Stiffness

Yang Cao
Department of Computer Science
http://courses.cs.vt.edu/~cs6404
Summary

• What is stiffness
• Stiffness in deterministic regime
  – Test Problem
  – Explicit Method and Implicit Method

• Stiffness in Stochastic regime
  – Test Problem
  – Explicit tau-leaping and Implicit tau-leaping
  – Damping Effect
  – Trapezoidal tau-leaping and Down-Shifting Method
Stiffness presents in most multiscale systems, particularly, when the system has the following feature.

- Exhibit slow and fast time scales.
- The fast scales are stable.
- Fast reactions almost cancel each other while slow reactions determine the trend.
- In both the deterministic and stochastic regimes, stiffness is a major reason that makes a simulation unnecessarily slow.
Stiffness in Deterministic Simulation

\[ S_1 \xrightarrow{c_1} \emptyset \]
\[ S_1 + S_1 \xrightarrow{c_2} S_2 \]
\[ S_2 \xrightarrow{c_3} S_1 + S_1 \]
\[ S_2 \xrightarrow{c_4} S_3 \]

\[ \begin{align*}
\dot{x}_1 &= -c_1 x_1 - c_2 x_1^2 + 2c_3 x_2 \\
\dot{x}_2 &= \frac{c_2}{2} x_1^2 - c_3 x_2 - c_4 x_2 \\
\dot{x}_3 &= c_4 x_2 
\end{align*} \]

\[ \begin{align*}
c_1 &= 1 \\
c_2 &= 10 \\
c_3 &= 1000 \\
c_4 &= 0.1 \\
x_1(0) &= 400 \\
x_2(0) &= 798 \\
x_3(0) &= 0 
\end{align*} \]

Solved by ODE45 and Ploted at every 100 steps (total 50353 steps)

Solved by ODE15s Plot at every step (32 steps in total)
Stiff example: Decaying-dimerizing reaction set

- Plot of reaction propensities against time using SSA.
- The fast red and blue reactions almost cancel each other.

**Decaying-Dimerizing Reaction Set**

\[
\begin{align*}
S_1 & \rightarrow 0 \quad c_1 = 1 \\
S_1 + S_1 & \rightarrow S_2 \quad c_2 = 0.002 \\
S_2 & \rightarrow S_1 + S_1 \quad c_3 = 0.5 \\
S_2 & \rightarrow S_3 \quad c_4 = 0.04
\end{align*}
\]

- *Exact SSA run*
  - 2000 steps (reactions) per dot
  - 527,928 steps (reaction events) total
  - X(3) ends at 17125
  - T ends at 43.86
Deterministic Case

For any ODE (ordinary differential equation)

\[ x' = f(x, t) \]

by local linearization, it can be reduced to

\[ x' = Ax \]

Suppose \( A \) has eigenvalues \( \lambda_1, \cdots, \lambda_n \), there exists nonsingular matrix \( T \) such that

\[ T^{-1}AT = \text{diag}\{\lambda_1, \cdots, \lambda_n\} \]

Let \( y = T^{-1}x \). We see a process to convert any ODE to a simple linear test problem

\[ x' = \lambda x \]

We assume the original ODE is stable, thus

\[ \text{Re} (\lambda) \leq 0 \]
Absolute Stability

Think about the case when explicit Euler method applied to the test problem.

\[ x_{n+1} = x_n + hf(x_n) = (1 + \lambda h)x_n \]

We note that the original system should have the following property

\[ |x_{n+1}| \leq |x_n| \]

The absolute stability requires that the numerical solution for test problem also satisfy this property. Thus

\[ |1 + \lambda h| \leq 1 \]

This leads to

\[ h \leq \frac{2}{- \text{Re}(\lambda)} \]

When \(|\text{Re}(\lambda)|\) is large, the stepsize will have to be very small.
Euler Method Applied to Test Problem

For a test problem

\[ x' = -100x \]

