

重力と熱流下における粒子集団の様相

理学部理学科物理学コース 学籍番号 20S2035Y 山本 凜

最終更新日: 2023 年 12 月 28 日

目次

1	実験	3
1.1	追実験	4
1.2	重力と熱流を同時にかける	6
1.3	重力を先にかけて, 熱流を後からかける	9
1.4	重力のみをかける	10
1.5	熱流のみをかける	10
付録 A	ソースコード	10

1 実験

この章では, 行ったシミュレーションの設定についてそれぞれ説明をしていく.

系の上下両端のポテンシャルエネルギー差 mgL_y と運動エネルギー差 $k_B\Delta T$ の比を χ として以下のように設定する.

$$\chi \equiv \frac{k_B\Delta T}{mgL_y}$$

壁ポテンシャルまわりの無次元パラメータを 3 つ用意する.

R_d : 乾き具合.

R_t : 壁の厚み.

R_a : 濡れ具合.

これらを用いて, 壁-粒子間相互作用 LJ ポテンシャルは以下のように書き表す.

$$\varepsilon^{\text{wall}} = (1.0 - R_d) \times \varepsilon$$

$$\sigma^{\text{wall}} = (0.5 + R_t) \times \sigma$$

$$r_{\text{cut}}^{\text{wall}} = \left(2^{1/6} + R_a\right) \times \sigma^{\text{wall}}$$

パラメータ (R_d, R_t, R_a) を変えることによって, 壁-粒子間相互作用 LJ ポテンシャルが変わる. このときに, どのようにして粒子集団の様相が変化するかをみる.

R_a と R_t を少しずつ変えた系でシミュレーションをして, 粒子集団の様相の変化を見たい. 以下に示すが, それらを動かす範囲である.

$$R_a: 0.0 \sim 3.0 - 2^{1/6} = 1.877538\dots$$

$$R_t: 0.0 \sim 0.5$$

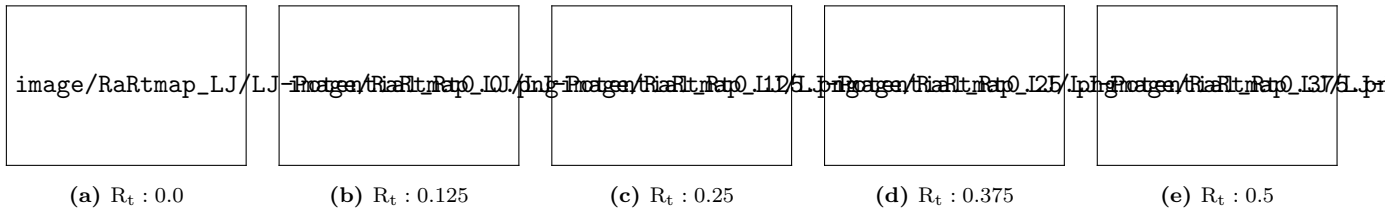


図 1

本章の以降の実験は特記がない限り以下のパラメータに近い値で行うものとする.

- $N = 1250$: 粒子数

- $\rho\sigma^2 = 0.4$: 粒子数密度
- $L_x/\sigma = 39.528471 \simeq 39.5$: 系の x 幅
- $L_y/\sigma = 79.0569414 \simeq 79.0$: 系の y 幅
- $k_B T/\varepsilon = 0.43$: 初期温度
- $k_B \Delta T/\varepsilon = 0.04$: 熱浴の温度差
- $mg\sigma/\varepsilon = 0.0003999718779659611 \simeq 4.0 \times 10^{-4}$: 粒子にかかる重力の大きさ
- $dt\sqrt{\varepsilon/m\sigma^2} = 0.005$: シミュレーションにおける時間刻み.

以下に記すのは, 今後解析をする際に示すシミュレーションについての時間に関する説明である.

t_i : シミュレーション開始時から, 物理量を解析する際にデータを採用し始める時間. これ以降は定常状態であるとみなす.

t_f : シミュレーション開始時から, シミュレーションの終了時までの時間.

いずれの実験の場合も $t\sqrt{\varepsilon/m\sigma^2} = 0$ の時点では粒子は以下の画像のように, 系に規則正しく並べられているとする.



図 2: $N = 1250$

1.1 追実験

壁を完全に濡らしている状態を考えたいので, $r_{\text{cut}}^{\text{wall}} = 3.0\sigma$ に設定する.

また, 重力をかけた状態で粒子が下に落ちきり, 緩和しているとみなせるまでシミュレーションを行ってから, 熱流をかけている. この時点でのシミュレーションではデータを解析することはない.

