Detailed population balance modelling of soot in flames

Prof. Markus Kraft

10th IMACS Seminar on Monte Carlo Methods

Linz/Austria July 6-10 2015

Chemical Engineering and Biotechnology
Co-authors

- Wolfgang Wagner
- Mike Goodson
- Jethro Akroyd
- Sebastian Mosbach
- Michael Balthasar
- Jasdeep Singh
- Robert Patterson
- Matt Celnik
- Abhijeet Raj
- Neal Morgan
- Markus Sander
- Dongping Chen
- Zackwan Zainuddin
- Will Menz
- Edward Yapp
Laminar premixed stabilized flame
Soot Formation

Burner

Reaction Zone

Temperature

Flame

Carbon Addition Reactions

Particle Inception by PAHs

Oxidation by O$_2$ and OH,
Trajectory of a soot particle
Trajectory of a soot particle
Detailed population balance model

Gas-phase chemistry up to pyrene

PAH dynamics beyond pyrene

Soot and PAH dynamics
Particle representation

- Aggregate
  - Connectivity
  - Joint surface area

- Primary particle
  - PAHs rigidly stick

- PAH
  - Edge C
  - Site types
  - Fringe length

Sintering level
Detailed population balance model

PAH + PAH
Kinetic Monte-Carlo Aromatic Site (KMC-ARS)

Soot
PAH-Primary Particle (PAH-PP)
Each particle is represented as:

\[ P_q = P_q \left( p_1, \ldots, p_n(p_q), C \right) \]

where \( C \) is the connectivity matrix.

Type space: PAH-PP model

- Connectivity matrix $\mathbf{C}$ stores the common surface area between two primary particles:

$$
\mathbf{C}(P_q) = \begin{pmatrix}
0 & \cdots & 0 & \cdots & 0 \\
C_{21} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
C_{i1} & \cdots & C_{ij} & \cdots \\
\vdots & \vdots & \cdots & \ddots
\end{pmatrix}
$$

- Element $C_{ij}$ of matrix $\mathbf{C}$ has the following properties:

$$
C_{ij} = \begin{cases} 
0, & \text{non-neighbouring} \\
S_{\text{sph}}(p_i, p_j) \leq C_{ij} \leq s(p_i) + s(p_j), & \text{neighbouring}
\end{cases}
$$
Type space: KMC-ARS model

- Each primary $p_i$ is represented as:

$$p_i = p_i(m_1, \ldots, m_{n(p_i)})$$

where $m$ is the exact structure of a PAH
Type space: KMC-ARS model

- Each PAH $m_a$ is fully described by its edge carbon atoms:

$$m_a = m_a(c_1, \ldots, c_{n(m_a)})$$

- Surface reactions on edge carbon atoms
Type space: KMC-ARS model

- Each carbon atom $c_b$ has spatial coordinates:

$$c_b = c_b(\eta_i, \eta_j)$$

where $\eta$ are units in the horizontal and vertical direction.
KMC-ARS model

- Surface reactions happen when gas-phase molecules react with sites on a PAH

- Armchair growth reaction
KMC-ARS transformations

- Surface reactions are described by a set of 20 jump processes:

1. Free-edge ring growth
2. Armchair ring growth
3. 6-member bay closure
4. Benzene addition
5. Free-edge ring desorption
6. Free-edge oxidation by O₂
7. Free-edge oxidation by OH
8. Armchair oxidation by O₂
9. Armchair oxidation by OH
10. Zig-zag 5-member ring addition
11. 5-member ring desorption
12. Armchair 6- to 5-member ring conversion
13. Free-edge 5- to 6-member ring conversion
14. Armchair 5- to 6-member ring conversion
15. 5-member ring migration
16. 6-member ring rearrangement at bay
17. 6-member ring rearrangement at bay
18. 6-member ring desorption at bay
19. Zig-zag oxidation by O₂
20. Zig-zag oxidation by OH
**KMC-ARS transformation**

<table>
<thead>
<tr>
<th></th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fuel composition</strong> (mol%)</td>
<td>50 C\textsubscript{2}H\textsubscript{4}, 50 O\textsubscript{2}</td>
</tr>
<tr>
<td><strong>Velocity</strong> (cm/s)</td>
<td>54</td>
</tr>
<tr>
<td><strong>Pressure</strong> (bar)</td>
<td>0.15</td>
</tr>
<tr>
<td><strong>Equivalence ratio</strong> (-)</td>
<td>3.0</td>
</tr>
</tbody>
</table>

