Available Methods
Numerical Solution

- The ME

\[
\frac{\partial p_x(x; t)}{\partial t} = \sum_{m=1}^{M} \left\{ \pi_m (x - s_m)p_x(x - s_m; t) - \pi_m (x)p_x(x; t) \right\}, \quad t > 0
\]

of the population process can be expressed as a linear system of coupled first-order differential equations:

\[
\frac{dp(t)}{dt} = Pp(t), \quad t > 0
\]

- \( p(t) \) is a \( K \times 1 \) vector that contains the nonzero probabilities \( p_x(x; t) \).

- \( P \) is a large \( K \times K \) sparse matrix whose structure can be inferred from the ME.
Numerical Solution

- For large networks, matrix $P$ is highly sparse.
- When the columns of the net stoichiometric matrix $S$ are all different from each other, the $i$-th column of $P$ contains at most $M + 1$ nonzero elements.
- The off-diagonal elements are given by
  \[ \pi_m(x_i) > 0, \text{ for } m = 1, 2, ..., M, \]
  where $x_i$ is the $i$-th state.
- The diagonal element is given by
  \[ -\sum_{m=1}^{M} \pi_m(x_i) < 0 \]
- Consequently, the elements of each column of matrix $P$ sum to zero.
Example

Consider a Markovian reaction network comprised of five species and the following five reactions ($N = M = 5$):

\[
\begin{align*}
X_1 + X_2 &\xrightarrow{\kappa_{1,2}} X_3 \\
X_3 &\xrightarrow{\kappa_{2,3}} X_1 + X_2 \\
X_4 &\xrightarrow{\kappa_{3,4}} X_3 \\
X_4 &\xrightarrow{\kappa_{4,4}} X_5 \\
X_5 &\xrightarrow{\kappa_{5,5}} X_4
\end{align*}
\]

Always initialized with $x(0) = x_1$

\[
P = \begin{bmatrix}
-k_5 & k_4 & 0 & 0 \\
k_5 & -(k_3 + k_4) & 0 & 0 \\
0 & k_3 & -k_2 & k_1 \\
0 & 0 & k_2 & -k_1
\end{bmatrix}
\]

\[
\frac{dp(t)}{dt} = Pp(t)
\]

State-space

\[
P(t) = \begin{bmatrix}
p(x_1; t) \\
p(x_2; t) \\
p(x_3; t) \\
p(x_4; t)
\end{bmatrix}
\]

\[
S = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
1 & -1 & 1 & 0 & 0 \\
0 & 0 & -1 & -1 & 1 \\
0 & 0 & 0 & 1 & -1
\end{bmatrix}
\]

\[
x_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad x_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\]
Example

\[
\begin{align*}
\pi_1(x) &= \kappa_1 x_1 x_2 \\
\pi_2(x) &= \kappa_2 x_3 \\
\pi_3(x) &= \kappa_3 x_4 \\
\pi_4(x) &= \kappa_4 x_4 \\
\pi_5(x) &= \kappa_5 x_5
\end{align*}
\]

\[
\begin{pmatrix}
0 \\
0 \\
0 \\
1
\end{pmatrix} = x_1
\]

\[
\begin{pmatrix}
0 \\
0 \\
0 \\
1
\end{pmatrix} - \begin{pmatrix}
0 \\
0 \\
0 \\
1
\end{pmatrix} = x_2 = x_1 - s_4
\]

\[
\frac{\partial p_x(x_1; t)}{\partial t} = \pi_1(x_1 - s_1) p_x(x_1 - s_1; t) - \pi_1(x_1) p_x(x_1; t) + \pi_2(x_1 - s_2) p_x(x_1 - s_2; t) - \pi_2(x_1) p_x(x_1; t) + \pi_3(x_1 - s_3) p_x(x_1 - s_3; t) - \pi_3(x_1) p_x(x_1; t) + \pi_4(x_1 - s_4) p_x(x_1 - s_4; t) - \pi_4(x_1) p_x(x_1; t) + \pi_5(x_1 - s_5) p_x(x_1 - s_5; t) - \pi_5(x_1) p_x(x_1; t)
\]

\[
= \pi_4(x_1 - s_4) p_x(x_1 - s_4; t) - \pi_1(x_1) p_x(x_1; t) - \pi_2(x_1 - s_2) p_x(x_1 - s_2; t) + \pi_3(x_1 - s_3) p_x(x_1 - s_3; t) + \pi_4(x_1) p_x(x_1; t) + \pi_5(x_1 - s_5) p_x(x_1 - s_5; t) - \pi_5(x_1) p_x(x_1; t)
\]

\[
= \pi_4(x_1 - s_4) p_x(x_1 - s_4; t) - \pi_1(x_1) p_x(x_1; t) - \pi_2(x_1 - s_2) p_x(x_1 - s_2; t) + \pi_3(x_1 - s_3) p_x(x_1 - s_3; t) + \pi_4(x_1) p_x(x_1; t) + \pi_5(x_1 - s_5) p_x(x_1 - s_5; t) - \pi_5(x_1) p_x(x_1; t)
\]

impossible states

zero propensities

© John Goutsias (JHU), 2020
Numerical Solution & FSP

- Note that (assuming $K < \infty$)
  \[
  \frac{dp(t)}{dt} = Pp(t) \iff p(t) = \exp(tP)p(0)
  \]

- Therefore, solving the ME is equivalent to evaluating the matrix exponential $\exp(tP)$.

