

# FROM PETRI NETS TO PARTIAL DIFFERENTIAL EQUATIONS

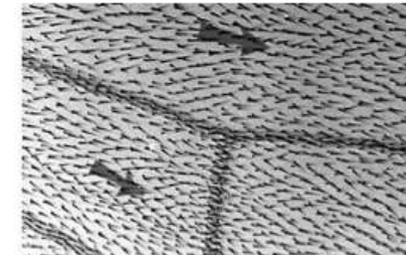
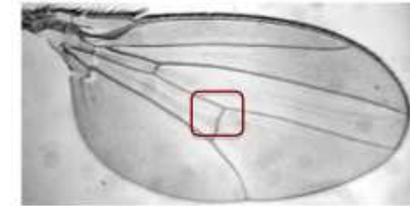
- A PETRI NET PERSPECTIVE ON SYSTEMS AND SYNTHETIC BIOLOGY -

MONIKA HEINER

BRANDENBURG TECHNICAL UNIVERSITY COTTBUS-SENFTEMBERG  
COMPUTER SCIENCE INSTITUTE

## □ FRAMEWORK

- > *unifying four paradigms:*  
QPN - SPN - CPN - HPN
- > *our toolbox :*  
Snoopy - Marcie - Charlie - Patty - S4

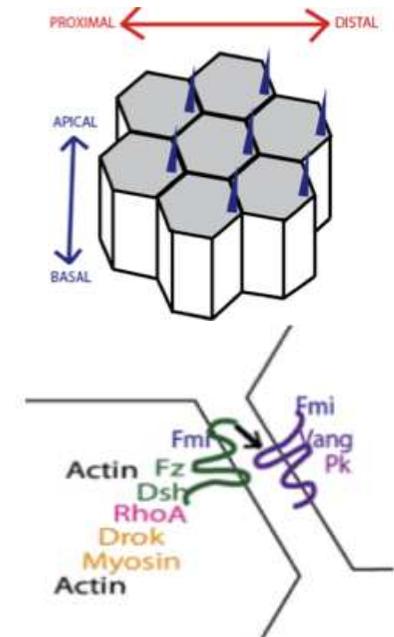


## □ MODELLING BIO PETRI NETS

- > *composition from standard components*
- > *bottom-up (reverse engineering)*
- > *genome-controlled model generation*

## □ COLOURED PETRI NETS

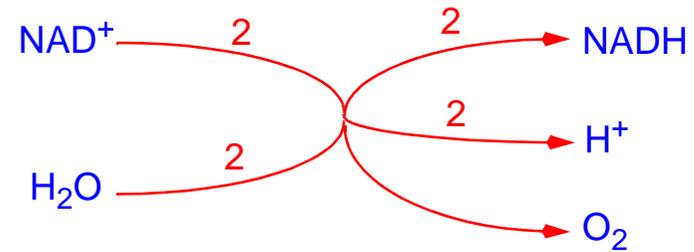
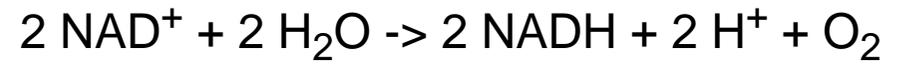
- > *colouring space -> PDE*
- > *Turing patterns*
- > *planar cell polarity*



# **THE PETRI NET FRAMEWORK**

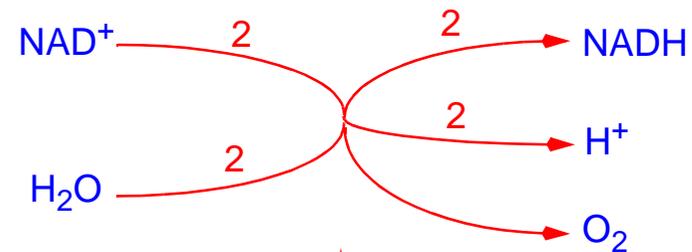
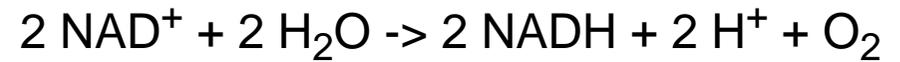
...

**ARE NETWORKS  
OF BIOCHEMICAL  
REACTIONS**

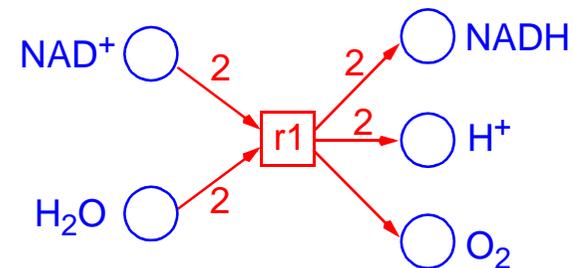


...  
**ARE NETWORKS  
 OF BIOCHEMICAL  
 REACTIONS**

...  
**NATURALLY  
 EXPRESSIBLE AS  
 PETRI NETS**



**hyper-arcs**

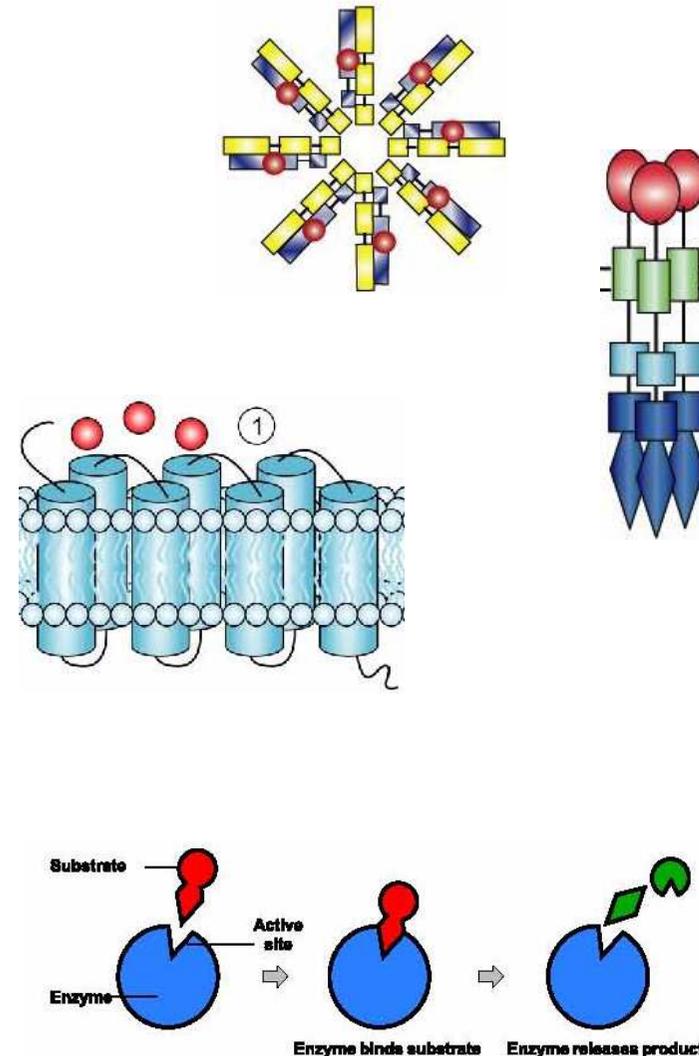


## □ places → model variables

- > (bio-) chemical compounds
  - > proteins
  - > protein conformations
  - > complexes
  - > genes, . . . , etc.
- . . . in different locations*

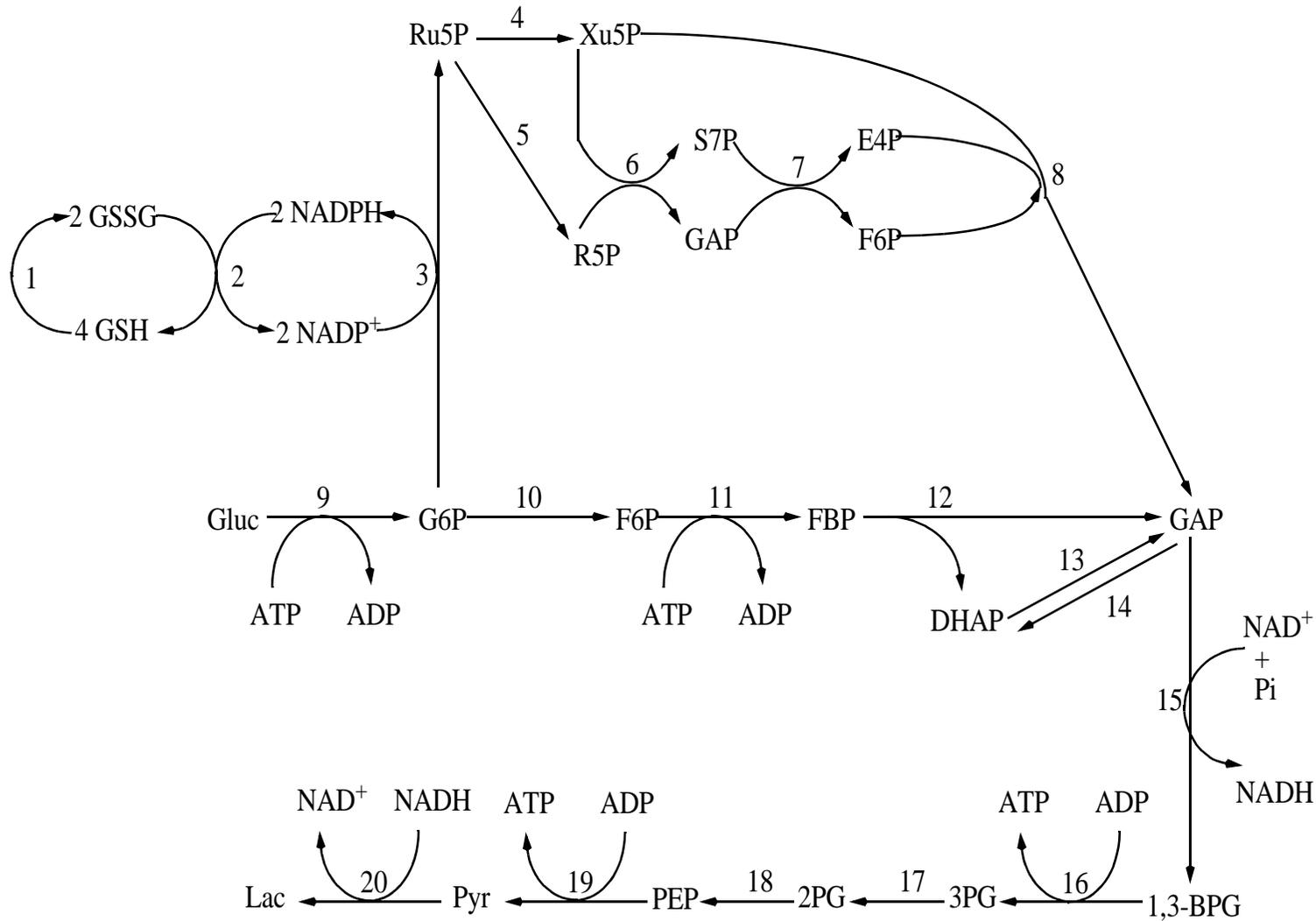
## □ transitions → atomic events

- > (stoichiometric) chemical reaction
  - > complexation / decomplexation
  - > phosphorylation / dephosphorylation
  - > conformational change
  - > transport step, . . . , etc.
- . . . in different locations*



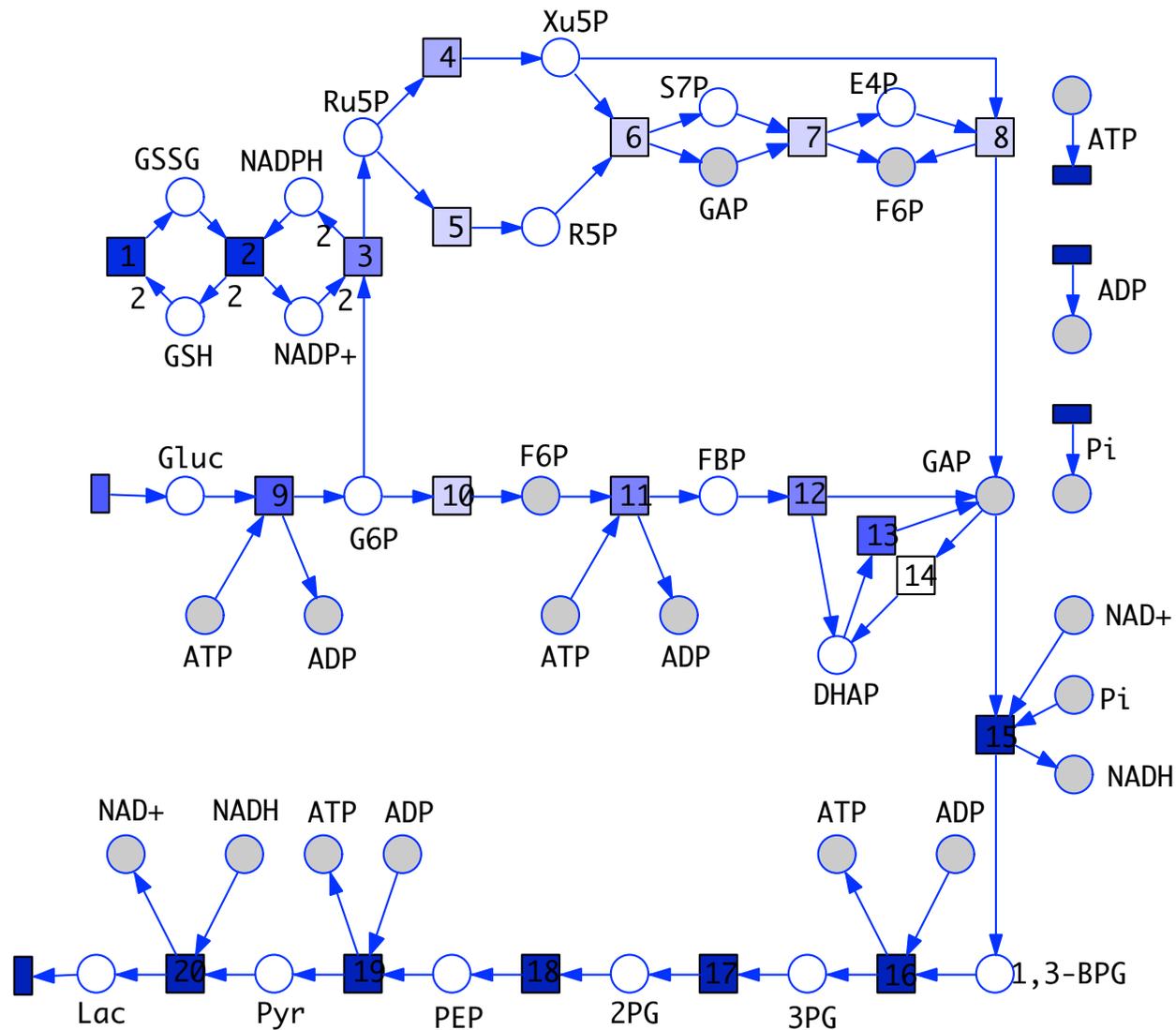
# Ex1 - Glycolysis and Pentose Phosphate Pathway

[Reddy 1993]

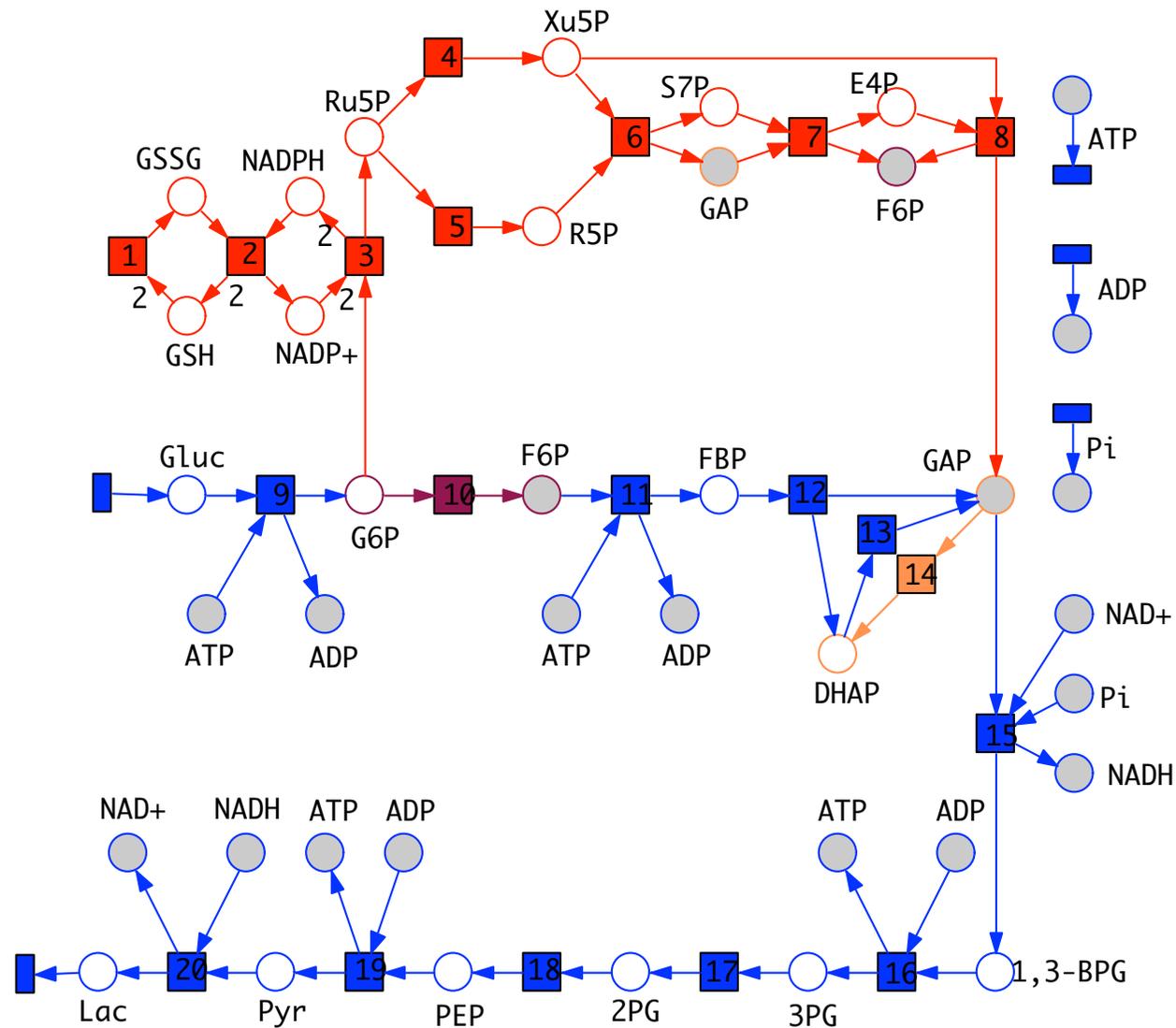


# Ex1 - Glycolysis and Pentose Phosphate Pathway

[Reddy 1993]  
[Heiner 1998]

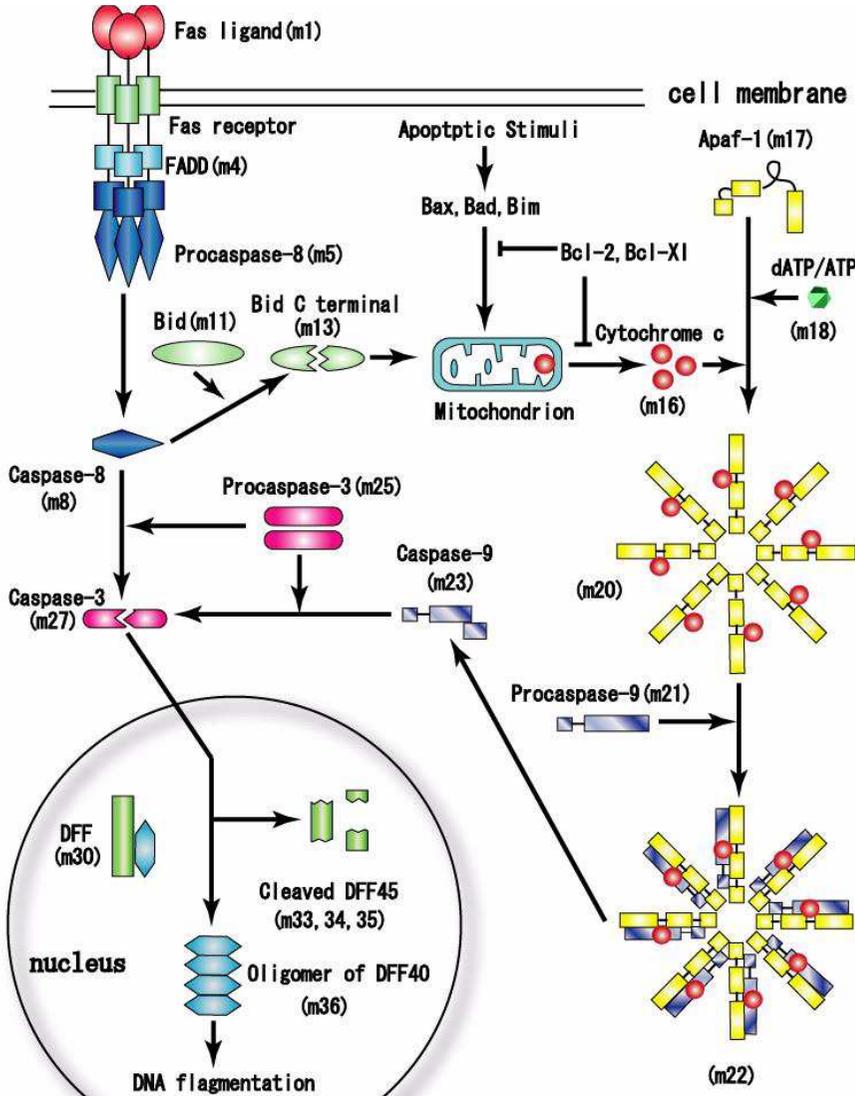


# Ex1 - Glycolysis and Pentose Phosphate Pathway

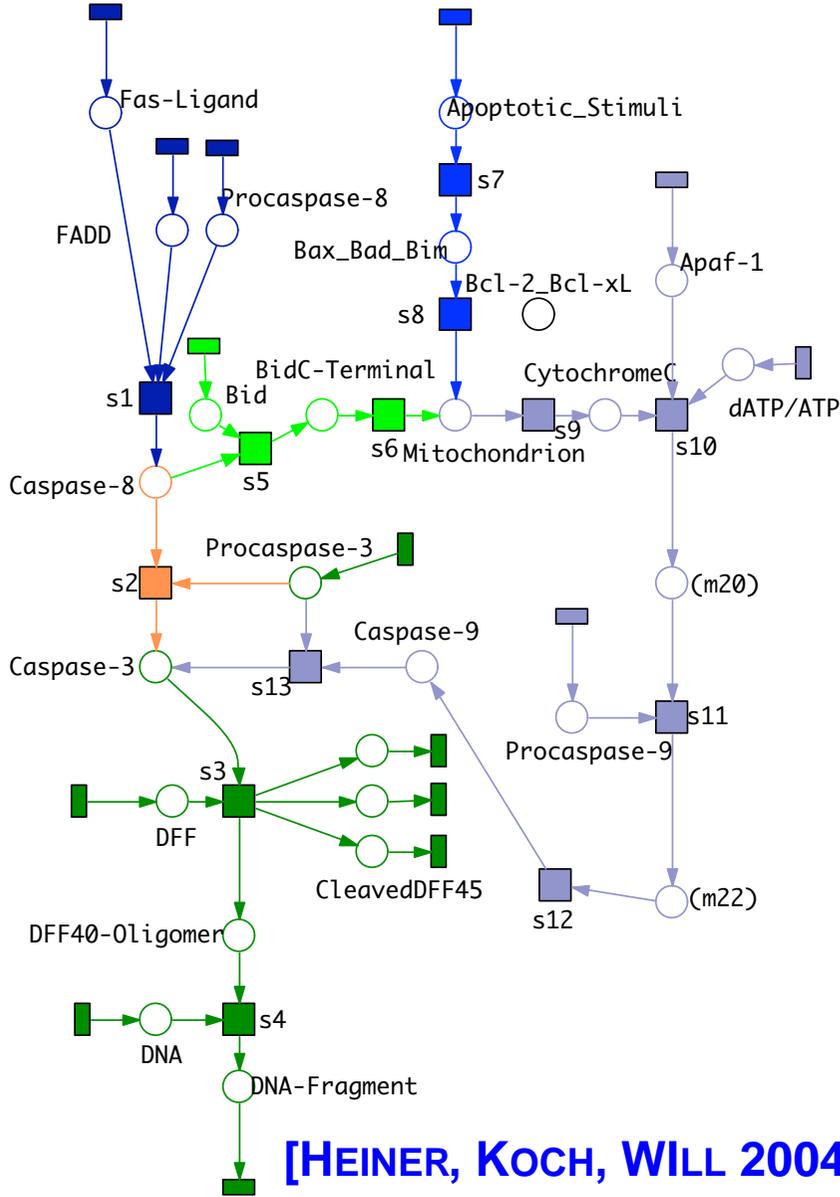


[Reddy 1993]  
[Heiner 1998]  
[Heiner 2009]

# Ex2 - APOPTOSIS IN MAMMALIAN CELLS

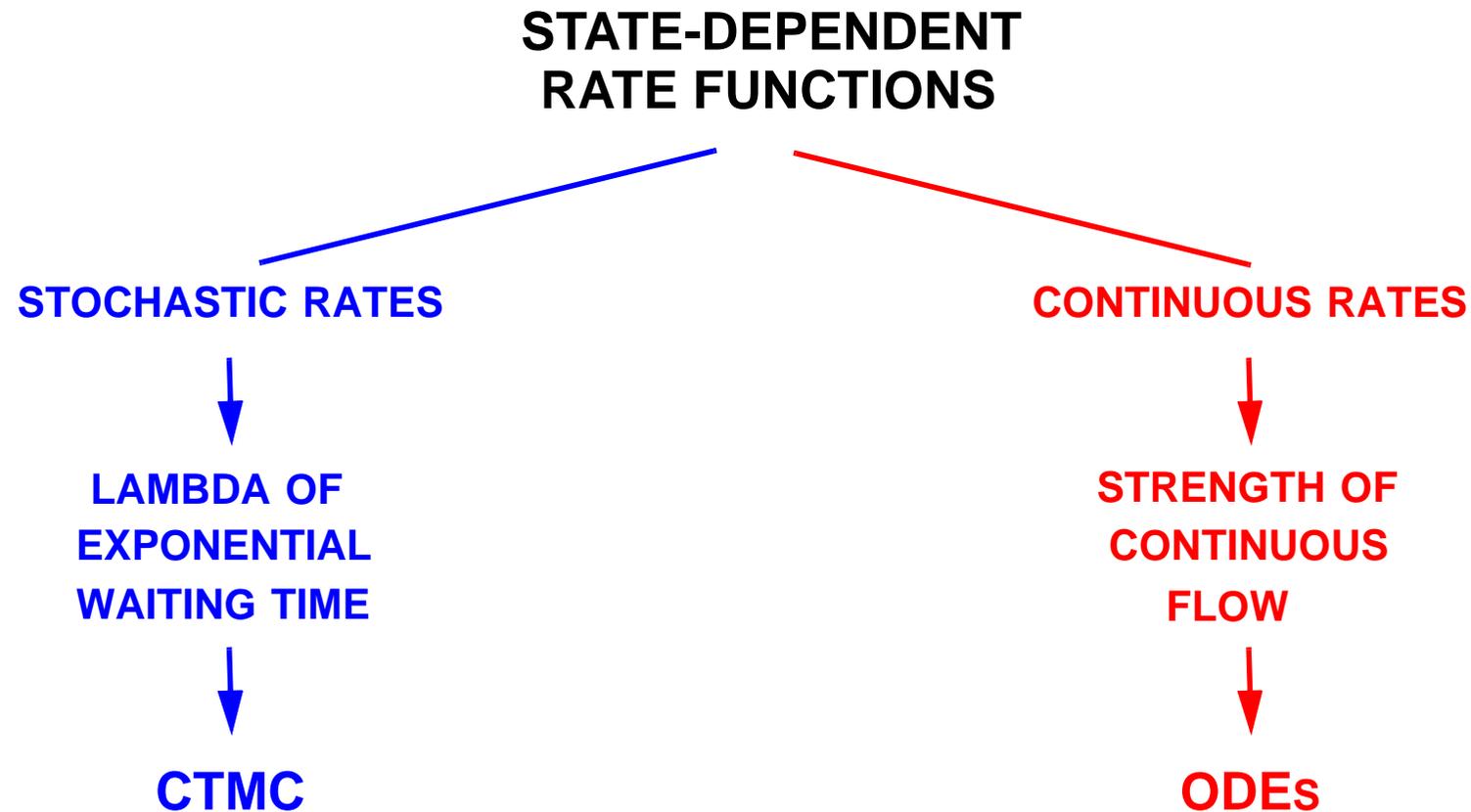


[GON 2003]



[HEINER, KOCH, WILL 2004]

## STATE-DEPENDENT RATE FUNCTIONS

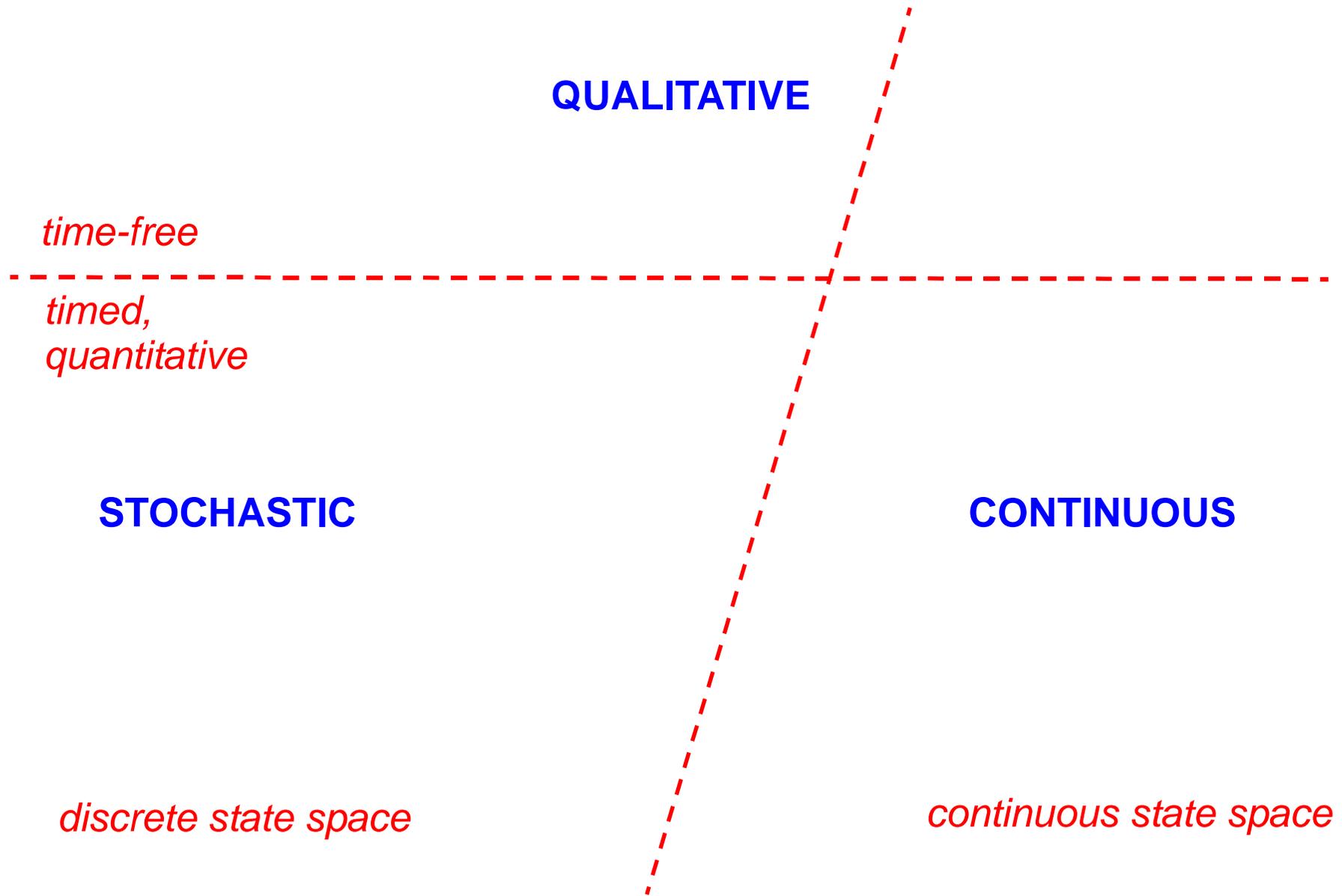


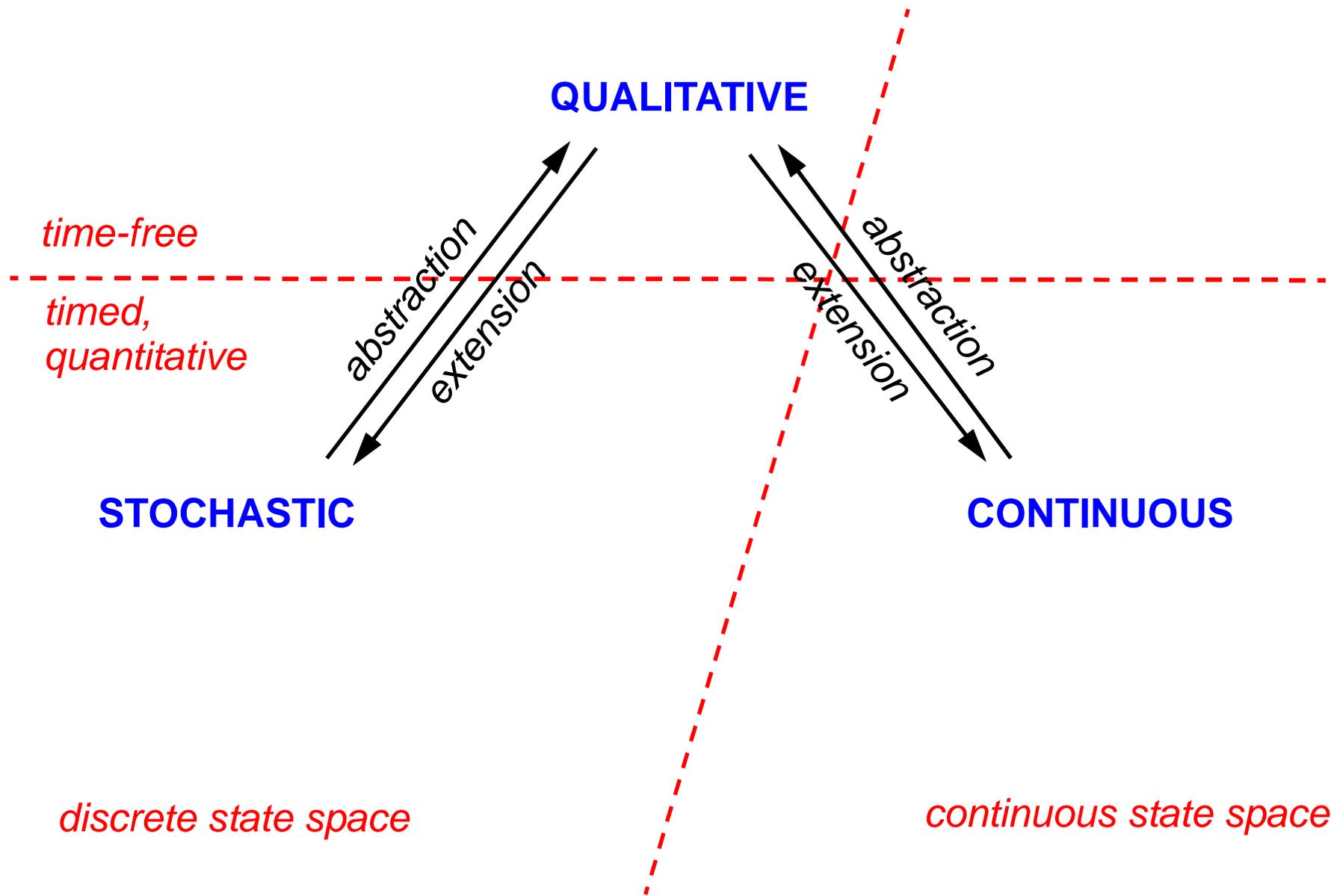
-> supported by, e.g., COPASI, Dizzy, ..., Snoopy

