

Square-body scalar transport

A functional spreadsheet and browser model:
accuracy, assumptions, and worked examples

Alex Blythe 24 October 2026 Version 1.0

Scope. This model solves a scalar equation in a prescribed horizontal velocity field. It does not solve fluid velocity, pressure, drag, or vortex shedding. Close agreement between implementations establishes agreement on the stated discrete equations, not physical accuracy of flow around a solid.

1. Problem and reconstruction

The model reconstructs the transport calculation visible in user-supplied screenshots from [@cfdbyhand](#). The original workbook and narration were unavailable. This is an independently authored implementation, with attribution for teaching inspiration rather than a claim of affiliation or exact reproduction.

The transported scalar ϕ is dimensionless. For constant prescribed $u \geq 0$, density ρ , and transport coefficient Γ , the modeled equation is

$$\rho u \frac{\partial \phi}{\partial x} = \Gamma \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right). \quad (1)$$

The rectangle is $0 \leq x \leq L$, $0 \leq y \leq H$. The square is a region where grid-node scalar values are held at ϕ_s . It is not an impermeable, no-slip obstacle in a solved velocity field. The prescribed velocity remains unchanged.

Quantity	Baseline value	Provenance
u, ρ, Γ	1 m/s, 1 kg/m ³ , 0.1 kg/(m s)	Visible screenshot inputs
$\Delta x, \Delta y$	0.05 m each	Visible screenshot spacing
L, H	3 m, 2 m	Reconstruction assumptions
Square corner and side	(0.8, 0.75) m; 0.5 m	Reconstruction assumptions
ϕ_{in}, ϕ_s	1, 0	Reconstructed boundary setup
Grid	61 $\times$ 41 nodes	Derived from domain and spacing

The left boundary holds $\phi = \phi_{\text{in}}$. The right boundary copies its left neighbor, approximating $\partial \phi / \partial x = 0$. Top and bottom copy the adjacent interior row, approximating $\partial \phi / \partial y = 0$. Boundary copying is first-order. Inlet conditions take precedence at the left corners. Square nodes, including their geometric boundary, are fixed. Geometry changes snap to included nodes with a 10^{-9} m comparison allowance.

The rectangle's transverse boundaries represent zero scalar gradient. They are not evidence that a hydrodynamic symmetry condition has been satisfied. Background discretization references are provided by NIST FiPy [1, 2].

2. Discretization and accuracy

For an interior node and positive u , the model uses upwind advection and centered diffusion. In conductance notation,

$$D_x = \Gamma \frac{\Delta y}{\Delta x}, \quad D_y = \Gamma \frac{\Delta x}{\Delta y}, \quad F = \rho u \Delta y, \quad (2)$$

$$a_W = D_x + F, \quad a_E = D_x, \quad a_N = a_S = D_y, \quad (3)$$

$$\phi_P = \frac{a_W \phi_W + a_E \phi_E + a_N \phi_N + a_S \phi_S}{a_P}, \quad a_P = a_W + a_E + a_N + a_S. \quad (4)$$

At baseline, $D_x = D_y = 0.1$, $F = 0.05$, and $a_P = 0.45$. The denominator is included. The cell Péclet number is $F/D_x = 0.5$.

