

# MULTI-VARIABLE MARKOV PROCESS: A MATRIX PRODUCT STATES TREATMENT

COLLABORATION:

---

BIN XU – PHYSICS

PEIQI WANG – ORFE

LIANGSHENG ZHANG – PHYSICS

PETER GJELTEMA – MAE

JUN XIONG – PHYSICS

# CONTENTS

- Transitional matrix method of Markov process
- Matrix product states
- The Algorithm and Program structures
- Some results

# MARKOV PROCESS

- An emotional boy: happy or angry
- IF he is happy:
  - 50% -> Angry
  - 50% -> Remain happy
- IF he is angry:
  - 50% -> Remain angry
  - 50% -> Happy

Transitional Matrix

$$v_{t+1} = T \cdot v_t$$

$$\begin{pmatrix} 0.5P_{Angry} + 0.5P_{Happy} \\ 0.5P_{Angry} + 0.5P_{Happy} \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} P_{Angry} \\ P_{Happy} \end{pmatrix}$$

# MARKOV PROCESS

Transitional Matrix

- 2 emotional boys
- With probability  $p$ , they will change their states **TOGETHER**
- Otherwise, they remain in their states

$$\begin{pmatrix} 1-p & 0 & 0 & p \\ 0 & 1-p & p & 0 \\ 0 & p & 1-p & 0 \\ p & 0 & 0 & 1-p \end{pmatrix} \begin{pmatrix} P_{AA} \\ P_{AH} \\ P_{HA} \\ P_{HH} \end{pmatrix}$$

$$T = (1-p)I + p\sigma^X \otimes \sigma^X$$

$$\sigma^X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

# MARKOV PROCESS

- N emotional boys
- With probability p, a pair nearest neighbors will change states TOGETHER
- Otherwise, they remain their states

Transitional Matrix

$$T = pI + (1 - p) \sum_{i=1}^{n-1} \frac{1}{n-1} \sigma_i^x \otimes \sigma_{i+1}^x$$

$$\sigma_i^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The T matrix is  $2^n$  dimensional!

# MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA \dots AA) \\ P(AAA \dots AH) \\ P(AAA \dots HA) \\ P(AAA \dots HH) \\ \vdots \\ P(HHH \dots HH) \end{pmatrix}$$

$2^N$  states

- Read as a function

$$P : \{A, H\}^{\otimes N} \rightarrow \mathbb{R}$$

- How do we represent a function?
  - Table of value
  - Analytical relations
    - (Taylor expansion, etc.)
  - Algebraic relations
    - (product of matrices)

# MATRIX PRODUCT STATES (MPS)

- Algebraic representation of the “state function”

$$P : \{A, H\}^{\otimes N} \rightarrow \mathbb{R}$$

$$\sigma_1 \sigma_2 \cdots \sigma_N \mapsto M_{\sigma_1}^{[1]} M_{\sigma_2}^{[2]} M_{\sigma_3}^{[3]} \cdots M_{\sigma_N}^{[N]}$$

- We need to store  $2N$  matrices instead of  $2^N$  numbers
- for example:

$$P(AHA \cdots HA) = M_A^{[1]} M_H^{[2]} M_A^{[3]} \cdots M_H^{[N-1]} M_A^{[N]}$$

- In the memory

$M_A^{[1]}$	$M_A^{[2]}$	$\dots$	$M_A^{[N]}$
$M_H^{[1]}$	$M_H^{[2]}$	$\dots$	$M_H^{[N]}$

# MATRIX PRODUCT STATES (MPS)

- Algebraic representation of the “state function”

$$P : \{A, H\}^{\otimes N} \rightarrow \mathbb{R}$$

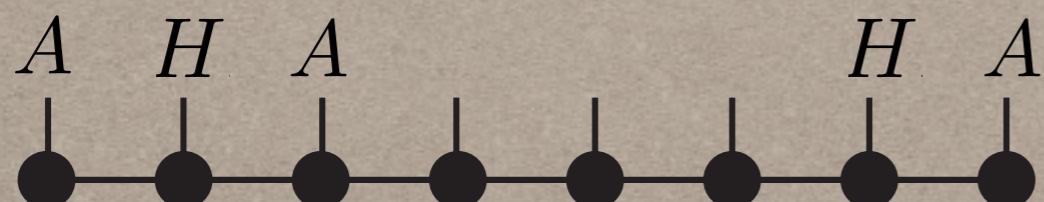
$$\sigma_1 \sigma_2 \cdots \sigma_N \mapsto M_{\sigma_1}^{[1]} M_{\sigma_2}^{[2]} M_{\sigma_3}^{[3]} \cdots M_{\sigma_N}^{[N]}$$

