Simulation of the Belousov-Zhabotinsky Chemical Oscillator using Python

By: Steven Selverston
For: Prof. Jim Crutchfield
PHY-150
University of California, Davis
6/2/2009
The famous B-Z reaction creates fantastic designs in small dishes.
Background:

• Originally found by Boris Belousov sometime in the 1950s—could not get the work published initially
• Study was continued in 1961 by Anatol Zhabotinsky

Simple oscillatory behavior like that shown to the right, can last for several hours before equilibrium is reached*

80-step mechanisms have been proposed

<table>
<thead>
<tr>
<th>1. 80-step mechanisms</th>
<th>2. 80-step mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. İON + H₂ → H⁺ + H₂O</td>
<td>2. İON + H₂ → H⁺ + H₂O</td>
</tr>
<tr>
<td>2. İO⁻ + H⁺ → İO⁻ + H₂</td>
<td>3. İO⁻ + H₂O⁺ → İO⁻ + H₂</td>
</tr>
<tr>
<td>3. İO⁻ + H₂O⁺ → İO⁻ + H₂O</td>
<td>4. İO⁻ + H₂O⁺ → İO⁻ + H₂O</td>
</tr>
<tr>
<td>4. İO⁻ + H₂O⁺ → İO⁻ + H₂O</td>
<td>5. İO⁻ + H₂O⁺ → İO⁻ + H₂O</td>
</tr>
<tr>
<td>5. İO⁻ + H₂O⁺ → İO⁻ + H₂O</td>
<td>6. İO⁻ + H₂O⁺ → İO⁻ + H₂O</td>
</tr>
<tr>
<td>6. İO⁻ + H₂O⁺ → İO⁻ + H₂O</td>
<td>7. İO⁻ + H₂O⁺ → İO⁻ + H₂O</td>
</tr>
</tbody>
</table>

Moral of the story: The chemistry details are extremely complicated
Simpler mechanisms are better for modeling: here, an 11-step mechanism
ODEs: the “Oregonator”

Field-Noyes Version:

\[
\begin{align*}
\frac{dX}{dt} &= k_3 AY - k_2 XY + k_5 BX - 2k_4 X^2 \\
\frac{dY}{dt} &= -k_3 AY - k_2 XY + f k_j Z \\
\frac{dZ}{dt} &= 2k_5 BX - k_j Z
\end{align*}
\]

\[A = B \equiv [\text{BrO}_3^-] \]
\[X \equiv [\text{HBrO}_2] \]
\[Y \equiv [\text{Br}^-] \]
\[Z \equiv [\text{Ce}^{4+}]\]

All rate constants estimated from empirical experiments

Tyson’s Version

• Convenient non-dimensionalization
• Simplest three-variable BZ model

\[
\frac{dx}{dt} = \frac{(qy - xy + x(1-x))}{\epsilon}
\]

\[
\frac{dy}{dt} = \frac{(-qy - xy + fz)}{\epsilon'}
\]

\[
\frac{dz}{dt} = x - z
\]

J.J. Tyson, 1985
Oscillations and Traveling Waves in Chemical Systems
Edited by Field & Burger  pp. 111-112
Limit cycles, but no chaos

Green = $\log_{10}(\text{scaled } \text{Ce}^{4+})$

Red = $\log_{10}(\text{scaled } \text{Br}^-)$

Blue = $\log_{10}(\text{scaled } \text{HBrO}_2)$

Corresponding 3D Attractor

* 2D Plot made by matplotlib
* 3D Plot made by MayaVi
* Enthought Python Distribution
Continuous-flow Stirred Tank Reactors (CSTR)

- Avoids equilibrium by using an open system
- Chaos results from interactions from two frequencies on different timescales*
  - frequency1: BZ limit cycle
  - frequency2: BrMA cycling concentration

\[
\tau = \text{volume (m}^3\)/\text{flowrate (m}^3/\text{s})
\]

\[
k_f = 1/\tau = \text{inverse residence time (similar to flowrate)}
\]

