Effect of stiffness on phase separation

14.10.2025

Parameters

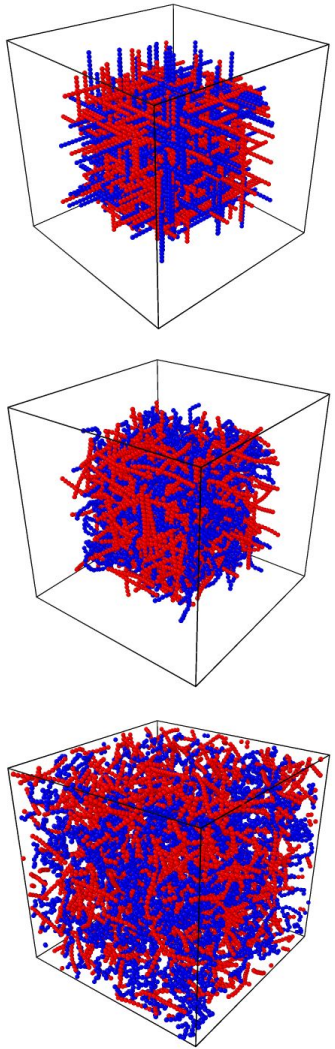
- Number of chains (N - type A, M - type B)
- Particle types (T1(head), T2(body), T3(tail) - type A, T4(head, T5(body), T6(tail) - type B)
- Angle coefficients (a1 - type A, a2 - type B)

Pair coefficient - fixed 1.0 repulsive
FENE

Simulation setup

- Initial configuration - each chain contains 15 beads
- Relaxation - 10^5 steps
- Equilibration - 10^5 steps

Data Analysis: Equilibration run
101 frames - 10^5 steps



Data analysis methods

- End-to end distance (R)
- Number density (ρ)

End-to-End Distance

Measures the straight-line distance between the first and last monomer of a polymer chain. Useful for understanding chain conformation and flexibility.

$$R_{ee} = \|\mathbf{r}_N - \mathbf{r}_1\| \tag{1}$$

where:

- R_{ee} is the end-to-end distance
- $\mathbf{r}_1$ is the position vector of the first monomer
- $\mathbf{r}_N$ is the position vector of the last monomer
- $\|\cdot\|$ denotes the Euclidean norm

1D Cartesian Number Density

Measures the actual concentration of particles along one coordinate axis. Gives the true number of particles per unit volume at each position.

$$\rho_\alpha(x) = \frac{N_\alpha(x)}{A \cdot \Delta x} \tag{5}$$

where:

- $\rho_\alpha(x)$ is the number density of type α at position x
- $N_\alpha(x)$ is the number of α particles in slab $[x, x + \Delta x]$
- $A = L_y \cdot L_z$ is the cross-sectional area perpendicular to x-axis
- Δx is the slab thickness ($\Delta x = \Delta y = \Delta z = 0.5\sigma$)

Data analysis methods

- Normalized probability density (p)
- Local composition fraction (ϕ)
- 2D density difference map (pB-pA)

1D Cartesian Probability Density

Tells you the likelihood of finding a particle at each position along an axis, properly normalized. Useful for statistical analysis and comparing distributions.

$$p_\alpha(x) = \frac{N_\alpha(x)}{N_{\text{total}} \cdot A \cdot \Delta x} \tag{6}$$

where:

- $p_\alpha(x)$ is the probability density of finding type α at position x
- N_{total} is the total number of α particles in the system

Local Composition Fraction

Tells you the local concentration or fraction of a specific type of particle in a small region of space. It's especially useful in multicomponent systems (like polymer blends, colloids, or mixtures) to study phase separation, layering, or clustering. The local composition fraction of component A is given by:

$$\phi_A(\mathbf{r}) = \frac{p_A(\mathbf{r})}{p_A(\mathbf{r}) + p_B(\mathbf{r})} \tag{7}$$

where:

- $\phi_A(\mathbf{r})$ is the local composition fraction of component A at position $\mathbf{r}$
- $p_A(\mathbf{r})$ is the probability density of A particles at position $\mathbf{r}$
- $p_B(\mathbf{r})$ is the probability density of B particles at position $\mathbf{r}$
- where $\mathbf{r}$ is x, y or z.

2D Probability Density Difference Map

Provides a normalized view of spatial segregation patterns, showing where each particle type is more likely to be found. Values range from -1 to 1 for easy interpretation.

$$\Delta p(x,y) = \frac{p_A(x,y) - p_B(x,y)}{\max|p_A(x,y) - p_B(x,y)|} \tag{13}$$

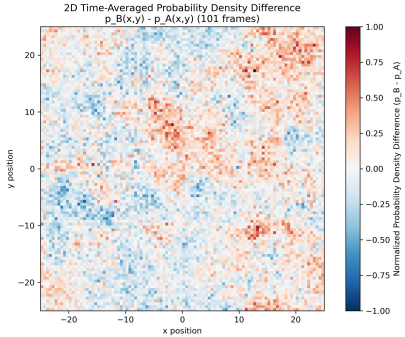
where:

- $p_A(x,y) = \dfrac{N_A(x,y)}{N_{\text{total},A} \cdot \Delta x \Delta y L_z}$ is the 2D probability density of type A
- $p_B(x,y) = \dfrac{N_B(x,y)}{N_{\text{total},B} \cdot \Delta x \Delta y L_z}$ is the 2D probability density of type B
- $\Delta p(x,y) \in [-1,1]$ indicates spatial segregation
- Positive values: regions dominated by type A
- Negative values: regions dominated by type B

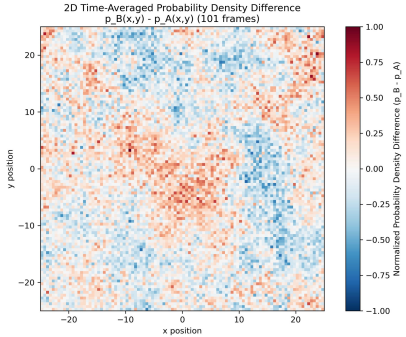
Effect of angle coefficient

N = 50, M = 50

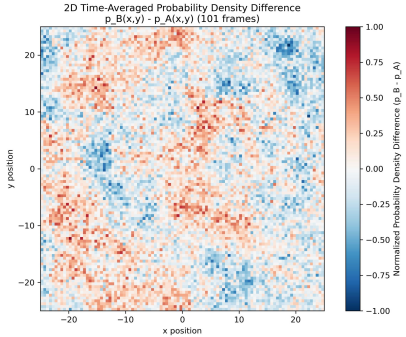
a1 = 1, a2 = 1



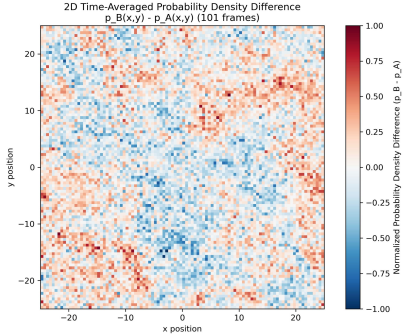
a1 = 1, a2 = 20



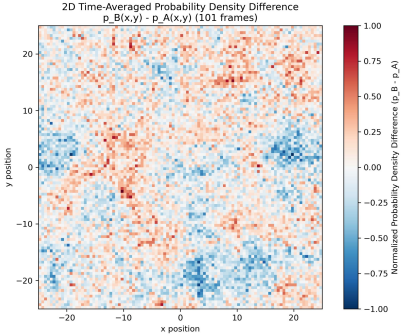
a1 = 1, a2 = 60



a1 = 20, a2 = 20



a1 = 20, a2 = 40



a1 = 20, a2 = 60

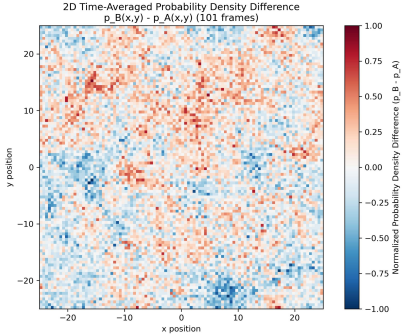
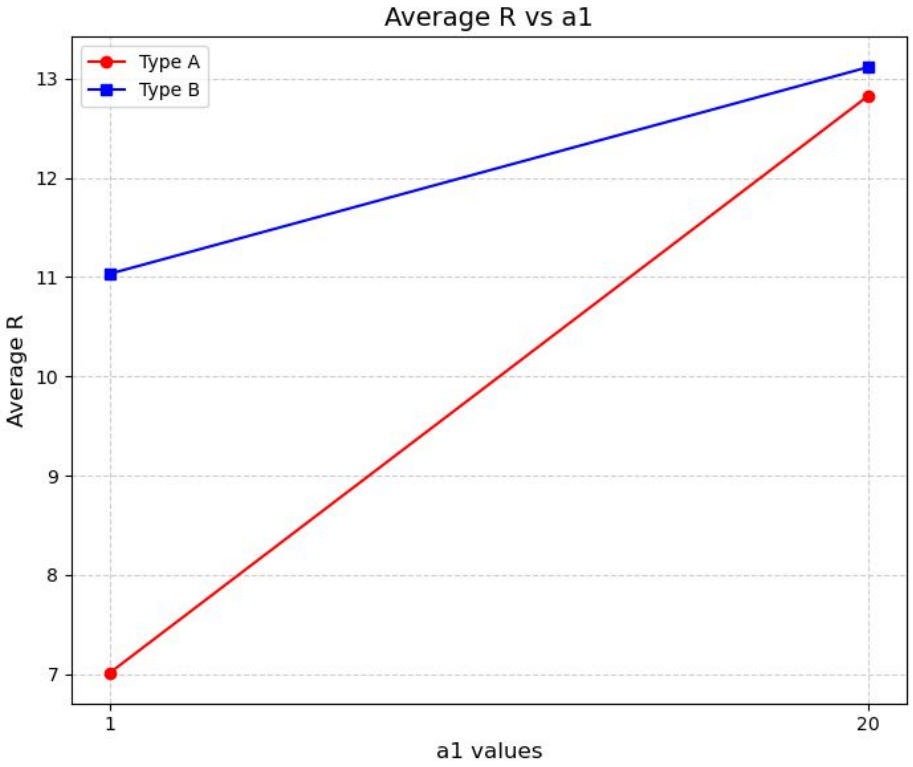
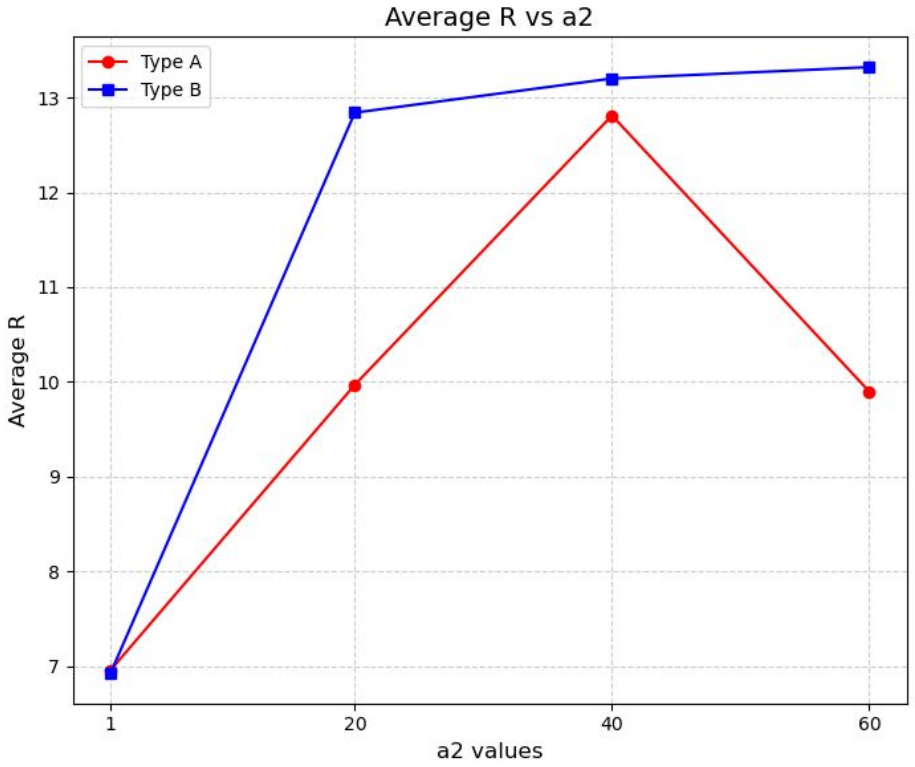


