

# Learning Representation & Behavior: Manifold and Spectral Methods for Markov Decision Processes and Reinforcement Learning



Sridhar Mahadevan, U. Mass, Amherst



Mauro Maggioni, Yale Univ.

# Outline

- **Part One**
  - History and Motivation (9:00-9:15)
  - Overview of framework (9:15-9:30)
  - Technical Background (9:30-10:30)
  - Questions: (10:30-10:45)
- **Part Two**
  - Algorithms and implementation (11:15-11:45)
  - Experimental Results (11:45-12:30)
  - Discussion and Future Work (12:30-12:45)
  - Questions (12:45-1:00)

# Mathematical Foundations

- What we will assume:
  - Basic knowledge of machine learning, Markov decision processes and reinforcement learning
  - Linear algebra , graph theory, and statistics
- What we will introduce:
  - Least squares techniques (for solving MDPs)
  - Spectral graph theory: matrices  $\Leftrightarrow$  graphs
  - Fourier and wavelet bases on graphs
  - Continuous manifolds

# Tutorial Decomposition

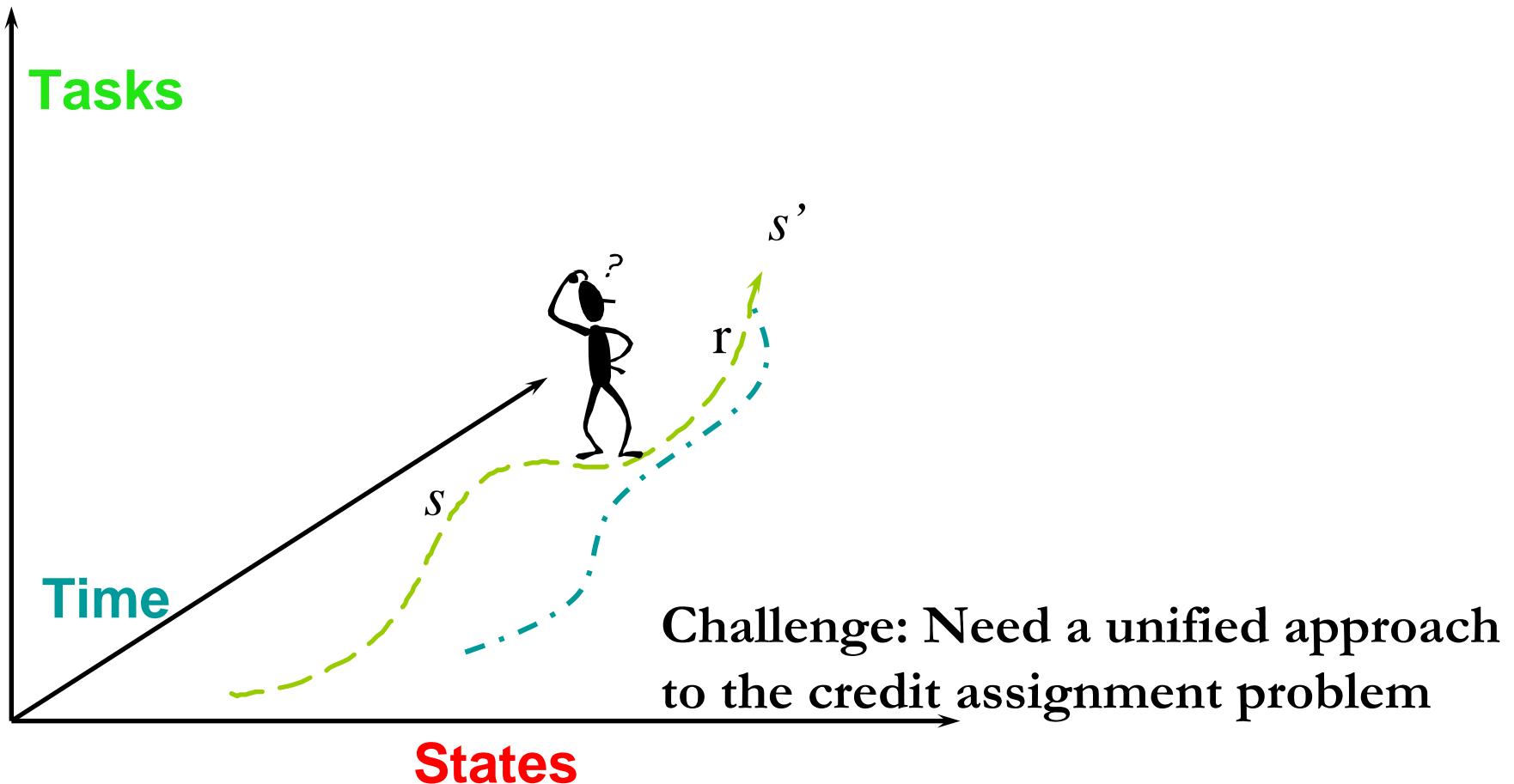
- Sridhar Mahadevan
  - History and motivation
  - Overview of the framework
  - Fourier (Laplacian) approach: global bases
- Mauro Maggioni
  - Harmonic analysis on graphs and manifolds
  - Diffusion wavelets: local bases
- Both:
  - Algorithms, experiments, implementation
  - Future work

# Outline

- **Part One**
  - History and Motivation (9:00-9:15)
  - Overview of framework (9:15-9:30)
  - Technical Background (9:30-10:30)
  - Questions: (10:30-10:45)
- **Part Two**
  - Algorithms and implementation (11:15-11:45)
  - Experimental Results (11:45-12:30)
  - Discussion and Future Work (12:30-12:45)
  - Questions (12:45-1:00)

# Credit Assignment Problem

(Minsky, Steps Toward AI, 1960)



# Samuel's Checker Player

(Arthur Samuel, 1950s)



- Samuel's work laid the foundations for many later ideas: temporal-difference learning, parametric function approximation, evaluation functions,...
- However, his work did not address a crucial problem: **learning of representation**

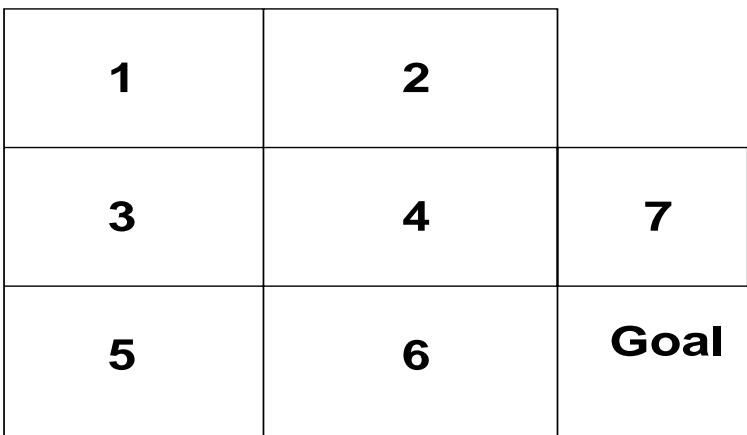
# Polynomial Bases

(Samuel, 1959; Koller and Parr, UAI 2000)

One basis function applied to all states

$\Phi$								
All basis functions applied to one state	1	1	1	1	1	1	1	1
	1	2	4	8	16	32	64	
	1	3	9	27	81	243	729	
	1	4	16	64	256	1024	4096	
	1	5	25	125	625	3125	15625	
	1	6	36	216	1296	7776	46656	
	1	7	49	343	2401	16807	117649	
	$i^0$	$i^1$	$i^2$	$i^3$	$i^4$	$i^5$	$i^6$	

# How to find a good basis?



$$e_1 = [1, \dots, 0]$$

$$e_i = [0, \dots, i, \dots, 0]$$

Any function on this graph  
is a vector in  $\mathbb{R}^7$

The question we want to ask is  
how to construct a basis set for  
approximating functions on this graph

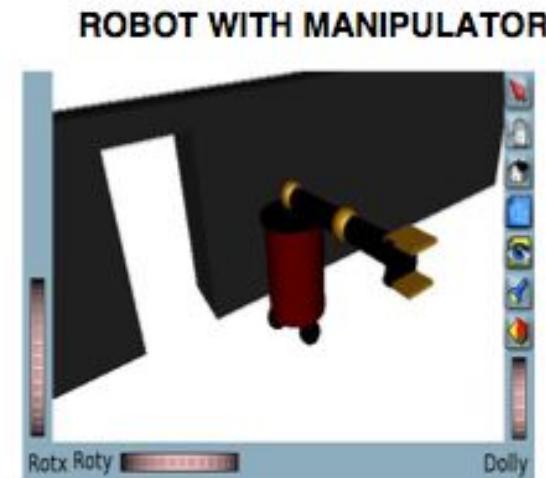
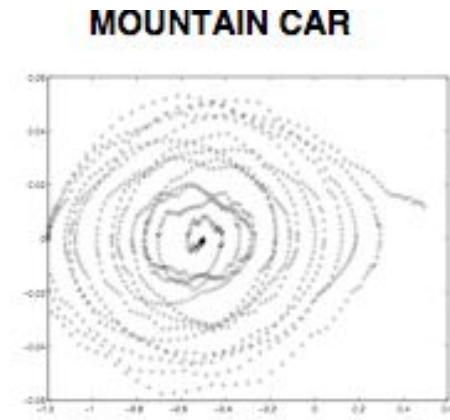
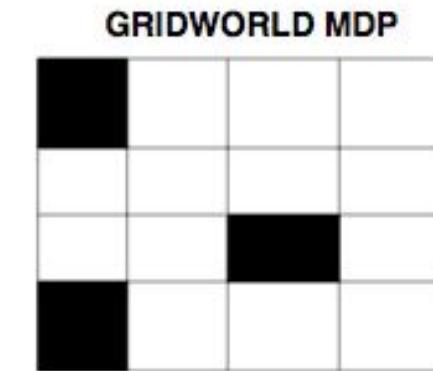
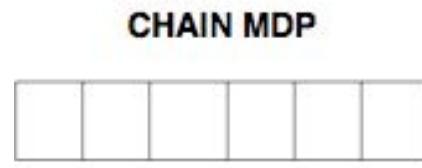
Solution 1: use the *unit basis*

Solution 2: use *polynomials or RBFs*

Neither of these exploit geometry

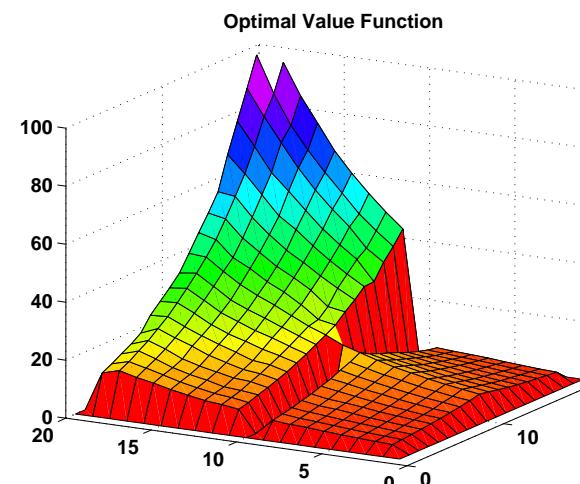
# Structural Credit Assignment: Automating Value Function Approximation

- Many approaches to value function approximation
  - Neural nets, radial basis functions, support vector machines, kernel density estimation, nearest neighbor
- How to automate the design of a function approximator?
- We want to go beyond model selection!

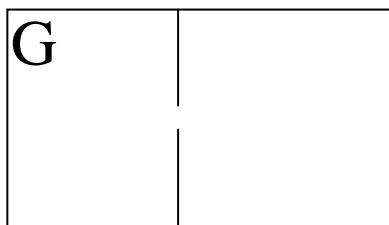


# Standard Approaches to VFA Can Easily Fail!

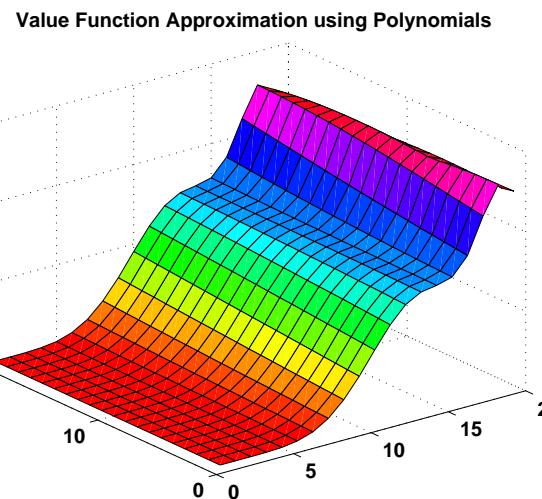
(Dayan, Neural Comp, 1993; Drummond, JAIR 2003)



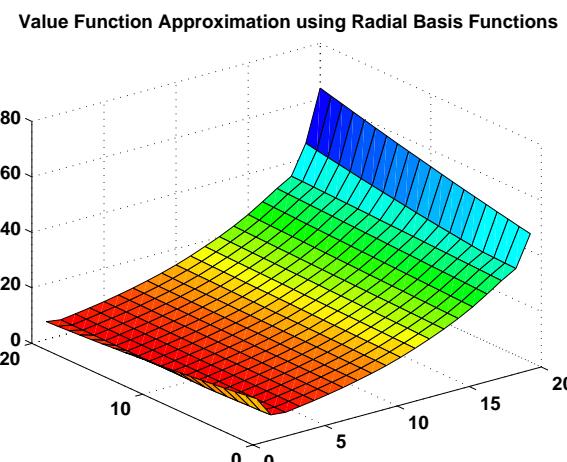
OPTIMAL VF



Multi-room  
environment



POLYNOMIAL

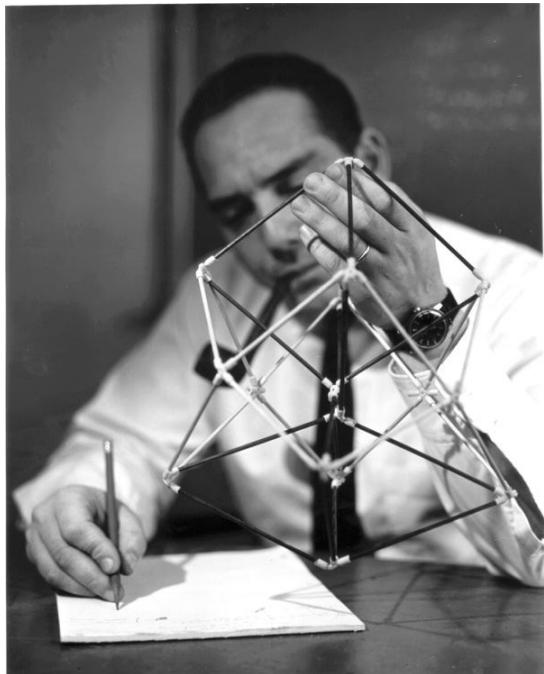


RADIAL BASIS FUNCTION

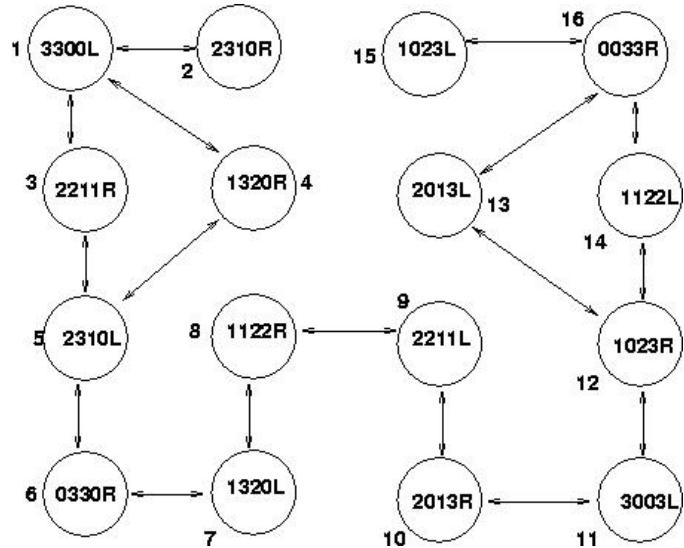
These approaches measure distances  
in ambient space, not on the manifold!