- In the memory

$M_A^{[1]}$	$M_A^{[2]}$	$\dots$	$M_A^{[N]}$
$M_H^{[1]}$	$M_H^{[2]}$	$\dots$	$M_H^{[N]}$

- Pictorial representation

$$P(AHHA \cdots HA) = M_A^{[1]} M_H^{[2]} M_A^{[3]} \cdots M_H^{[N-1]} M_A^{[N]}$$

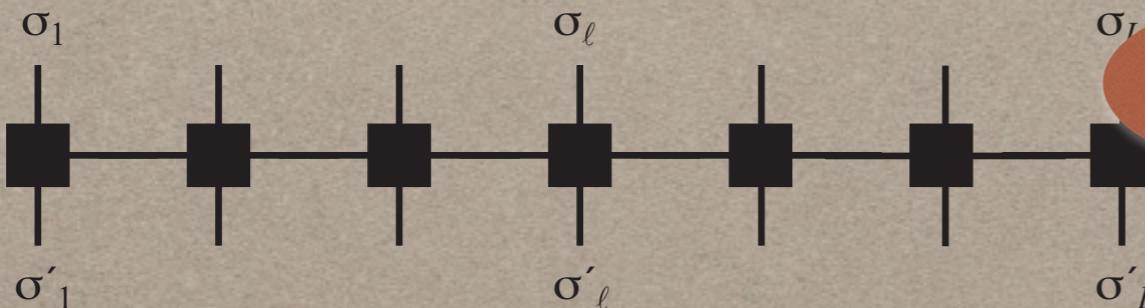


# MATRIX PRODUCT STATES (MPS)

- So we solve an exponentially hard problem in polynomial time!
- “Really? That sounds too good to be true!”
- I cheated a little bit: the dimension of these matrices (bound dimension) will be in general  $2^N$
- But for many “interesting” problems, a rather small matrix approximates the state very nicely!
- We are just exploiting the hidden structure of the model!
  - Remember image compression using Fourier analysis or wavelet? This is the same logic!

# MATRIX PRODUCT OPERATORS (MPO)

- The transitional matrix is a product of matrices, where each matrix operates on a single site.
- Pictorial representation



first  
middle

last

$$T = pI + (1 - p) \sum_{i=1}^{n-1} \frac{1}{n-1} \sigma_i^x \otimes \sigma_{i+1}^x$$