- $N = 5000$
- $\rho\sigma^2 = 0.4$
- $L_x/\sigma = 79.0 \dots$
- $L_y/\sigma = 158.1 \dots$

- $k_B T / \varepsilon = 4.3$
- $k_B \Delta T / \varepsilon = 0.0$
- $mg\sigma / \varepsilon = 2.0 \times 10^{-4}$
- $t_f \sqrt{\varepsilon / m\sigma^2} = 5.0 \times 10^5$

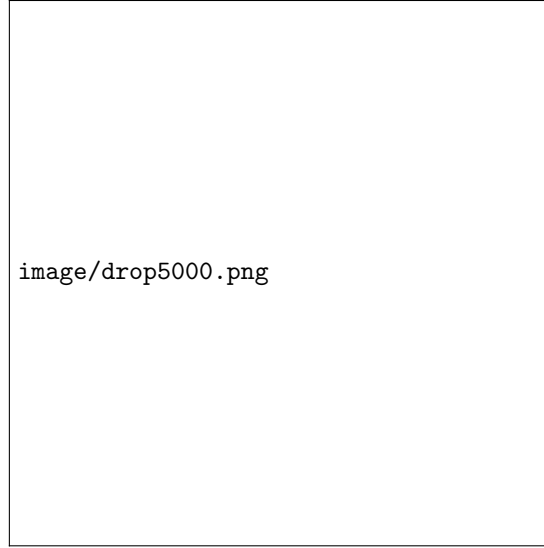


図 3: $t\sqrt{\varepsilon/m\sigma^2} = 5.0 \times 10^5$ のスナップショット

続いて重力をかけた緩和後の系で、熱浴の温度差を改めて以下のようにつけ、熱流をかけてシミュレーションをしている。このシミュレーションでデータを解析する。

- $\chi = k_B \Delta T / mgL_y = 1.265$
- $k_B \Delta T / \varepsilon = 0.04$
- $t_i \sqrt{\varepsilon / m\sigma^2} = 5.0 \times 10^5$
- $t_f \sqrt{\varepsilon / m\sigma^2} = 1.0 \times 10^6$

($R_d = 0.0, R_t = 0.5, R_a = 3.0 - 2^{1/6}$) で実験を行う。これは、($\varepsilon^{\text{wall}} = \varepsilon, \sigma^{\text{wall}} = \sigma, r_{\text{cut}}^{\text{wall}} = 3.0\sigma^{\text{wall}}$) に対応しているので、 $\phi_{\text{LJ}}^{\text{wall}} = \phi_{\text{LJ}}$ ということになり、先行研究と同じ結果を得ることができる。



図 4: Ay100_rho0.4_T0.43_dT0.04_Rd0.0_Rt0.5_Ra1.877538_g0.0004_run1.0e8

1.2 重力と熱流を同時にかける

以下のように, R_a と R_t を少しずつ変えた系を設定して, それぞれシミュレーションをした.

	$R_a : 0.0$	$R_a : 0.469$	$R_a : 0.938$	$R_a : 1.408$	$R_a : 1.877$
$R_t : 0.0$	a	b	c	d	e
$R_t : 0.125$	f	g	h	i	j
$R_t : 0.25$	k	l	m	n	o
$R_t : 0.375$	p	q	r	s	t
$R_t : 0.5$	u	v	w	x	y

パラメータを確認する.

- $N = 1250$
- $\rho\sigma^2 = 0.4$
- $L_x/\sigma = 39.5 \dots$
- $L_y/\sigma = 79.0 \dots$
- $k_B T/\varepsilon = 0.43$
- $k_B \Delta T/\varepsilon = 0.04$
- $mg\sigma/\varepsilon = 4.0 \times 10^{-4}$
- $t_f \sqrt{\varepsilon/m\sigma^2} = 2.0 \times 10^5$

この際の粒子集団の様相は以下ようになる.

