Smart boundary conditions for surface layers of turbulent convection

M. Rheinhardt

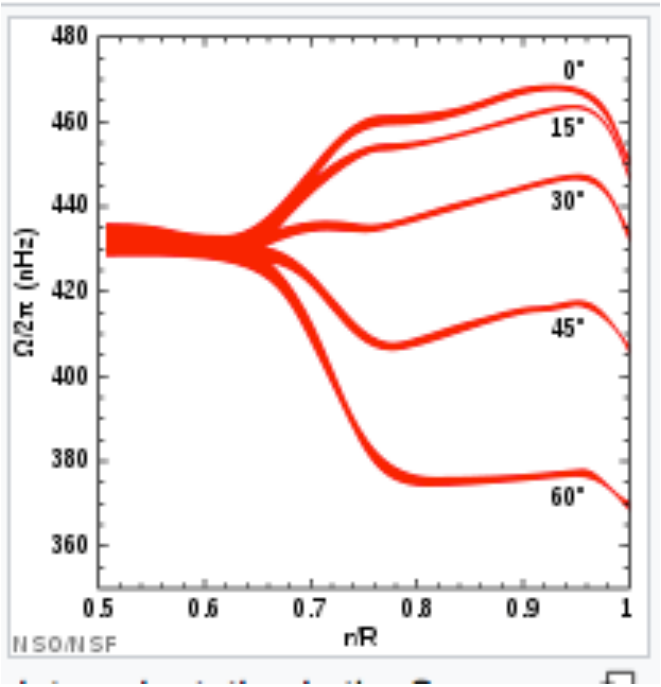
Aalto University

August 15, 2019

Motivation

Surface layers of turbulent convection:

- rapid transition from optically thick to optically thin
- $\longrightarrow$ strong density stratification
- $\longrightarrow$ surface shear
- $\longrightarrow$ explicit radiative heat transfer instead of diffusion approximation
- high spatial and temporal resolution necessary



Problem

simulation of convection/convective dynamo in entire convection zone with the resolution needed in the surface layer not feasible

$\longrightarrow$ employ

- strongly non-uniform grid
  $\longrightarrow$ time step set by smallest grid cells

or

- boundary condition which compactifies the surface layer; recipe:
  - locally simulate surface layer in (say) Cartesian box with physical BCs at upper boundary
  - figure out functional $\boldsymbol{F}(\{\boldsymbol{U}, \boldsymbol{B}, \rho, \boldsymbol{s}, \partial\boldsymbol{B}, \partial\boldsymbol{U}, \partial\rho, \partial\boldsymbol{s}\}|_{\partial V_{\rm low}})$ for which $\boldsymbol{F} = \boldsymbol{0}$ at lower boundary of surface layer $\partial V_{\rm low}$
  - $\boldsymbol{F}$ non-local, perhaps non-instantaneous
  - too ambitious!

A modest solution

employ time dependent Dirichlet BCs for all quantities
with the correct temporal correlation properties:

$$\int f_i(\boldsymbol{x}_k, t-\tau) f_j(\boldsymbol{x}_l, \tau) d\tau \quad \forall\, f_i, f_j \in \{\boldsymbol{U}, \boldsymbol{A}, \rho, s\} \quad \text{and} \quad \forall\, \boldsymbol{x}_k, \boldsymbol{x}_l \in \partial V_{\text{low}}$$

measured from the local box simulation

that is:
generate corresponding stochastic signals $f_i(\boldsymbol{x}_k, t)$

$\longrightarrow$ a task for Nigul!

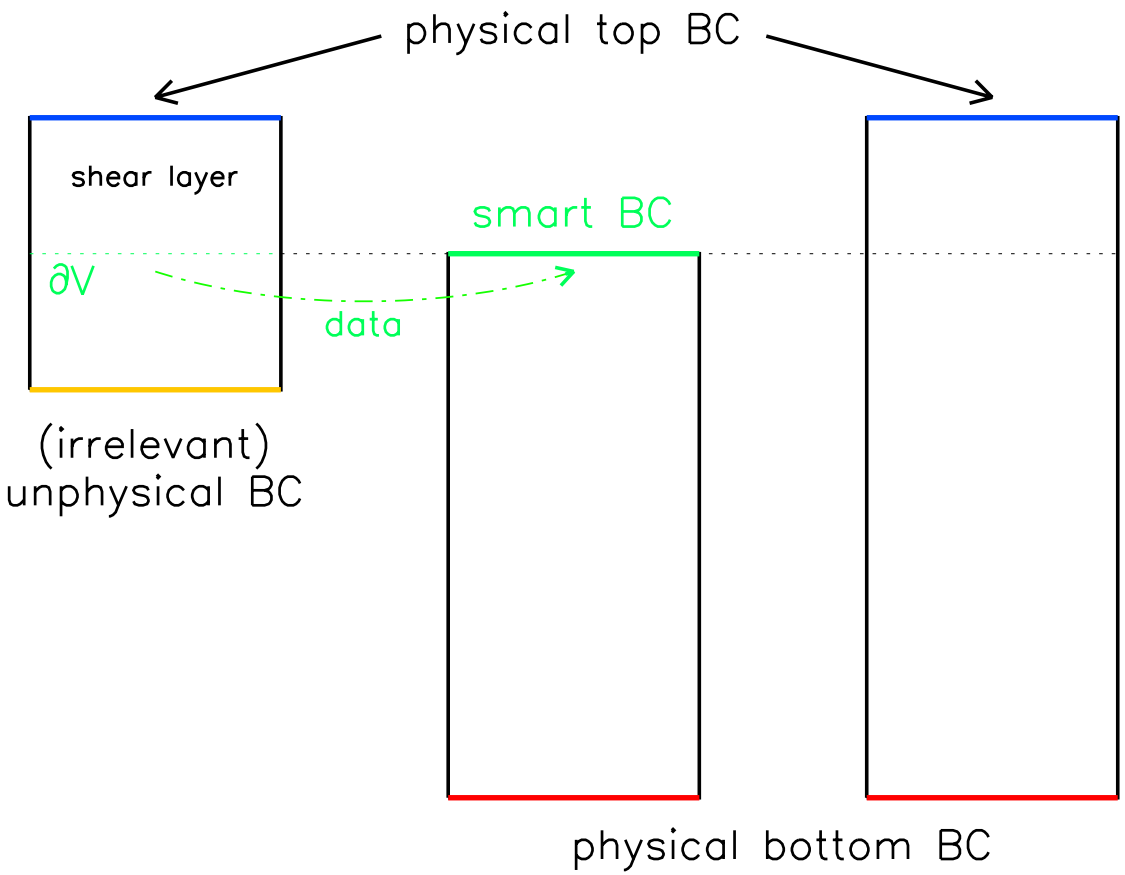
A humble solution

employ time dependent Dirichlet BCs on ∂V_{low} for all quantities with the quantities directly taken from the local box simulation

Problems:

- output cadence of local box simulation limited
- temporal interpolation
- *randomized repetition*

Scheme for testing



local box | global CZ simulation | reference

Implementation

- new BC type `'slc'` = data from slices for all simulated quantities
- refers to slices
  `['xy','xz','yz']` and
  `['xy2','xz2','yz2']`
  for lower and upper $[z, y, x]$ boundary, respectively
- data read from different run directory (`bc_slc_dir`)
  as “scattered array" (linked list)
- used in `set_from_slice_[xyz]` as Dirichlet condition,
  combined with one-sided difference formulae
- linear interpolation in time
- no randomized repetition yet