# Learning Representations by Global State Space Analysis

(Saul Amarel, 1960s)



Missionaries and Cannibals



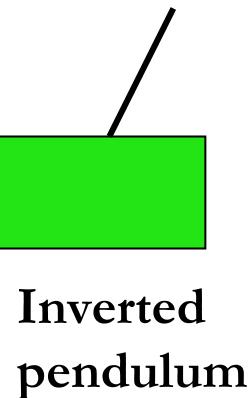
**Find symmetries and  
bottlenecks in state spaces**

# Outline

- **Part One**
  - History and Motivation (9:00-9:15)
  - Overview of framework (9:15-9:30)
  - Technical Background (9:30-10:30)
  - Questions: (10:30-10:45)
- **Part Two**
  - Algorithms and implementation (11:15-11:45)
  - Experimental Results (11:45-12:30)
  - Discussion and Future Work (12:30-12:45)
  - Questions (12:45-1:00)

# Overview of Framework

(Mahadevan, AAAI, ICML, UAI 2005; Mahadevan & Maggioni, NIPS 2005;  
Maggioni and Mahadevan, ICML 2006)



Samples from random walk  
on manifold in state space

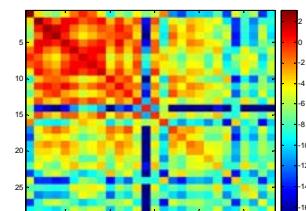
Build  
Graph

Fourier

Wavelets

$$\Delta\phi = \lambda\phi$$

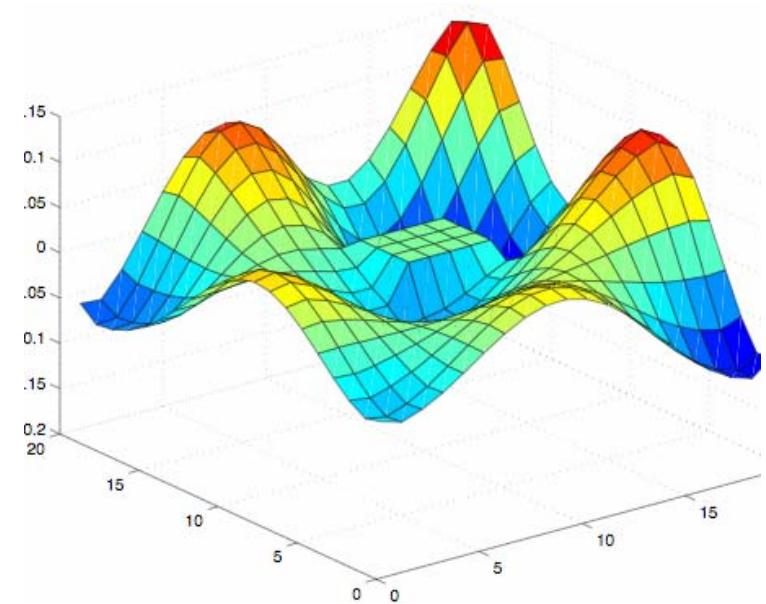
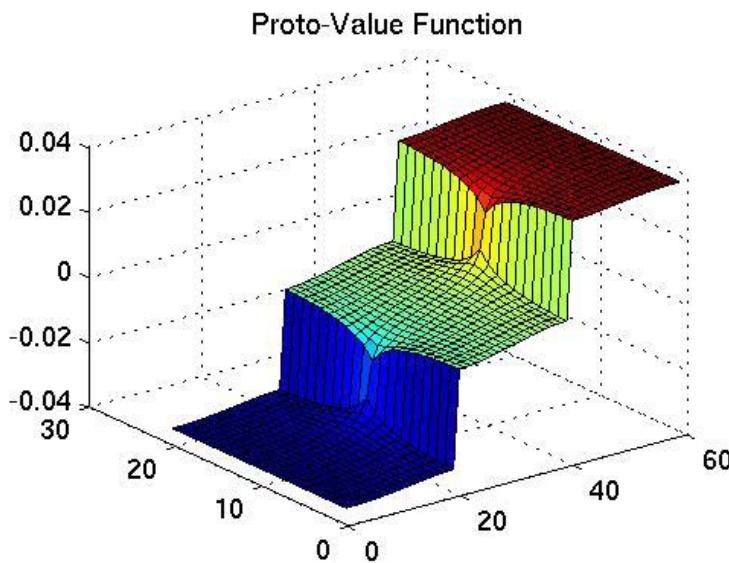
Eigenfunctions of  
Laplacian on state  
space: **global** basis  
functions



Dilations of  
diffusion operator:  
**local multiscale** basis  
functions

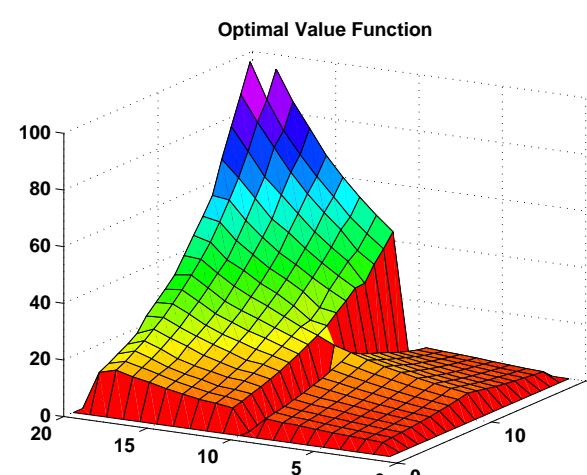
# Proto-Value Functions

(Mahadevan: AAAI 2005, ICML 2005, UAI 2005)

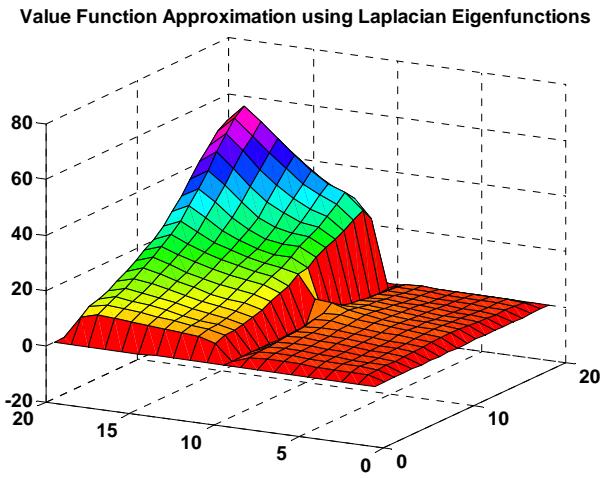


Proto-value functions are **reward-independent** global (or local) basis functions, **customized** to a state (action) space

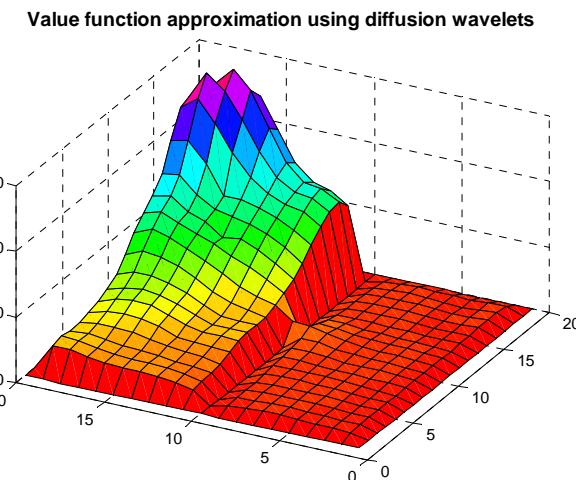
# Value Function Approximation using Fourier and Wavelet Bases



OPTIMAL VF



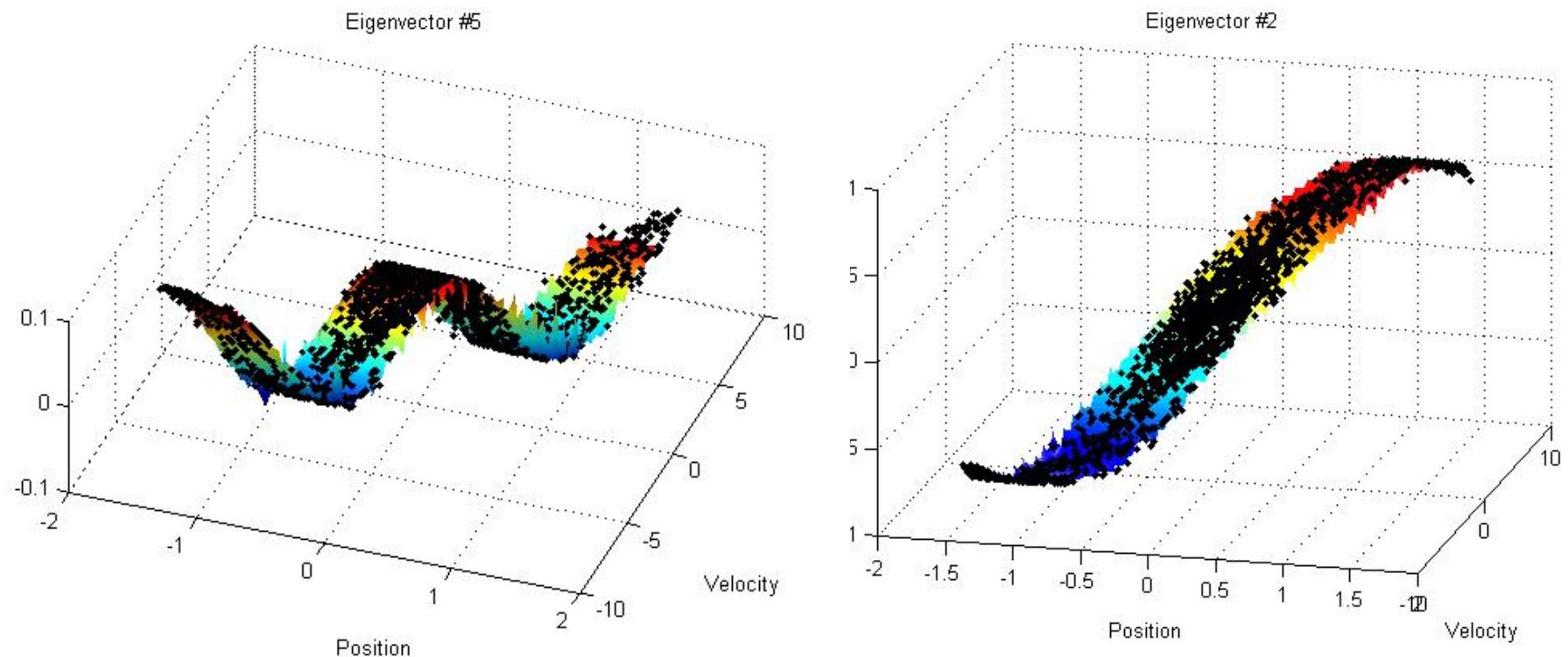
FOURIER BASIS



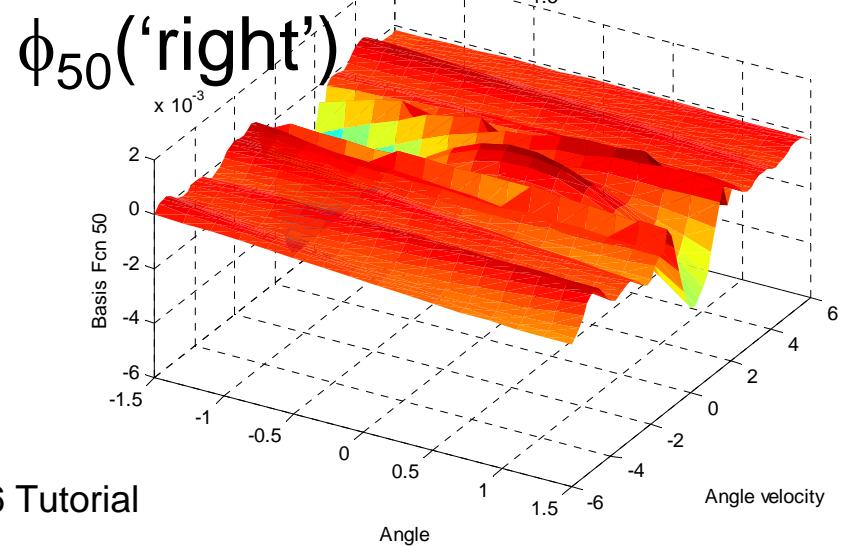
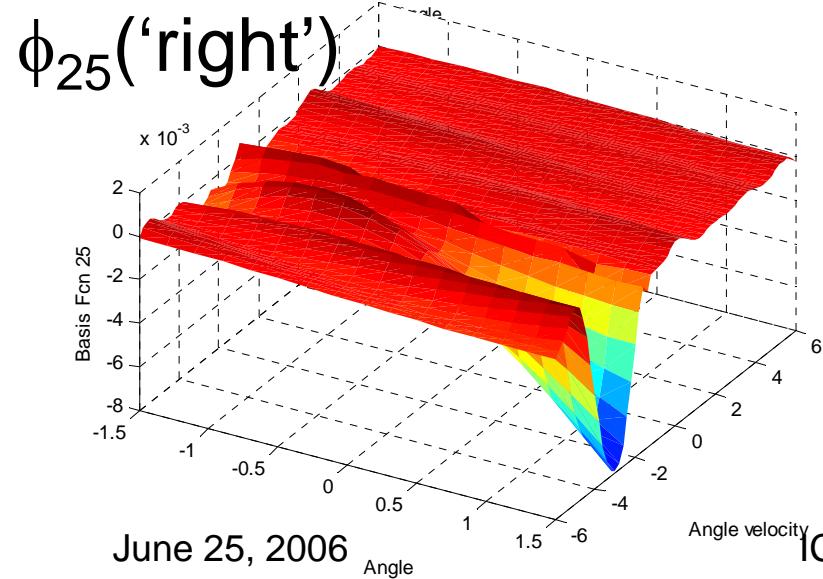
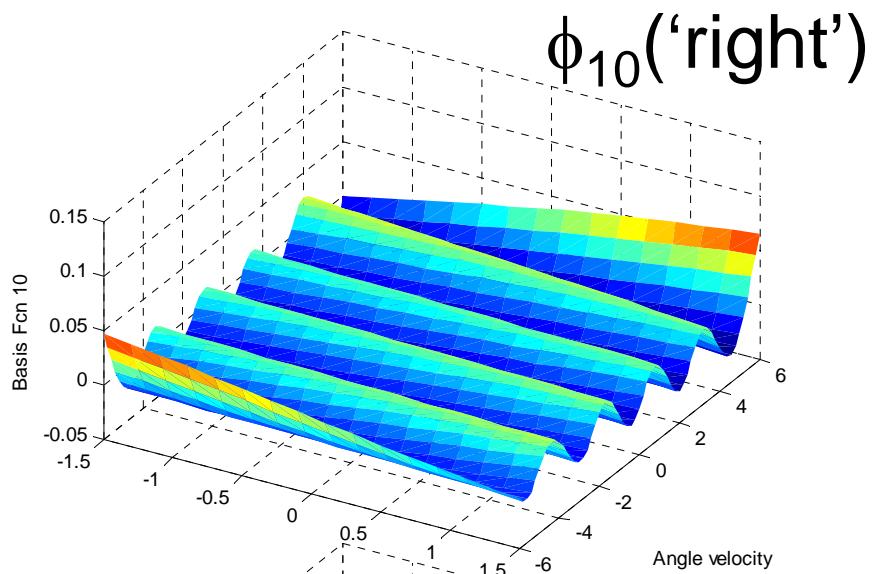
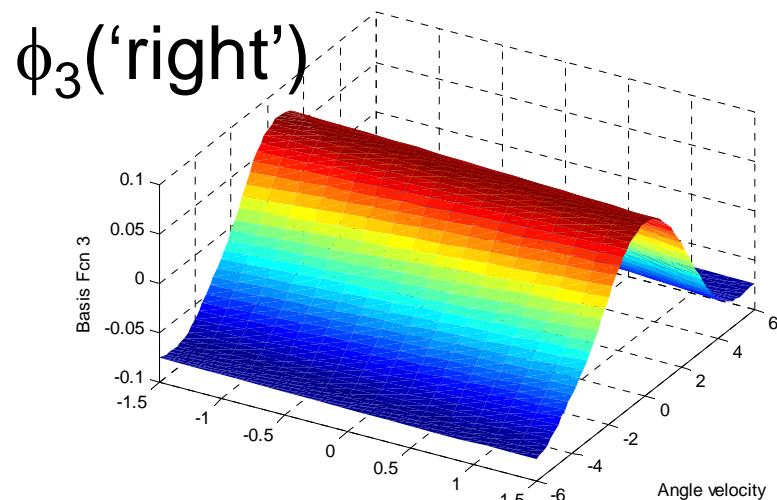
WAVELET BASIS

These bases are automatically learned from a set of transitions  $(s, a, s')$

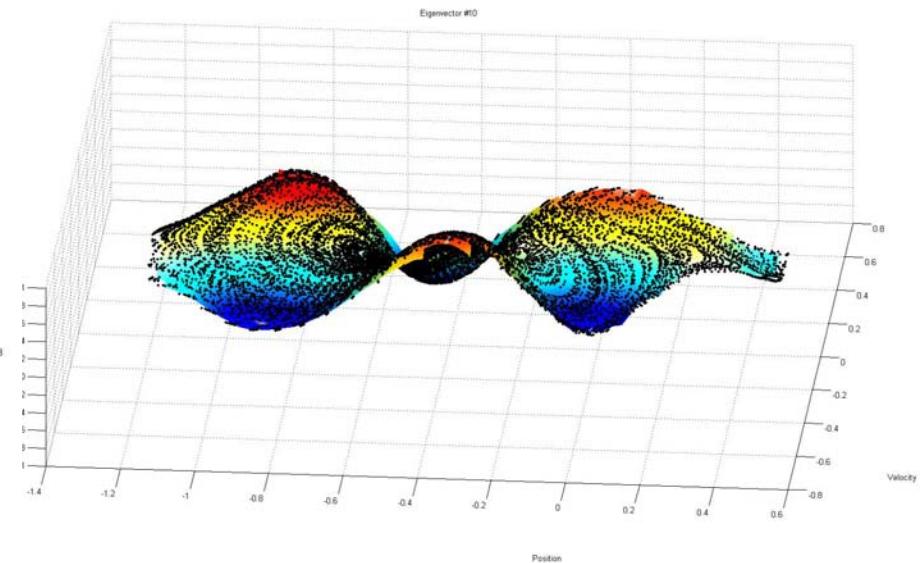
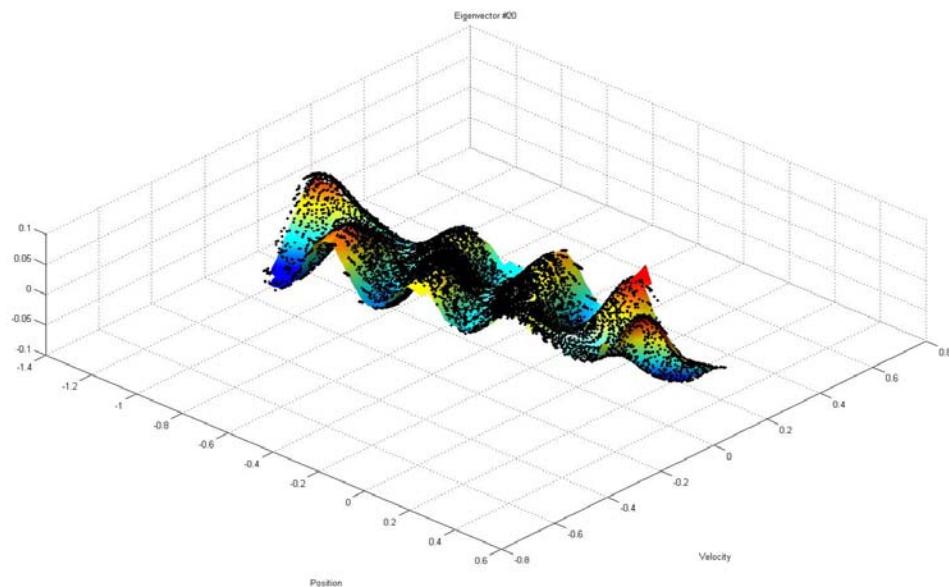
# Laplacian Proto-Value Functions: Inverted Pendulum