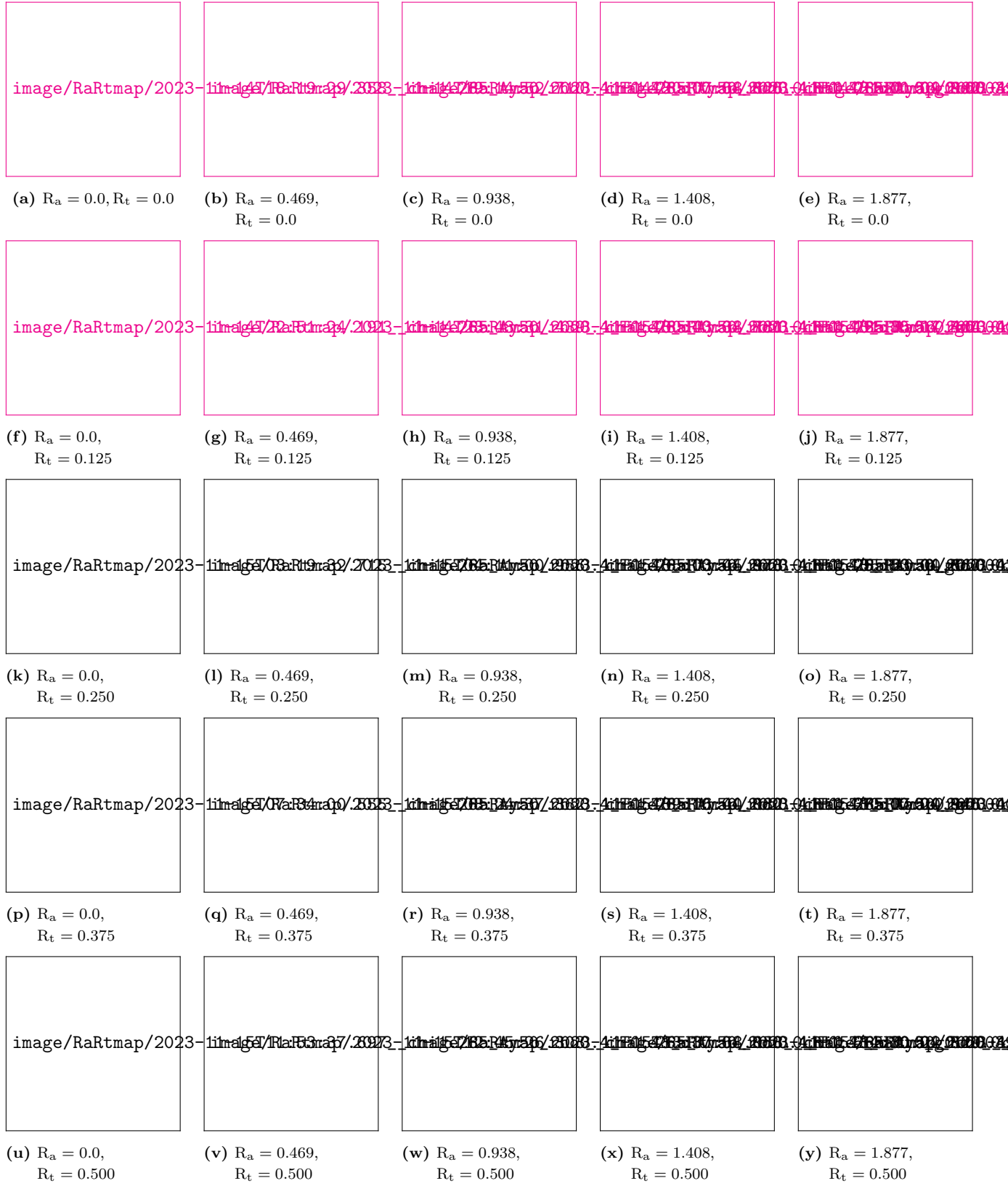


図 5: リンク先の動画は $t\sqrt{\varepsilon/m\sigma^2} = 200$ ごとに表示. スナップショットは $t\sqrt{\varepsilon/m\sigma^2} = t_f\sqrt{\varepsilon/m\sigma^2}$ 時.