J. Happold, Geschichtete polyzyklische aromatische Kohlenwasserstoffe als Bausteine der Rußbildung, Universität Stuttgart, Stuttgart, 2008

KMC-ARS transformation

<table>
<thead>
<tr>
<th>Values</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel composition (mol%)</td>
<td>50 C₂H₄, 50 O₂</td>
</tr>
<tr>
<td>Velocity (cm/s)</td>
<td>54</td>
</tr>
<tr>
<td>Pressure (bar)</td>
<td>0.15</td>
</tr>
<tr>
<td>Equivalence ratio (-)</td>
<td>3.0</td>
</tr>
</tbody>
</table>

J. Happold, Geschichtete polyzyklische aromatische Kohlenwasserstoffe als Bausteine der Rußbildung, Universität Stuttgart, Stuttgart, 2008

Particle processes: Inception

• Two PAHs in the gas-phase stick together introducing a new particle into the system consisting of one primary

• Inception is implemented as:

$$P_q(p_1(m_1), C) + P_r(p_1(m_1), C) \rightarrow P_s(p_1(m_1, m_2), C)$$
Particle processes: Coagulation

- Coagulation occurs when two particles stick to each other and assume point contact.

- The coagulation of particles $P_q$ and $P_r$ is implemented as:

  $$P_q \left(p_1, \ldots, p_{n(P_q)}, C\right) + P_r \left(p_1, \ldots, p_{n(P_r)}, C\right) \rightarrow P_s \left(p_1, \ldots, p_{n(P_q)}, p_{n(P_q)+1}, \ldots, p_{n(P_q)+n(P_r)}, C'\right)$$

- Primary particle $p_i$ from $P_q$ and a primary particle $p_j$ from $P_r$ are in point contact.
Particle processes: Sintering

- Rearrangement of PAHs to minimise the surface area of the particle
Particle processes: Condensation

- Particles can grow by the deposition of PAHs on the particle surface
- One of the particles is a single PAH species:

\[ P_q(p_1(m_1), C) + P_r(\ldots, p_k(m_1, \ldots, m_n(p_k)), \ldots, C) \rightarrow P_s(\ldots, p_k(m_1, \ldots, m_n(p_k), m(n(p_k)+1)), \ldots, C') \]
Particle processes: Surface reaction

- Surface reactions are described by a set of 20 jump processes:

1. **S1 Free-edge ring growth**
   
2. **S2 Armchair ring growth**
   
3. **S3 6-member bay closure**
   
4. **S4 Benzene addition**
   
5. **S5 Free-edge ring desorption**
   
6. **S6 Free-edge oxidation by O2**
   
7. **S7 Free-edge oxidation by OH**
   
8. **S8 Armchair oxidation by O2**
   
9. **S9 Armchair oxidation by OH**
   
10. **S10 Zig-zag 5-member ring addition**

11. **S11 5-member ring desorption**

12. **S12 Armchair 6- to 5-member ring conversion**

13. **S13 Free-edge 5- to 6-member ring conversion**

14. **S14 Armchair 5- to 6-member ring conversion**

15. **S15 5-member ring migration**

16. **S16 6-member ring rearrangement at bay**

17. **S17 6-member ring rearrangement at bay**

18. **S18 6-member ring desorption at bay**

19. **S19 Zig-zag oxidation by O2**

20. **S20 Zig-zag oxidation by OH**
Particle process: Surface rounding

- Surface rounding by mass addition
Particle process: Surface rounding

- Condensation or surface reaction increases its mass and volume.
- The change in volume results in the change in net common surface area of the primary particle $p_i$ with all its neighbouring primaries:

$$\Delta s(p_i) = \Delta v(p_i) \frac{2s}{d_{pri}(p_i)}$$

Where $s$ is the smoothing factor such that $0 \leq s \leq 2$

- The corresponding change in $C'$ is given by:

$$C_{ij} = \begin{cases} 0, & \text{non-neighbouring} \\ C_{ij} + \Delta s(p_i), & \text{neighbouring} \end{cases}$$
The overall model consists of the kinetic model and the stochastic particle model.