Table 1: End-to-end distance analysis for different angle coefficient combinations

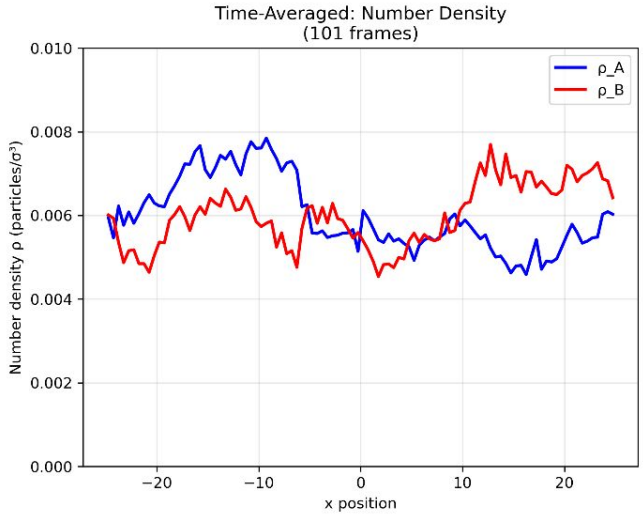
Angle Coefficients(a)		**End-to-End Distance(R)**	
a_1	a_2	$R_A(\sigma)$	$R_B(\sigma)$
1	1	6.904	6.861
1	20	7.081	12.845
1	60	6.965	13.315
20	20	12.839	12.826
20	40	12.810	13.196
20	60	12.827	13.315

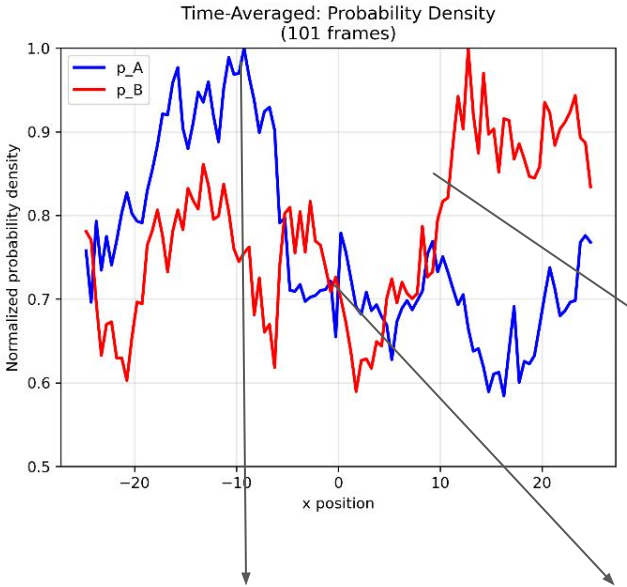
Note: a_1 and a_2 represent the angle coefficients for type A and type B polymer chains, respectively. R_A and R_B denote the time-averaged end-to-end distances.

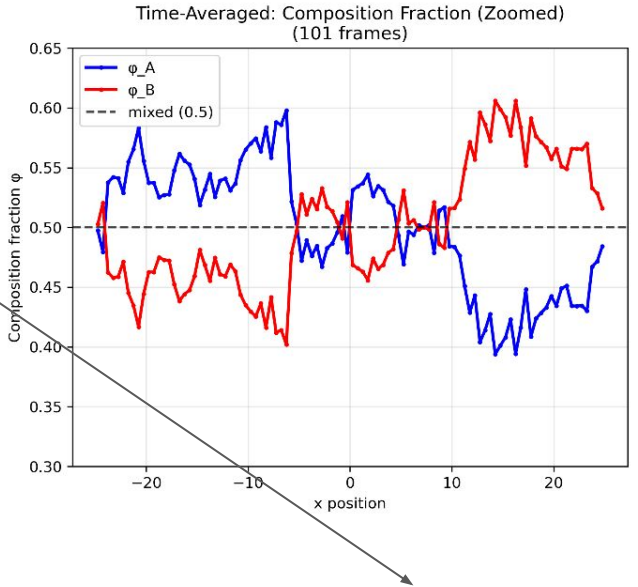

Average R vs a1
Type A
Type B
Average R
13
12
11
10
9
8
7
1
20
a1 values

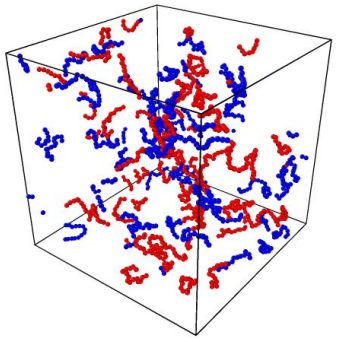

Average R vs a2
Type A
Type B
Average R
13
12
11
10
9
8
7
1
20
40
60
a2 values

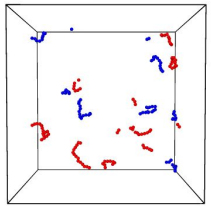
a1 = 1, a2 = 1

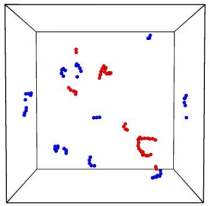

Time-Averaged: Number Density
(101 frames)
ρ_A
ρ_B
Number density ρ (particles/σ³)
x position

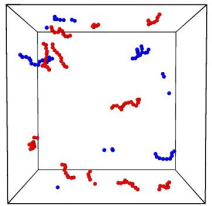

Time-Averaged: Probability Density
(101 frames)
p_A
p_B
Normalized probability density
x position

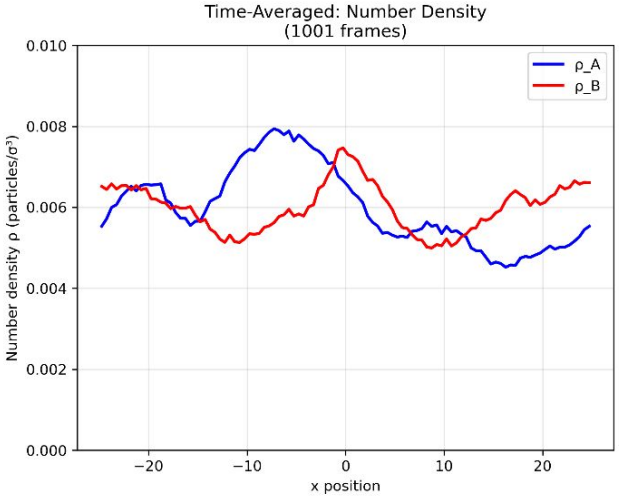

Time-Averaged: Composition Fraction (Zoomed)
(101 frames)
φ_A
φ_B
mixed (0.5)
Composition fraction φ
x position

