

# *Multiscale Simulation of Biochemical Systems*

Linda Petzold, Hong Li and Sotiria Lampoudi  
*University of California Santa Barbara*

Yang Cao  
Virginia Tech

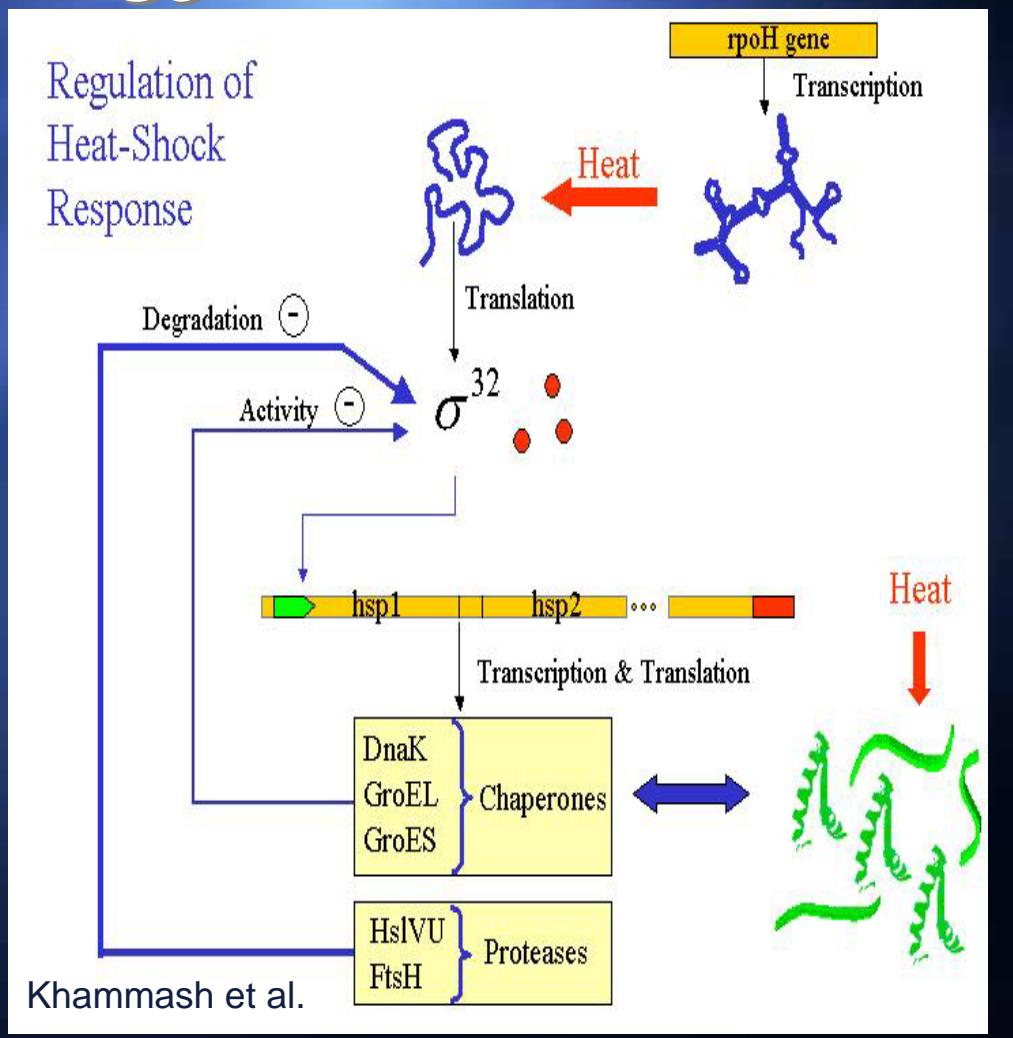
Daniel Gillespie  
*Caltech / UCSB (consultant)*

[www.engineering.ucsb.edu/~cse](http://www.engineering.ucsb.edu/~cse)

# Multiscale Problem in Systems Biology

In the heat-shock response in E. Coli, an estimated 20 - 30 sigma-32 molecules per cell play a key role in sensing the state of the cell and in regulating the production of heat shock proteins. The system cannot be simulated at the fully stochastic level due to:

- Multiple time scales (stiffness)
- The presence of exceedingly large numbers of molecules that must be accounted for in SSA



# *Outline*

- Stochastic Simulation Algorithm (SSA)
- Accelerated discrete stochastic simulation (tau-leaping)
- Stochastic partial equilibrium approximation (slow-scale SSA)
- StochKit software
- Under construction

# *Stochastic Simulation Algorithm*

*(Gillespie 1976)*

- Well-stirred mixture
- $N$  molecular species  $S_1, \dots, S_N$
- Constant temperature, fixed volume  $\Omega$
- $M$  reaction channels  $R_1, \dots, R_M$
- Dynamical state  $X(t) = (X_1(t), \dots, X_N(t))$  where  $X_i(t)$  is the number of  $S_i$  molecules in the system