重心位置 Y_g を系の y 幅でスケーリングして、時系列プロットすると、

$$Y_g \equiv \bar{y}_i = \frac{1}{N} \sum_i y_i$$

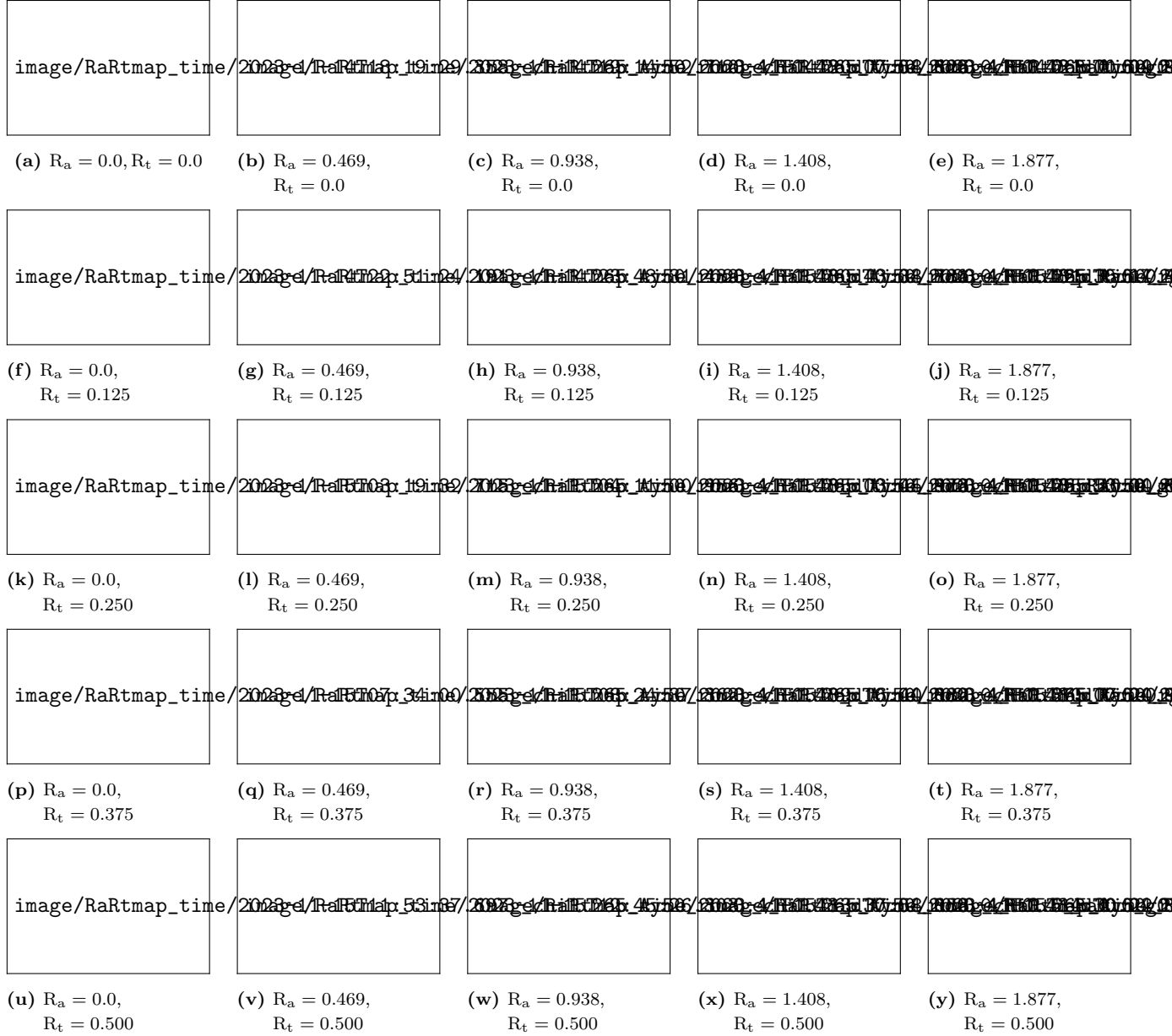


図 6: $t_i = 0, t_f = 2.0 \times 10^5, dt\sqrt{\varepsilon/m\sigma^2} = 0.005, t\sqrt{\varepsilon/m\sigma^2} = 200$ ごとにプロット。

