Data and code description for manuscript, "Natural convection in the cytoplasm: Theoretical predictions of buoyancy-driven flows inside a cell"

Nikhil Desai, Weida Liao and Eric Lauga

June 2024

1 Data description

We provide here a description of the data-files containing the COMSOL simulation data, used in the manuscript "*Natural convection in the cytoplasm: Theoretical predictions of buoyancy-driven flows inside a cell*", by Nikhil Desai, Weida Liao and Eric Lauga.

The data for a given figure is shared in the file identified by the figure number, e.g., the data for Fig. 7(b) will be in the file named `fig7b.txt`. The following types of data have been shared:

1. **Contour plot data for temperature, flow or concentration:** Any file named `figxx.txt` contains the contour plot data shown in figure number `xx`. This contour can be of the temperature field (e.g., the temperatures in Fig. 3), the velocity components inside the cell (e.g., Fig. 7(b)), or the solute concentration distribution (e.g., Fig. 9).

   Each such file contains 4 columns of numbers. Each row of the dataset corresponds to data for a single point inside the cell. Entries of the i-th row can be described as (x_i, y_i, z_i, p_i). The first three entries are spatial coordinate data (x_i, y_i, z_i) of points inside the cell. The fourth entry p_i contains the value of the temperature/flow-field component/solute concentration at the point (x_i, y_i, z_i).

   The data has been provided for a sufficiently large number of points, such that the value of the property p_q at any query point (x_q, y_q, z_q) can be obtained via linear interpolation of existing data, without any practical loss of accuracy.

   Note: In files named `fig3_...txt`, the label `top` corresponds to figures in the top row, and `bot` corresponds to figures in the bottom row of Fig. 3. The labels `Temp`, `ux` and `uz` are files containing data for the temperature, x'-velocity and z'-velocity, respectively.

2. **Velocity vector data:** Files named `figxx_vectors.txt` contain data for the velocity vectors from figure number `xx`.

   Each such file contains 4 columns of numbers. Each row corresponds to data for a single point inside the cell. Entries of the i-th row can be described as $(c_{1,i}, c_{2,i}, u_{1,i}, u_{2,i})$. The first two entries contain coordinate data ((x_i, y_i), or (y_i, z_i), or (x_i, z_i)) and the second two entries contain data for the velocity vectors (respectively, $(u_{x,i}, u_{y,i})$, or $(u_{y,i}, u_{z,i})$, or $(u_{x,i}, u_{z,i})$).

3. **Contour plot for the maximum local Péclet number:** This data is for solute transfer simulations (due to cytoplasmic convection + molecular diffusion) of Section 6. It is provided in a MATLAB datafile called `fig10a.mat`. The file contains three matrices of size 11×7, named '`D`', '`dT`' and '`Pe_l_max`:

   (a) `D` is a matrix where each row is a copy of 7 distinct solute diffusivity values, say D_j, ranging from $D_7 = 0.01$ μm^2 s^{-1} to $D_1 = 100$ μm^2 s^{-1}.

   (b) `dT` is a matrix where each column is a copy of 11 distinct values of the nucleus-to-membrane temperature difference, say ΔT_i, ranging from $\Delta T_1 = 0.01$ K to $\Delta T_{11} = 10$ K.

   (c) `Pe_l_max` is a matrix whose i-th row ($1 \leq i \leq 11$) and j-th column ($1 \leq j \leq 7$) correspond to the maximum local Péclet number for mass-transport simulations with ΔT given by the i-th row of `dT` and D given by the j-th column of `D`, i.e. $\mathrm{Pe}^{\ell}_{\mathrm{max,(i,j)}}$ corresponds to simulation with ΔT_i and D_j.

4. **Line plots for the maximum and minimum vertical flow inside cells with different nucleus size and placement:** This is the data for the line plots in Fig. 13(a-b). It is provided in two `.mat` files: `kappa=2e-1.mat` for Fig. 13(a) and `kappa=43e-2.mat` for Fig. 13(b). Each of these files contains one variable named `kappa`, which is the value of the nucleus-to-cell radius ratio for which the data has been recorded. Additionally, there is an array named `theta_e` which contains the values of the 'nucleus angle' θ_e (see Fig. 1) in the manuscript, for which the maximum and minimum velocities have been plotted in Figs. 13(a-b). Both `kappa=2e-1.mat` and `kappa=43e-2.mat` contain data arrays named as `epd_uz_max` or `epd_uz_min`. These correspond respectively, to the values of the maximum or minimum velocity along the z' direction for eccentricity given by $e = 0.\mathtt{d}$.

2 Code description

We provide here a description of the MATLAB code used to generate all results for the bi-spherical coordinates calculations in the manuscript "*Natural convection in the cytoplasm: Theoretical predictions of buoyancy-driven flows inside a cell*", by Nikhil Desai, Weida Liao and Eric Lauga. The source code directory contains six different functions and one 'driver script' named `flow_solver.m`. We describe here what each MATLAB script/function does.

Function driver

`flow_solver.m` is the main script that can be used to generate results from the paper. In this script, the user must first prescribed the following:

(a) `kappa`: radius of nucleus,

(b) `e`: eccentricity of the nucleus,

(c) `N_BiSp`: the number of unknown functions $U_n(\xi)$ (also called velocity modes) in the expansion given by eqn. (34) in the manuscript's Appendix,

(d) `M_xi`: the number of points over which the velocity modes, $U_n(\xi)$, are discretized (see eqn. (34) and the text just before eqn. (41) in the manuscript's Appendix), and,

(e) `plot_vel`: the velocity component (x or z) which the user desires to be plotted; the user can also choose to plot the velocity magnitude.

