

Convervative averaging

Chatgpt

December 9, 2025

It is the "most common and clean case" in physics and numerical modeling (finite volume, EM, CFD, etc.): two "nonuniform grids" that cover the "same physical domain" (same total length), and you want to transfer "cell averages" conservatively from one grid to another.

Let's go step by step carefully.

1 Problem setup

You have a 1D physical domain $([x_0, x_N])$.

- "Source grid" (old grid): edges $(x_{1/2}, x_{3/2}, \dots, x_{N_s+1/2})$ cell averages $(f_i), (i = 1, \dots, N_s)$
- "Target grid" (new grid): edges $(X_{1/2}, X_{3/2}, \dots, X_{N_t+1/2})$ unknown averages $(F_j), (j = 1, \dots, N_t)$

Both grids cover "exactly the same domain": $[x_0, x_{N_s}] = [X_0, X_{N_t}] = [x_0, x_N]$.

2 Conservative averaging principle

We want to ensure "total integral is preserved:"

$$\sum_i f_i, \Delta x_i = \sum_j F_j, \Delta X_j \quad (1)$$

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$, $\Delta X_j = X_{j+1/2} - X_{j-1/2}$. The physical requirement is that the "integrated quantity" in each target cell equals the "overlap-weighted sum" of source cell integrals. For each target cell (j) :

$$F_j = \frac{1}{\Delta X_j} \sum_i f_i, L_{ij} \quad (2)$$

where (L_{ij}) is the "overlap length" between source cell (i) and target cell (j) :

$$L_{ij} = \max(0, \min(x_{i+1/2}, X_{j+1/2}) - \max(x_{i-1/2}, X_{j-1/2})) \quad (3)$$

This simple expression guarantees "strict conservation" over the entire domain.

3 Implementation (robust + exact for same-length grids)

Here's a clean, working Python version (but same logic applies in C or Fortran):

```
import numpy as np

def conservative_average_same_length(xe, f, Xe):
    """
    Conservative remap (finite-volume average) from source grid xe->Xe
    when both grids cover the same domain.

    Parameters
    -----
    xe : array of shape (Ns+1,) - source cell edges
```

```

f : array of shape (Ns,) - source cell averages
Xe : array of shape (Nt+1,) - target cell edges

Returns
-----
F : array of shape (Nt,) - target cell averages (conservative)
"""
Ns = len(f)
Nt = len(Xe) - 1
F = np.zeros(Nt, dtype=float)

i = 0
for j in range(Nt):
    X1, X2 = Xe[j], Xe[j+1]
    dx_target = X2 - X1
    s = 0.0
    # accumulate source contributions to target cell j
    while i < Ns and xe[i+1] <= X1:
        i += 1 # skip source cells to the left
    k = i
    while k < Ns and xe[k] < X2:
        overlap = max(0.0, min(xe[k+1], X2) - max(xe[k], X1))
        if overlap > 0.0:
            s += f[k] * overlap
        k += 1
    F[j] = s / dx_target
return F

xe = np.array([0.0, 1.0, 2.5, 4.0]) # source edges (3 cells)
f = np.array([2.0, 3.0, 1.0]) # source cell averages
Xe = np.array([0.0, 1.5, 3.0, 4.0]) # target edges (3 cells)

F = conservative_average_same_length(xe, f, Xe)

# check conservation:
source_total = np.sum(f * np.diff(xe))
target_total = np.sum(F * np.diff(Xe))

print("Target averages:", F)
print("Source integral:", source_total)
print("Target integral:", target_total)

```

- This is $O(N_s + N_t)$ and strictly conservative.
- Works even when the grids are nonuniform and not aligned.

4 Summary

To perform "conservative averaging between two nonuniform grids that span the same physical domain":

$$F_j = \frac{1}{\Delta X_j} \sum_i f_i \cdot \max(0, \min(x_{i+1/2}, X_{j+1/2}) - \max(x_{i-1/2}, X_{j-1/2})) \quad (4)$$

This preserves the "integrated quantity" exactly, even when the grids are: nonuniform, not aligned, and have different cell counts.

5 Derive an averaging rule from a conservation law — with a concrete example

5.1 Example 1 — 1D layered conduction (why harmonic mean appears)

Consider two horizontal layers in series (current/heat/flux flows vertically through both). Layer 1: thickness t_1 , conductivity σ_1 . Layer 2: t_2, σ_2 . Apply a vertical potential difference ΔV across the combined thickness $T = t_1 + t_2$.