1.3 重力を先にかけて、熱流を後からかける

以下は、追実験のときと同じように、まず重力のみをかけて、粒子集団が落ちきってから熱流をかけると同時に測定を開始するものである。

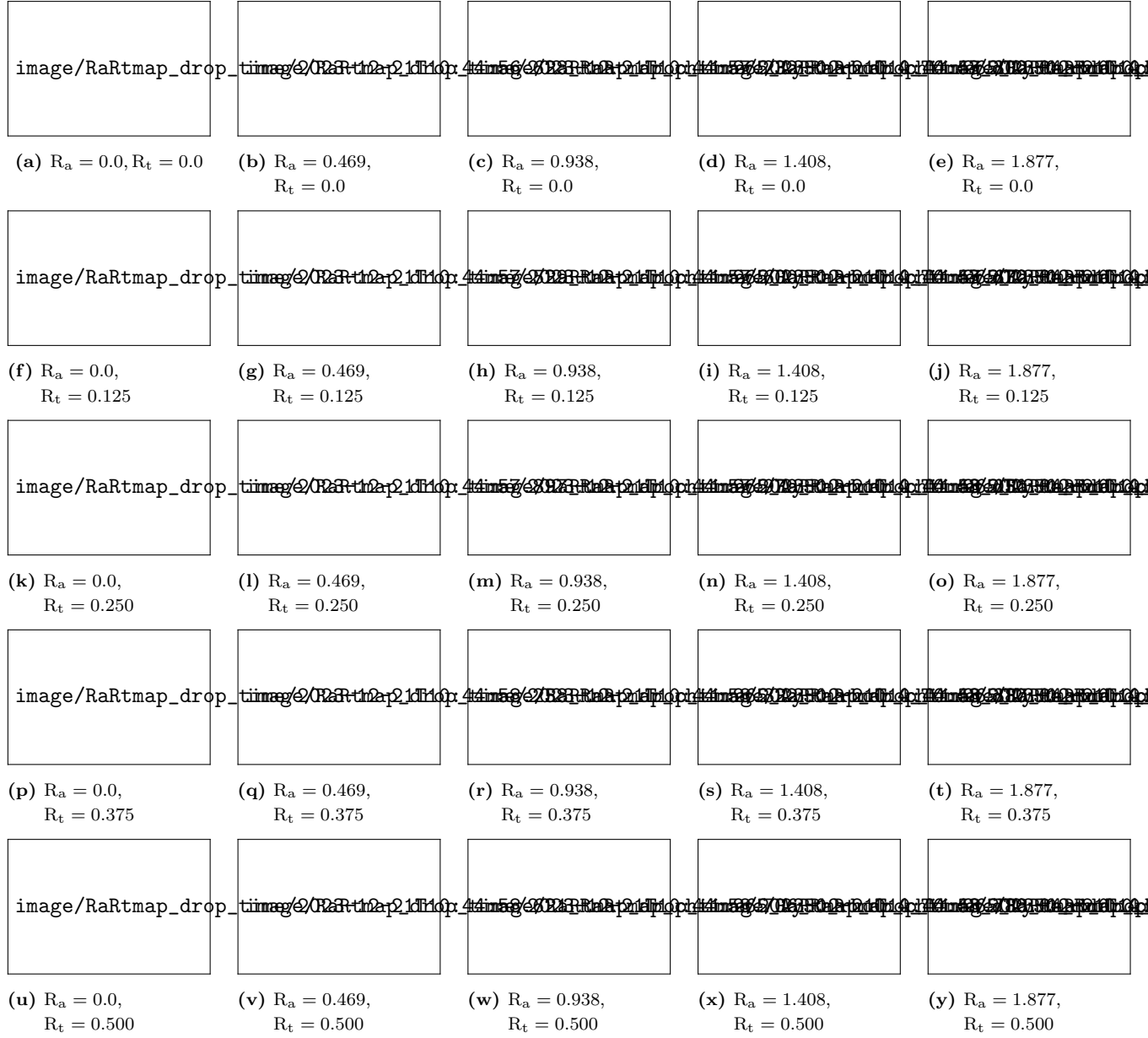


図 7: $t_i = 2.0 \times 10^5, t_f = 4.0 \times 10^5, dt\sqrt{\varepsilon/m\sigma^2} = 0.005, t\sqrt{\varepsilon/m\sigma^2} = 200$ ごとにプロット.

1.4 重力のみをかける

- $N = 1250$
- $\rho\sigma^2 = 0.4$
- $L_x/\sigma = 39.5 \dots$
- $L_y/\sigma = 79.0 \dots$
- $k_B T/\varepsilon = 0.43$
- $k_B \Delta T/\varepsilon = 0.0$
- $mg\sigma/\varepsilon = 4.0 \times 10^{-4}$
- $t_f \sqrt{\varepsilon/m\sigma^2} = 2.0 \times 10^5$

1.5 熱流のみをかける

- $N = 1250$
- $\rho\sigma^2 = 0.4$
- $L_x/\sigma = 39.5 \dots$
- $L_y/\sigma = 79.0 \dots$
- $k_B T/\varepsilon = 0.43$
- $k_B \Delta T/\varepsilon = 0.04$
- $mg\sigma/\varepsilon = 0.0$
- $t_f \sqrt{\varepsilon/m\sigma^2} = 2.0 \times 10^5$

付録 A ソースコード

ソースコード 1: in.2DLJ_mod

```
1 # 2d Lennard-Jones
2
3
4 # 出力関係のパラメータ
5 variable run equal PLACEHOLDER_run
6 variable thermo equal ${run}/1000 # 分母の数がlogで生成される行数になる。
7 variable dump equal ${run}/1000 # 分母の数がlammppstrjで生成される行数になる。
8 variable image_step equal ${run}/4 # 分母の数+1枚の画像を作成。
9
10 # 重要なパラメータ
11 variable SEED equal 202035
12 variable Ay equal PLACEHOLDER_Ay # 粒子生成に用いるy方向でのセル数。
13 variable Ax equal ${Ay}/2 # 粒子生成に用いるx方向でのセル数。
14 variable rho equal PLACEHOLDER_rho # 密度。密度と粒子数から体積が決まる。
15 variable trange equal 8 # 各熱浴の幅。
```

```

16 variable gap equal 0.5 # box と catom のずれ. ずらさないと粒子が消えてしまう.
17 # lo,hi が単に座標の小さい大きいであることに注意.
18 variable T equal PLACEHOLDER_T # 各熱浴の目標温度の中間, これを初期温度に設定.
19 variable dT equal PLACEHOLDER_dT
20 variable tlo equal ${T}+(${dT})/2 # 座標の小さい方の熱浴の目標温度.
21 variable thi equal ${T}-${dT}/2 # 座標の大きい方の熱浴の目標温度.
22 variable g equal PLACEHOLDER_g # 重力加速度.
23 # 粒子-粒子間のLJ ポテンシャル
24 variable epsilon_pair equal 1.0 # LJ ポテンシャルの epsilon; ポテンシャルの深さ.
25 variable sigma_pair equal 1.0 # LJ ポテンシャルの sigma; 衝突直径.
26 variable rc_pair equal 3.0 # 典型的なカットオフ長.
27 # 壁-粒子間のLJ ポテンシャル
28 # variable r_thickness equal PLACEHOLDER_rt # 壁の厚みに相当する.
29 # variable r_attractive equal PLACEHOLDER_ra # 引力ポテンシャルに影響.
30 variable epsilon_wall equal 1.0 # LJ ポテンシャルの epsilon; ポテンシャルの深さ.
31 variable sigma_wall equal 1.0 # LJ ポテンシャルの sigma; 衝突直径.
32 variable rc_wall equal 3.0 # WCA ポテンシャルになるようなカットオフ長+alpha*
    sigma_wall.
33
34 # 領域関係のパラメータ
35 # 縦長のとき
36 variable box_xlo equal 0 # x の小さい方の直線.
37 variable box_xhi equal ${Ax} # x の大きい方の直線.
38 variable box_ylo equal -${gap} # y の小さい方の直線.
39 variable box_yhi equal ${Ay}-${gap} # y の大きい方の直線.
40 variable coldlo equal -${gap} # 熱浴で温度の低い方の小さい方の直線.
41 variable coldhi equal -${gap}+${trange} # 熱浴で温度の低い方の大きい方の直線.
42 variable hotlo equal ${Ay}-${gap}-${trange} # 熱浴で温度の高い方の小さい方の直線.
43 variable hothi equal ${Ay}-${gap} # 熱浴で温度の高い方の大きい方の直線.
44 # # 横長のとき
45 # variable box_xlo equal -${gap}
46 # variable box_xhi equal ${Ax}-${gap}
47 # variable box_ylo equal 0
48 # variable box_yhi equal ${Ay}
49 # variable coldlo equal -${gap}
50 # variable coldhi equal -${gap}+${trange}
51 # variable hotlo equal ${Ay}-${gap}-${trange}
52 # variable hothi equal ${Ay}-${gap}
53
54
55 # 系の設定
56 units lj # LJ 単位系.
57 atom_style atomic # 粒子.
58 dimension 2 # 次元.
59 timestep 0.005 # MD シミュレーションの timestep.
60 boundary p f p # x=1,y=m,z=n の直線が周期境界条件.