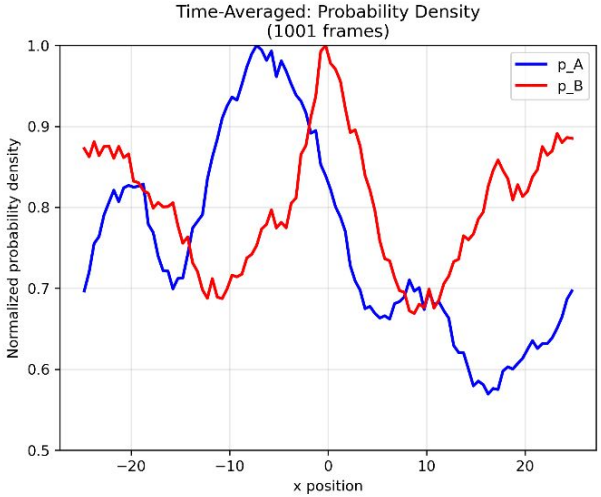


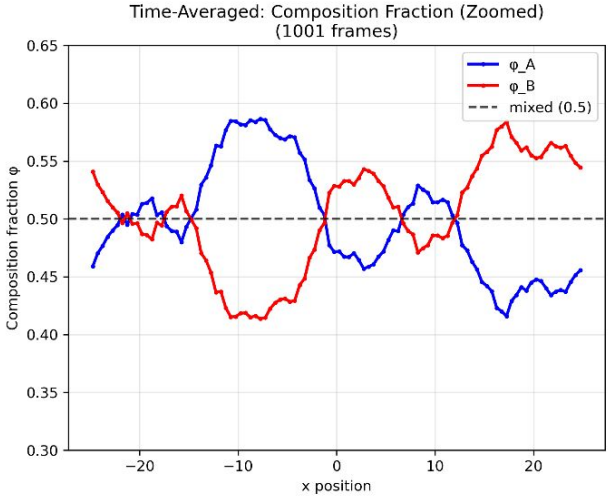




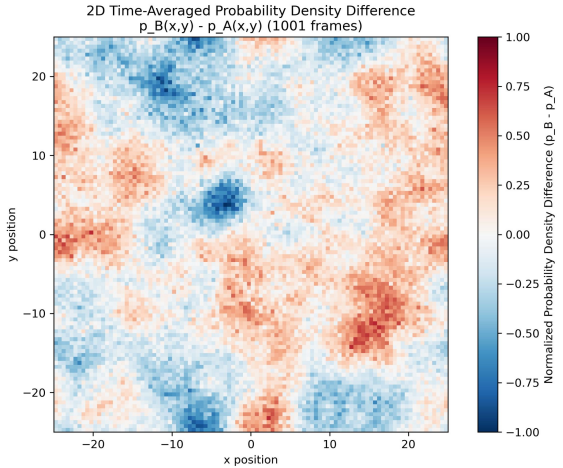




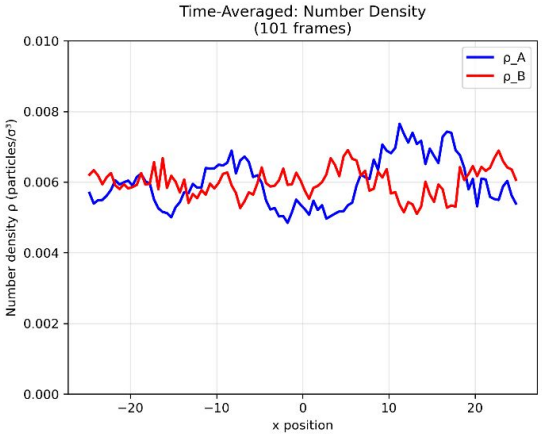


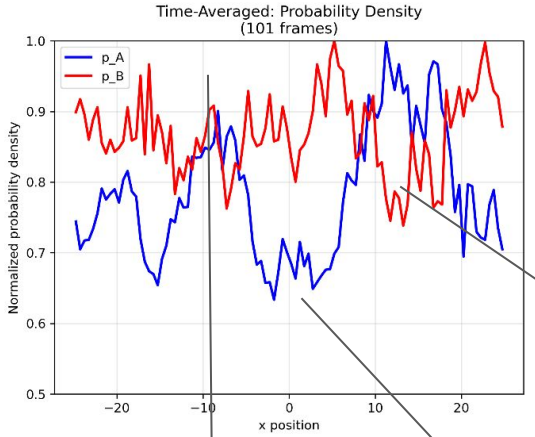


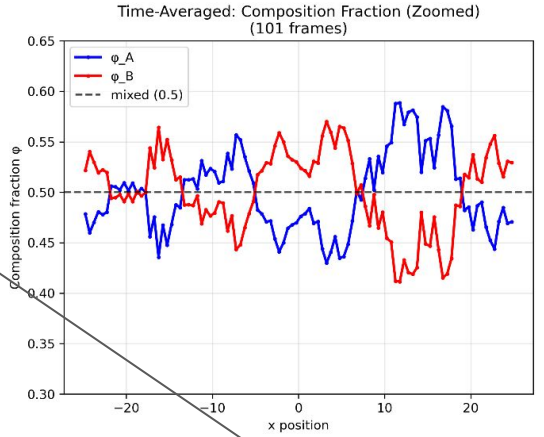
a1 = 1, a2 = 1
Thermostat ->100
Timesteps -> 10^6

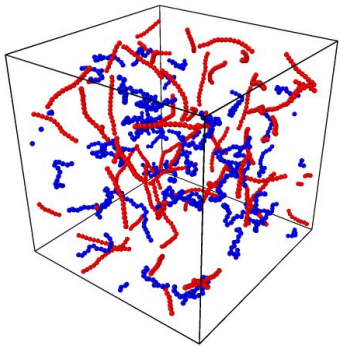


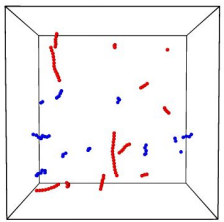
a1 = 1, a2 = 20

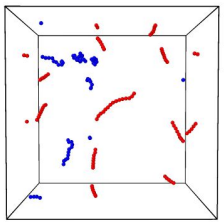

Time-Averaged: Number Density
(101 frames)
ρ_A
ρ_B
Number density ρ (particles/σ³)
x position

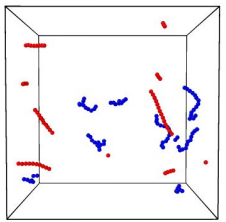

Time-Averaged: Probability Density
(101 frames)
p_A
p_B
Normalized probability density
x position


Time-Averaged: Composition Fraction (Zoomed)
(101 frames)
φ_A
φ_B
mixed (0.5)
Composition fraction φ
x position

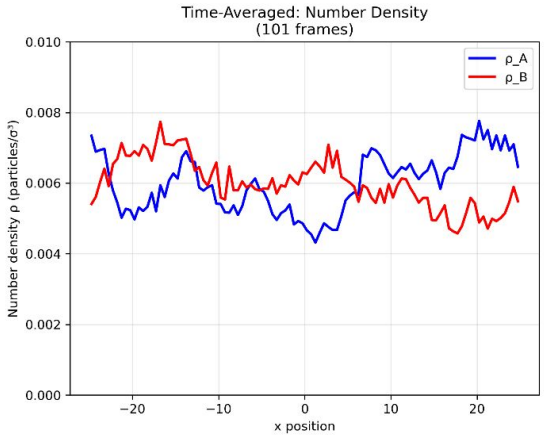


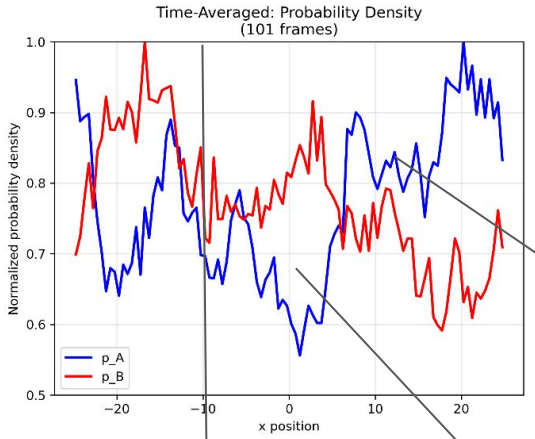


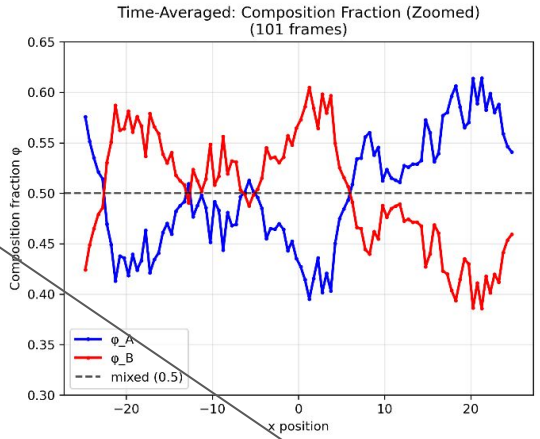


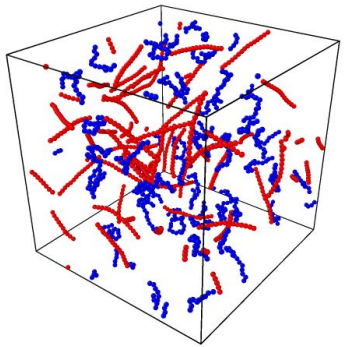


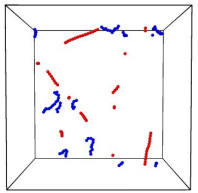
a1 = 1, a2 = 60

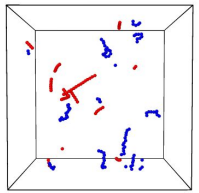

Time-Averaged: Number Density
(101 frames)
Number density ρ (particles/σ³)
x position
ρ_A
ρ_B

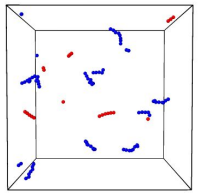

Time-Averaged: Probability Density
(101 frames)
Normalized probability density
x position
p_A
p_B


Time-Averaged: Composition Fraction (Zoomed)
(101 frames)
Composition fraction φ
x position
φ_A
φ_B
mixed (0.5)

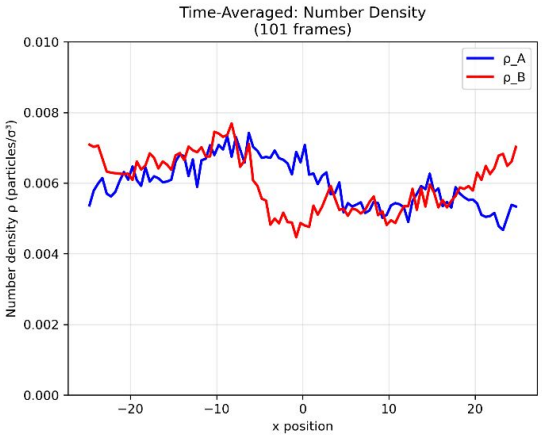


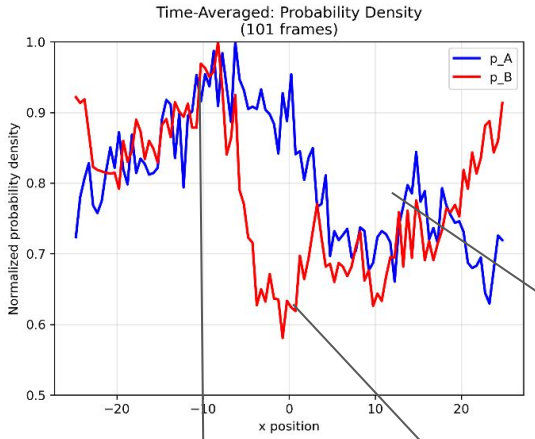


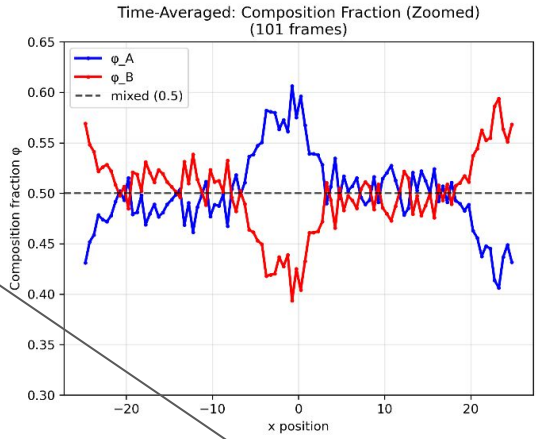


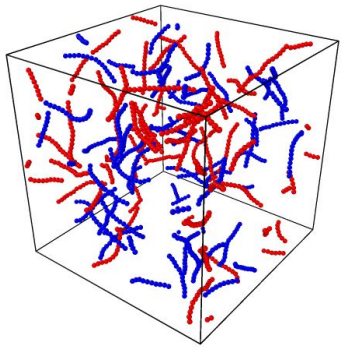


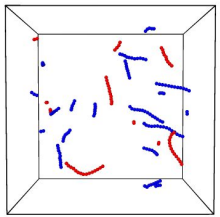
a1 = 20, a2 = 20

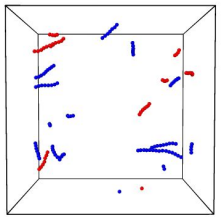

Time-Averaged: Number Density
(101 frames)
ρ_A
ρ_B
Number density ρ (particles/σ³)
x position