$$T = T^{[1]} T^{[2]} \dots T^{[N]}$$

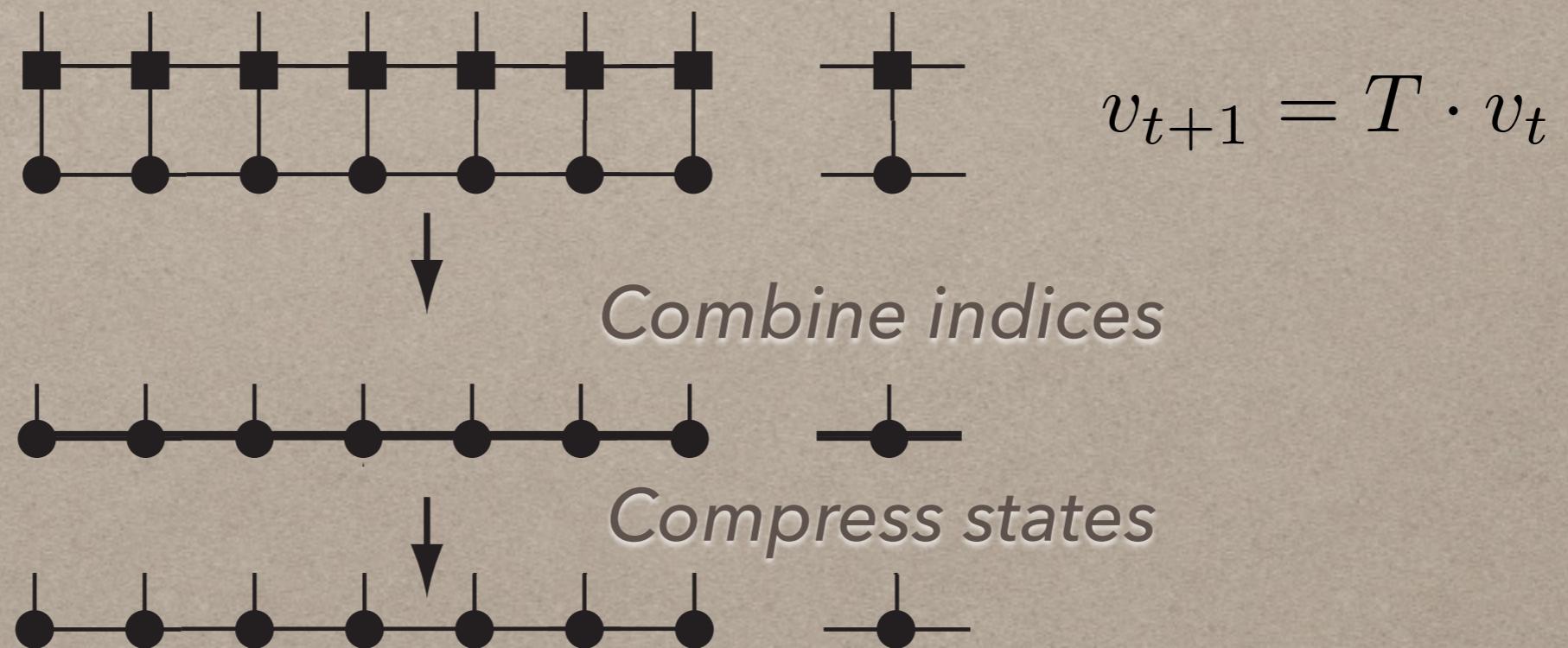
$$T^{[1]} = \begin{bmatrix} pI & (1-p)\sigma_1^X & I \end{bmatrix}$$

$$T^{[i]} = \begin{bmatrix} I & 0 & 0 \\ \sigma_i^X & 0 & 0 \\ 0 & (1-p)\sigma_i^X & I \end{bmatrix}$$