# *Stochastic Simulation Algorithm*

- Propensity function  $a_j(x)dt$  = the probability, given  $X(t) = X$ , that one  $R_j$  reaction will occur somewhere inside  $\Omega$  in the next infinitesimal time interval  $[t, t + dt]$
- When that reaction occurs, it changes the state. The amount by which  $X_i$  changes is given by  $v_{ij} =$  the change in the number of  $S_i$  molecules produced by one  $R_j$  reaction
- $X(t)$  is a jump Markov process

# *Stochastic Simulation Algorithm*

- Draw two independent samples  $r_1$  and  $r_2$  from  $U(0,1)$

and take  $\tau = \frac{1}{a_0(X)} \ln\left(\frac{1}{r_1}\right)$

$j =$  the smallest integer satisfying  $\sum_{j'=1}^j a_{j'}(x) > r_2 a_0(x)$

- Update  $X$

$$X \leftarrow X + v_j$$

# *Accelerated Discrete Stochastic Simulation: Tau-leaping Method*

- Given a subinterval of length  $\tau$ , if we could determine how many times each reaction channel fired in each subinterval, we could forego knowing the precise instants at which the firings took place. Thus we could leap from one subinterval to the next.
- How long can that subinterval be? Tau-leaping is exact for constant propensity functions, thus  $\tau$  is selected so that no propensity function changes ‘appreciably.’

# *Tau-leaping Method: Some Details*

- Agrees with SSA in the small step size limit
- Equivalent to Forward Euler in the SDE and ODE regimes

## Update Formula

$$\Delta X = \nu P(a(x), \tau)$$

where

$\Delta X$  Change of state

$x$  Current state

$\tau$  Time step

$P(a, \tau)$  Poisson variable with parameters  $a$  and  $\tau$

$a_j(x)$  Propensity functions

$\nu_{ij}$  Change in species  $i$  due to reaction  $j$

# *Mathematical Foundations for Tau-Leaping*

- Order conditions and stability analysis for tau-leaping methods
- Tau-leaping is first order accurate in the moments

# *Implicit Tau for Stiff Discrete Stochastic Systems*

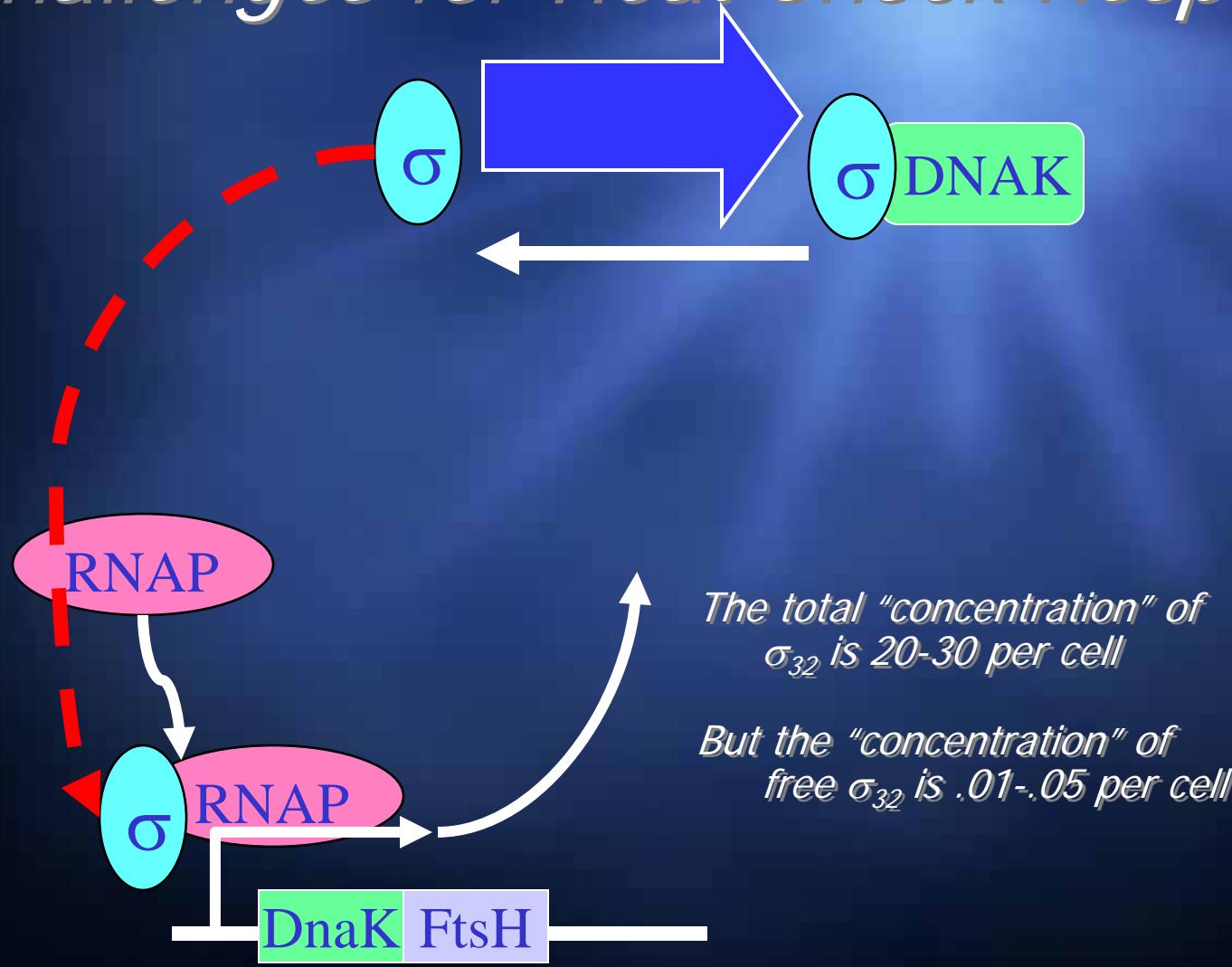
$$\Delta X = \nu \tau a(x + \Delta X) + \nu P(a(x), \tau) - \nu \tau a(x)$$