These models can be coupled using operator splitting techniques:

- Strang splitting
- Predictor-Corrector algorithm

Simplest algorithm to solve the population balance is the Direct Simulation Algorithm (DSA).

Numerical issues with DSA addressed by Stochastic Weighted Algorithm (SWA).

Techniques to improve computational efficiency.
Operator splitting technique

- Divide and conquer strategy

\[
\frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = f(x, y) \equiv f_1(x, y) + f_2(x, y)
\]

- First-order splitting: first solve for

\[
\frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = f_1(x, y)
\]

followed by

\[
\frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = f_2(x, y)
\]
Operator splitting technique

- Second-order splitting: Strang splitting

\[
\frac{d}{dt} (x_y) = \frac{1}{2} f_1 (x, y)
\]

\[
\frac{d}{dt} (x_y) = f_2 (x, y)
\]

\[
\frac{d}{dt} (x_y) = \frac{1}{2} f_1 (x, y)
\]
Operator splitting: Examples

• Lotka-Volterra predator-prey equations

\[
\frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x(y - 2) \\ y(1 - x) \end{pmatrix}
\]

• With \( z = (x, y)^T \), a simple splitting is

\[
f_1(x, y) = \begin{pmatrix} x(y - 2) \\ 0 \end{pmatrix}, \quad f_2(x, y) = \begin{pmatrix} 0 \\ y(1 - x) \end{pmatrix}
\]
Operator splitting: Predictor-Corrector

- Consider the differential equation

\[ \frac{dz}{dt} = f(z) \]

- We approximate \( f(z) \) by a polynomial so that

\[ z_{n+1} = z_n + h \left( \beta_0 z_{n+1}' + \beta_1 z_n' + \beta_2 z_{n-1}' + \beta_3 z_{n-2}' + \cdots \right) \]

Where \( z_n' = f(z_n), z_{n-1}' = f(z_{n-1}), z_{n-2}' = f(z_{n-2}), \cdots \)
Operator splitting: Predictor-Corrector

1. Predictor step: explicit method to predict $z_{n+1}$

2. Corrector step: implicit method to use predicted $z_{n+1}$ to correct $z_{n+1}$

\[ z_{n+1} = z_n + h \left( \beta_0 z_{n+1} + \beta_1 z_n' + \beta_2 z_{n-1}' + \beta_3 z_{n-2}' + \cdots \right) \]

\[ z_{n+1} = z_n + h \left( \beta_0 z_{n+1}' + \beta_1 z_n' + \beta_2 z_{n-1}' + \beta_3 z_{n-2}' + \cdots \right) \]
Operator splitting technique

- The rates of certain particle processes depend on the concentration of certain gas-phase species.

Inception (collision efficiency model):

- The gas-phase chemistry and the particle processes are coupled
- Technique to couple a stochastic particle model to gas-phase chemistry
State space

The state of the system \((Q)\) at any time consists of two components

1. The first component \((Q_1)\) contains the concentration of the chemical species \(C:\)
   \[
   Q_1 = \{C_k : k \in \{1, \ldots, N_g\}\}
   \]

2. The second component \((Q_2)\) is the stochastic particle system:
   \[
   Q_2 = \{P_q : q \in \{1, \ldots, N\}\}
   \]
Operator splitting technique

- Evolution of the state of the system $Q$:

$$\frac{d}{dt} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} = \begin{pmatrix} G_1(Q_1) \\ G_2(Q_1, Q_2) \end{pmatrix} + \begin{pmatrix} P_1(Q_1, Q_2) \\ P_2(Q_1, Q_2) \end{pmatrix}$$

- Operator $G$ represents the effects of gas-phase chemical reactions on the system and operator $P$ indicates the effects of the particle processes on the system.

- Subscripts 1 and 2 denote gas-phase and particle-phase, respectively.

- The gas-phase chemistry is best solved using an implicit ODE solution technique, as there are only a finite number of gas-phase variables.

Operator splitting: Strang splitting

**Input:** State of the system $Q_0 = Q_{1,0} + Q_{2,0}$ at initial time $t_0$; Final time $t_f$. **Output:** State of the system $Q_f$ at final time $t_f$.