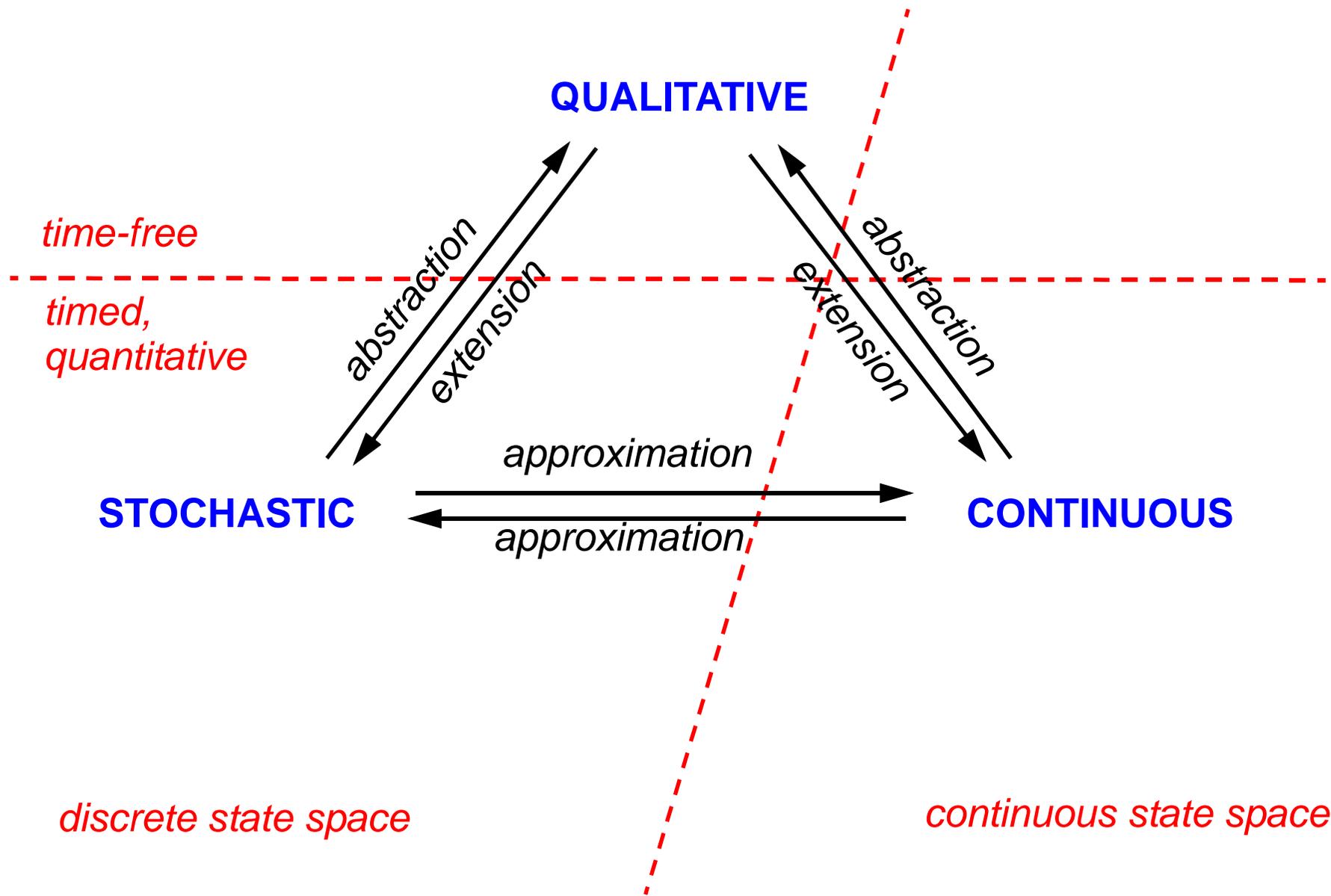
**QUALITATIVE**

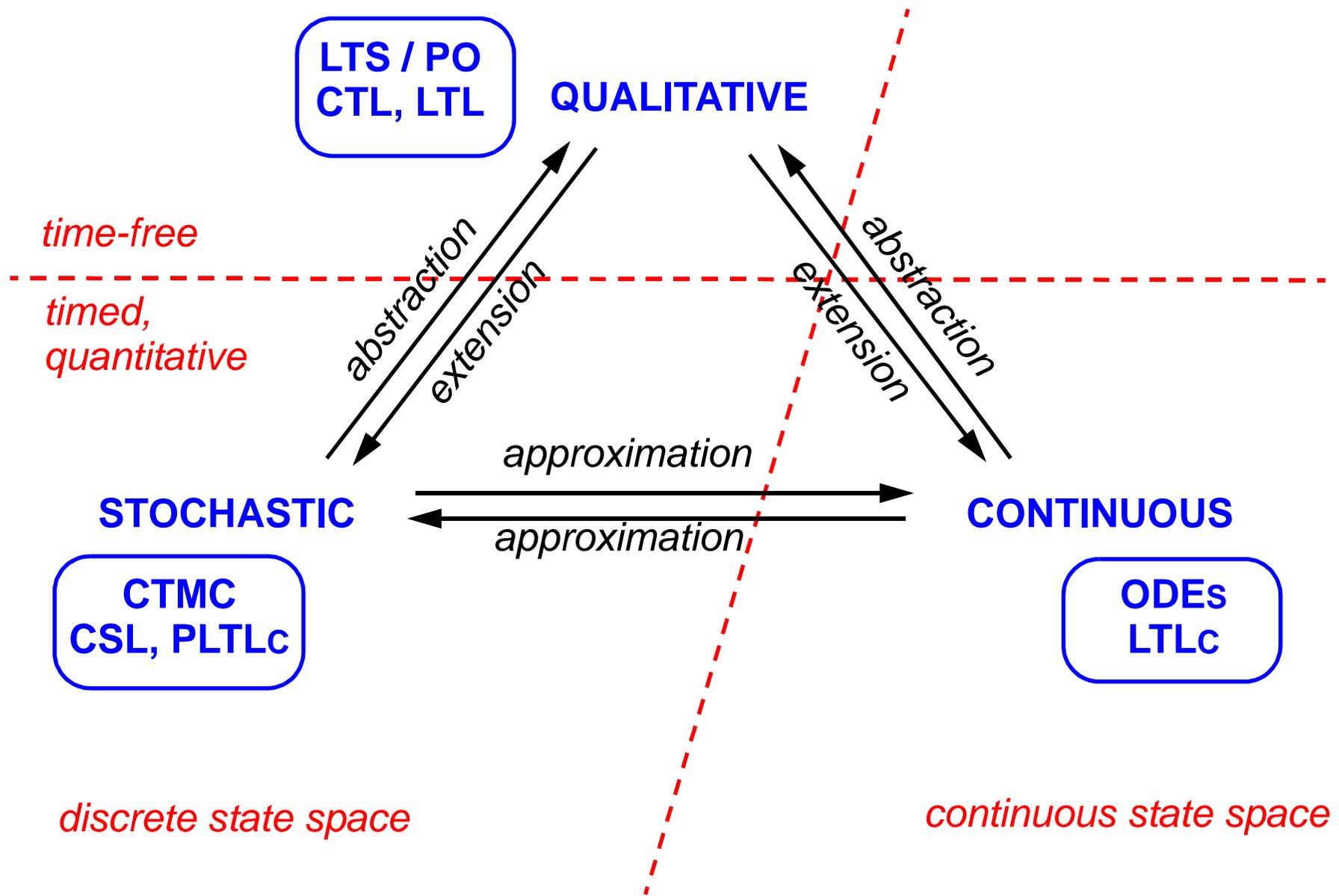
**STOCHASTIC**

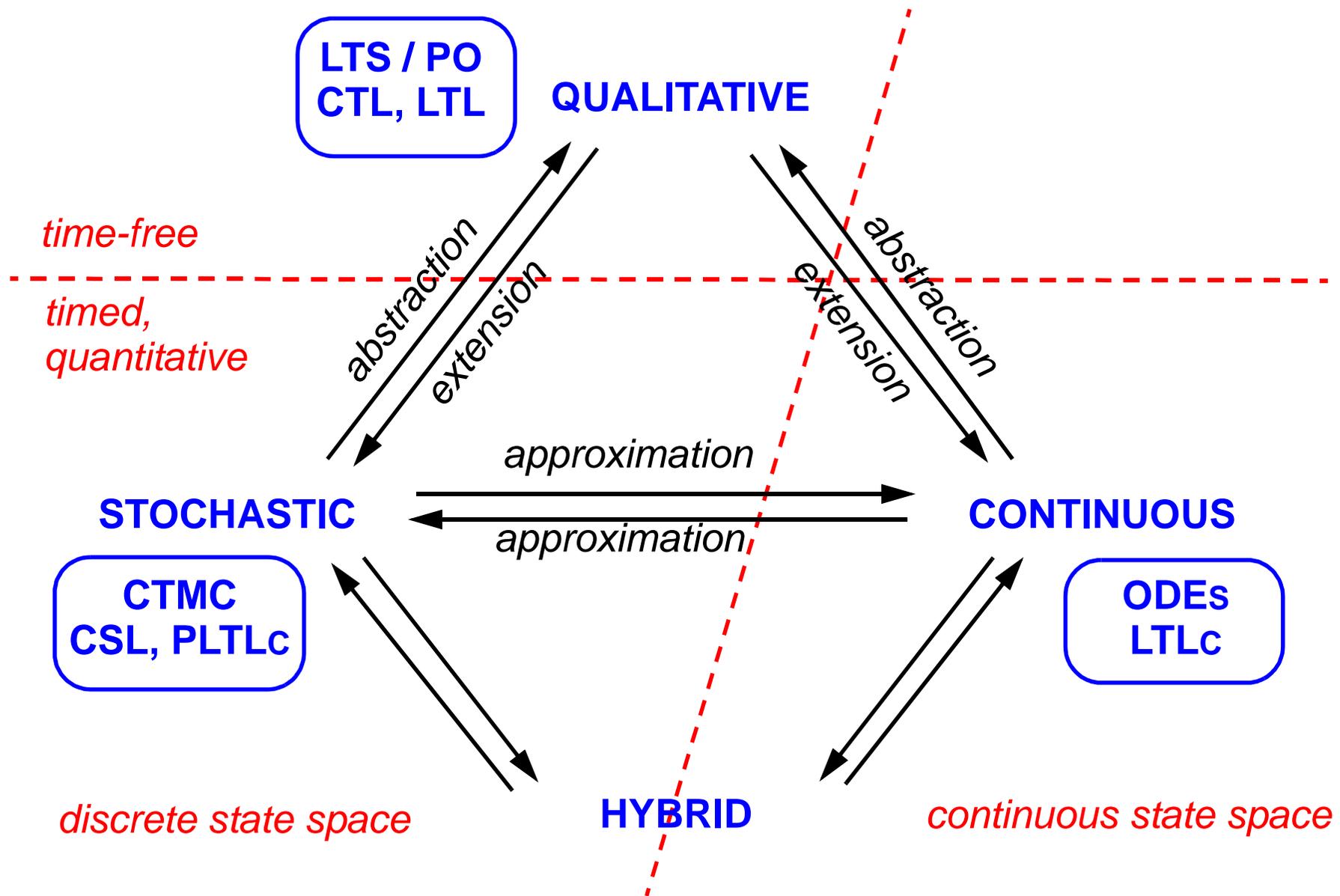
**CONTINUOUS**

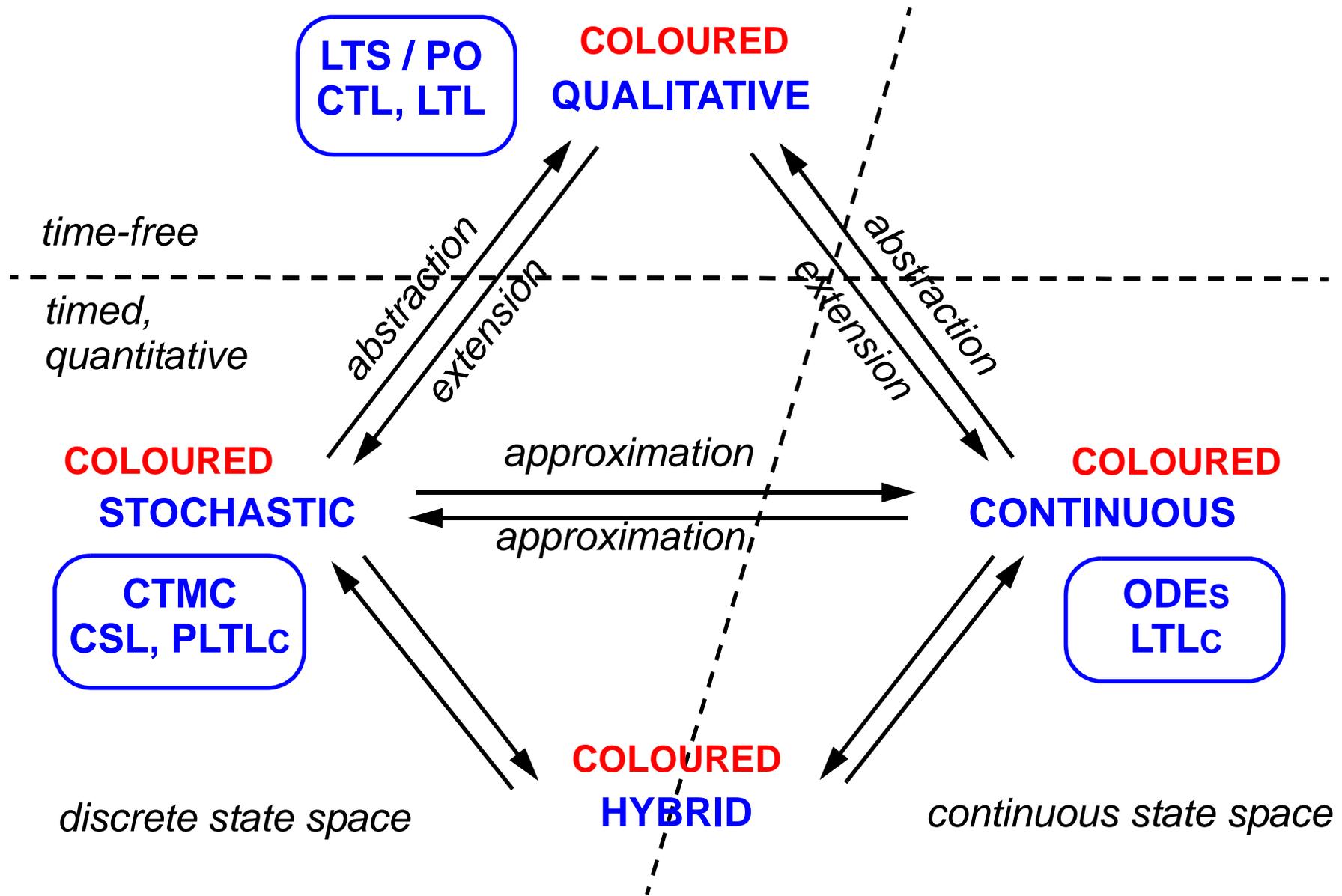






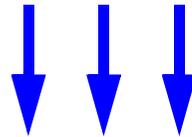




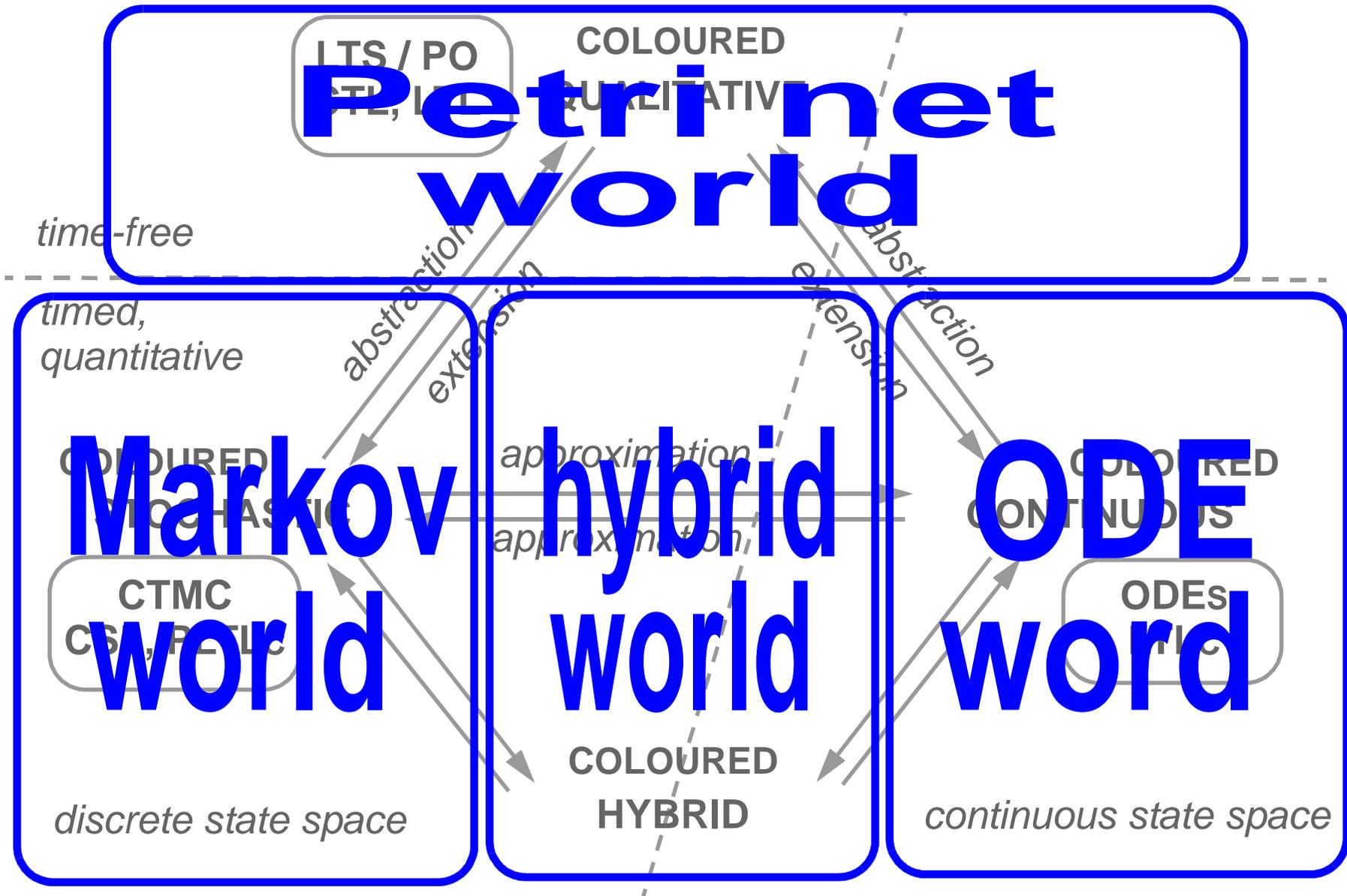


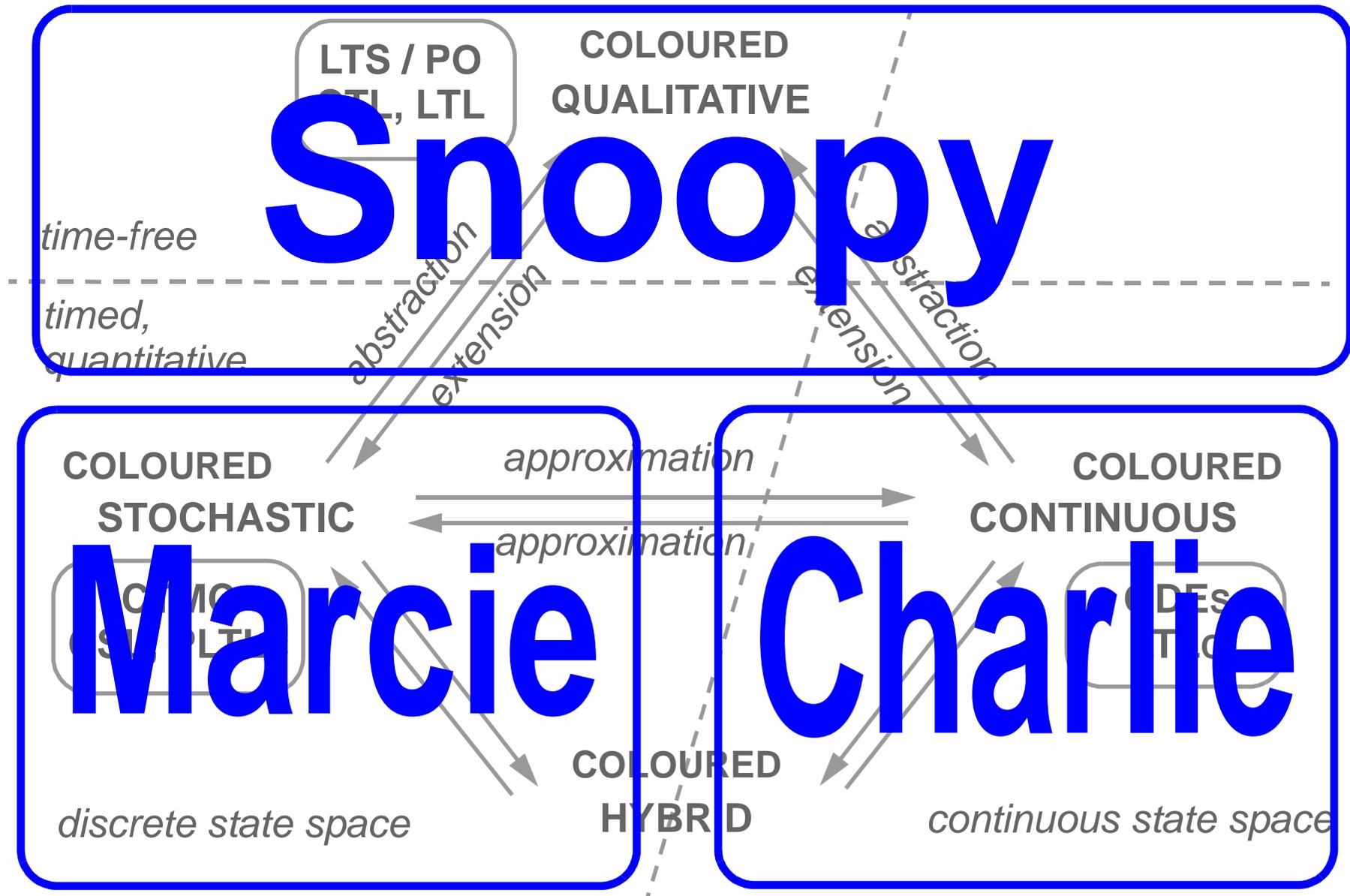
# 4x2

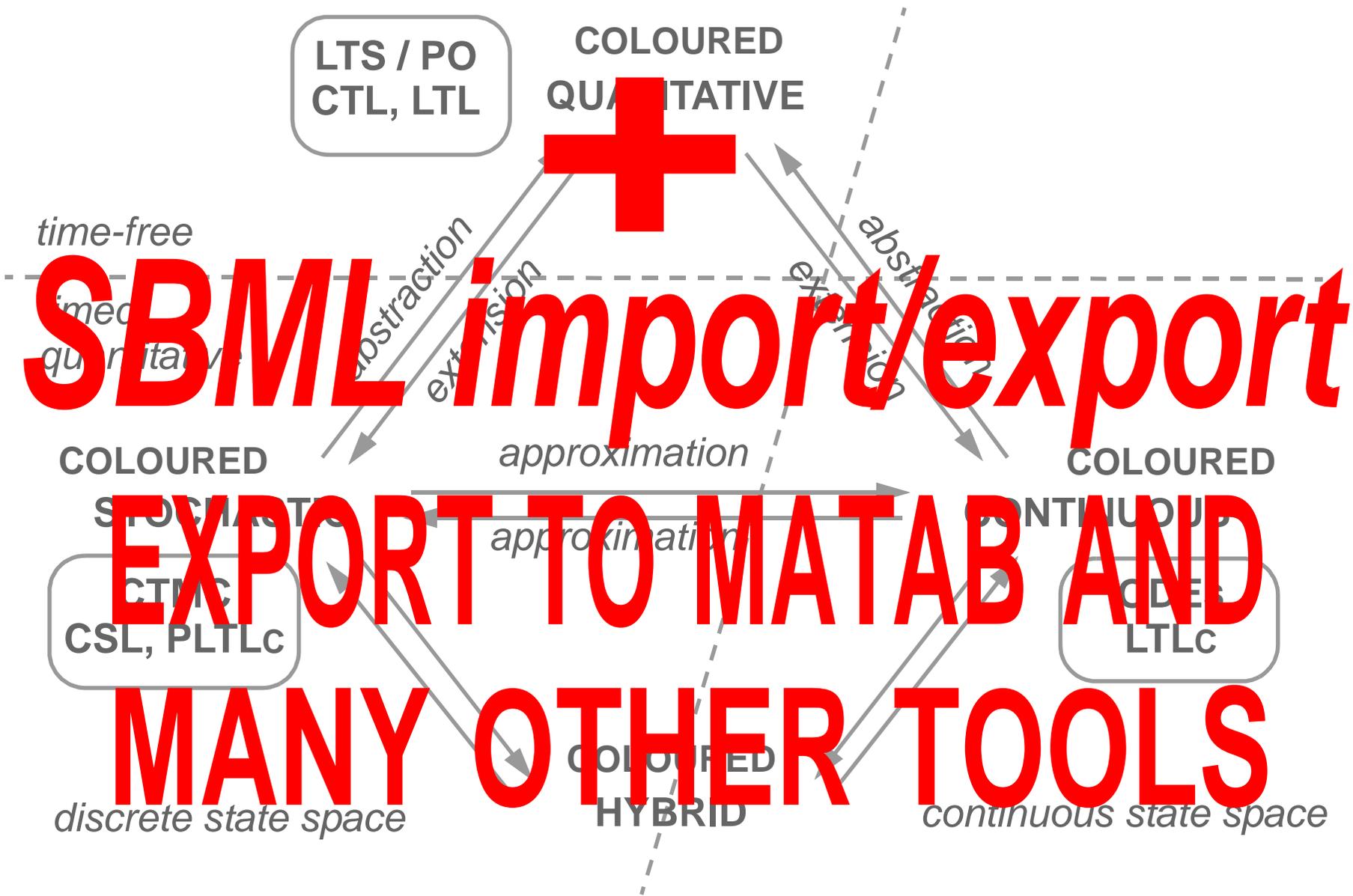
MODELS SHARING STRUCTURE



**QUANTITATIVE MODEL = QUALITATIVE MODEL  
+  
RATE FUNCTIONS  
(KINETICS)**

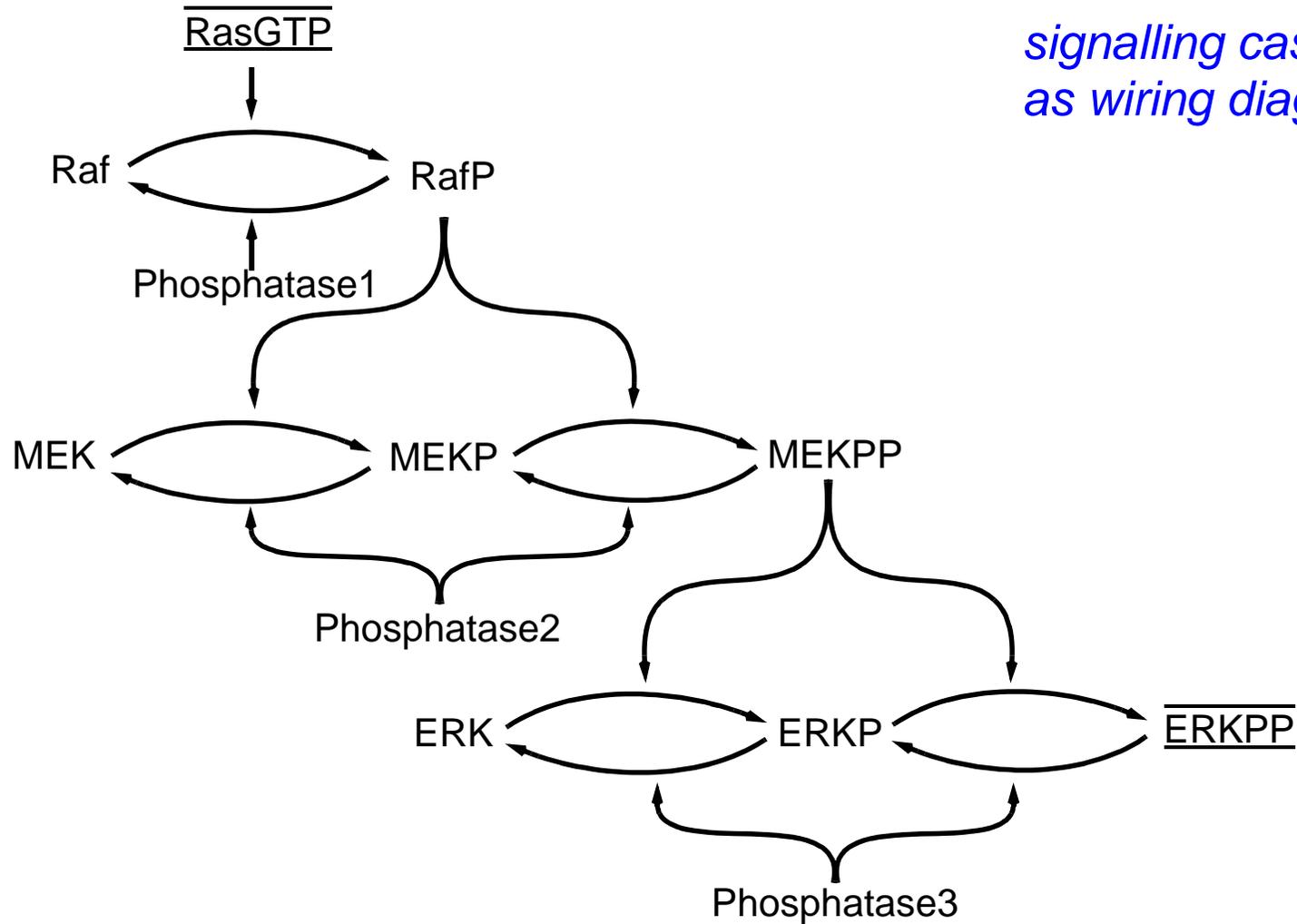






# MODELLING BIO (PETRI) NETS

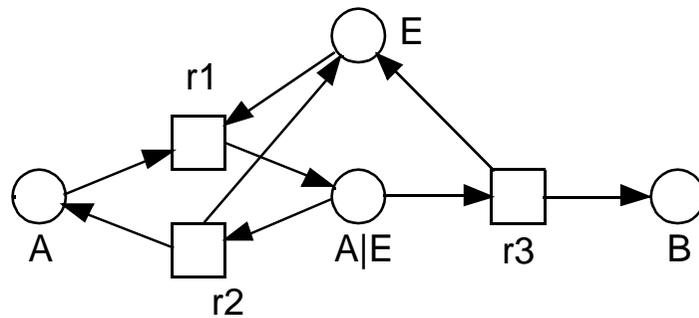
# **APPROACH 1**



*signalling cascade  
as wiring diagram*

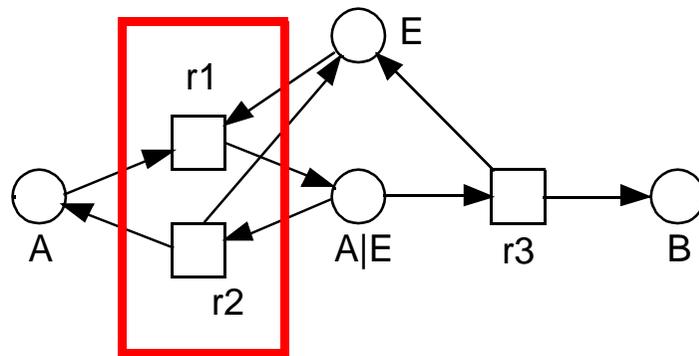


*enzymatic reaction,  
mass-action kinetics*



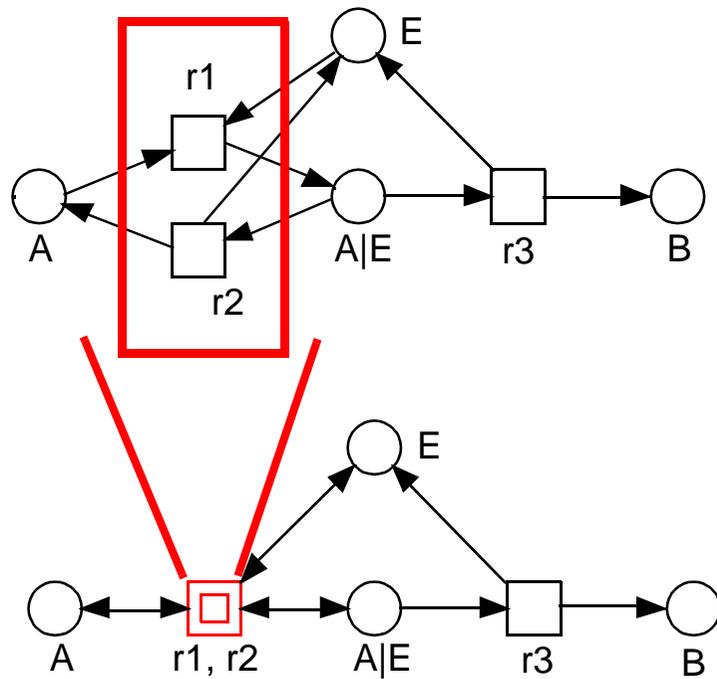


*enzymatic reaction,  
mass-action kinetics*

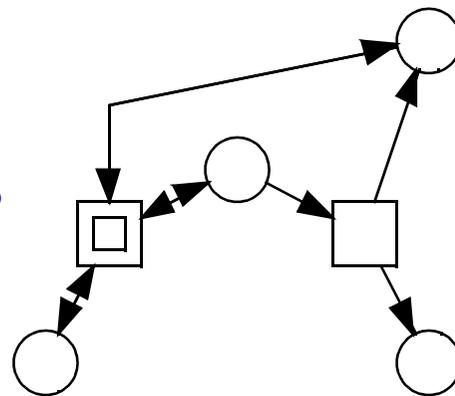


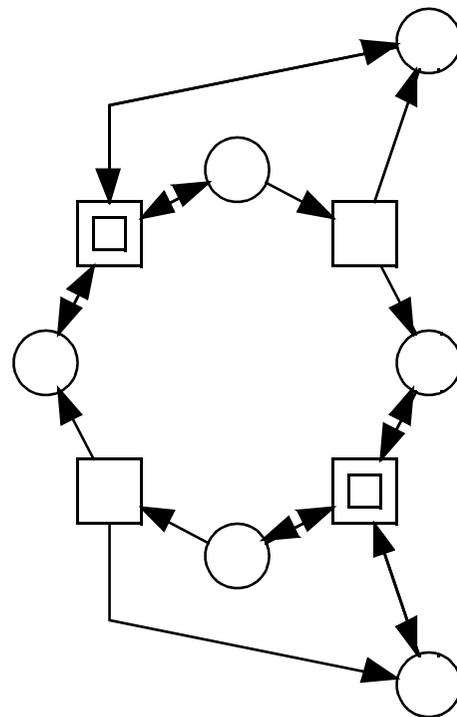


*enzymatic reaction,  
mass-action kinetics*



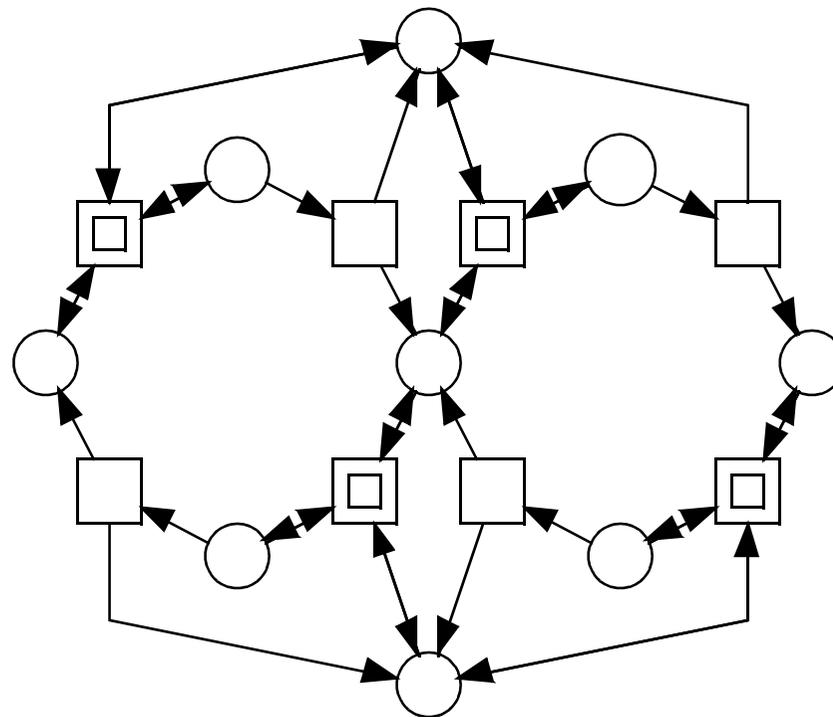
SINGLE  
MASS-ACTION STEP



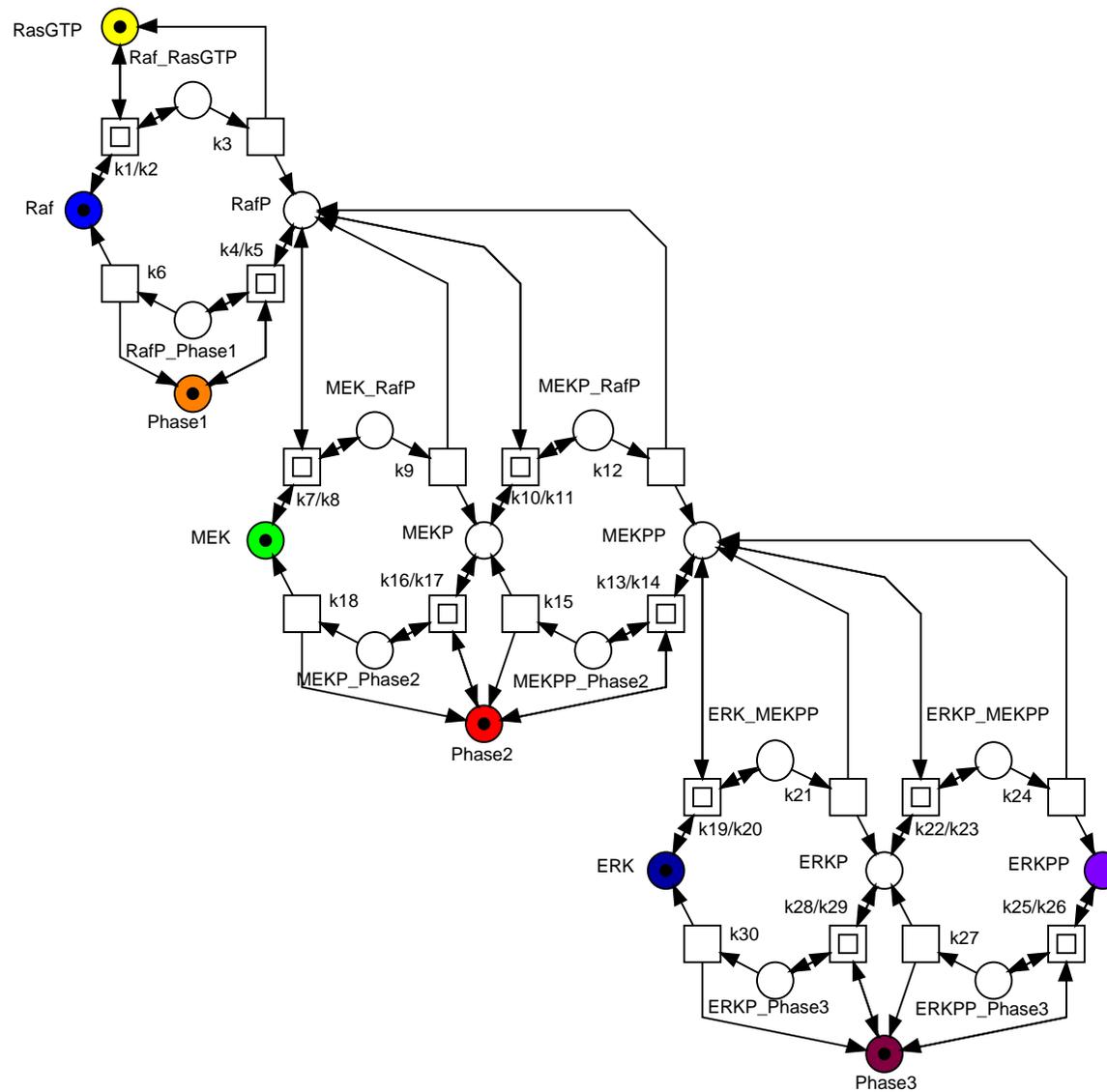


*SINGLE  
PHOSPHORYLATION / DEPHOSPHORYLATION*

## DOUBLE PHOSPHORYLATION / DEPHOSPHORYLATION

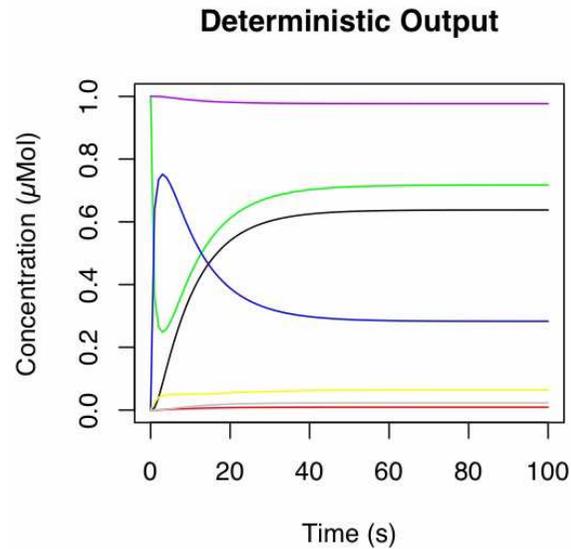
