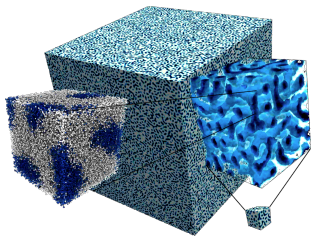
# SOMA: Scaling Soft Polymers to many GPUs via OpenACC

With Applications to Battery Electrolytes

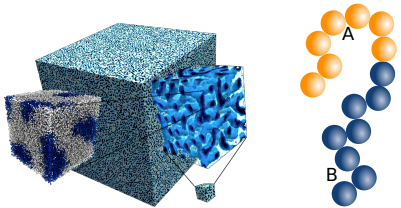
Institute for Theoretical Physics, Georg-August University Göttingen

February 11, 2020



# Why are complex polymer melts interesting?

Self-assembled nanostructures




Appealing HPC object to study

SOMA
yes, soft matters

L. Schneider and M. Müller, Comput. Phys. Commun. **235C**, 463–476 (2019)

## Applications

- battery materials
- molecular sieves
- micro electronics

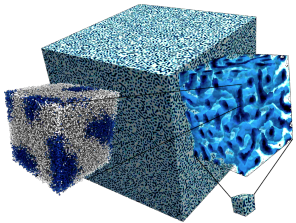
## HPC implementation

- polymers are fractal objects
- straightforward coarse-graining
- short-range vs. long-range phenomena

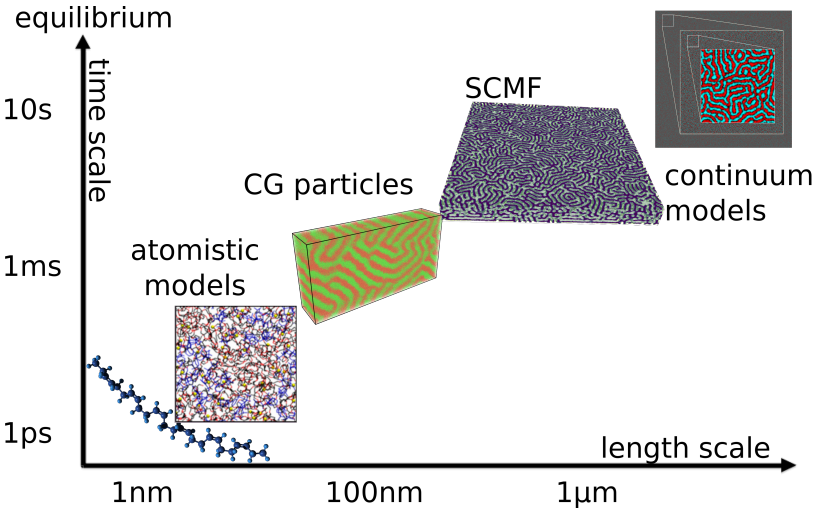
# Overview

1. How to model and simulate engineering scale polymer melts?
   - Why OpenACC for GPUs?
   - Slow, weak interactions via fluctuating fields
   - Parallel hierarchy for strong interactions
   - Detours: implementation quirks
2. Large-scale metastable networks: battery electrolytes
   - Goals for electrolyte materials
   - Percolating network structures
   - Diffusive transport properties
3. Next steps: modular computing
   - Dedicated analysis server

SOMA
yes, soft matters



# Hierarchy of coarse-grained models

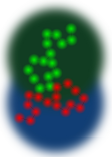



- omission of atomistic details
- large length scales
- long time scales
- DPD enables hydrodynamics
- SCMF unlocks engineering scales
- continuum models would allow full product simulation

# Coarse-graining: single bead $\leftrightarrow$ many atoms

## Coarse-graining

- Gaussian chains
- fewer degrees of freedom
- universality
- higher parallelism
- soft interactions



density $\hat{\phi}_A, \hat{\phi}_B$ based Monte-Carlo

- harmonic bond potential: $R_{e0}$
  - $V_h(\boldsymbol{r}) = \frac{k_2}{2}\boldsymbol{r}^2$
- restrain density fluctuations: $\kappa_0 N$
  - $\mathscr{H}_{\text{fluc.}}[\hat{\phi}_A, \hat{\phi}_B] \propto \int \mathrm{d}\boldsymbol{r}\ \frac{\kappa_0 N}{2}\left(\hat{\phi}_A(\boldsymbol{r}) + \hat{\phi}_B(\boldsymbol{r}) - 1\right)^2$
- microphase separation: $\chi_0 N$
  - $\mathscr{H}_{\text{sep.}}[\hat{\phi}_A, \hat{\phi}_B] \propto \int \mathrm{d}\boldsymbol{r}\ \chi_0 N\ \hat{\phi}_A(\boldsymbol{r})\hat{\phi}_B(\boldsymbol{r})$

$$N = 2^{14}$$
$$\Downarrow$$
$$N = 2^{7}$$

SOMA
yes, soft matters

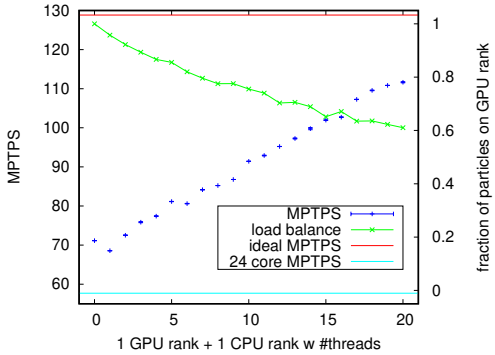
https://gitlab.com/InnocentBug/SOMA
L. Schneider and M. Müller, Comput. Phys. Commun. **235C**, 463–476 (2019)

# Why OpenACC?

- OpenACC:
  - pragma based accelerator description
- open standard
- support for the implementation (GPU-Hackathon)
- implemented in PGI and partly in GCC
- CPU and GPU implementation:
  - single code base
  - CPU and GPU version work together
- high acceptance:
  - modifications only require C

