

Introduction to STELLA

Structural *Thinking* *Experiential* *Learning*
Laboratory with *Animation*

Commerical Products

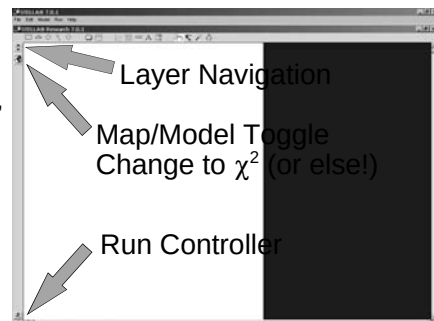
Graphical Interface Differential Equation Solvers (GIDES)

- Stella
 - Free run-time version
- Berkeley Madonna
 - Free run-time version
 - Solves "stiff" differential equations
- VisSim (Mathcad)
- Simulink (Matlab)
- Simile
- Model Maker

Stella Interface

Stella "Layers" – Modeling

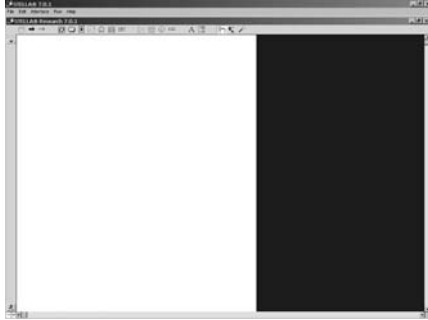
- Construct Model using Building Blocks, Tools, Objects
- Outputs



Stella Interface

Stella "Layers" – Mapping

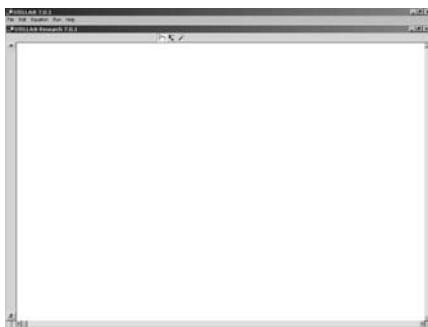
- Text
- Output Tables and Graphs
- Input using Slides and Dials
- Pictures
- Quick Time Movies



Stella Interface

Stella "Layers" – Equation

- Initial Values of Stocks
- Connector Information
- Differential Equations Representing the Time Dependence of Stocks



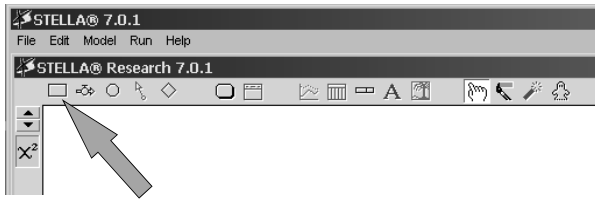
Stella Interface

Menu / Icons



Stella Interface

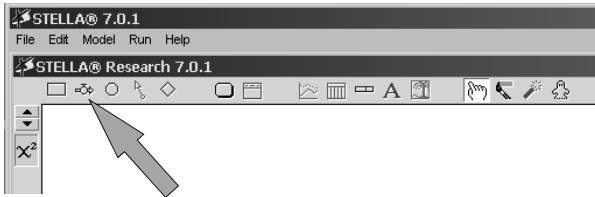
Building Blocks – Stocks



Value Undergoing a Change

Stella Interface

Building Blocks – Flow



Change of Stock with Respect to Time

Stella Interface

Building Blocks – Converter

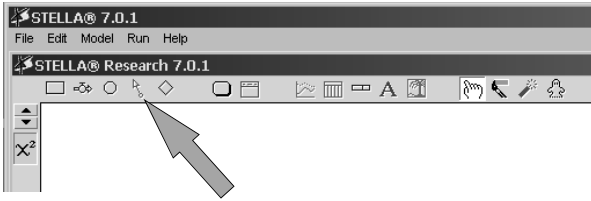


Constants

Transformation Equations

Stella Interface

Building Blocks – Connectors



Links Building Blocks

Stella Interface

Tools – Hand

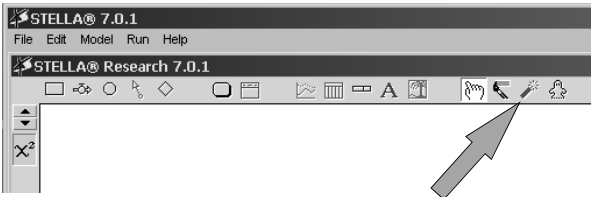


General Purpose Editing Tool

Cursor Pointer

Stella Interface

Tools – Dynamite



Delete (No “Undo”)