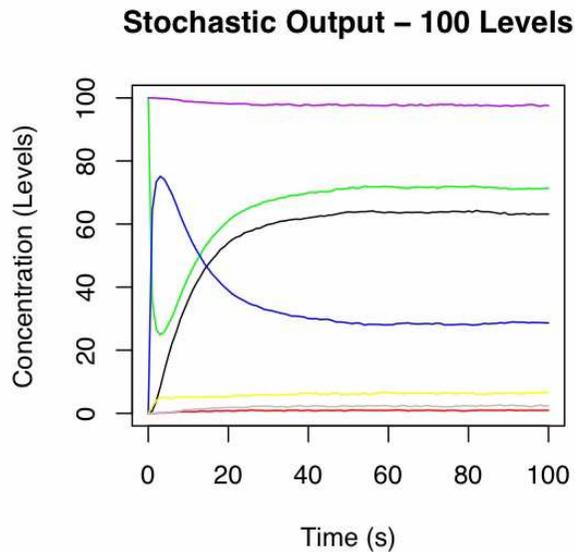
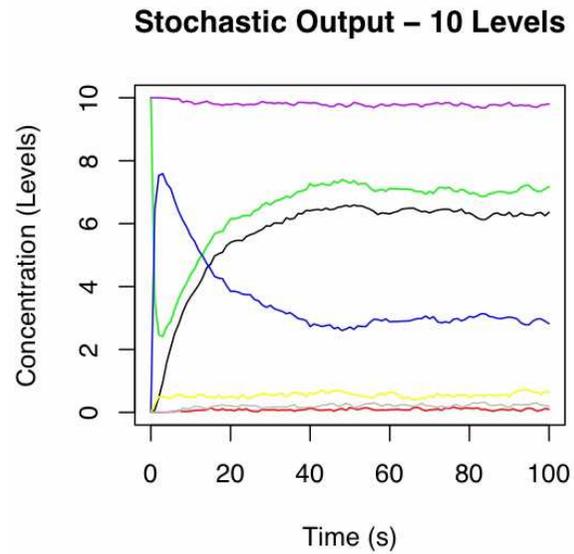
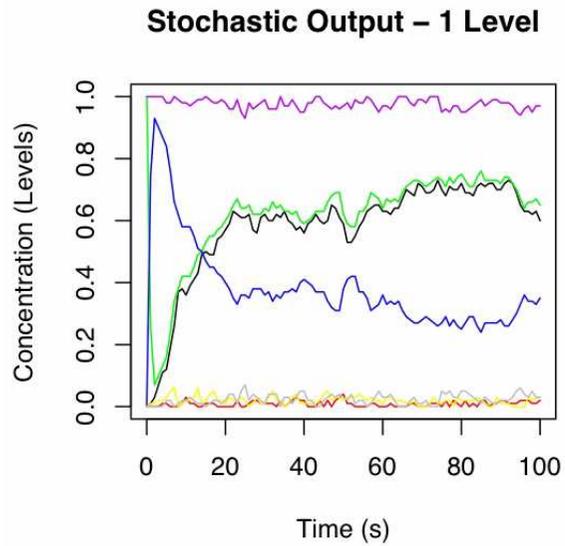


# SIGNALLING CASCADE AS PETRI NET



[GILBERT,  
HEINER,  
LEHRACK 2007]

[HEINER,  
GILBERT,  
DONALDSON 2008]



*signalling cascade  
as wiring diagram*

**[GILBERT,  
HEINER,  
LEHRACK 2007]**

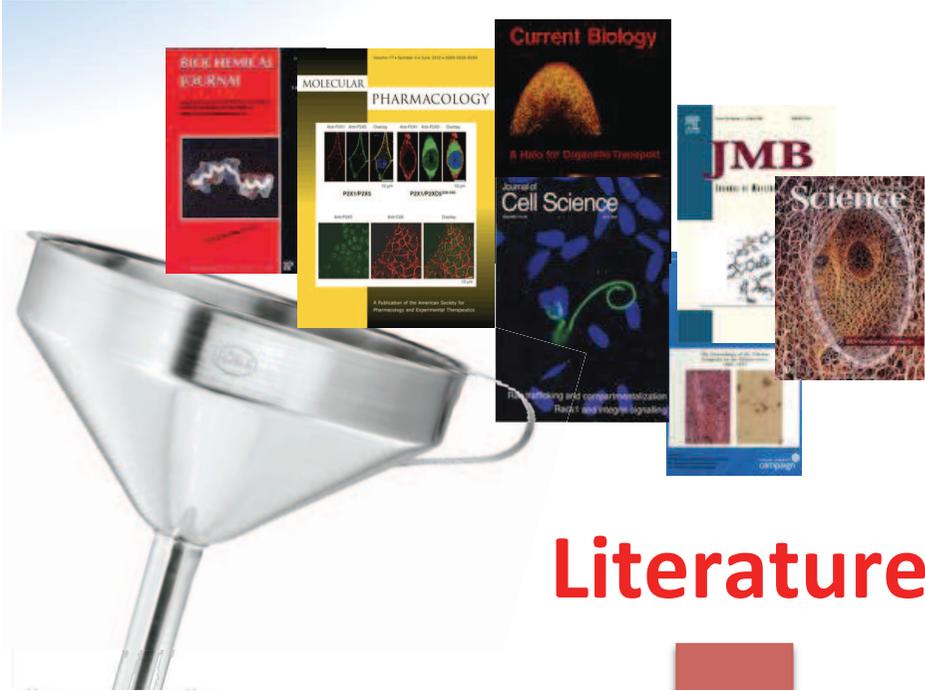
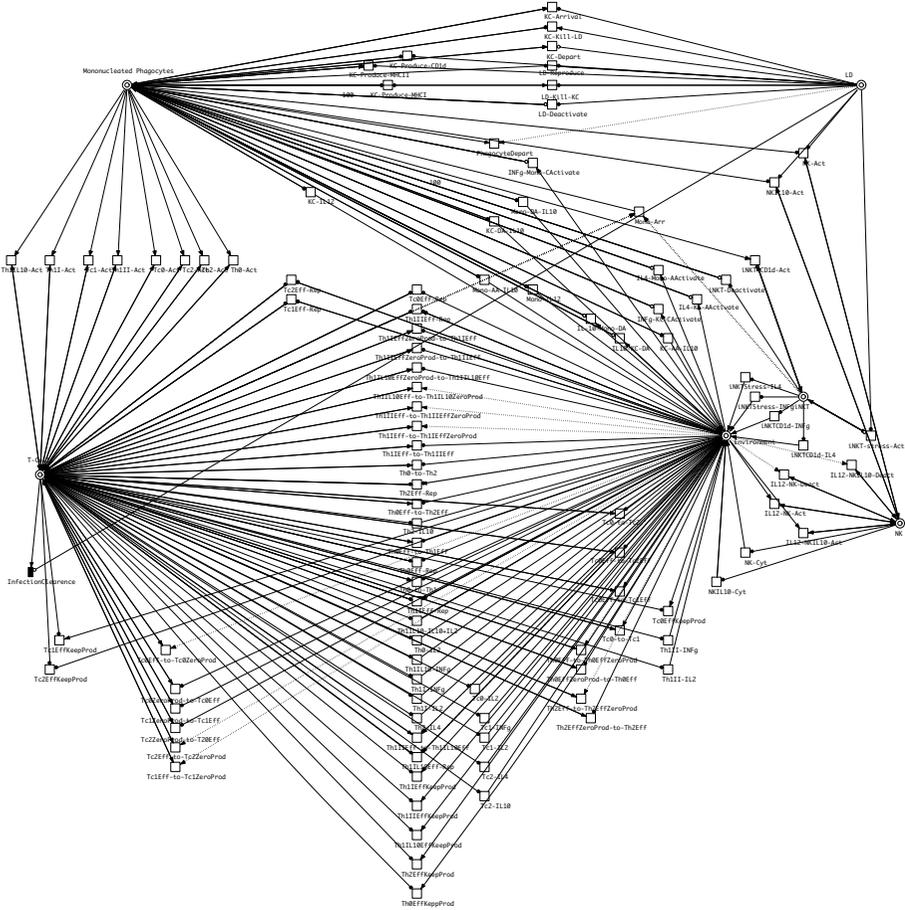
**[HEINER,  
GILBERT,  
DONALDSON 2008]**

# APPROACH 2

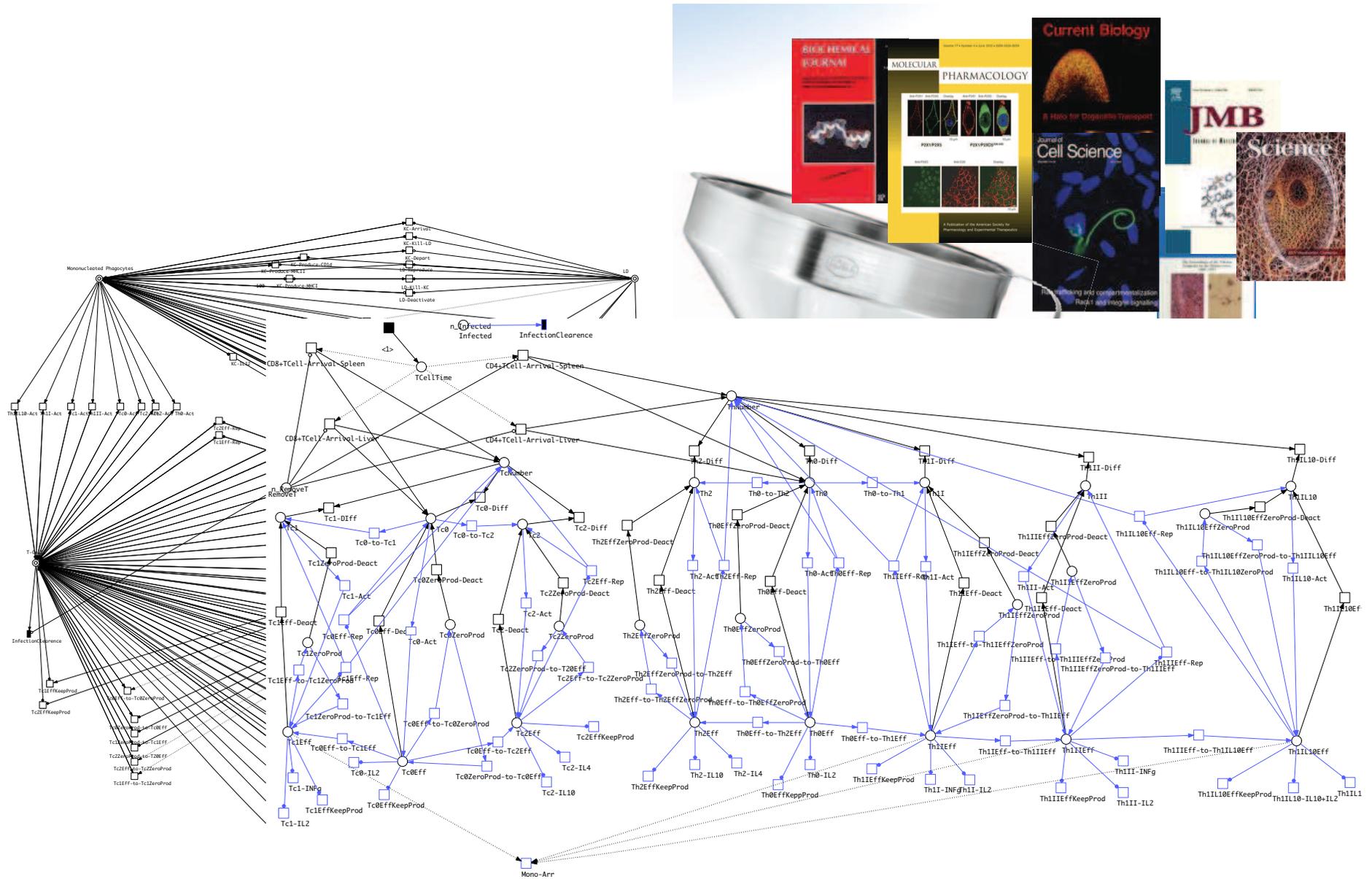


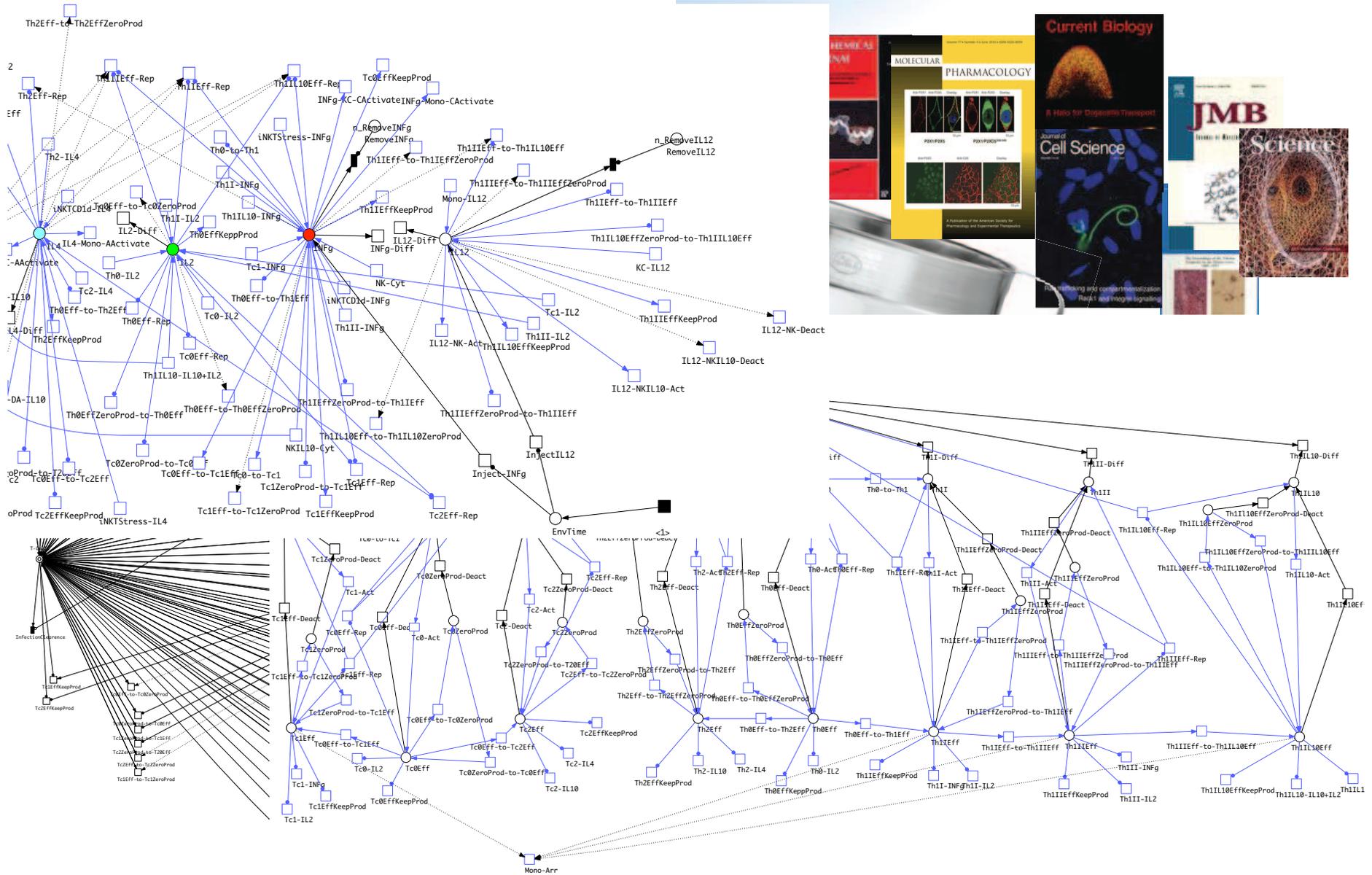
Literature





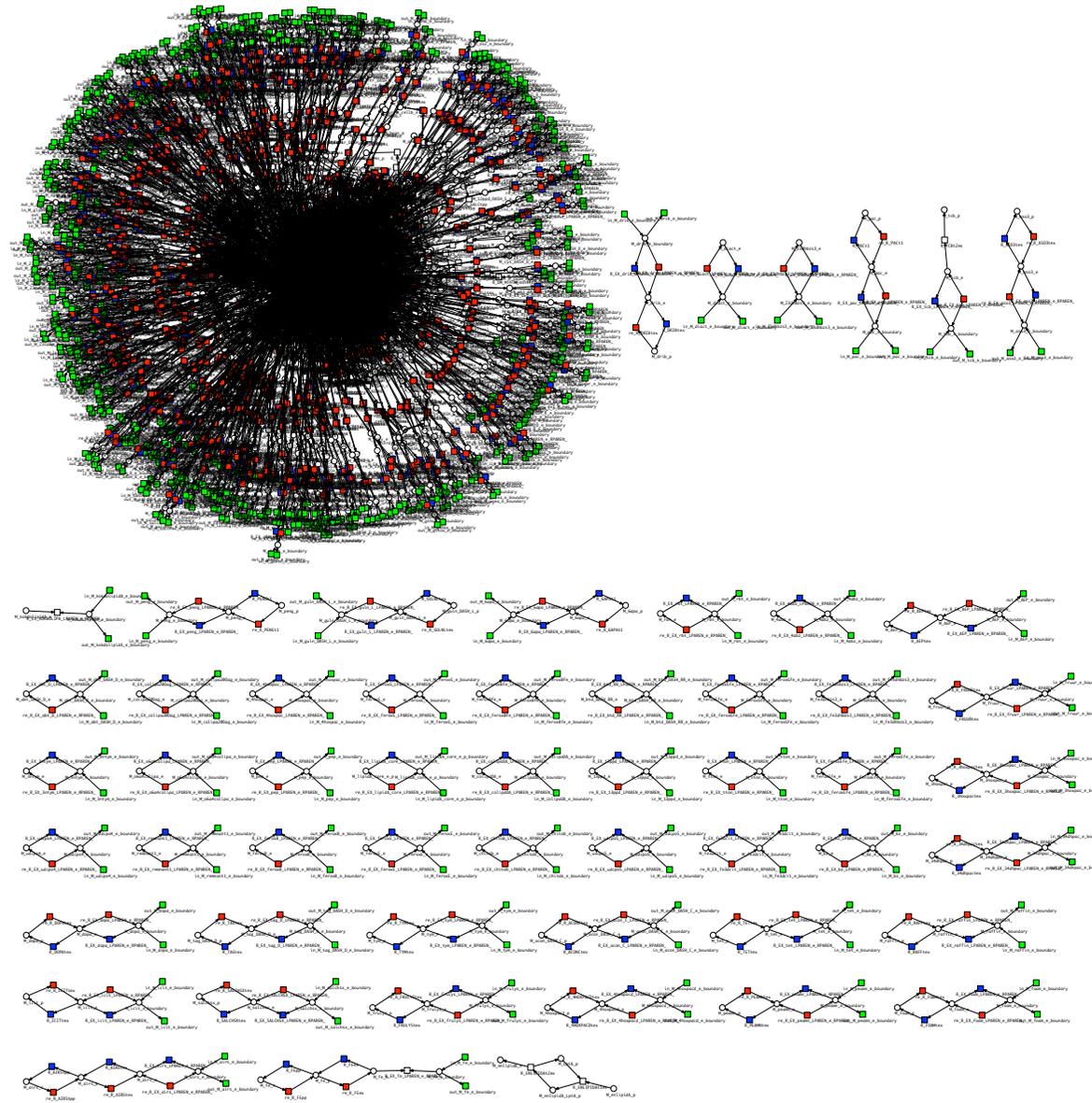
Literature



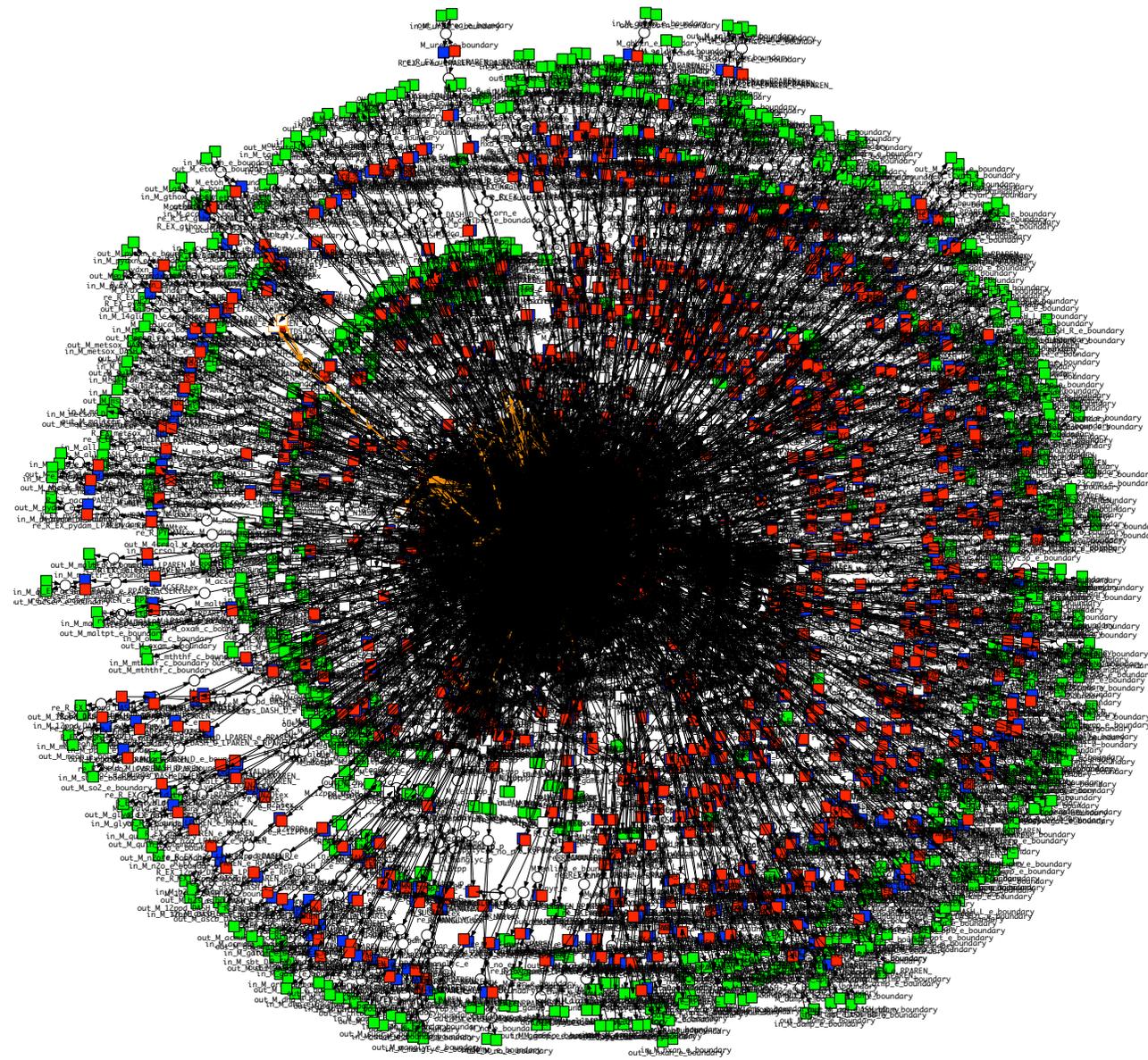


# APPROACH 3

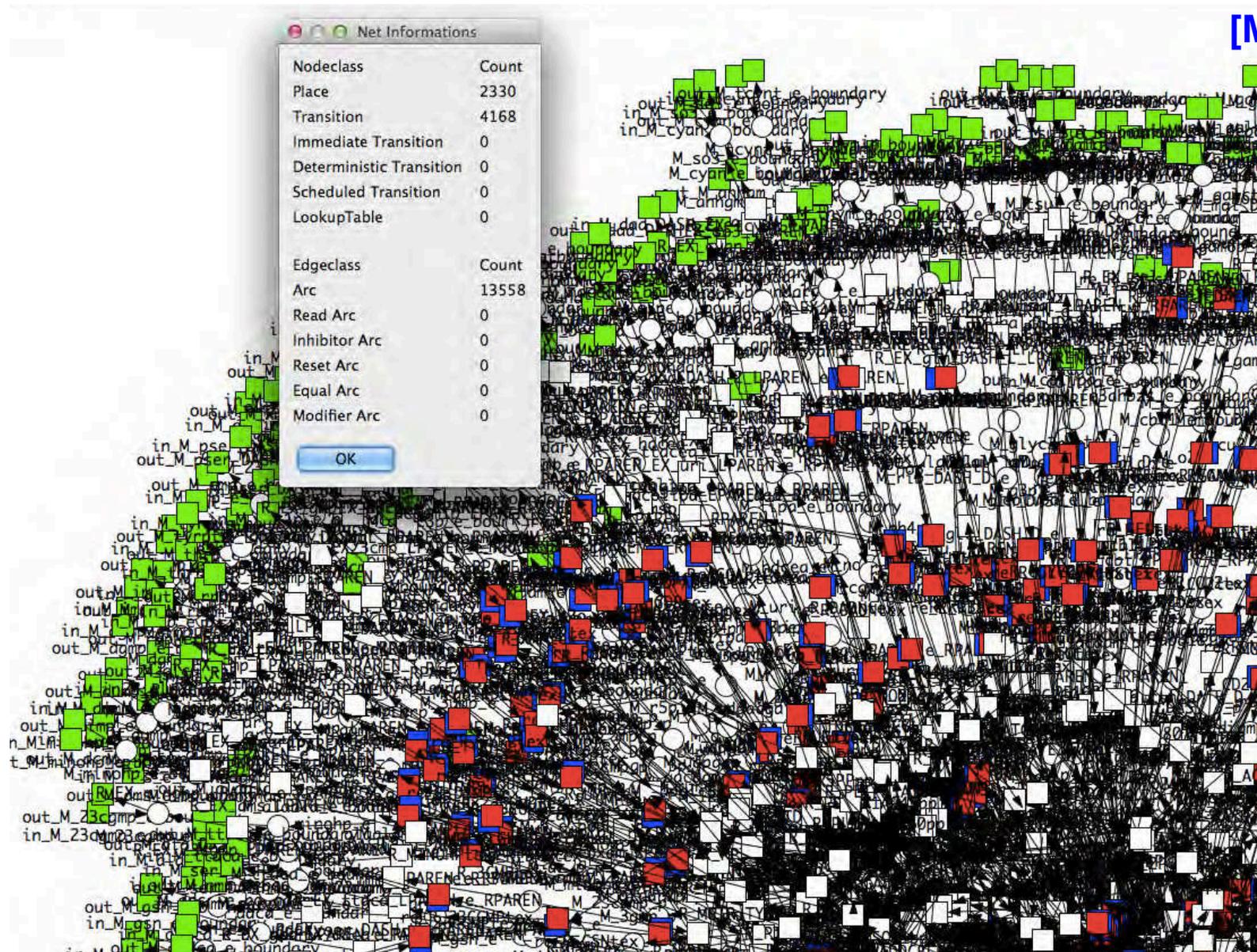
[Monk 2013]



[Monk 2013]



[Monk 2013]



# HOW TO SURVIVE LARGER MODELS ?

# HOW TO SURVIVE LARGER MODELS ?

-> MODULAR MODELLING

# HOW TO SURVIVE LARGER MODELS ?

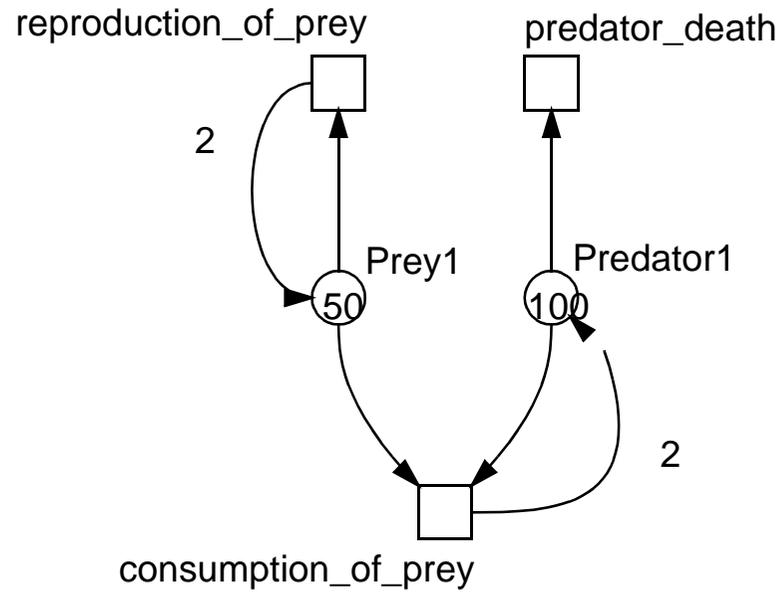
-> MODULAR MODELLING

-> COLOURED PETRI NETS

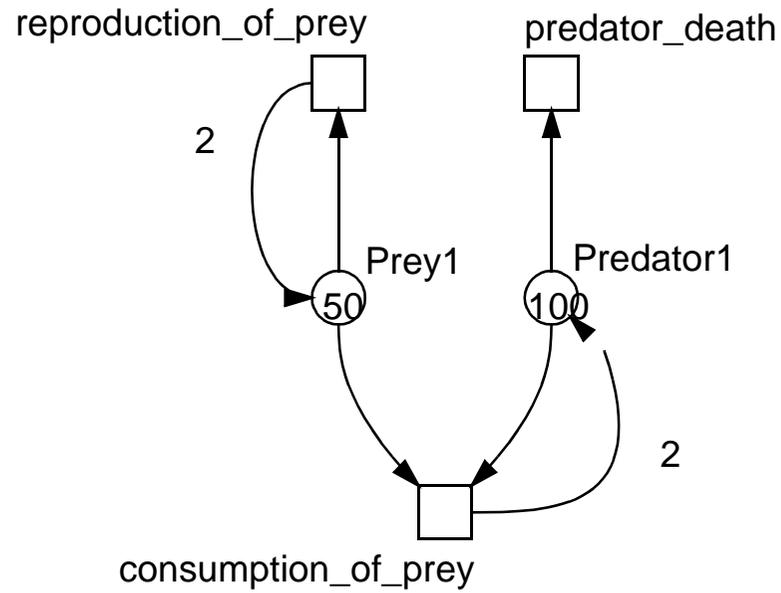
# ... AND THEN THERE WAS COLOUR



*Kew Gardens, 24/04/2011*

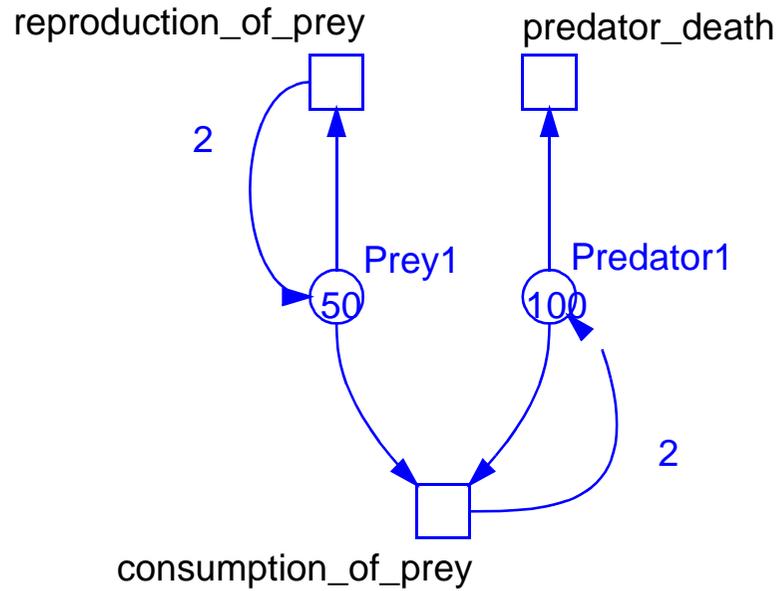


# EX: PREY - PREDATOR

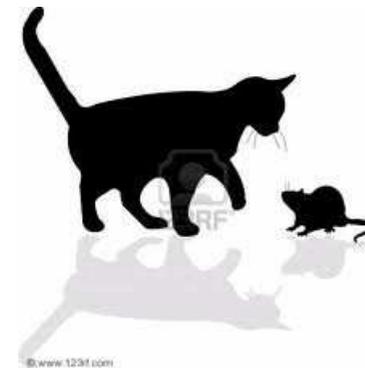
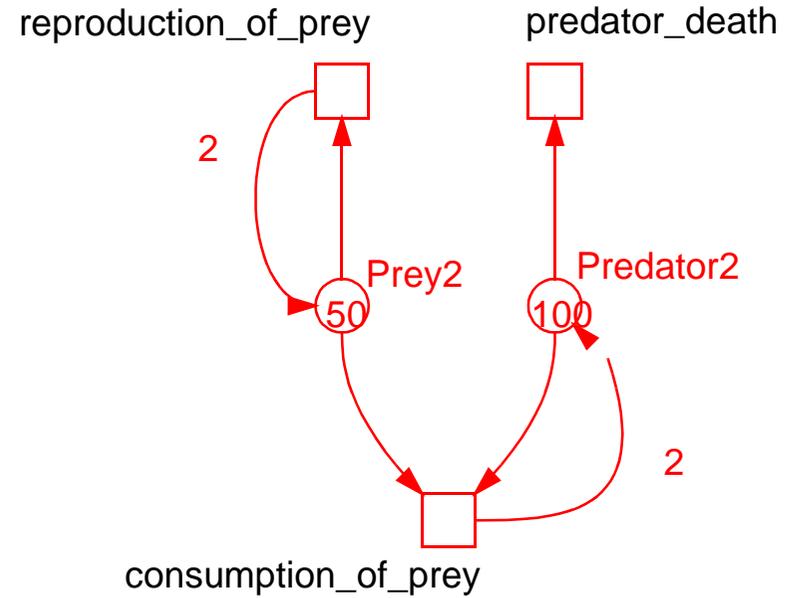