$t_i \leftarrow t_0, Q_i \leftarrow Q_0$;

while $t_i < t_f$ do

Integrate over time interval $[t_i, t_i + \frac{h}{2}]$

\[
\frac{d}{dt} \begin{pmatrix} Q_1^1 \\ Q_2^1 \end{pmatrix} = \begin{pmatrix} G_1(Q_1^1) \\ G_2(Q_1^1, Q_2^1) \end{pmatrix}
\]

With initial conditions

\[
\begin{pmatrix} Q_1^1(t_i) \\ Q_2^1(t_i) \end{pmatrix} = \begin{pmatrix} Q_{1,i} \\ Q_{2,i} \end{pmatrix}
\]

Solve over time interval $[t_i, t_i + h]$ 

\[
\frac{d}{dt} \begin{pmatrix} Q_1^2 \\ Q_2^2 \end{pmatrix} = \begin{pmatrix} P_1(Q_1^2, Q_2^2) \\ P_2(Q_1^2, Q_2^2) \end{pmatrix}
\]

With initial conditions 

\[
\begin{pmatrix} Q_1^2(t_i) \\ Q_2^2(t_i) \end{pmatrix} = \begin{pmatrix} Q_1^1(t_i + h/2) \\ Q_2^1(t_i + h/2) \end{pmatrix}
\]

DSA

Integrate over time interval $[t_i + \frac{h}{2}, t_i + h]$

\[
\frac{d}{dt} \begin{pmatrix} Q_1^3 \\ Q_2^3 \end{pmatrix} = \begin{pmatrix} G_1(Q_1^3) \\ G_2(Q_1^3, Q_2^3) \end{pmatrix}
\]

With initial conditions 

\[
\begin{pmatrix} Q_1^3(t_i) \\ Q_2^3(t_i) \end{pmatrix} = \begin{pmatrix} Q_1^2(t_i + h) \\ Q_2^2(t_i + h) \end{pmatrix}
\]

ODE solver

Assign solution at $t_{i+1} = t_i + h$

\[
\begin{pmatrix} Q_{1,i+1} \\ Q_{2,i+1} \end{pmatrix} \leftarrow \begin{pmatrix} Q_1^3(t_i + h) \\ Q_2^3(t_i + h) \end{pmatrix}
\]

$i \leftarrow i + 1$
Numerical issues

- Large magnitude source term in ODE system competing with a similarly sized sink term in the population balance
- Splitting step size very small to keep numerical error sufficiently low

Numerical issues

• The system of chemical reactions is inevitably stiff and therefore has to be treated with implicit methods, which involve the inversion of Jacobian matrices

• DSA leads to a discontinuous change in the particle system which makes the reuse of intermediate results impossible

• Reinitialize the ODE solver every splitting step

Predictor-Corrector algorithm: Predictor step

**Input:** Approximate solution $Q_1, Q_2, Q_3, \ldots, Q_i$ at times $t_1 < t_2 < t_3, \ldots, t_i$ where $Q_t = Q_{1,t} + Q_{2,t}$

**Output:** State of the system $Q_{i+1}$ at time $t_{i+1}$.

- Fit a vector of polynomials $P^0(t)$ to $m$ previous points $P_1(Q_{1,i-m}, Q_{2,i-m}), P_1(Q_{1,i-m+1}, Q_{2,i-m+1}), \ldots, P_1(Q_{1,i}, Q_{2,i})$

- Solve over time interval $[t_i, t_{i+1}]$
  
  \[
  \frac{d}{dt} \left( \begin{array}{c}
  \tilde{Q}_1^0(t) \\
  \tilde{Q}_2^0(t)
  \end{array} \right) = \left( \begin{array}{c}
  G_1(\tilde{Q}_1^0) \\
  G_2(\tilde{Q}_1^0, \tilde{Q}_2^0)
  \end{array} \right) + (P^0(t))_0
  \]

  With initial condition
  
  \[
  \left( \begin{array}{c}
  \tilde{Q}_1^0(t_i) \\
  \tilde{Q}_2^0(t_i)
  \end{array} \right) = \left( \begin{array}{c}
  Q_{1,i} \\
  Q_{2,i}
  \end{array} \right)
  \]

- Fit a vector of polynomials $Q^0(t)$ to $Q_1,i-m+1, Q_{1,i-m+2}, Q_{1,i}, Q_1^0(t_{i+1})$