Stella Interface

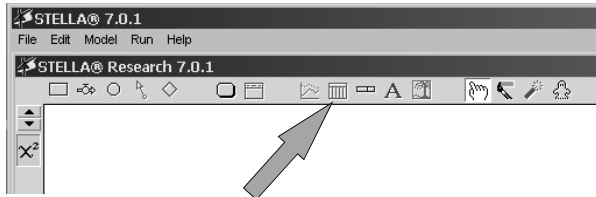
Objects – Graph Pad



Graphing Results

Stella Interface

Objects – Table Pad



Spreadsheet-like Table of Results

RUN SPECS		
Length of simulation:	Unit of time:	Run Mode:
From: 0	☐ Hours	☒ Normal
To: 10	☐ Days	☐ Cycle-time
DT: 0.1	☐ Weeks	
☐ DT as fraction	☐ Months	Interaction Mode:
Pause interval: INF	☐ Quarters	☒ Normal
	☐ Years	☐ Flight Sim
	☒ Other	
	Time	
Integration Method:	Sim Speed:	
☒ Euler's Method	0 real secs = 1 unit time	
☐ Runge-Kutta 2		
☐ Runge-Kutta 4	Min run length: 0 secs	
☐ Analyze Mode: stores run results in memory (0.0 MB required)		
		Cancel OK

[illegible][illegible][illegible]

Solving of Differential Equations

Runge-Kutta 2

$$y(t_m + \Delta t) = y_m + \frac{\Delta t}{2}(\alpha_1 + \alpha_2)$$

$$\alpha_1 = F(t_m, y_m)$$

$$\alpha_2 = F(t_m + \Delta t, y_m + \alpha_1 \Delta t)$$

First 3 Terms

Solving of Differential Equations

Runge-Kutta 4

$$y(t_m + \Delta t) = y_m + \frac{\Delta t}{6}(\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4)$$

$$\alpha_1 = F(t_m, y_m)$$

$$\alpha_2 = F\left(t_m + \frac{\Delta t}{2}, y_m + \frac{\alpha_1 \Delta t}{2}\right)$$

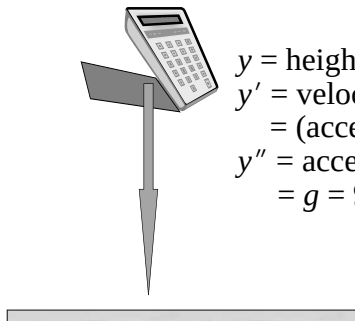
$$\alpha_3 = F\left(t_m + \frac{\Delta t}{2}, y_m + \frac{\alpha_2 \Delta t}{2}\right)$$

$$\alpha_4 = F(t_m + \Delta t, y_m + \alpha_3 \Delta t)$$

First 5 Terms

Simple Model – Falling Calculator

System



y = height

y' = velocity

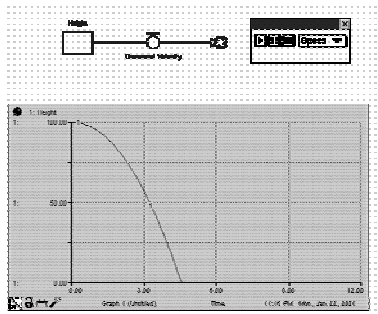
= (acceleration)(time)

y'' = acceleration

= $g = 9.8 \text{ m s}^{-2}$

Simple Model – Falling Calculator

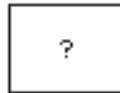
Stella Model



Noname 1



height



☐ height
☒ Reservoir ☐ Conveyor ☐ Queue ☐ Oven

☐ Non-negative
☐ Array

Allowable Inputs

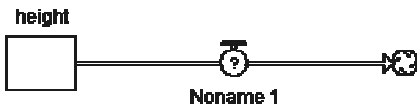
E	[	]	^
7	8	9	÷
4	5	6	/
1	2	3	-
0	.	+	
<			

☐ INITIAL(height) = ...