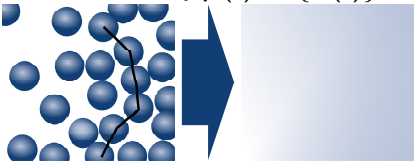
OpenACC

More Science, Less Programming



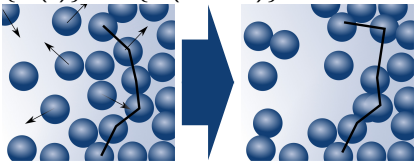
# Single-Chain-in-Mean-Field algorithm

1. calculate density $\rho(t) \leftarrow \{\boldsymbol{r}_i(t)\}$



2. bond force-biased Monte-Carlo

$\{\boldsymbol{r}_i(t)\} \overset{\rho(t)}{\rightarrow} \{\boldsymbol{r}_i(t+\Delta t)\}$



3. repeat

K. C. Daoulas and M. Müller, J. Chem. Phys. **125**, 184904 (2006)

Implementation:

**Step 1**

- simple reduction problem
- non-bonded: calculation on a grid

**Step 2**

- bond force-biased Monte-Carlo
- exact bond energies
- non-bonded particles are independent

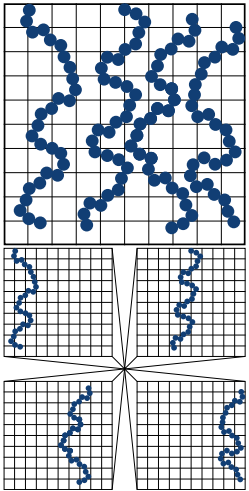
# Density field: atomic reductions

- assign particles on a grid: atomic reductions

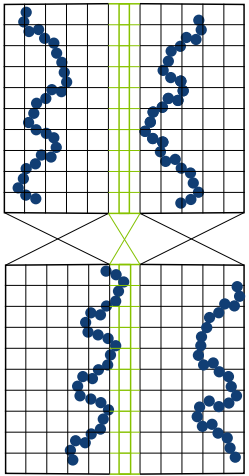
```
#pragma acc parallel loop gang num_gangs(n_polymers)
#pragma omp parallel for
for (uint64_t i = 0; i < n_polymers; i++){
#pragma acc loop vector
  for (unsigned int j = 0; j < N; j++){
    unsigned int monotype = get_particle_type(...);
    unsigned int idx = coord_to_index_unified(...);
#pragma acc atomic update
#pragma omp atomic
    p->fields_32[idx] += 1;}}
```

- 16-bit MPI communication between GPUs

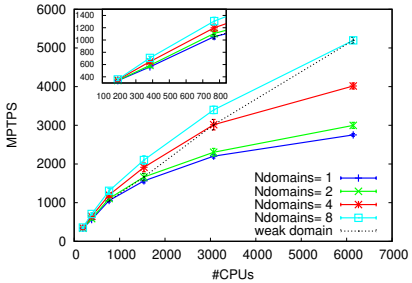
```
#pragma acc update self(p->fields_unified[:])
  MPI_Allreduce( ... , MPI_UINT16_T, ... );
#pragma acc update device(p->fields_unified[:])
```



# What is necessary for engineering scales: spatial domain decomposition!

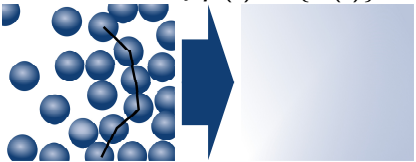


- less memory per `MPI` rank
- all-to-all vs next-neighbor communication
- engineering scale $nN = 4 \cdot 10^9$
- communication bound problem
- linear scaling if constant #CPUs per domain



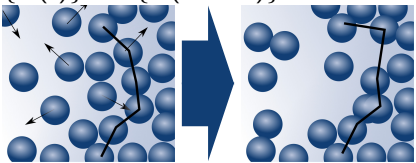
# Single-Chain-in-Mean-Field algorithm

1. calculate density $\rho(t) \leftarrow \{\boldsymbol{r}_i(t)\}$



2. bond force-biased Monte-Carlo

$\{\boldsymbol{r}_i(t)\} \overset{\rho(t)}{\rightarrow} \{\boldsymbol{r}_i(t+\Delta t)\}$



3. repeat

K. C. Daoulas and M. Müller, J. Chem. Phys. **125**, 184904 (2006)

Implementation:

**Step 1**

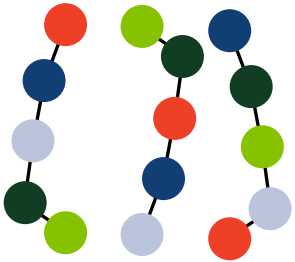
- simple reduction problem
- non-bonded: calculation on a grid

**Step 2**

- bond force-biased Monte-Carlo
- exact bond energies
- non-bonded particles are independent

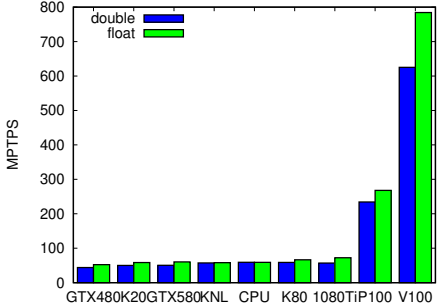
# GPU parallel level: independent molecules

Polymer iteration:



- same color: $\Rightarrow$ parallel
- iterate colors
- 3 threads
- 5 iterations

```
#pragma acc parallel loop
#pragma omp parallel for
for(uint64_t mol=0; mol < Nmol; mol++){
#pragma acc loop seq
  for(int mono=0;mono<getN(mol);mono++){
    const unsigned int i = rng(N);
    smart_MC(i); }}
```




