A Numerical Implementation of MHD Bi-Axial Stretching with Cattaneo–Christov Heat Flux

This appendix describes the staged numerical solver implemented in the provided Jupyter Python script, reproducing the work of Usama et al. [1].

A.1 Coupling Structure

The system exhibits strict feed-forward one-way coupling:

$$(f,g) \xrightarrow{M,\alpha} \theta \xrightarrow{f,g,\mathrm{Pr},\beta_1,S_0} h \xrightarrow{f,g,\theta,M,\mathrm{Gr}} \tag{1}$$

Three sequential solution stages eliminate global stiffness.

1. Stage 1: Solve momentum subsystem $(f(\eta), g(\eta))$.
2. Stage 2: Solve temperature subsystem $\theta(\eta)$ (given interpolants of $f+g$ and $f'+g'$). Apply homotopy continuation for non-zero β_1, S_0.
3. Stage 3: Solve free-stream buoyancy subsystem $h(\eta)$ (given interpolants of f, $f+g$, and θ).

A.2 Key ODE System (first-order reduced form for `scipy.integrate.solve_bvp`)

The momentum subsystem transforms the sixth-order boundary value problem into a system of six first-order ODEs with state vector $\boldsymbol{y} = [f,\, f',\, f'',\, g,\, g',\, g'']$.

$$\begin{cases} f' = y_1, \quad f'' = y_2, \quad f''' = -(f+g)f'' + (f')^2 + Mf', \\ g' = y_4, \quad g'' = y_5, \quad g''' = -(f+g)g'' + (g')^2 + Mg'. \end{cases} \tag{2}$$

Boundary conditions for momentum:

$$f(0) = 0,\ f'(0) = 1,\ f'(\infty) = 0, \quad g(0) = 0,\ g'(0) = \alpha,\ g'(\infty) = 0.$$

Temperature subsystem state vector $\boldsymbol{y} = [\theta,\, \theta']$. The second-order Cattaneo–Christov ODE:

$$\theta'' = -\frac{\mathrm{Pr}(f+g)\theta' - \beta_1\mathrm{Pr}(f'+g')\theta' + \beta_1\mathrm{Pr}S_0(f+g)\theta' + \mathrm{Pr}S_0\theta}{1 - \beta_1\mathrm{Pr}(f+g)^2}. \tag{3}$$

Boundary conditions: $\theta(0) = 1$, $\theta(\infty) = 0$. The denominator is regularized in-code for intermediate iterates:

$$\mathrm{denom} = \begin{cases} \Delta(\eta), & |\Delta(\eta)| \geq 10^{-6}, \\ \mathrm{sign}(\Delta(\eta) + 10^{-30}) \cdot 10^{-6}, & |\Delta(\eta)| < 10^{-6}, \end{cases} \quad \Delta(\eta) = 1 - \beta_1\mathrm{Pr}(f+g)^2. \tag{4}$$

This regularization only protects Newton iterations; physical simulations are restricted to $\Delta_{\min} > 0$. Free-stream buoyancy subsystem state vector $\boldsymbol{y} = [h,\, h']$:

$$h'' = -(f+g)h' + hf' + Mh + \mathrm{Gr}\,\theta, \tag{5}$$

Boundary conditions: $h(0) = 0$, $h(\infty) = 1$.