```

```

61 lattice sq  $\{\rho\}$  # 粒子の初期配置. sq; 正方形セルの左隅に1つ置く.
62 region box block  $\{\text{box\_xlo}\}$   $\{\text{box\_xhi}\}$   $\{\text{box\_ylo}\}$   $\{\text{box\_yhi}\}$  -0.1 0.1 # 系の領域
    設定.
63 region catom block 0  $\{\text{Ax}\}$  0  $\{\text{Ay}\}$  -0.1 0.1 # 粒子生成の領域設定.
64 create_box 1 box # 系の生成.
65 create_atoms 1 region catom # 粒子の生成.
66 mass 1 1.0 # 粒子の設定.
67 velocity all create  $\{\text{T}\}$   $\{\text{SEED}\}$  dist gaussian # 粒子に温度
    t を目標とする初期速度をガウス分布に従って与える.
68
69 # 縦長のとき
70 region cold block INF INF  $\{\text{coldlo}\}$   $\{\text{coldhi}\}$  -0.1 0.1 # 熱浴Cの領域.
71 region hot block INF INF  $\{\text{hotlo}\}$   $\{\text{hothi}\}$  -0.1 0.1 # 熱浴Hの領域.
72 # # 横長のとき
73 # region cold block  $\{\text{coldlo}\}$   $\{\text{coldhi}\}$  INF INF -0.1 0.1 # 冷たい熱浴の領域.
74 # region hot block  $\{\text{hotlo}\}$   $\{\text{hothi}\}$  INF INF -0.1 0.1 # 暖かい熱浴の領域.
75
76 # 各熱浴領域の温度を計算
77 compute Tcold all temp/region cold # c_Tcold で cold 熱浴領域の温度を取得.
78 compute Thot all temp/region hot # c_Tcold で cold 熱浴領域の温度を取得.
79
80 # 粒子-粒子間相互作用ポテンシャル
81 pair_style lj/cut  $\{\text{rc\_pair}\}$ 
82 pair_coeff 1 1  $\{\text{epsilon\_pair}\}$   $\{\text{sigma\_pair}\}$   $\{\text{rc\_pair}\}$ 
83 pair_modify shift yes # ポテンシャルエネルギーが0になる距離がカットオフ長になるように全
    体的にシフトアップする.
84
85 # 高速化コマンド. neighbor list に入れる距離指定.
86 neighbor 0.3 bin
87 neigh_modify every 1 delay 0 check yes
88
89 # システムに他の操作がない場合にnve アンサンブルに一致するだけで、今回の系はlangevin 熱浴
    を用いた nvt アンサンブルであることに注意.
90 fix 1 all nve
91
92 # 壁-粒子間相互作用ポテンシャル
93 # 縦長のとき
94 fix wallylo all wall/lj126 ylo EDGE  $\{\text{epsilon\_wall}\}$   $\{\text{sigma\_wall}\}$   $\{\text{rc\_wall}\}$ 
    units box pbc yes
95 fix wallyhi all wall/lj126 yhi EDGE  $\{\text{epsilon\_wall}\}$   $\{\text{sigma\_wall}\}$   $\{\text{rc\_wall}\}$ 
    units box pbc yes
96 # # 横長のとき
97 # fix wallxlo all wall/lj126 xlo EDGE  $\{\text{epsilon\_wall}\}$   $\{\text{sigma\_wall}\}$   $\{\text{rc\_wall}\}$ 
    units box pbc yes
98 # fix wallxhi all wall/lj126 xhi EDGE  $\{\text{epsilon\_wall}\}$   $\{\text{sigma\_wall}\}$   $\{\text{rc\_wall}\}$ 
    units box pbc yes