\+ simple independent units

\+ parallelism scales with system

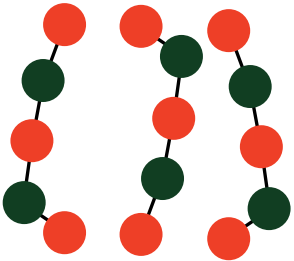
– large polymers (networks)

– no dynamic bonds

L. Schneider and M. Müller, Comput. Phys. Commun. **235C**, 463–476 (2019)

# GPU parallel level: independent sets of not-bonded beads

Independent set iteration:



- higher parallelism
- 9 (6) threads
- 2 iterations

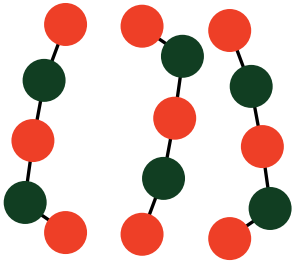
```
#pragma acc parallel loop
#pragma omp parallel for
for (uint64_t mol=0; mol < Nmol; mol++){
//Generate rdm permutation of the sets
//and store result in set_permutation[]

#pragma acc loop seq
  for(int i_set=0; i_set < Nsets;
      i_set++){
    const int set_id =
        set_permutation[i_set];
#pragma acc loop vector
    for(int i_p=0; i_p < setL[set_id];
        i_p++){
      unsigned int ip = sets[...];
      smart_MC(ip);}}
```

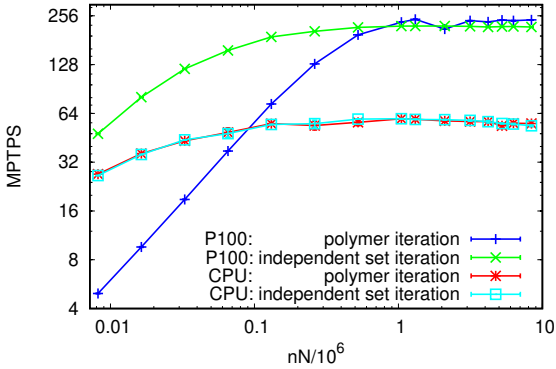
\+ full utilization of parallelism
\+ networks or many nodes
− non-trivial set decomposition
− additional memory and complexity

# GPU parallel level: independent sets of not-bonded beads

Independent set iteration:



- higher parallelism
- 9 (6) threads
- 2 iterations




\+ full utilization of parallelism

\+ networks or many nodes

− non-trivial set decomposition

− additional memory and complexity

L. Schneider and M. Müller, Comput. Phys. Commun. **235C**, 463–476 (2019)

# Pseudo random number definition

## Many random numbers in parallel

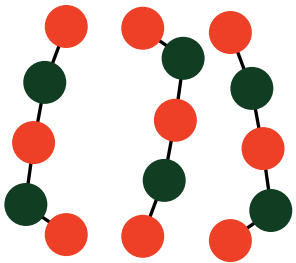
1. streamable pRNGs
   - 1 seed, multiple streams
   - (small) state of pRNG
   - example: PCG **O'NEILL (2014)**
2. hash based pRNGs
   - hash time step, thread, seed and ID
   - requires good hash functions
   - example: random123

     **Salmon, Moraes, et al. (2011)**
3. precompute random numbers
   - memory intensive
   - number of RNGs must be known
   - example: `curand`

- SOMA uses PCG32
- 128 bit state (SOMA is memory bound)
- number of RNGs per step unknown
- easy C implementation
- multiple streams
- one RNG state per parallel thread
- SOMA offers other RNG for verification
  - Mersenne Twister (2504 bytes state)
- SOMA is offers bit-wise reproducible trajectories
  - parallel reductions only on integer numbers

# Optimized architecture storage

- SOMA is memory bound
  - loading RNG state
  - position access (optimized)
  - density field access (“random”)
- reduction of memory access
- utilize different types of GPU memory



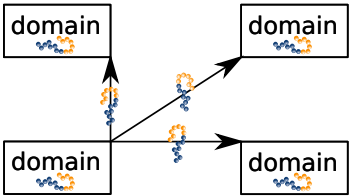
- typical scenario:
  - many molecules (polymers)
  - same bond architecture
- introduce types of molecules
- store bond architecture only once per type
- load architecture to shared GPU memory
- OpenACC: automatically

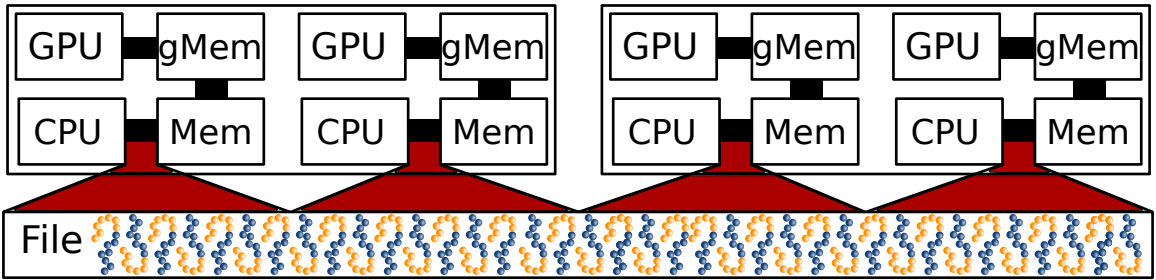
**trade off**

- complex storage
- reduced flexibility
- large molecule become complicated (networks)

# Large systems require parallel IO

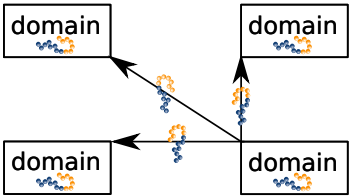
- HDF5 hides MPI-IO which hides the hardware
- HDF5 is widespread and fits in many pipelines
- enables handling files $>$ memory
- domain decomposition requires 2nd communication

