(Refined) Mean Field Approximation for Heterogeneous Systems

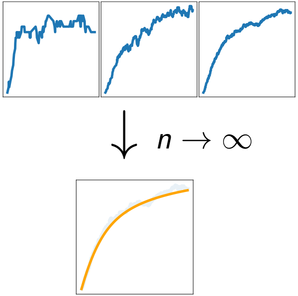
Sebastian Allmeier [1]   Nicolas Gast [1]

[1]Inria Grenoble - Rhône-Alpes;
Univ. Grenoble Alpes

Table of Contents

Table of Contents

Classical Mean Field Setting

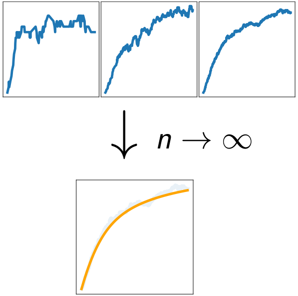


Mean-field methodology:

- $M_s^{(n)}(t) = \frac{1}{n}\{\#\text{ objects in state } s \text{ at time } t\}$
- $\lim_{n\to\infty} M_s^{(n)}(t) = \text{ODE}$

Works for: - objects w/ homogeneous transitions
- groups of objects w/ similar statistical behavior

Classical Mean Field Setting



Mean-field methodology:

- $M_s^{(n)}(t) = \frac{1}{n}\{\#\text{ objects in state } s \text{ at time } t\}$
- $\lim_{n\to\infty} M_s^{(n)}(t) = \text{ODE}$

Works for: - objects w/ homogeneous transitions
- groups of objects w/ similar statistical behavior

Based on object independence assumption.

Can be made more accurate by considering refinements which take dependencies into account.

Classical Mean Field Setting

Aim: Want to analyze transient and steady-state behavior.
Problems:

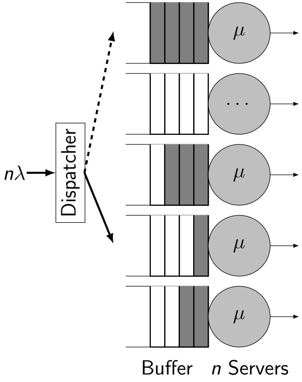
- direct analysis complicated $\rightarrow$ exponentially growing state space $\mathcal{S}^n$
- inefficient and inaccurate simulations

Many modeling possibilities

load balancing, epidemic modeling, caching, communication protocols, SSD garbage collection, malware propagation, ...

Example: Power-of-2-choice model

introduced by Mitzenmacher in 2001

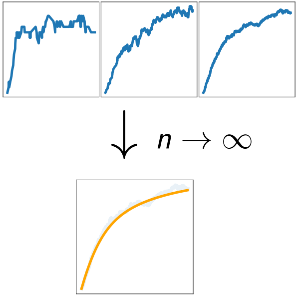


$M_s^{(n)}(t) \sim$ percentage of servers with at least s jobs

- arrival rate $n\lambda$ (λ per server, < 1)
- dispatcher univ. and independently samples two servers
- job added to shorter queue
  (equality broken at random)
- hom. server service rate (usually $\mu = 1$)

a simple & effective distribution scheme

Accuracy homogeneous case / known results



For $m(t)$ the mean field approximation of the occupancy measure $M(t)$

- $\sup_{s \leq t} \left\| M^{(n)}(s) - m(s) \right\| \xrightarrow{n \to \infty} 0$ in probability [1]
- $\mathbb{E}\left[\left\| M^{(n)}(t) - m(t) \right\| \right] = O(\frac{1}{\sqrt{n}})$
- $\mathbb{E}\left[M^{(n)}(t) \right] - m(t) = O(\frac{1}{n})$

[1] $(\forall \epsilon > 0 \ \mathbb{P}(|M^{(n)} - m| > \epsilon) \to 0)$

Table of Contents

We need to model heterogeneity

Importance of Heterogeneity

- heterogeneity has a dramatic impact, e.g. for caching or epidemic modelling
- many homogeneous models ignore heterogeneity
- in general **no theoretical guarantees** for accuracy