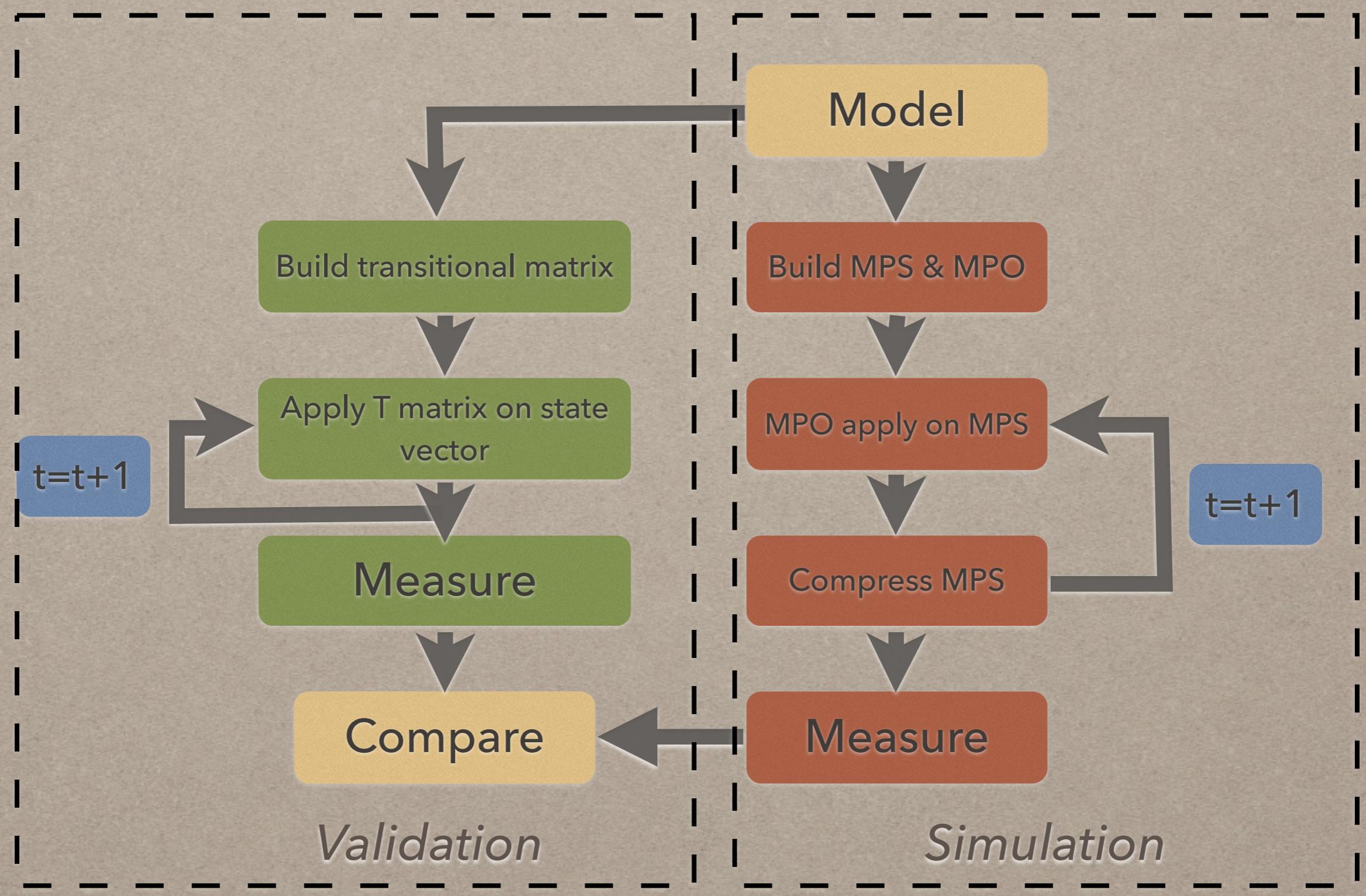
$$T^{[N]} = \begin{bmatrix} I \\ \sigma_N^X \\ 0 \end{bmatrix}$$