A.3 Workflow Flowchart

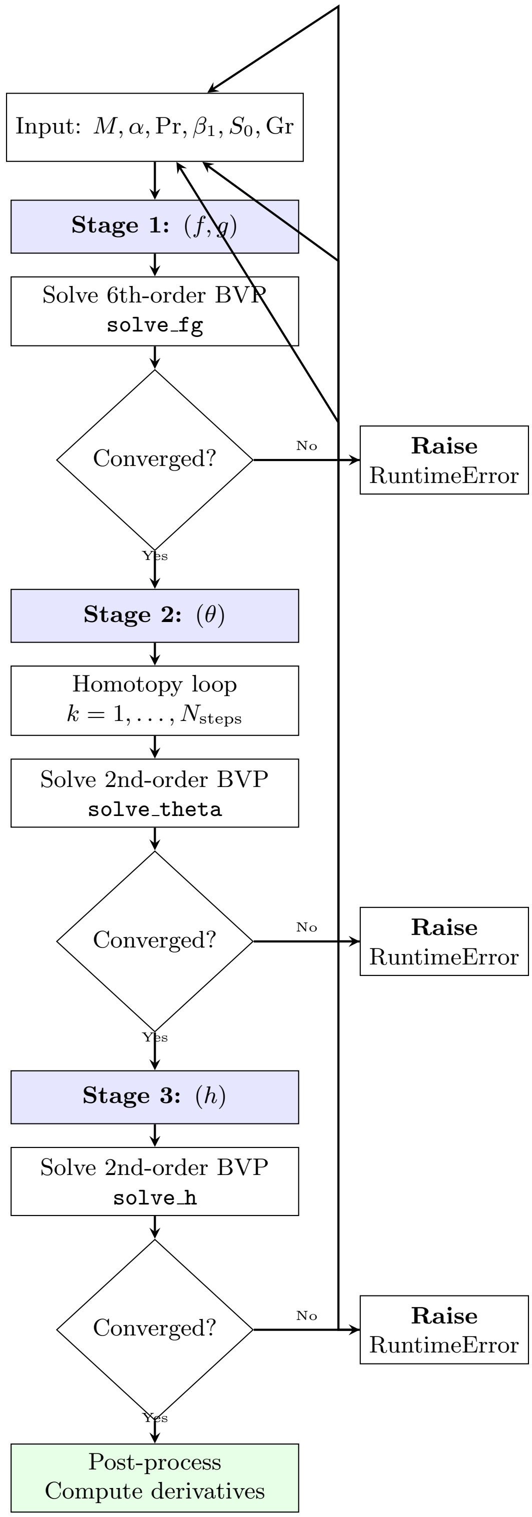

Figure 1: Workflow flowchart for the Python Jupyter implementation. Homotopy is internal to `solve_theta`. Runtime exceptions are raised for failed BVP convergence.

A.4 Pseudocode

Algorithm 1: Staged solver from Jupyter Python script

Input: $M, \alpha, \mathrm{Pr}, \beta_1, S_0, \mathrm{Gr}$; domain lengths $L_{fg} = 8.0,\ L_\theta \in [4, 6],\ L_h = 8.0$; initial nodes $n_0 = 200$; tolerance tol $= 10^{-9}$; homotopy steps $N_{\text{steps}} = 10$; regularization floor $\varepsilon = 10^{-6}$

Output: Solutions f, g, θ, h; surface derivatives $f''(0), g''(0), \theta'(0), h'(0)$

Stage 1: Momentum Subsystem (f, g);

Initialize $\eta \leftarrow \mathrm{linspace}(0, L_{fg}, n_0)$;

Construct initial guess: $f_0(\eta) = 1 - e^{-\eta},\ g_0(\eta) = \alpha(1 - e^{-\eta})$;

Assemble state vector $y_0 = [f_0,\ f_0',\ f_0'',\ g_0,\ g_0',\ g_0'']^T$;

Call `solve_bvp` for 6th-order system:

$$\begin{cases} f''' + (f+g)f'' - (f')^2 - Mf' = 0, \\ g''' + (f+g)g'' - (g')^2 - Mg' = 0, \end{cases}$$

with BCs: $f(0) = 0,\ f'(0) = 1,\ f'(\infty) = 0,\ g(0) = 0,\ g'(0) = \alpha,\ g'(\infty) = 0$;

if *not sol.success* **then**

  Raise RuntimeError("Momentum solve failed");

end

Construct interpolants: $F(\eta) = f(\eta) + g(\eta),\ F'(\eta) = f'(\eta) + g'(\eta)$;

Stage 2: Temperature Subsystem (θ);

$\eta_\theta \leftarrow \mathrm{linspace}(0, L_\theta, n_0)$;

$y_\theta \leftarrow [e^{-\eta_\theta},\ -e^{-\eta_\theta}]^T$;

if $\beta_1 = 0$ *and* $S_0 = 0$ **then**

  $N_{\text{steps}} \leftarrow 1$;

end

for $k = 1$ **to** N_{steps} **do**

  $t \leftarrow k/N_{\text{steps}}$;