# EX: PREY - PREDATOR

sub-system1



sub-system2



❑ **definitions**

*colourset* CS = 1-2;

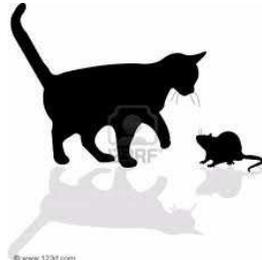
*var* x : CS;

❑ **better:**

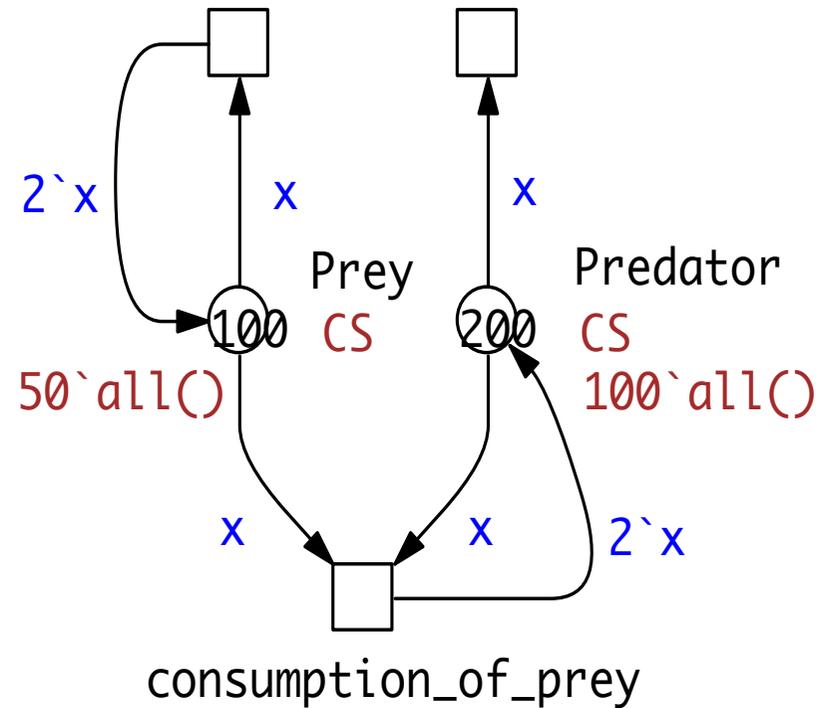
*const* SIZE = 2;

*colourset* CS = 1-SIZE;

*var* x : CS;



reproduction\_of\_prey predator\_death



❑ **definitions**

*colourset* CS = 1-2;

*var* x : CS;

❑ **better:**

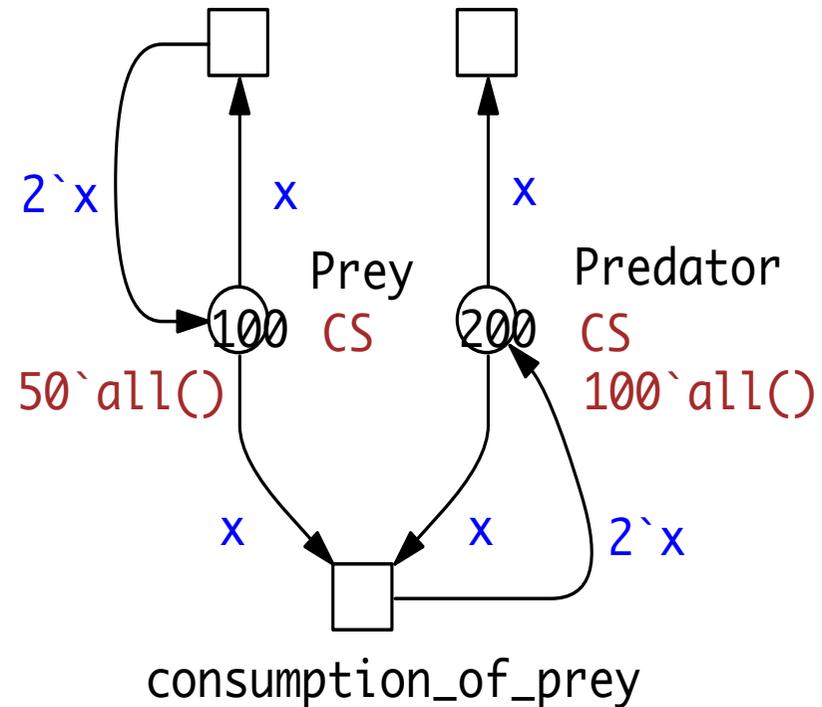
*const* SIZE = 2;

*colourset* CS = 1-SIZE;

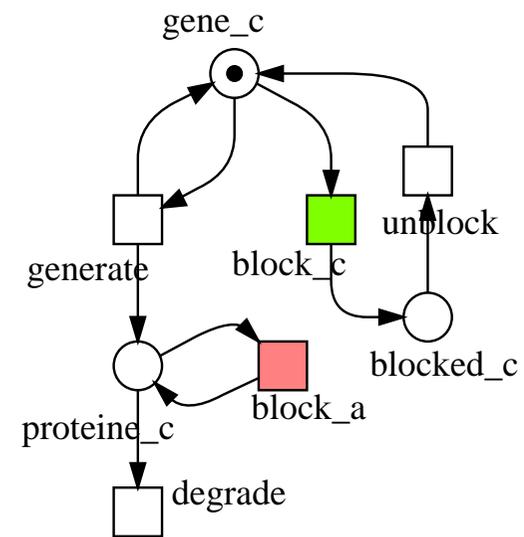
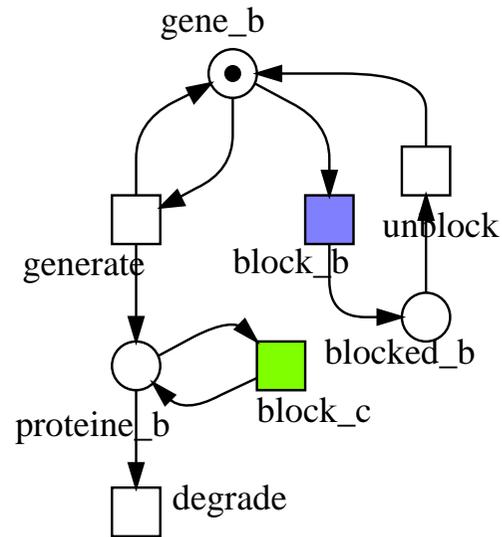
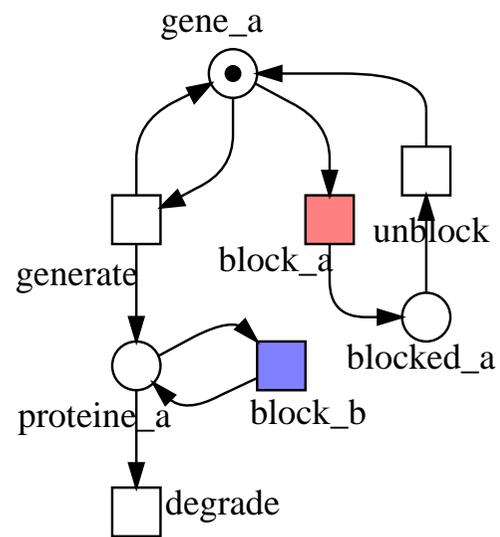
*var* x : CS;

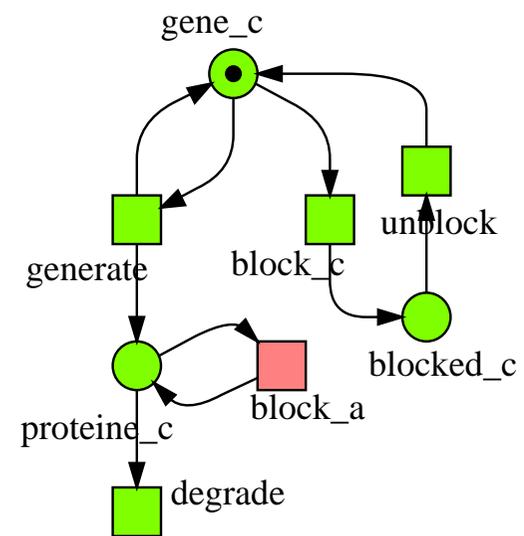
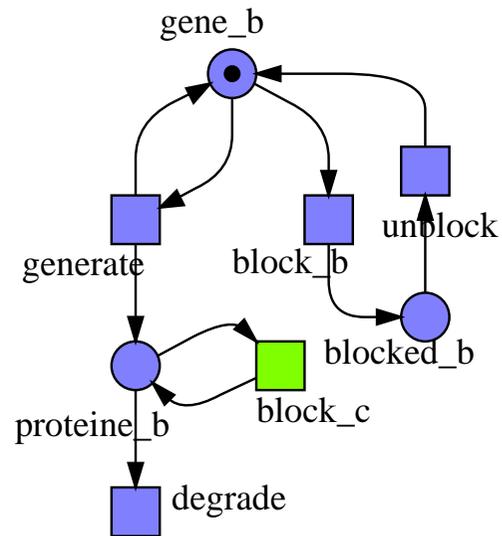
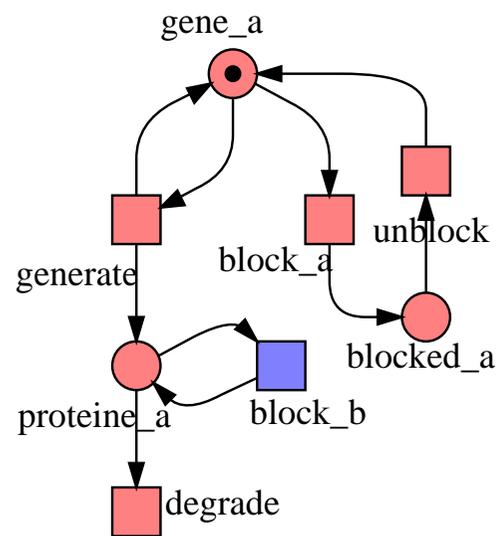


reproduction\_of\_prey predator\_death



❑ **changing SIZE adapts the model to various scenarios**

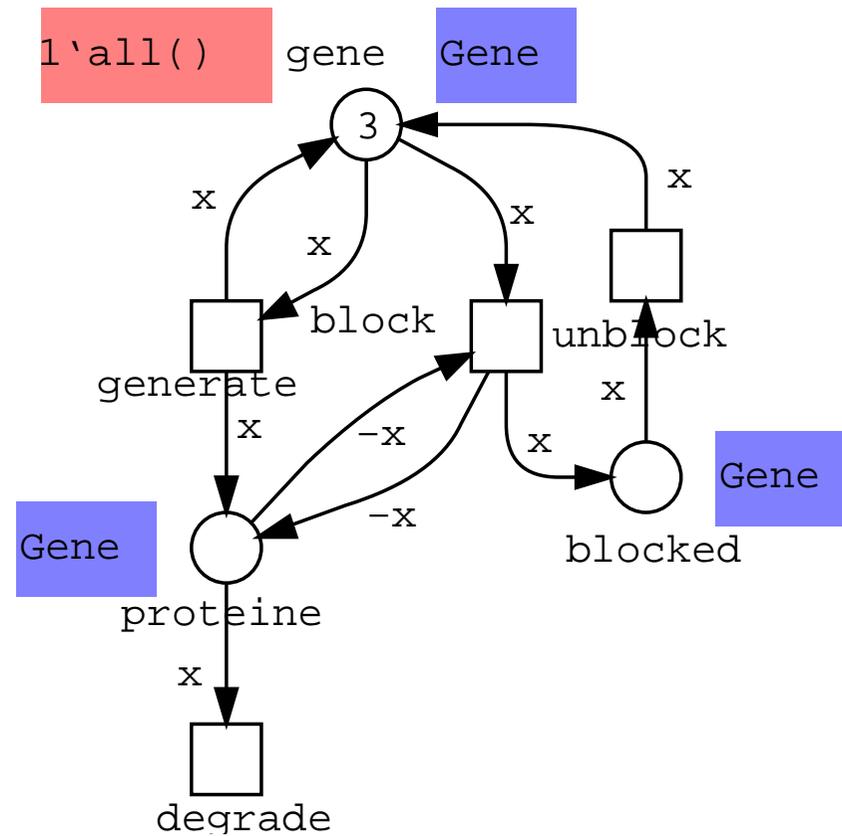




## □ definitions

*colorset* Gene = enum a-c;

var x : Gene;



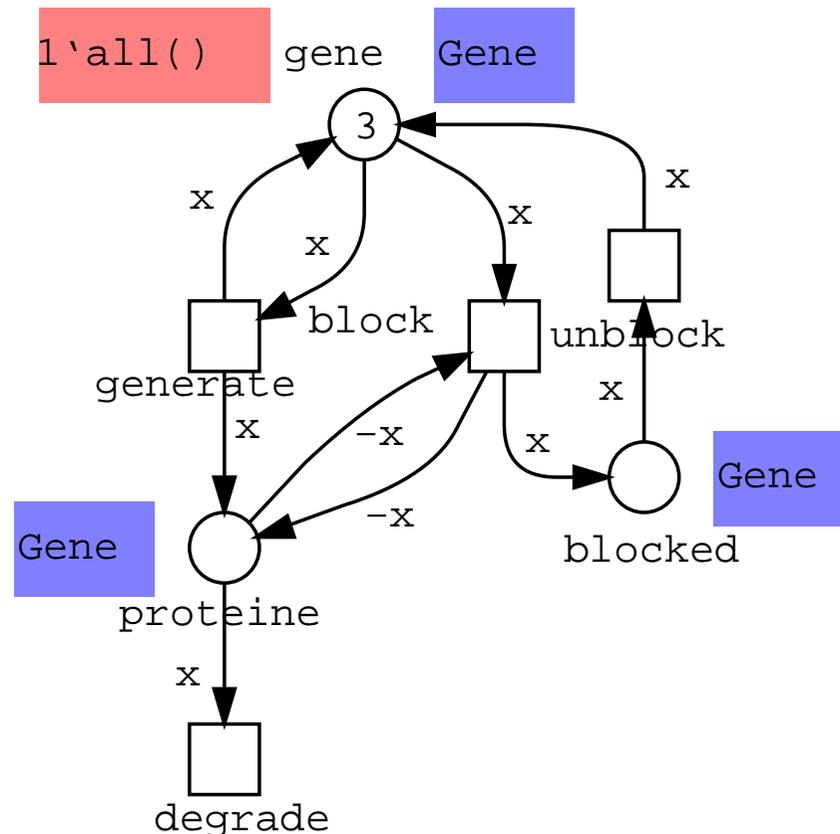
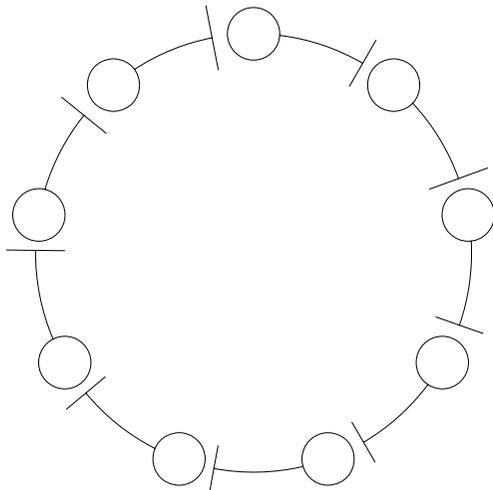
## ❑ definitions

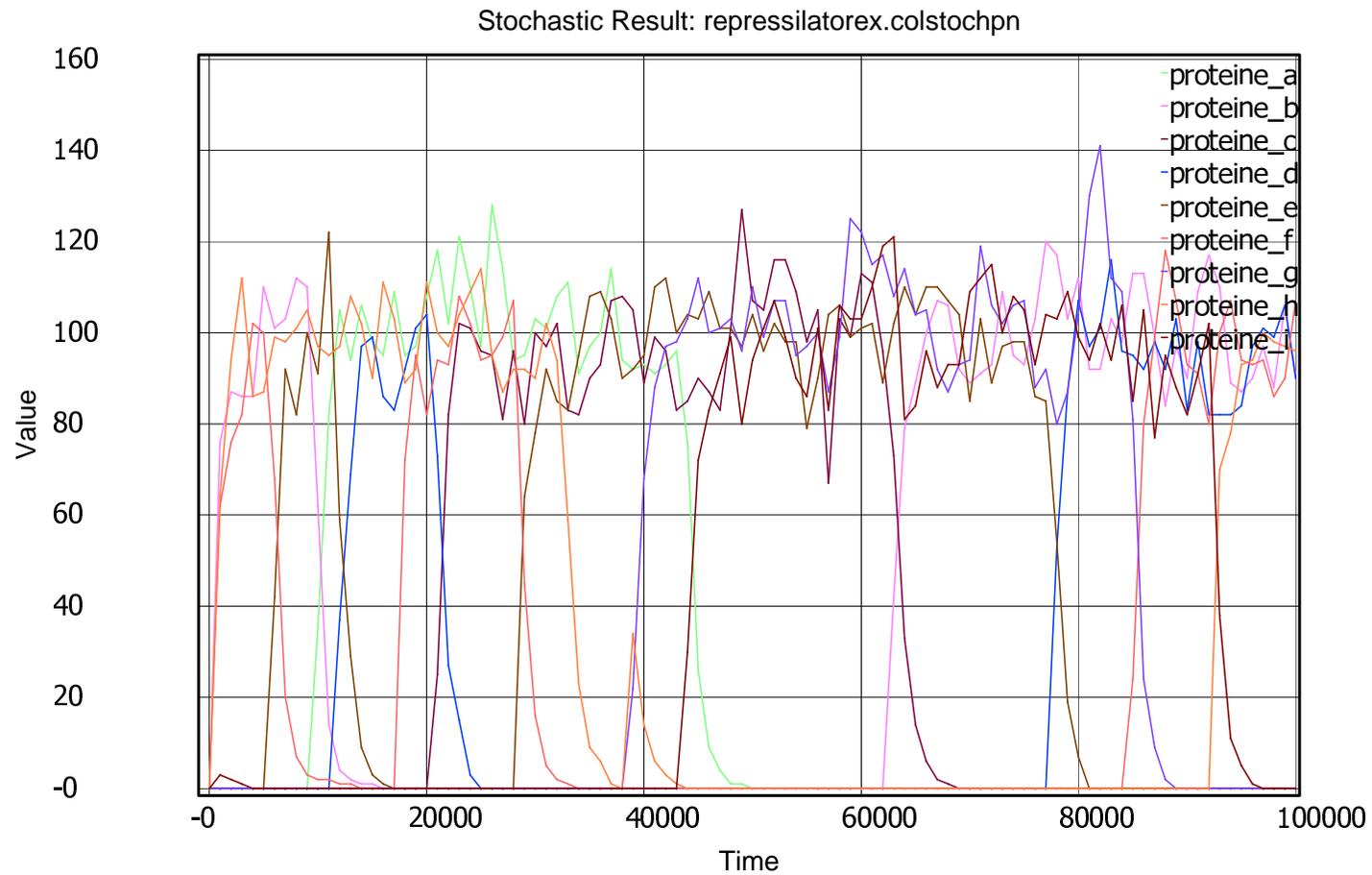
*colorset* Gene = enum a-c;

var x : Gene;

## ❑ model scaling

*colorset* Gene = enum a-i;





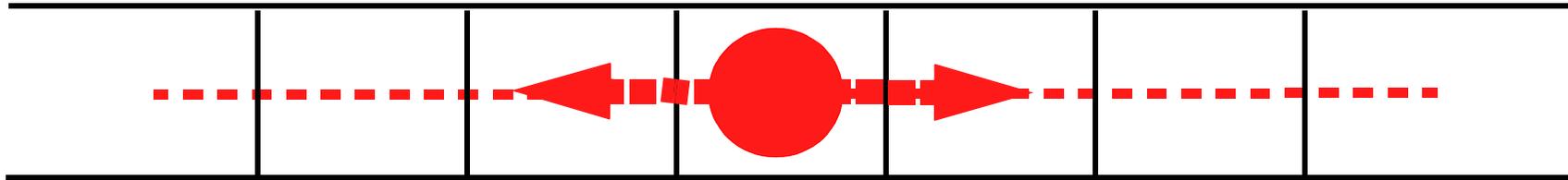
# COLOURING SPACE

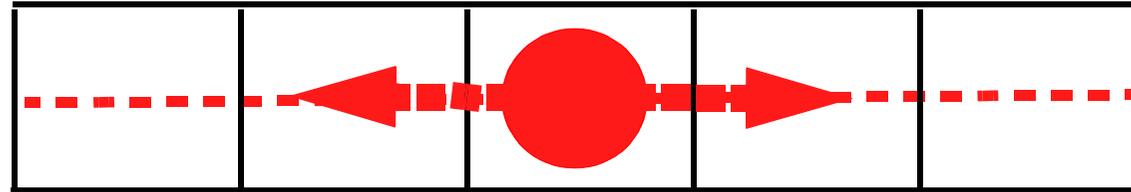
## EXAMPLE 1:

### DIFFUSION IN SPACE

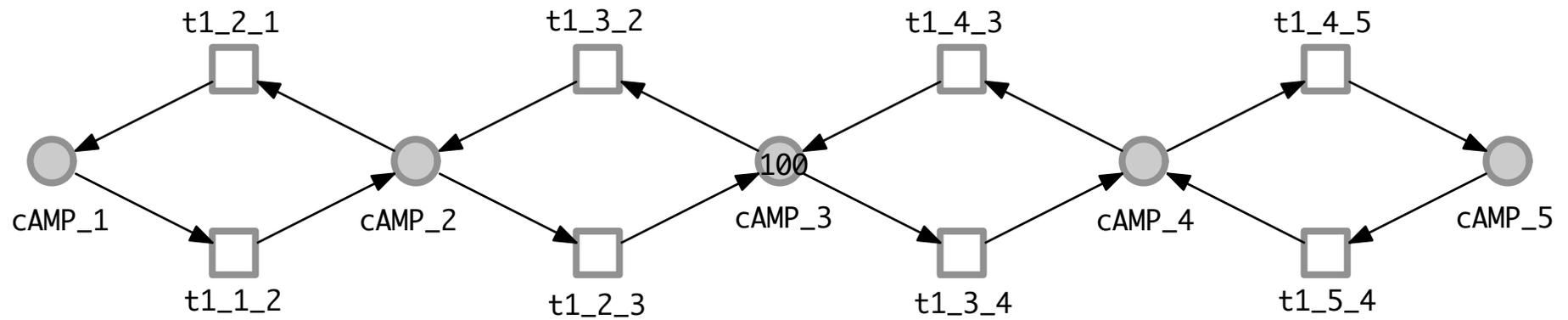
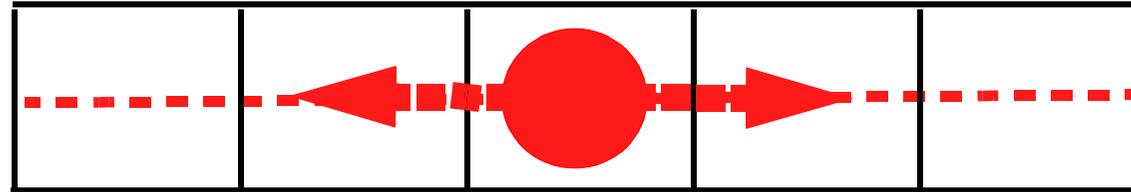


*Richmond, 13/09/2011*





# Ex1: DIFFUSION - 1D



### □ definitions

```
const D1 = 5;           // grid size  
const MIDDLE = D1/2;  
colorset CS = 1-D1;    // grid positions  
var x,y : CS;
```

### □ definitions

```
const D1 = 5;           // grid size
```

```
const MIDDLE = D1/2;
```

```
colorset CS = 1-D1;    // grid positions
```

```
var x,y : CS;
```

```
function neighbour1D (CS x,a) bool:
```

```
  // a is neighbour of x
```

```
  ( a=x-1 | a=x+1 ) & ( 1<=a ) & ( a<=D1 );
```

## □ definitions

```
const D1 = 5;           // grid size
```

```
const MIDDLE = D1/2;
```

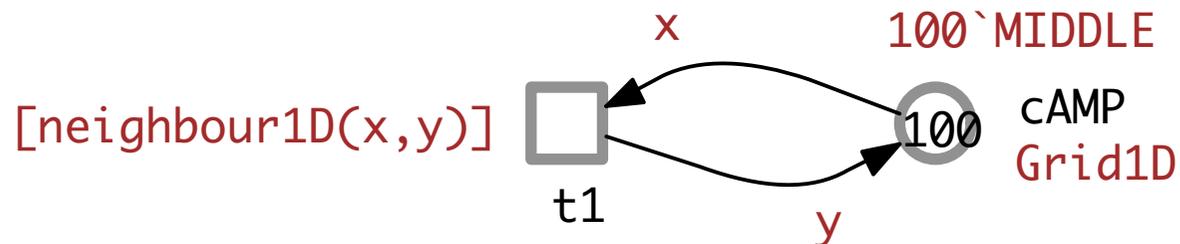
```
colorset CS = 1-D1;    // grid positions
```

```
var x,y : CS;
```

```
function neighbour1D (CS x,a) bool:
```

```
  // a is neighbour of x
```

```
  ( a=x-1 | a=x+1 ) & ( 1<=a ) & ( a<=D1 );
```



## □ definitions

```
const D1 = 5;           // grid size
```

```
const MIDDLE = D1/2;
```

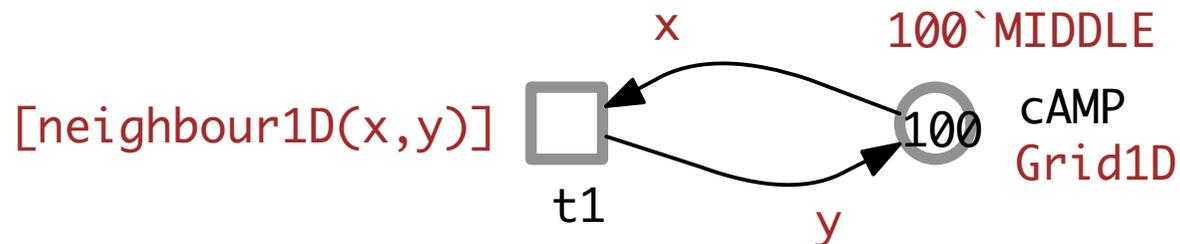
```
colorset CS = 1-D1;    // grid positions
```

```
var x,y : CS;
```

```
function neighbour1D (CS x,a) bool:
```

```
  // a is neighbour of x
```

```
  ( a=x-1 | a=x+1 ) & ( 1<=a ) & ( a<=D1 );
```



## □ movement = changing colour

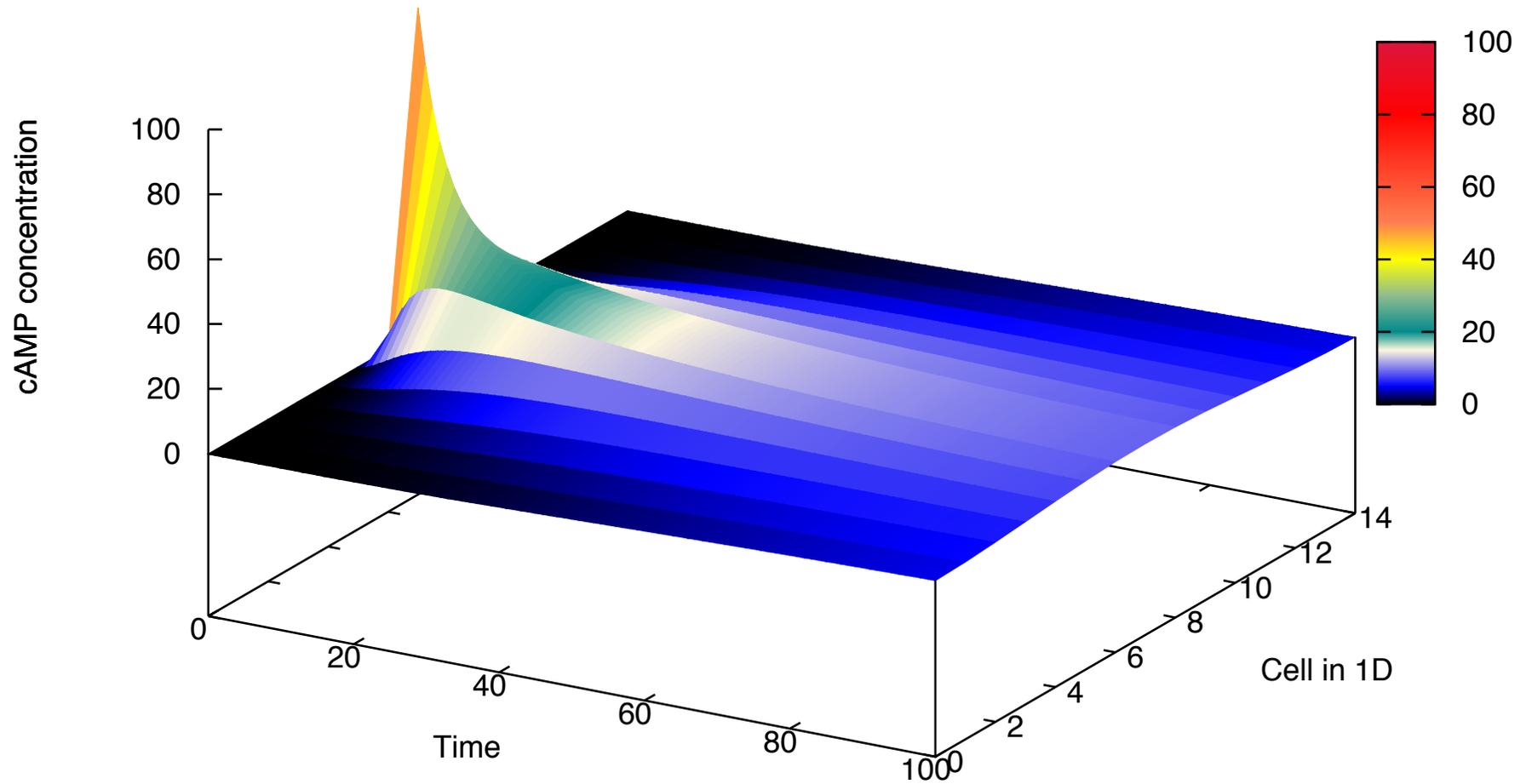
$$\frac{dc_1}{dt} = k \cdot c_2 - k \cdot c_1$$

$$\frac{dc_2}{dt} = k \cdot c_1 + k \cdot c_3 - 2 \cdot k \cdot c_2$$

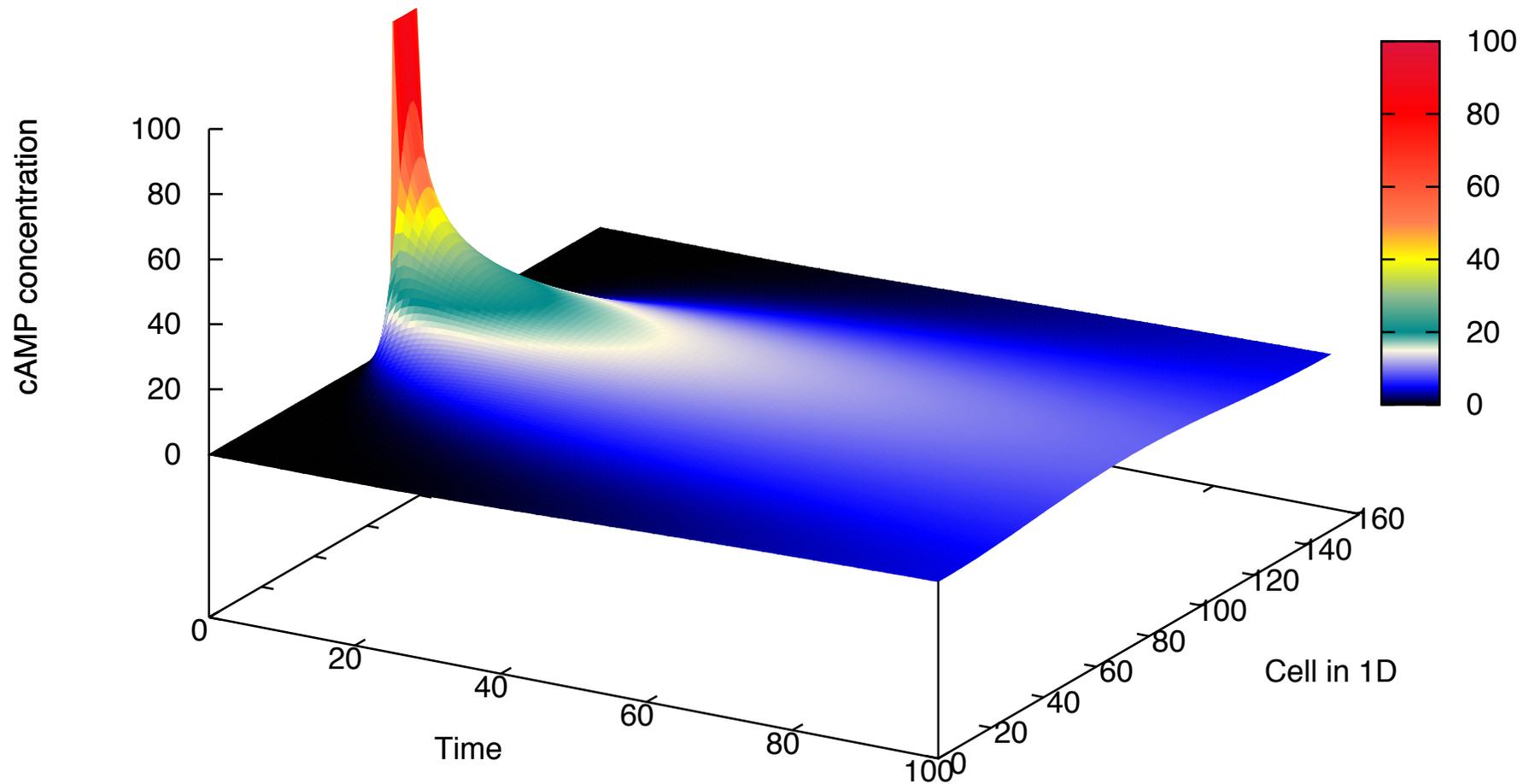
$$\frac{dc_3}{dt} = k \cdot c_2 + k \cdot c_4 - 2 \cdot k \cdot c_3$$