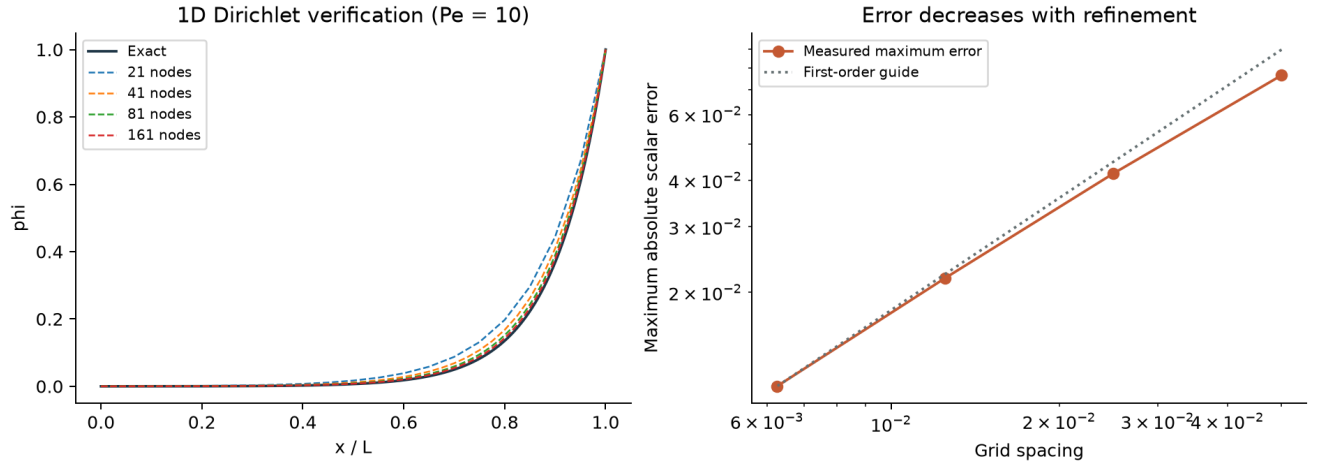
Upwind advection is first-order. A Taylor expansion of the backward derivative introduces leading streamwise numerical diffusion

$$\Gamma_{\text{num}} \simeq \frac{\rho u \Delta x}{2} = 0.025 \text{ kg}/(\text{m s}). \quad (5)$$

This is 25% of the specified baseline Γ . **It is a truncation-error coefficient ratio, not a 25% solution-error claim.** Its derivation assumes a smooth solution away from boundaries and corners. NIST discusses the upwind scheme's excess numerical diffusion [2].

Analytical check. A separate one-dimensional problem uses $L = 1$, $\rho = u = 1$, $\Gamma = 0.1$, $\phi(0) = 0$, and $\phi(1) = 1$. These Dirichlet conditions differ from the square example. Its exact solution is