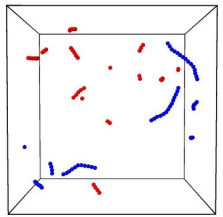

Time-Averaged: Probability Density
(101 frames)
p_A
p_B
Normalized probability density
x position


Time-Averaged: Composition Fraction (Zoomed)
(101 frames)
φ_A
φ_B
mixed (0.5)
Composition fraction φ
x position

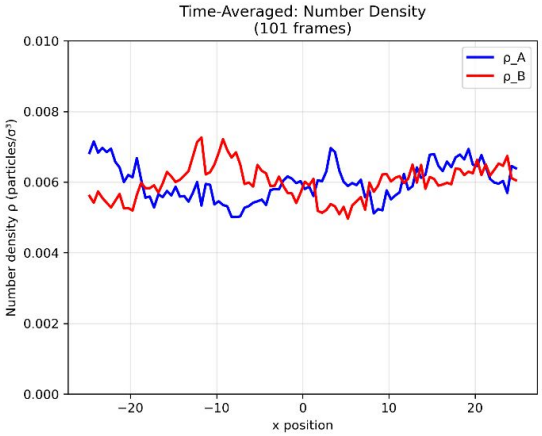


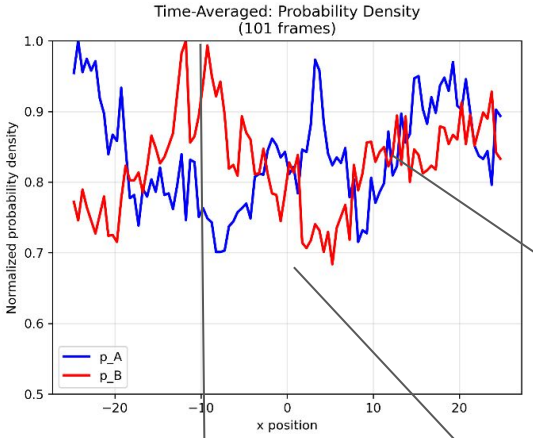


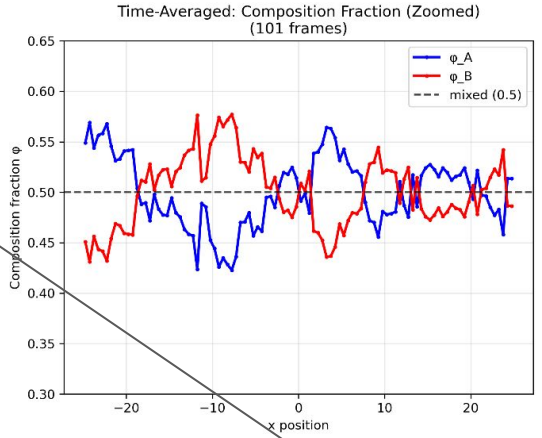


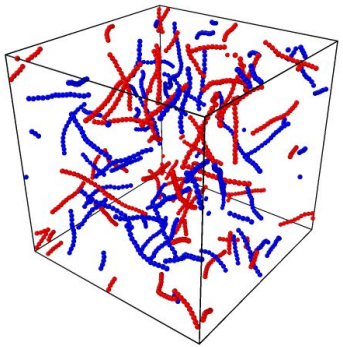


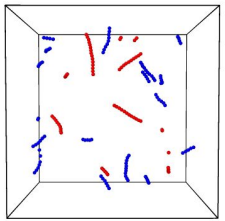
a1 = 20, a2 = 40

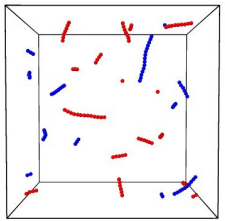

Time-Averaged: Number Density
(101 frames)
ρ_A
ρ_B
Number density ρ (particles/σ³)
x position

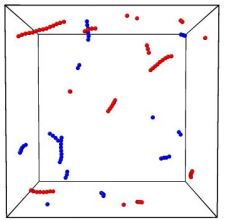

Time-Averaged: Probability Density
(101 frames)
ρ_A
ρ_B
Normalized probability density
x position


Time-Averaged: Composition Fraction (Zoomed)
(101 frames)
φ_A
φ_B
mixed (0.5)
Composition fraction φ
x position

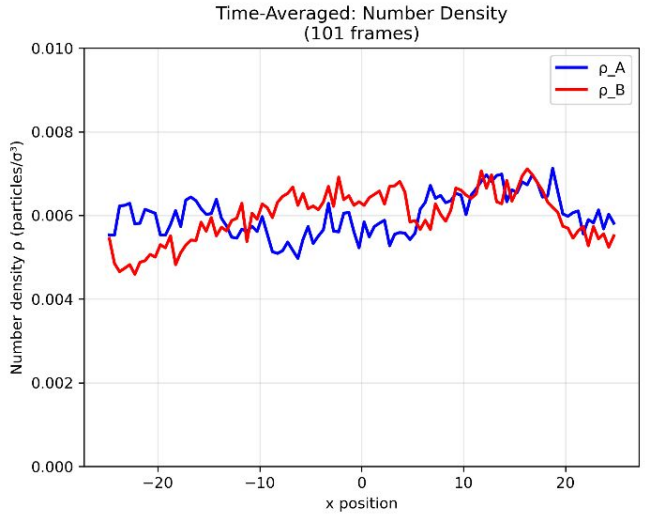


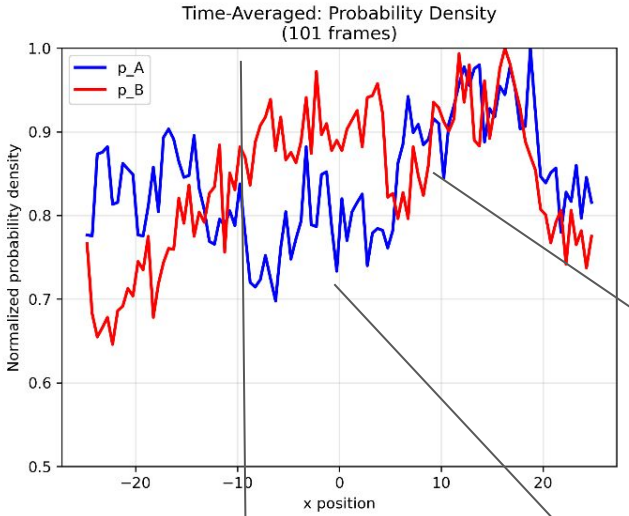


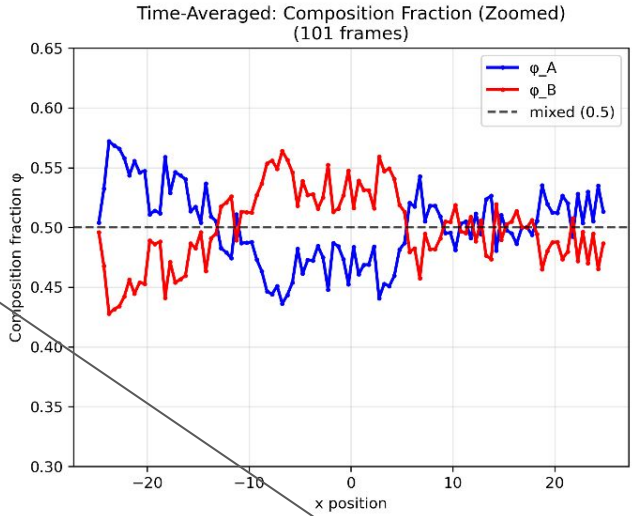


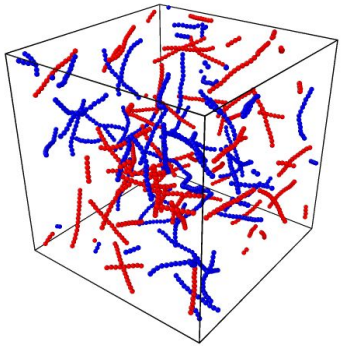


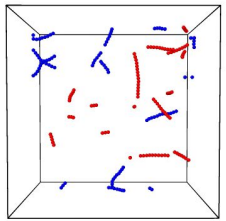
a1 = 20, a2 = 60

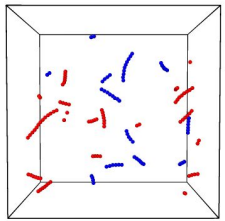

Time-Averaged: Number Density
(101 frames)
ρ_A
ρ_B
Number density ρ (particles/σ³)
x position

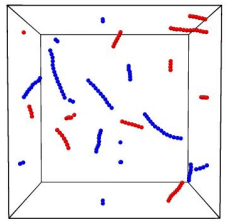

Time-Averaged: Probability Density
(101 frames)
p_A
p_B
Normalized probability density
x position

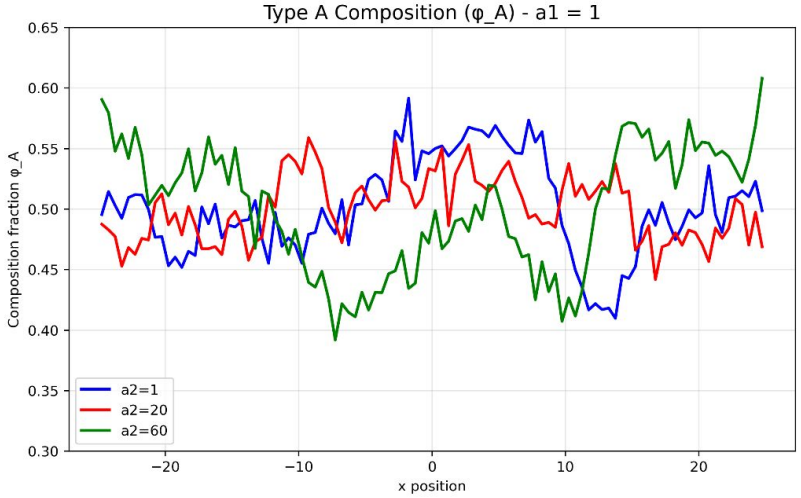

Time-Averaged: Composition Fraction (Zoomed)
(101 frames)
φ_A
φ_B
mixed (0.5)
Composition fraction φ
x position