$$\frac{dc_4}{dt} = k \cdot c_3 + k \cdot c_5 - 2 \cdot k \cdot c_4$$

$$\frac{dc_5}{dt} = k \cdot c_4 - k \cdot c_5$$

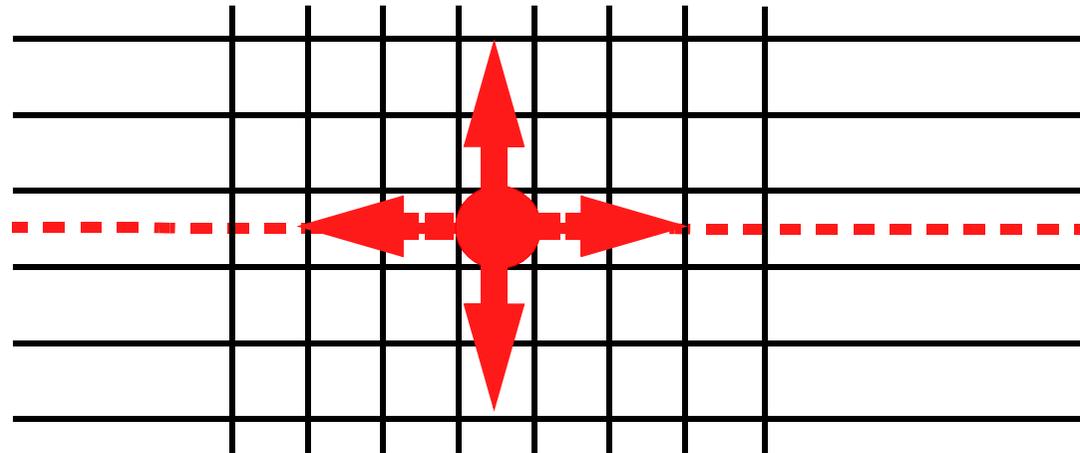


## 15 GRID POSITIONS

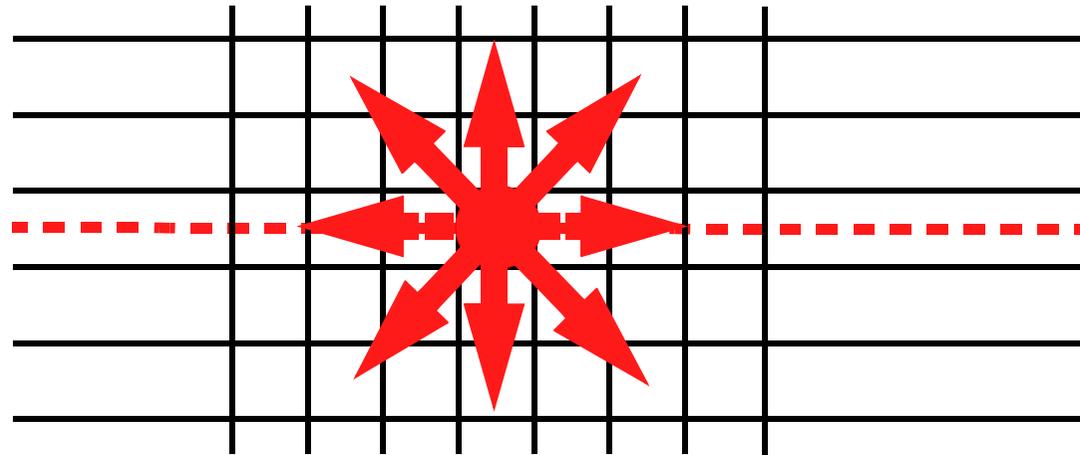


## 150 GRID POSITIONS, SCALING OF INITIAL MARKING AND RATES

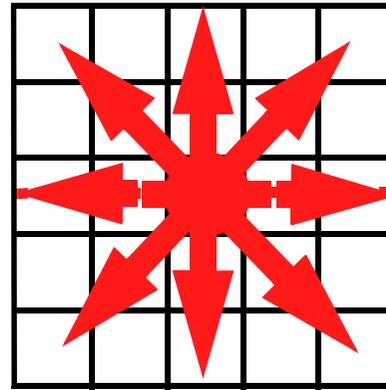
## ❑ SCHEME



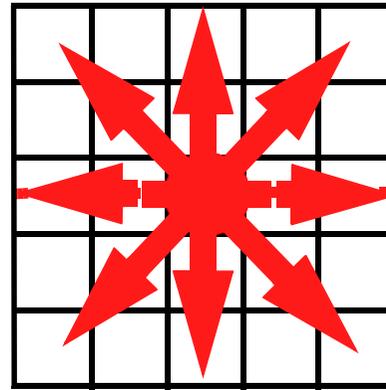
## ❑ SCHEME



## ❑ SCHEME



## ❑ SCHEME



## ❑ definitions

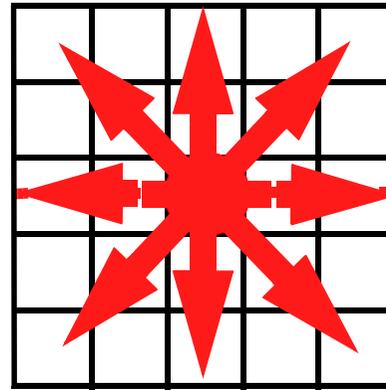
*const D1 = 5;*

*// grid size first dimension*

*const D2 = D1;*

*// grid size second dimension*

## ❑ SCHEME

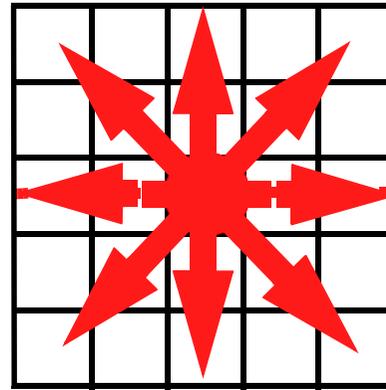


## ❑ definitions

```
const D1 = 5;           // grid size first dimension
const D2 = D1;         // grid size second dimension
const MIDDLE = D1/2;

colorset CD1 = 1-D1;   // row index
colorset CD2 = 1-D2;   // column index
colorset Grid2D = CD1 x CD2; // 2D grid
```

## ❑ SCHEME



## ❑ definitions

```
const D1 = 5;           // grid size first dimension
const D2 = D1;         // grid size second dimension
const MIDDLE = D1/2;

colorset CD1 = 1-D1;   // row index
colorset CD2 = 1-D2;   // column index
colorset Grid2D = CD1 x CD2; // 2D grid

var x, a : CD1;
var y, b : CD2;
```

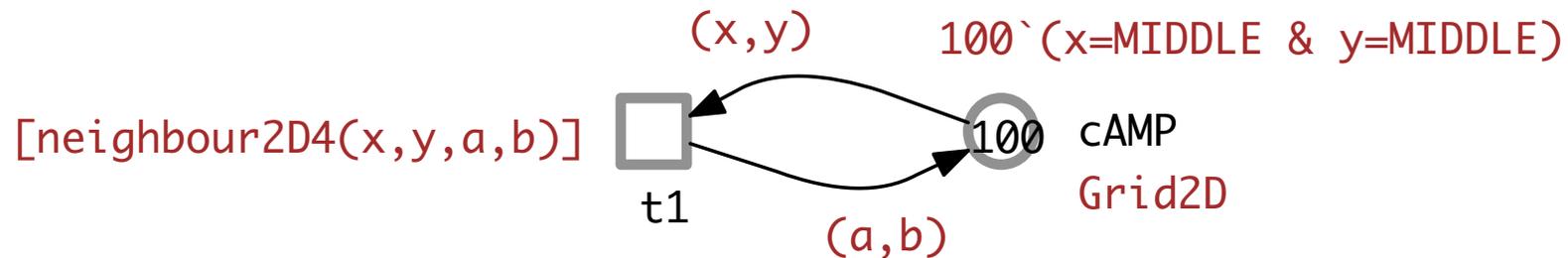
## □ four neighbours

**function** neighbour2D4 (CD1 x, CD2 y, CD1 a, CD2 b) **bool**:

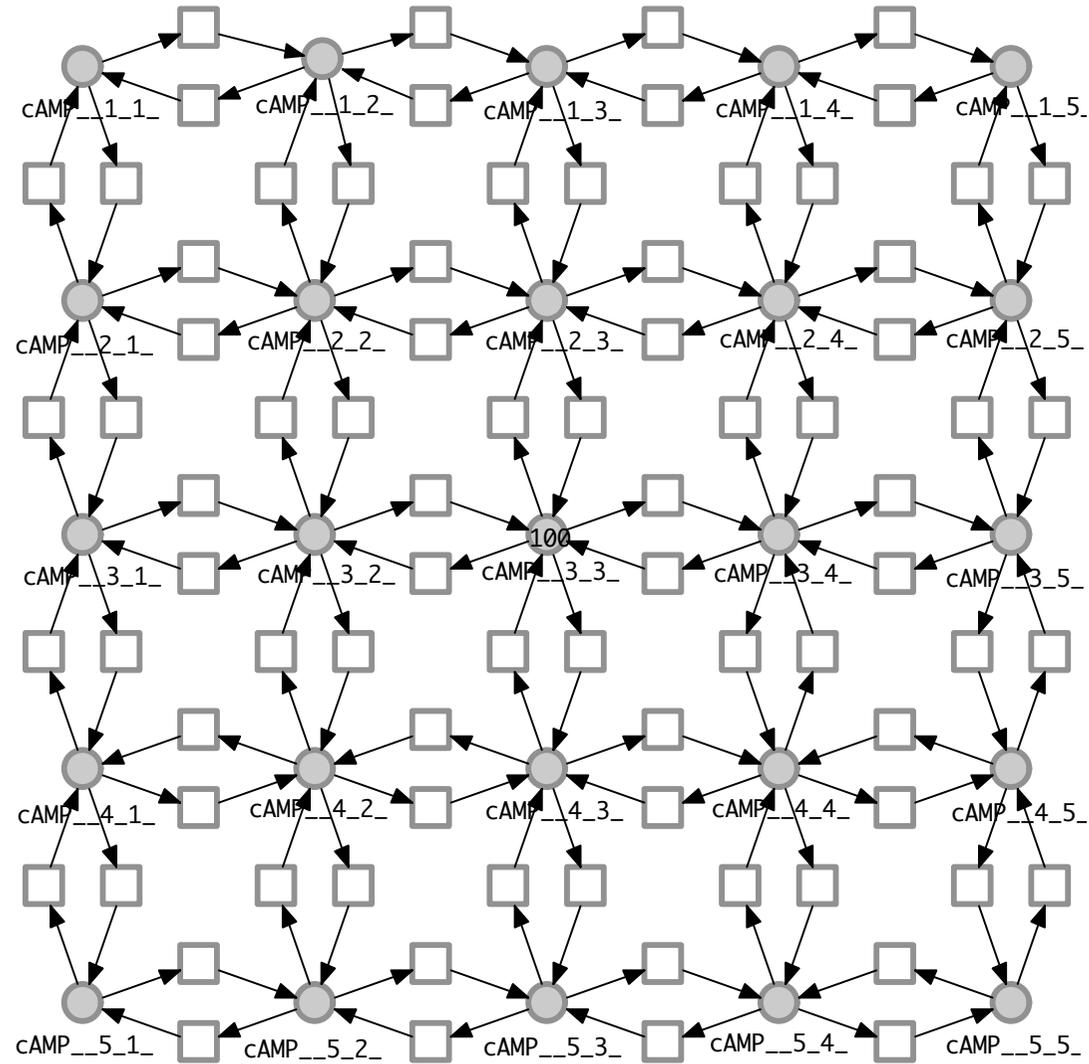
// (a,b) is one of the up to four neighbours of (x,y)

(a=x & b=y-1) | (a=x & b=y+1)

| (b=y & a=x-1) | (b=y & a=x+1);



# Ex1: DIFFUSION - 2D4 NEIGHBOURHOOD



## □ eight neighbours

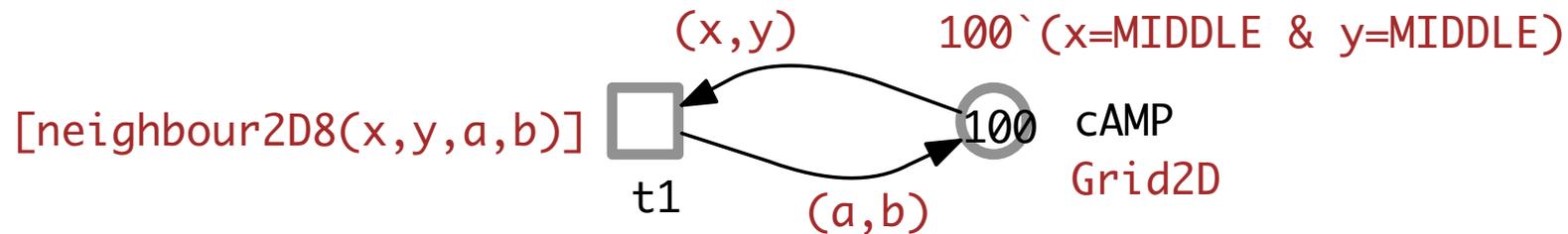
**function** neighbour2D8 (CD1 x, CD2 y, CD1 a, CD2 b) **bool**:

// (a,b) is one of the up to eight neighbours of (x,y)

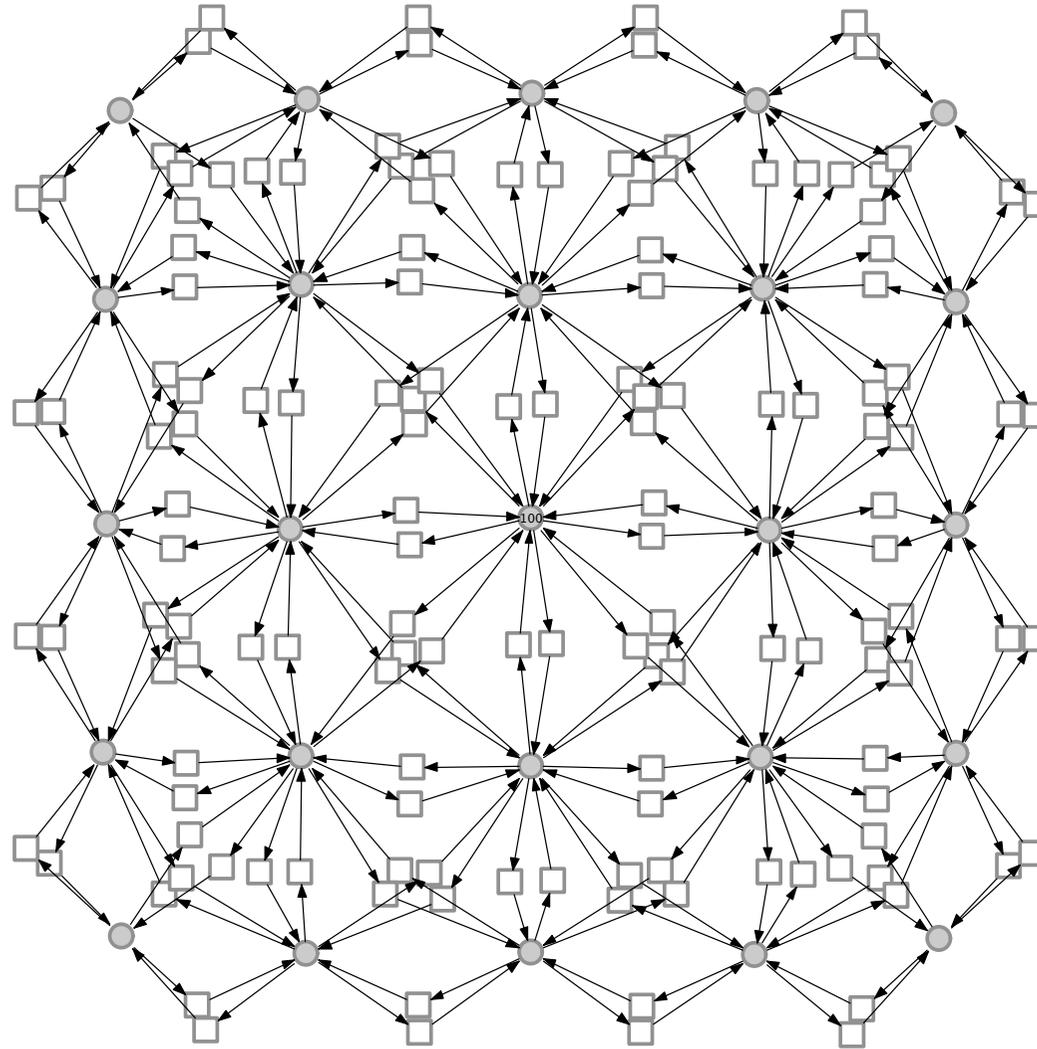
(a=x-1 | a=x | a=x+1) & (b = y-1 | b=y | b=y+1)

& (!(a=x & b=y))

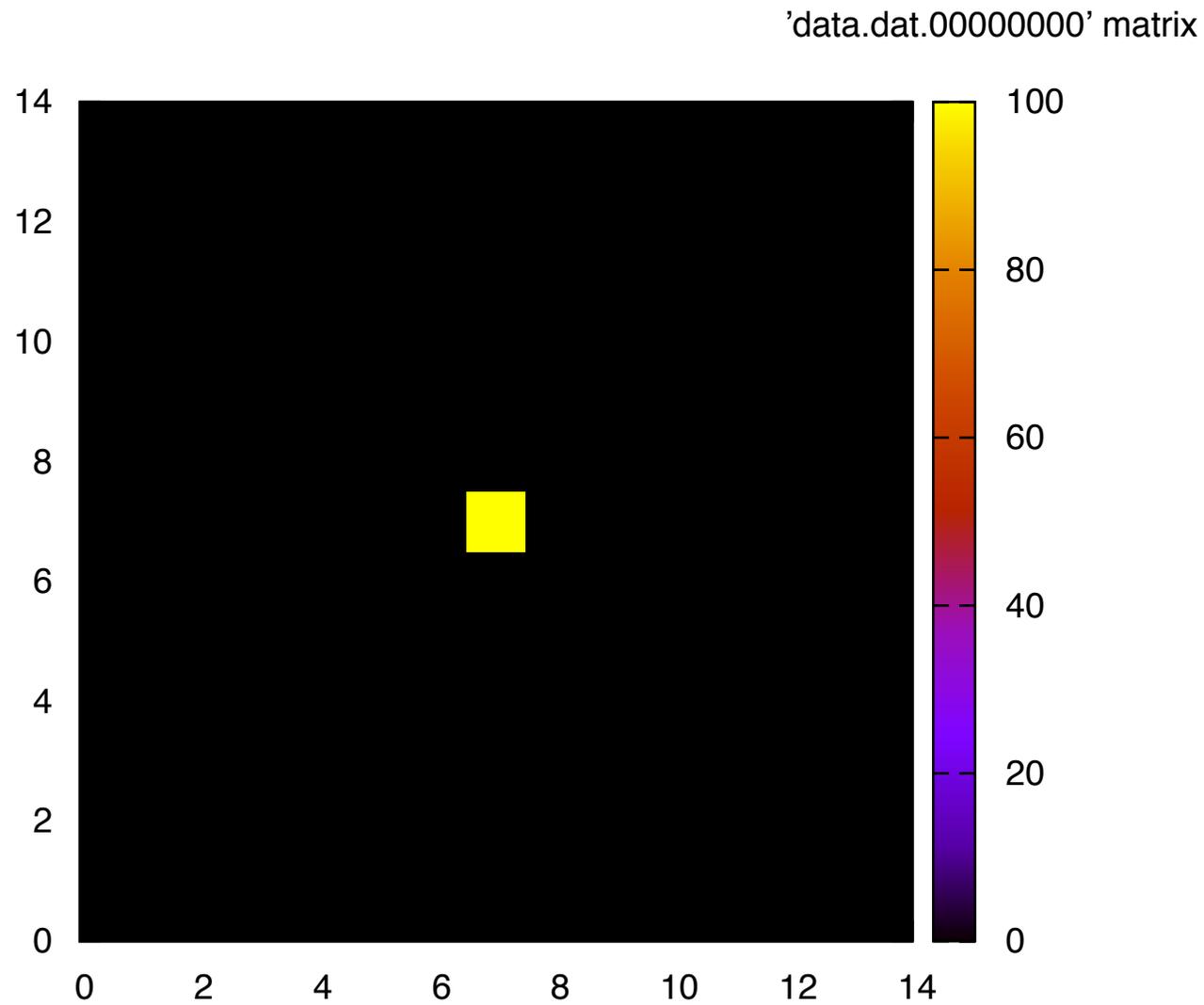
& (1<=a & a<=D1) & (1<=b & b<=D2);



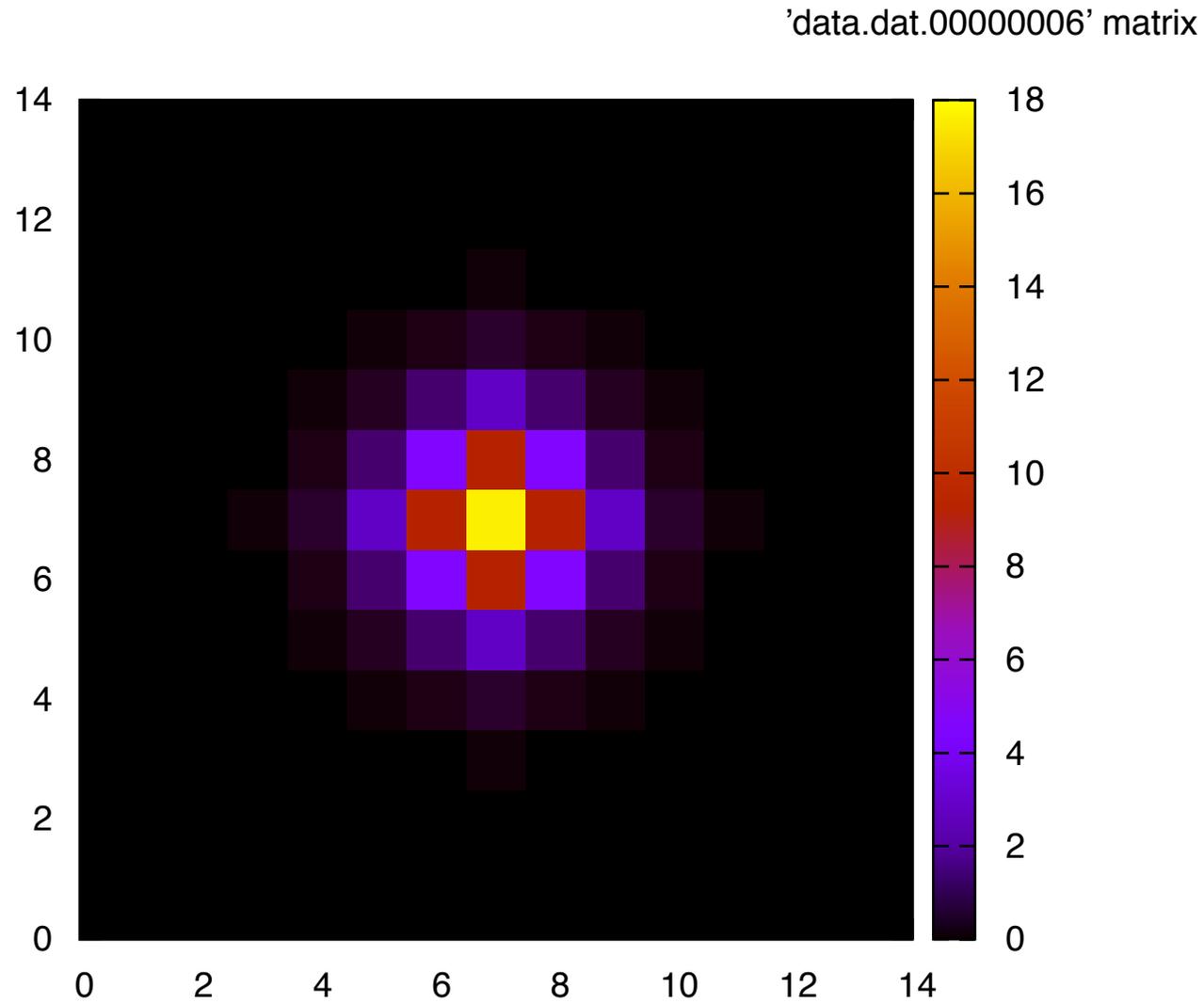
# Ex1: DIFFUSION - 2D8 NEIGHBOURHOOD



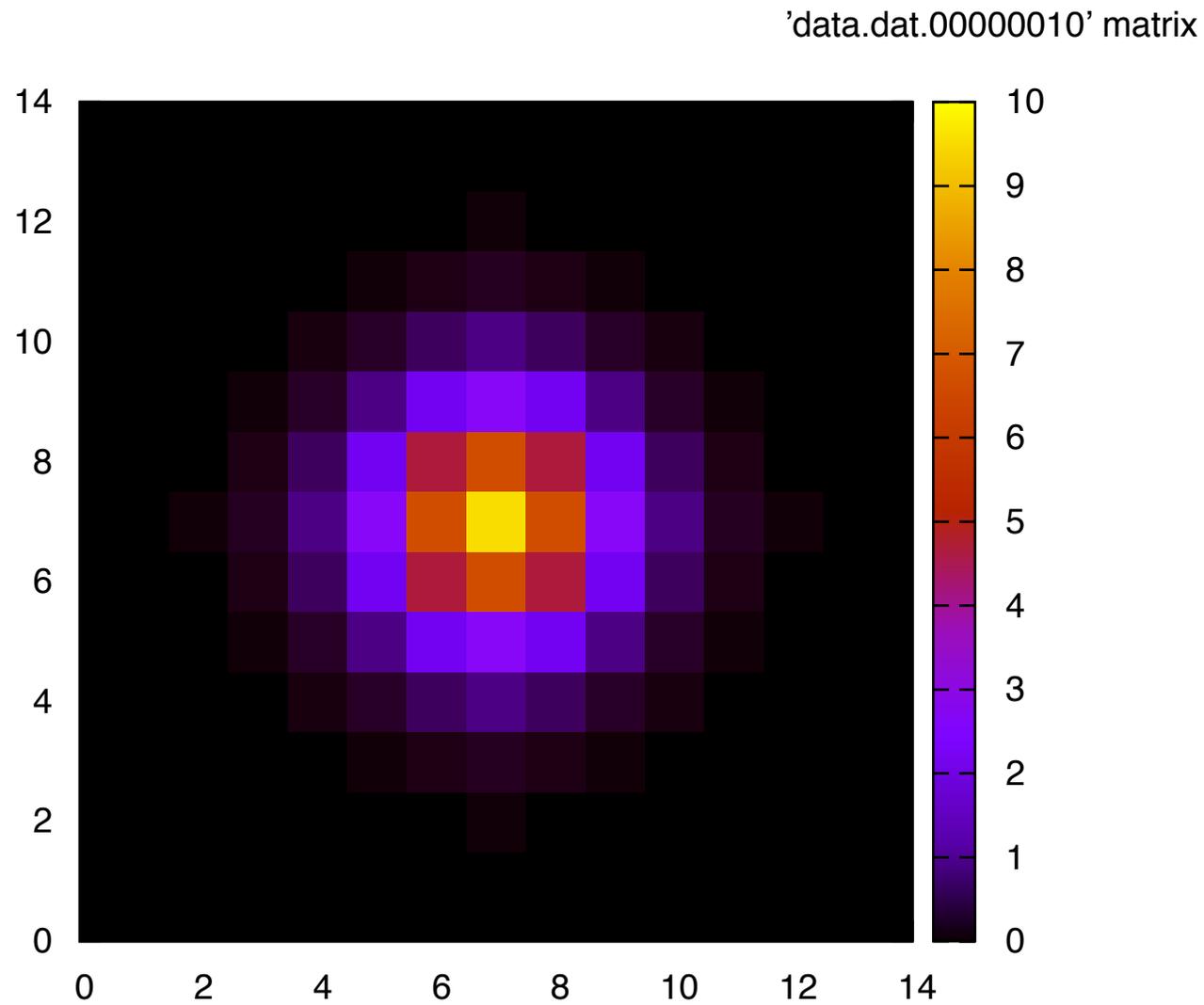
# Ex1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



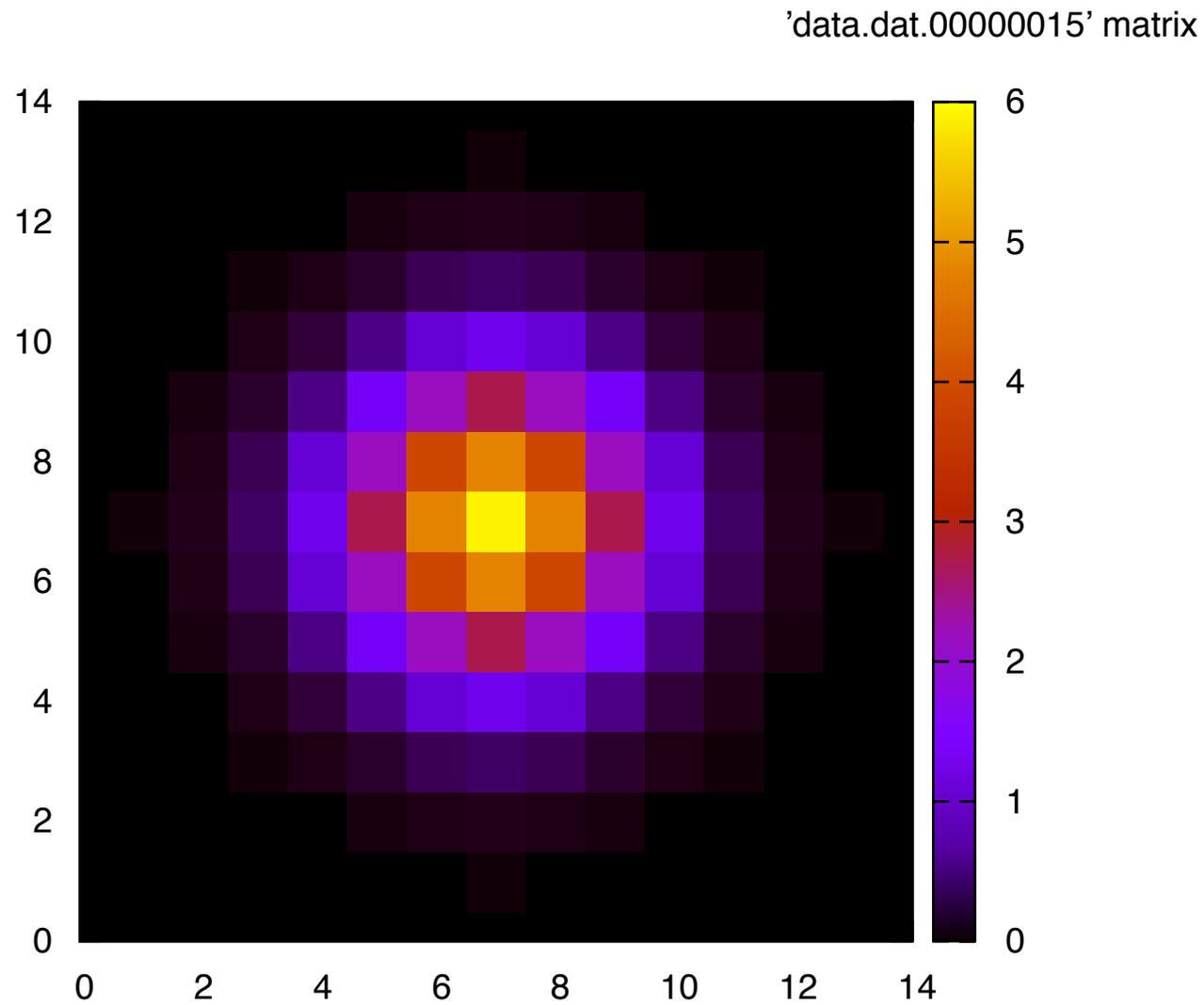
# Ex1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



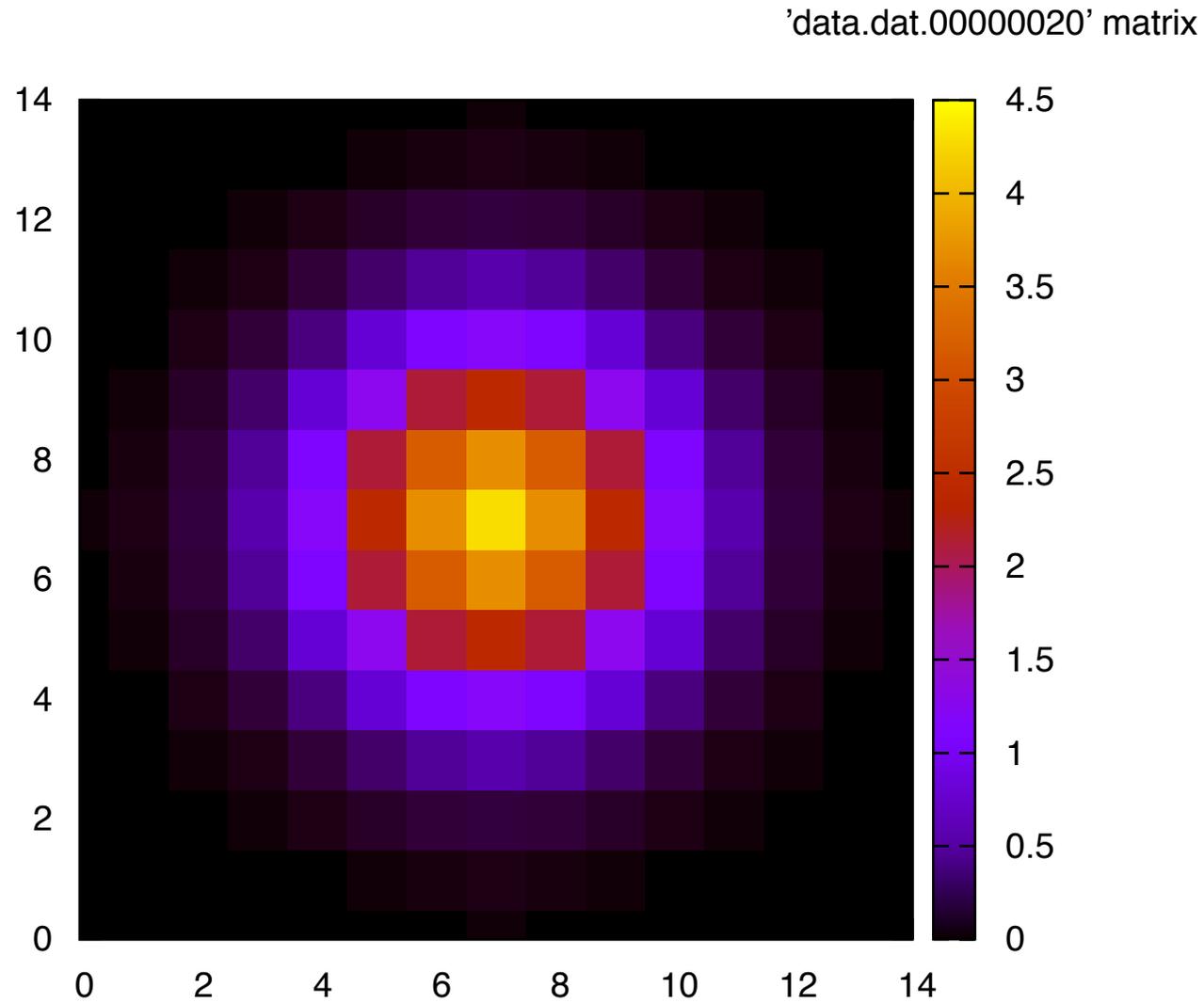
# Ex1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