Units...

100

Document Message... Cancel OK



Downward_Velocity

☒ UNIFLOW ☐ BIFLOW

☐ Unit conversion

Required Inputs

Buildins

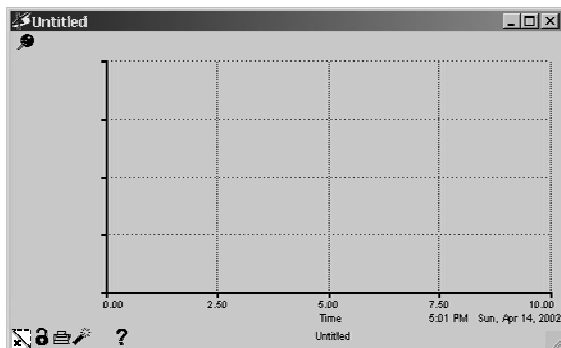
- ABS
- AND
- ARCTAN
- ARRAYMEAN
- ARRAYSTDDEV
- ARRAYSUM

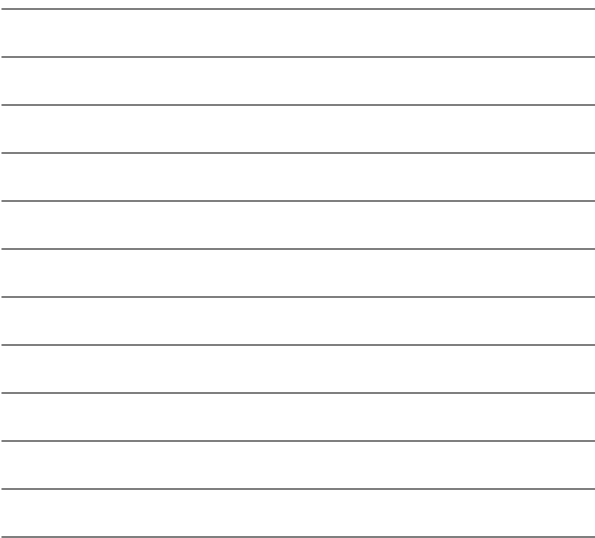
Downward_Velocity = ...

Units...

3.8*time

Become Graphical Function Document Message... Cancel OK



[illegible]

STELLA® 7.0.1

File Edit Equation Run Help

STELLA® Research 7.0.1

height(t) = height(t - dt) + (- Downward_Velocity) * dt
 INIT height = 100
 OUTFLOWS:
 Downward_Velocity = 9.8*time

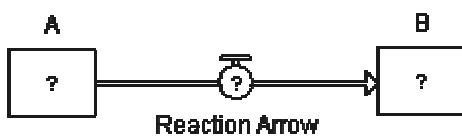
Simple Model – Falling Calculator

Stella Model – Accuracy

- Air Friction
- Bounce

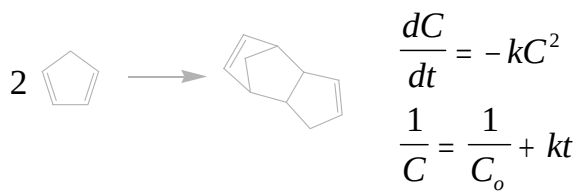
Chemical Kinetics

Sure looked like a natural fit to me!!

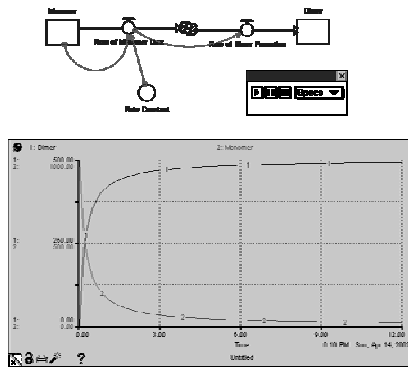


Kinetics Model

Dimerization of Cyclopentadiene
2nd Order Diels-Alder Reaction



Kinetics Model



Monomer

?