# MATRIX PRODUCT OPERATORS (MPO)

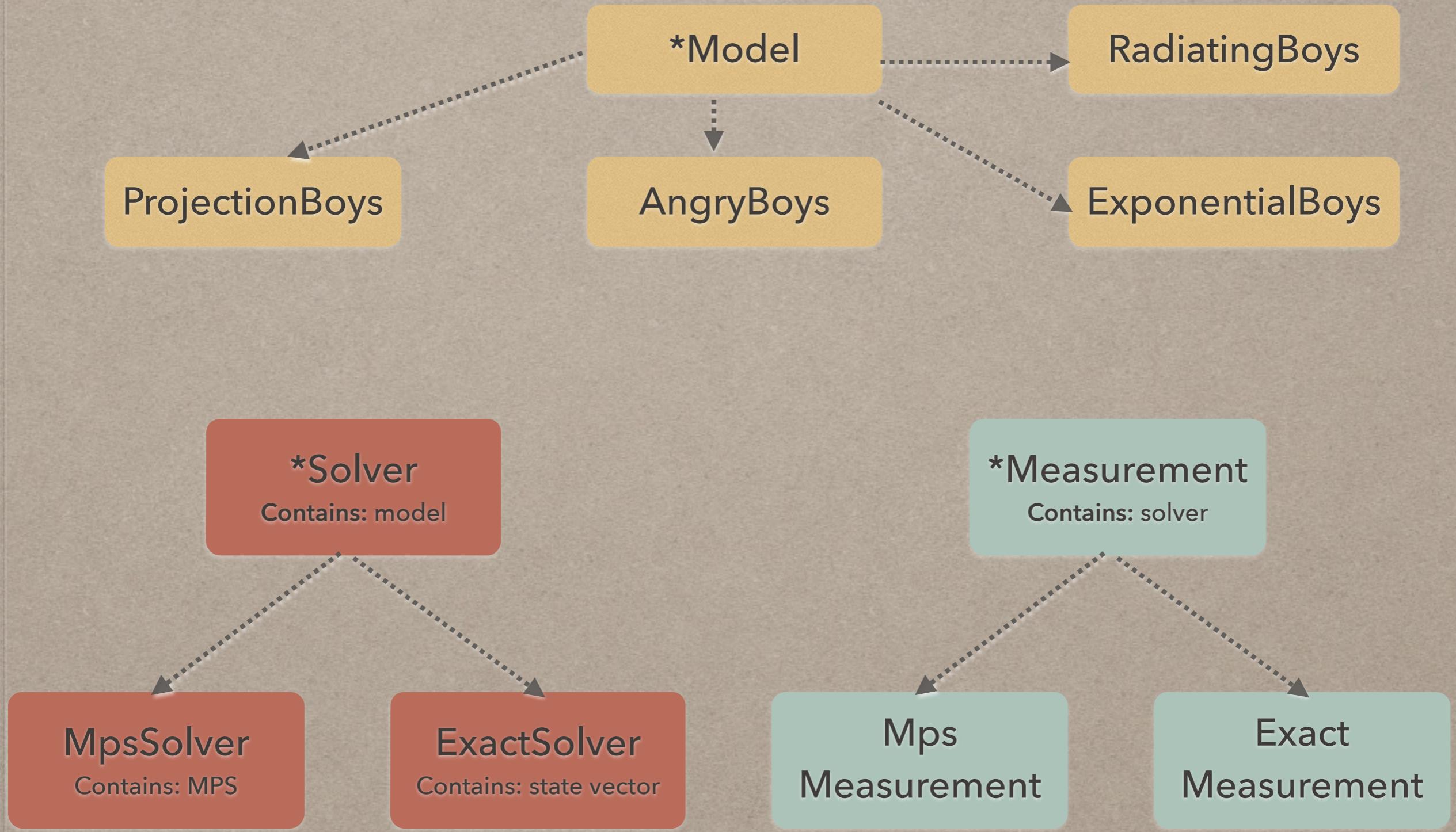
- Applying MPO of the transitional matrix on the MPS of a state vector



# ALGORITHM



# CLASS STRUCTURE



# THE PROGRAM

 [binarybin / MPS\\_Proba](#) Unwatch ▾ 5 Unstar 4 Fork 1

branch: [master](#) [MPS\\_Proba](#) / [src](#) / [+](#)

variance measurement bug

 [binarybin](#) authored 16 hours ago latest commit [172f10aedb](#)

..		
<a href="#">angryboys.py</a>	improved model representation (for all models)	14 days ago
<a href="#">exactmeasurement.py</a>	variance measurement bug	16 hours ago
<a href="#">exactsolver.py</a>	Abolished output channels; build MPS and exact representations of mod...	14 days ago
<a href="#">exponentialboys.py</a>	improved model representation (for all models)	14 days ago
<a href="#">measurement.py</a>	debugged exact measurement	16 hours ago
<a href="#">model.py</a>	improved model representation (for all models)	14 days ago
<a href="#">mpsmeasurement.py</a>	debugged exact measurement	16 hours ago
<a href="#">mpssolver.py</a>	ProjectionBoys model added: involving pi_plus and pi_minus projection...	14 days ago
<a href="#">proba.py</a>	debugged exact measurement	16 hours ago
<a href="#">projectionboys.py</a>	improved model representation (for all models)	14 days ago
<a href="#">radiatingboys.py</a>	fixed a bug in radiatingboys model	14 days ago
<a href="#">solver.py</a>	ProjectionBoys model added: involving pi_plus and pi_minus projection...	14 days ago

# RESULTS