  $\beta_1^{(k)} \leftarrow \beta_1 \cdot t, \quad S_0^{(k)} \leftarrow S_0 \cdot t$;

  Solve 2nd-order BVP:

$$\theta''\Delta + \mathrm{Pr}F\theta' - \beta_1 \mathrm{Pr}F'\theta' + \beta_1 \mathrm{Pr}S_0 F\theta' + \mathrm{Pr}S_0\theta = 0,$$

  where $\Delta = 1 - \beta_1 \mathrm{Pr}F^2$, with regularization $\widetilde{\Delta}$;

  BCs: $\theta(0) = 1,\ \theta(\infty) = 0$;

  if *not sol.success* **then**

    Raise RuntimeError("Temperature solve failed at step k");

  end

  Warm start: update η_θ, y_θ from current solution;

end

Construct temperature interpolant $\Theta(\eta)$;

Stage 3: Free-Stream Subsystem (h);

$\eta_h \leftarrow \mathrm{linspace}(0, L_h, n_0)$;

$y_h \leftarrow [1 - e^{-\eta_h},\ e^{-\eta_h}]^T$;

Solve 2nd-order BVP:

$$h'' + Fh' - hf' - Mh - \mathrm{Gr}\,\Theta = 0,$$

BCs: $h(0) = 0,\ h(\infty) = 1$;

if *not sol.success* **then**

  Raise RuntimeError("Free-stream solve failed");

end

Compute wall derivatives: $f''(0),\ g''(0),\ \theta'(0),\ h'(0)$ from solution objects;

return solution objects and surface quantities;

A.5 Chebyshev Pseudospectral Verification

The spectral solver is used only for momentum subsystem (f, g) as independent verification, matching the original paper.

Algorithm 2: Chebyshev pseudospectral solver for momentum subsystem (Python implementation)

Input: M, α, number of nodes N, physical domain length L
Output: Nodal solutions f_j, g_j, wall second derivative $f''(0)$
Generate Chebyshev–Gauss–Lobatto nodes: $x_j = \cos(\pi j/N), \quad j = 0, 1, \ldots, N$;
Build differentiation matrix D_x via `cheb()` function;
Map reference domain $[-1, 1]$ to physical domain $[0, L]$: $\eta_j = L(1 - x_j)/2$;
Scale derivative matrix: $D = -(2/L) \cdot D_x$;
Compute higher derivatives: $D_2 = D \cdot D, \quad D_3 = D \cdot D \cdot D$;
Assemble unknown vector: $\boldsymbol{u} = [f_0, \ldots, f_N,\ g_0, \ldots, g_N]^T$;
Define residual function $R(\boldsymbol{u})$: Extract f, g from $\boldsymbol{u}$;
Compute derivatives Df, D_2f, D_3f and Dg, D_2g, D_3g;
Evaluate ODE residuals at interior collocation points:

$$R_f = D_3f + (f + g)D_2f - (Df)^2 - MDf,$$

$$R_g = D_3g + (f + g)D_2g - (Dg)^2 - MDg$$

Enforce boundary conditions by replacing first and last rows;
Initial guess $\boldsymbol{u}_0$ from adaptive collocation solution `solve_fg`;
Solve nonlinear system using `scipy.optimize.root` with `method="lm"` (Levenberg–Marquardt), xtol $= 10^{-13}$;
Extract f_j, g_j from solution;
Compute wall derivative: $f''(0) = (D_2f)[0]$;
return nodal arrays and wall derivative;

A.6 Numerical Parameters

The following numerical parameters are hard-coded in the Python script:

- Adaptive collocation solver: `scipy.integrate.solve_bvp`, relative tolerance tol $= 10^{-9}$, `max_nodes=100000`.
- Domain truncation lengths: $L_{fg} = 8.0$, $L_h = 8.0$, $L_\theta \in \{4.0, 5.0\}$.
- Initial mesh points for BVP: $n_0 = 200$.
- Homotopy steps: $N_{\text{steps}} = 10$ (auto set to 1 for $\beta_1 = S_0 = 0$).
- Regularization floor for temperature denominator: $\varepsilon = 10^{-6}$; offset 10^{-30} for sign stability.
- Spectral solver: Levenberg–Marquardt, xtol $= 10^{-13}$, maxiter=20000; convergence study node set $N \in \{4, 6, 8, 10, 12, 16, 20, 24, 28, 32, 40\}$.
- Plot style: MATLAB default color palette, serif font, inward-pointing ticks, light grid lines.