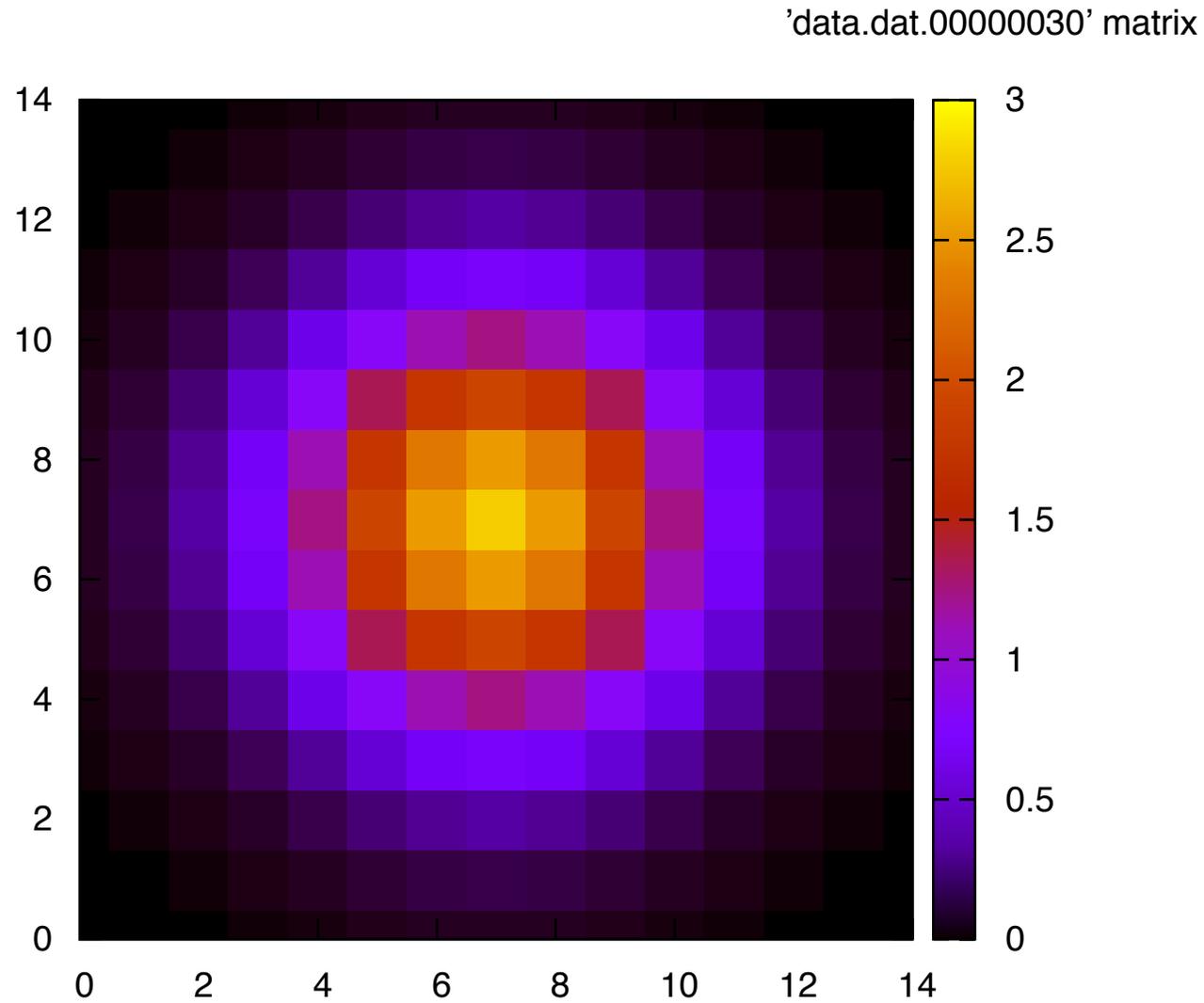
# Ex1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



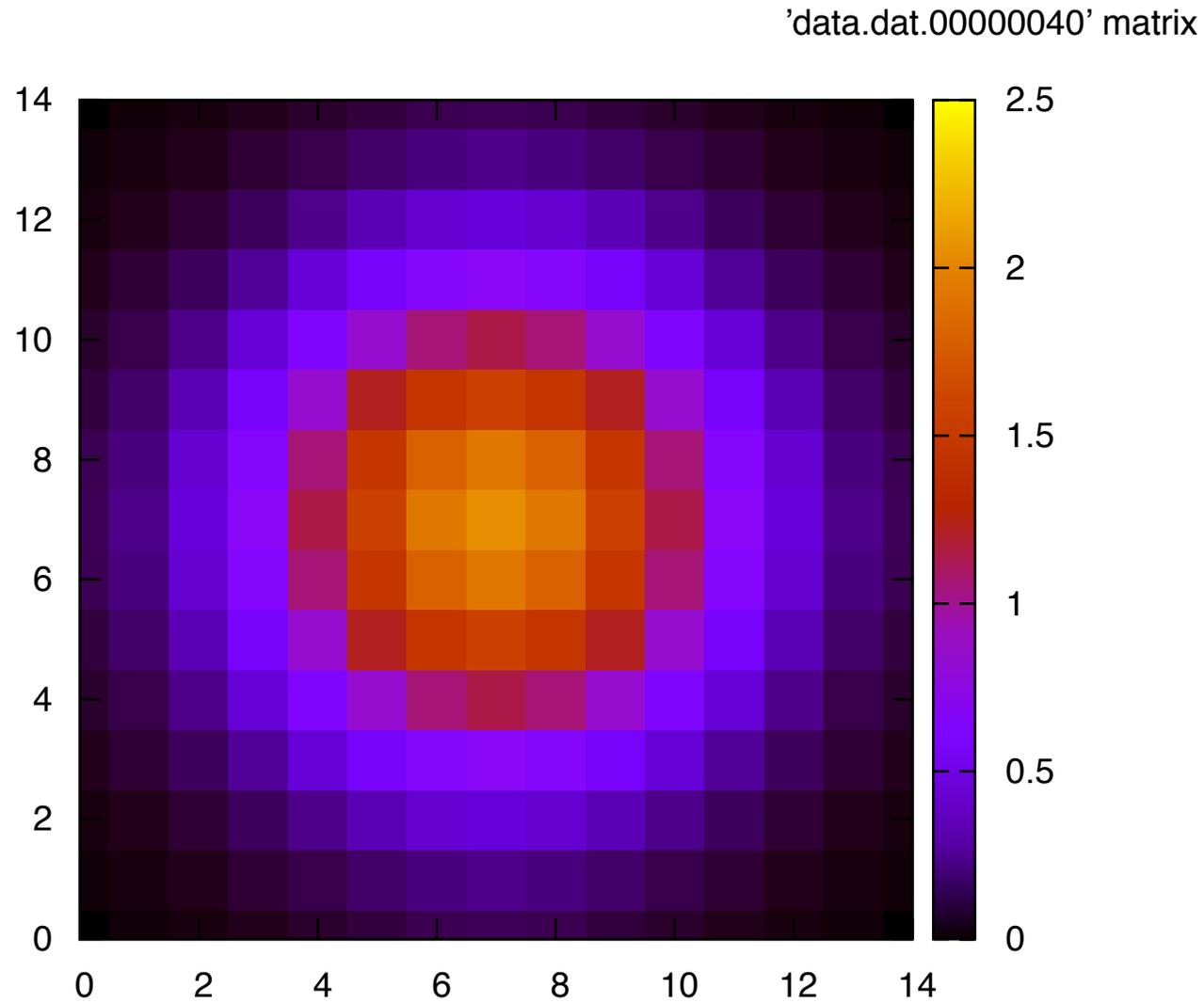
# Ex1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



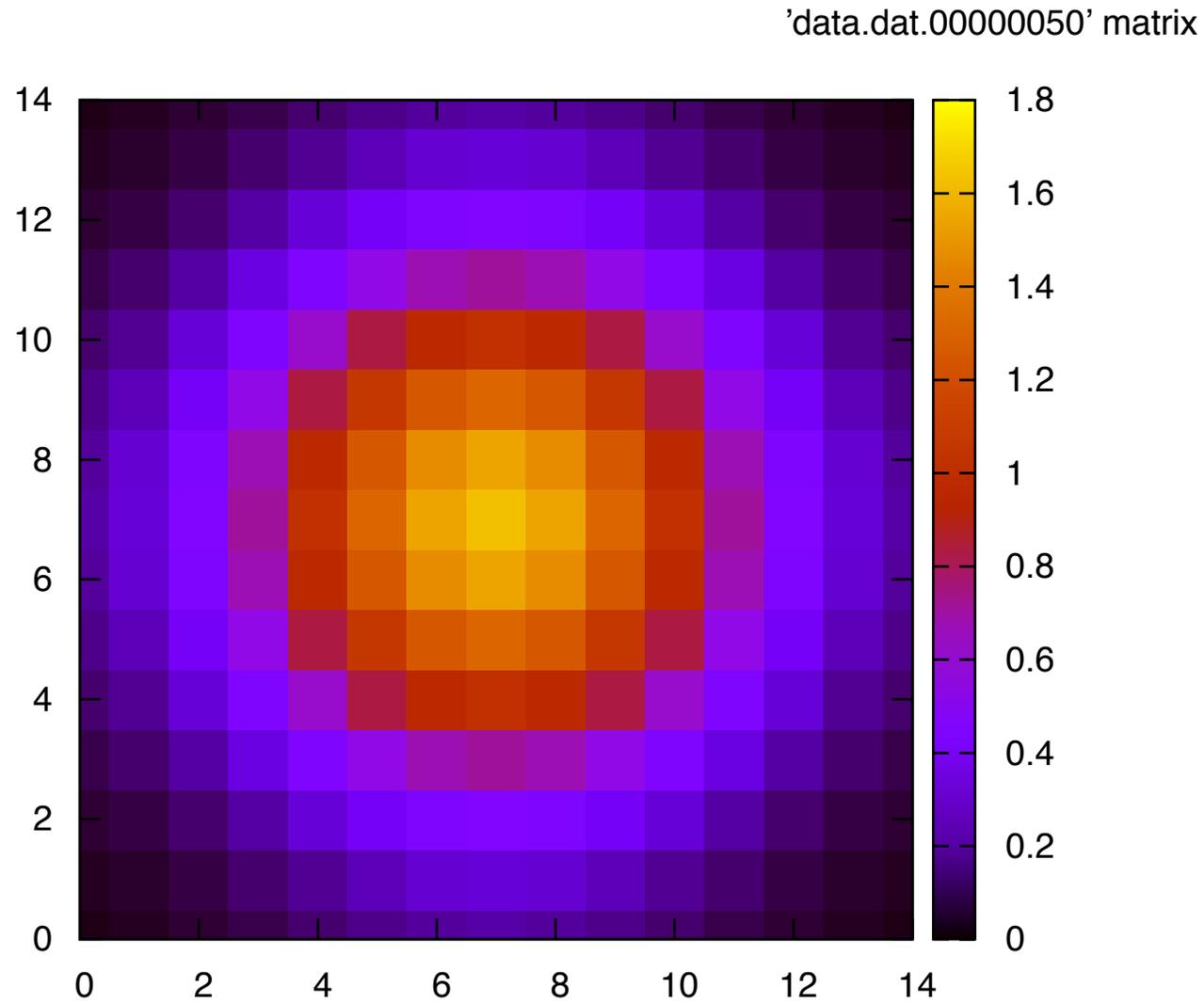
# Ex1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15

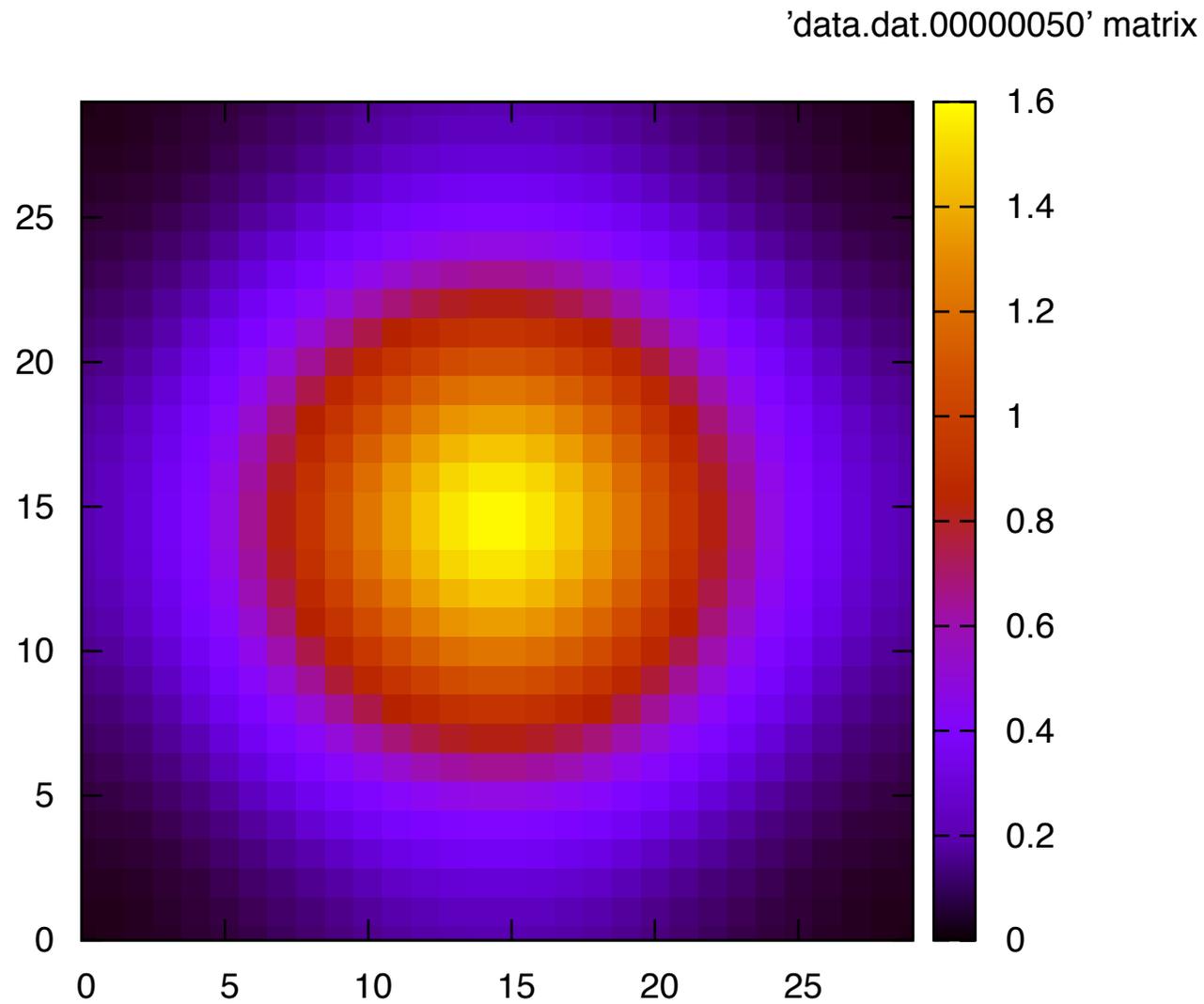


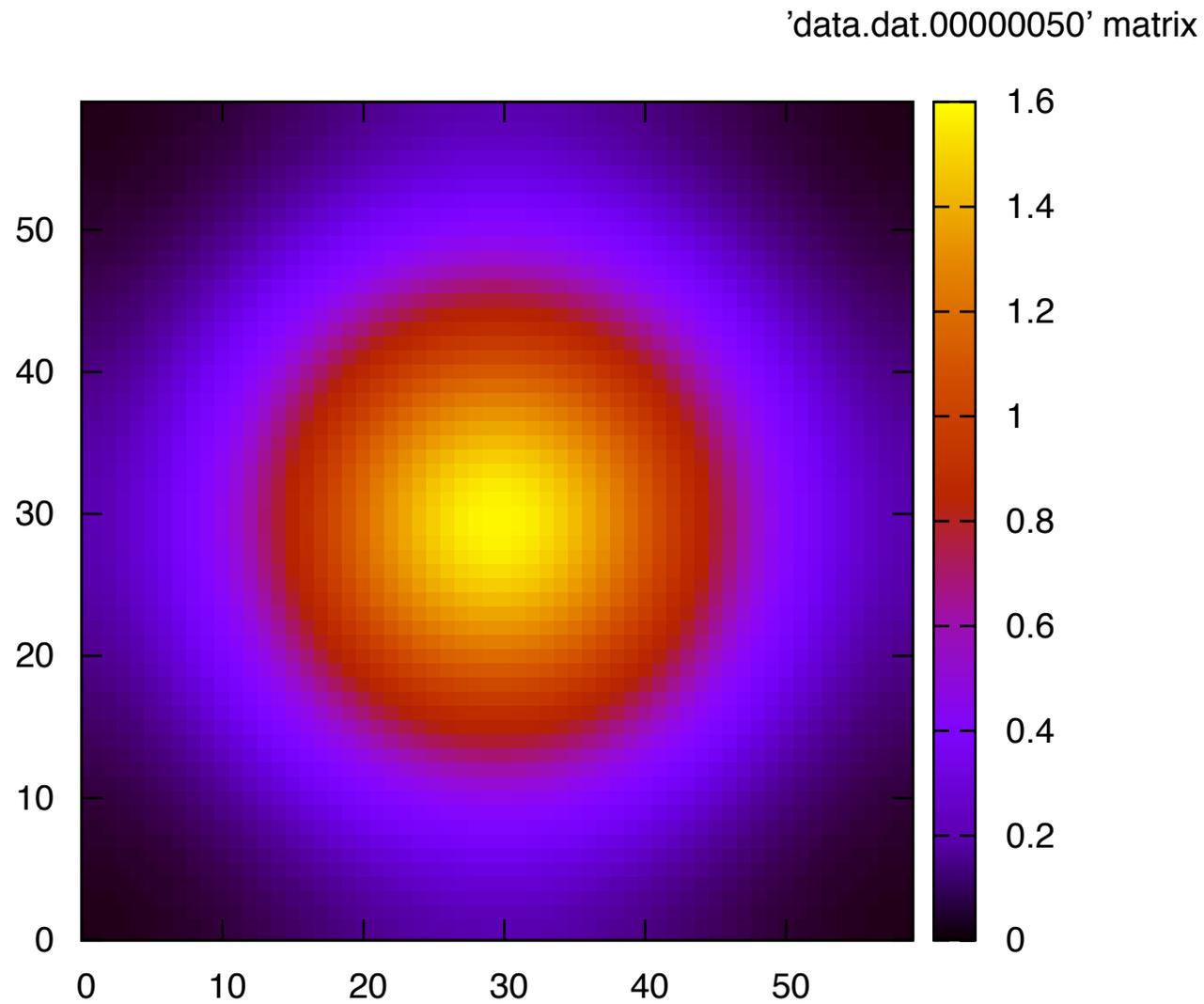
# Ex1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15

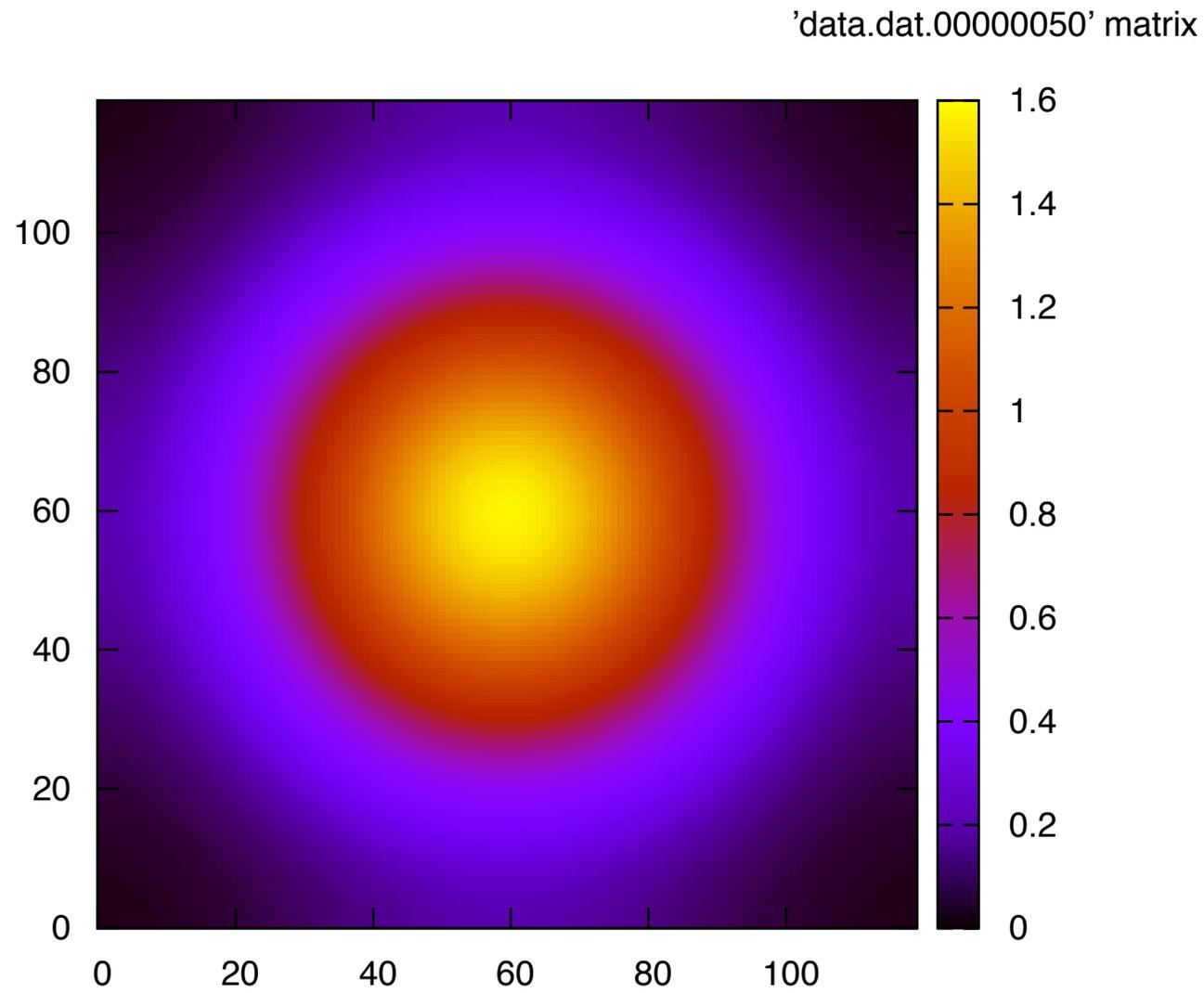


# Ex1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15









# EXAMPLE 2: TURING PATTERNS

Liu, Blätke, Heiner, Yang:  
Modelling and simulating reaction–diffusion systems using coloured Petri nets;  
Computers in Biology and Medicine, 53:297–308, October 2014

### “How the Leopard Got Its Spots”

#### □ **morphogenesis**

- > *developmental pattern formation in bio systems*
- > *the process that controls the organized spatial distribution of cells*
- > *tiger stripes, leopard spots, the precisely spaced rows of alligator teeth, etc.*

#### □ **Turing's theory of biological pattern formation, 1952**

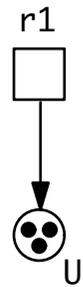
- > *patterns form as result of*  
*the interactions between two chemicals*  
*that spread throughout a system at different rates*

#### □ **highly simplified and idealised take on biological patterning**

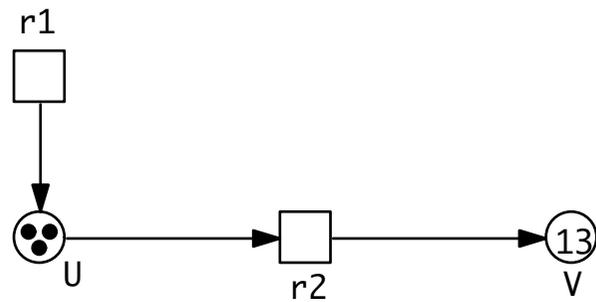
#### □ **mathematical challenge**

- > *For which parameters do stable/oscillating Turing patterns exist ?*
- > *analysis of stability, multistability, bifurcation of non-linear PDE*

r1: -> U

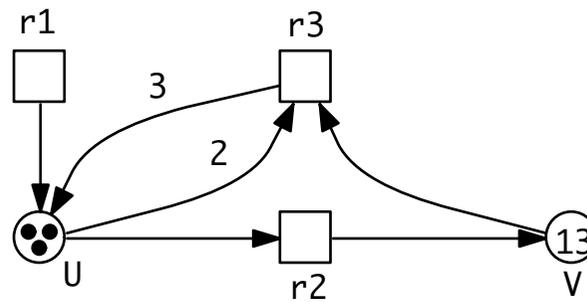


r1:        -> U  
r2:        U -> V



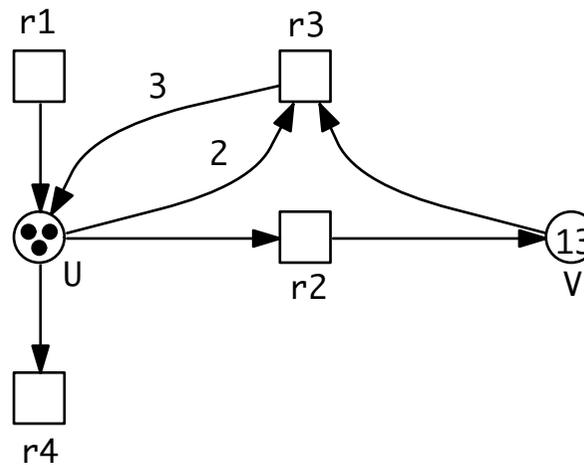
## EX2: TURING PATTERNS

r1:  $\rightarrow U$   
r2:  $U \rightarrow V$   
r3:  $2U + V \rightarrow 3U$



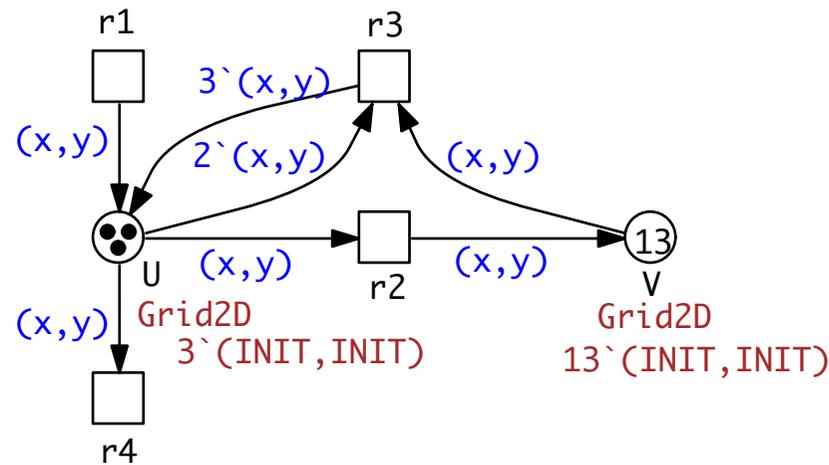
## EX2: TURING PATTERNS

r1:  $\rightarrow U$   
r2:  $U \rightarrow V$   
r3:  $2U + V \rightarrow 3U$   
r4:  $U \rightarrow$

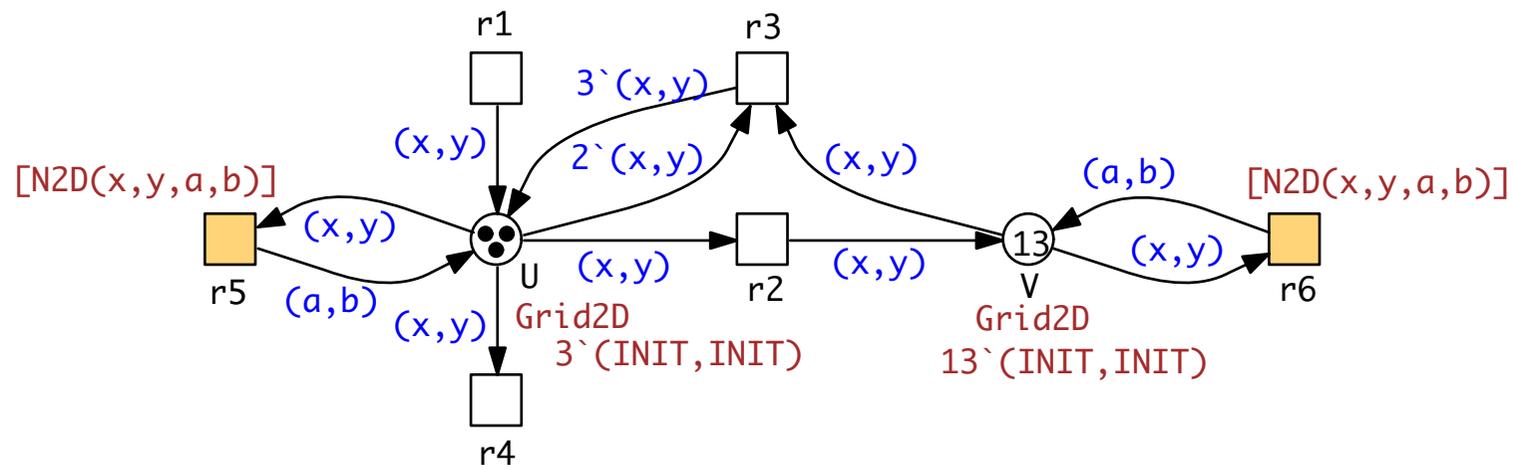
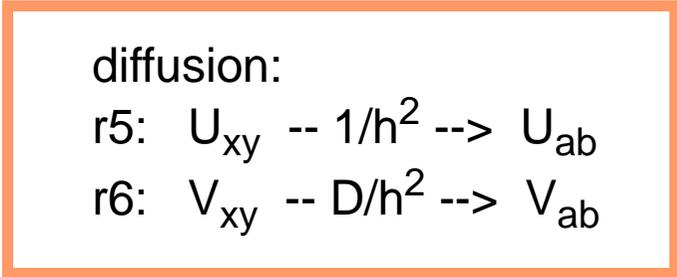
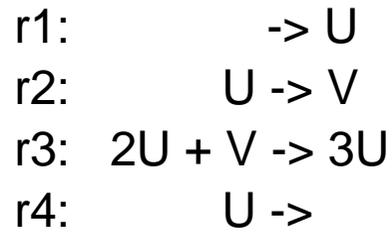


- r1:  $\rightarrow U$
- r2:  $U \rightarrow V$
- r3:  $2U + V \rightarrow 3U$
- r4:  $U \rightarrow$

## adding SPACE



# Ex2: TURING PATTERNS



*r1 - r4 follow mass action kinetics with rate constants:*  
*r1: A, r2: B, r3: 1, r4: 1;*

### □ reactions

-> *version of Brusselator model, <http://en.wikipedia.org/wiki/Brusselator>*

### □ parameters

-> *Pena, Perez-Garcia: Stability of Turing patterns in the Brusselator model, Physical Review 2001*

-> *A = 4.5, B = 0.04 ..0.98, D = 128, h = 0.8*

### □ unfolding

-> *runtime (constraint solver, 4 threads): 128 sec*

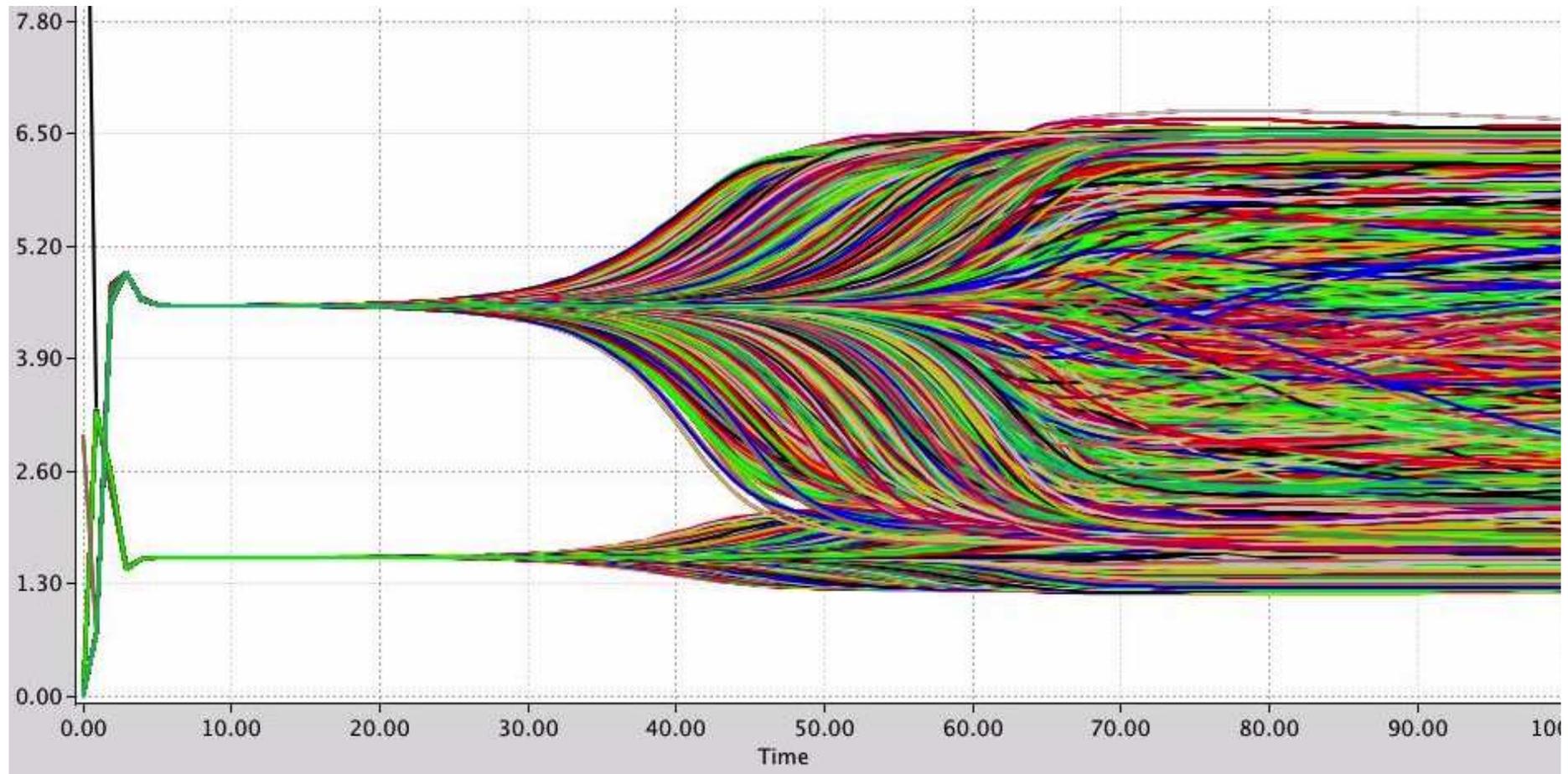
-> *places: 32,768, transitions: 324,616*

### □ continues simulation

-> *BDF (Backward Differentiation Formulae, higher-order stiffly stable solver)*

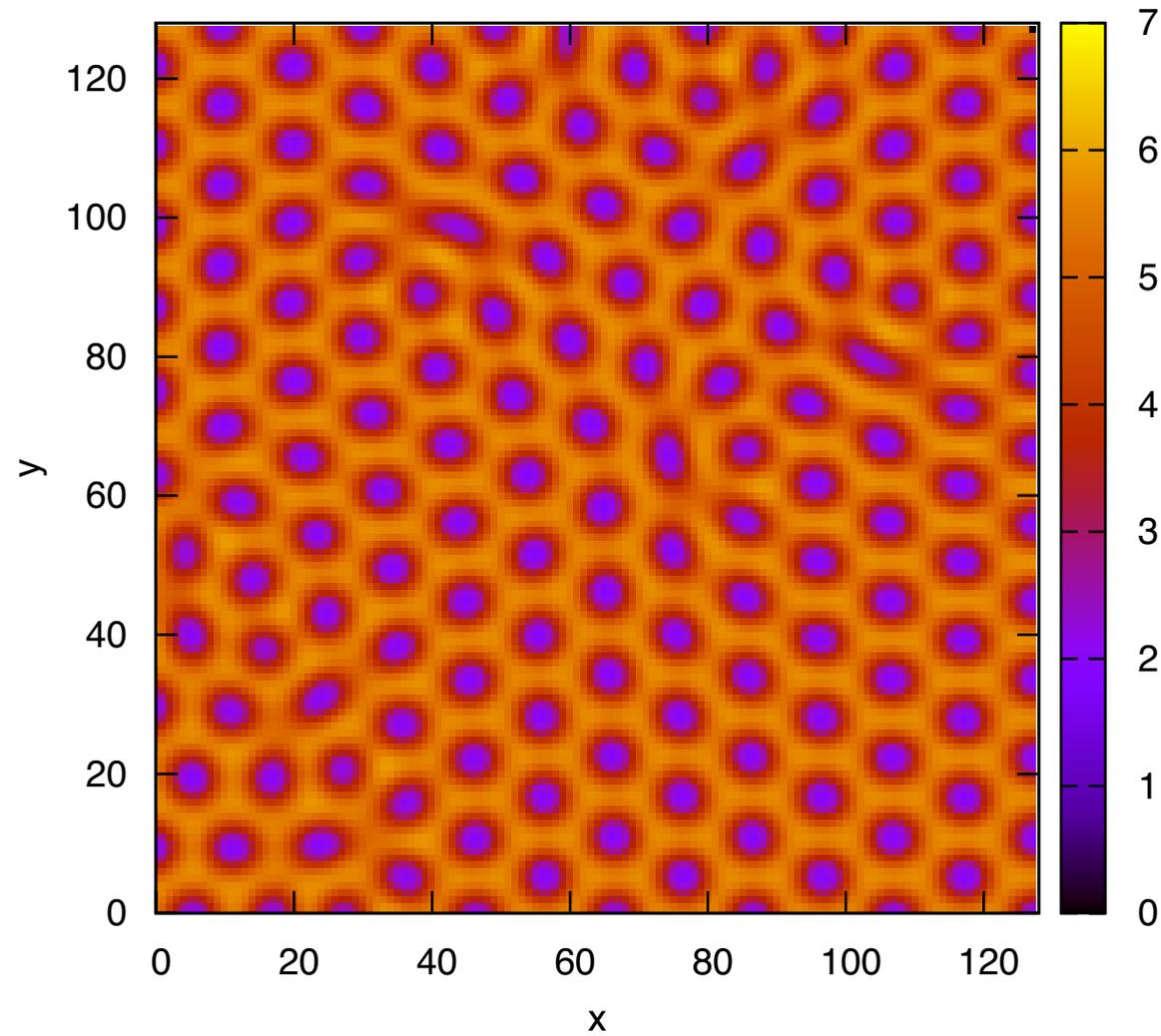
-> *simulation time: 5,000 -> runtime: about 30h*

