SDS package: A Stokesian dynamics and electric charge transfer simulator for particulate suspensions

Mehdi Karzar-Jeddi

April 14, 2020

Contents

1 SDS input and output setup **5**

1.1 Introduction 5

1.2 Requirements 5

1.3 Input files 5

1.3.1 `input.txt` 6

1.3.2 `location.txt` 8

1.3.3 `force.txt` 8

1.3.4 `rc.txt` 9

1.4 Output files 9

1.4.1 `dump.dis_xxx.cfg` 9

1.4.2 `tec_xxx.dat` 9

1.4.3 `velocity.txt` 10

1.5 Parameters 10

1.6 Reference 10

2 Study cases **11**

2.1 Two particles with different sizes under external force 11

2.2 A column of five same size particles under external force 11

2.3 A cluster of 5 by 5 randomly distributed particles under shear 12

2.4 Time-reversibility of Stoeks flow 12

2.5 Sedimentation of uncharged carbon beads to charged current collector surface 13

Chapter 1

SDS input and output setup

1.1 Introduction

This manual describes input set up and output files of SDS package, a Stokessian Dynamics Simulator. SDS simulates time evolution of particle configurations in a quasi steady creeping flow conditions. SDS also simulates electric charge transfer between particles. Time evolution of particles configuration and hydrodynamic interactions between particles is simulated using Stokesian dynamics and the electric potential interactions between particles are simulated using Resistor-Capacitor formulation. Details of theoretical formulations are published in two publications that are listed in Reference section.

1.2 Requirements

The SDS code can run on both Windows and Linux servers. On a Windows server, the code can be compiled using Microsoft Visual Studio. Perquisite for using Microsoft Visual Studio to compile SDS is installation of LAPACK package, linear algebra package. Instructions on how to install LAPACK in Microsoft Visual Studio are provided in:

https://icl.cs.utk.edu/lapack-for-windows/VisualStudio_install.html

On a Linux server, the code can be compiled using `g++` compiler. Before installing SDS, LAPACK package has to get installed and the links for LAPACK in `makefile` has to be updated to LAPACK installation address.

1.3 Input files

There are three input files for setting up simulation: `input.txt`, `locationtxt`, and `forcetxt`. Input files include header before parameters that are between `% ... %` and sometimes there are notes after

introducing the parameters. **It is important to keep the headers and comments untouched**. In this section all of input files are introduced.

1.3.1 `input.txt`

`input.txt` defines most of the setup for simulation and how the code should run. This file consists of input values for many of variable and flags for set up the code. This is to let the code understand what is the time step, what is the length of simulations, what is the number of particles, what are the initial configuration, should lubrication approximation accounted, and so. Note that all parameters in input file should have a value regardless if they are being used or not. The enteries are to set up in few categories:

Set up flags and parameters

- `deltat`: is size of time step for transient simulation
- `nstep`: is the total number of time steps
- `sstep`: is starting time step or initial time step (if you are starting from initial time, you may set it to 0)
- `nsamp`: is the periodic saving time steps
- `version`: defines the mobility tensor version, 0 is for F, 1 is for FT, and 2 is for FTS. Note that for simulation near to wall ** version bigger than 1 is not allowed.
- `flag_poly`: set it to 0 for monodisperese particle size and set it to 1 for polydisperse size distribution
- `flag_mat`: the code was being developed in two versions Matrix based and non matrix based. Later I decided to develop the code based on mobility matrix so keep it at 1
- `flag_lub`: set it to 0 to solve set up with out lubrication assumption and set it to 1 for with lubrication consideration.
- `flag_prob`: 0 is to define problem in free space, 1 is to set problem near a no-slip wall.
- `flag_fix`: 0 is to set mobile particles only simulations, 1 is to set fixed particles next to mobile particle at one side, 2 is to set up fixed particles at both sides of simulation cell.
- `flag_per`: 0 value is to set free space simulation, 1 is to set periodic cell in x direction only, 2 is to set peridic cell in xy- plane, and 3- is to set xy- Lees-Edwards periodic boundary conditions.
- `flag_cper`: is to set charge of incoming particles in periodic cell. 0 means particles in periodic cells will have zero charge, and 1 means that the particles in periodic cells will exchange charge with the main cell. **Note that `flag_cper` flag has to be consistent with `flag_per`.**