A.7 Important Code-Specific Observations

1. **Interpolation handling for θ in `solve_h`**: The temperature solution is defined only on shorter domain $[0, L_\theta]$. In the Python code, values for $\eta > L_\theta$ are clipped and set to zero. This is a practical numerical approximation consistent with fast decay of $\theta(\eta)$.
2. **Regularization is only for intermediate iterates**: The floor on denom stabilizes Newton updates during homotopy steps. For physically admissible parameter sets $\Delta_{\min} > 0$, the regularization floor is never activated. If $\Delta(\eta)$ becomes negative inside domain, solver may produce unphysical profiles without warning; the current code does not implement an automatic $\Delta_{\min}$ reject check (this is a suggested improvement).

3. **Homotopy only on temperature solve**: Momentum subsystem (f, g) is solved in one shot without continuation. Only β_1 and S_0 are ramped incrementally inside `solve_theta`.

4. **Spectral solver initial guess**: The pseudospectral routine uses the adaptive-collocation solution as initial guess. Without this high-quality starting guess, Levenberg–Marquardt may fail to converge.

5. **Safe β_1 clamping in plotting functions**: In figure-generation functions for high-Prandtl numbers, β_1 is heuristically clamped to avoid singular-point regimes. This is purely for plotting robustness and is not part of the original mathematical formulation.

Validation Strategy

The following validation strategy is implemented in the script:

- Benchmark limit: $\beta_1 = 0, S_0 = 0, M = 0, \mathrm{Gr} = 0$; compare wall derivatives $f''(0), g''(0), \theta'(0)$ against Crane-Wang reference values.

- Spectral convergence test: compute error $|f''_{\text{spectral}}(0) - f''_{\text{collocation}}(0)|$ over increasing Chebyshev node count N; verify exponential decay of error.

- Visual reproduction: six pairs of sub-figures replicate the original paper's figures 2–7.

Further insight into the multi-dimensional parametric landscape of the MHD bi-axial stretching flow with Cattaneo–Christov heat flux is provided by the composite heatmap analysis in Figure 2. This multi-panel visualization consolidates admissibility bounds, surface thermal gradients, wall-shear behaviour, and critical-threshold characteristics across the extended parameter space.