**Stepsize = 0.0001, 0.01, 0.018, 0.02**

**Stepsize = 0.021**
More General Test Problem

For any nonlinear function $g(t)$, one can always construct the test problem

$$x' = \lambda(x - g(t)) + g'(t)$$

where $\text{Re}(\lambda) \ll 0$. Or even $x' = \lambda(x - g(t))$

See numerical results (Forward Euler Method) for

$$x' = -100(x - \sin(t)), \quad x(0) = 1$$

Stepsize = 0.001, 0.01, 0.018, 0.02

Stepsize = 0.021
Stiffness

When the absolute stability requirement is far more strict than the accuracy requirement, it is considered stiff. For example, consider the following system:

\[
\begin{align*}
    x_1' &= -0.1 \ x_1 \\
    x_2' &= -100 \ x_2
\end{align*}
\]

This system has two time scale. The fast variable \( x_2 \) converges to its stable state very quickly. The dynamics of the whole system is then represented by the slow variable \( x_1 \). To accurately calculate \( x_1 \), the stepsize can be as large as 0.5 (5% error control). But the absolute stability for the whole system requires a stepsize that is smaller than 0.02.
Implicit Method

Backward (Implicit) Euler method is given by

\[ x_{n+1} = x_n + hf(x_{n+1}) \]

When applied to the test problem, it leads to

\[ x_{n+1} = \frac{1}{1 - \lambda h} x_n \]

Again, the absolute stability requires

\[ (*) \quad \left| \frac{1}{1 - \lambda h} \right| \leq 1 \]

But if \( \text{Re}(\lambda) \leq 0 \), (*) is satisfied for all \( h > 0 \)

Since the absolute stability does not add extra requirement on the stepsize for implicit method, implicit method is a natural choice for stiff problems.
Implicit Euler Method Applied to Test Problem

For test problems

\[ x' = -100x \quad \text{and} \quad x' = -100 \left( x - \sin(t) \right) \]

Stepsize = 0.0001, 0.01, 0.02, 0.04, 0.1

Stepsize = 0.001, 0.02, 0.05, 0.1
Another Point of View: Explicit vs Implicit Method

**Explicit Euler Method**  
\[ x_{n+1} = x_n + hf(x_n) \]

\[ x_n \rightarrow x_{n+1} = x_n + hf(x_n) \rightarrow x_{n+1} \]  

**Backward Euler Method**  
\[ x_{n+1} = x_n + hf(x_{n+1}) \]

\[ x_n \rightarrow x_{n+1} = x_n + hf(x_{n+1}) \rightarrow x_{n+1} \]  

feedback
Newton Iteration

To solve any nonlinear equation

\[ F(x) = 0 \]

Newton iteration gives

\[ x_{k+1} = x_k - \left[F'(x_k)\right]^{-1} F(x_k), \quad k = 0, 1, \ldots \]

Implicit Euler method leads to

\[ F(x) = x - x_n + hf(x) = 0 \]

To solve it, Newton iteration yields

\[ x_{k+1} = x_k - \left[I - hJ(x_n)\right]^{-1} \left(x_k - x_n + hf(x_k)\right) \]

where

\[ J(x_n) = \frac{\partial F}{\partial x}(x_n) \]
Multiscale Challenges for Heat Shock Response
The multiscale behavior can be modeled in the following simple model:

\[ S_1 + S_2 \leftrightarrow S_3 \]
\[ S_1 + S_4 \rightarrow S_5 \]

or a simpler model

\[ S_1 
\leftrightarrow S_2 \]
\[ S_1 \rightarrow S_3 \]

**Features**

- Fast and slow reactions
- Fast reactions usually “less important” than slow ones
- SSA will spend most of its time simulating the fast reactions
Explicit tau leaping applied to stiff problem

Explicit Tau-leaping

\[ X_{n+1} = x_n + vP(\nu x_n \tau) \]

Test Problem (Reversible Isomerization) not a decaying process

\[ S_1 \leftrightarrow S_2 \]

\[ c_1 = c_2 = 100 \]

\[ S_1 \xrightarrow{c_1} \varnothing \]

\[ c_1 = 100 \]

Sample trajectories – stepsizes .005, .009, .02
Stability Analysis

For reversible reactions

\[ S_1 \leftrightarrow S_2 \]

The conservation law is satisfied

\[ X_1 + X_2 = x_t \]

Thus the state can be represented by

\[ X = X_1, \quad X_2 = x_t - X. \]

Tau-leaping method yields

\[ X_{n+1} = x_n + P(c_2(x_t - x_n)\tau) - P(c_1x_n\tau). \]