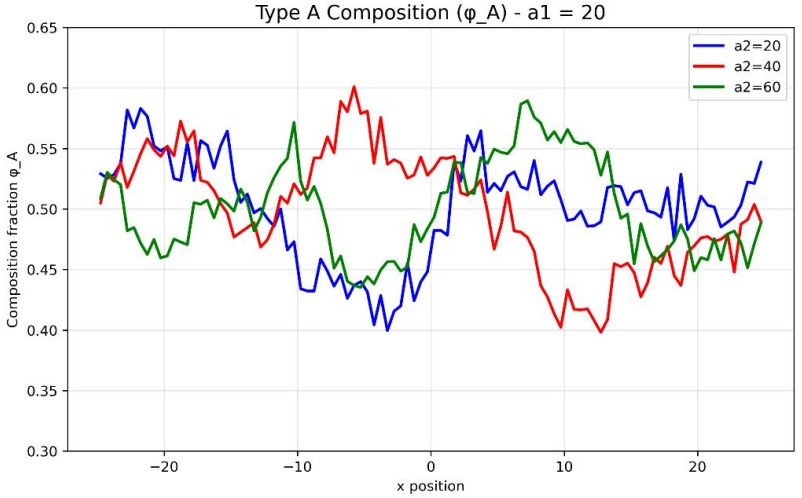


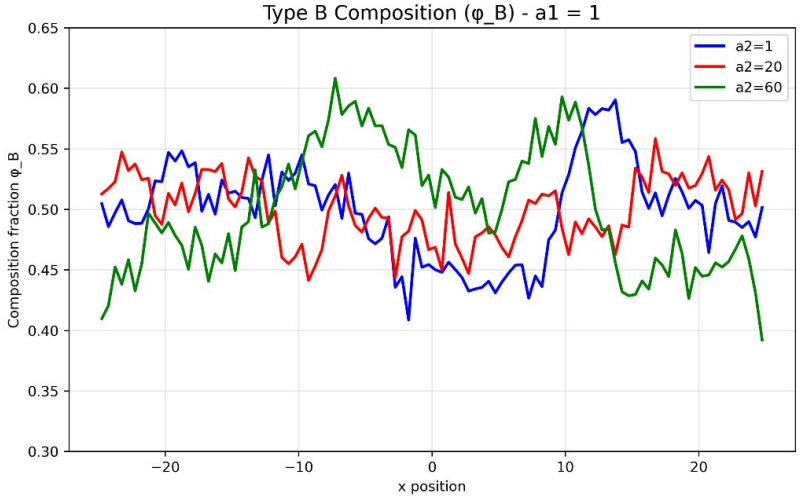


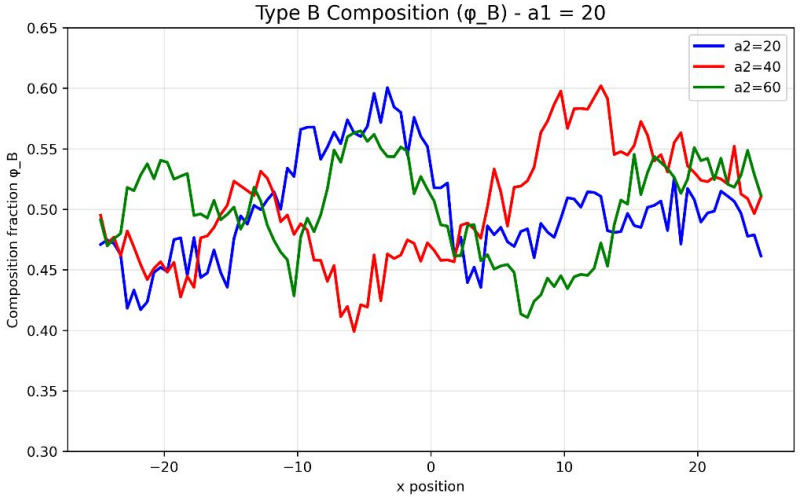





Type A Composition (φ_A) - a1 = 1
Composition fraction φ_A
x position
a2=1
a2=20
a2=60


Type A Composition (φ_A) - a1 = 20
Composition fraction φ_A
x position
a2=20
a2=40
a2=60

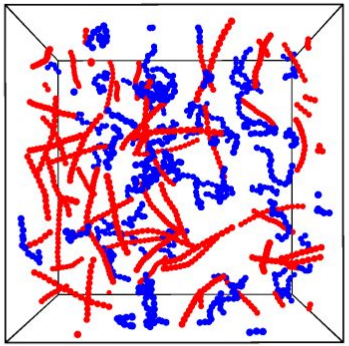

Type B Composition (φ_B) - a1 = 1
Composition fraction φ_B
x position
a2=1
a2=20
a2=60


Type B Composition (φ_B) - a1 = 20
Composition fraction φ_B
x position
a2=20
a2=40
a2=60

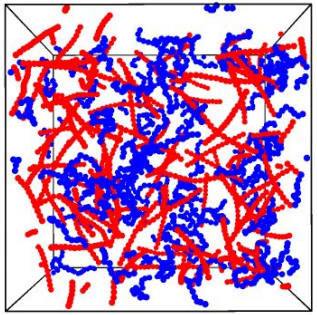
Effect of concentration
a1 = 1, a2 = 60

Comparison (a1 = 1, a2 = 60)

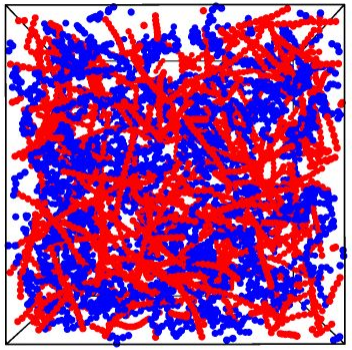
N + M = 100



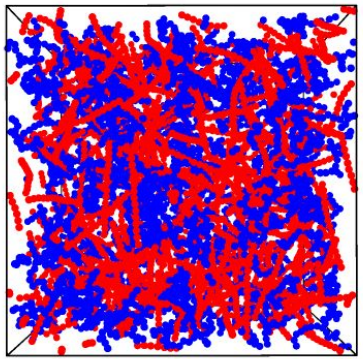
N + M = 200



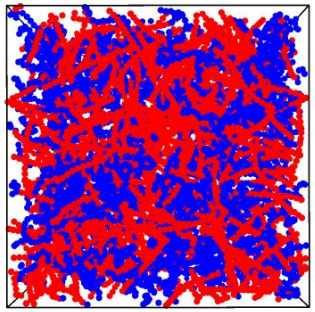
N + M = 500



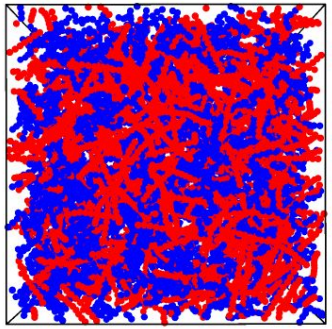
N + M = 600



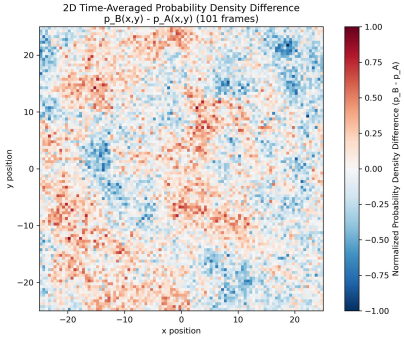
N + M = 833



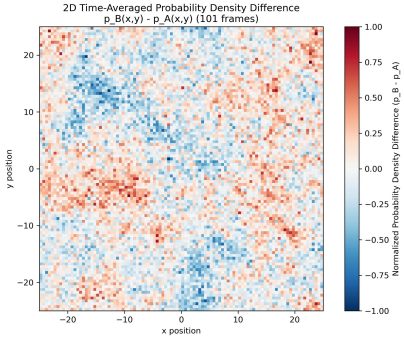
N + M = 1000



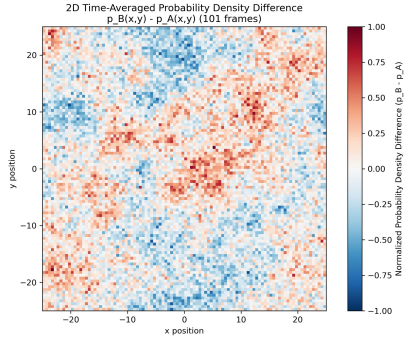
N + M = 100



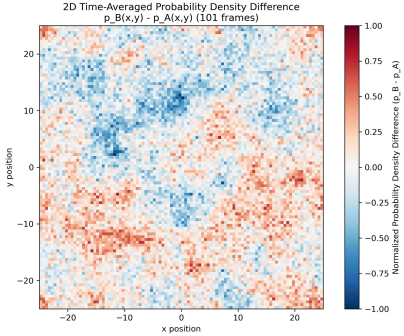
N + M = 200



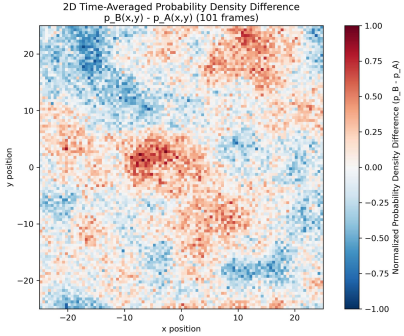
N + M = 500



N + M = 600



N + M = 833



N + M = 1000

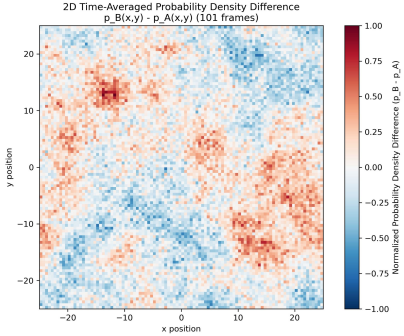
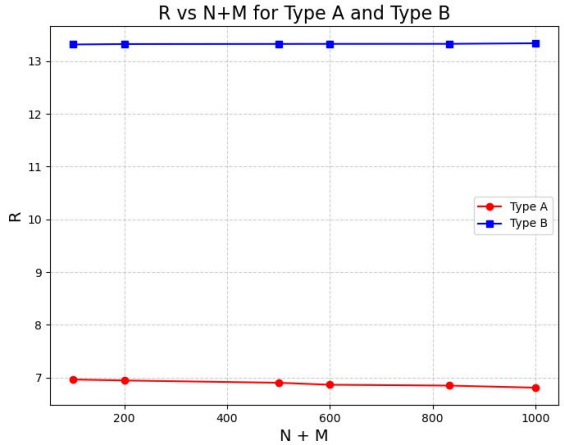