Physics / conservation used : the same vertical flux J (current density) flows through both layers (steady state, series). Ohm/Fourier law locally: $J = \sigma \frac{\Delta V_{layer}}{t_{layer}}$. The total voltage drop equals sum of drops:

$$\Delta V = \Delta V_1 + \Delta V_2 = \frac{J t_1}{\sigma_1} + \frac{J t_2}{\sigma_2}. \quad (5)$$

Define an effective conductivity σ_{eff} for the whole thickness so that

$$J = \sigma_{eff} \frac{\Delta V}{T}. \quad (6)$$

Substitute ΔV and cancel J :

$$\frac{1}{\sigma_{eff}} = \frac{t_1}{T} \frac{1}{\sigma_1} + \frac{t_2}{T} \frac{1}{\sigma_2}. \quad (7)$$

So

$$\sigma_{eff} = \frac{T}{\frac{t_1}{\sigma_1} + \frac{t_2}{\sigma_2}} \quad (8)$$

— the “weighted harmonic mean” (weights = thickness fractions).

5.1.1 Numerical check (step-by-step arithmetic)

Let $t_1 = 2$ m, $t_2 = 3$ m, $\sigma_1 = 0.10$ S/m, $\sigma_2 = 0.01$ S/m. Compute:

- $T = 2 + 3 = 5$.
- $t_1/\sigma_1 = 2/0.10 = 20$.
- $t_2/\sigma_2 = 3/0.01 = 300$.
- Sum = $20 + 300 = 320$.
- $\sigma_{eff} = 5/320 = 0.015625$ S/m.

If instead you average resistivity $\rho = 1/\sigma$ the series average is arithmetic:

$$\rho_{eff} = \frac{t_1 \rho_1 + t_2 \rho_2}{T}, \quad (9)$$

and $1/\rho_{eff} = \sigma_{eff}$ — same number (check: $\rho_1 = 10$, $\rho_2 = 100$; $\rho_{eff} = (2 \cdot 10 + 3 \cdot 100)/5 = 320/5 = 64 \Omega \cdot m$; $1/64 = 0.015625$ S/m).

Interpretation: harmonic mean arises because you conserve flux (same J) and add the drops.

5.2 Example 2 — parallel conduction (arithmetic mean)

If instead the layers are side-by-side and subject to the same potential difference (i.e., they are *in parallel* electrically or thermally), the total flux is the sum of the fluxes:

$$J_{tot} = J_1 + J_2 = \sigma_1 \frac{\Delta V}{t} + \sigma_2 \frac{\Delta V}{t} = (\sigma_1 + \sigma_2) \frac{\Delta V}{t}. \quad (10)$$

Defining σ_{eff} so that $J_{tot} = \sigma_{eff} \frac{\Delta V}{t}$ gives

$$\sigma_{eff} = \frac{1}{2}(\sigma_1 + \sigma_2) \quad (11)$$

(for two equal-width parallell pieces; in general a weighted arithmetic mean). So **parallel requires arithmetic mean, series requires harmonic mean**.

5.3 Example 3 — finite-volume face conductivity (useful for discretization on nonuniform grid)

In finite-volume / two-point flux approximation you demand “flux continuity across a cell face”. Consider two adjacent control volumes with centers at distances d_L and d_R from the face, with conductivities σ_L and σ_R . The face conductivity σ_f is chosen so that the flux computed from the left side equals the flux from the right and equals the flux through the face.

From linear interpolation of potential between centers, one obtains

$$\sigma_f = \frac{d_L + d_R}{\frac{d_L}{\sigma_L} + \frac{d_R}{\sigma_R}} \quad (12)$$

which is a “distance-weighted harmonic mean”. (If $d_L = d_R$ this reduces to the simple harmonic mean.)

5.3.1 Small numeric example

Let $d_L = d_R = 1$, $\sigma_L = 0.10$, $\sigma_R = 0.01$:

- numerator = $1 + 1 = 2$.
- denominator = $1/0.10 + 1/0.01 = 10 + 100 = 110$.
- $\sigma_f = 2/110 \approx 0.0181818$ S/m.

This is exactly what many FD/FV codes use for face transmissibility when coefficients are discontinuous.

5.4 A brief note about anisotropy / tensors

When conductivity (or diffusivity) is a tensor, the normal component across an interface averages like the scalar normal conductivity (harmonic for series of layers normal to flow), while tangential components (parallel to layers) average like the parallel case (arithmetic). For full tensors you must transform to the local coordinate system relative to the interface, average appropriate components, and transform back.