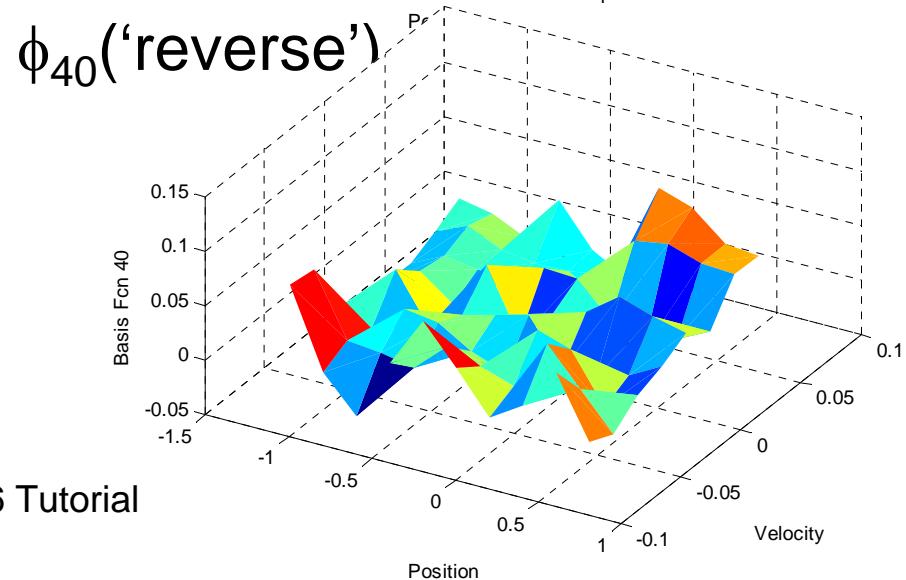
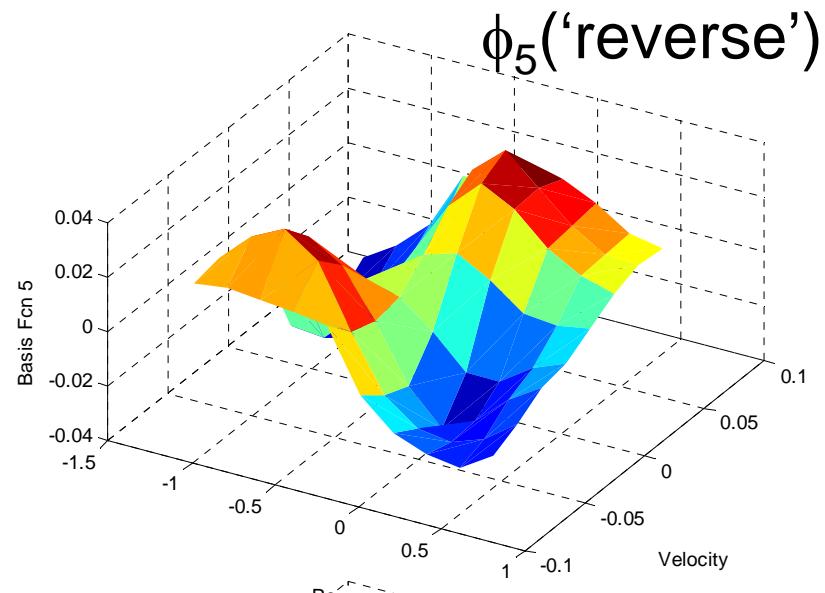
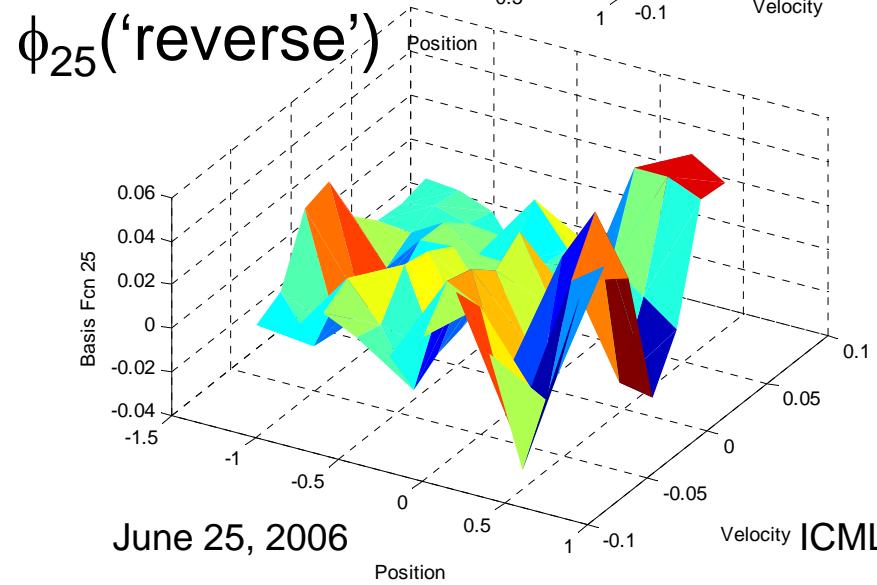
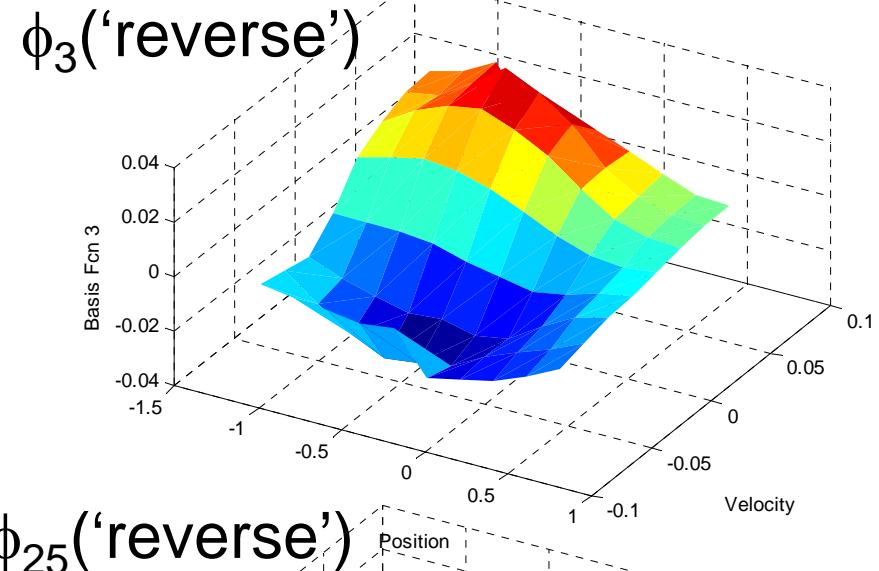
# Laplacian PVFs: Inverted Pendulum



# Laplacian Proto-Value Functions: Mountain Car



# Laplacian PVFs: Mountain Car



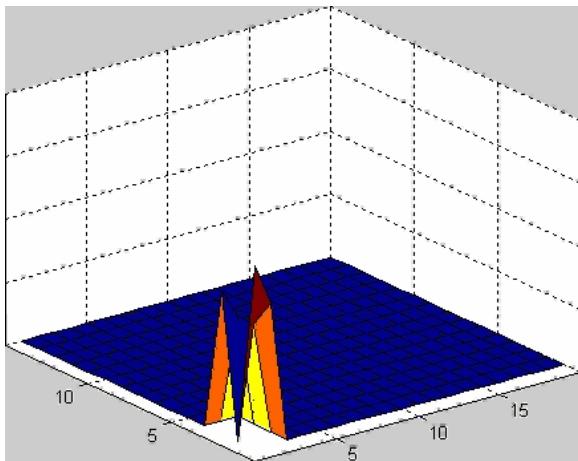
# Multiresolution Manifold Learning

- Fourier methods, like Laplacian manifold or spectral learning, rely on eigenvectors
  - Eigenvectors are useful in analyzing long-term global behavior of a system (e.g, PageRank)
  - They are rather poor at short or medium term transient analysis (or locally discontinuities)
- Wavelet methods [Daubechies, Mallat]
  - Inherently multi-resolution analysis
  - Local basis functions with compact support
- Diffusion wavelets [Coifman and Maggioni, 2004]
  - Extend classical wavelets to graphs and manifolds

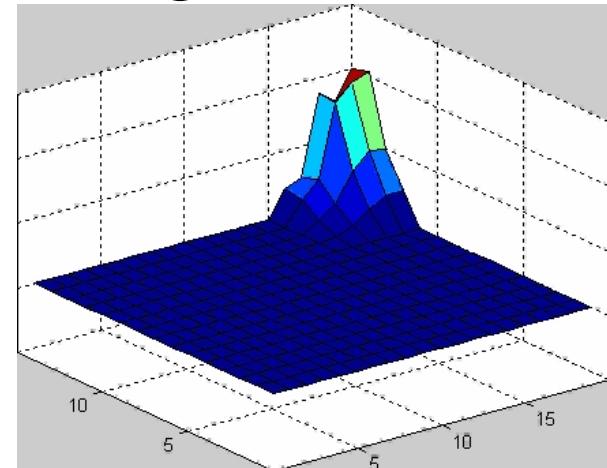
# Multiscale Analysis: Diffusion Wavelet Bases

(Coifman and Maggioni, 2004; Mahadevan and Maggioni, NIPS 2005)

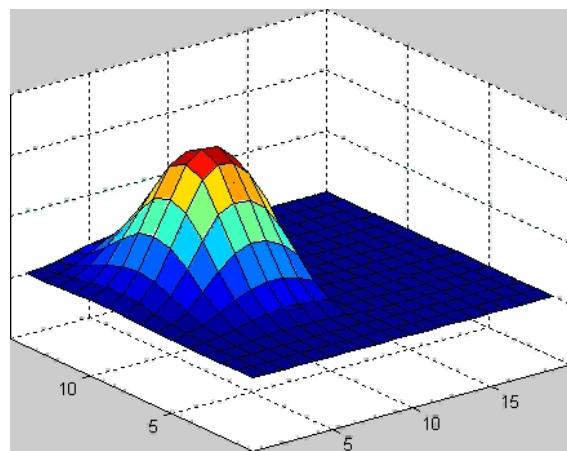
$\delta$  functions  $\rightarrow$  Global eigenvectors



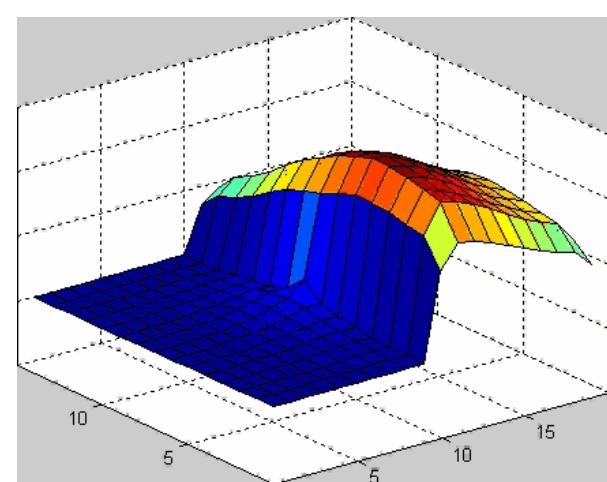
Level 2



Level 3



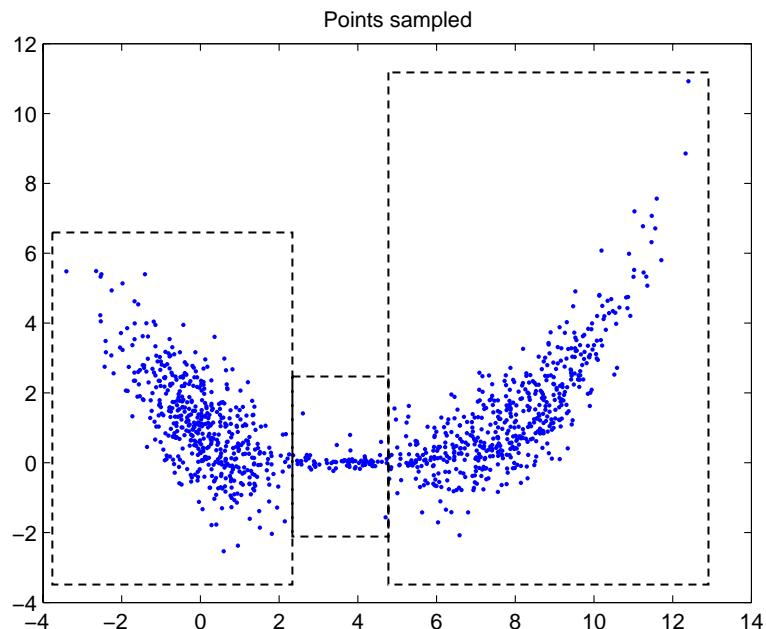
Level 4



Level 5

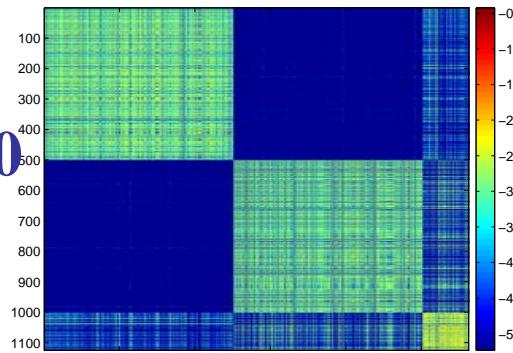
# Automatic Multiscale Analysis of Transition Matrices

(Maggioni and Mahadevan, ICML 2006)

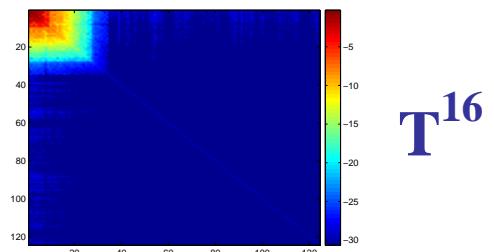


Random walk on a continuous  
two-room environment

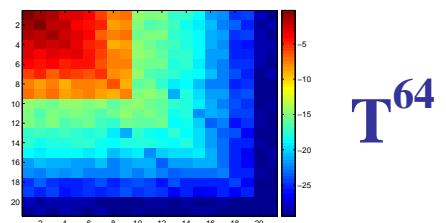
**1000x1000**



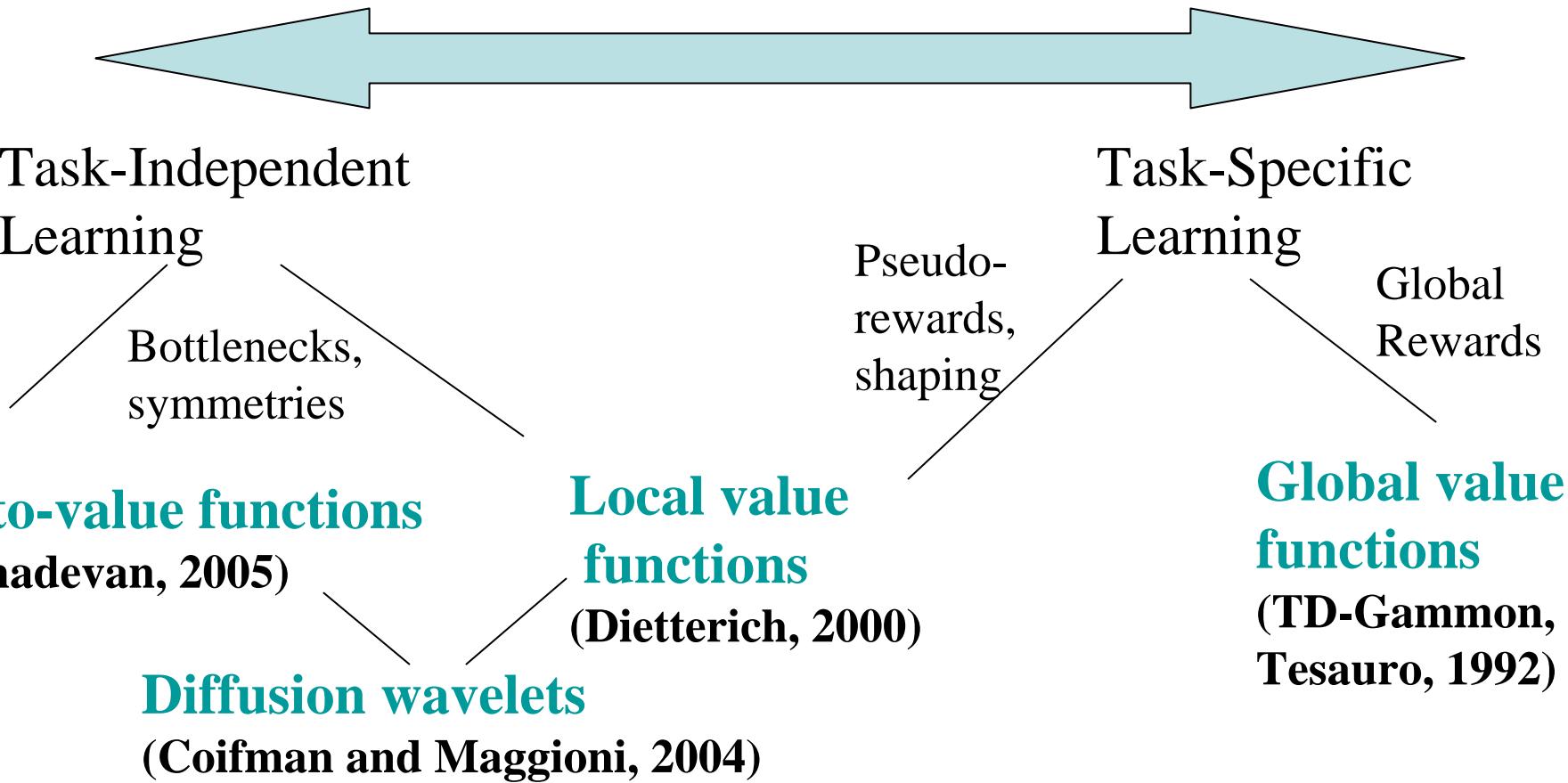
**100 x 100**



**20 x 20**



# Combining Reward-Specific and Reward-Independent Learning



# Learning Representation and Behavior

- Unified framework for credit assignment problem:
  - Harmonic analysis on graphs and manifolds
  - Fourier and wavelet bases (for value functions, transition matrices, policies, and more...)
  - Unifies RL with manifold and spectral learning
- Novel representations and algorithms:
  - Automatic basis function construction (values, policies)
  - Representation Policy iteration with an adaptive basis
  - Multiscale diffusion policy evaluation
  - A new representation for temporally extended actions
  - Transfer learning by representation sharing
  - Extendable to POMDPs and PSRs