- `periodic`: 0 is to use non=Ewald summation formulation and 1 is to set for Ewald summation for far-field hydrodynamic interactions. **Note that Ewald formulation is an underdevelopment part of the code for treatment of far-field interactions and work on this is not finished yet. Do not set the flag to 1.**

- `flag_fread`: is to set initial particles configuration to generated by code (value 0) or to be read from file (value 1)

- `flag_config`: is a parameter to set the initial particle configuration from a file: 0: read the location and initial charge from `location.txt`, 1: simple crystal (SC), 2: BCC cruystal, 3: FCC crystal, 4: SC with random disturbance, and 5: read a former run results `dump.dis_xxx.cfg`, where `xxx` is the value of `sstep`. Note that if you set the input to 5 you still need to enter number of mobile and total number of particles in `location.txt`.

- `phi`: uniform volume fraction of initiral particle configuration

- `nx`: is the number of initial particles in x direction

- `ny`: is the number of initial particles in y direction

- `nz`: is the number of initial particles in z direction

Far field velocities $\mathbf{U}_\infty$

Far field velocity in x, y, and z directions are introduced as

- `U_x`

- `U_y`

- `U_z`

Far field strain rate $\mathbf{E}_{ii}$

Three dimensional far field strain is introduced as

- `E_xx`

- `E_xy`

- `E_xz`

- `E_yz`

- `E_zz`

Lubrication parameters

These are parameters for how the lubrication function has to be introduced.

1.3.2 location.txt

This file defines initial location of particles in case the `flag_config` is set to 0. The following information need to be filed in exact same order (after the header)

1. `np`: number of particles.
2. `nm`: number of mobile particles (number of fixed particles is equal to `np-nm`.
3. `xlo`: lower limit of simulation cell in x direction.
4. `xhi`: upper limit of simulation cell in x direction.
5. `ylo`: lower limit of simulation cell in y direction.
6. `yhi`: upper limit of simulation cell in y direction.
7. `zlo`: lower limit of simulation cell in z direction.
8. `zhi`: upper limit of simulation cell in z direction.
9. `xlo`: lower limit of simulation cell in x direction.
10. `particle_id x y z a charge`
11. `particle_id x y z a charge`
12. `...`

where `particle_id` is the id number of particle, `x` is the x location of particle, `y` is the y location of particles, `z` is the z location of the particles, `a` is the radius of particle, and `charge` is the initial charge of the particles.

1.3.3 force.txt

This file is to set up external force that is applied to each of the mobile particles or to set up velocity to mobile particles. This file will be read if `flag_fread` set to 1. The entry has to be in tow batches. The first batch is for mobile particles in following format

```
prticle_id fx fy fz tx ty tz exx exy exz eyz ezz
```

where `fi` is external force acting on particle in direction `i`, `ti` is external torque acting on particle in direction `i`, and `eij` is external strain rate applied on particle location. The second batch is for setting velocity for fixed particles in following format

`prticle_id ux uy uz ox oy oz exx exy exz eyz ezz`

where `ui` is transnational velocity of particle in direction `i`, `ti` is rotational velocity of particle in direction `i`, and `eij` is external strain rate applied on particle location.

1.3.4 `rc.txt`

This file sets up the input for charge transfer calculations. All inputs are in a single batch and has to enter in following order right after the header

1. `Rw`: the electrical resistance between particles and wall in Ohms
2. `Rij`: the electrical resistance between two particles in contact in Ohms
3. `Resr`: equivalent series resistance in Ohms
4. `Cap`: capacitance of each particle in Farad
5. `touch`: is the touch limit for two particle as percent of particle radii
6. `nx`
7. `ny`

`nx` and `ny` were supposed to be used in a post processing scheme which is deactivated at the moment. Never the less a value should be introduced for each of them. Note that in an earlier implementation of charge transfer, the value of each capacitor was supposed to enter through `capacitance.txt` file. However, in later version of charge transfer model this option commented out, instead a unique value for all capacitors can be entered in `rc.txt`

1.4 Output files

Out put files for simulation are

1.4.1 `dump.dis_xxx.cfg`

cfg files include the location of particles at time step xxx. cfg files can be read by Ovito pacakge.