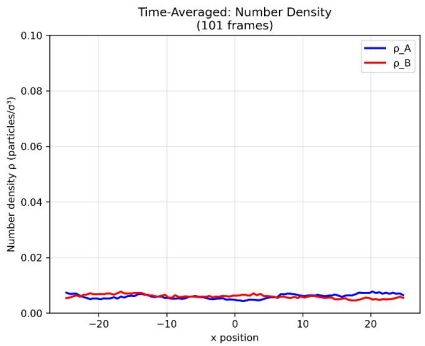
Table 2: End-to-end distance analysis for different concentrations

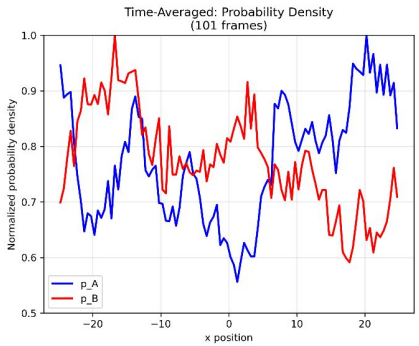
Angle Coefficients(a)		**End-to-End Distance(R)**	
N	M	$R_A(\sigma)$	$R_B(\sigma)$
50	50	6.965	13.315
100	100	6.946	13.323
250	250	6.903	13.325
300	300	6.865	13.326
416	417	6.849	13.327
500	500	6.810	13.338

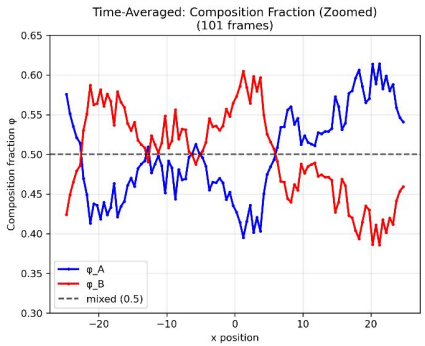
Note: N and M represent the chain numbers for type A and type B polymer chains, respectively. R_A and R_B denote the time-averaged end-to-end distances.