- Only the **mean** part is implicit
- Tends to the Backward Euler scheme as populations get larger

# *Properties of Implicit Tau*

- First-order accurate in the moments
- For 'large' stepsizes, implicit tau damps the noise in 'fast' species, whereas explicit tau method amplifies the noise
- The distribution information of the fast species can be easily recovered by taking a few small time steps whenever this information is needed - this 'down-shifting' works because the system is a Markov process!

# *A Closer Look at the Multiscale Challenges for Heat Shock Response*



# *Stochastic Partial Equilibrium Approximation – Slow-Scale SSA*

- In deterministic simulation of chemical systems, the partial equilibrium approximation assumes that the fast reactions are always in equilibrium. These fast reactions are thus treated as algebraic constraints.
- In stochastic simulation, the states keep changing. The stochastic partial equilibrium approximation is based on the assumption that the **distributions** of the fast species remain unchanged by the fast reactions.
- Slow-scale approximation Lemma – devil is in the details.

# *Heat Shock Response Model*

- Stochastic Model involves 28 species and 61 chemical reactions. This is a moderate-sized model.
- 12 fast reactions were chosen for the SPEA. The fast reactions were identified from a single SSA simulation to be the ones that fired most frequently. These 12 reactions fire 99% of the total number of times for all reaction channels.
- CPU time for the multiscale SSA
  - Without down-shifting: 3 hours for 10,000 runs
  - With down-shifting: 4 hours for 10,000 runs
- Accuracy:
  - Without down-shifting: slow species distributions accurate
  - With down-shifting: all species distributions accurate
- Speedup for 10,000 simulations (SSA takes 10 days on 1.5 Ghz Workstation)
  - Without down-shifting: 80X faster
  - With down-shofting: 60X faster



- SSA – Fastest SSA Algorithm!
- Constant-stepsize explicit, implicit, trapezoidal tau-leaping modules
- Adaptive-stepsize, non-negativity preserving explicit tau/SSA (IMEX tau/SSA soon)
- Slow-scale SSA
- SBML Converter
- Parallel (across an ensemble)
- More to come! – Adaptive, hybrid, spatially-dependent!



# *Under Construction*

Next-generation algorithms and software for well-mixed systems:

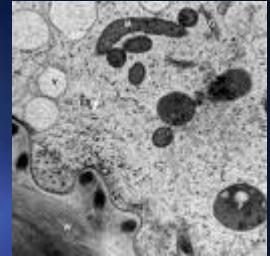
- **Automatic Method and Stepsize Selection**
  - Latest algorithm efficiently and reliably adapts stepsizes for both the stiff and nonstiff cases, detects stiffness and switches between explicit and implicit tau, preserves nonnegativity and uses SSA for reactions involving any species with small population
  - Needs to be combined with ssSSA to handle efficiently problems with fast reactions involving species with small population – slow-scale tau-leaping automatically partitions the reversible reactions in the system
  - Fully-adaptive algorithms and software still to come



- Ultra-fast SSA (computing ensembles)
  - SSA on GPU architecture – initial implementation of our fastest SSA algorithm on \$600 NVIDIA GPU card runs more than 250x faster than workstation alone!
- SSA on the Grid



PNY 8800 GeForce PCIe Graphics Card, 16x8 processors



## Spatially inhomogeneous stochastic systems

- Molecular crowding - how do the propensities change? (*Gillespie, Lampoudi, Petzold 2007*)
- What is an appropriate cell size?
- Parallelization
  - When you can exploit problem structure to decouple the SSA simulations in grid cells between larger time steps, parallelization is easy – fractional step HMKMC Method (*Zheng, Stephens, Braatz, Alkire, Petzold, 2007* - Electrodeposition of copper interconnects in chip fabrication)
  - Otherwise, this is an unsolved problem (we have some ideas)