# Manifold and Spectral Learning

- **Spectral methods** are based on computing eigenvectors of a normalized “affinity” matrix
  - [Shi and Malik, IEEE PAMI 1997]
  - [Ng, Jordan, and Weiss, NIPS 2001]
  - PageRank [Page, Brin, Motwani, Winograd, 1998]
- **Manifold methods** model the local geometry of the data by constructing a graph
  - [Roweis and Saul; Tenenbaum, de Silva, Langford, Science 2000]
  - [Belkin and Niyogi, MLJ 2004]
  - [Weinberger, Sha, Saul, ICML 2004]
- These methods are closely related to kernel PCA
  - [Scholkopff, Smola and Muller, 2001]
  - [Bengio et al, Neural Computation, 2004]

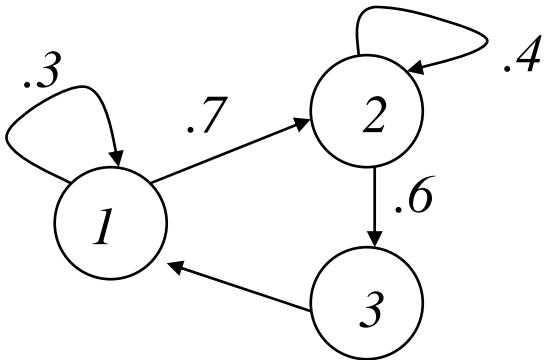
# “Curse of Dimensionality”

- Two active subfields in machine learning
  - Learning in inner product spaces: kernel methods
  - Manifold learning: nonlinear dimensionality reduction
- Fourier and wavelet bases on graphs
  - Based on analysis of the heat kernel of a graph
  - Basis for Hilbert space of functions on a graph
  - Nonlinear low-dimensional embedding of the graph
- Measure distances “intrinsically” in data space, not in ambient space!
  - Distance is based on diffusions (heat flow)
  - Closely connected to random walks on graphs

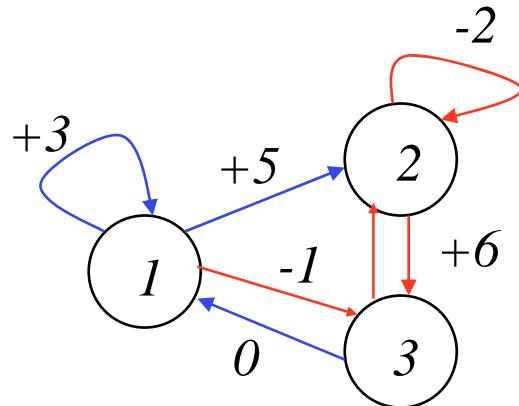
# Outline

- **Part One**
  - History and Motivation (9:00-9:15)
  - Overview of framework (9:15-9:30)
  - Technical Background (9:30-10:30)
  - Questions: (10:30-10:45)
- **Part Two**
  - Algorithms and implementation (11:15-11:45)
  - Experimental Results (11:45-12:30)
  - Discussion and Future Work (12:30-12:45)
  - Questions (12:45-1:00)

# Finite State Probabilistic Models

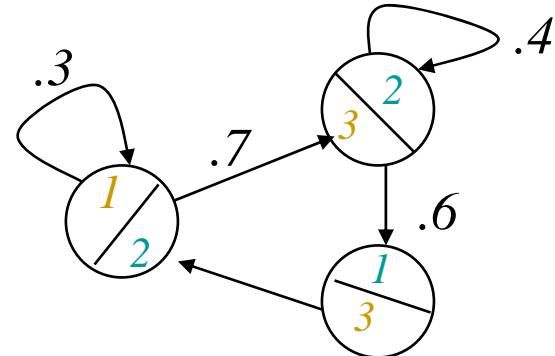


**Markov Chain**

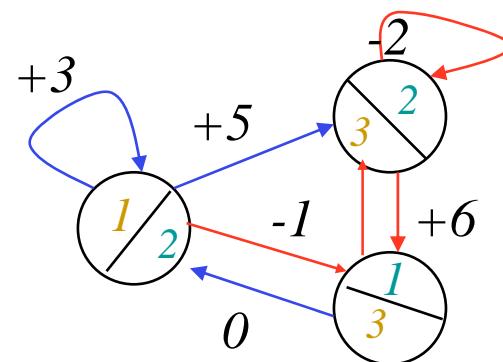


**Markov Decision Process**  
June 25, 2006

*How to learn  
multi-scale  
representations for  
analysis using  
these models?*



**Hidden Markov Model**



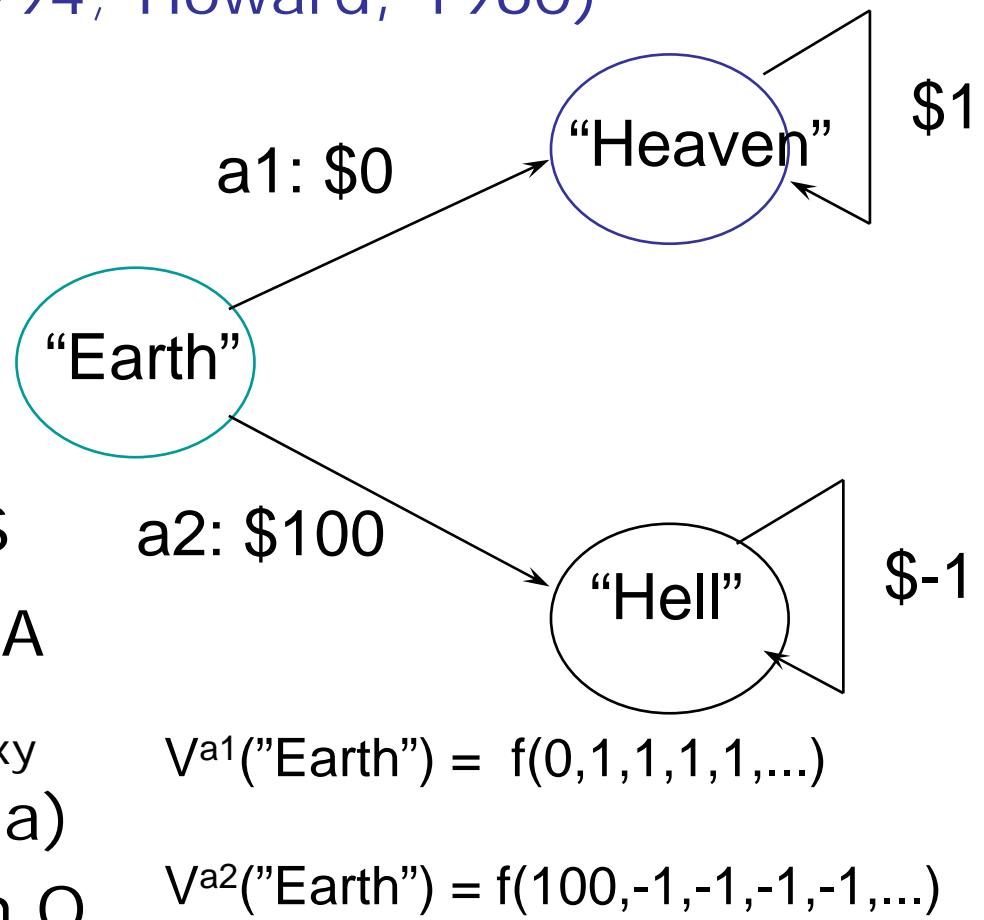
**Partially Observable Markov  
Decision Process**

# Markov Decision Processes

(Puterman, 1994; Howard, 1960)

- Specification:

- A finite set of states  $S$
- A finite set of actions  $A$
- A transition matrix  $P^a_{xy}$
- A reward function  $r(s,a)$
- An optimality criterion  $O$



# Infinite Horizon Markov Decision Processes

- Graded myopia: “*discounted sum of rewards*”

- Maximize

$$\sum_t \gamma^t r_t$$

- Hell if  $\gamma < 0.98$ , otherwise Heaven

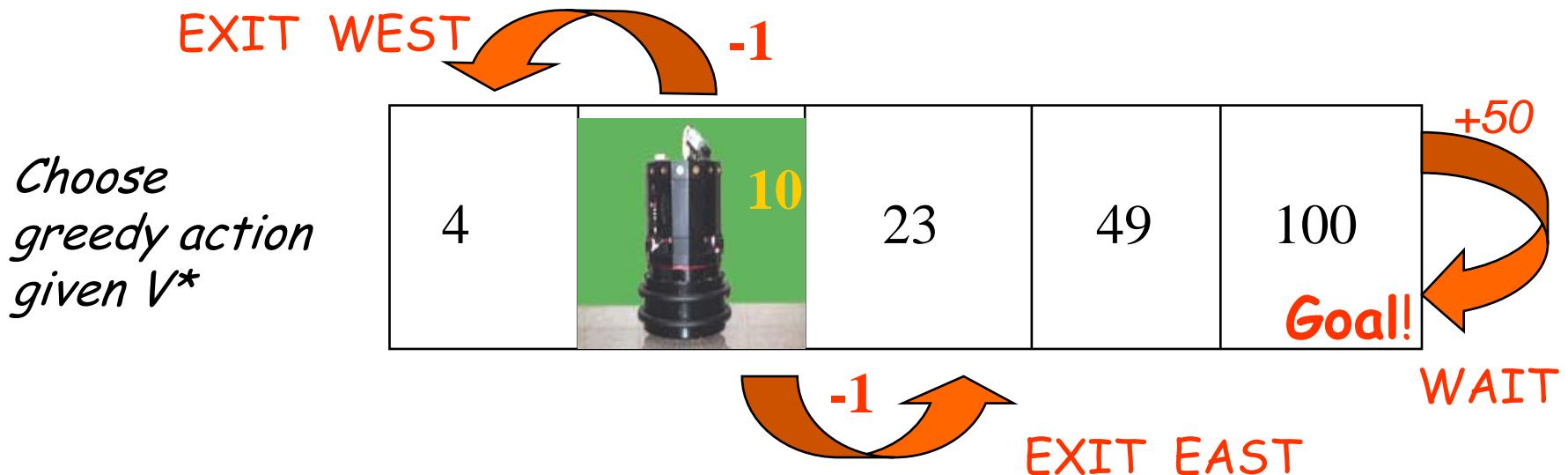
- Maximize “*average-adjusted sum of rewards*”

$$\lim_{n \rightarrow \infty} \frac{\sum_{t=1}^n r_t}{n}$$

Always go to heaven!

# Bellman Optimality Equation

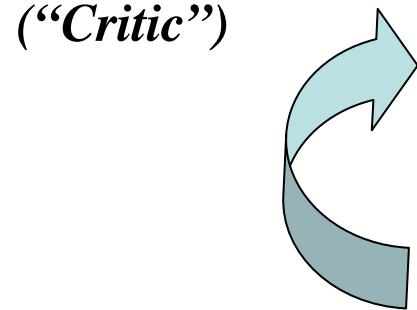
$$V^*(x) = \max_{a \in A(x)} \left( r(x, a) + \gamma \sum_y P_{xy}^a V^*(y) \right)$$



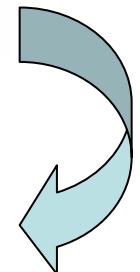
# Policy Iteration

(Howard, PhD, MIT, 1959)

*Policy Evaluation:*  $V^\pi(x) = r(x, \pi(x)) + \gamma \sum_y P_{xy}^{\pi(x)} V^\pi(y)$



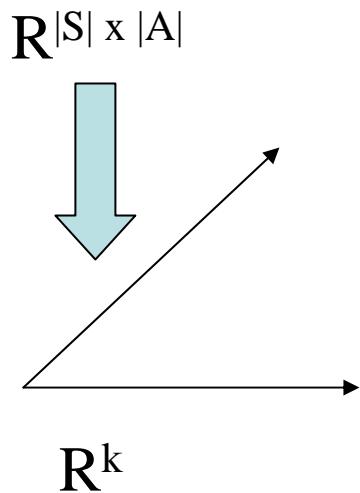
*Policy Improvement:*  $\pi'(x) = \operatorname{argmax}_a \left( r(x, a) + \gamma \sum_y P_{xy}^a V^\pi(y) \right)$



# Value Function Approximation

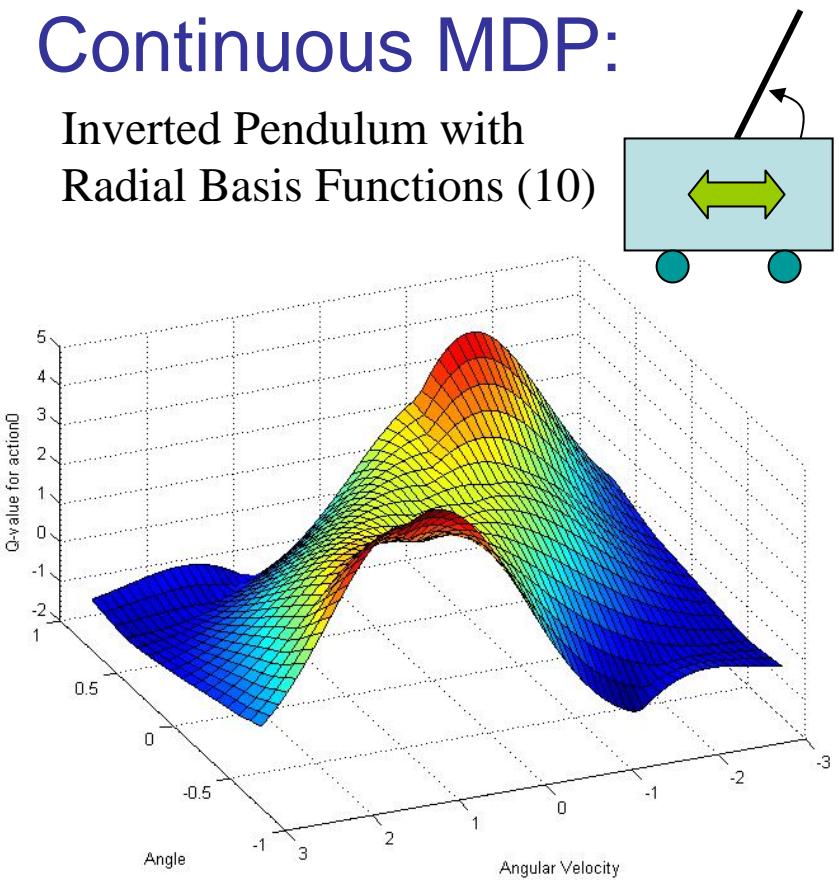
Dimensionality Reduction

Discrete MDP:



Continuous MDP:

Inverted Pendulum with  
Radial Basis Functions (10)



# Linear Value Function Approximation

- Consider a linear architecture for approximating value functions

$$V(s) \approx \sum_i \phi_i(s) w_i$$

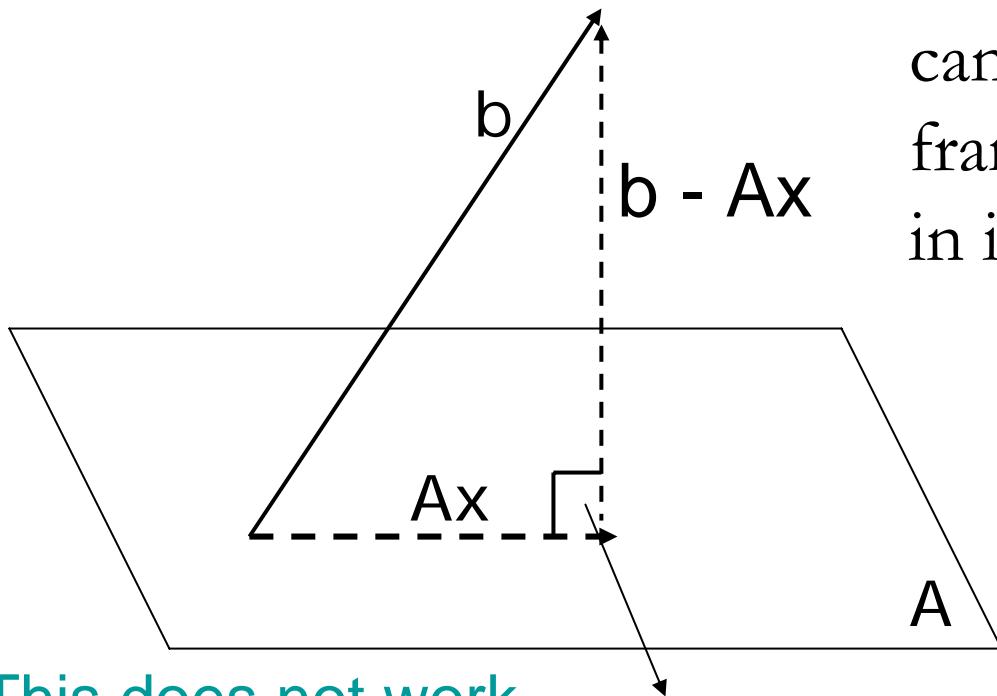
- Policy evaluation is no longer straightforward!

$$\sum_i \phi_i(s) w_i \neq r(s, \pi(s)) + \gamma \sum_{s'} P_{s,s'}^a \sum_i \phi_i(s') w_i$$

- This equation is not guaranteed to be solvable since the RHS may be outside column space of  $\Phi$

# Least-Squares Projection

(Strang, 2003; Deutsch, 2001)



Many problems in learning  
can be formalized using the  
framework of projections  
in inner product spaces

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

$$\mathbf{P} = \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$$

This does not work  
for  $L_\infty$  norm

Extends to infinite-dimensional spaces  
provided subspace is “complete”

# Bellman Residual Method

(Munos, ICML 03; Lagoudakis and Parr, JMLR 03)