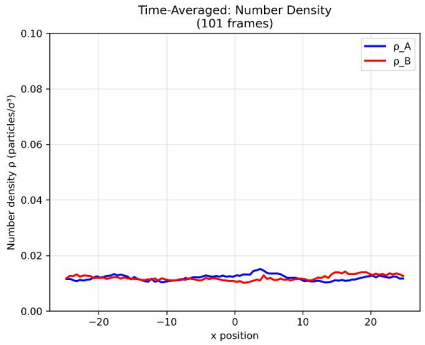
N = 50
M = 50
x

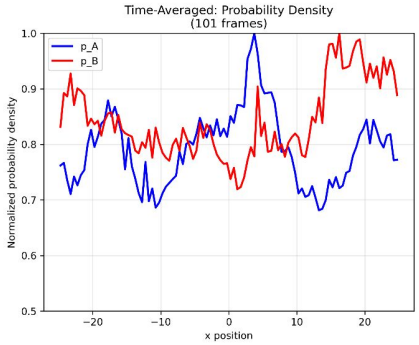


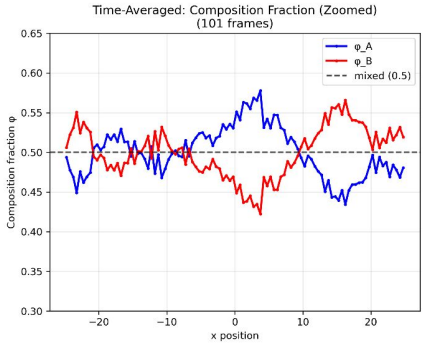




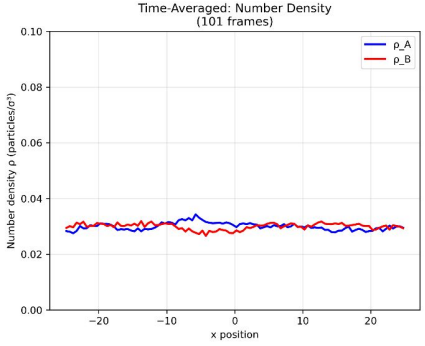
N = 100
M = 100
x

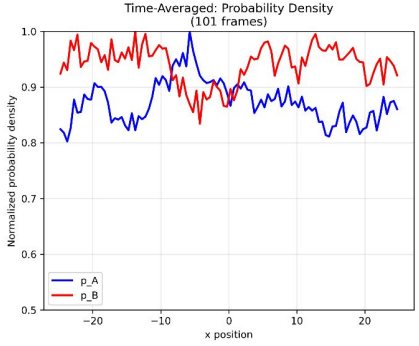


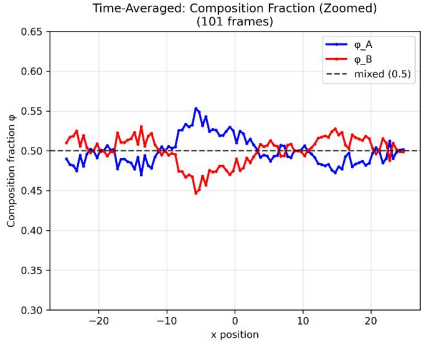




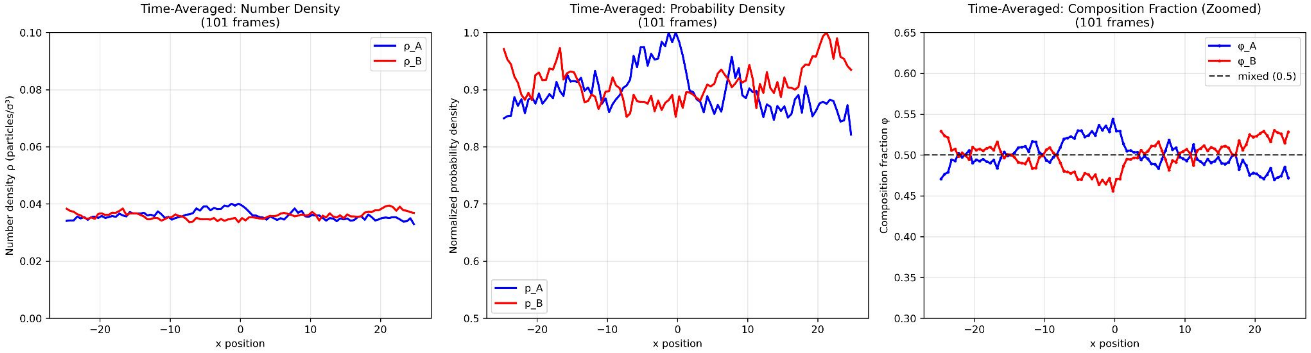
N = 250
M = 250
x



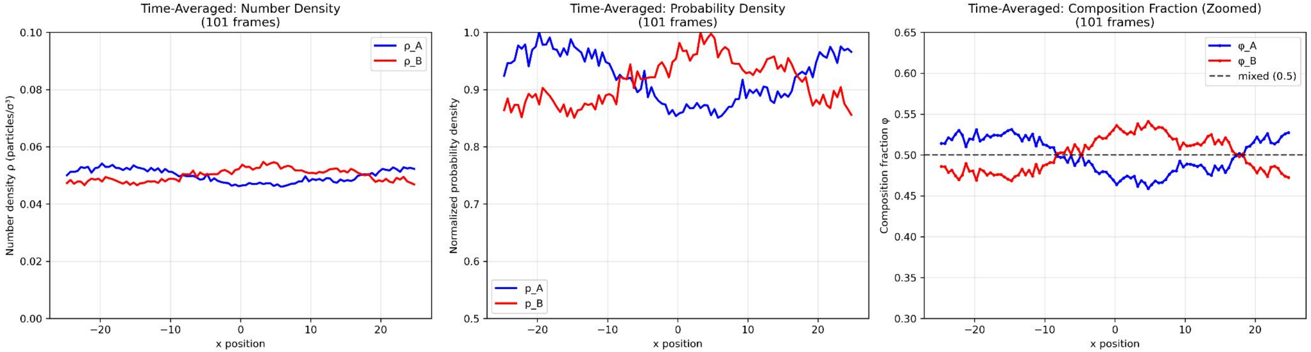




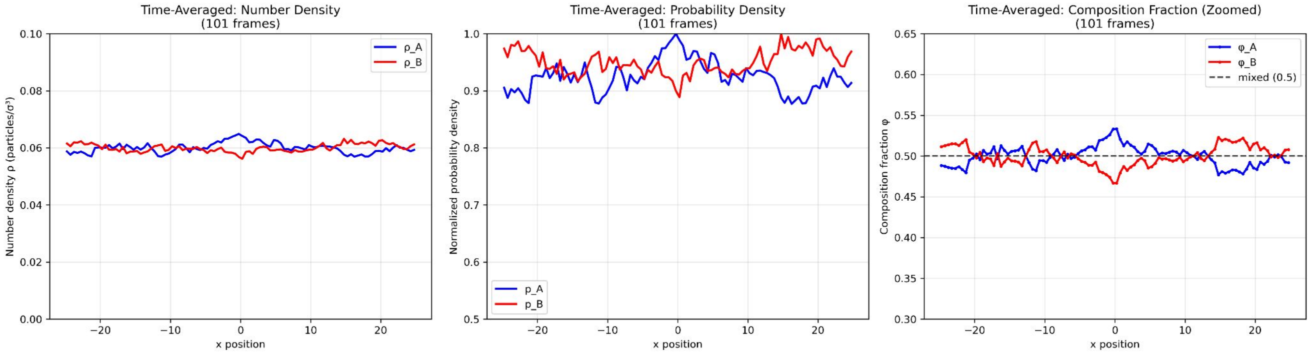
N = 300
M = 300
x

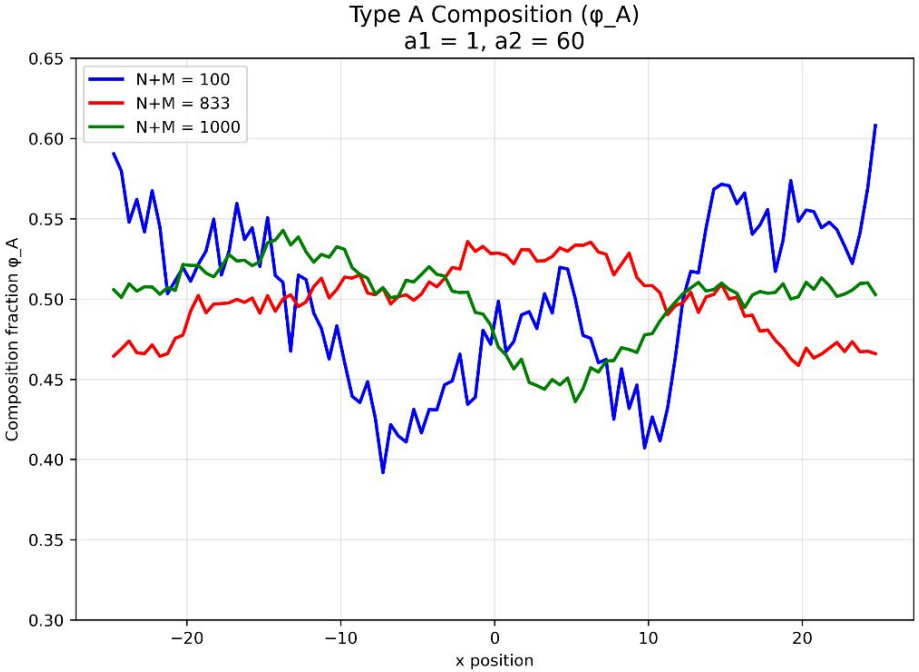


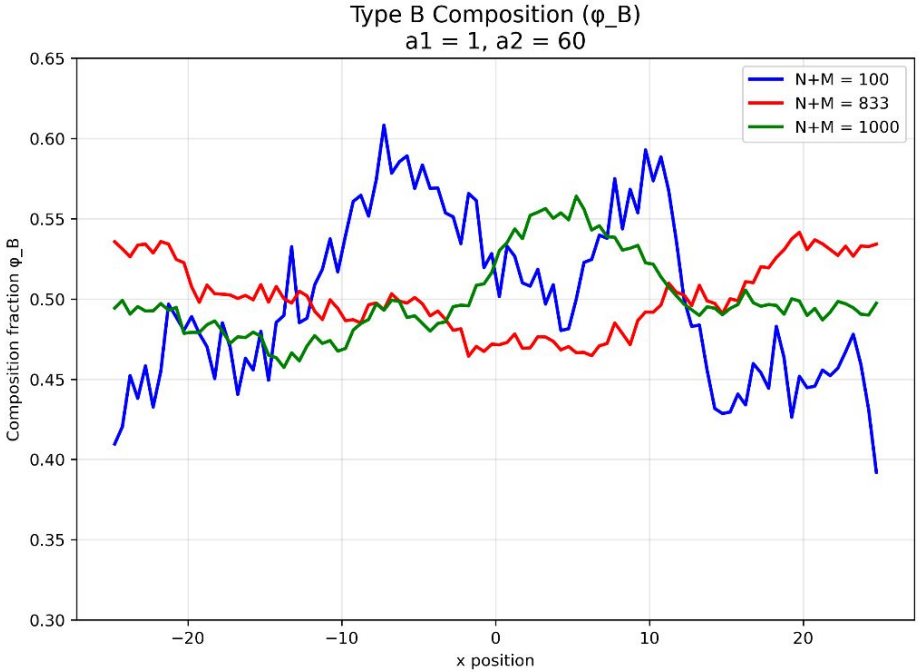
N = 417
M = 416
x



N = 500
M = 500
x




Type A Composition (φ_A)
a1 = 1, a2 = 60
N+M = 100
N+M = 833
N+M = 1000
Composition fraction φ_A
x position
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
−20
−10
0
10
20


Type B Composition (φ_B)
a1 = 1, a2 = 60
N+M = 100
N+M = 833
N+M = 1000
Composition fraction φ_B
x position
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
−20
−10
0
10
20

Analysis

```python
# Slab profile functions with 0.5σ bin size
def slab_profile_number_density(coords, mask, axis=0, R=25.0):
    """Formula 1: 1D Number Density ρ_α(x) = N_α(x) / (A * Δx)"""
    dx = 0.5  # σ = 0.5
    bins = np.arange(-R, R + dx, dx)

    # Cross-sectional area
    area = (2 * R) * (2 * R)

    vals = coords[:, axis]
    cnt, _ = np.histogram(vals[mask], bins=bins)
    density = cnt / (area * dx)  # Number density

    centers = 0.5 * (bins[:-1] + bins[1:])
    return centers, density

def slab_profile_probability_density(coords, mask, total_particles, axis=0, R=25.0):
    """Formula 2: 1D Probability Density p_α(x) = N_α(x) / (N_total,α * A * Δx)"""
    dx = 0.5  # σ = 0.5
    bins = np.arange(-R, R + dx, dx)

    # Cross-sectional area
    area = (2 * R) * (2 * R)

    vals = coords[:, axis]
    cnt, _ = np.histogram(vals[mask], bins=bins)
    probability_density = cnt / (total_particles * area * dx)  # Probability density

    centers = 0.5 * (bins[:-1] + bins[1:])
    return centers, probability_density

def calculate_composition_fraction_probability(p_A, p_B):
    """Calculate composition fraction using probability density: φ_A = p_A / (p_A + p_B)"""
    phi_A = np.zeros_like(p_A)
    phi_B = np.zeros_like(p_B)
    nonzero = (p_A + p_B) > 0
    phi_A[nonzero] = p_A[nonzero] / (p_A[nonzero] + p_B[nonzero])
    phi_B[nonzero] = p_B[nonzero] / (p_A[nonzero] + p_B[nonzero])
    return phi_A, phi_B
```

Simulation input file

Relaxation and repulsive run

```
# --- Initialization ---
units       lj
boundary    p p p

atom_style  angle
pair_style  lj/cut 1.12246
pair_modify shift yes
bond_style fene
angle_style harmonic

# --- Read structure ---
read_data   solution.data

# --- Bonded Interactions ---
bond_coeff 1 30.0 1.5 1.0 1.0
bond_coeff 2 30.0 1.5 1.0 1.0
angle_coeff 1 1.0 180.0
angle_coeff 2 60.0 180.0

pair_coeff * * 1.0 1.0

# --- Energy Minimization ---
minimize 1e-6 1e-8 10000 100000

# --- Neighbor setup ---
neighbor     0.4 bin
neigh_modify delay 10

# --- Velocity Initialization & Dynamics ---
velocity all create 1.0 12345 rot yes dist gaussian

fix integrator all nve
fix thermostat all langevin 1.0 1.0 100.0 12345

# --- Output settings ---
thermo_style multi
thermo 500

# --- Dump unwrapped coordinates for python analysis ---
dump 1 all atom 1000 dump.relaxation
dump_modify 1 scale yes

# Run simulation
run 100000

# --- Save final configuration ---
write_data data.relaxation
```

in.task_1_60 ●

2 types > _Task1000 > 1_60 > in.task_1_60

```
# --- Initialization ---
units       lj
boundary    p p p

atom_style  angle
pair_style  lj/cut 1.12246
pair_modify shift yes
bond_style fene
angle_style harmonic

# --- Read structure ---
read_data   data.relaxation

# --- Bonded Interactions ---
bond_coeff 1 30.0 1.5 1.0 1.0
bond_coeff 2 30.0 1.5 1.0 1.0
angle_coeff 1 1.0 180.0
#angle_coeff 2 60.0 180.0

pair_coeff * * 1.0 1.0

# --- Neighbor setup ---
neigh_modify delay 10

# --- Velocity Initialization & Dynamics ---
velocity all create 1.0 12345 rot yes dist gaussian

fix integrator all nve
fix thermostat all langevin 1.0 1.0 100.0 12345

# --- Output settings ---
thermo_style multi
thermo 500

# --- Dump unwrapped coordinates for python analysis ---
dump 1 all atom 1000 dump.task
dump_modify 1 scale yes

# Run simulation
run 100000

# --- Save final configuration ---
write_data data.task



print "Simulation done"
```

1D Cartesian Probability Density

Tells you the likelihood of finding a particle at each position along an axis, properly normalized. Useful for statistical analysis and comparing distributions.

$$p_\alpha(x) = \frac{N_\alpha(x)}{N_{\text{total},\alpha} \cdot A \cdot \Delta x} \tag{6}$$

where:

- $p_\alpha(x)$ is the probability density of finding type α at position x
- $N_{\text{total},\alpha}$ is the total number of α particles in the system

```
N_total_A = len(coords_transformed[A_mask])
N_total_B = len(coords_transformed[B_mask])

for axis_idx, axis_name in enumerate(['x', 'y', 'z']):
    # Calculate instantaneous profiles
    centers, rho_A_inst = slab_profile_number_density(coords_transformed, A_mask, axis=axis_id
    centers, rho_B_inst = slab_profile_number_density(coords_transformed, B_mask, axis=axis_id
    centers, p_A_inst = slab_profile_probability_density(coords_transformed, A_mask, N_total_A
    centers, p_B_inst = slab_profile_probability_density(coords_transformed, B_mask, N_total_B
```

Local Composition Fraction

Tells you the local concentration or fraction of a specific type of particle in a small region of space. It's especially useful in multicomponent systems (like polymer blends, colloids, or mixtures) to study phase separation, layering, or clustering. The local composition fraction of component A is given by:

$$\phi_A(\mathbf{r}) = \frac{\rho_A(\mathbf{r})}{\rho_A(\mathbf{r}) + \rho_B(\mathbf{r})} \tag{7}$$

where:

- $\phi_A(\mathbf{r})$ is the local composition fraction of component A at position $\mathbf{r}$
- $\rho_A(\mathbf{r})$ is the number density of A particles at position $\mathbf{r}$
- $\rho_B(\mathbf{r})$ is the number density of B particles at position $\mathbf{r}$
- where $\mathbf{r}$ is x, y or z.

```
def calculate_composition_fraction(rho_A, rho_B):
    """Calculate composition fraction: φ_A = ρ_A / (ρ_A + ρ_B)"""
    phi_A = np.zeros_like(rho_A)
    phi_B = np.zeros_like(rho_B)
    nonzero = (rho_A + rho_B) > 0
    phi_A[nonzero] = rho_A[nonzero] / (rho_A[nonzero] + rho_B[nonzero])
    phi_B[nonzero] = rho_B[nonzero] / (rho_A[nonzero] + rho_B[nonzero])
    return phi_A, phi_B
```

Proper fixes

1D Cartesian Probability Density

Tells you the likelihood of finding a particle at each position along an axis, properly normalized. Useful for statistical analysis and comparing distributions.

$$p_\alpha(x) = \frac{N_\alpha(x)}{N_{\text{total}} \cdot A \cdot \Delta x} \tag{6}$$

where:

- $p_\alpha(x)$ is the probability density of finding type α at position x
- N_{total} is the total number of α particles in the system

```
coords_transformed = (coords - 0.5) * 2 * R
A_mask = np.isin(types, [1, 2, 3])
B_mask = ~A_mask

N_total = len(coords_transformed[A_mask]) + len(coords_transformed[B_mask])

for axis_idx, axis_name in enumerate(['x', 'y', 'z']):
    # Calculate instantaneous profiles
    centers, rho_A_inst = slab_profile_number_density(coords_transformed, A_mask, axis=axis_i
    centers, rho_B_inst = slab_profile_number_density(coords_transformed, B_mask, axis=axis_i
    centers, p_A_inst = slab_profile_probability_density(coords_transformed, A_mask, N_total,
    centers, p_B_inst = slab_profile_probability_density(coords_transformed, B_mask, N_total,
```

Local Composition Fraction

Tells you the local concentration or fraction of a specific type of particle in a small region of space. It's especially useful in multicomponent systems (like polymer blends, colloids, or mixtures) to study phase separation, layering, or clustering. The local composition fraction of component A is given by:

$$\phi_A(\mathbf{r}) = \frac{p_A(\mathbf{r})}{p_A(\mathbf{r}) + p_B(\mathbf{r})} \tag{7}$$

where:

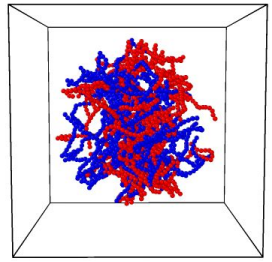
- $\phi_A(\mathbf{r})$ is the local composition fraction of component A at position $\mathbf{r}$
- $p_A(\mathbf{r})$ is the probability density of A particles at position $\mathbf{r}$
- $p_B(\mathbf{r})$ is the probability density of B particles at position $\mathbf{r}$
- where $\mathbf{r}$ is x, y or z.