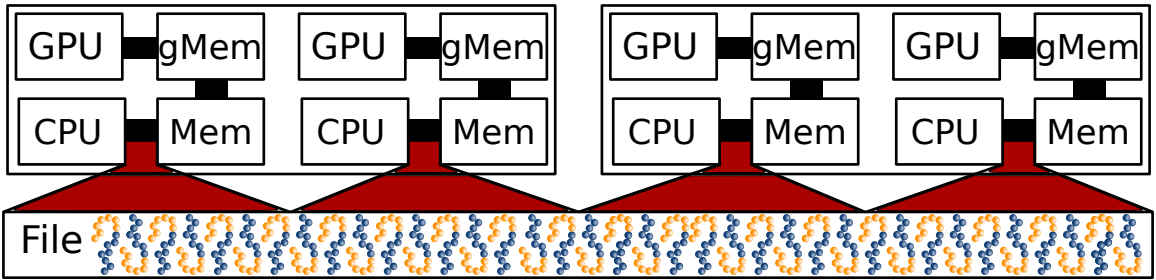


# Large systems require parallel IO

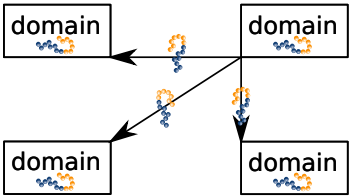
- HDF5 hides MPI-IO which hides the hardware
- HDF5 is widespread and fits in many pipelines
- enables handling files $>$ memory
- domain decomposition requires 2nd communication

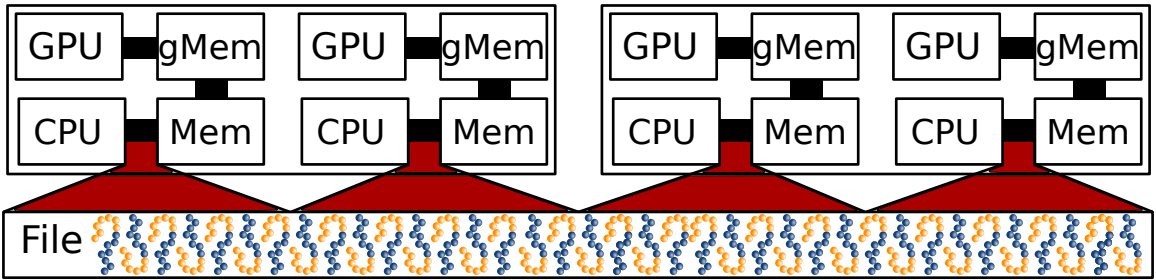


# Large systems require parallel IO

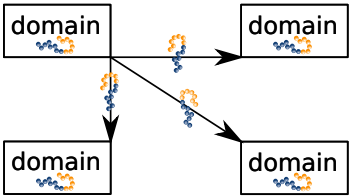
- HDF5 hides MPI-IO which hides the hardware
- HDF5 is widespread and fits in many pipelines
- enables handling files $>$ memory
- domain decomposition requires 2nd communication

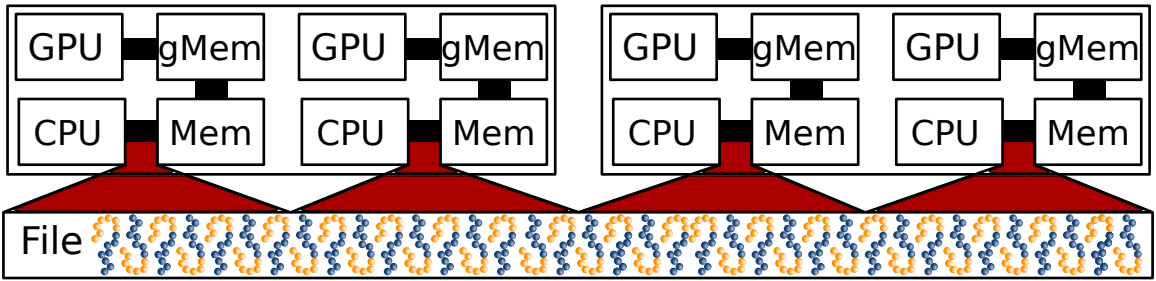


# Large systems require parallel IO

- HDF5 hides MPI-IO which hides the hardware
- HDF5 is widespread and fits in many pipelines
- enables handling files $>$ memory
- domain decomposition requires 2nd communication