- Let us write the Bellman equation in matrix form as

$$\Phi w^\pi \approx R^\pi + \gamma P^\pi \Phi w^\pi$$

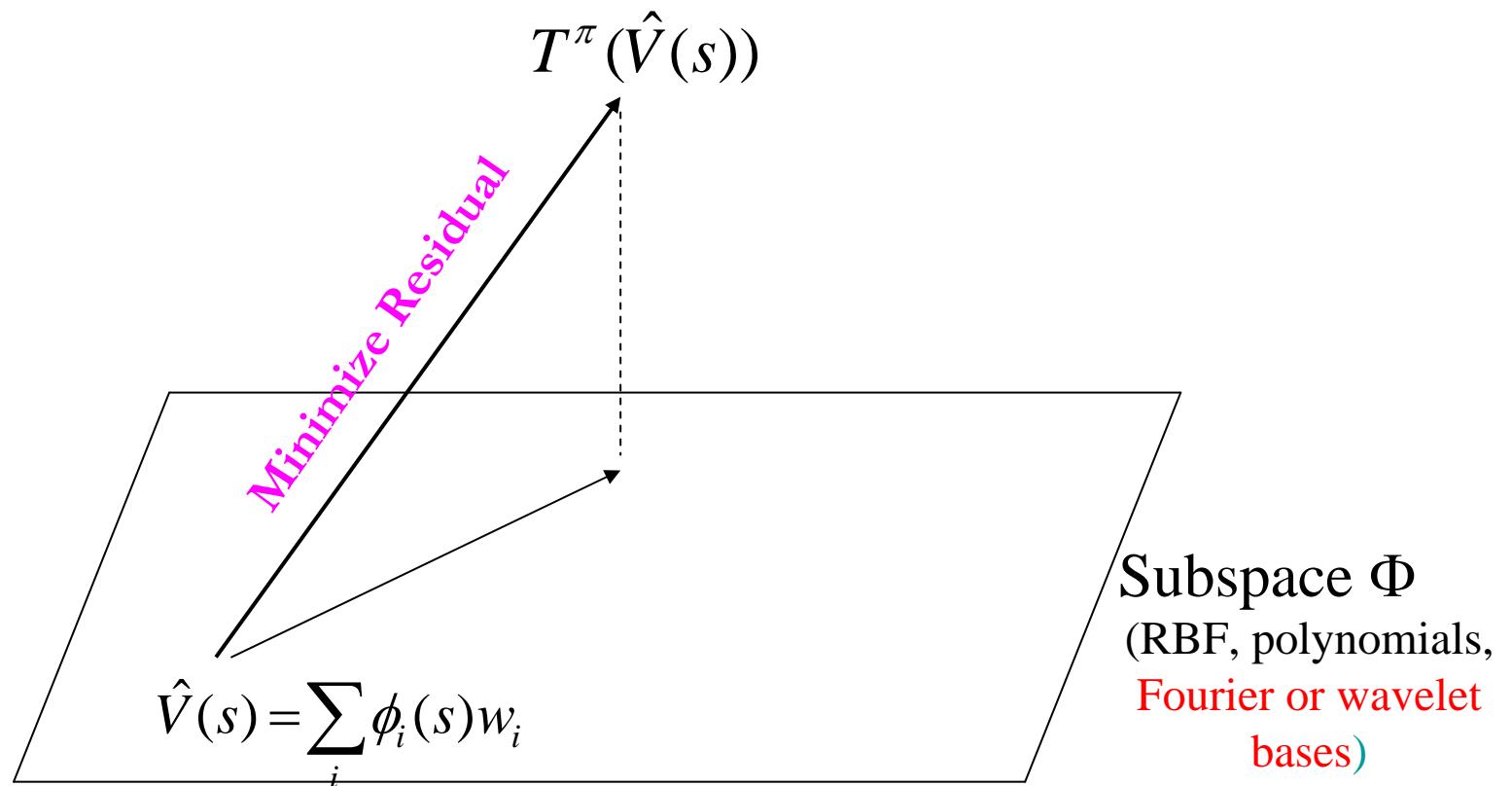
- Collecting the terms, we rewrite this as

$$(\Phi - \gamma P^\pi \Phi) w^\pi \approx R^\pi$$

- The least-squares solution is

$$w^\pi = [(\Phi - \gamma P^\pi \Phi)^T (\Phi - \gamma P^\pi \Phi)]^{-1} (\Phi - \gamma P^\pi \Phi)^T R^\pi$$

# Bellman Residual Method



# Bellman Fixpoint Method

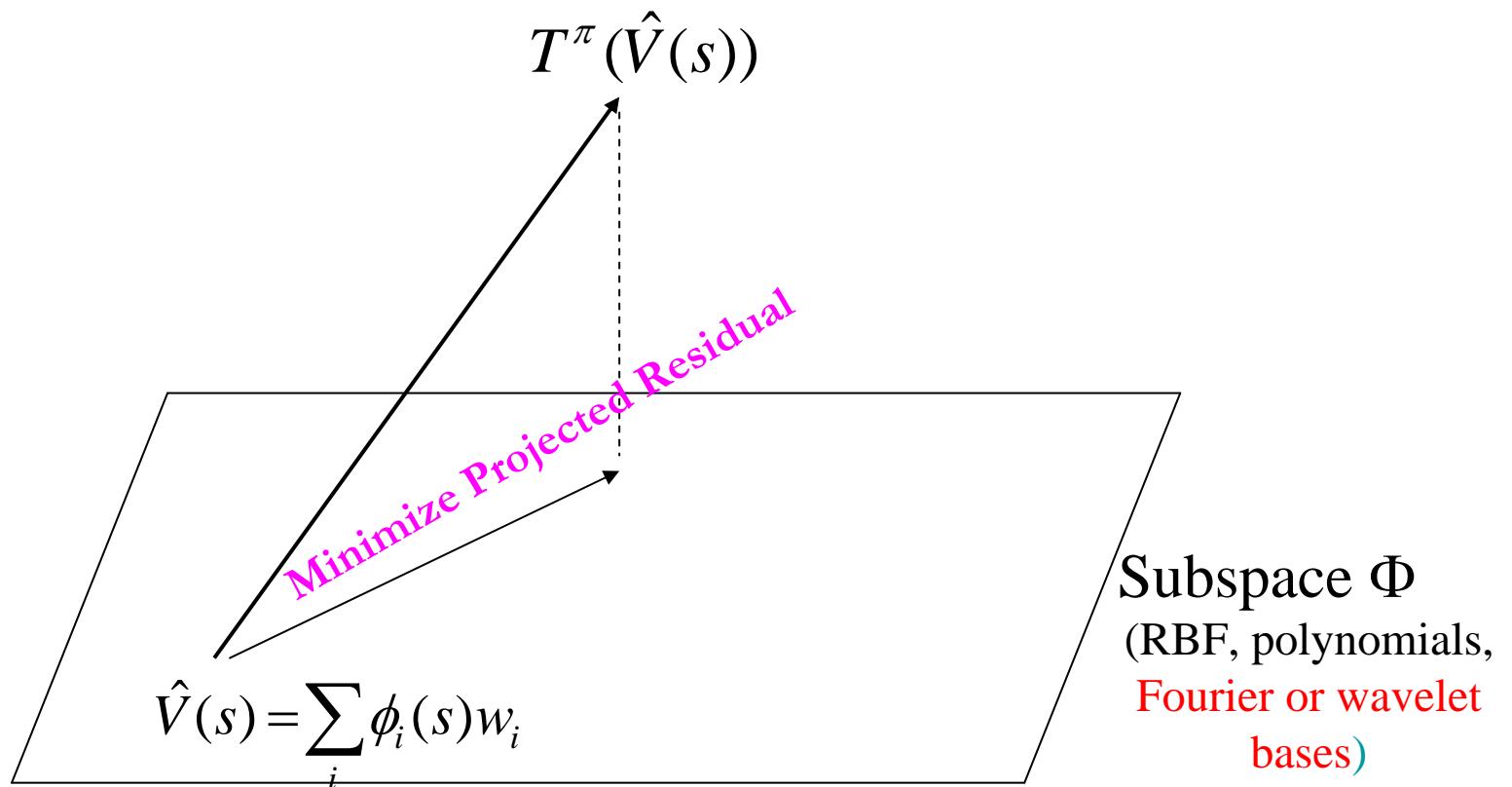
- Another way to obtain a least-squares solution is to project the backed-up value function  $T^\pi(V^\pi)$

$$P = \Phi (\Phi^T \Phi)^{-1} \Phi^T$$

- The least-squares projected weights then becomes

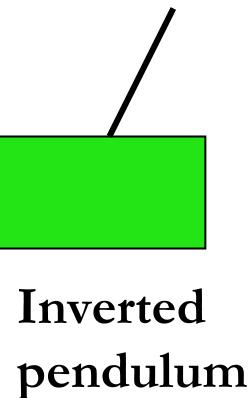
$$W^\pi = (\Phi^T \Phi)^{-1} \Phi^T [R^\pi + \gamma P^\pi V^\pi]$$

# Bellman Fixpoint Method



# Overview of Framework

(Mahadevan, AAAI, ICML, UAI 2005; Mahadevan & Maggioni, NIPS 2005;  
Maggioni and Mahadevan, ICML 2006)



Samples from random walk  
on manifold in state space

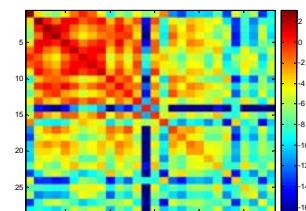
Build  
Graph

Fourier

Wavelets

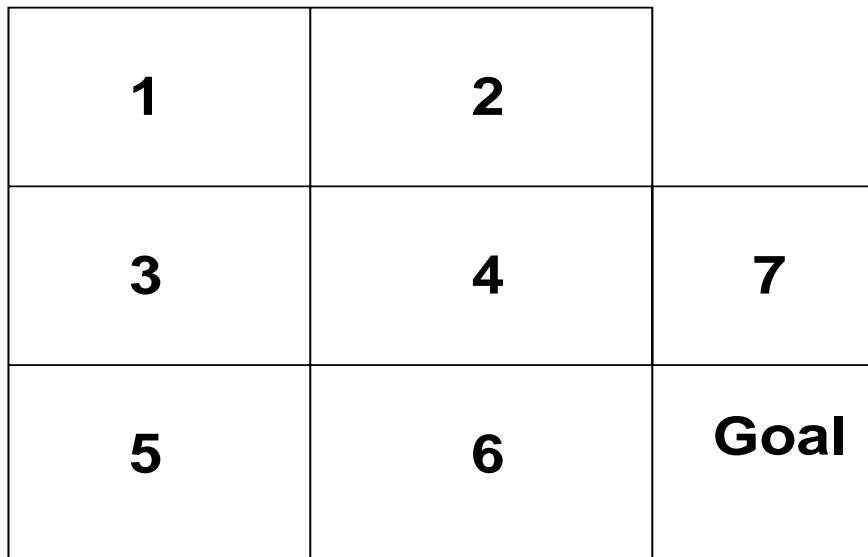
$$\Delta\phi = \lambda\phi$$

Eigenfunctions of  
Laplacian on state  
space: **global** basis  
functions



Dilations of  
diffusion operator:  
**local multiscale** basis  
functions

# Graph Adjacency Matrix



Adjacency Matrix

0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	0	1	1	0	0
0	1	1	0	0	1	1
0	0	1	0	0	1	0
0	0	0	1	1	0	0
0	0	0	1	0	0	0

# Spectral Theorem

- From basic linear algebra, we know that since the adjacency matrix  $A$  is symmetric, we can use the spectral theorem

$$A = V \Lambda V^T$$

- $V$  is a matrix of orthonormal eigenvectors,  $\Lambda$  is a diagonal matrix of eigenvalues
- Eigenvectors satisfy the following property:

$$A x = \lambda x$$

# Diagonalized Adjacency Matrix

$$\Lambda = \begin{matrix} & 2.5243 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 1.0000 & 0 & 0 & 0 & 0 \\ \Lambda = & 0 & 0 & 0.7923 & 0 & 0 & 0 \\ & 0 & 0 & 0 & -0.7923 & 0 & 0 \\ & 0 & 0 & 0 & 0 & -1.0000 & 0 \\ & 0 & 0 & 0 & 0 & 0 & -2.5243 \end{matrix}$$

$$V = \begin{matrix} & -0.3213 & 0.5000 & -0.3831 & 0.3831 & 0.5000 & 0.3213 \\ & -0.3419 & 0.5000 & 0.0900 & 0.0900 & -0.5000 & -0.3419 \\ V = & -0.4692 & -0.0000 & -0.3935 & -0.3935 & -0.0000 & -0.4692 \\ & -0.5418 & 0.0000 & 0.4544 & -0.4544 & -0.0000 & 0.5418 \\ & -0.3213 & -0.5000 & -0.3831 & 0.3831 & -0.5000 & 0.3213 \\ & -0.3419 & -0.5000 & 0.0900 & 0.0900 & 0.5000 & -0.3419 \\ & -0.2146 & -0.0000 & 0.5735 & 0.5735 & 0.0000 & -0.2146 \end{matrix}$$

# Inner Product Spaces

- An *inner product space* is a vector space associated with an inner product (e.g,  $\mathbb{R}^n$ )
- The set of all functions  $\Phi$  on a graph  $G = (V, E)$  forms an inner product space, where the inner product is defined as
$$\langle f, g \rangle = \sum_i f(i) g(i)$$
- An *operator*  $O$  on an inner product space of functions is a mapping  $O: \Phi \rightarrow \Phi$

# Adjacency Operator

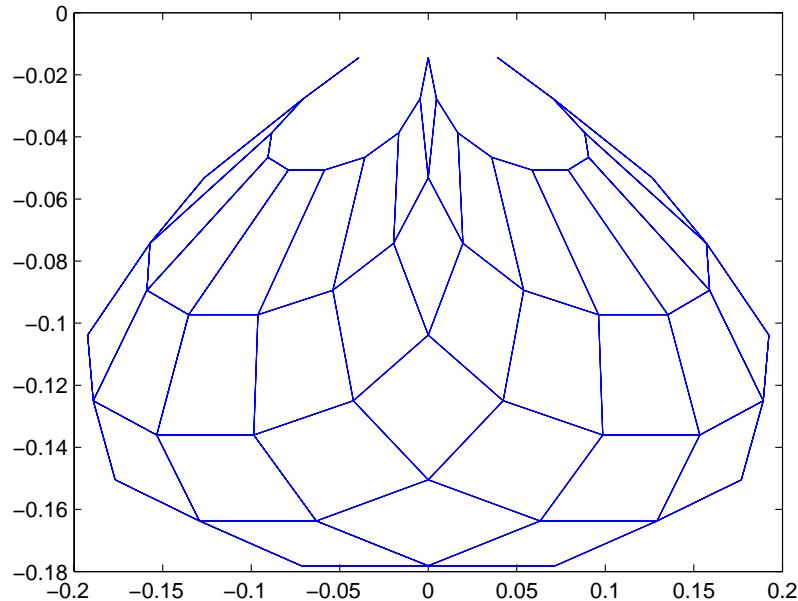
- Let us now revisit the adjacency matrix and treat it as an operator
- What is its effect on functions on the graph?
- It is easy to see that

$$A f(i) = \sum_{j \sim i} f(j)$$

0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	0	1	1	0	0
0	1	1	0	0	1	1
0	0	1	0	0	1	0
0	0	0	1	1	0	0
0	0	0	1	0	0	0

# Graph Embedding: 10x10 Grid

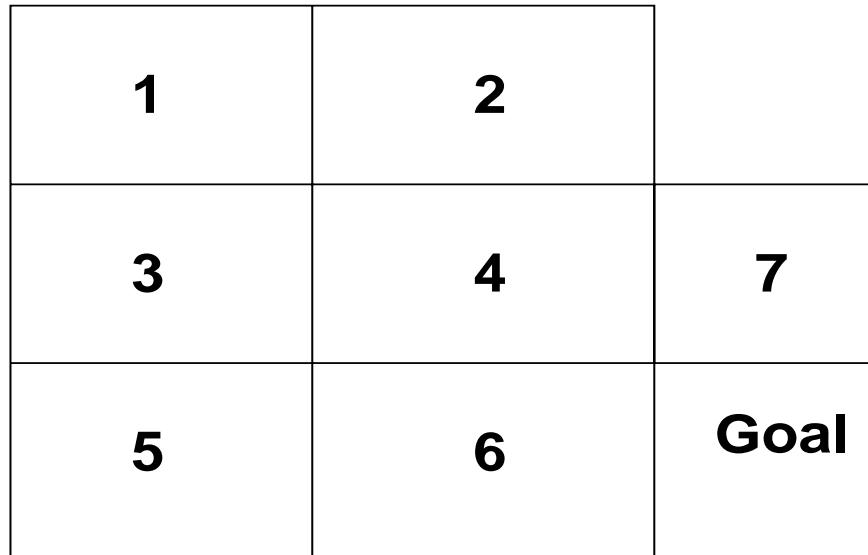
$$\phi: V \rightarrow \mathbb{R}$$



Here,  $\phi$  was defined using the eigenvectors associated with the largest two eigenvalues (99<sup>th</sup> and 100<sup>th</sup>)

# Combinatorial Graph Laplacian

(Fiedler, 1973; Cvetkovic et al, 1980; Chung, 1997)