$$\phi(x) = \frac{\exp(10x) - 1}{\exp(10) - 1}. \quad (6)$$

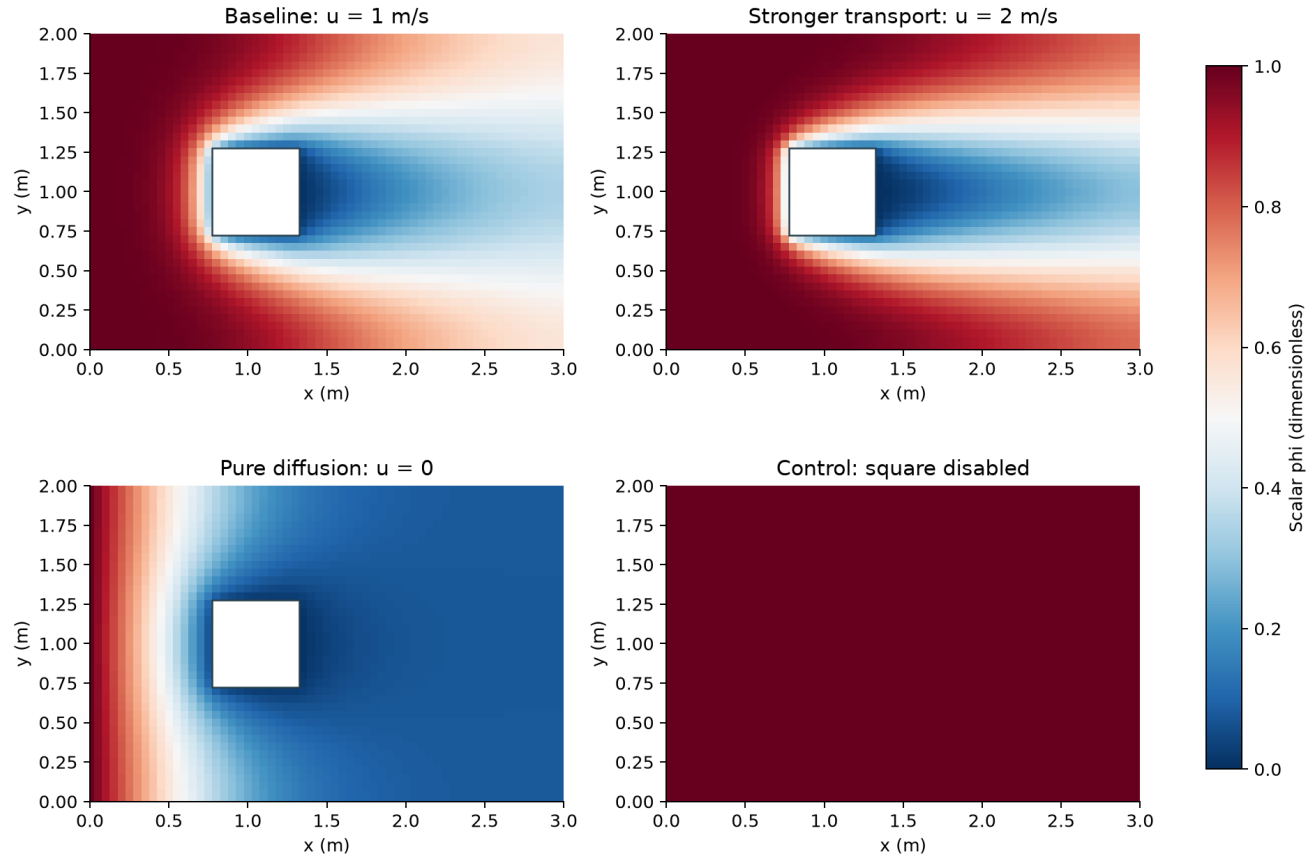


Nodes	h	$\ e\ _\infty$	Observed order
21	0.05000	0.07643	—
41	0.02500	0.04167	0.875
81	0.01250	0.02184	0.932
161	0.00625	0.01120	0.964

The observed order approaches one. Error is absolute scalar error; dividing by near-zero exact values would give misleading relative errors. This test verifies the scheme, not an error bound for the square geometry.

3. Worked examples

All four examples use the baseline domain, density, transport coefficient, and grid. The square is shown as white because its scalar is prescribed, not because velocity has been solved around it. A fixed color scale permits comparisons.



Example	Change	u (m/s)	$\phi(3, 1)$
Baseline	None	1	0.338126
Stronger transport	Double velocity	2	0.290217
Pure diffusion	Remove advection	0	0.077516
Uniform control	Disable square clamp	1	1.000000

Interpretation. Increasing transport speed changes the extent and recovery of the scalar deficit. Removing advection leaves a diffusion problem driven by the inlet and the square clamp. Disabling the square admits the exact uniform solution $\phi = 1$, which checks the boundary setup and constant preservation.

These pictures do not show streamlines, pressure, momentum loss, or vorticity. The scalar deficit downstream is not an aerodynamic wake prediction. No physical interpretation such as concentration or normalized temperature is assigned without the additional constitutive and boundary assumptions that interpretation requires.

4. Executed numerical evidence

Independent assembly and solution. A Python implementation assembles the linear equations as a sparse matrix and uses SciPy `spsolve`. It does not call the JavaScript solver. GS below denotes the separate Gauss–Seidel reference. The normalized residual is

$$r_\infty = \max_P \left| \frac{(A\phi - b)_P}{A_{PP}} \right|, \quad (7)$$

including boundary rows. This has scalar units and is equivalent to the maximum normalized stencil defect. It is not a continuum error estimate.

Case	$\ \phi_{LU} - \phi_{GS}\ _\infty$	LU residual
baseline	3.88e-11	4.93e-16
faster	1.38e-11	7.77e-16
diffusion	1.17e-09	4.51e-16
uniform	2.00e-14	6.78e-16

Four direct solutions also remain within the specified scalar bounds, allowing 10^{-10} roundoff. The largest LU-to-GS difference is 1.17×10^{-9} , in pure diffusion. Two implementations can share a modeling error; the analytical test addresses an additional, different question.

Native workbook checks. Microsoft Excel 16.0, build 20326 recalculated the workbook, changed velocity, disabled the square, selected pure diffusion, restored defaults, and saved/reopened the result. All 2,501 field formulas and the iteration settings were retained.