1.4.2 `tec_xxx.dat`

tec files are similar to cfg files but they can be read by tecplot. Note that output for tec files might be nulled.

1.4.3 `velocity.txt`

1.5 Parameters

1.6 Reference

- M. Karzar-Jeddi, H. Luo, P.T. Cummings, Mobilities of polydisperse hard spheres near a no-slip wall, *Computers & Fluids*, **176**, 40-50, 2018.

- M. Karzar-Jeddi, H. Luo, P.T. Cummings, K.B. Hatzell, Computational modeling of particle hydrodynamics and charging process for the flowable electrodes of carbon slurry, *Journal of Electrochemical Society*, **166**(12), A2643-A2653, 2019.

Chapter 2

Study cases

In this chapter we present some of the studied cases using SDS package. Input files for some of the cases are provided in github page of the package in: https://github.com/mkjeddi/SDS

2.1 Two particles with different sizes under external force

The first example is a simulation of time evolution of configuration of two unequal particles, $i = 1$ and 2, under external force (see Fig. 2.1). The ratio of radius of particles is $a_1/a_2 = 2$. Input files for this example are provided in github link.

2.2 A column of five same size particles under external force

This case is a simulation of time evolution of configuration of particles under an external force (see Fig. 2.2). The external force is applied in normal to column. Input files for this example are provided in github

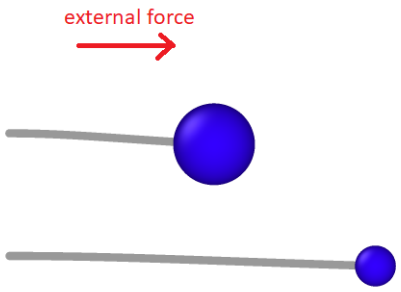

Figure 2.1: Two particles with radius ration of $a_1 / a_2 = 2$ under effect of external force $6\pi a_i \eta$. The trajectory of particles from start point is presented by grey line.

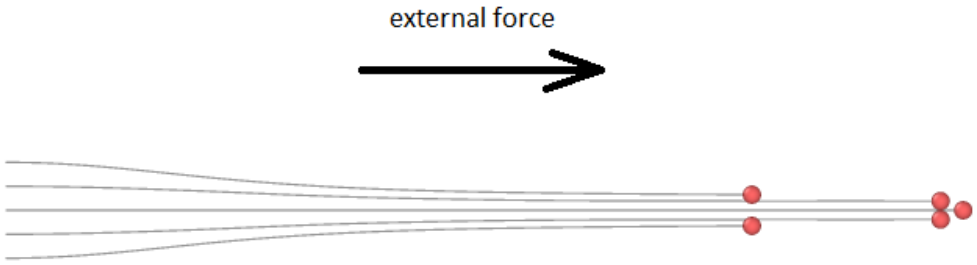

Figure 2.2: Trajectory of five identical hard spheres under equal external force starting from initial configuration of a column.