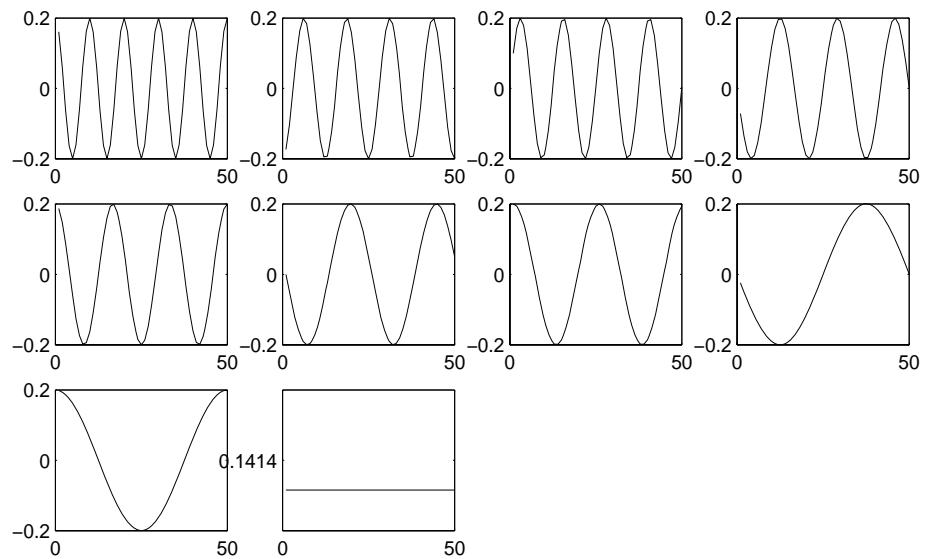
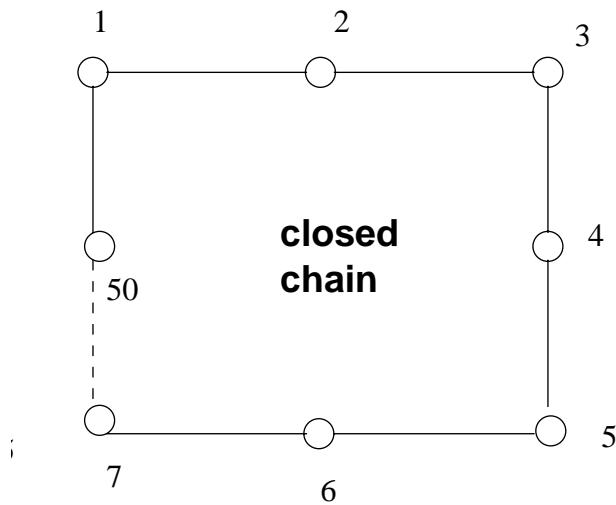
Laplacian Matrix =  $D - A$

2	-1	-1	0	0	0	0
-1	2	0	-1	0	0	0
-1	0	3	-1	-1	0	0
0	-1	-1	4	0	-1	-1
0	0	-1	0	2	-1	0
0	0	0	-1	-1	2	0
0	0	0	-1	0	0	1

Row sums

Negative of weights

# One-Dimensional Chain MDP



Eigenvectors of the  
Graph Laplacian

# Fourier Approach: Laplacian Eigenfunctions

- In 1807, Joseph Fourier discovered trigonometric functions are a complete basis for approximating any smooth function, while solving the heat equation
  - The trigonometric functions  $e^{i w t}$  diagonalize any time-invariant linear operator  $L$ 
$$L e^{t w t} = h(w) e^{i w t}$$
$$L f = \int_{-\infty}^{\infty} h(w) e^{-i w t} dt$$
- Fourier's insight has been generalized extensively over the past 200 years
  - In the study of continuous manifolds, the eigenfunctions of the Laplacian form a discrete orthonormal basis [Rosenberg, 1997]
  - For vector spaces associated with a discrete graph, the eigenfunctions of the graph Laplacian form a complete basis [Chung, 1997; Cvetkovic, 1980]

# Simple Properties of the Laplacian

- The Laplacian is *positive semidefinite*

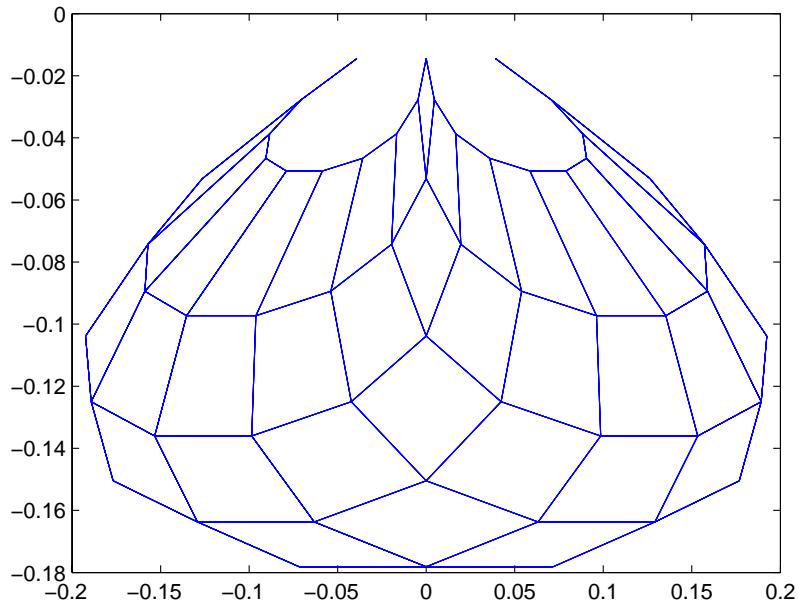


- The Laplacian for this graph is  $[1 \ -1; \ -1 \ 1]$
- Note that  $x^T L x = (x_1 - x_2)^2$
- We can express the Laplacian of any graph as the sum of the Laplacians of the same graph with all edges deleted, except for one.
- This implies that

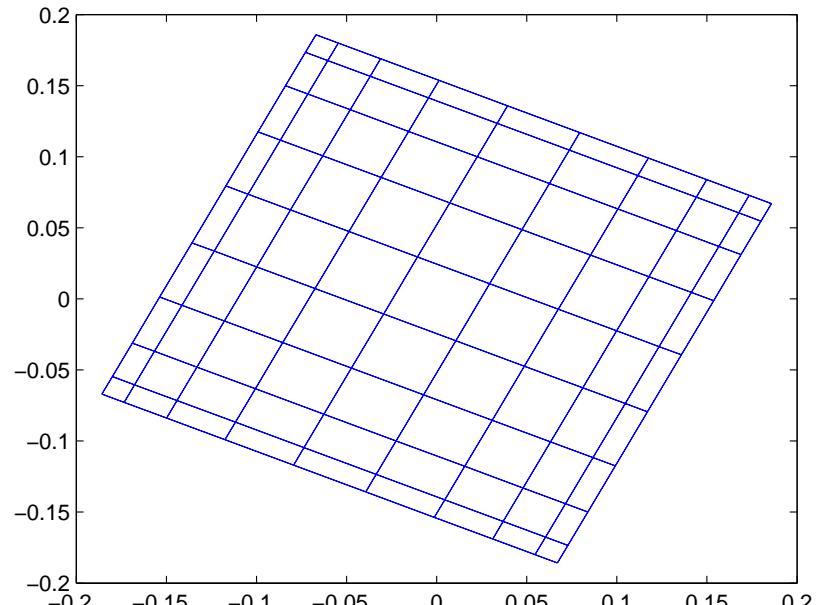
$$\langle x, Lx \rangle = x^T L x = \sum_{u \sim v} (x_u - x_v)^2$$

# Embedding a Grid

Adjacency

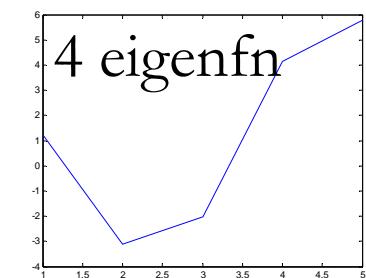
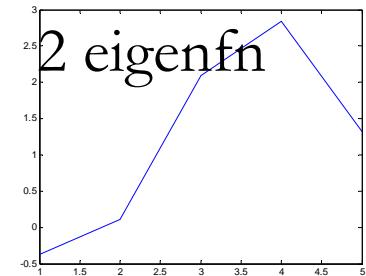
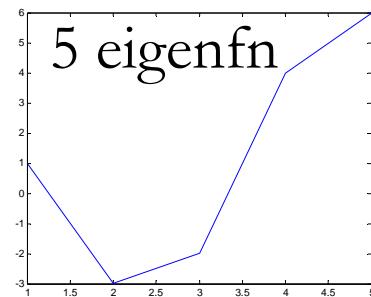
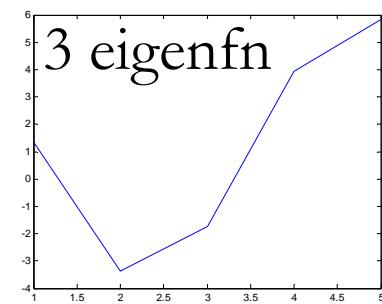
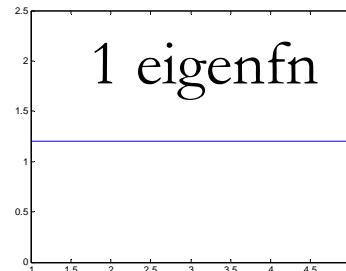
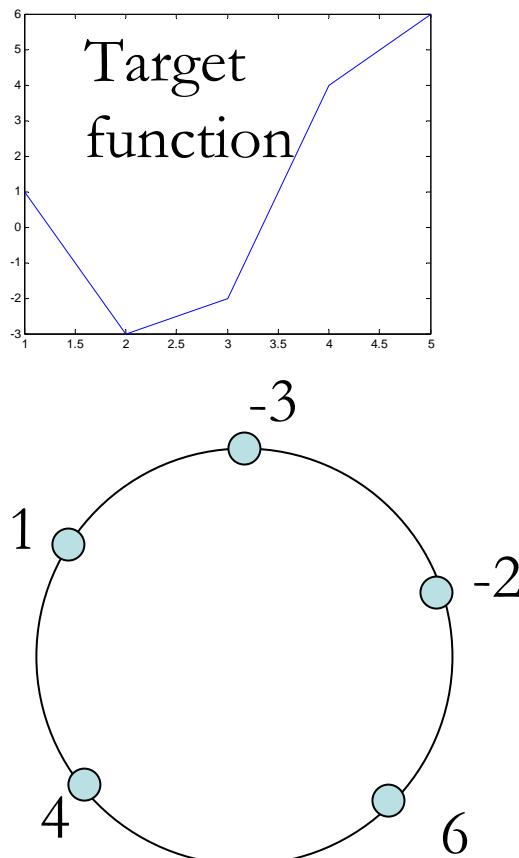


Laplacian

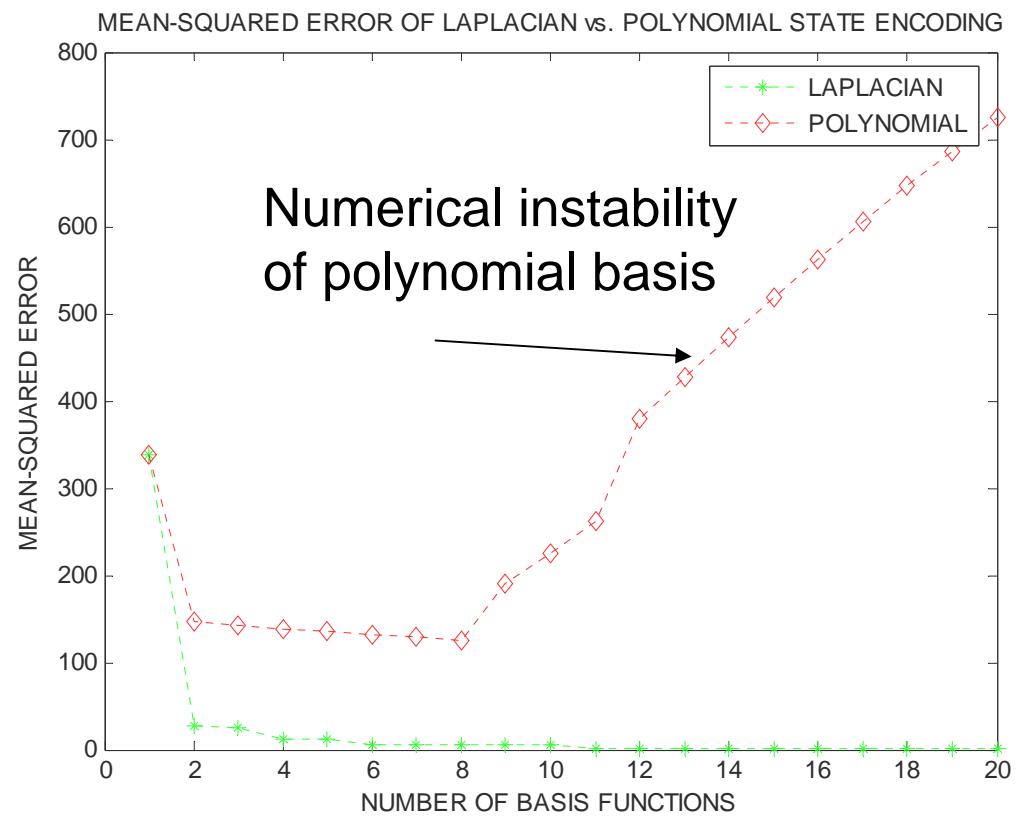
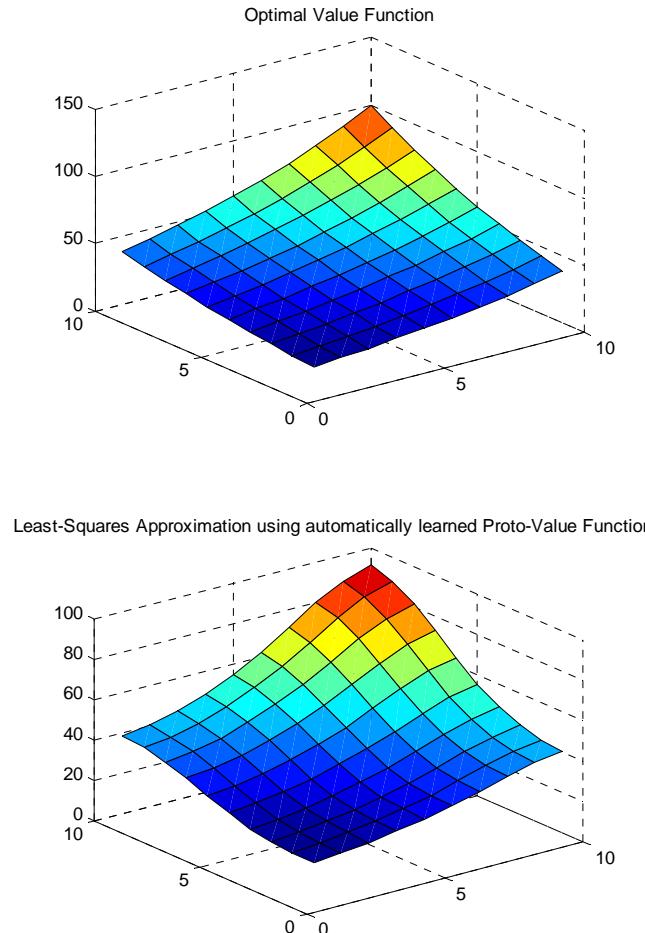


We'll discuss later why  
the Laplacian embedding  
of the grid is so regular

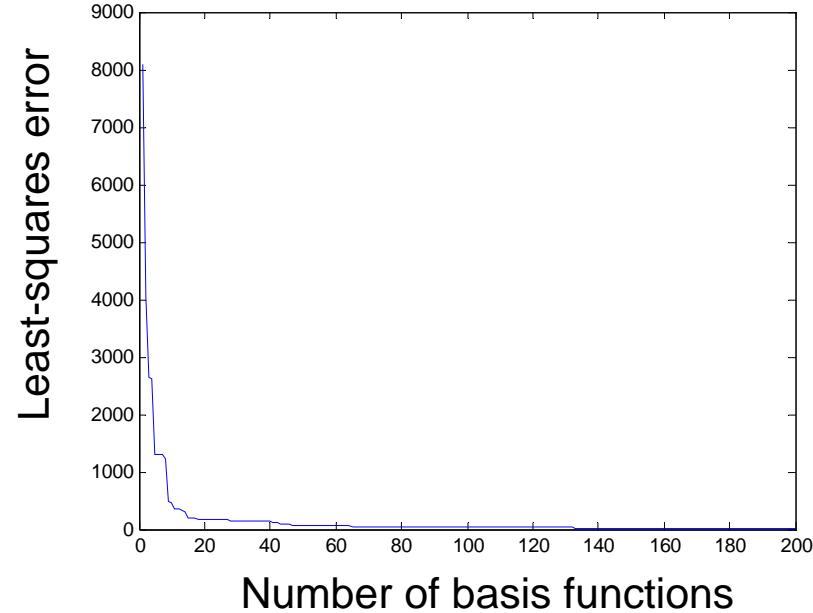
# Least-Square Projections onto Laplacian Eigenspaces



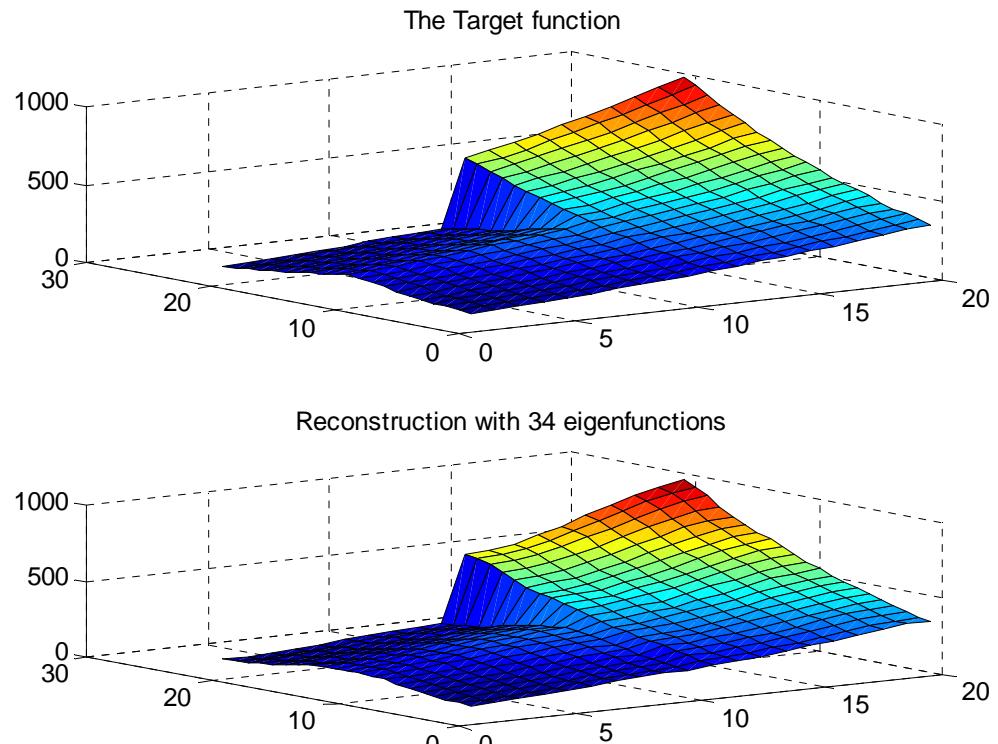
# Laplacian vs. Polynomial Approximation on a Grid