☐ Monomer
☒ Reservoir
☐ Conveyor
☐ Queue
☐ Oven

☒ Non-negative
☐ Array

Allowable Inputs

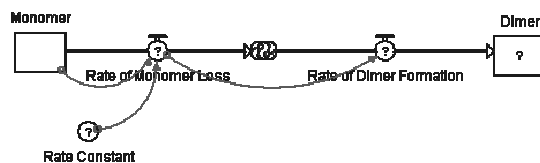
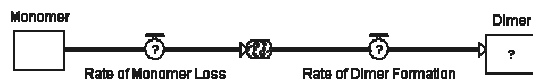
E	(	)	~
7	8	9	*
4	5	6	/
1	2	3	-
0	.	+	
<			

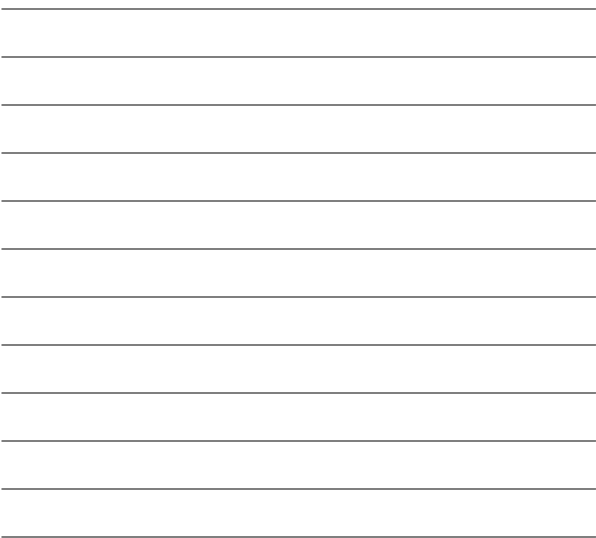
☐ INITIAL(Monomer) = ...

1000

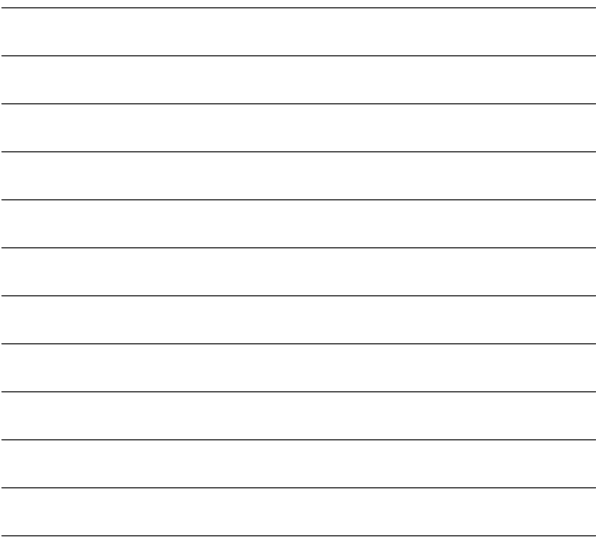
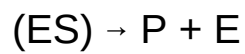
Units...

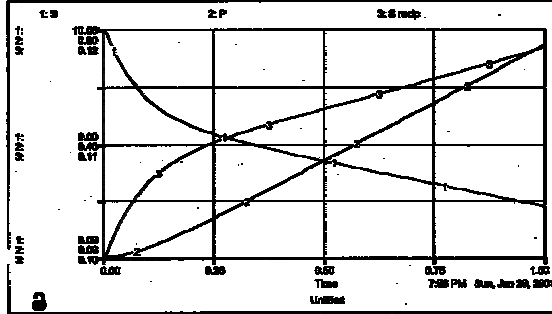
Document Message... Cancel OK





Len Soltzberg (Simmons College)