Mean Field Approximation

Experiments suggest similar behavior as for homogeneous case.
Advantages remain: simplify analysis + fast computation

Key Question

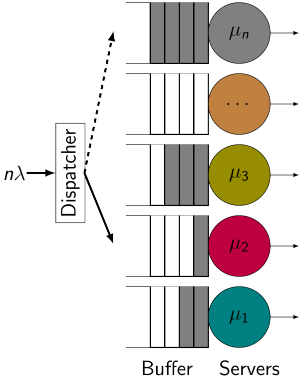
Can we mathematically justify that mean field approximation is a valid technique?

Example: Heterogeneous Power-of-2-choice model (Het. JSQ(2))

represent model using indicators: $X^{(n)}_{(k,s)}(t) = \mathbf{1}_{\{\text{item } k \text{ is in state } s \text{ at time } t\}}$

Example: Heterogeneous Power-of-2-choice model (Het. JSQ(2))

represent model using indicators: $X^{(n)}_{(k,s)}(t) = \mathbf{1}_{\{\text{item } k \text{ is in state } s \text{ at time } t\}}$

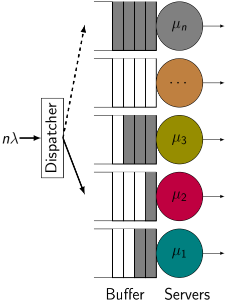


$X_{(k,s)}(t) \sim$ is the buffer of server k is filled with s jobs at time t

- arrival rate $n\lambda$
- uniform and indep. sampling of two servers
- job added to shorter queue (**pairwise transitions**)
- **heterogeneous** server rates μ_i (**unilateral transitions**)

Transitions

$$X^{(n)}_{(k,s)}(t) = \mathbf{1}_{\{\text{item } k \text{ is in state } s \text{ at time} t\}}$$



Departure

$$\mathbf{X} \mapsto \mathbf{X} - e_{(k,s)} + e_{(k,s-1)} \qquad \text{(unilateral)}$$

$$\text{at rate } \mu_k \; X_{(k,s)}$$

Arrival

$$\mathbf{X} \mapsto \mathbf{X} + e_{(k,s+1)} - e_{(k,s)} \qquad \text{(pairwise)}$$

$$\text{at rate } \lambda/n \; X_{(k,s)} X_{(k',s)}$$

$$\text{or } 2\lambda/n \; X_{(k,s)} X_{(k',j)} \quad (j > s)$$

The Generic Heterogeneous Framework

use alternative but equivalent representation: $X^{(n)}_{(k,s)}(t) = \mathbf{1}_{\{S^{(n)}_k(t)=s\}}$

Allowed Dynamics

Interactions of d objects $\mathbf{k} = (k_1, \ldots, k_d)$ jumping from states $\mathbf{s}$ to $\mathbf{s}'$ at rate $\frac{1}{n^{d-1}} r_{\mathbf{k},\mathbf{s}\to\mathbf{s}'}$:

- $\mathbf{X} \mapsto \mathbf{X} - e_{(k,s)} + e_{(k,s')}$

  at rate $r_{k,s\to s'} X_{(k,s)}$ (unilateral)
- $\mathbf{X} \mapsto \mathbf{X} - e_{(k_1,s_1)} + e_{(k_2,s_1')} - e_{(k_2,s_2)} + e_{(k_2,s_2')}$

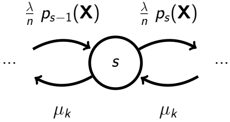
  at rate $\frac{1}{n} r_{\mathbf{k},\mathbf{s}\to\mathbf{s}'} X_{(k_1,s_1)} X_{(k_2,s_2)}$ (pairwise)
- $\ldots$