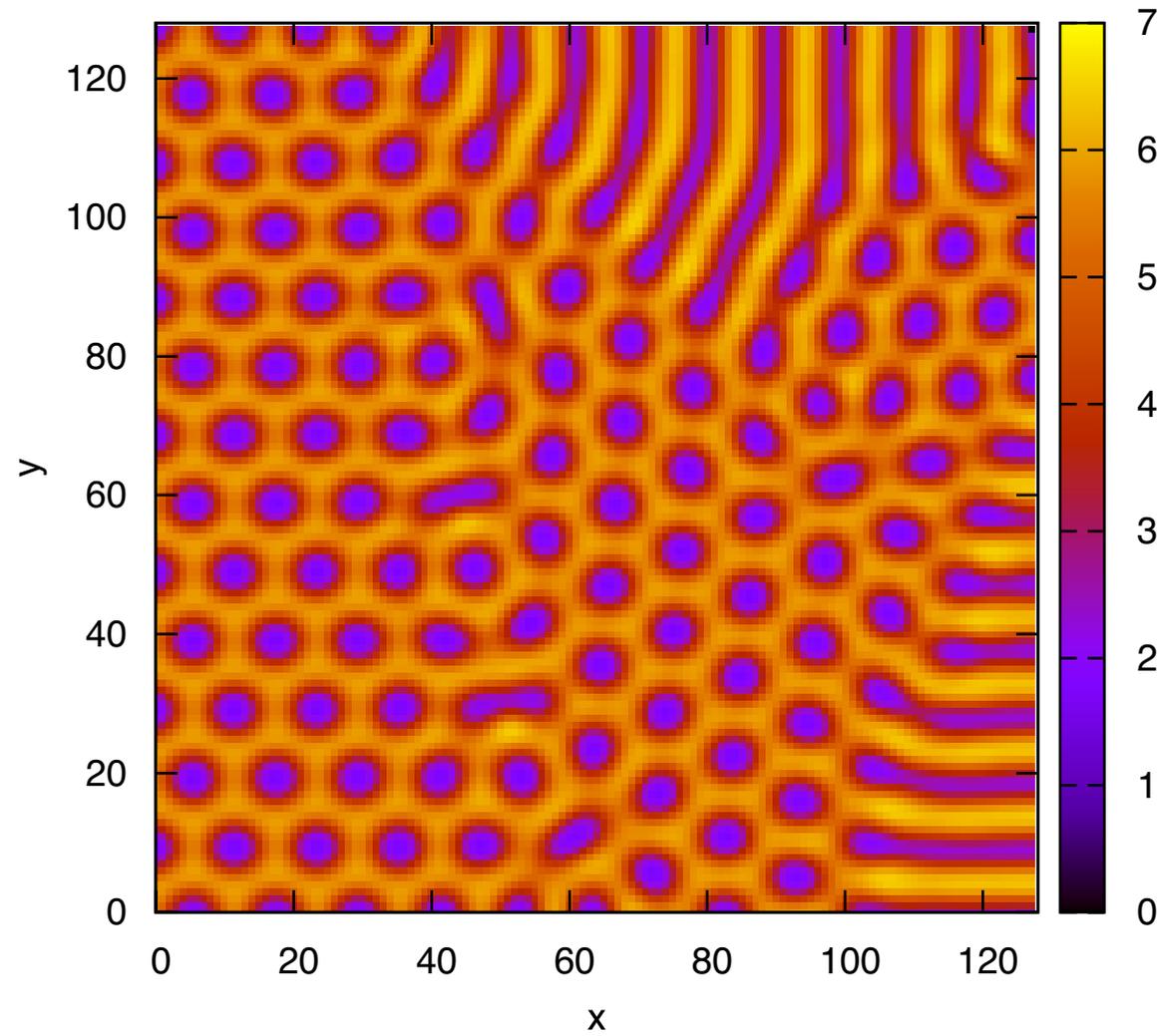
-> EVOLUTION OF  $U, V$  IN ALL GRID POSITIONS OVER TIME

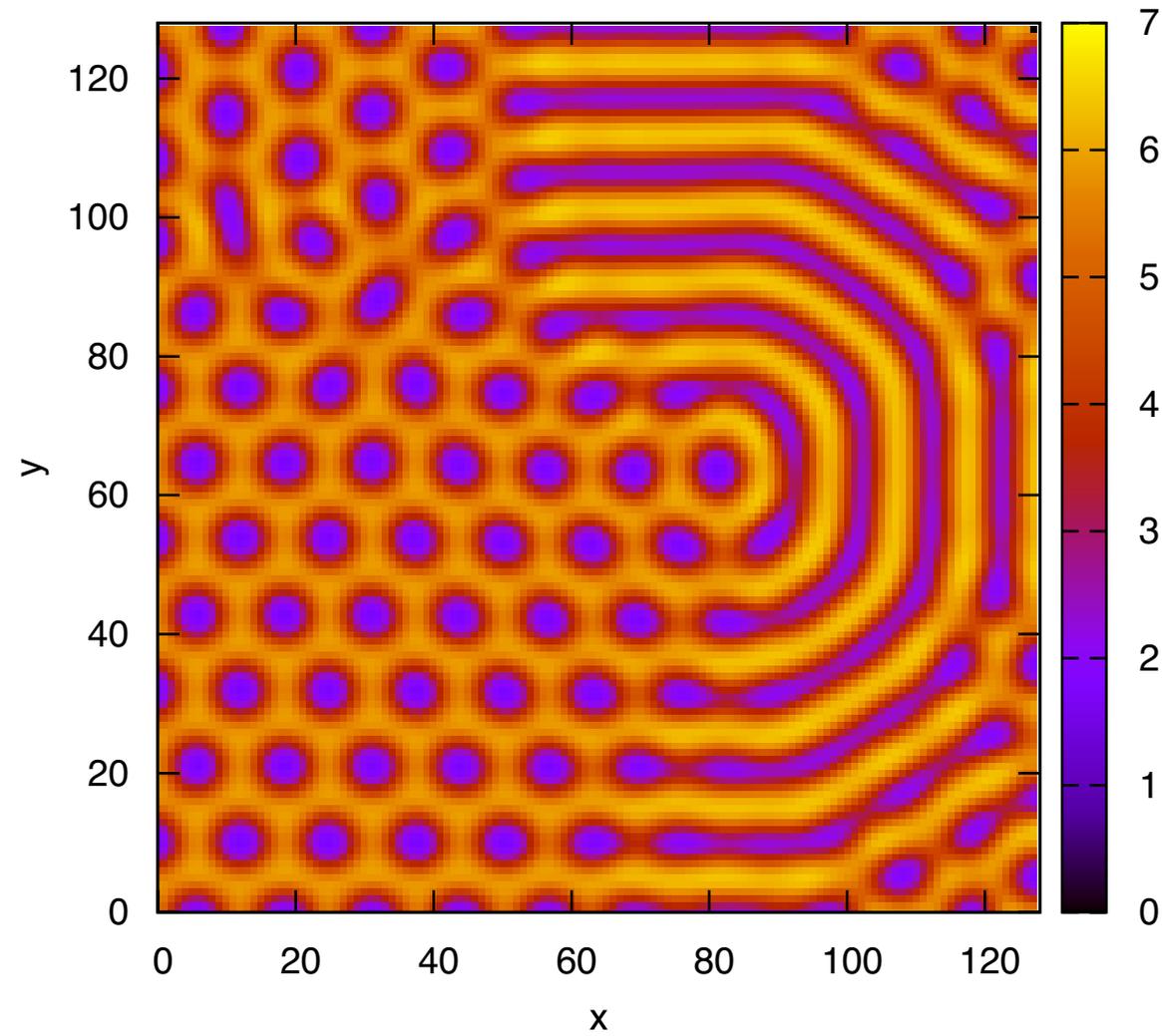


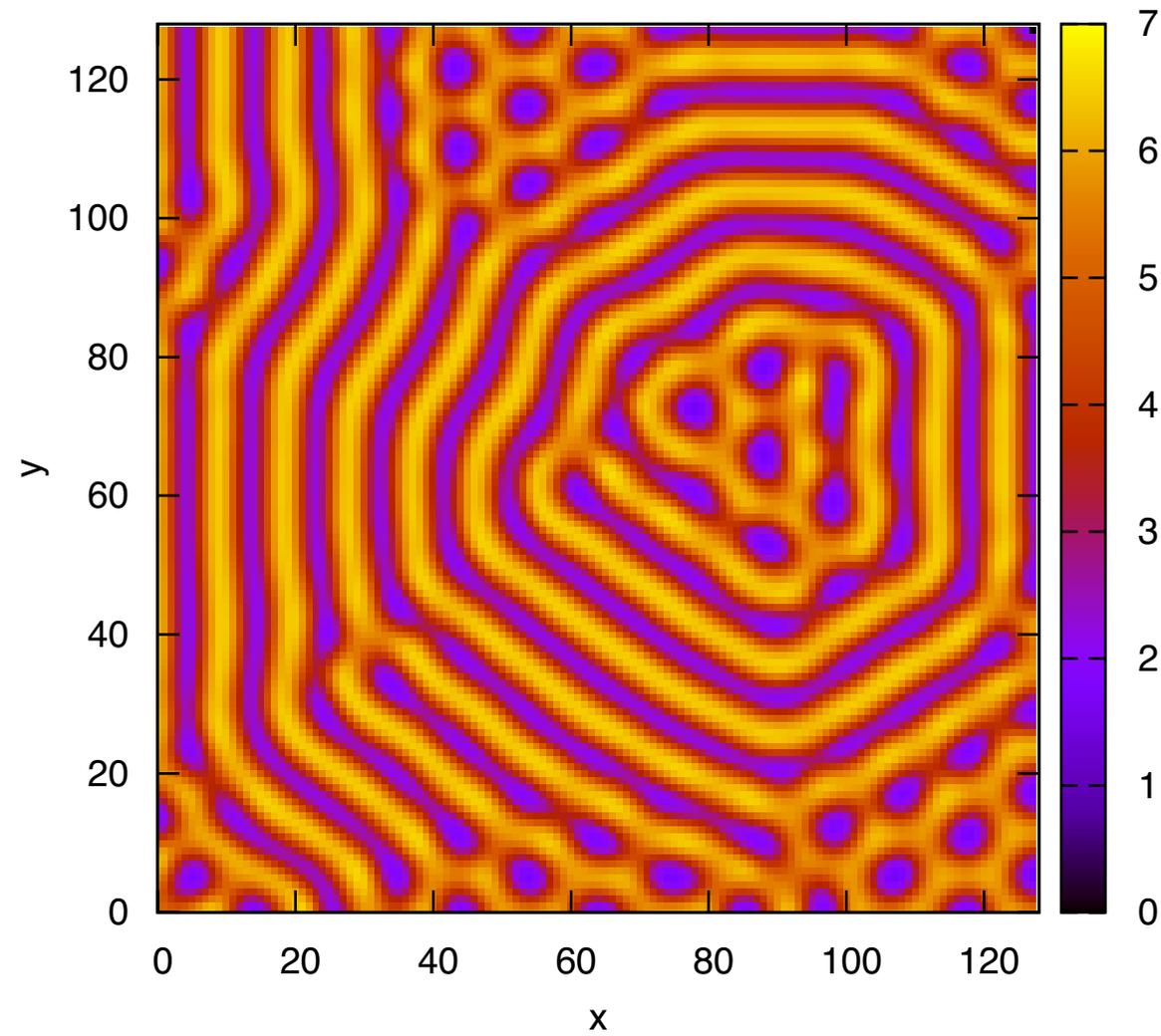
## Ex2: TURING PATTERNS

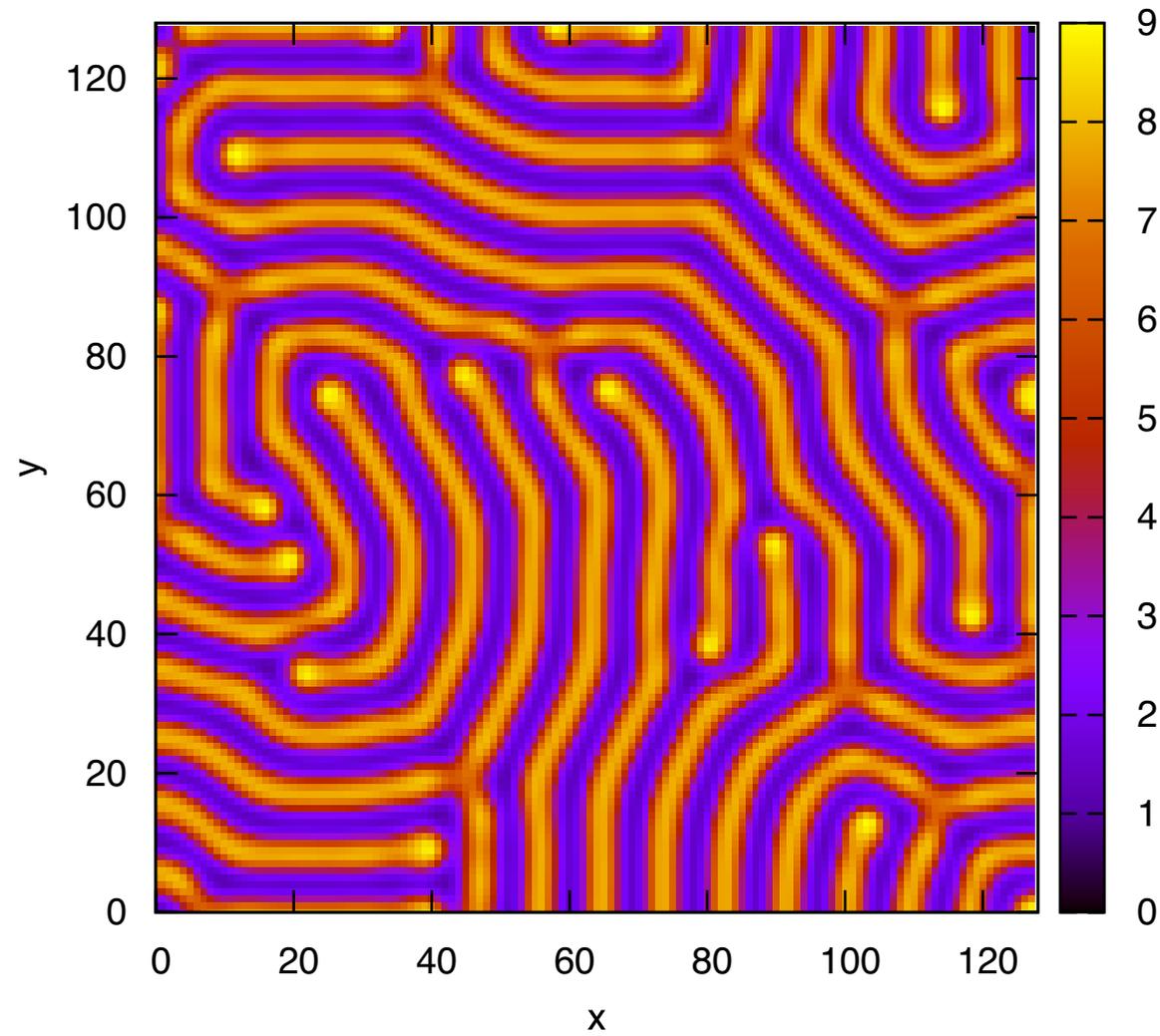
-> U IN ALL GRID POSITIONS AT TIME 5,000

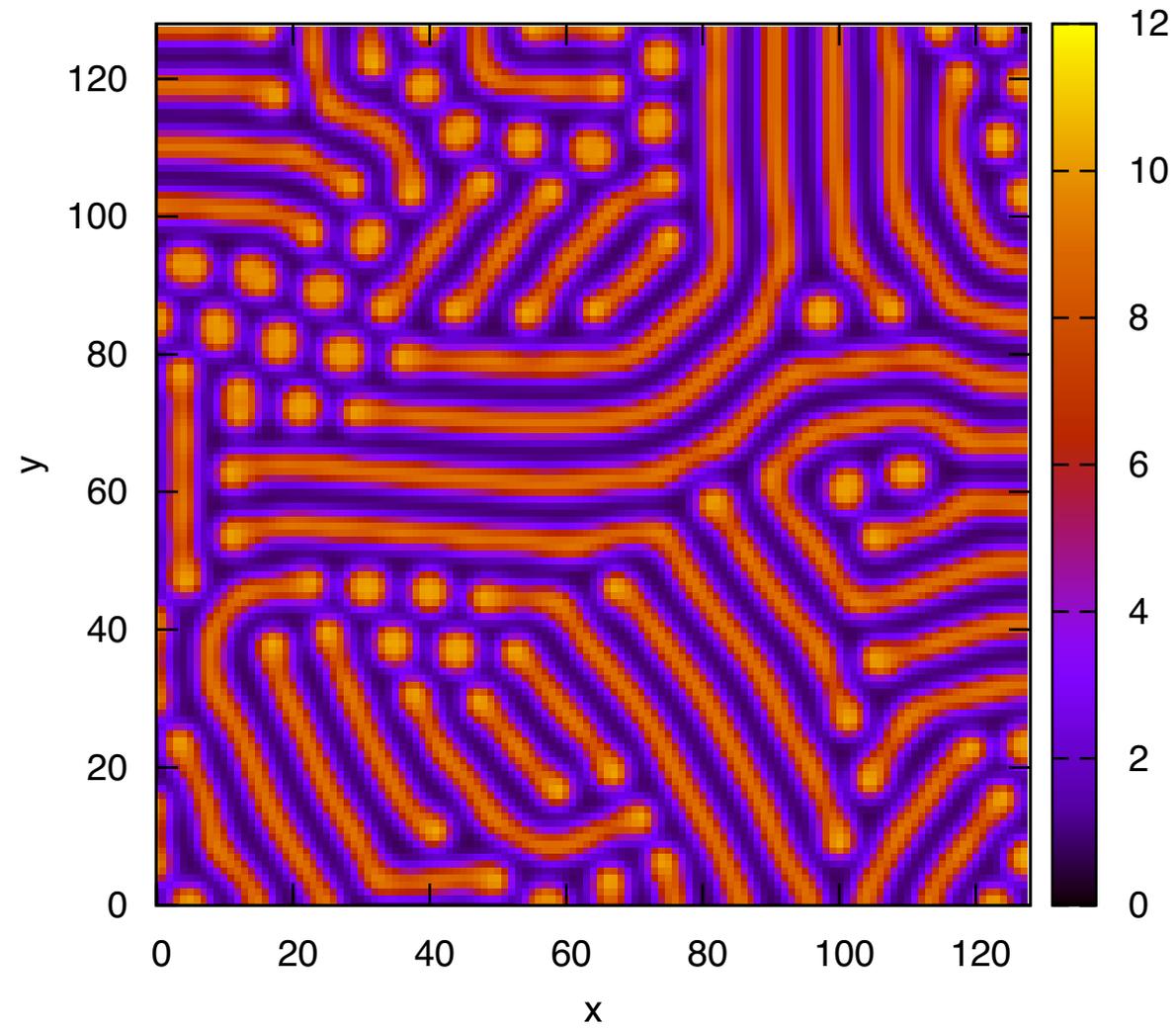


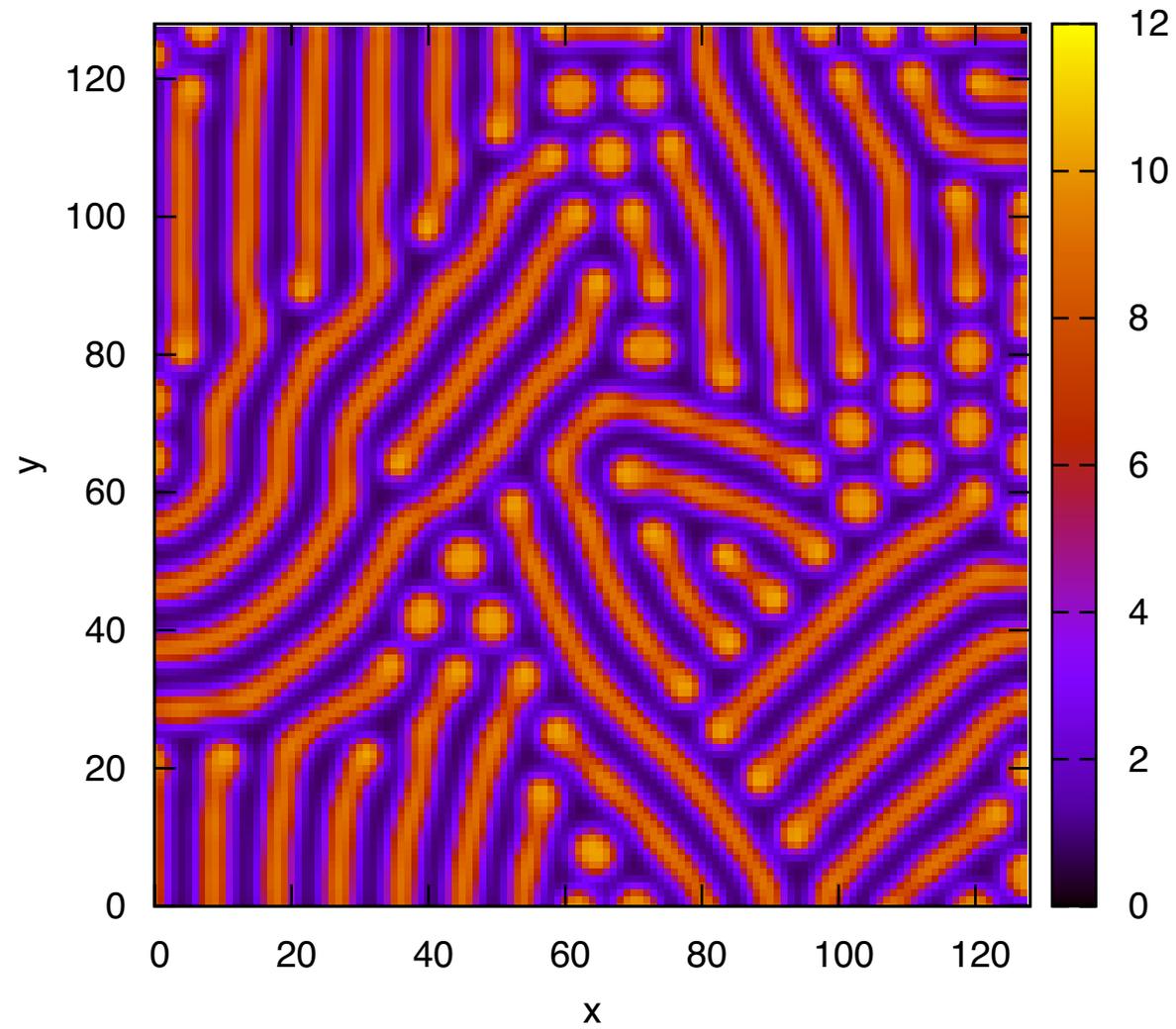


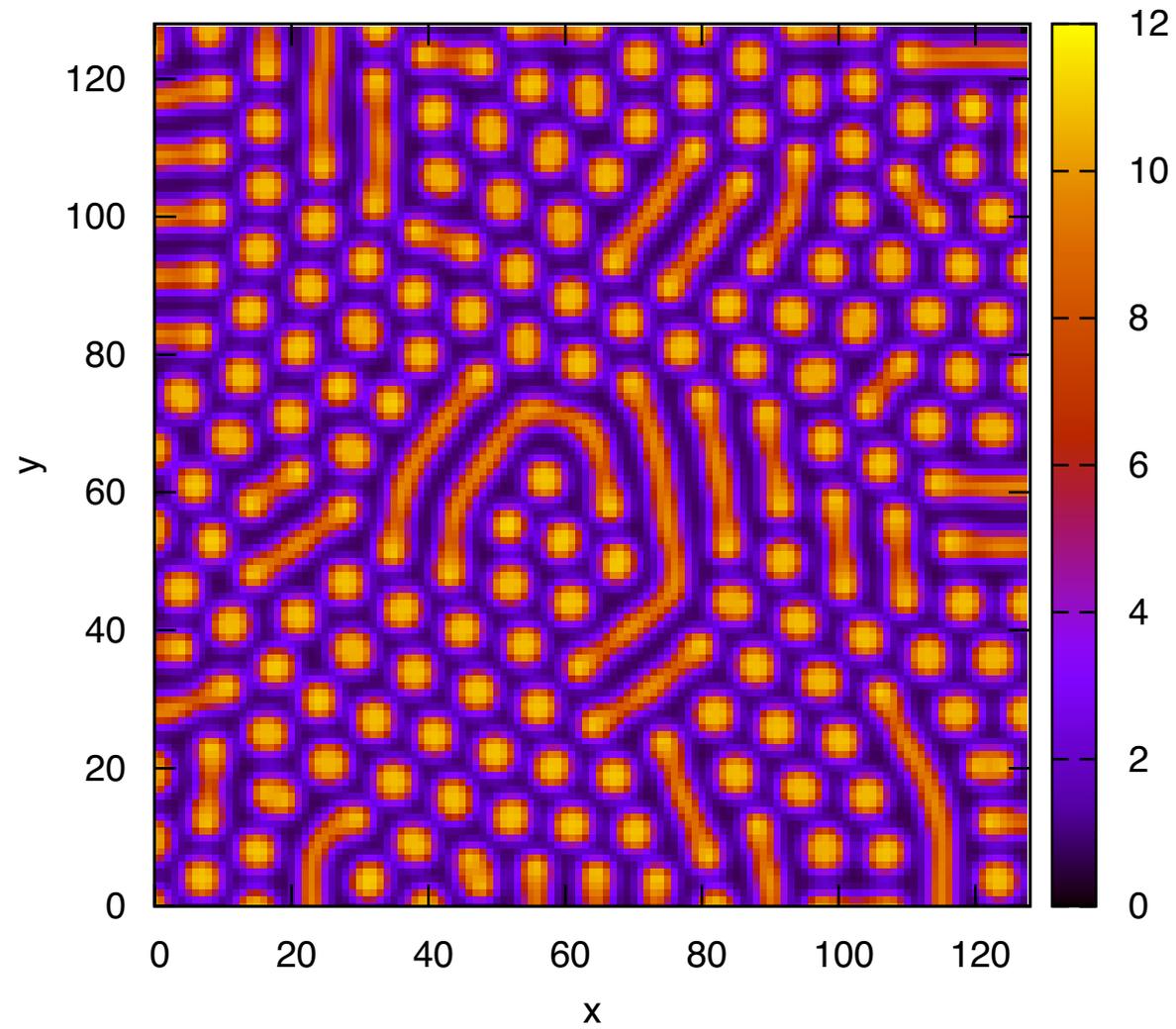


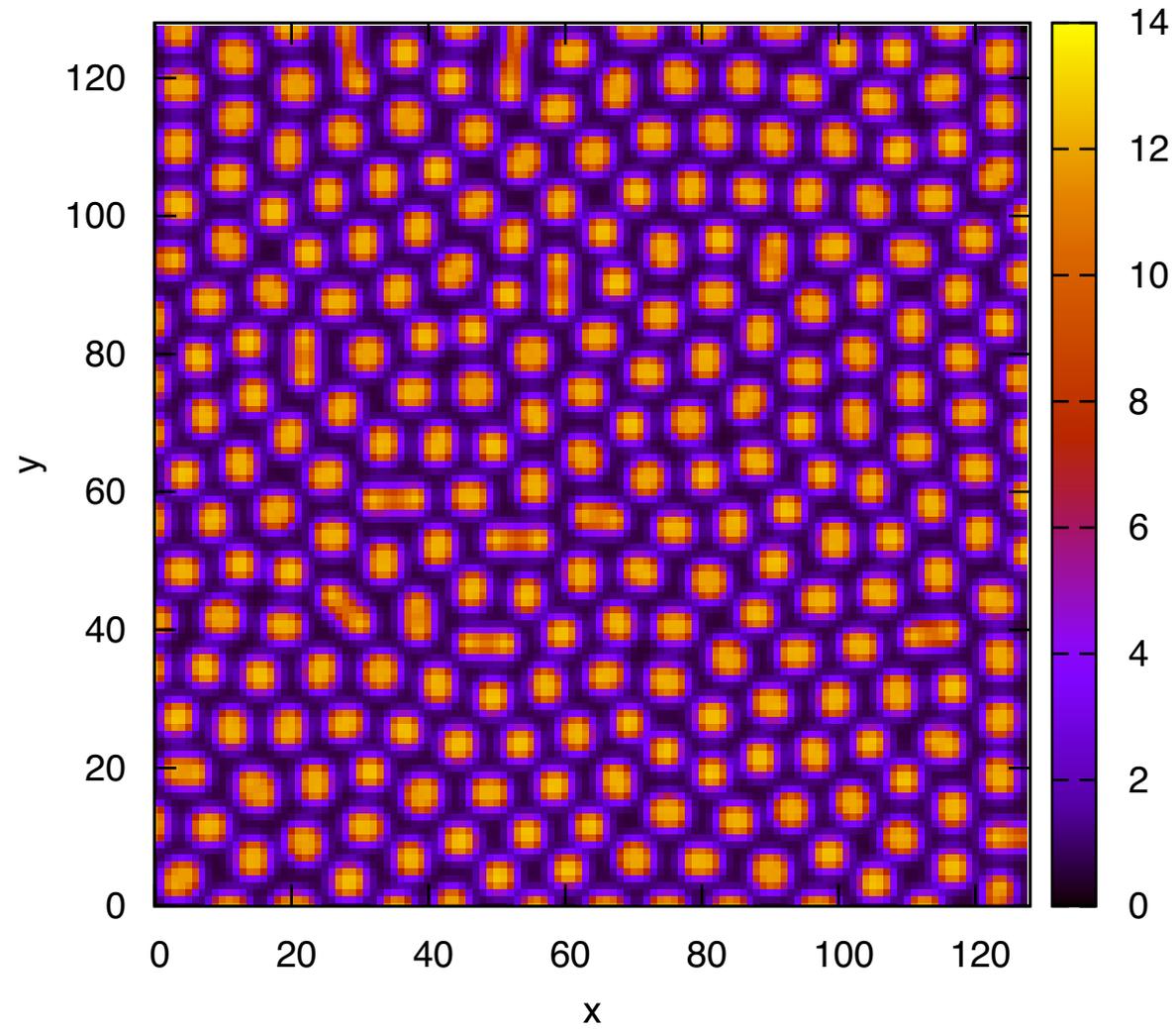








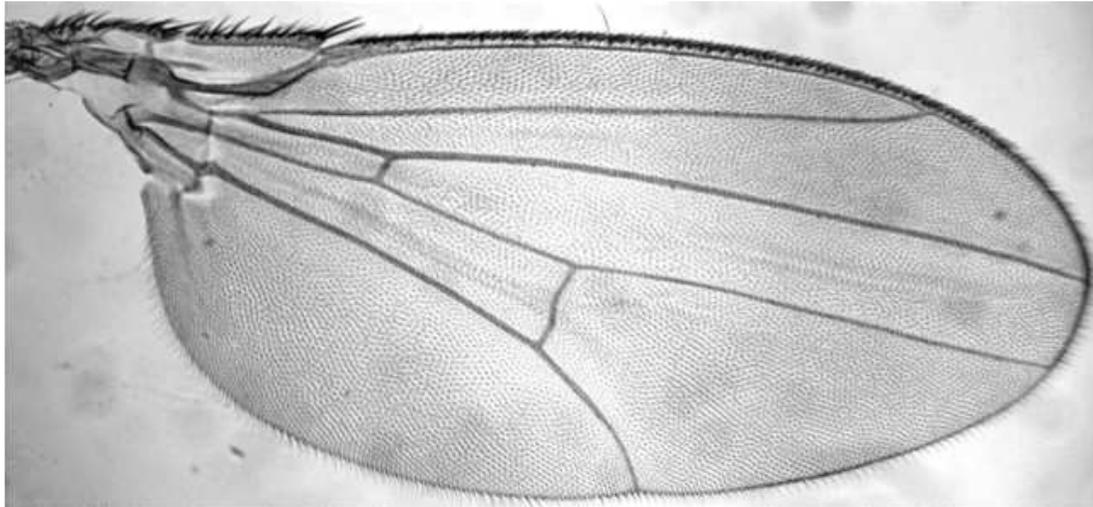




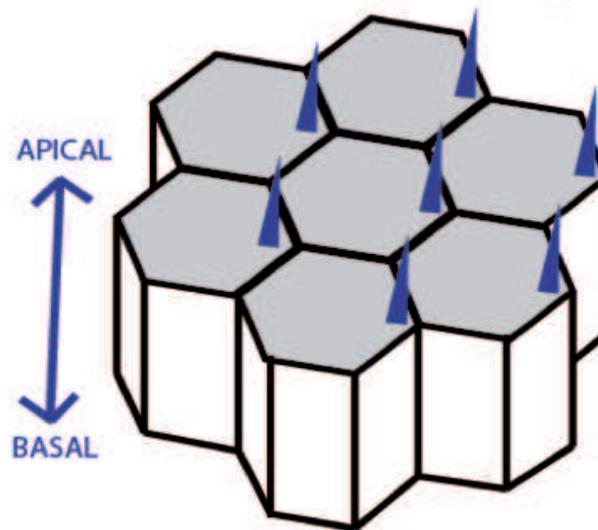
# EXAMPLE 3: PLANAR CELL POLARITY IN FLY WING

Gao, Gilbert, Heiner, Liu, Maccagnola, Tree:  
Multiscale Modelling and Analysis of Planar Cell Polarity in the Drosophila Wing;  
IEEE/ACM Transactions on Computational Biology and Bioinformatics, 10(2):337-351, 2013

# EX3 - PLANAR CELL POLARITY

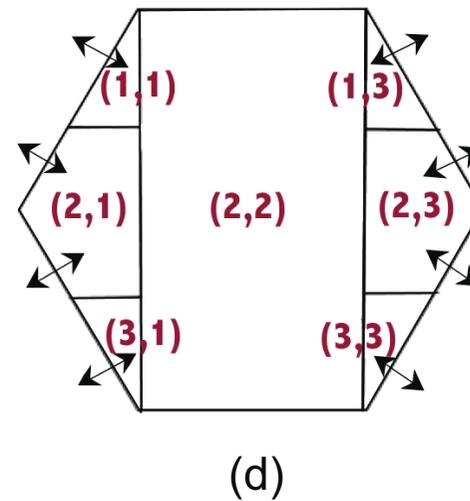
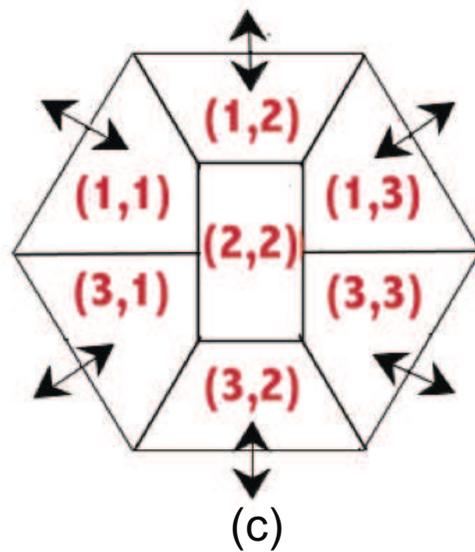
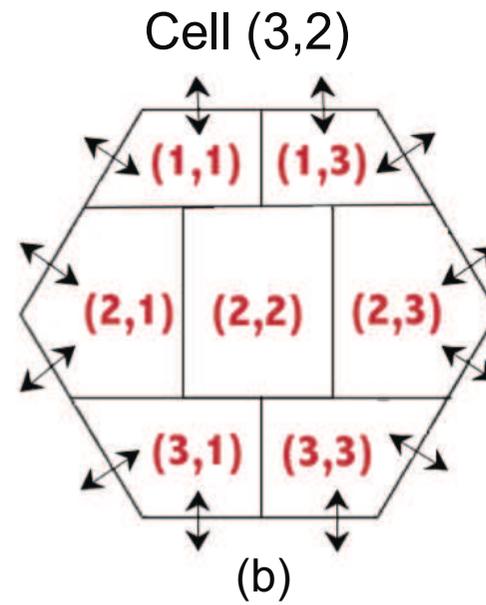
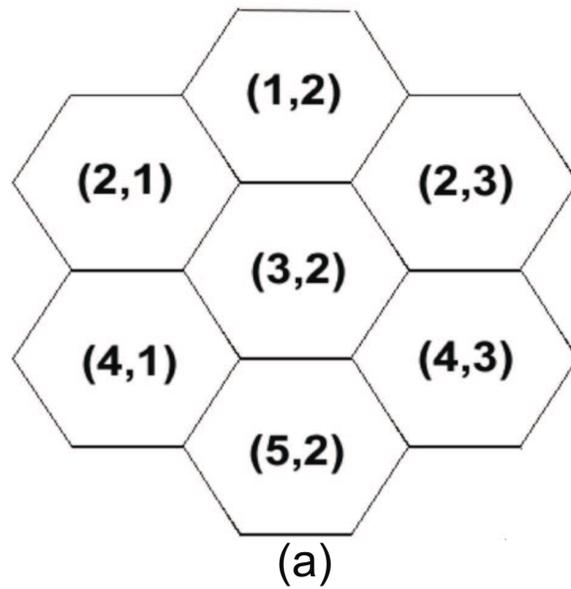