Case	Excel vs. GS maximum difference	Excel residual
baseline	1.89e-12	9.33e-13
faster	1.33e-09	9.23e-11
uniform	4.10e-09	9.71e-11
diffusion	1.16e-07	9.99e-11

Excel uses at most 10,000 iterations per calculation with maximum change 10^{-10} . A small change or residual is not an accuracy certificate. Pure diffusion may take substantially longer than the baseline. The browser worker’s message path and invalid-input handling were exercised in a Node worker harness; this is not a claim of cross-browser UI validation.

Square-case refinement. Three aligned grids preserve the physical square coordinates and include its boundary nodes. Common-node changes compare adjacent grids at shared fluid nodes.

Grid	h_x (m)	$\phi(2,1)$	$\phi(3,1)$	Common-node change
61×41	0.0500	0.194275	0.338126	–
121×81	0.0250	0.199701	0.348439	0.04701
241×161	0.0125	0.202145	0.353281	0.02721

The outlet-center value changes by about 0.01516 between the first and finest grids. The maximum common-node differences remain 0.04701 and 0.02721. **The baseline is mesh-dependent.** The finest result is a comparison, not an exact solution. Corner behavior and first-order boundary treatment affect the trend. No formal error bound, grid-convergence index, or grid-independence claim is made.

5. Use, reproduction, and limits

Workbook. Open `Square_Body_Transport.xlsx` in desktop Excel and edit amber inputs on Model. Field contains the cell equations and residual. Allow iterative calculation and press F9 if necessary. The workbook settings are 10,000 iterations and maximum change 10^{-10} . Other open workbooks can affect Excel calculation settings. No macros, external workbook links, or private service connections are required.

Browser. The worker solves the same nodal model locally. Geometry stays on the fixed grid. Input changes require a new calculation; any retained result is explicitly identified as the previous completed solve. Stop cancels the worker. A completed field can be downloaded as CSV. The UI reports iteration count, residual, cell Péclet number, and outlet-center scalar. A stalled iteration limit is reported as unconverged, not accepted.

Reproduce. The public package contains the browser solver and worker, independent verification script, versioned inputs, workbook reference fields, Excel verification summary, CSVs, accuracy measurements, and a SHA-256 manifest. In a new environment with Python, install `numpy scipy matplotlib`, then run:

```
python verify_accuracy.py -output .
```

Run this from an extracted reproduction package containing the supplied reference JSON files. The script rebuilds the sparse, analytical, and refinement checks and figures. It does not rerun Excel. The retained Excel measurements describe the recorded desktop run. Compile `main.tex` inside the LaTeX source directory with a normal LaTeX engine or Tectonic.

Remaining limits. The model is linear, steady, two-dimensional, and limited to nonnegative prescribed horizontal velocity with positive density and diffusion. The square prescribes scalar; it does not enforce no-penetration or no-slip in a solved velocity field. There is no turbulence, transition, compressibility, momentum coupling, pressure correction, or unsteady shedding model. Domain adequacy and experimental agreement are unestablished. The screenshot-derived values do not define a validated material or flow regime. The results are educational numerical evidence, not an aerodynamic design release.

Next physical extension. Introduce continuity and both momentum equations, a pressure projection or correction, no-slip/no-penetration boundaries, and appropriate unsteady treatment. Verify against independent reference quantities before reporting drag, separation, or wake dynamics. Those additions would change the model's governing problem rather than merely improve its color plot.

References

- [1] NIST FiPy, *Finite Volume Method*. <https://pages.nist.gov/fipy/en/stable/numerical/discret.html>. Accessed 24 October 2026. Background reference; this implementation uses a nodal stencil, not FiPy itself.
- [2] NIST FiPy, *Numerical Schemes*. <https://pages.nist.gov/fipy/en/3.4.5/numerical/scheme.html>. Accessed 24 October 2026.
- [3] CFDbyHand, Instagram profile @cfdbyhand. <https://www.instagram.com/cfdbyhand/>. Teaching inspiration reviewed from user-supplied screenshots; original workbook not obtained.
- [4] Alex Blythe, *Square-body scalar transport, version 1.0: executable evidence and downloads*. <https://alex-blythe.com/software/aero/square-body-transport/>. 24 October 2026. See `accuracy-results.json`, `excel-verification.json`, and `browser-solver-checks.json`.