```

```

99
100 # langevin 熱浴
101 fix hot all langevin ${T} ${T} 1.0 ${SEED} tally no # 熱浴
    Hが温度 T になるようにする.
102 fix cold all langevin ${T} ${T} 1.0 ${SEED} tally no # 熱浴
    Cが温度 T になるようにする.
103 fix_modify hot temp Thot
104 fix_modify cold temp Tcold
105
106 # 重力場
107 fix Gravity all gravity ${g} vector 0 -1 0
108
109 run 90000 # t が tau になるまで実行.
110
111 unfix hot # 熱浴H についての設定の解除.
112 unfix cold # 熱浴C についての設定の解除.
113
114 fix hot all langevin ${tlo} ${tlo} 1.0 ${SEED} tally no # 熱浴が温度
    tlo になるようにする.
115 fix cold all langevin ${thi} ${thi} 1.0 ${SEED} tally no # 熱浴が温度
    thi になるようにする.
116 fix_modify hot temp Thot
117 fix_modify cold temp Tcold
118
119
120 # 重心計算 (Center of Mass)
121 compute CoM all com # c_CoM[1]でXg, c_CoM[2]でYg を取得.
122
123
124 # 出力コマンド
125 # VMD
126 dump id all custom ${dump} output.lammpstrj id x y vx vy
127
128 # 画像
129 dump 2 all image ${image_step} image.*.jpg type type
130 dump_modify 2 pad 3
131
132 # YAML
133 fix extra all print ${thermo} ""
134   _timestep:_(step)
135   _temp:_(temp)
136   _pe:_(pe)
137   _TotE:_(etotal)
138   _xg:_(c_CoM[2])
139   _Tcold:_(c_Tcold)
140   _Thot:_(c_Thot)"" file output.yaml screen no

```

```

141
142 # log
143 thermo_style custom step temp pe etotal c_CoM[2] c_Tcold c_Thot # 出力する物理量.
144
145 # 一次元プロファイル(今は温度と密度だけ計算と出力)
146 compute chunk all chunk/atom bin/1d y lower 3.0 units box
147 fix temp all ave/chunk 100000 1 100000 chunk temp file temp_profile.profile
148 fix rhop all ave/chunk 100000 1 100000 chunk density/number file rho_profile.
    profile
149
150 thermo ${thermo} # 熱力学量の出力.
151 thermo_modify norm no # 示量的な熱力学量に調整.
152
153 run ${run} # 実行.

```

ソースコード 2: lammps_modexe.jl

```

1  #=
2  lammps ファイルの実行及び出力ファイルの保管.
3  - lammps ファイルを適切にパラメータ処理する必要がある.
4  - lammps ファイル実行時に出力される*.log ファイル,*.yaml ファイルが指定した同一フォルダに
    ,それぞれのフォルダを作成して保管される.
5  - lammps ファイルと同一ディレクトリにある*.lammprstファイルは削除.
6  - パラメータごとにlammps ファイルを編集して繰り返し実行させる.
7  =#
8
9  using Glob # *を使ってパターンマッチングするためのパッケージ.
10 using Dates #日時を取得するパッケージ.
11
12 lammpsfile = glob("in.*")[1] # 実行ファイルを指定.
13 file_list = ["log", "yaml", "lammprst"] # 扱う出力ファイルの拡張子.
14 outputpath = "/Users/2023_2gou/Desktop/r_yamamoto/Research/outputdir" # 出力ディ
    レクトリのパス.
15 remark_text="rain"
16
17 #=
18 パラメータを指定.
19 各パラメータの要素数の積数回分だけlammps が実行されるので大きくしすぎないように注意.
20 =#
21 Ay_range = range(50,length=1) # 偶数にする.
22 rho_range = range(0.4,length=1) # 密度.
23 T_range = range(0.43,length=1) # 初期温度.
24 dT_range = range(0.04,length=1) # 熱浴の温度の差.
25 g_range = range(4e-4,length=1) # 重力.
26 # rt_range = range(0.0,length=1) # 壁の厚み.
27 # ra_range = range(3.0-1.122462,length=1) # 濡れ具合.