Take the mean value at both sides

\[ \langle X_{n+1} \rangle = [1 - (c_1 + c_2)\tau]\langle X_n \rangle + c_2\bar{x}_t. \]

Let \[ \lambda = c_1 + c_2. \]

The stability requires:

\[ |1 - \lambda\tau| \leq 1 \]

which leads to

\[ \tau \leq \frac{2}{\lambda} \]
Implicit Tau-Leaping

The update formula

\[ X_{n+1} = x_n + \nu \tau a(X_{n+1}) + \nu \left[ P(a(x_n)\tau) - a(x_n)\tau \right] \]

- Based on the (explicit) tau method
- Only the mean part is implicit
- Reduces to the Backward Euler scheme in the SDE and ODE regimes
- Agrees with SSA in the small step size limit
- Better suited for stiff problems
Implicit tau applied to stiff problem

Reversible Isomerization Reaction

\[ S_1 \leftrightarrow S_2 \]

\[ c_1 = c_2 = 100 \]

Sample trajectories, stepsizes .005, .05
Damping Effect

Although the implicit tau-leaping does not have the stability problem, it does have a side effect called “damping effect”. It will damp the variance when the stepsize is relatively large.

![Graphs showing the effect of different step sizes on the variance.](image)
There are two ways to deal with the damping effect.

1. Trapezoidal Formula

\[
X_{n+1} = x_n + \frac{1}{2} \nu \tau \left[ a(X_{n+1}) - a(x_n) \right] + \nu P(a(x_n) \tau).
\]

2. Down-Shifting Method
Time for a break?!!!
Stiffness II

Yang Cao
Department of Computer Science
http://courses.cs.vt.edu/~cs6404

Virginia Tech
What if $\sigma$ is at a very low copy number?
Michaelis Menton Equation

\[ E + S \xrightleftharpoons[k_1]{k_{-1}} ES \xrightarrow{k_2} P + E \]

Partial Equilibrium Assumption
\[ k_1[E][S] = k_{-1}[ES] \]

Steady State Assumption
\[ k_1[E][S] = k_{-1}[ES] + k_2[ES] \]

Overall reaction rate
\[ S \xrightarrow{E} P \]

\[ v = \frac{k_2 [E]_T [S]}{K_M + [S]} \]

where
\[ K_M = \frac{k_{-1} + k_2}{k_1} \]

Henrici 1903
Michaelis-Menton 1913
Briggs-Haldane 1925
Michaelis Menton Equation

\[ [E][S] = k_M [ES] \]

where

\[ K_M = \frac{k_{-1} + k_2}{k_1} \]

Let \[ [E]_T = [ES] + [E] \]. The above equation becomes

\[ ([E]_T - [ES])[S] = k_M [ES] \]

Thus

\[ [ES] = \frac{[E]_T [S]}{K_M + [S]} \]

Since the production rate satisfies:

\[ \frac{d [P]}{dt} = k_2 [ES] \]

We obtain the overall reaction rate equation

\[ \nu = \frac{k_2 [E]_T [S]}{K_M + [S]} \quad S \longrightarrow^E P \]
Michaelis Menton Equation

M-M equation matches with the data very well when substrate is in excess of enzyme.

\[ [E] \ll [S]. \]

Second Assumption: \[ [E] \ll [S]. \] Thus \[ [S] \approx [S]_T. \]

Otherwise, species S will stay in the form of ES most time. All we know is \[ [S]_T \]

When substrate is in excess

When enzyme is in excess
Related Work

- **SSA**
  - Direct Method and First Reaction Method, Gillespie, 1976, 1977
  - Next Reaction Method, Gibson and Bruck, 2000
  - Optimized Direct Method, Cao et al. 2004

- **Poisson Tau-Leaping Methods**
  - Implicit tau, Rathinam, Petzold, Cao, Gillespie, 2003
  - Trapezoidal tau, Cao et al. 2004
  - Binomial tau-leaping, Tian and Burrage 2004

- **Hybrid Methods**
  - Rao and Arkin, 2003
  - Haseltine and Rawlings, 2002
  - Slow-scale SSA, Cao, Gillespie and Petzold. 2004, 2005
Formulate the Multiscale Difficulty in Simple Models