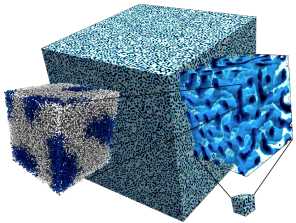


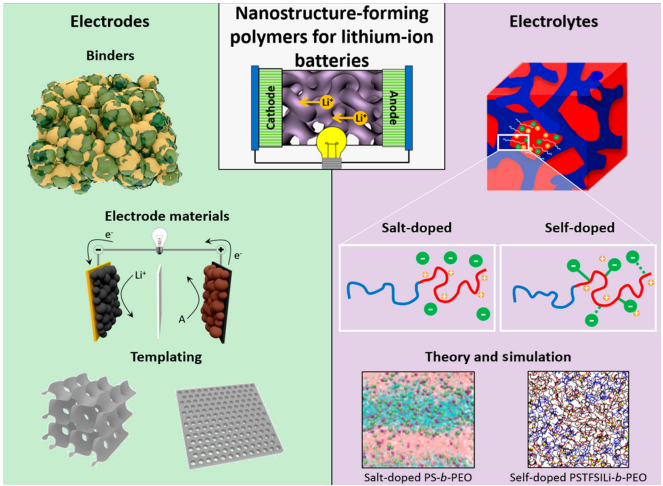
# Overview



SOMA
yes, soft matters



# Next generation of polymeric lithium ion batteries




## Goals

- ion conductivity
- mechanical stability

## Diblock copolymer materials

- self-assembled nanostructures
- block A: conductivity
- block B: stability

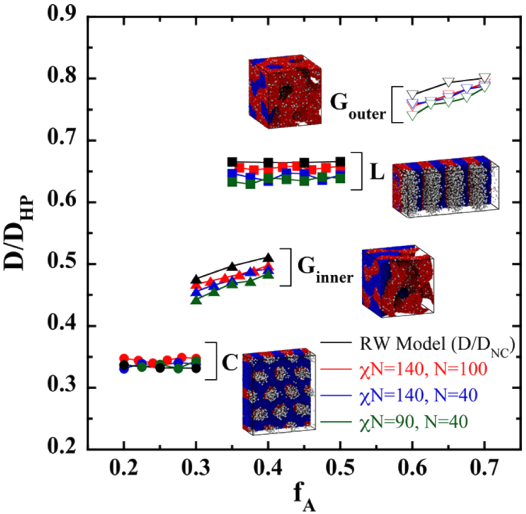
M. A. Morris, H. An, et al., ACS Energy Lett. **2**, 1919–1936 (2017)

# Ion transport in polymer electrolytes

- polymer electrolytes: conduction of ions
- recently investigations
  Shen et.al., Alshammasi et.al., and Zhang et.al.
- investigations of molecular details
- interface roughness effects
- equilibrium morphologies

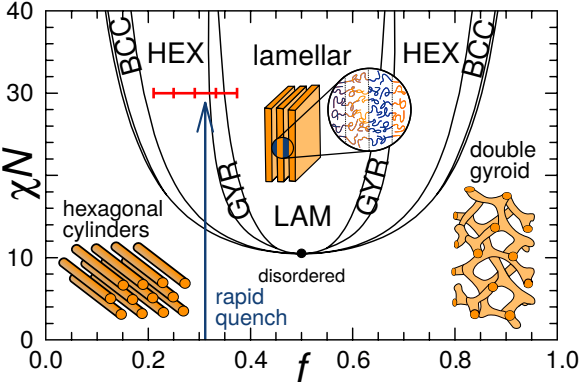
**In this talk!**

- effect of large-scale morphologies
- non-equilibrium meta-stable states



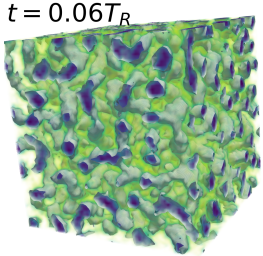

K.-H. Shen, J. R. Brown, et al., ACS Macro Letters **7**, 1092–1098 (2018)
M. S. Alshammasi and F. A. Escobedo, Macromolecules **51**, 9213–9221 (2018)
Z. Zhang, J. Krajniak, et al., ACS Macro Letters **8**, 1096–1101 (2019)

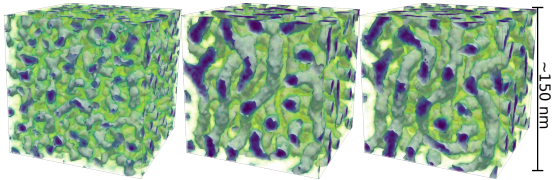
# Metastable network phases




- $\chi N = 30$
- $f \in \left[\frac{7}{32}, \frac{12}{32}\right]$

- rapid quench
- metastable
- $t = 6\,T_R$
- 3D diffusion?

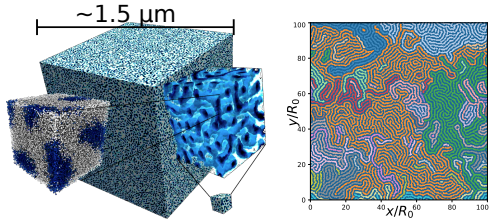


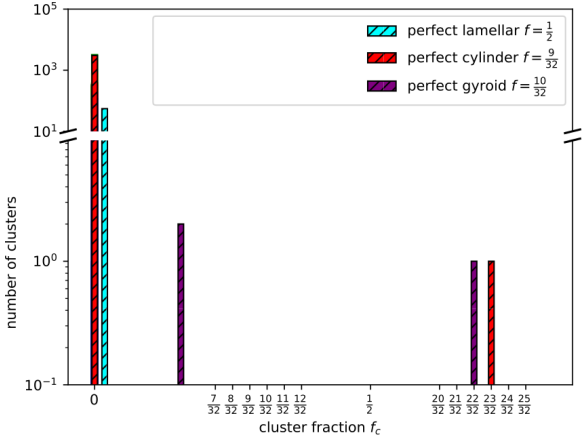

adapted from M. W. Matsen, J. Phys: Condens. Matter **14**, R21 (2001)

L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019)

# 3D network structures for conductivity and stability



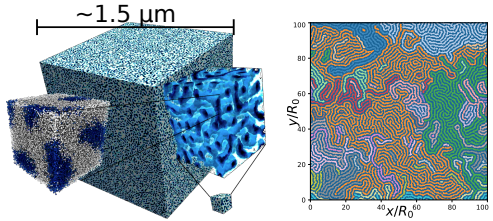

- non-periodic network structures
- $\approx 4.1 \cdot 10^9$ particles
- engineering scale: $L = 0.8$-$2.7\mu$m
- A-phase: omnidirectional diffusion
- B-phase: mechanical stability



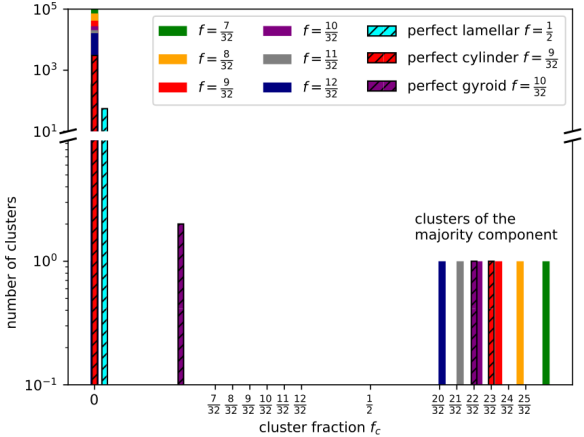

- $f > \frac{7}{32}$: 3D percolating cluster for A and B

L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019)