- Solve over time interval $[t_i, t_{i+1}]$
  
  \[
  \frac{d}{dt} \tilde{Q}_2^0 = B_2(Q^0(t), \tilde{Q}_2^0)
  \]

  With initial condition
  
  \[
  \tilde{Q}_2^0(t_i) = \tilde{Q}_2^0(t_{i+1})
  \]

\[
\left( \tilde{Q}_1^0(t_{i+1}), \tilde{Q}_2^0(t_{i+1}) \right)^T \text{ for } t_{i+1}
\]
Predictor-Corrector algorithm: Corrector step

**Input:** Approximate solution $Q_1, Q_2, Q_3, \ldots, Q_i$ at times $t_1 < t_2 < t_3, \ldots, t_i$ where $Q_t = Q_{1,t} + Q_{2,t}$

**Output:** State of the system $Q_{i+1}$ at time $t_{i+1}$

$j \leftarrow 1$

while $j < J$ do

Fit a vector of polynomials $P_j(t)$ to $m + 1$ points $P_1(Q_{1,i-m+1}, Q_{2,i-m+1}), P_1(Q_{1,i-m+2}, Q_{2,i-m+1}), \ldots, P_1(Q_{1,i}, Q_{2,i}), P_1(Q_1^{-1}(t_{i+1}), Q_2^{-1}(t_{i+1}))$

Solve over time interval $[t_i, t_{i+1}]$

\[
\frac{d}{dt} \begin{pmatrix} \bar{Q}_1^j \\ \bar{Q}_2^j \end{pmatrix} = \begin{pmatrix} G_1(Q_1^j) \\ G_2(Q_1^j, Q_2^j) \end{pmatrix} + (P_j(t))
\]

With initial condition

\[
\begin{pmatrix} \bar{Q}_1^j(t_i) \\ \bar{Q}_2^j(t_i) \end{pmatrix} = \begin{pmatrix} Q_{1,i} \\ Q_{2,i} \end{pmatrix}
\]

end

Fit a vector of polynomials $Q_j^j(t)$ to $Q_{1,i-m+1}, Q_{1,i-m+2}, Q_{1,i}, \bar{Q}_1^j(t_{i+1})$

Solve over time interval $[t_i, t_{i+1}]$

\[
\frac{d}{dt} \bar{Q}_2^j = B_2(Q^j(t), \bar{Q}_2^j)
\]

With initial condition

$\bar{Q}_2^j(t_i) = \bar{Q}_2^j(t_{i+1})$

$j \leftarrow j + 1$
Numerical efficiency

![Graphs showing numerical efficiency](image-url)
Direct Simulation Algorithm (DSA)

- Solve the population balance equations
- Every process is directly included in the Monte-Carlo algorithm

Direct Simulation Algorithm (DSA)

**Input:** State of the system $Q_0$ at time $t_0$; Final time $t_f$. **Output:** State of the system $Q_f$ at final time $t_f$.

$t_i \leftarrow t_0, Q_i \leftarrow Q_0$;

**while** $t_i < t_f$ **do**

1. **if** $N_{PAH} < N_g$ **and** $N > N_{max}$
   - **No**
   - **Yes**
     - Uniformly select and remove a particle from the ensemble

2. **if** $N < (0.5 \times N_{max})$
   - **No**
   - **Yes**
     - Uniformly select two particles $P_q$ and $P_r$ and perform process

3. **if** $m = surf$
   - **No**
   - **Yes**
     - Uniformly select a particle $P_q$ and perform process

4. **if** $R_{tot}(Q) = R_{inc}(Q) + R_{coag}(Q) + R_{cond}(Q) + R_{surf}(Q)$
   - **No**
   - **Yes**
     - Calculate an exponentially distributed waiting time $\tau$

5. **if** $P(m) = \frac{R_m(Q)}{R_{tot}(Q)}$
   - **No**
   - **Yes**
     - Choose a process $m$ with probability

6. **end**

**Sinter/round all particles and update their sintering level**

**Double the ensemble and adjust the sample volume**

**Increment** $t \leftarrow t + \tau$
Direct Simulation Algorithm (DSA)

- Problems with a variable size particle ensemble:
  - Removing particles until no particles remain
  - Attempting to add a particle when there is insufficient space
Doubling algorithm