• The multiscale behavior can be modeled in the following simple model:

\[
S_1 + S_2 \leftrightarrow S_3 \\
S_1 + S_4 \rightarrow S_5
\]

or a simpler model

\[
S_1 \leftrightarrow S_2 \\
S_1 \rightarrow S_3
\]

• Features
  – Fast and slow reactions
  – Fast reactions usually “less important” than slow ones
  – SSA will spend most of its time simulating the fast reactions
Partition the System

• **Partition the reactions:**
  
  - Fast reactions: \( R^f = \{ R_1^f, \ldots, R_M^f \} \)
  
  - Slow reactions: \( R^s = \{ R_1^s, \ldots, R_M^s \} \)
  
  - Criterion:
    
    \[ a_j^f(x) \ll a_j^s, \ \forall j, j' \text{ and for "most" } x \]

• **Partition the species:**
  
  \[ S = (S^f, S^s), \ X(t) = (X_f(t), X_s(t)) \]

  - Fast species: \( S^f = \{ S_1^f, \ldots, S_N^f \} \)
  
  - Slow species: \( S^s = \{ S_1^s, \ldots, S_N^s \} \)
  
  - Criterion: A species is fast if its population gets changed by at least one fast reaction; otherwise, this species is slow.
Virtual Fast Process \( \hat{X}^f(t) \)

- \( \hat{X}^f(t) = \) the fast species populations driven by only the fast reaction channels
- \( \hat{X}^f(t) \) is \( X^f(t) \) with all the slow reactions “turned off”
- \( X^f(t) \) is a physically \textit{real} but non-Markovian process. It does \textit{not} satisfy an ordinary master equation.
- \( \hat{X}^f(t) \) is a physically \textit{fictitious} but Markovian process. It \textit{does} satisfy an ordinary master equation: the ME for the fast species, driven by only the fast reactions, with all slow species populations held constant.
(Stochastic) Partial Equilibrium Approximation

- In deterministic simulation of chemical systems, the partial equilibrium approximation assumes that the fast reactions are always in equilibrium and the fast state variables are at quasi-steady state.

- In stochastic simulation, the state variables 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.
Stochastic Partial Equilibrium

• In the virtual system, the states change with time. The behavior is quite different from the situation in the deterministic model.

• But the distribution of the states at different time points are similar to each other. Moreover, the temporal distribution is similar to the ensemble distribution.

\[ S_1 + S_2 \xleftrightarrow{} S_3 \]
• **Assumption 1:** \( \hat{X}^f(t) \) must be stable; i.e.,

\[
\lim_{t \to \infty} \hat{P}^f(x^f, t \mid x_0, t_0) = \hat{P}(x^f, \infty \mid x_0) \text{ must exist.}
\]

• **Assumption 2:** \( \hat{X}^f(t) \to \hat{X}^f(\infty) \) in a time that is very small compared to the time between slow reaction events.

• **Reasoning of the two assumptions:**
  – The two assumptions should be satisfied whenever stiffness is present. Then the fast reactions will be “less important” than the slow ones, so skipping over them will make sense.
  – If the two conditions cannot be satisfied, the fast reactions are not less important than the slow ones, and skipping over them is not a good idea.
Slow Scale Propensity Theorem

- **Theorem:** With $X(t) = (x^f, x^s)$, let $\Delta_s$ be a time increment that is large compared to the time for $\hat{X}^f(t) \rightarrow \hat{X}^f(\infty)$, but small compared to the time between slow reactions. Then the probability that one $R^s$ will occur in $[t, t + \Delta_s)$ is approximately $\bar{a}^s_j(x^s; x^f)\Delta_s$, where

$$\bar{a}^s_j(x^s; x^f) = \sum_{x^f'} \hat{P}(x^f', \infty | x^f, x^s)a^s_j(x^f', x^s).$$

- $\bar{a}^s_j(x^s; x^f)$ is called the slow-scale propensity function for $R^s_j$.
- It’s the average of $a^s_j(x^f, x^s)$ w.r.t. $\hat{X}^f(\infty)$.
- It replaces $a^s_j(x^f, x^s)$ on the time scale of the slow reactions.
The Slow-Scale SSA

Initialize: Given $X(t_0) = (x^f_0, x^s_0)$, set $t \leftarrow t_0, x^s \leftarrow x^s_0, x^f \leftarrow x^f_0$.