# 3D network structures for conductivity and stability



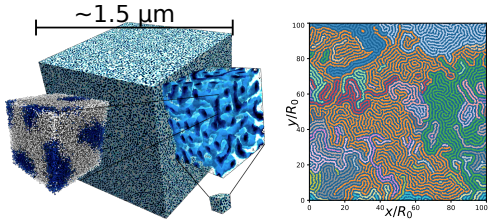

- non-periodic network structures
- $\approx 4.1 \cdot 10^9$ particles
- engineering scale: $L = 0.8$-$2.7\mu$m
- A-phase: omnidirectional diffusion
- B-phase: mechanical stability



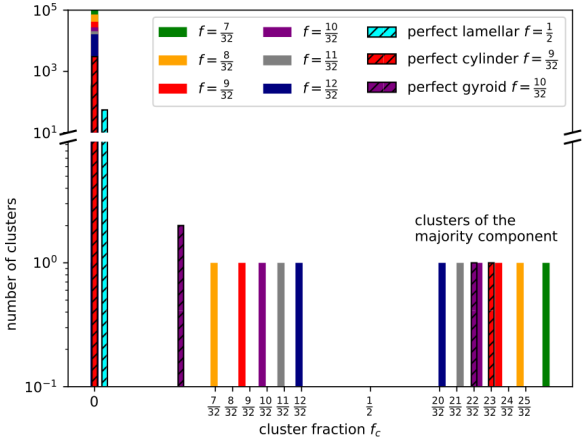

- $f > \frac{7}{32}$: 3D percolating cluster for A and B

L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019)

# 3D network structures for conductivity and stability



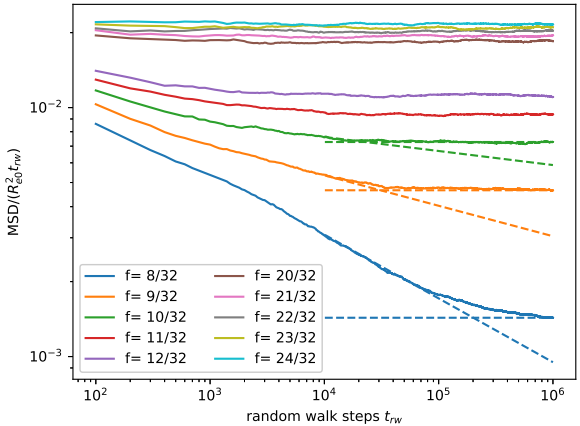

- non-periodic network structures
- $\approx 4.1 \cdot 10^9$ particles
- engineering scale: $L = 0.8$-$2.7\mu$m
- A-phase: omnidirectional diffusion
- B-phase: mechanical stability



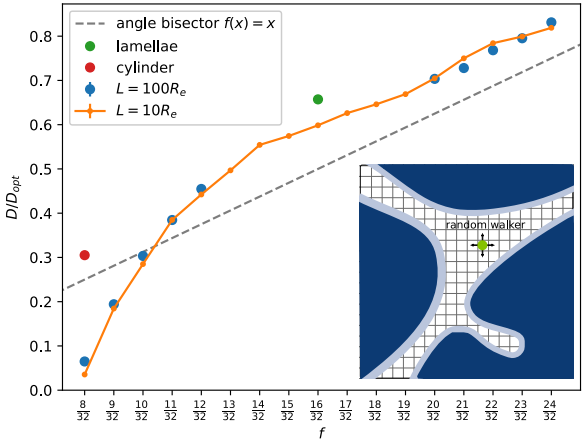

- $f > \frac{7}{32}$: 3D percolating cluster for A and B

L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019)

# Diffusive ion transport




- subdiffusion for $\sqrt{MSD} < (18 \pm 1) R_{e0}$



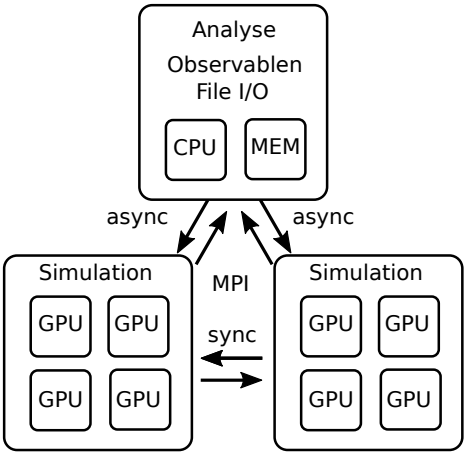

- linear for $f > 1/2$
- decrease for $f < \frac{13}{32}$

Continuous 3D transport of ions!

L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019)

# Dedicated analysis server

- large check-point files ($\approx$ 100GB) unsuitable for analysis
- solution: on-the-fly analysis
- problem: simulation resources idle during analysis
- idea: dedicated analysis server
- simulation snapshot sent via asynchronous MPI
- analysis and simulation parallel on different resources
- easy extension of the analysis server
  - GPU parallel not necessary



# Summary

## Coarse-grained models

- parallel smart-MC
- implemented for GPUs

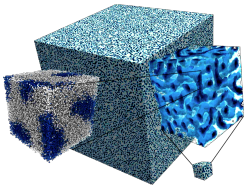
## Battery electrolytes

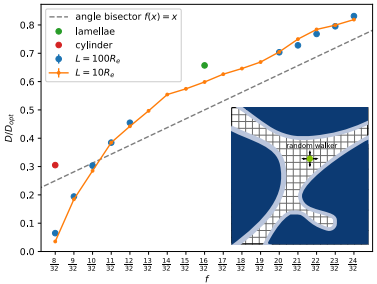
- metastable networks
- 3D conductivity

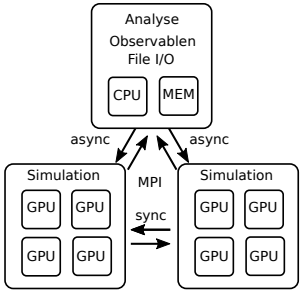
## Outlook

- dedicated analysis server
- other GPU manufacturers?