- Unfortunately, this computation is not possible, since the size of matrix $P$ is prohibitively large.

- To reduce the size, we could employ a method known as finite-state projection (FSP) method.

- FSP requires appropriate truncation of the state-space to determine the possible (non-zero probability) set of states and development of a computationally feasible algorithm for calculating the matrix exponential.
One of the best methods for computing matrix exponentials is known as the Krylov subspace approximation (KSA) method.

For a sufficiently small time-step \( \tau > 0 \), this method approximates the vector \( p(t + \tau) = \exp(\tau P)p(t) \) when \( P \) is a large and sparse matrix.

This is done by using a polynomial series expansion of the form:

\[
\hat{p}(t + \tau) = c_0 p(t) + c_1 \tau P p(t) + \cdots + c_{K-1}(\tau P)^{K-1} p(t)
\]
Numerical Solution – KSA Method

\[ \hat{p}(t + \tau) = c_0 p(t) + c_1 \tau Pp(t) + \cdots + c_{K_0-1}(\tau P)^{K_0-1} p(t) \]

- The coefficients \( c_0, c_1, \ldots, c_{K_0-1} \) are estimated by minimizing the least-squares error (LSE) \( \| p(t + \tau) - \hat{p}(t + \tau) \|_2^2 \).

- The optimal \( K_0 \)-order polynomial approximation of \( p(t + \tau) \) is a point in the \( K_0 \)-dimensional Krylov subspace:

\[ \mathcal{K}(t) = \text{span}\left\{ p(t), \tau Pp(t), \ldots, (\tau P)^{K_0-1} p(t) \right\} \]
Numerical Solution – KSA Method

- This element can be approximated by

\[
\hat{p}(t + \tau) = \|p(t)\|_2 V(t) \exp\{\tau H(t)\} e_1
\]

- \(e_1\) is the first column of the \(K_0 \times K_0\) identity matrix.

- \(V(t)\) is a \(K \times K_0\) matrix whose columns form an orthonormal basis for the Krylov subspace \(K(t)\).

- \(H(t)\) is a \(K_0 \times K_0\) Hessenberg matrix (upper triangular with an extra subdiagonal).

- Both matrices are computed by a well-known procedure in linear algebra, known as the Arnoldi procedure.
The KSA method reduces the problem of calculating the exponential of a large and sparse $K \times K$ matrix $P$ to the problem of calculating the exponential of the much smaller and dense $K_0 \times K_0$ matrix $H$ ($K_0 = 30 - 50$ is usually sufficient for many applications).

Computation of the reduced-size problem can be done by standard methods.

The KSA method can be recursively implemented using:

$$\hat{p}((j + 1)\tau) = \exp\{\tau P\} \hat{p}(j\tau)$$

$$= \|\hat{p}(j\tau)\|_2 V(j\tau) \exp\{\tau H(j\tau)\} e_1$$
Numerical Solution – FSP/KSA Method

- Practical implementation of the FSP algorithm is difficult.
- The main issue here is the size of the truncated state-space, which is usually very large.
- For this reason, these methods are most often limited to relatively small reaction networks.
- The KSA method is based on several approximations whose cumulative effect may appreciably affect its accuracy, numerical stability and computational efficiency.
- The KSA method does not guarantee that $\hat{p}(j\tau)$ is a probability vector!!
The ME associated with the DA process can be expressed as a linear system of coupled first-order differential equations:

$$\frac{\partial p_z(z, t)}{\partial t} = \sum_{m=1}^{M} \{a_m(z - e_m)p_z(z - e_m, t) - a_m(z)p_z(z, t)\}, \quad t > 0$$

associated with the DA process can be expressed as a linear system of coupled first-order differential equations:

$$\frac{dq(t)}{dt} = Qq(t), \quad t > 0$$

- $q(t)$ is a $Q \times 1$ vector that contains the nonzero probabilities $p_z(z, t)$.
- $Q$ is a large $Q \times Q$ sparse matrix whose structure can be inferred from the ME.
Ordering the elements in the $Z$ state-space lexicographically results in a matrix $Q$ that is lower triangular.

In this case, and for a given time step $\tau > 0$, we can use the implicit Euler method for solving differential equations to estimate $q(t)$ at discrete time points $j\tau$.