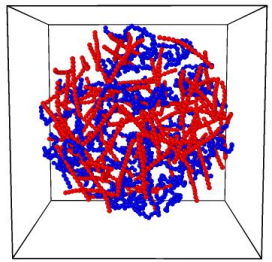
```
calculate_composition_fraction_probability(p_A, p_B):
"""Calculate composition fraction using probability density: φ_A = p_A / (p_A + p_B)"""
phi_A = np.zeros_like(p_A)
phi_B = np.zeros_like(p_B)
nonzero = (p_A + p_B) > 0
phi_A[nonzero] = p_A[nonzero] / (p_A[nonzero] + p_B[nonzero])
phi_B[nonzero] = p_B[nonzero] / (p_A[nonzero] + p_B[nonzero])
return phi_A, phi_B
```

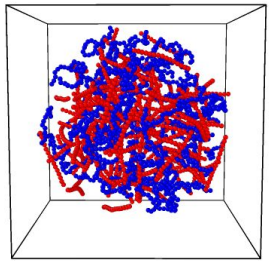
Simulation setup

- Initial configuration - each chain contains 15 beads
- Relaxation - 10^6 steps
- Equilibration - 10^6 steps

Data Analysis: Equilibration run
101 frames - 10^6 steps







Data analysis methods

- Radial Distribution Function(g(r))
- Radial probability density (p)
- Local composition fraction (ϕ)

Radial Probability Density

*I should have used this formula, however I used normalized radial number density. The radial probability density is given by:

$$p(r) = \frac{n(r)}{4\pi R_0^2 \Delta r}. \tag{3}$$

where:

- $p(r)$ is the radial probability density at distance r
- $n(r)$ is the number of particles in the spherical shell at radius r
- R_0 is the reference radius
- Δr is R_0 - r
- r is the position

Radial Distribution Function

The radial distribution function answers: "Given a particle here, how likely am I to find another particle at distance r from it compared to a completely random distribution (ideal gas distribution)?"

$$g(r) = \frac{1}{\rho N} \left\langle \sum_{i=1}^{N} \sum_{j \neq i}^{N} \delta(r - r_{ij}) \right\rangle \tag{2}$$

where:

- N = total number of particles
- $\rho = \frac{N}{V}$ = number density (particles per unit volume)
- r_{ij} = distance between particles i and j
- δ = Dirac delta function
- $\langle \cdots \rangle$ = ensemble or time average

Local Composition Fraction

Tells you the local concentration or fraction of a specific type of particle in a small region of space. It's especially useful in multicomponent systems (like polymer blends, colloids, or mixtures) to study phase separation, layering, or clustering. The local composition fraction of component A is given by:

$$\phi_A(\mathbf{r}) = \frac{p_A(\mathbf{r})}{p_A(\mathbf{r}) + p_B(\mathbf{r})} \tag{7}$$

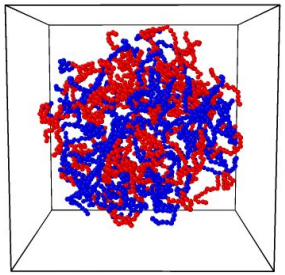
where:

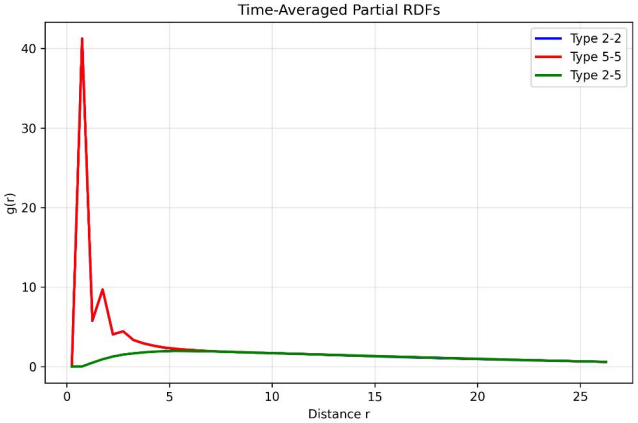
- $\phi_A(\mathbf{r})$ is the local composition fraction of component A at position $\mathbf{r}$
- $p_A(\mathbf{r})$ is the probability density of A particles at position $\mathbf{r}$
- $p_B(\mathbf{r})$ is the probability density of B particles at position $\mathbf{r}$
- where $\mathbf{r}$ is x, y or z.

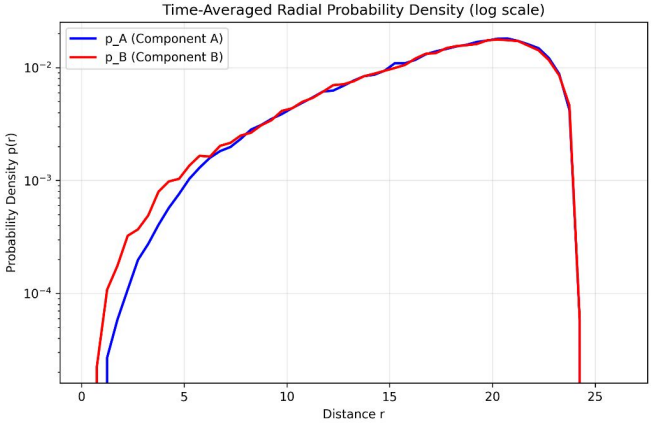
Effect of angle coefficient

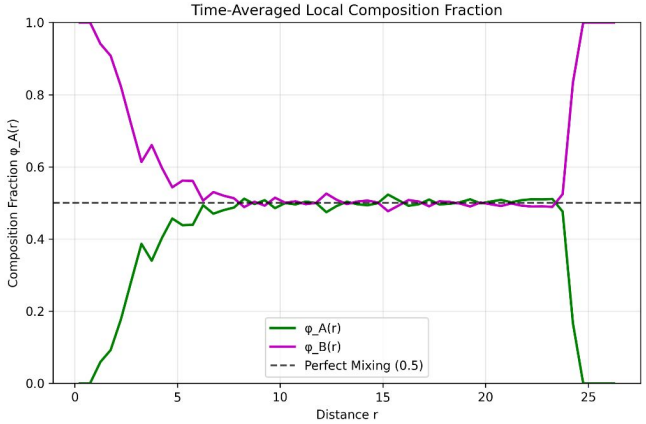
N = 100, M = 100

a1 = 1, a2 = 1



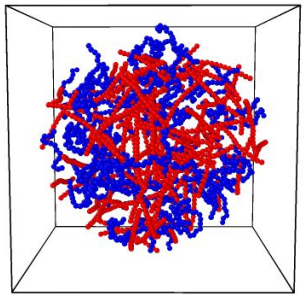


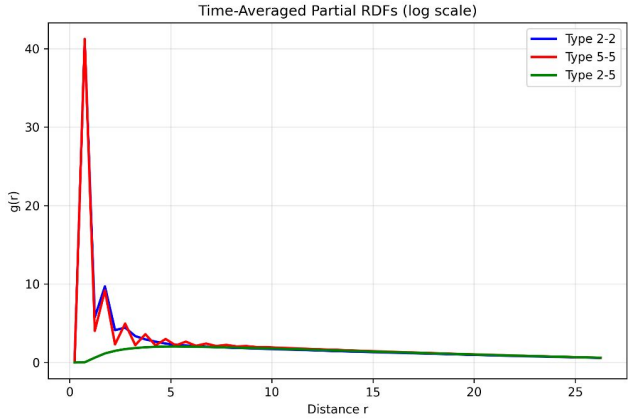


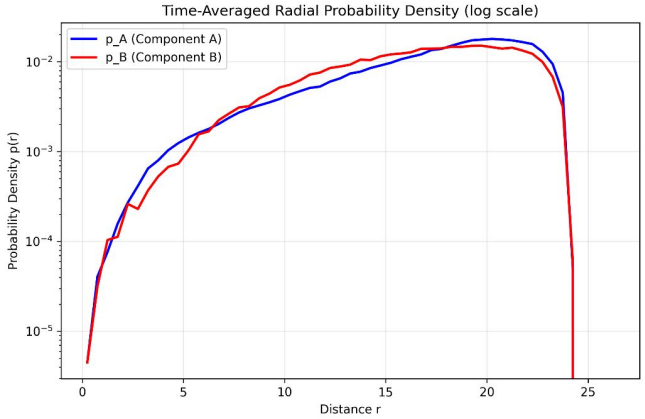


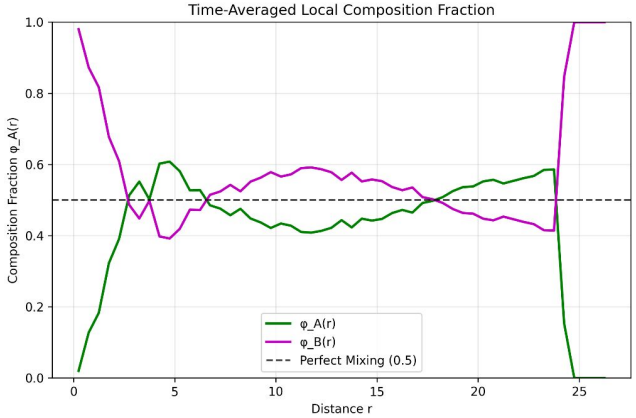
System Summary:
- Total particles: 3000
- Component A types: 1, 2, 3
- Component B types: 4, 5, 6
- Sphere radius: 26.33
- Box size: 50.00
- Frames averaged: 51
- Using second half of trajectory

a1 = 1, a2 = 60









System Summary:
- Total particles: 3000
- Component A types: 1, 2, 3
- Component B types: 4, 5, 6
- Sphere radius: 26.32
- Box size: 50.00
- Frames averaged: 51
- Using second half of trajectory