SOMA
yes, soft matters



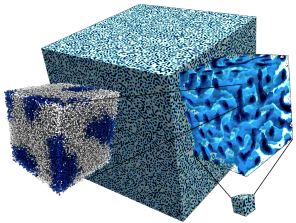


# Overview

1. How to model and simulate engineering scale polymer melts?
   - Why OpenACC for GPUs?
   - Slow, weak interactions via fluctuating fields
   - Parallel hierarchy for strong interactions
   - Detours: implementation quirks
2. Large-scale metastable networks: battery electrolytes
   - Goals for electrolyte materials
   - Percolating network structures
   - Diffusive transport properties
3. Next steps: modular computing
   - Dedicated analysis server

SOMA
yes, soft matters



# References I


L. Schneider and M. Müller, Comput. Phys. Commun. **235C**, 463–476 (2019).

K. C. Daoulas and M. Müller, J. Chem. Phys. **125**, 184904 (2006).

M. E. O'NEILL, ACM Transactions on Mathematical Software (2014).

J. K. Salmon, M. A. Moraes, et al., in Proceedings of 2011 international conference for high performance computing, networking, storage and analysis (2011), pp. 1–12.

M. A. Morris, H. An, et al., ACS Energy Lett. **2**, 1919–1936 (2017).

K.-H. Shen, J. R. Brown, et al., ACS Macro Letters **7**, 1092–1098 (2018).

M. S. Alshammasi and F. A. Escobedo, Macromolecules **51**, 9213–9221 (2018).

Z. Zhang, J. Krajniak, et al., ACS Macro Letters **8**, 1096–1101 (2019).

M. W. Matsen, J. Phys: Condens. Matter **14**, R21 (2001).

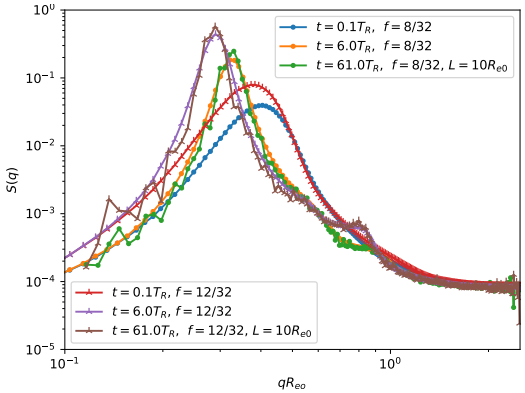
# References II


L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019).

D. Ben-Avraham and S. Havlin, Cambridge university press, (2000).

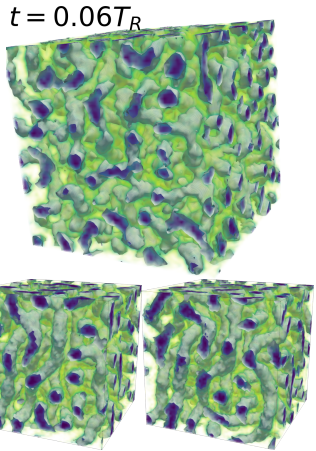
H. J. Heijmans, SIAM review **37**, 1–36 (1995).

# Metastable network phases



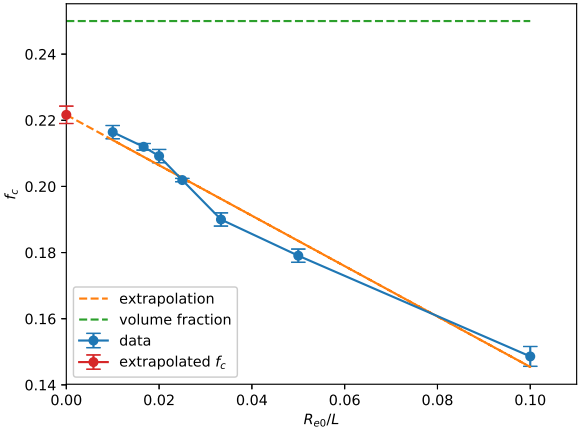

structure factor $S(|\boldsymbol{q}|)$

- rapid quench
- metastable
- $t = 6T_\mathrm{R}$
- 3D diffusion?

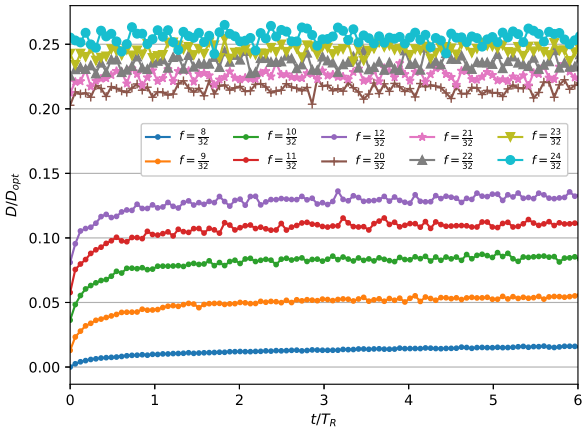



L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019)