PROXIMAL ← → DISTAL

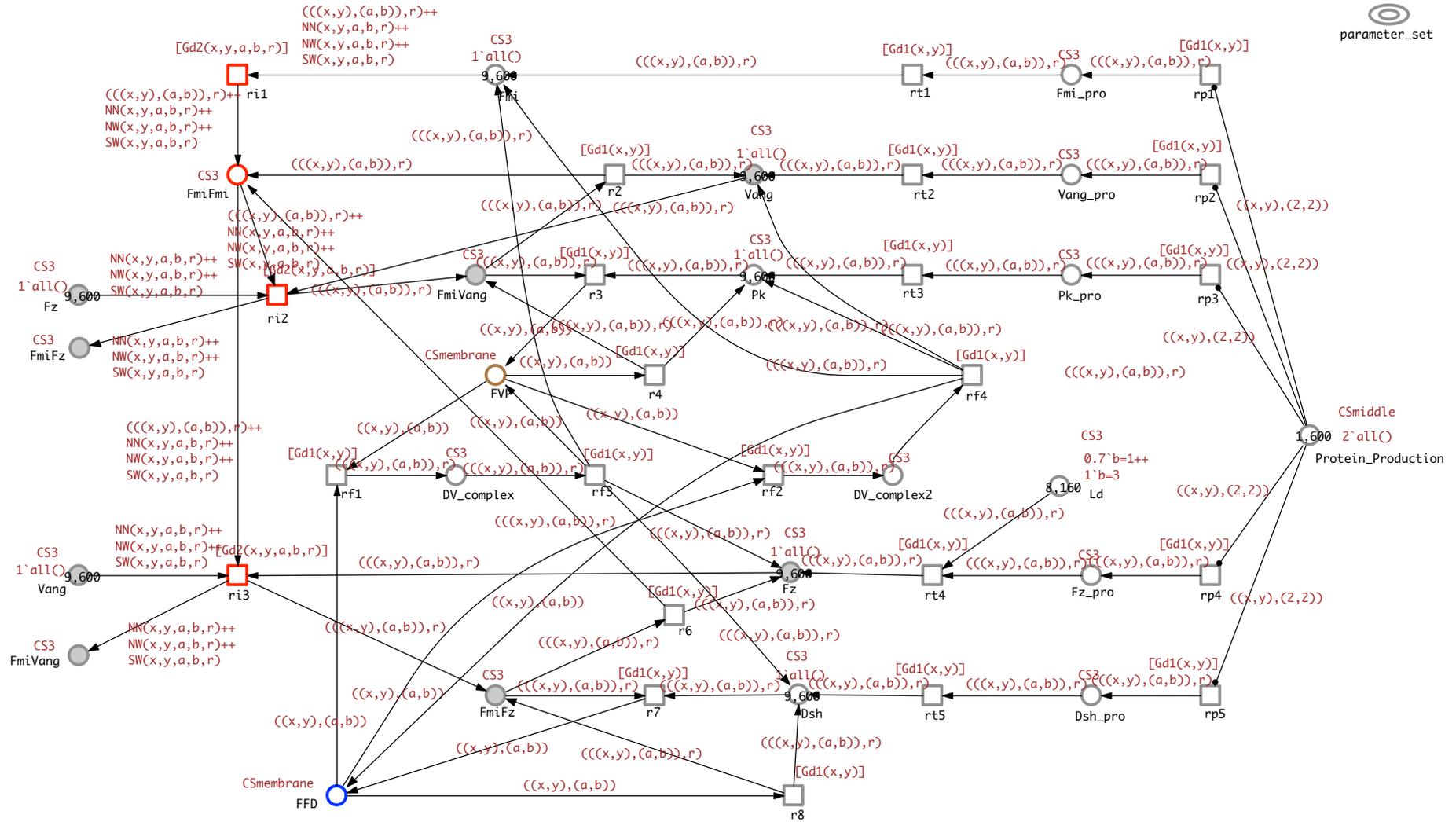


[BioPPN 2011]  
[CMSB 2011]

# EX3 - PLANAR CELL POLARITY

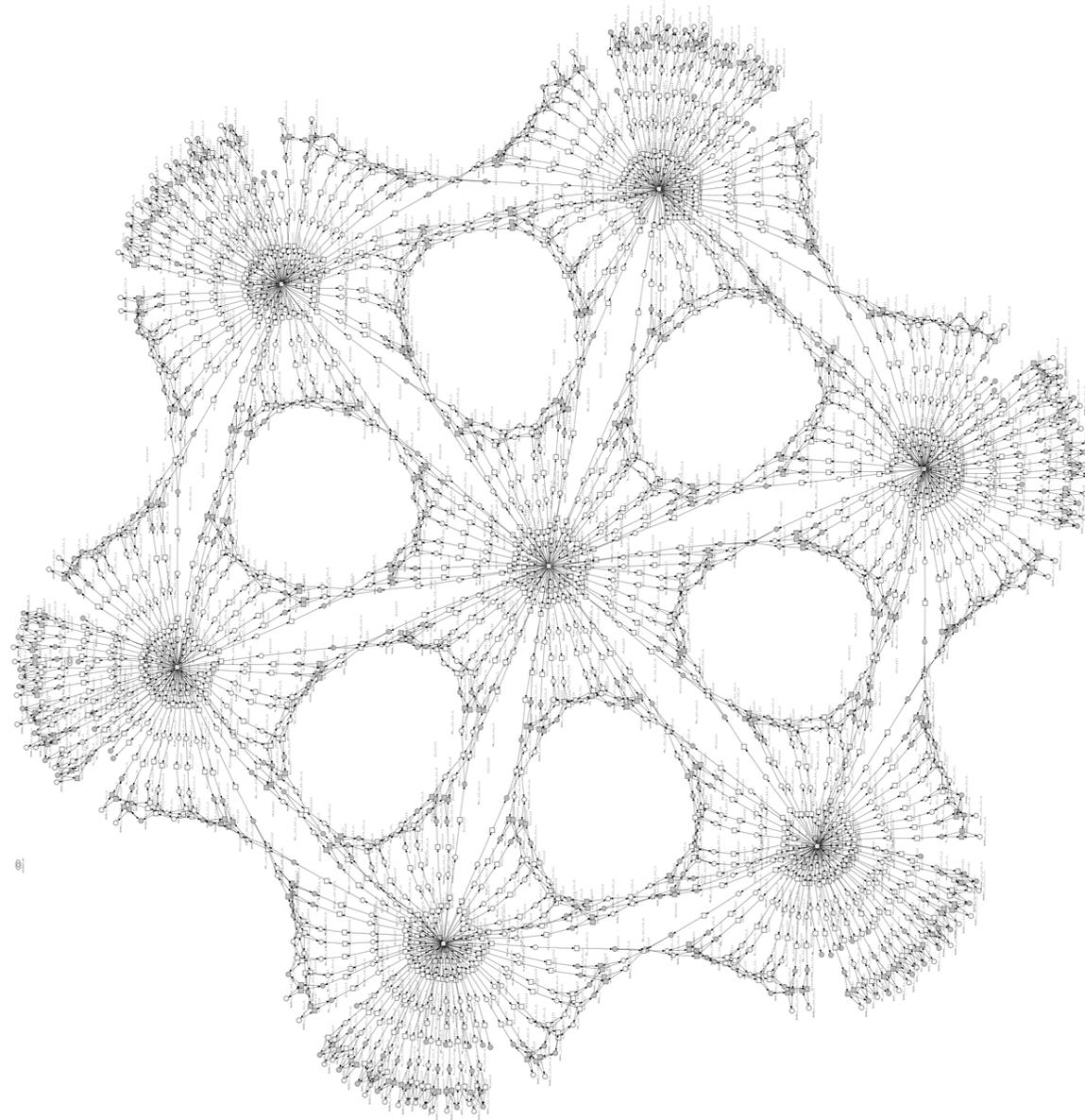


# EX3: PLANAR CELL POLARITY

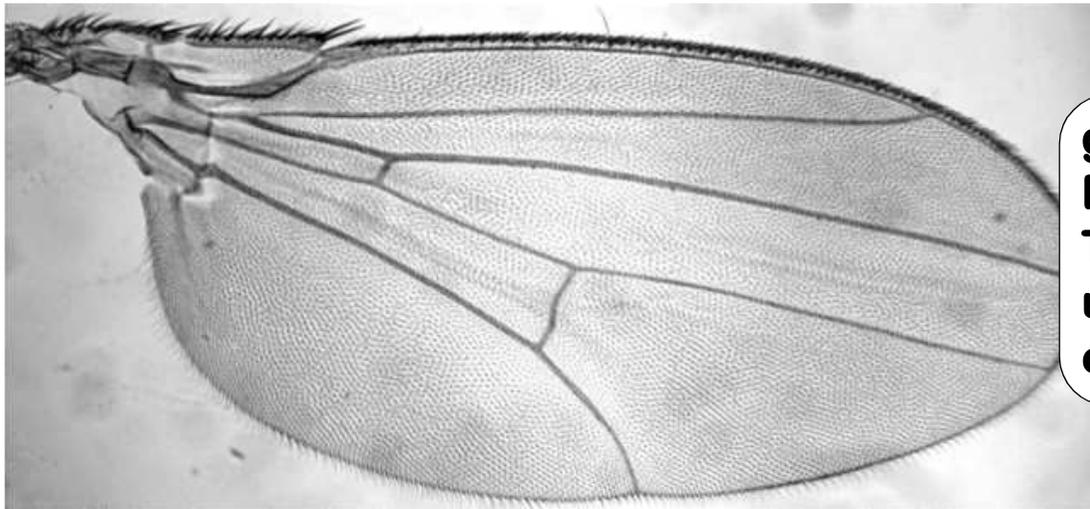


[QIAN GAO, PHD THESIS 2013]

# EX3: PLANAR CELL POLARITY, PLAIN MODEL (7 CELLS)

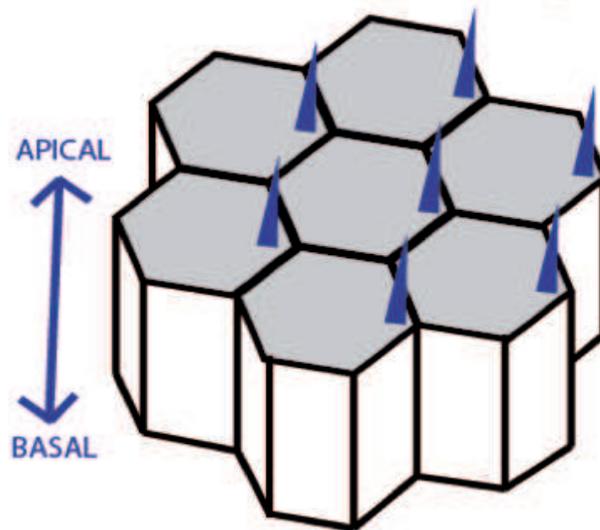


# EX3 - PLANAR CELL POLARITY



<b>grid size:</b>	<b>40 x 40</b>
<b>PLACES:</b>	<b>164,000</b>
<b>TRANSITIONS:</b>	<b>229,686</b>
<b>unfolding:</b>	<b>4 min</b>
<b>cont. simulation:</b>	<b>2 h</b>

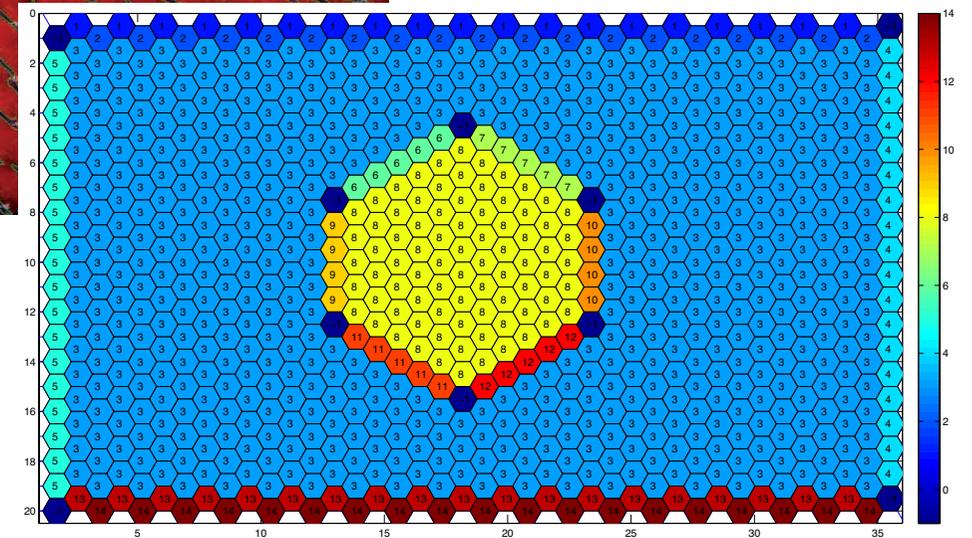
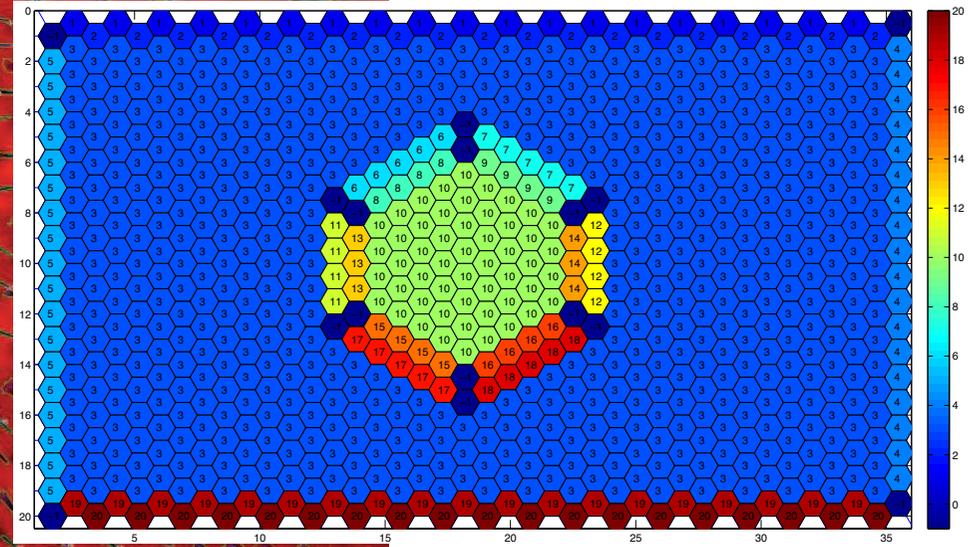
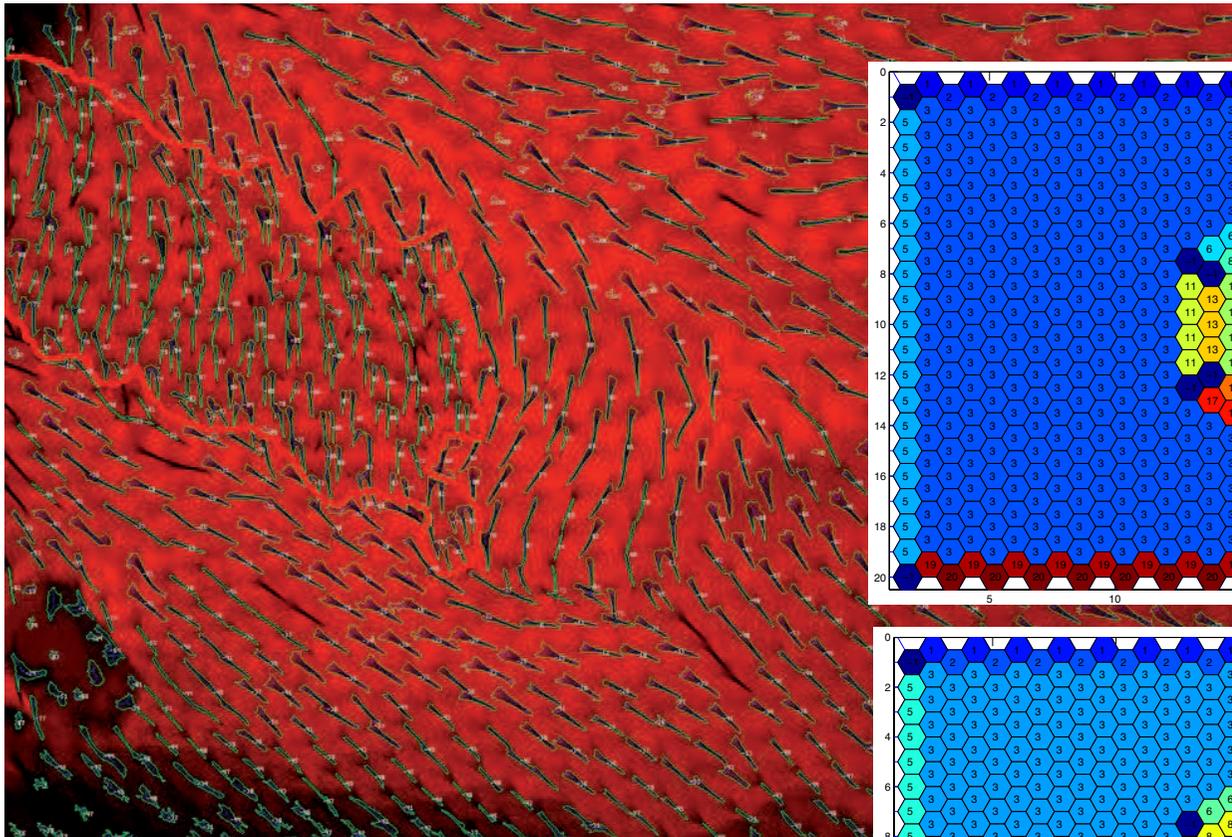
PROXIMAL ← → DISTAL



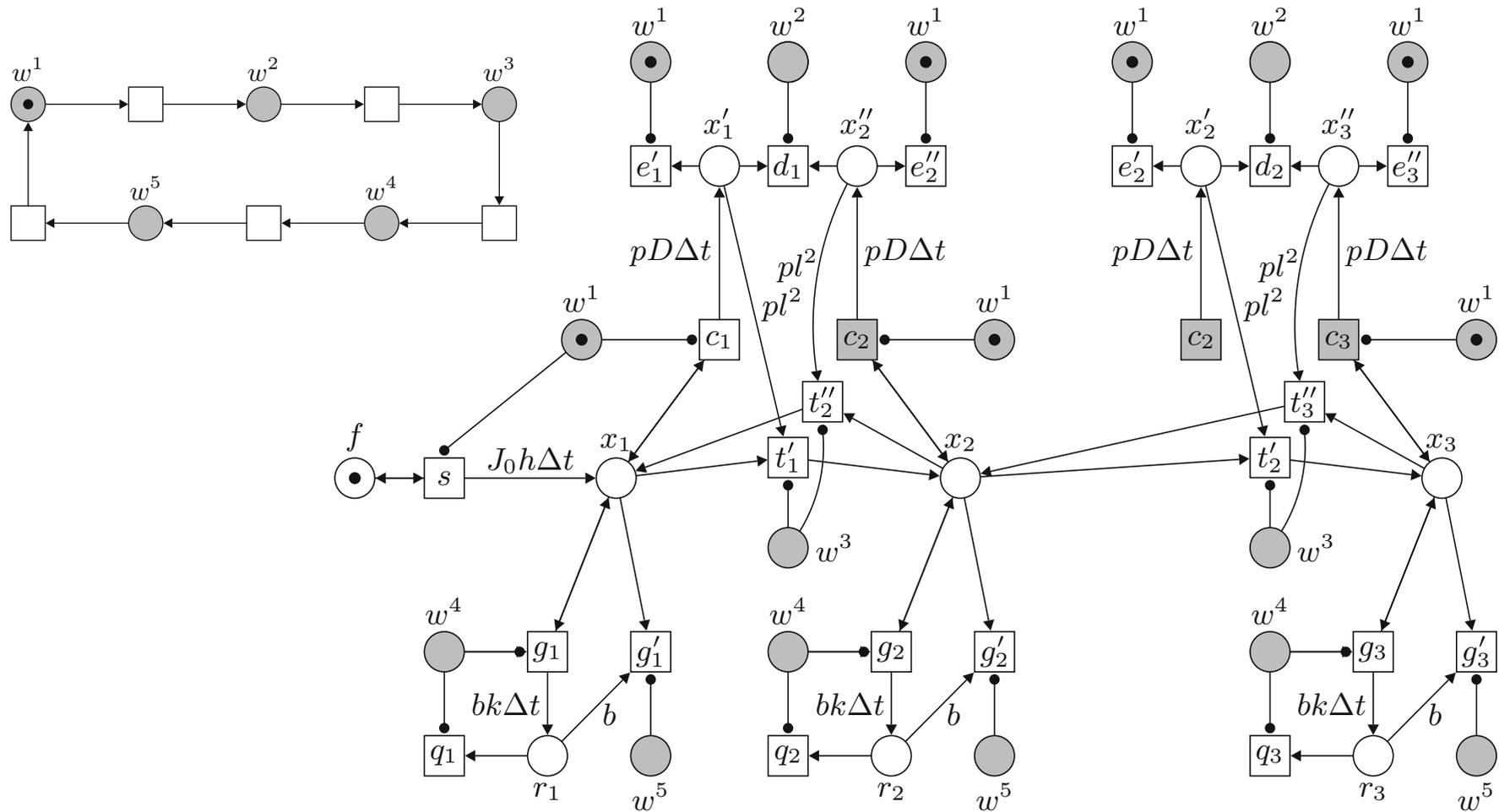
[BioPPN 2011]  
[CMSB 2011]



# EX3 - PLANAR CELL POLARITY



[TCBB 2012]



Bertens, Kleijn, Hille, Heiner, Koutny, Verbeek:  
 Modeling biological gradient formation: combining partial differential equations and Petri nets;  
 Natural Computing 2015.

## ❑ FRAMEWORK

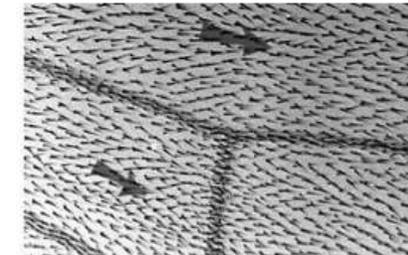
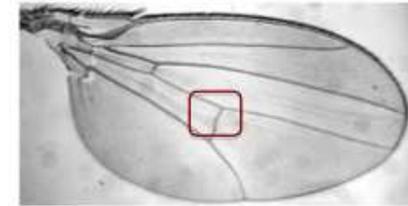
- > QPN - SPN - CPN - HPN
- > uncoloured / coloured

## ❑ MODELLING BIO PETRI NETS

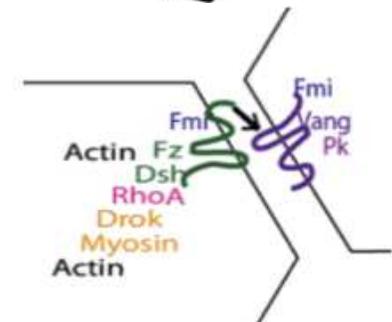
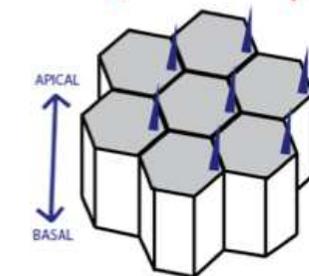
## ❑ COLOUR

- > gradients
- > Turing pattern
- > fly wing

## ❑ WHAT NEXT ?



PROXIMAL ← → DISTAL



## ❑ FRAMEWORK

- > QPN - SPN - CPN - HPN
- > uncoloured / coloured

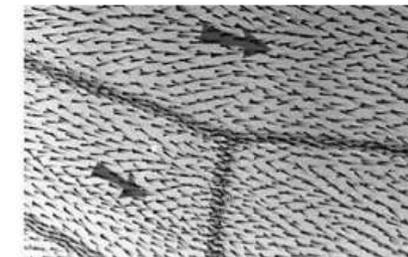
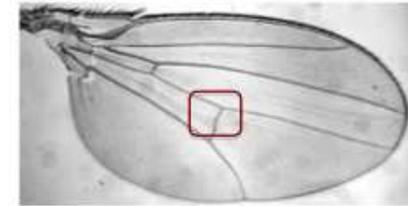
## ❑ MODELLING BIO PETRI NETS

## ❑ COLOUR

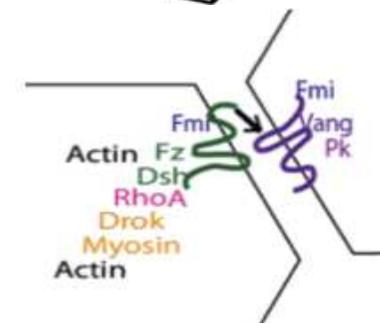
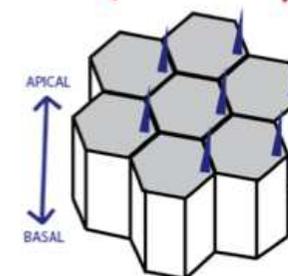
- > gradients
- > Turing pattern
- > fly wing

## ❑ WHAT NEXT ?

-> MODELLING 4 ANALYSING



PROXIMAL ← → DISTAL



## ❑ FRAMEWORK

- > QPN - SPN - CPN - HPN
- > uncoloured / coloured

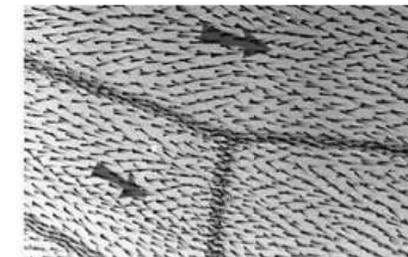
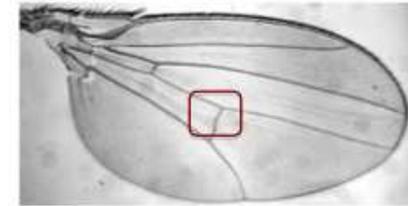
## ❑ MODELLING BIO PETRI NETS

## ❑ COLOUR

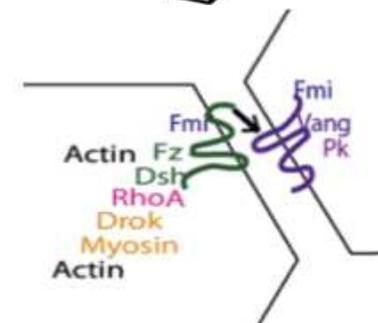
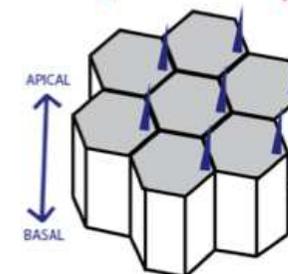
- > gradients
- > Turing pattern
- > fly wing

## ❑ WHAT NEXT ? -> MODELLING 4 ANALYSING

- > how to analyse coloured Petri nets ?
- > model checking
- > efficient hybrid simulation

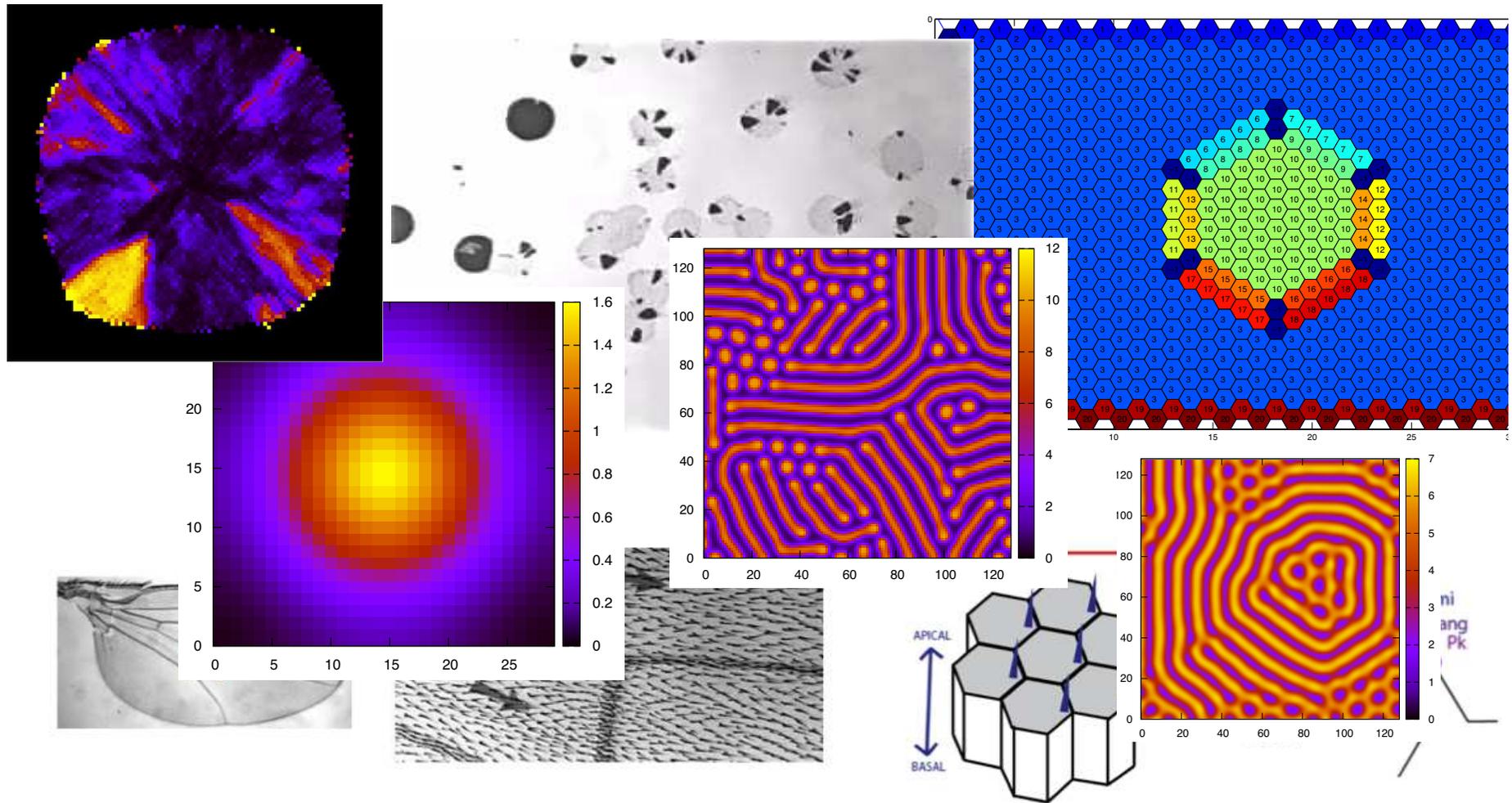


PROXIMAL ← → DISTAL



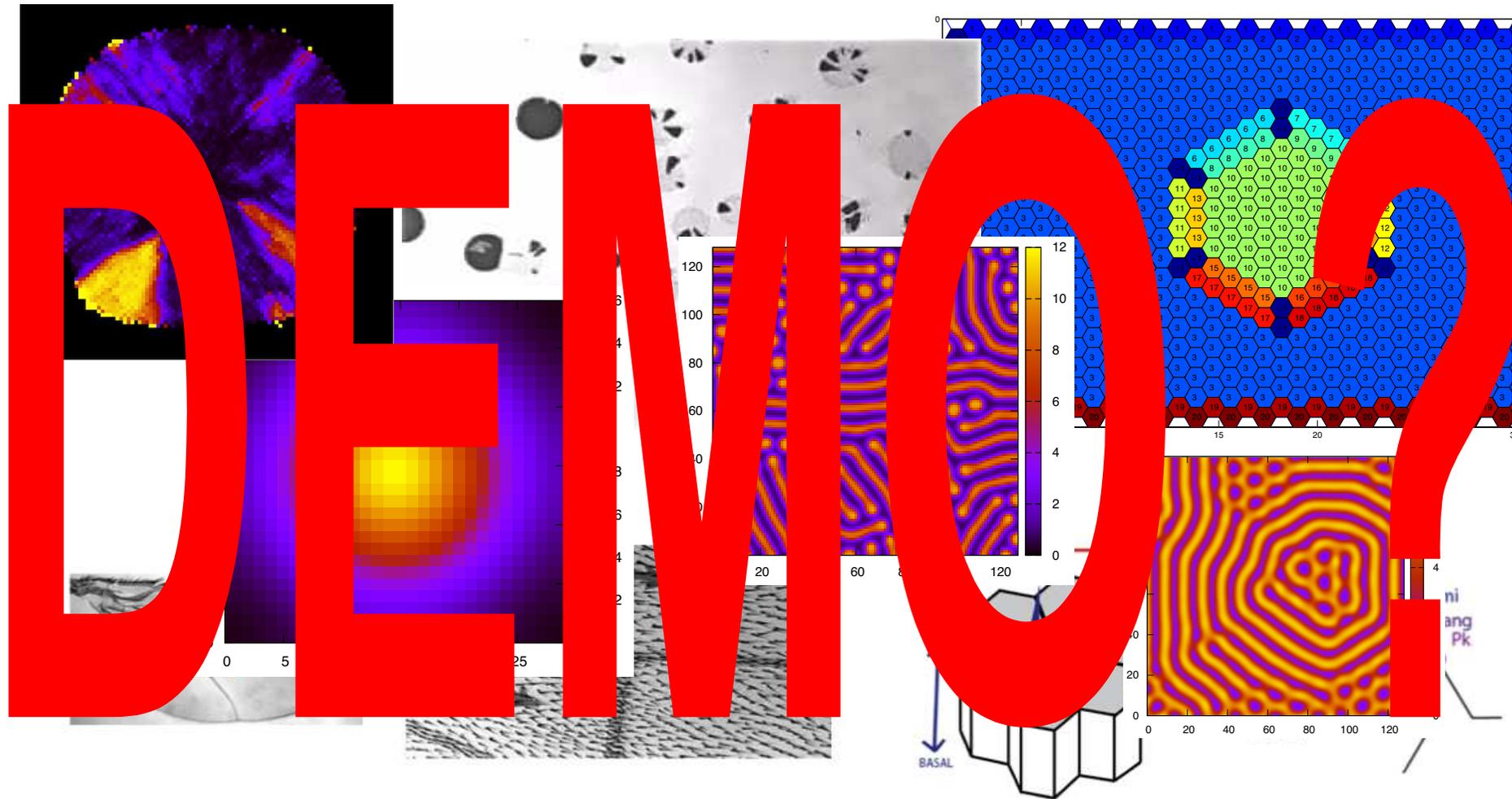
- ❑ **David Gilbert**  
Brunel University London, UK
  
- ❑ **Wolfgang Marwan, Mary Ann Blätke**  
Otto-von-Guericke University Magdeburg,  
Germany
  
- ❑ **Fei Liu**  
Harbin University, China
  
- ❑ **Mostafa Herajy**  
Port Said Universit, Egypt
  
- ❑ **Jetty Kleijn, Fons Verbeek**  
Leiden University, NL





[HTTP://WWW-DSSZ.INFORMATIK.TU-COTTBUS.DE](http://www-dssz.informatik.tu-cottbus.de)

[HTTP://MULTISCALEPN.BRUNEL.AC.UK](http://multiscalepn.brunel.ac.uk)



[HTTP://WWW-DSSZ.INFORMATIK.TU-COTTBUS.DE](http://www-dssz.informatik.tu-cottbus.de)

[HTTP://MULTISCALEPN.BRUNEL.AC.UK](http://multiscalepn.brunel.ac.uk)