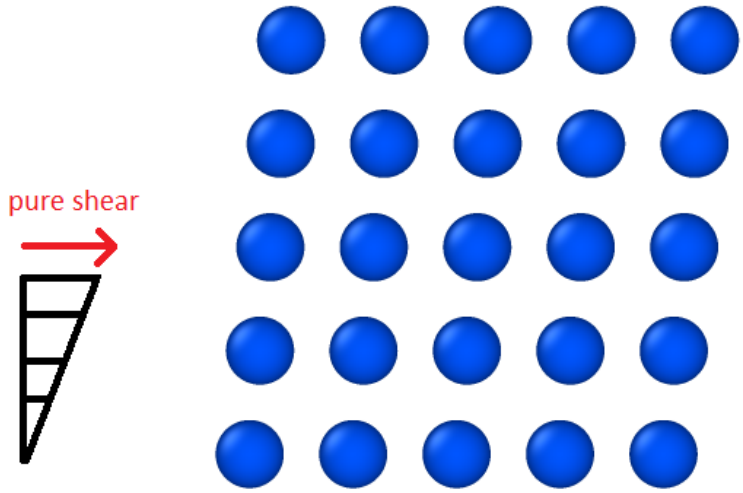

Figure 2.3: Time evolution of mono-layer of identical hard sphere under pure shear.

link.

2.3 A cluster of 5 by 5 randomly distributed particles under shear

This case is simulation of time evolution of a mono-layer of 5 by 5 identical particles initially configured in simple crystal form (see Fig. 2.3). For this simulation there is no external force on particles however the pure shear is set by setting E_{xy}= 1 and Ω_z=-1. Boundary conditions for this simulation is Lees-Edwards boundary conditions. Input files for this example are provided in github link.

2.4 Time-reversibility of Stoeks flow

Time evolution of configuration of a chain of particles under pure shear (see Fig. 2.4). Then a pure shear is applied in direction normal to the orientation of the chain. After running simulation for 200 time steps with

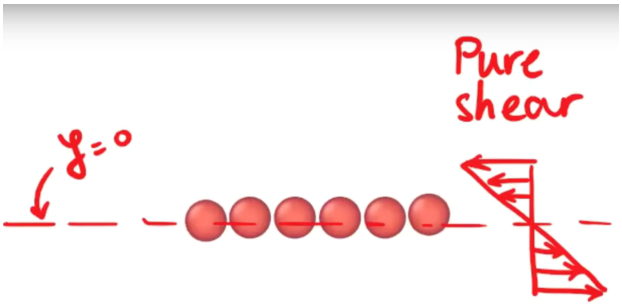

Figure 2.4: Time evolution of six particles under pure shear and then reverse the shear direction. This setup is to test the code with reversibility of Stokes flow (see simulation video in YouTube link).

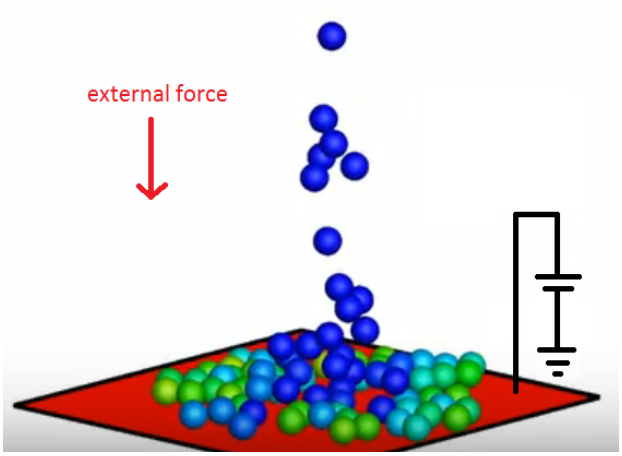

Figure 2.5: A setup to simulate sediment of carbon current capacitors into a current collector (see simulation video in YouTube link). Color coding is showing the electric potential of the particles, red color shows fully charged and blue color presents uncharged particles.

a dimensionless time step of 0.001, the shear direction is reversed. As it is shown in the video the chain of particles unfolds to its initial configuration. an animation of the test is presented in the following YouTube link: YouTube.

2.5 Sedimentation of uncharged carbon beads to charged current collector surface

In this case the sedimentation of carbon beads, flow capacitors, on a current collector surface is simulated (see Fig. 2.5). The initial electric charge of the capacitors is considered to be zero and the current collector is a conducting plate connected to an electric source. Initially the flow capacitors are distributed in a vertical column with random disturbance. The beads, then start moving under gravity force toward the plate while they are interacting hydro dynamically with each other and with the current collector surface which is simulated as a no-slip surface. During this interactions particles exchange electric charges with each other and with the current collector.