# Finite-size effect for cluster fraction $f_c$

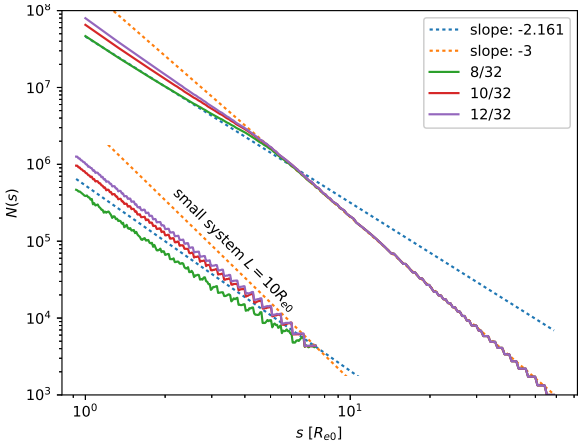



- systematic finite-size effect: $f_c = f_c^* - \alpha/L$




- convergences to plateau after coarsening

# Space-filling characteristics of 3D network structures




- not space-filling $s < s^* \approx (5.4 - 6.4)R_{e0}$
- space-filling on large length scales
- characteristics of an overcritical cluster
- simulations of large sizes required

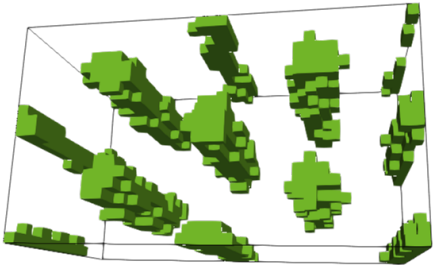
## Box counting algorithm

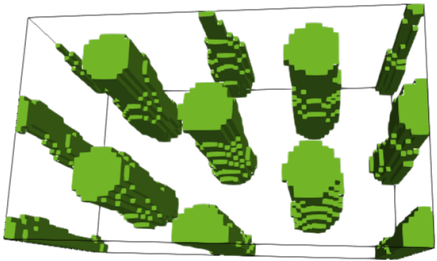
1. divide system if box of length $s$
2. count boxes that contain structure $N(s)$
- $N(s) \propto s^{-d_f}$ fractal dimension

D. Ben-Avraham and S. Havlin, Cambridge university press, (2000)

L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019)

# Grid smoothing for diffusion analysis





1. time average to reduce fluctuations
2. increase grid resolution
3. apply opening $\bigcirc$ and closing $\bullet$
   - $\rho \bigcirc s = (\rho \ominus s) \oplus s$
   - $\rho \bullet s = (\rho \oplus s) \ominus s$

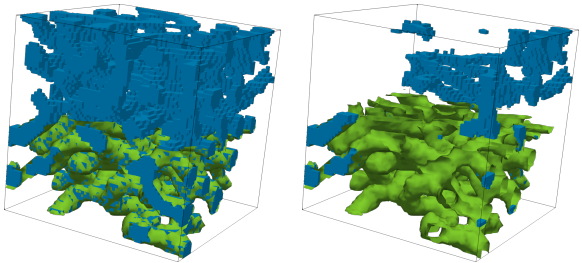
$$(\rho \ominus s)(\boldsymbol{r}) = \inf_{\boldsymbol{r}' \in s} [\rho(\boldsymbol{r} + \boldsymbol{r}') - s(\boldsymbol{r}')]$$

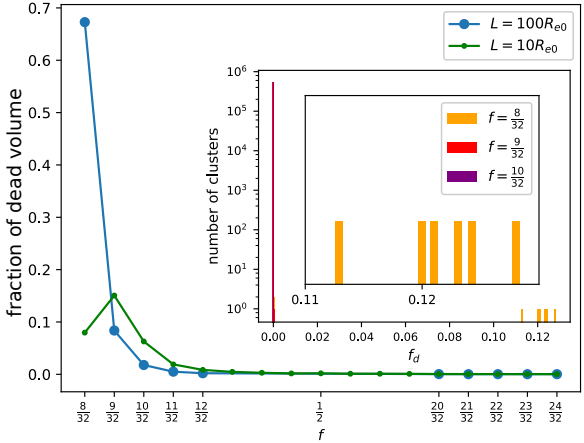
$$(\rho \oplus s)(\boldsymbol{r}) = \sup_{\boldsymbol{r}' \in s} [\rho(\boldsymbol{r}) - s(\boldsymbol{r} - \boldsymbol{r}')]$$

H. J. Heijmans, SIAM review **37**, 1–36 (1995)

# Dead-end analysis

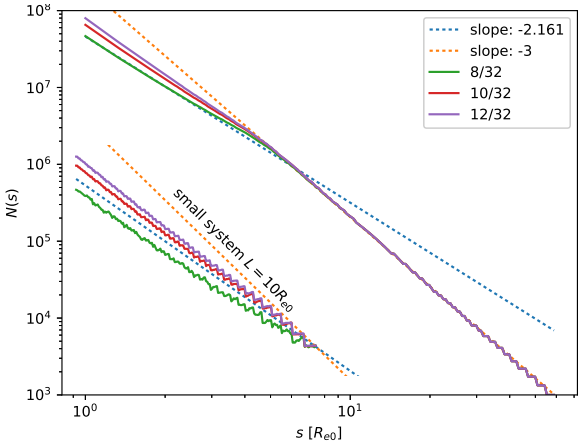
- diffusion $\neq$ directed transport
- dead-ends in network structures
- impact on small volume fractions $f$
- underestimated by small simulations






L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019)

# Space-filling characteristics of 3D network structures




- not space-filling $s < s^* \approx (5.4 - 6.4)R_{e0}$
- space-filling on large length scales
- characteristics of an overcritical cluster
- simulations of large sizes required

**Box counting algorithm**

1. divide system if box of length $s$
2. count boxes that contain structure $N(s)$
- $N(s) \propto s^{-d_f}$ fractal dimension

D. Ben-Avraham and S. Havlin, Cambridge university press, (2000)

L. Schneider and M. Müller, Macromolecules **52**, 2050–2062 (2019)