1. In state $(x^f, x^s)$ at time $t$, evaluate $\bar{a}_j^s(x^s; x^f)$ for $j = 1, \ldots, M_s$.

2. Compute $\bar{a}_0^s = \sum_{j=1}^{M_s} \bar{a}_j^s(x^s; x^f)$. With $r_1, r_2 \in U(0,1)$, take

$$
\tau = \frac{1}{\bar{a}_0^s(x^s; x^f)} \ln \left( \frac{1}{r_1} \right),
$$

$$
\mu = \text{smallest integer satisfying } \sum_{j=1}^{\mu} \bar{a}_j^s(x^s; x^f) \geq r_2 \bar{a}_0^s(x^s; x^f).
$$

3. Advance system to the next slow reaction

$$
t \leftarrow t + \tau,
$$

$$
\begin{cases}
    x^s_i &\leftarrow x^s_i + \nu_{i\mu}^{ss} \\ (i = 1, \ldots, N_s), \\
    x^f_i &\leftarrow x^f_i + \nu_{i\mu}^{fs} \\ (i = 1, \ldots, N_f)
\end{cases}
$$

Then “relax” the fast variables: $x^f \leftarrow \text{sample of } \hat{x}^f(\infty)$.

4. Record $X(t) = (x^f, x^s)$. Return to 1, or else stop.
Examples

\[ S_1 \leftrightarrow S_2 \]
\[ S_1 \rightarrow S_3 \]

The virtual fast process will be: \[ S_1 \leftrightarrow S_2 \]

A conservation law is satisfied for the virtual fast process \[ x_1 + x_2 = x_t \]

The whole system can be reduced to a process: \[ S_t \xrightarrow{c_i} S_3 \]

where \( S_t \) represents the sum of \( S_1 \) and \( S_2 \)

The overall reaction rate is calculated as

\[ x_1 + x_2 = x_t \]
\[ c_1 \langle x_1(\infty) \rangle = c_2 \langle x_2(\infty) \rangle \quad \Rightarrow \quad \langle x_1(\infty) \rangle = \frac{c_2}{c_1 + c_2} x_t \]

Thus

\[ C_t = \frac{c_2 c_3}{c_1 + c_2} \]
Examples (Michaelis- Menton)

\[ S_1 + S_2 \xleftrightarrow{c_3} S_3 \rightarrow S_1 + S_4 \]

The virtual fast process will be:

\[ S_1 + S_2 \xleftrightarrow{} S_3 \]

Conservation laws are satisfied for the virtual fast process:

\[ x_1 + x_3 = x_{t1} \]
\[ x_2 + x_3 = x_{t2} \]

We assume \( x_1 \ll x_2 \). Then \( x_{t2} \approx x_2 \)

The whole system can be reduced to a process:

\[ S_2 \xrightarrow{c_i} S_4 \]

The overall reaction rate is calculated as:

\[ x_1 + x_3 = x_{t1} \]
\[ c_1 \langle x_1(\infty) \rangle x_{t2} \approx c_2 \langle x_3(\infty) \rangle \]

\[ \Rightarrow \quad \langle x_3(\infty) \rangle = \frac{c_1 x_{t2}}{c_1 x_{t2} + c_2} x_{t1} \]

Let \( k_m = \frac{c_2}{c_1} \)

Then

\[ \alpha_s(x) = \frac{c_3 x_{t1} x_2}{x_2 + k_m} \]
Stochastic Model involves 28 species and 61 chemical reactions. This is a moderate-sized model.

CPU time on 1.46 Ghz Pentium IV Linux workstation for one SSA simulation is 90 seconds.

CPU time for 10,000 simulations is more than 10 days.

12 fast reaction channels 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 reaction channels 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
Heat Shock Response Model Results

Histogram (10,000 samples) of a slow species mRNA(DNAK) (left) and a fast species DNAK with (right) and without downshifting method (middle) solved by the original SSA method (purple solid line with ‘o’) and the MSSA method (red dashed line with ‘+’), for the HSR model.
Thanks! Questions?

Plato is my friend, Aristotle is my friend, but my best friend is truth --- Newton