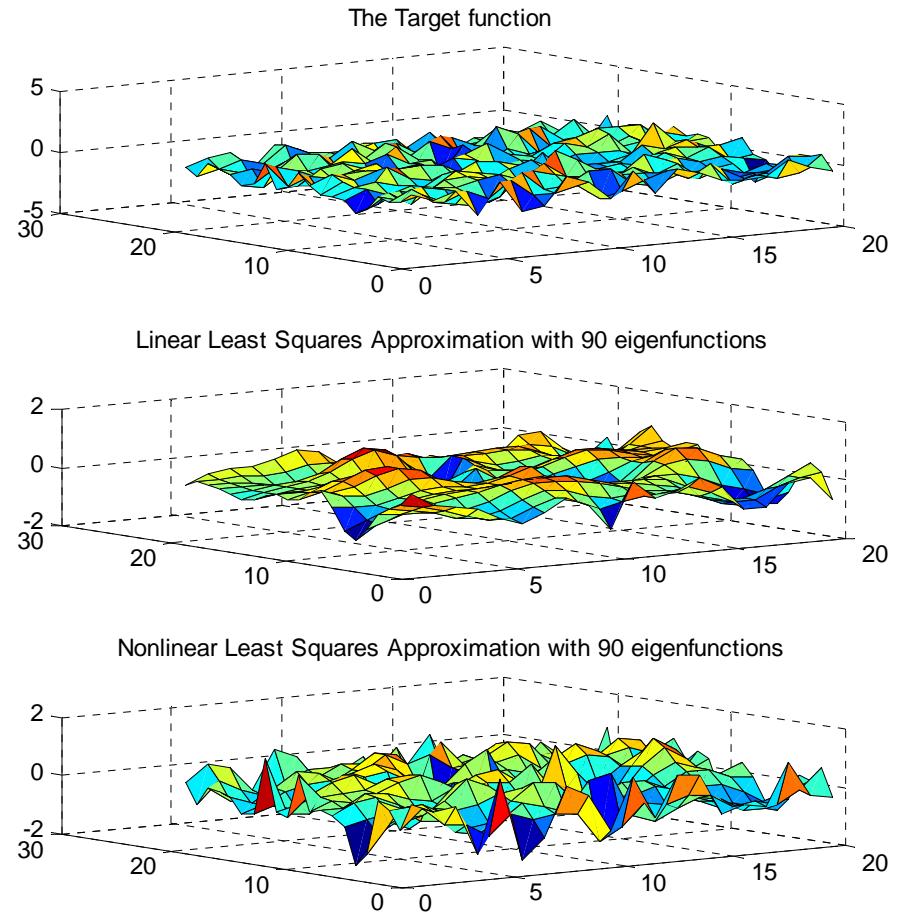
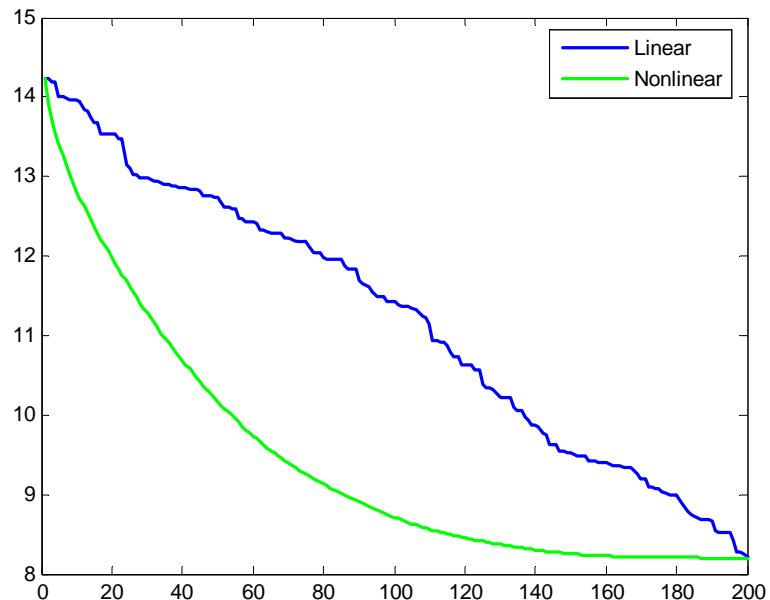
# Linear Least-Squares Approximation with Laplacian Bases



$$f' = \sum_{i=1}^n \langle f, \phi_i \rangle \phi_i$$



# Linear vs. Nonlinear LS Approximation of Random Function



# Graph Embedding

- Consider the following optimization problem mapping, where  $y_i \in R$  is a mapping of the  $i^{\text{th}}$  vertex to the real line
$$\text{Min}_w \sum_{i,j} (y_i - y_j)^2 w_{i,j} \text{ s.t. } y^\top D y = 1$$
- The best mapping is found by solving the generalized eigenvector problem

$$W \phi = \lambda D \phi$$

- If the graph is connected, this can be written as

$$D^{-1} W \phi = \lambda \phi$$

# Normalized Graph Laplacian and Random Walks

- Given an undirected weighted graph  $G = (V, E, W)$ , the random walk on the graph is defined by the transition matrix

$$P = D^{-1}W$$

- Random walk matrix is not symmetric
- Normalized Graph Laplacian

$$\mathcal{L} = D^{-1/2} (D - W) D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

- The random walk matrix has the same eigenvalues as  $(I - \mathcal{L})$

$$D^{-1}W = D^{-1/2} (D^{-1/2} W D^{-1/2}) D^{1/2} = D^{-1/2} (I - \mathcal{L}) D^{1/2}$$

# Operators on Graphs

Operator	Spectrum
Adjacency = $A$	Real, $ \lambda  \leq d_v$
C. Laplacian = $D - A$	PSD, $\lambda \geq 0$
N. Laplacian = $I - D^{-1/2}AD^{-1/2}$	PSD, $\lambda \in (0, 2)$
Random Walk = $D^{-1}A$	$\lambda \in \mathcal{R}$
DWT = $\mathcal{T}$	$\sigma(\mathcal{T}^k)$ decays fast

# Diffusion Analysis

- Fourier vs. wavelet analysis
  - Local vs. global analysis
  - Multiresolution modeling
- Diffusion wavelets [Coifman and Maggioni, 2004]
  - Generalization of wavelets to graphs and manifolds
  - Provides a way to learn multiscale basis functions
  - Automatic hierarchical abstraction of Markov process on graphs

## Multiscale analysis vs. Fourier Analysis

- Fourier vs. Wavelet Analysis:
  - Qualitative ideas: global vs. multiscale
  - Motivations: approximation, compression, denoising, computational efficiency; connections with Harmonic Analysis and approximation theory
  - Multiresolution modeling: multiscale approximation of processes
- Diffusion Wavelets
  - A way of generalizing wavelets to graphs and manifolds
  - Can be learnt once a graph is given
  - Automatically generates hierarchical view of a Markov process

## Qualitative ideas: multiscale vs. global

The setup is as before: we construct a set of basis functions adapted to the geometry of the explored state space, and project a policy iteration algorithm in a subspace spanned by those basis functions.

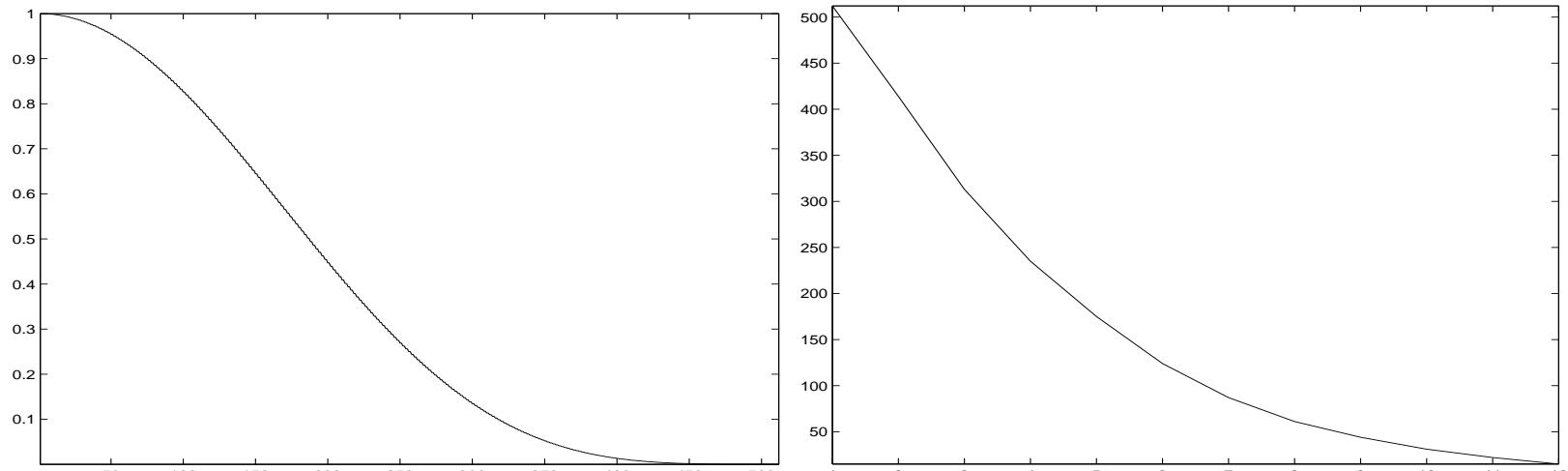
Instead of global Fourier-modes, we will use wavelet-like, multiscale basis elements. They are also built from a diffusion operator  $T$  on a graph. We denote them by  $\phi_{j,k}$ , where  $j$  will indicate scale, and  $k$  location. These allow to represent efficiently a broader class of functions than Fourier eigenfunctions, for example functions which are piecewise smooth and not globally smooth.

This wavelet analysis is multiscale in at least three ways:

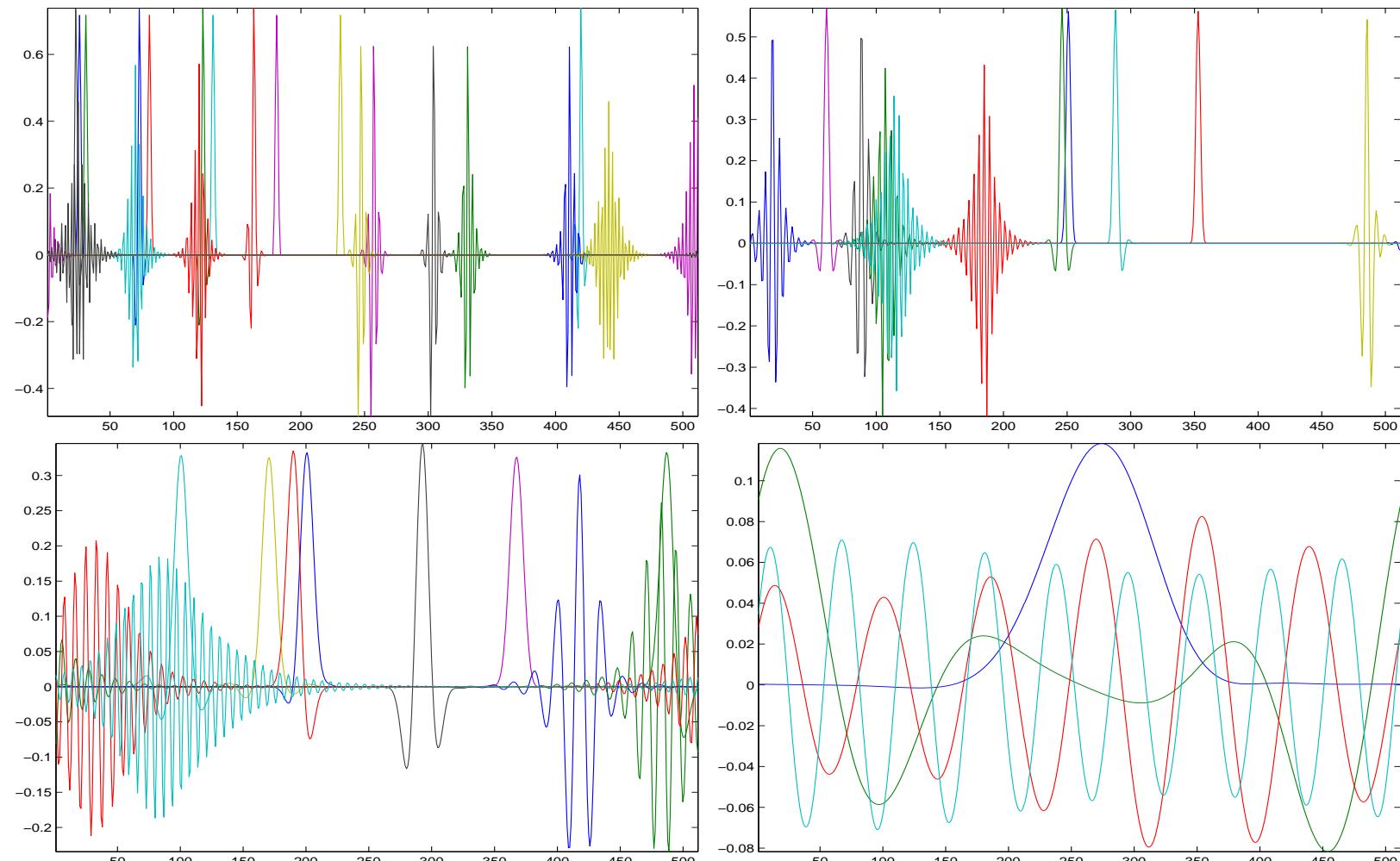
- in *space*: basis elements are localized, the elements at scale  $j$  have support of roughly diameter  $\delta^j$ , for some  $\delta > 1$ ;
- in *frequency*: basis elements at scale  $j$  have Fourier transform essentially supported in  $[\epsilon^{2^{-j}}, \epsilon^{2^{-j+1}}]$ ;
- in *time*: it is possible to represent  $T^{2^j}$  on  $\{\phi_{j,k}\}_k$  by a small matrix, with great precision.

## What do diffusion wavelets look like? (I)

For pedagogical purposes, we illustrate the construction in a very simple example. We consider the Laplacian on the circle  $\mathbb{T}$ , since even in this case the multiresolution analysis we introduce is new.

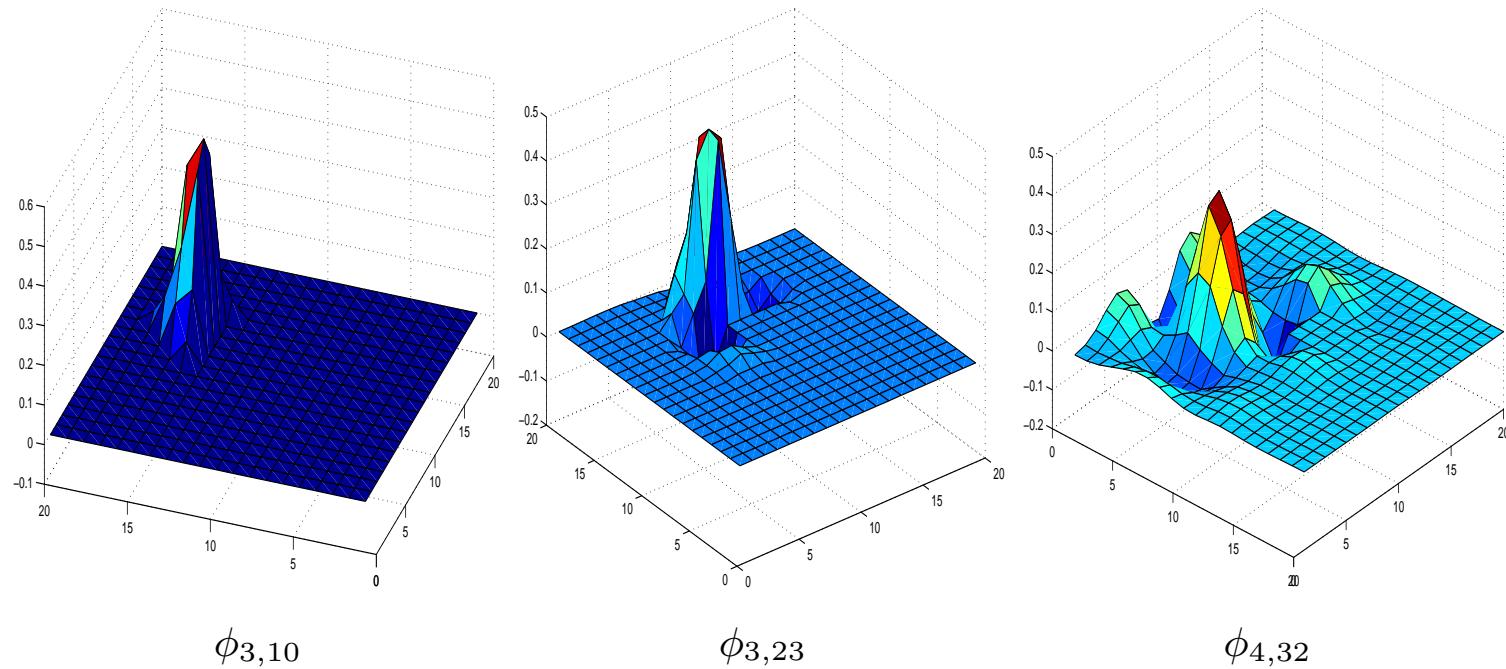


Left: the spectrum of  $T$ . Right: the dimension of  $V_j$  as a function  $j$ .