☐ $E(t) = E(t - dt) + (dEdt) * dt$
 INIT E = 1
 INFLOWS:
 $\Rightarrow dEdt = -k1 * E * S + km1 * ES + k2 * ES$
☒ $ES(t) = ES(t - dt) + (dESdt) * dt$
 INIT ES = 0
 INFLOWS:
 $\Rightarrow dESdt = -k2 * ES - km1 * ES + k1 * E * S$
☐ $P(t) = P(t - dt) + (dPdt) * dt$
 INIT P = 0.0001
 INFLOWS:
 $\Rightarrow dPdt = k2 * ES$
☐ $S(t) = S(t - dt) + (dSdt) * dt$
 INIT S = 10
 INFLOWS:
 $\Rightarrow dSdt = -k1 * E * S + km1 * ES$
☐ $k1 = 1$
☐ $k2 = 1$
☐ $km1 = 1$
☐ $\ln P = \text{LOGN}(P)$
☐ $S_recp = 1/S$

Oscillating Reactions

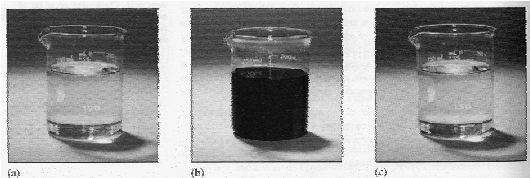
Criteria

- Two or more coupled reactions
- Autocatalytic

Oscillating Reactions

Briggs-Rauscher

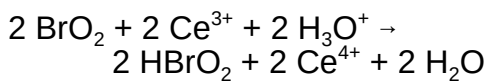
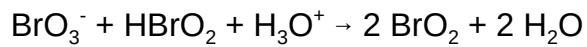
- H_2O_2
- KIO_3 and H_2SO_4
- $\text{HOOCCH}_2\text{COOH}$, MnSO_4 , starch



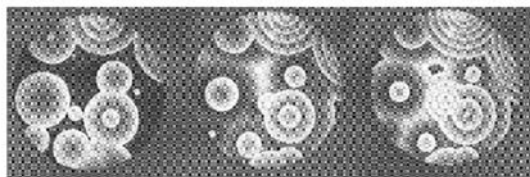
~15 s for each cycle

Oscillating Reactions

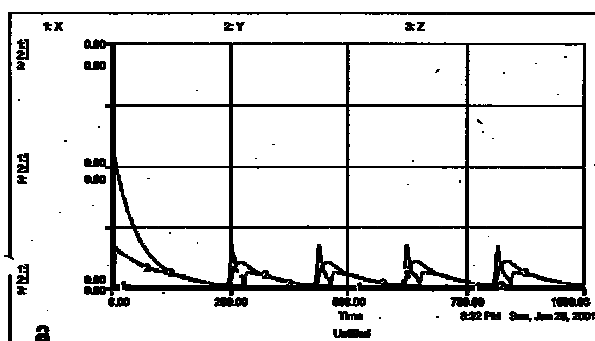
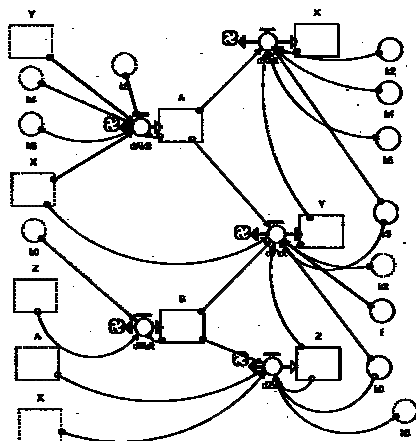
Belousov-Zhabotinskii



- 18 reversible steps
- 21 different chemical species



A = BrO_3^-
X = HBrO_2
Z = Ce^{4+}
P = HOBr
B = organic
Y = Br^-



☐ A) $A() = A() - d() + (dA()) * d()$
INIT A = 0.05
INFLUENCE:
 $\Rightarrow dA() = -3^2 A^2 Y + 15^2 A^2 X + 1^2 X^2$

☐ B) $B() = B() - d() + (dB()) * d()$
INIT B = 0.02
INFLUENCE:
 $\Rightarrow dB() = -10^2 B^2 Z$