Given an estimate $\hat{q}(j\tau)$ of $q(j\tau)$, we can obtain an estimate $\hat{q}((j+1)\tau)$ of $q((j+1)\tau)$ by solving the following system of linear equations:

$$(I - \tau Q)\hat{q}((j+1)\tau) = \hat{q}(j\tau)$$

$Q \times Q$ identity matrix

https://en.wikipedia.org/wiki/Lexicographic_order
https://en.wikipedia.org/wiki/Backward_Euler_method
This is because, from the differential equation

\[
\frac{dq(t)}{dt} = Qq(t)
\]

we have that

\[
\frac{dq((j+1)\tau)}{dt} = Qq((j+1)\tau)
\]

\[
\Rightarrow \frac{q((j+1)\tau) - q(j\tau)}{\tau} \simeq Qq((j+1)\tau)
\]

\[
\Rightarrow (I - \tau Q)q((j+1)\tau) \simeq q(j\tau)
\]
The previous step is possible for any value of $\tau$ and can be efficiently done by a standard forward substitution algorithm. The resulting method is always stable, producing a valid probability vector at each iteration. Its accuracy can be controlled by a single parameter, the step size $\tau$.

We can obtain $\hat{p}_X(x; t) = \sum_{z \in \mathcal{B}(x)} \hat{p}_Z(z; t)$, where $\mathcal{B}(x) = \{ z : x = x_0 + S z \}$.

https://en.wikipedia.org/wiki/Triangular_matrix#Forward_and_back_substitution
The implicit Euler (IE) method is computationally superior to KSA when the cardinality of the DA state-space is not much larger than the cardinality of the population state-space.

This is not always possible: the DA process is non-decreasing as opposed to the population numbers that can either increase or decrease.

The IE method can only be used when the number of reaction events are sufficiently constrained or remain small during a time interval of interest.
**Example:** In the case of the simple SIR model of epidemiology, the net stoichiometric matrix is given by

\[
S = \begin{bmatrix}
-1 & 0 \\
1 & -1 \\
0 & 1
\end{bmatrix}
\]

and the matrix \( S^T S \) is invertible.

Consequently, there is a one to one correspondence between the \( X \) and \( Z \) state-spaces, since

\[
X(t) = x_0 + SZ(t) \iff Z(t) = (S^T S)^{-1} S^T [X(t) - x_0]
\]

In this case, the IE method is preferable to the KSA method.
Numerical Solution – Implicit Euler Method

- **Example**: Modeling a well-documented 1978 influenza epidemic in an English boarding school.
- We can use the IE method to compute the exact solution of the underlying ME.
- There is a total of 763 students in the school.
- Stochastic SIR model:

  \[
  \begin{align*}
  X_1 & \leftrightarrow S \\
  X_2 & \leftrightarrow I \\
  X_3 & \leftrightarrow R \\
  X_1 + X_2 & \rightarrow 2X_2 \\
  X_2 & \rightarrow X_3 \\
  \pi_1(x_1, x_2, x_3) &= \kappa_1 x_1 x_2 \\
  \pi_2(x_1, x_2, x_3) &= \kappa_2 x_2 \\
  \kappa_1 &= 0.00218 \text{ / day} \\
  \kappa_2 &= 0.44036 \text{ / day}
  \end{align*}
  \]
Initial conditions:

\[ X_1(0) = 762 \]
\[ X_2(0) = 1 \]
\[ X_3(0) = 0 \]

The \( Z \) state space is a 2-D rectangular grid of points from \((0,0)\) to \((762,763)\).

It contains a total of \(763 \times 764 = 582,932\) points.

KSA method \[ \rightarrow \] 4,328 seconds of CPU time.

IE method \[ \rightarrow \] 52 seconds of CPU time.
Numerical Solution – Implicit Euler Method

see video-4-1.mov
Numerical Solution – Implicit Euler Method

Snapshot of the joint conditional probability mass function

$$\Pr[S(t), I(t) \mid I(t) > 0]$$

at the end of the 6-th day of the influenza epidemic.
The mean profiles (green lines) and the ±1 standard deviation profiles (red lines) of
(a) Susceptible students.
(b) Infected students.
(c) Recovered students.

Monte Carlo estimates of the mean and standard deviation profiles of the infected students
are depicted in (d).

Blue circles in (b) mark available data.
Numerical Solution – Implicit Euler method

(a) Evolution of the expected number of recovered students (green line) and the ±1 standard deviations (red lines), given that at least one student is always infected.

(b) The Fano factor (variance/mean) associated with the results in (a) as a function of time.

(c) Dynamic evolution of the probability of no infection $\Pr[I(\infty) = 0]$.

(d) The probability mass function $\Pr[S(\infty) = s, I(\infty) = 0]$ at steady-state.