Definition of the Mean Field

Obtaining the drift

$$f^{(n)}(\mathbf{X}(t)) \approx \sum_{\substack{\text{set of jumps} \\ \text{from } X^{(n)}(t)}} \text{jump} \times \text{jump rate}$$

Zoom on single item
for het. JSQ(2)



The Mean Field Approximation is solution to the IVP:

$$\frac{d}{dt}\phi^{(n)}(t) = f^{(n)}(\phi^{(n)}(t)); \quad \phi^{(n)}(0) = \mathbf{X}^{(n)}(0)$$

Results

- Lipschitz continuous and smooth drift (**by definition**)
- Assumption: uniform bound on rates independent of system size n

Theorem

For the mean field approximation $\phi(t)$ and refinement term $\mathbf{v}(t)$

$$\mathbb{P}\,(\text{item } k \text{ in state } s \text{ at time } t) = \phi_{(k,s)}(t) + O(1/n),$$
$$\mathbb{P}\,(\text{item } k \text{ in state } s \text{ at time } t) = \phi_{(k,s)}(t) + v_{(k,s)}(t) + O(1/n^2).$$

$\rightarrow$ Allows to obtain approximations for linear transformations such as average values (e.g. average queue length)

Refinement & Proof

The heterogeneous Framework allows to define a **refinement term $\boldsymbol{v(t)}$** following the ideas of [Gast, Van Houdt] [Gast et al.].

The **refined mean field approximation** $\underline{\phi(t) + \mathbf{v}(t)}$ increases accuracy by considering dependencies among objects in the approximation.

Proof Idea (Mean Field)

- $\mathbb{P}\,(\text{item } k \text{ in state } s \text{ at time } t) = \mathbb{E}\left[X_{(k,s)}(t)\right]$
- use generator comparison of stochastic and deterministic process (related to Stein's method)
- apply Taylor approximation and bound remainder
- use transition rate structure to obtain $1/n$ bounds

Classical Mean Field Setting

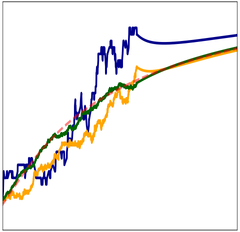
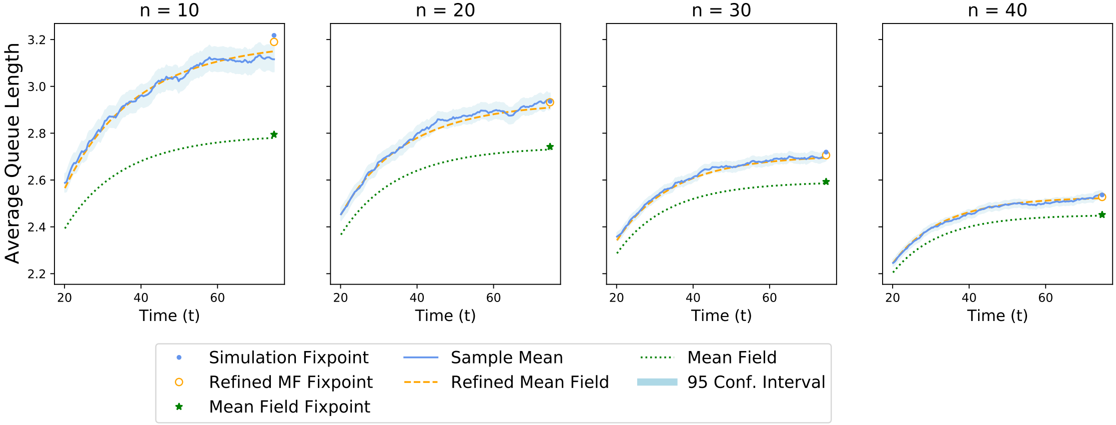


