A Gpu Implementation of Von Mises-Fisher Distribution Algorithm for Polymer Conformation Analysis

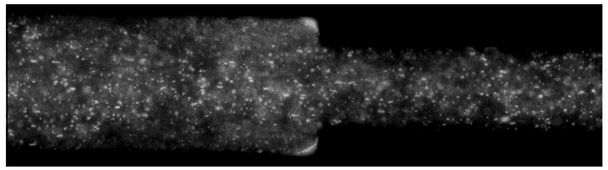
Aidos Abzhanov, Bakytzhan Kallemov

Center for Energy Research, Nazarbayev University, Astana, Kazakhstan

NAZARBAYEV UNIVERSITY

Motivation

- Multiscale models for polymer



Advantages:

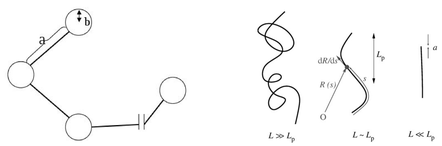
- generally compare better to experiments than pure macroscopic models
- avoid potentially harmful closure operation
- more descriptive and universal

Disadvantages:

- Highly computationally demanding
- no universal approach to the problem

Kramers bead-rod freely jointed polymer model

- Kinetic model of the polymers
- N beads connected by $N-1$ rigid rods.



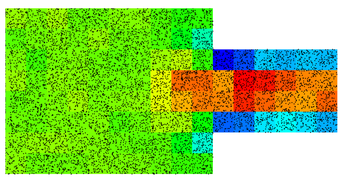
- For D-dimensional space, total degrees of freedom can be huge $\mathcal{O}(2 \times N \times D \times 10^{23})$
- Each of the bead is subjected to constrained Langevin equation

$$d\mathbf{x}_\alpha = \mathbf{v}_\alpha dt$$
$$d\mathbf{v}_\alpha = \gamma(\mathbf{u}(\mathbf{x}_\alpha) - \mathbf{v}_\alpha)dt + \mathbf{f}_{\text{constraint}}dt + \sigma d\mathbf{W}$$

Constraint, corresponds to Kuhn length:

$$\theta_{\alpha+1,\alpha} = \frac{1}{2}\left(\|\mathbf{x}_{\alpha+1} - \mathbf{x}_\alpha\|^2 - a^2\right) = 0.$$

A multiscale framework



Lagrangian markers represent a kinetic state, advected with fluid flow

Fokker-Planck equation for PDF [**?**]

$$\frac{\partial \Psi}{\partial t} = -\mathbf{v}\cdot\nabla_{\mathbf{x}}\Psi - \nabla_{\mathbf{v}}\cdot(\mathbf{F}(\mathbf{x},\mathbf{v})\Psi) + \frac{1}{2}\mathbf{\Gamma}_{\alpha\delta}(\mathbf{x})\mathbf{\Gamma}_{\beta\delta}(\mathbf{x})\frac{\partial^2\Psi}{\partial\mathbf{v}_\alpha\partial\mathbf{v}_\beta}$$

$$\Psi(\mathbf{q}|\mathbf{x},t) \rightarrow \{\mathbf{q}(t),\mathbf{v}(t) \rightarrow \mathbf{q}(t+\Delta t),\mathbf{v}(t+\Delta t)\} \rightarrow \Psi(\mathbf{q}|\mathbf{x}+u\Delta t, t+\Delta t)$$

von Mises-Fisher distribution

- Directional distribution on unit hypersphere in d- dimensions, $d \geq 2$

$$f(\mathbf{x};\mu,\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2}I_{d/2-1}(k)}\exp(\kappa\mu^T\mathbf{x})$$

  - μ - mean direction, $\|\mu\| = 1$
  - κ - concentration parameter, $\kappa \geq 0$
- for $d = 2$ known as von Mises distribution
- for $d = 3$ known as Fisher distribution

$\kappa = 1$ $\kappa = 10$ $\kappa = 100$ 1

Mixture model

- Used to characterize distribution in multiple directions

$$f(\mathbf{x};\{\alpha\},\{\mu\},\{\kappa\}) = \sum_{h=1}^{k}\alpha_h f_h(\mathbf{x};\mu_h,\kappa_h),$$

where α_h - weight of h direction, $\sum\alpha_h = 0, \alpha_h \geq 0$

- $\kappa_1 = 20\ \alpha_1 = 0.5\ \mu_1 = (0,1,0)$
- $\kappa_2 = 20\ \alpha_2 = 0.25\ \mu_2 = (-0.66,-0.66,0.33)$
- $\kappa_3 = 30\ \alpha_3 = 0.25\ \mu_3 = (0.707,0,-0.707)$

Polymer comformation model

- We encode conformation of polymer with mixture of von Mises-Fisher distribution.
- Instead of storing coordinates of each beads of every polymer in ensemble, we store parameters of corresponding mixture model

GPU Parallelization

- Computation of polymer's conformation is done by EM clusterization algorithm [**?**], [**?**] applied to the polymer's rod directions
- This is a time-consuming process when computed for ensemble of polymer's realizations.
- We can improve overall performance by moving computations from CPU to GPU

CUDA Implementation

- Each polymer realization is computed on a seperate CUDA block
- Shared memory is used to store raelizations's local variables, like data points, μ, κ, α
- `kernel<<<ensemble size, polymer size>>>(data points, mu, kappa, alpha)`

Experimental Results

Size of ensemble, ($d = 3, k = 2$)	CPU (sec)	GPU (sec)	Acceleration
10	0.06	0.06	1
20	0.13	0.08	1.6
30	0.24	0.08	3
40	0.30	0.08	3.7
50	0.36	0.08	4.5
60	0.42	0.09	4.6
70	0.50	0.12	4.1

Reference

Acknowledgment

This work was supported by 055 research program from the MES of the Republic of Kazakhstan.