The above inputs are then passed to the function `axisymm_flow_solve_BiSp.m`.

Axisymmetric flow solver in bi-spherical coordinates

`axisymm_flow_solve_BiSp.m` is the main function, which actually solves for the flow inside the cell. This function accepts the inputs from above sub-section and returns the following as output:

(a) `uz_max`: the global maximum vertical velocities inside the cell (plotted in Fig. 5(a) in main manuscript).

(b) `uz_min`: the global minimum vertical velocities inside the cell (plotted in Fig. 5(b) in main manuscript).

(c) `Un`: an array of size `N_BiSp` $\times$ `M_xi`, of discretized velocity modes $U_n(\xi_m)$, where $1 \leq n \leq$ `N_BiSp` and $1 \leq m \leq$ `M_xi`, which can be used to reconstruct the fluid's velocity, by using eqns. (30), (31) and (34) in the manuscript's Appendix. Note that the velocity modes are output in the sequence $[\,\{U_1(\xi_1)..U_1(\xi_M)\}, \{U_2(\xi_1)..U_2(\xi_M)\}, ... \{U_N(\xi_1)..U_N(\xi_M)\}\,]$, where $N =$ `N_BiSp` and $M =$ `M_xi`.

`axisymm_flow_solve_BiSp.m` uses other important functions to carry out the computations and generate the final plots. These are described next.

Supporting functions

1. `theta_modes.m`: This function returns the values of the n^{th} temperature mode (eqn. (28) in Appendix) and its derivative. It requires, as input, the following variables:

   (a) `n`: the 'mode number',

(b) `xi`: an array of values of the bi-spherical coordinate ξ for which the user desires the output,

(c) `xi_i`: the value of the ξ-coordinate for the nuclear membrane, and,

(d) `xi_o`: the value of the ξ-coordinate for the cell membrane.

2. `lgwt.m`: Consider a continuous function $f(x)$ which is defined on the interval $[a, b]$ and which can be evaluated at any x, with $a \leq x \leq b$; according to the N-point Gauss–Legendre quadrature rule,

$$\int_a^b f(x)\, dx \approx \sum_{i=1}^{N} f(x_i) \times w_i,$$

where the x_i are called Legendre–Gauss nodes and the w_i are called Legendre–Gauss weights. The function `lgwt.m` computes (and returns) the Legendre-Gauss nodes and weights, needed to compute the definite integrals with respect to the bi-spherical coordinate χ. It requires, as input, the following variables:

(a) `N`: the number of Legendre–Gauss nodes (and weights), which decides the order of accuracy of the definite integral,

(b) `a`: the lower limit of integration (e.g., $\chi = -1$),

(c) `b`: the upper limit of integration (e.g., $\chi = 1$).

3. `I_lin_BiSp_proj.m`: This function constructs and returns the $\mathscr{I}_{lin}$ matrix (from eqns. (40) and (41) in the Appendix). It requires, as input, the following variables:

(a) `N_BiSp`: the number of bi-spherical velocity modes in the expansion given by eqn. (34) in the manuscript's Appendix,

(b) `xi`: an array of values of the bi-spherical coordinate ξ for which the user desires the output,

(c) `chi`: an array of values of the Gauss–Legendre nodes (see description for `lgwt.m`) for the bi-spherical coordinate χ, with respect to which the definite integrals are performed to evaluate $\mathscr{I}_{lin}$,

(d) `wts`: an array of values of the Gauss–Legendre weights corresponding to the array `chi` (see description for `lgwt.m`), used in evaluation of the definite integrals required for computing $\mathscr{I}_{lin}$, and,

(e) `LegPols`: the full set of Legendre polynomials required for evaluation of the definite integrals in $\mathscr{I}_{lin}$.

4. `chi_vals_surf.m`: This function returns the values of the bi-spherical coordinate χ for points separated by a constant angular separation, lying on the surface of the nucleus (inner sphere) or cell membrane (outer sphere). For example, the χ = const. curves in Fig. 15 of the manuscript have been plotted at equi-angular separations on the inner sphere. The function `chi_vals_surf.m` requires, as input, the following variables:

(a) `e`: eccentricity of the nucleus,

(b) `kappa`: radius of nucleus,

(c) `N_th`: the number of points on the surface of the sphere,

(d) `in_or_out`: the sphere ('inner sphere'=nucleus surface & 'outer sphere'=cell membrane) for which these points are needed.

The result of this function is used in `plot_results.m` (described next).

5. `plot_results.m`: This function plots the fluid velocity inside the cell. It requires, as input, the following variables:

(a) `Un`: the array of discretized velocity modes $U_n(\xi_m)$ (see description of `axisymm_flow_solve_BiSp.m` for structure of the `Un` array),

(b) `xi`: the array of the discretized points ξ_m over which the modes $U_n(\xi_m)$ have been defined/computed,

(c) `N_modes`: number of bi-spherical velocity modes in the expansion given by eqn. (34) in the manuscript's Appendix,

(d) `chi`: values of the bi-spherical coordinate χ, evaluated along points separated by a constant angular separation, on the surface of the nucleus (inner sphere) or cell membrane (outer sphere); this is the output of the function `chi_vals_surf.m`,

(e) `e`: eccentricity of the nucleus,

(f) `kappa`: radius of nucleus,

(g) `plot_vel`: user-specified input, for which component of the velocity x or z needs to be plotted; there is also an option to plot the magnitude of the velocity.