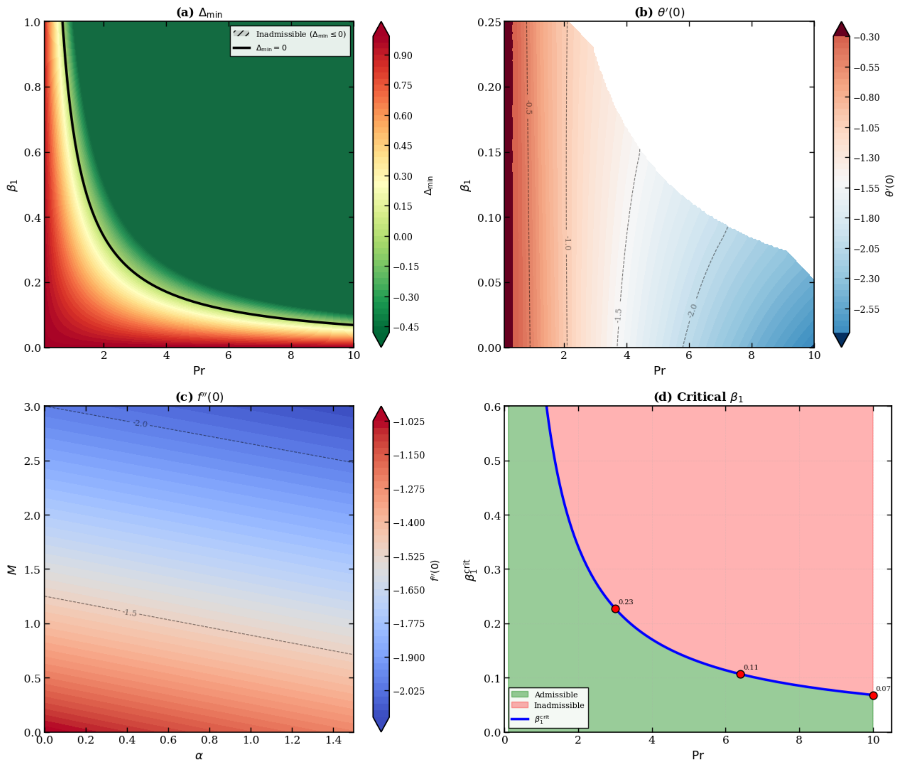

Figure 2: Composite heatmap analysis for the MHD bi-axial stretching flow with Cattaneo–Christov heat flux. (a) Admissibility margin $\Delta_{\min}$ in (Pr, β_1) space; the solid black contour marks $\Delta_{\min} = 0$, separating admissible and inadmissible domains. (b) Surface thermal gradient $\theta'(0)$ in (Pr, β_1) space; dashed contours denote constant values of the wall temperature gradient. (c) Wall-shear derivative $f''(0)$ in (α, M) parameter space as obtained from momentum-only BVP solutions. (d) Numerically evaluated critical boundary $\beta_1^{\text{crit}}(\mathrm{Pr})$, with green shading for admissible parameters $\beta_1 < \beta_1^{\text{crit}}(\mathrm{Pr})$ and red shading for the inadmissible singular regime; discrete computed critical points are overlaid.

Figure 2(a) presents the admissibility margin $\Delta_{\min} = 1 - \beta_1 \mathrm{Pr} \max_\eta (f+g)^2$ evaluated over the (Pr, β_1) plane for fixed $M = 1.0$, $\alpha = 1.0$. Positive values of $\Delta_{\min}$ correspond to well-posed non-singular temperature-equation solutions, whereas $\Delta_{\min} \le 0$ identifies parameter combinations where the denominator of the Cattaneo–Christov similarity ODE vanishes within the boundary-layer domain. The $\Delta_{\min} = 0$ boundary exhibits a hyperbolic-like decay with increasing Pr: low-Pr fluids tolerate substantially larger thermal-relaxation parameters β_1, while high-Pr conditions severely constrain the permissible magnitude of β_1. This demonstrates that non-Fourier thermal-relaxation effects can only be studied within a tightly bounded parameter subspace for high-Prandtl-number liquids.

The surface thermal gradient $\theta'(0)$ is shown in Figure 2(b), computed via homotopy-continued BVP solves within the admissible parameter domain only. The magnitude of $\theta'(0)$ increases monotonically with both Pr and β_1. At low Pr, the wall gradient is weakly sensitive to variations in β_1, indicating diffusive transport dominates over finite-relaxation effects. For $\mathrm{Pr} \gtrsim 2$, β_1 exerts a progressively stronger influence on the surface heat flux, reflecting the growing interplay between boundary-layer thinning and finite thermal wave speed. All contour results are masked in the inadmissible region, where no physically consistent similarity-solution exists.

Figure 2(c) isolates the momentum-field response by mapping the wall-shear derivative $f''(0)$ across stretching ratio α and magnetic parameter M. Each grid point corresponds to an independent solve of the sixth-order momentum boundary-value problem, with no thermal coupling. Increasing magnetic parameter M suppresses near-wall velocity gradients and reduces wall shear, consistent with Lorentz-force flow retardation. Larger transverse stretching ratios α elevate the magnitude of $f''(0)$, enhancing momentum transport within the bi-axially stretched boundary-layer. This panel quantifies the relative importance of magnetic damping versus transverse stretching for surface shear stress, independent of thermal-relaxation parameters.