Illustration of $\phi(\mathbf{X}(\tau), t-\tau)$

- want to study $\mathbb{E}\left[\mathbf{X}(t)\right] - \phi(t)$ by relating their generators
- define $\psi(\tau) = \mathbb{E}\left[\phi(\mathbf{X}(\tau), t-\tau)\right]$ with $\phi(\mathbf{X}(\tau), t-\tau)$ the mean field approximation starting in $\mathbf{X}(\tau)$
- rewrite $\mathbb{E}[\mathbf{X}(t)] - \phi(\mathbf{X}(0), t) = \psi(t) - \psi(0)$
- justify $\psi(t) - \psi(0) = \int_0^t \frac{d}{d\tau}\psi(\tau)d\tau$
- $\frac{d}{d\tau}\psi(\tau) \sim$ Expected difference of generators evaluated at $\psi(s)$
- can be bounded using Taylor + rate structure

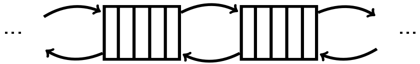
Table of Contents

Increasing Mean Field Accuracy as n grows
Almost exact Refined MF



Approximation Error for a Cache model

The RANDOM cache model in a nutshell:



promote item k
demote random item

	Mean field		Refined mean field		Simulation	
n	Error	(time)	Error	(time)	Error	(time)
10	0.0142	(10ms)	0.00197	(10ms)	0.00026	(4.3s)
30	0.0050	(14ms)	0.00022	(17ms)	0.00047	(4.9s)
50	0.0031	(17ms)	0.00008	(30ms)	0.00055	(5.7s)

RMF is more accurate than simulations of 10^8 requests

Computation Time for the Cache model

	Transient up to $T = 1000$	
n	mean field	refined m.f.
10	30ms	180ms
30	30ms	370ms
50	35ms	1s
100	60ms	14s
300	170ms	–*
1000	970ms	–*

* "–" means that the ODE solver did not finish before 30 seconds.

Numerical Toolbox

rmf_tool – A library to Compute (Refined) Mean Field Approximation(s)

Model Implementation

In [2]:
```
import rmf_tool.src.rmf_tool as rmf

# This code creates an object that represents a "density dependent population process"
ddpp = rmf.DDPP()

# Set parameters
mu, _lambda, K = 1.0, 0.9, 9
```

In [4]:
```
# Add transitions using mathematical formulation:
for i in range(K):
    if i >= 1:
        ddpp.add_transition(e(i), eval('lambda x: _lambda*(x[{}]*x[{}] - x[{}]*x[{}] )'.format(i-1,i-1,i,i) )) # arrivals
    if i < K-1 and i > 0:
        ddpp.add_transition(-e(i), eval('lambda x: mu*(x[{}] - x[{}])'.format(i,i+1))) # removals
```

supports Density Dependent Population Processes & heterogeneous Framework

Numerical Toolbox

In [15]:
```
# Set nitial state
ddpp.set_initial_state(e(0))

# Calculate mean field
T, x_transient = ddpp.ode(time=30)
```

In [16]:
```
# Simulate a trajectory for N=50
T_n50, X_n50 = ddpp.simulate(N=50,  time=30)
# and for N=1000
T_n1k, X_n1k = ddpp.simulate(N=1000,time=30)
```

Allows to easily obtain (for transient and steady state):
mean field, refined mean field approximation, simulations

Table of Contents

Open Questions

- steady-state results for heterogeneous systems
- heterogeneous load balancing (stability)
- two / multiple time scale models

Takeaways

The heterogeneous framework and derived Mean Field Approximation(s)

- can be applied to **fully** and **partially heterogeneous** interaction models
- can be **efficiently implemented**
  (i.e., by using the rmftool: `https://pypi.org/project/rmftool/`
  - 'pip install rmftool')
- exhibit **small hidden constants** in practice
- can be **refined** through the refinement term $\mathbf{v}$

Takeaways

The heterogeneous framework and derived Mean Field Approximation(s)

- can be applied to **fully** and **partially heterogeneous** interaction models
- can be **efficiently implemented**
  (i.e., by using the rmftool: `https://pypi.org/project/rmftool/`
  - 'pip install rmftool')
- exhibit **small hidden constants** in practice
- can be **refined** through the refinement term $\mathbf{v}$