```

```

28 run_range= range(1e7,length=1) # run step.
29
30 # 多重ループを用いてパラメータごとに実験を実行.
31 for Ay in Ay_range,
32     rho in rho_range,
33     T in T_range,
34     dT in dT_range,
35     g in g_range,
36     # rt in rt_range,
37     # ra in ra_range,
38     run_value in run_range # 変数をrun にしてしまうと julia の run('')と競合してしまう.
39
40     template_script = read(lammpsfile, String) # lammps ファイルを読み込む.
41     # パラメータ編集.
42     mod_script = replace(template_script,
43         "PLACEHOLDER_Ay" => string(Ay),
44         "PLACEHOLDER_rho" => string(rho),
45         "PLACEHOLDER_T" => string(T),
46         "PLACEHOLDER_dT" => string(dT),
47         "PLACEHOLDER_g" => string(g),
48         # "PLACEHOLDER_rt" => string(rt),
49         # "PLACEHOLDER_ra" => string(ra),
50         "PLACEHOLDER_run" => string(run_value)
51     )
52
53     tempfile = "in.temp_script" # 仮lammps ファイル.
54     fp = open(tempfile, "w") # 仮ファイルを作成して開く.
55     write(fp, mod_script) # 仮ファイルにパラメータを書き込む.
56     close(fp)
57
58     n = string(now()) # 実験日時の記録.
59     parameter = "Ay$(Ay)_rho$(rho)_T$(T)_dT$(dT)_g$(g)_run$(run_value)"
60     run('mpirun -n 4 lmp_mpi -log output.log -in $(tempfile)') # lammps の実行.
61     run('rm $(tempfile)') # 仮ファイルを削除.
62
63     # 出力ファイルの保管.
64     for file in file_list
65
66         # 読み込みに失敗したら次のループに進む.
67         try
68             readfile = glob("output.$(file)") [1] # 読み込みファイルを指定.
69             script = read(readfile, String) # 読み込みファイルを読み込む.
70             writepath = joinpath(outputpath, "$(file)dir", "$(n)_$(remark_text)$(
                parameter)_$(readfile)") # 書き込みファイルの絶対パス.
71             fp = open(writepath, "w") # 書き込みファイルを作成して開く.
72

```

```

73         if file == "log" # log ファイルのとき.
74             println(fp, "実験日時:␣$(n)") # 実験日時の書き込み.
75             println(fp, parameter) # コピー用.
76             println(fp, "備考欄:␣$(remark_text)") # 特別なことをした時の書き込み.
77             rm(readfile)
78         end
79
80         write(fp, script) # 読み込んだテキストを書き込む.
81         close(fp) # 書き込みファイルを閉じる.
82
83         if file == "yaml" # yaml ファイルのとき.
84             rm(readfile)
85         end
86
87         if file == "lampoostrj" # lampoostrj ファイルのとき.
88             rm(readfile) # lampoostrj ファイルは重いので、移動後は削除.
89         end
90
91     catch
92
93     end
94 end
95
96 end

```

ソースコード 3: plot_LJpotential.jl

```

1 # 汎用LJ ポテンシャル描画セル.
2 # パッケージ.
3 using Plots
4
5 # 関数定義.
6 function theta(r) # 階段関数.
7     return r > 0 ? 1 : 0
8 end
9 function phi(epsilon, sigma, r) # LJ ポテンシャル.
10     return 4.0 * epsilon * ((sigma/r)^12 - (sigma/r)^6)
11 end
12 function phi_tilde(r, epsilon, sigma, rc) # シフトアップとカットオフ.
13     return (phi(epsilon, sigma, r) - phi(epsilon, sigma, rc)) * theta(rc - r)
14 end
15
16 # 粒子-粒子LJ ポテンシャルのパラメータ.
17 epsilon = 1.0
18 sigma = 1.0
19 rc = 3.0 * sigma

```



```

20
21 # Rd, Rt, Ra の配列.
22 Rd_values = range(0.0, length=1)
23 Rt_values = range(0.0,0.5, length=5)
24 Ra_values = range(1.877, length=1)
25
26 # プロット概形.
27 plot(xlabel="r/σ", ylabel="εcf^95/ε")
28 xlims!(0.2,2.5)
29 ylims!(-1.5,3.0)
30 title!("LJ-Potential vs. r")
31 xlabel!("r/σ")
32 ylabel!("εcf^95/ε")
33
34 # 粒子-粒子LJ ポテンシャルのプロット.
35 plot!(r -> phi_tilde(r, epsilon, sigma, rc), label="Potential_pair; ε=$(round(
    epsilon,digits=1)), σ=$(round(sigma,digits=1)), rc=$(round(3.0,digits=2)) σ",
    linestyle=:dash)
36
37 # プロットの追加.
38 for Rd in Rd_values,
39     Rt in Rt_values,
40     Ra in Ra_values
41     # 壁-粒子LJ ポテンシャルのパラメータ.
42     epsilon_wall = (1.0 - Rd) * epsilon
43     sigma_wall = (0.5 + Rt) * sigma
44     rc_wall = ((2 ^ (1 / 6)) + Ra) * sigma_wall
45     # 打つ点を調整.
46     x_values = range(Rt+0.3,3.0,length=10000)
47     y_values = phi_tilde.(x_values, epsilon_wall, sigma_wall, rc_wall)
48     # 壁-粒子LJ ポテンシャルのプロット.
49     plot!(x_values, y_values, label="Potential_wall; εw=$(round(epsilon_wall,
        digits=1)) ε, σw=$(round(sigma_wall,digits=1)) σ, rcw=$(round((2^(1/6)) +
        Ra,digits=2)) σw=$(round(((2^(1/6)) + Ra)*sigma_wall,digits=2)) σ",
        linestyle=:dash)
50 end
51
52 display(plot!())
53 # savefig("")
54
55 ccall(:jl_tty_set_mode, Int32, (Ptr{Cvoid}, Int32), stdin.handle, true)
56 read(stdin, 1)

```

参考文献