Sample volume, $V$

Maximum capacity, $N_{\text{max}} = 8$ particles
Doubling algorithm

- Maintain statistically significant number of particles

Problems with DSA

• Each stochastic particle represents the same number of physical particles

• DSA yields relatively little information about the rarest particles

  \[ N_{\text{benzene}} \sim 10^{19} \text{ cm}^{-3} \] and \[ N_{\text{soot}} \sim 10^{11} \text{ cm}^{-3} \]

• Large amounts of computational effort is spent on very common PAHs/particles in order to obtain useful estimates concerning the concentrations of rarer, physically significant large particles
Stochastic weighted algorithm (SWA)

- A pair of particles coagulate such that:

\[(x_i, w_i), (x_j, w_j) \rightarrow (x_i + x_j, \gamma(x_i, w_i, x_j, w_j)), (x_j, w_j)\]

Where \(w\) is the statistical weight and \(\gamma\) is a weight transfer function.

- Coagulation events do not reduce the number of stochastic particles.
- Adjust the weight that is attached to each stochastic particle.
- This weight is proportional to the number of ‘real’ particles represented by the stochastic particle.

---

Stochastic weighted algorithm (SWA) – Case study

\[
\text{TEOS} \quad \text{Si(OC}_2\text{H}_5)_4 \rightarrow \text{Silicic acid} \quad \text{Si(OH)}_4 \rightarrow \text{Silica}
\]

<table>
<thead>
<tr>
<th>Model parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>(M_{0,\text{max}}) (m(^3))</td>
<td>(1.55 \times 10^{18})</td>
</tr>
<tr>
<td>(A_s) (sm(^{-1}))</td>
<td>(1.1 \times 10^{16})</td>
</tr>
<tr>
<td>(E_s) (K)</td>
<td>(1.2 \times 10^5)</td>
</tr>
<tr>
<td>(d_{\text{pri, crit}}) (nm)</td>
<td>4.4</td>
</tr>
<tr>
<td>(s)</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process settings</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial temperature (°C)</td>
<td>900</td>
</tr>
<tr>
<td>Residence time (s)</td>
<td>0.8</td>
</tr>
<tr>
<td>Initial TEOS fraction</td>
<td>250 ppm</td>
</tr>
<tr>
<td>Initial total pressure (atm)</td>
<td>1.0</td>
</tr>
</tbody>
</table>
• SWA solution should converge in the limit to DSA solution
• SWA1 = conservation of weight; SWA3 = conservation of mass

• SWA shows convergence for fewer stochastic particles
• SWA has a smaller confidence interval than DSA
Stochastic weighted algorithm (SWA)

START
Select two particles for coagulation

Direct Simulation Algorithm
Add particle 2 to particle 1
Delete particle 2
Is $N(t) < 0.5 \times N_{\text{max}}$?
Yes
Duplicate ensemble
No

Calculate majorant coagulation rate

Calculate true coagulation rate

Is the jump fictitious?
Yes
Adjust particle 1's weight
No
Add particle 2 to particle 1 (leave particle 2 unchanged)

Perform LPDA and sintering updates on particles

Update ensemble statistics

END
• Nonlinear processes cannot be neglected but are dominated in rate by linear ones
• Soot formation in laminar premixed ethylene flames

<table>
<thead>
<tr>
<th>Flame no.</th>
<th>Pressure (bar)</th>
<th>C/O</th>
<th>Cold gas velocity (cm/s)</th>
<th>$T_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>0.68</td>
<td>6.0</td>
<td>1880</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.69</td>
<td>5.9</td>
<td>1711</td>
</tr>
</tbody>
</table>
LPDA - JW1.69

<table>
<thead>
<tr>
<th>Number of tree levels</th>
<th>Time (mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSA</td>
<td>13</td>
</tr>
<tr>
<td>LPDA</td>
<td>13</td>
</tr>
</tbody>
</table>
Number of tree levels | Time (mins)
--- | ---
DSA | 13 | 8497
LPDA | 13 | 6.33
• The total jump rate $R$ for coagulation is:

$$R = \frac{1}{2} \sum_{i \neq j}^{N(t)} \frac{K(x_i, x_j)}{V_{\text{smp}}}$$

• The transition kernel is a popular choice:

$$K^{\text{tr}}(x_i, x_j) = \left(\frac{1}{K^{\text{fm}}(x_i, x_j)} + \frac{1}{K^{\text{sf}}(x_i, x_j)}\right)^{-1}$$
• Repeated selection of pairs of particles $x_i, x_j, i \neq j$ from a population 
  \{x_1, x_2, \ldots, x_n\}
• Selection done so that the probability of a pair being selected is:

$$\frac{K(x_i, x_j)}{R}, R = \sum_{i \neq j}^{N(t)} K(x_i, x_j)$$

• Particles of size $x_i$ and $x_j$ are removed and a particle of size $x_i + x_j$ is added
• Generation of the joint probability distribution of $i$ and $j$ requires summing over $O(N^2)$ terms

• Run time proportional to $N^2$
• A majorant is a function \( \hat{R} \geq K \) for which the computational of

\[
\hat{R} = \sum_{i \neq j}^{N(t)} \hat{K}(x_i, x_j)
\]

and the particle selection of the distribution

\[
\frac{\hat{K}(x_i, x_j)}{\hat{R}}
\]

are relatively fast.
• Possible to recover the distribution and rate defined by $K$ by rejecting the selection of a pair of particles $x_i, x_j$ with probability:

$$1 - \frac{K(x_i, x_j)}{\tilde{K}(x_i, x_j)}$$

• The full free-molecular kernel:

$$K_{fm}(x_i, x_j) = 2.2 \left( \frac{\pi k_B T}{2} \right)^{\frac{1}{2}} \left( \frac{1}{m_i} + \frac{1}{m_j} \right)^{\frac{1}{2}} (d_i + d_j)^2$$

• The free-molecular majorant kernel:

$$\tilde{K}_{fm}(x_i, x_j) = 2.2 k_{maj} \left( \frac{\pi k_B T}{2} \right)^{\frac{1}{2}} \left( \frac{1}{\sqrt{m_i}} + \frac{1}{\sqrt{m_j}} \right) (d_i^2 + d_j^2)$$
• The total majorant rate:

$$\sum_{i \neq j} \hat{K}_{\text{fm}}(x_i, x_j) = 2.2 k_{\text{maj}} \left( \frac{\pi k_B T}{2} \right)^{\frac{1}{2}} \left\{ (N(t) - 1) \sum_{i=1}^{N(t)} d_i^2 m_i^{-\frac{1}{2}} + \sum_{i=1}^{N(t)} m_i^{-\frac{1}{2}} \sum_{i=1}^{N(t)} d^2 + \sum_{i=1}^{N(t)} d_i^2 m_i^{-\frac{1}{2}} \right\}$$

• Independent generation of the indices $i$ and $j$ reduces the computational time to $O(N \log N)$

LPDA, fictitious jumps and majorant kernel

Select two particles for coagulation

Calculate majorant coagulation rate

Perform LPDA and sintering updates on particles

Calculate true coagulation rate

Add particle 2 to particle 1

Delete particle 2

Adjust particle 1’s weight

Add particle 2 to particle 1 (leave particle 2 unchanged)

Duplicate ensemble

Update ensemble statistics

Direct Simulation Algorithm

Stochastic Weighted Algorithm

Is \( N(t) < 0.5 \times N_{\text{max}} \) ?

Is the jump fictitious?

No

Yes

No

Yes

No
- Sample probe integrated into plate
- Temperature profiles and PSDs measured by Wang & co-workers

• Detailed population balance model describes the evolution of aggregates composed of primary particles which are in turn composed of individual polycyclic aromatic hydrocarbons

• Particles are subjected to inception, coagulation, condensation, sintering and a detailed set of surface growth jump processes

• Gas-phase chemistry and particle processes may be coupled through an operator splitting technique

• The simplest stochastic particle method for kinetic population balance equations is the direct simulation Monte Carlo method but has its flaws
• Stochastic weighted algorithm is shown to offer computational advantage over direct simulation algorithm in situations where interest is focused on the larger particles in a system

• Where nonlinear processes cannot be neglected but are dominated in rate by linear ones, deferment of the linear processes is shown to reduce run time by a factor of up to 1000

• Replacement of the actual coagulation kernel by a majorant kernel reduces run time proportionality from $N^2$ to $N$

• The detailed particle model was applied to a burner-stabilised stagnation flame and used to model quantities which may be directly compared with experiments