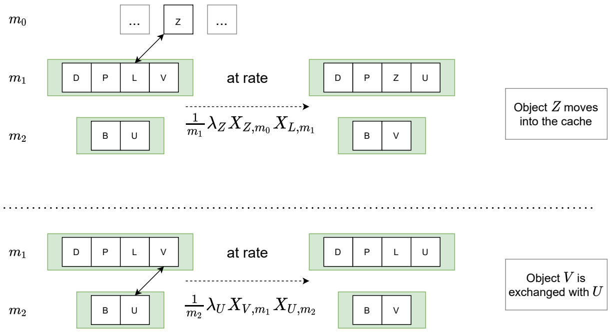
Thank you!

Sebastian Allmeier
sebastian.allmeier@inria.fr

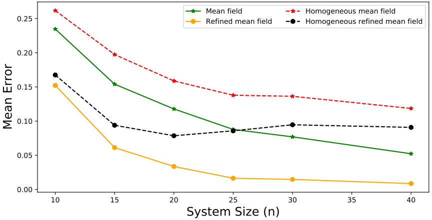
Mean Field and Refined Mean Field Approximations for Heterogeneous Systems: It Works!

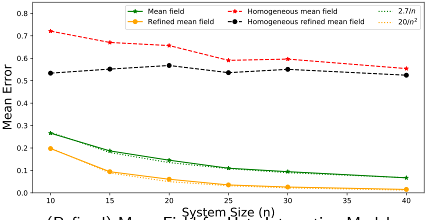
A Caching Example

Requests for object k arrive according to a Poisson process with intensity λ_k



Comparison of homogeneous and heterogeneous approximations





$$
\begin{aligned}
\dot{v}_{(k_1,s_1)}(t) &= \sum_{u\in\mathcal{I}} \frac{\partial f_{(k_1,s_1)}}{\partial x_u}(\mathbf{x}(t))v_u(t) + \frac{1}{2}\sum_{u,l\in\mathcal{I}} \frac{\partial^2 f_{(k_1,s_1)}}{\partial x_l \partial x_u}(\mathbf{x}(t))w_{u,l}(t),\\
\dot{w}_{(k_1,s_1),(k_2,s_2)}(t) &= \sum_{u\in\mathcal{I}} w_{u,(k_2,s_2)}(t)\frac{\partial f_{(k_1,s_1)}}{\partial x_u}(\mathbf{x}(t))\\
&\quad + \sum_{u\in\mathcal{I}} w_{u,(k_1,s_1)}(t)\frac{\partial f_{(k_2,s_2)}}{\partial x_u}(\mathbf{x}(t)) + Q_{(k_1,s_1),(k_2,s_2)}(\mathbf{x}(t)),\\
Q_{(k,s),(k_1,s_1)}(\mathbf{x}) &= \lim_{dt\to 0} \frac{1}{dt}\,\mathbb{E}\left[(\mathbf{X}(t+dt)-\mathbf{X}(t))^{\otimes 2}_{(k,s),(k_1,s_1)} \mid \mathbf{X}(t) = x\right]
\end{aligned}
$$

References

Nicolas Gast and Benny Van Houdt 2017
A Refined Mean Field Approximation. Proceedings of the ACM on Measurement and Analysis of Computing Systems 1, 2

Nicolas Gast, Luca Bortolussi, and Mirco Tribastone 2019
Size Expansions of Mean Field Approximation: Transient and Steady-State Analysis. Performance Evaluation 129 (Feb. 2019)

Sebastian Allmeier and Nicolas Gast 2021
Mean Field and Refined Mean Field Approximations for Heterogeneous Systems: It Works! Proc. ACM Meas. Anal. Comput. Syst., 6(1), (Feb 2022)

Sebastian Allmeier and Nicolas Gast 2021
rmf_tool – a library to compute (refined) mean field approximation(s). SIGMETRICS Perform. Eval. Rev. (Workshop).