☐ C) $C() = X() - d() + (dC()) * d()$
INIT X = 0.0001
INFLUENCE:
 $\Rightarrow dC() = 12^2 X^2 Y + 12^2 X^2 Y + 16^2 A^2 X^2 + 2^2 B^2 X^2 Z$

☐ D) $Y() = Y() - d() + (dY()) * d()$
INIT Y = 0.0001
INFLUENCE:
 $\Rightarrow dY() = -13^2 A^2 X^2 Y + 0.5^2 W^2 B^2 Z$

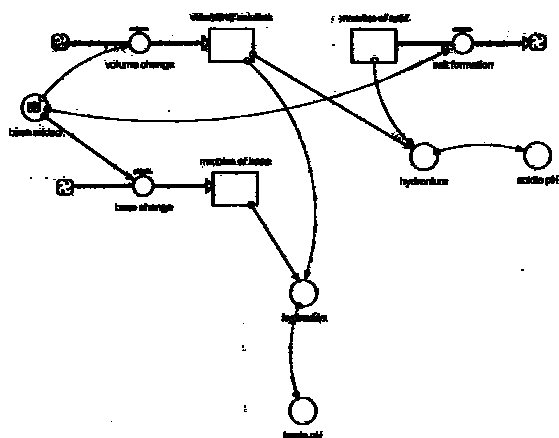
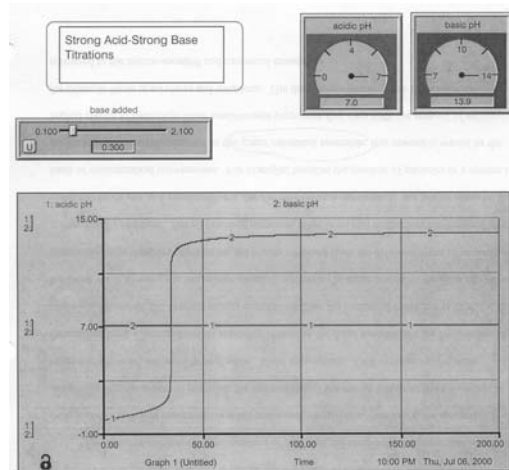
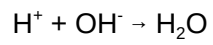
☐ E) $Z() = Z() - d() + (dZ()) * d()$
INIT Z = 0.001
INFLUENCE:
 $\Rightarrow dZ() = 2^2 A^2 X^2 - 10^2 B^2 Z$

☐ f = 1
☐ 10 = 1
☐ 12 = 0.5
☐ 13 = 1.28
☐ 14 = 2e3
☐ 15 = 8.0

Other Applications

Rod Schluter (Formerly of CofC)

Acid/Base Equilibrium



```

❑ moles_of_acid(t) = moles_of_acid(-dt) + (-salt_formation)*dt
INIT moles_of_acid = 10
OUTFLOW:
-> salt_formation = base_added
❑ moles_of_base(t) = moles_of_base(-dt) + (base_change)*dt
INIT moles_of_base = -moles_of_acid
INFLOW:
-> base_change = base_added
❑ volume_of_solution(t) = volume_of_solution(-dt) + (volume_change)*dt
INIT volume_of_solution = 10
INFLOW:
-> volume_change = base_added
⊙ acidic_pH = -LOG10(hydrogenium)
⊙ base_added = .1
⊙ basic_pH = 14-(-LOG10(hydroxide))
⊙ hydrogenium =
  IF(moles_of_acid/volume_of_solution>1.1e-7)THEN(moles_of_acid/volume_of_solution)ELSE(1e-7)
⊙ hydroxide =
  IF(moles_of_base/volume_of_solution>1.1e-7)THEN(moles_of_base/volume_of_solution)ELSE(1e-7)

```

Other Applications

Debye Theory for Monatomic Crystals

$$D(\theta_D / T) = \frac{3}{u^3} \int \frac{x^3}{e^x - 1} dx$$

$$U - U_0 = 3RTD$$

$$C_v = 3R \left(4D - 3 \frac{u}{e^u - 1} \right)$$

$$S = 3R \left[\frac{4}{3} D - \ln(1 - e^{-u}) \right]$$

$$A - U_0 = RT \left[3 \ln(1 - e^{-u}) - D \right]$$