления пространственно-неоднородного источника джоулева тепла. Нестационарное уравнение теплопроводности учитывает температурные зависимости плотности, теплоёмкости, теплопроводности и удельного электрического сопротивления, а также радиационные и испарительные тепловые потери с обеих поверхностей фольги. Уравнение дискретизируется на декартовой сетке с маской расчётной области и интегрируется методом переменных направлений; на каждом полушаге применяется метод простых итераций. *Результаты.* При токе 2,5 А и температуре контактов 78 К максимальная температура в стационарном состоянии составляет 829,803 К. Все исследованные методы решения трёхдиагональных систем дают одно и то же температурное поле после достижения сходимости. Самый быстрый метод разбиения в сочетании с графом задач CUDA выполняет расчёт за 36,3 с, обеспечивая ускорение в 24,1 раза по сравнению с последовательной реализацией на центральном процессоре. *Заключение.* Разработанная модель учитывает неоднородный электрический нагрев и нелинейные тепловые потери в фольге. Реализация на графическом процессоре обладает достаточной производительностью для проведения многократных расчётов при оптимизации геометрии фольги и рабочего тока.

Ключевые слова: нелинейная теплопроводность, джоулев нагрев, метод переменных направлений, метод прогонки, метод разбиения, CUDA