*Gyorgyi & Field, 1992 Nature 355 pp. 808-810
7-step mechanism

(1) Y + X + H  $\rightarrow$  2 V
(2) Y + A + 2H  $\rightarrow$  V + X
(3) 2X  $\rightarrow$  V
(4) 0.5X + A + H  $\rightarrow$  X + Z
(5) X + Z  $\rightarrow$  0.5X
(6) V + Z  $\rightarrow$  Y
(7) Z + M  $\rightarrow$

Y $\equiv$ Br$^-$
X $\equiv$ HBrO$_2$
Z $\equiv$ Ce(IV)
V $\equiv$ BrCH(COOH)$_2$
A $\equiv$ BrO$_3^-$
H $\equiv$ H$^+$
M $\equiv$ CH$_2$(COOH)$_2$

Gyorgyi & Field, 1992 Nature 355 pp. 808-810
chaotic three-variable model based on a 7-parameter rate equation

* k_f term represents the inverse residence time in a CSTR

Scaled Differential Equations:

\[
\begin{align*}
\frac{dx}{dt} &= To\{-k_1 H Y_o x y + k_2 A H^2 Y_o / X_o y - 2k_3 X_o x^2 \\
&\quad + 0.5k_4 A^{0.5} H^{1.5} X_o - 0.5(C - Z_o z)x^{0.5} \\
&\quad - 0.5k_5 Z_o x z - k_f x\}
\end{align*}
\]

\[
\begin{align*}
\frac{dz}{dt} &= To\{k_4 A^{0.5} H^{1.5} X^{0.5}(C/Z_o - z)x^{0.5} - k_5 X_o x z \\
&\quad - \alpha k_6 V_o z v - \beta k_7 M z - k_f z\}
\end{align*}
\]

\[
\begin{align*}
\frac{dv}{dt} &= To\{2k_1 H X_o Y_o / V_o x y + k_2 A H^2 Y_o / V_o y \\
&\quad + k_3 X_o^2 / V_o x^2 - \alpha k_6 Z_o z v - k_f z\}
\end{align*}
\]

Gyorgyi & Field, 1992 Nature 355 pp. 808-810
Period Doubling to Chaos

Decreasing flow rate parameter $k_f$
Chaotic Oregonator for CSTR at
kf = 0.00039  alpha = 666.7  beta = 0.3478  C = 0.000833  stepsize = 0.03
Chaotic Oregonator for CSTR at

\[ kf = 0.00039005 \text{ alpha } = 666.7 \text{ beta } = 0.3478 \text{ C } = 0.000833 \text{ stepsize } = 0.01 \]
Chaotic Oregonator for CSTR at
kf = 0.00039007 alpha = 666.7 beta = 0.3478 C = 0.000833 stepsize = 0.01
attractor formed at low ‘flowrate’ conditions

the attractor is virtually 2D
Higher Flow Rate Conditions

Chaotic Oregonator for CSTR at
\( k_f = 0.0022 \) alpha = 333.3 beta = 0.2609 C = 0.001 stepsize = 0.03

Chaotic Oregonator for CSTR at
\( k_f = 0.00216 \) alpha = 333.3 beta = 0.2609 C = 0.001 stepsize = 0.03
sensitive dependence on initial conditions

320 steps \[ k_f = 0.00216 \]

**Green**: IC = (0.446751, 5.275282, 0.393890)

**Black**: IC = (0.446780, 5.275270, 0.393895)
sensitive dependence on initial conditions

640 steps \( k_f = 0.00216 \)

**Green**: IC = (0.446751, 5.275282, 0.393890)

**Black**: IC = (0.446780, 5.275270, 0.393895)
\[ k_f = 0.0020798 \]
\[ k_f = 0.00208 \]

\[ \sim 1000 \text{ iterations} \]

\[ \sim 20000 \text{ iterations} \]
$k_f = 0.002081$
$k_f = 0.00216$
\[ k_f = 0.00216 \]
Caution: Stiff ODEs

• Runge-Kutta fails (At least without a variable step-size)
• Popular method for integrating these ODE’s is known as the GEAR method
• In Python, one can use SciPy’s `integrate.odeint()`
• Orbit Diagrams in MayaVi can take a very long time
Thanks

Questions?