The critical threshold curve $\beta_1^{\mathrm{crit}}(\mathrm{Pr})$ is displayed in Figure 2(d), directly extracted from the condition $\Delta_{\min} = 0$. The numerically sampled critical points confirm that the maximum allowable thermal-relaxation parameter decays rapidly as Pr grows. For $\mathrm{Pr} = 2$, $\beta_1^{\mathrm{crit}} \approx 0.23$; for $\mathrm{Pr} = 6.4$, $\beta_1^{\mathrm{crit}} \approx 0.11$; and for $\mathrm{Pr} = 10$, $\beta_1^{\mathrm{crit}} \approx 0.07$. These values provide practical upper bounds for β_1 when designing parametric numerical studies. Operating close to this critical boundary amplifies sensitivity to numerical noise and homotopy-solver residuals, so practical computations should employ β_1 values distinctly below $\beta_1^{\mathrm{crit}}(\mathrm{Pr})$.

Taken collectively, the composite parameter-space maps underscore that admissibility imposes hard constraints upon the Cattaneo–Christov similarity formulation. The inadmissible region signals the onset of interior-domain singularities in the transformed energy equation. Such singular behaviour suggests that for parameters beyond the $\beta_1^{\mathrm{crit}}(\mathrm{Pr})$ boundary, the present similarity reduction breaks down, and alternative formulations accounting for additional physics would be required. These contour results deliver concrete operational guidelines for parameter selection: for any specified Prandtl number, β_1 must remain sufficiently smaller than $\beta_1^{\mathrm{crit}}(\mathrm{Pr})$ to guarantee both mathematical well-posedness and numerical robustness of the boundary-layer solutions.

Nomenclature

Roman Symbols

Symbol	Description
a	stretching rate (s^{-1})
B_0	magnetic field strength (T)
c_p	specific heat at constant pressure (J kg^{-1} K^{-1})
f	axial stretching stream function
g	transverse stretching stream function
$\mathbf{g}$	gravitational acceleration vector (m s^{-2})
Gr	Grashof number
h	free-stream function
k	thermal conductivity (W m^{-1} K^{-1})
L	numerical truncation domain
M	magnetic parameter
N	number of Chebyshev grid points
p	pressure (Pa)
Pr	Prandtl number
$\mathbf{q}$	heat flux vector (W m^{-2})
Q_0	volumetric heat source (W m^{-3})
S_0	heat source parameter
T	temperature (K)
T_w	wall temperature (K)
T_∞	ambient temperature (K)
u, v, w	velocity components in x, y, z directions (m s^{-1})
U	free-stream velocity (m s^{-1})
x, y, z	Cartesian coordinates (m)

Table 1: Roman symbols used in the mathematical formulation.

Greek Symbols

Symbol	Description
α	transverse stretching ratio
α_{th}	thermal diffusivity (m^2 s^{-1})
β_1	Cattaneo–Christov relaxation parameter
β_T	thermal expansion coefficient (K^{-1})
Δ	leading coefficient / admissibility function
$\Delta_{\min}$	admissibility margin
ε	regularization floor parameter (10^{-6})
η	similarity variable
θ	dimensionless temperature
λ_t	thermal relaxation time (s)
μ	dynamic viscosity (Pa s)
ν	kinematic viscosity (m^2 s^{-1})
ρ	fluid density (kg m^{-3})
σ	electrical conductivity (S m^{-1})

Table 2: Greek symbols used in the mathematical formulation.

Mathematical Operators

Operator	Description
$(')$	differentiation with respect to η
∇	gradient operator
diag	diagonal matrix operator
$\mathcal{A}$	admissible parameter region
$\widetilde{\Delta}$	regularized admissibility function

Table 3: Mathematical operators used in the formulation.

Dimensionless Parameters

Parameter	Definition
M	$\dfrac{\sigma B_0^2}{\rho a}$
Pr	$\dfrac{\nu}{\alpha_{th}}$
Gr	$\dfrac{g\beta_T(T_w - T_\infty)}{Ua}$
S_0	$\dfrac{Q_0}{\rho c_p a}$
β_1	$\lambda_t a$

Table 4: Dimensionless parameters and their definitions.

References

[1] Syed M. Usama, Umair Umer, M. Ateeq Tahir, Zhigang Sun. Well-Posedness Diagnostics and Staged Numerical Solution of MHD Buoyant Flow over a Bi-Axially Stretching Surface with Cattaneo–Christov Thermal Relaxation.