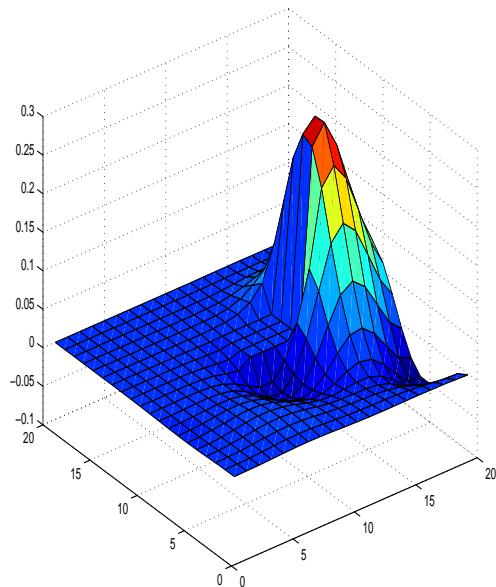
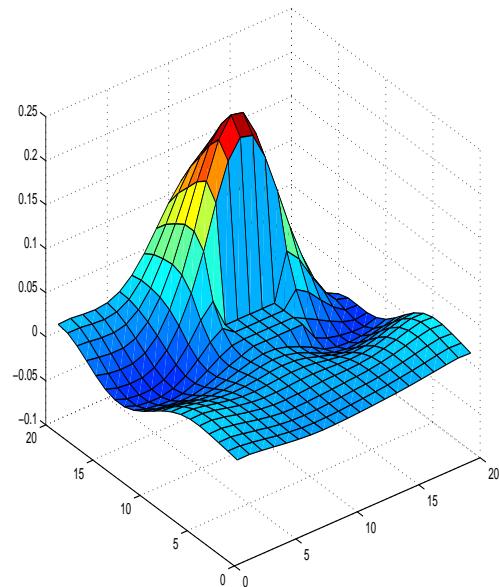
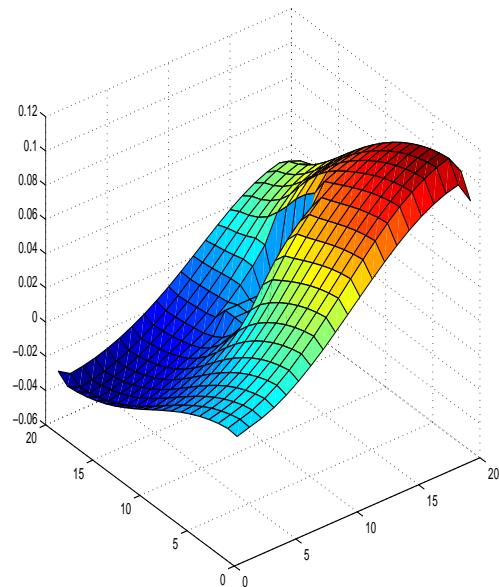


Some scaling functions in  $V_1$  (top left), in  $V_3$  (top right),  $V_6$  (bottom left) and  $V_{12}$  (bottom right).

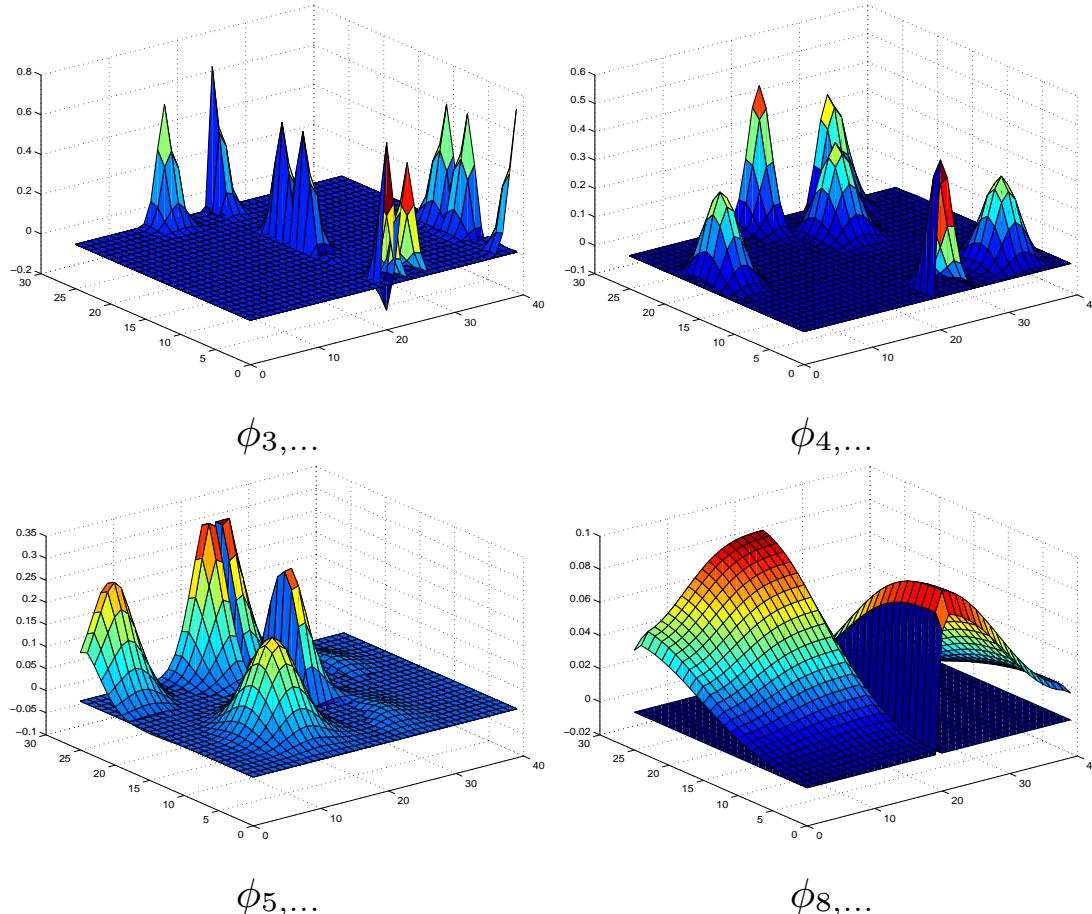
## What do diffusion wavelets look like? (II)



Room with an obstacle in the middle. Diffusion scaling functions are automatically adapted to the state space. First index denotes scale (the larger the index the coarser the scale) and the second one indexes the location. Compared to the eigenfunctions of the Laplacian, each of which has global support, the scaling functions are localized at different scales.

 $\phi_{5,5}$  $\phi_{6,8}$  $\phi_{9,2}$

## What do diffusion wavelets look like? (III)



Diffusion scaling functions on a discrete two-room spatial domain connected by a common door. All the scaling functions are naturally adapted to the state space.

## Connections with Harmonic Analysis, I

We need tools for efficiently working with functions on a manifold or graph: in particular efficient and stable representation for functions of interest (e.g. value functions). Assume a linear architecture:

$$f = \sum_k \alpha_k \phi_k$$

where  $f$  is a function in the class of functions we want to approximate,  $\phi_k$ 's are basis functions (“building blocks” or “templates”), and the coefficients  $\alpha_k$  contain the information for putting together the “building blocks” in order to reconstruct (or approximate)  $f$ .

What does **efficient** mean? Few, in proportion to how “complicate”  $f$  is, and efficiently-organized coefficients  $\alpha_k$ . Smoothness constraints become **sparsity** constraints. For example, it is useful for *linear* approximation, that  $|\alpha_k| \lesssim k^{-\gamma}$ . Or, for *nonlinear* approximation, that  $|\alpha_{\sigma(k)}| \lesssim k^{-\gamma}$ , for some permutation  $\sigma$  (possibly  $\sigma(k) \gg k!$ ).

## Connections with Harmonic Analysis, II

### [Enter Fourier]

- (i) Fourier: approximate solutions of the heat equation on an interval or rectangle with sine and cosine functions:  $\phi_k(x) = \sin(kx)$ .
- (ii) Fourier on Euclidean domains: instead of sines and cosines need the eigenfunctions of the Laplacian on the domain:  $\phi_k$ :

$$\Delta\phi_k = \lambda_k\phi_k .$$

- (iii) Fourier on manifolds and graphs: as above, with the natural Laplace-Beltrami operator, or the graph Laplacian.

The good and the bad: FFT,  $\phi_k$ 's are global approximants, and  $\alpha_k$  are not as sparse as one may wish. Example:  $f \in \mathcal{C}^s$  iff  $|\hat{f}(k)| = |\langle f, \phi_k \rangle| \leq k^{-s-1}$  (modulo a “small lie”).

## Connections with Harmonic Analysis, III

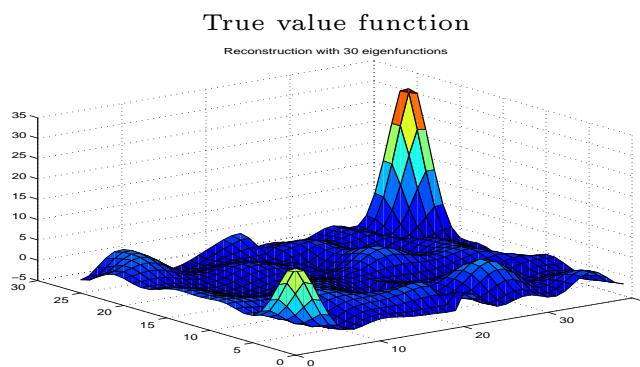
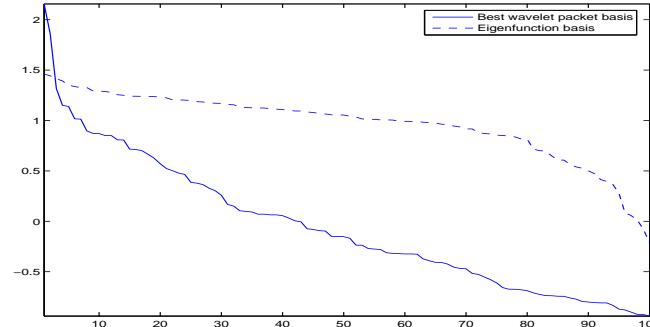
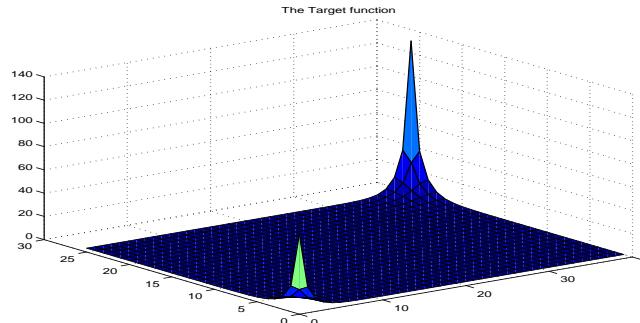
### Wavelets and Multiresolution Analysis:

Wavelets are concentrated both in time and frequency. Wavelets have to indices  $\phi_{j,k}$  is an “atom” concentrated in time at position  $k$ , width about  $2^{-j}$ , and concentrated around frequency  $2^j$ . They provide essentially the best possible building blocks for interesting and large classes of functions, i.e. much fewer  $\alpha_k$ ’s in the representation of these functions.

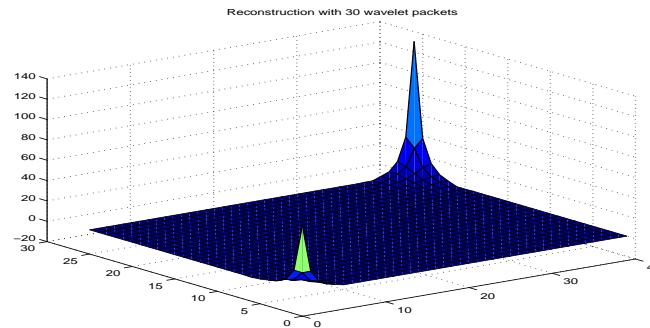
Initially constructed on  $\mathbb{R}$  (late 80’s), then on  $\mathbb{R}^n$ , and constructions on meshed surfaces (graphics, PDEs).

They characterize *local* regularity, vs. the global regularity characterized by Fourier coefficients.

## Example of nonlinear approximation



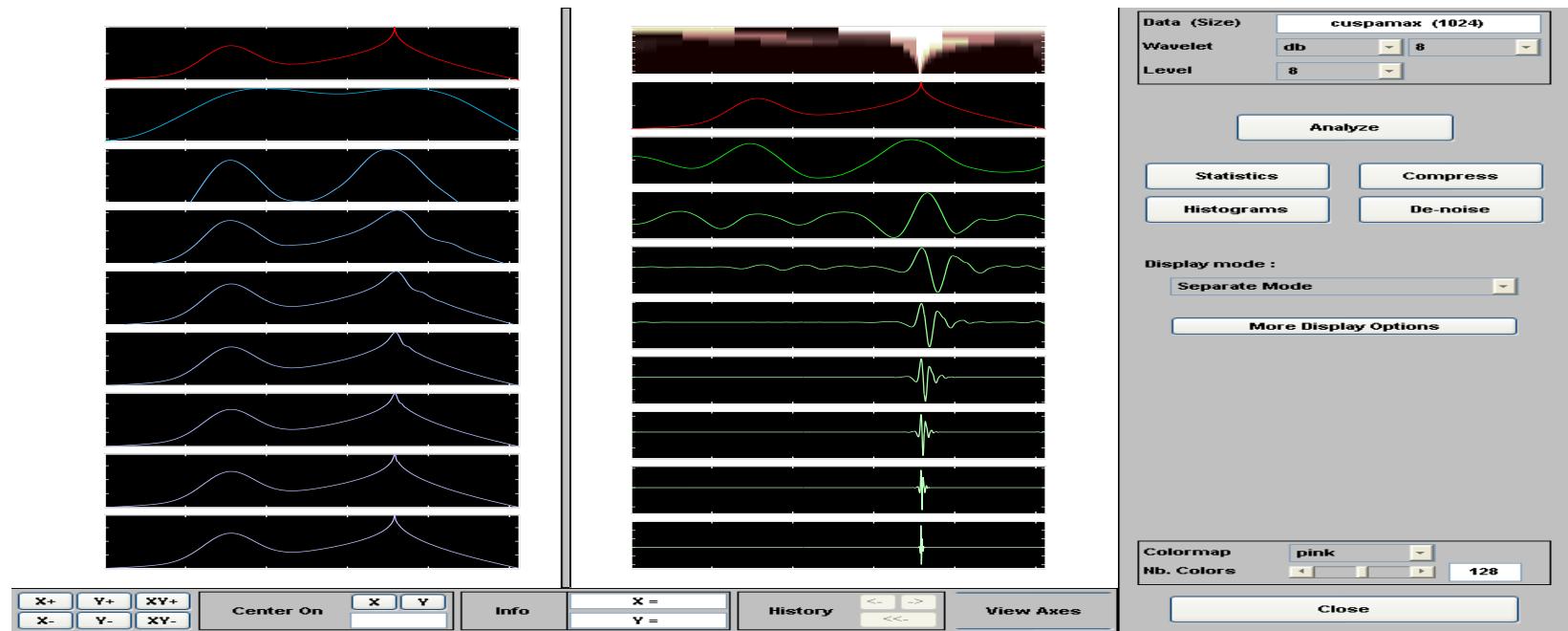
Fourier and DWT nonlinear approx



Fourier nonlinear approx. (30 coeffs)  
DWT nonlinear approx. (30 coeffs)

Approximation of a value function which is not very smooth because of localized peaks that induce local large gradients. This value function corresponds to a discrete two room domain connected by a door in the middle, with two (positive but different) rewards at the two opposite corners of the two rooms.

## Example of Wavelet Transform



Wavelet transform. First column, top to bottom: projections onto scaling subspaces  $V_j$  (Daubechies-8) at increasing resolution. Top of second column: wavelet transform of the original signal: horizontal axis corresponds to location, vertical axis to scales (finer scales at the bottom). Second plot: reconstructed signal. Other plots in second column: wavelet coefficients at increasing resolution. With 4.65% of the coefficients it is possible to recover 99.98% of the energy ( $\mathbb{L}^2$ -norm) of the signal.

## Multiscale geometry and Markov processes

In many situations the graph/manifold representing the state space contains clusters at different scales, separated by bottlenecks. A Markov process (e.g. associated with a policy) on such a space will be “nearly decomposable” (or lumpable), at different time-scales. For example in the two-room problem, at a certain large time scale, may be approximated by a two-state problem. In general there may be more bottlenecks and decompositions, depending on the time-scale at which the problem is considered.

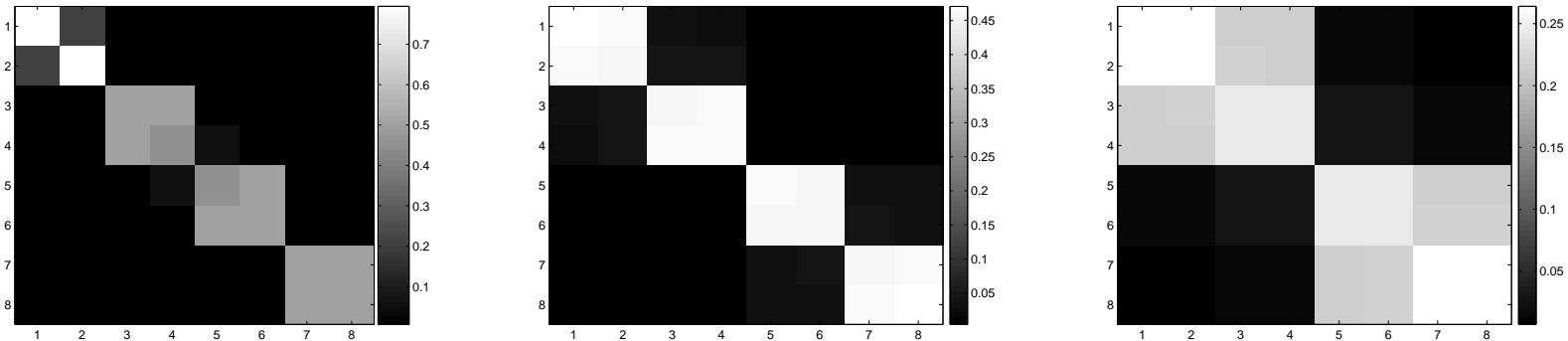
This generalizes Euclidean constructions of wavelets, much used in mathematical analysis and signal processing, and extends it not only to the analysis of functions, but also to the analysis of Markov processes.

## Abstraction of Markov chains

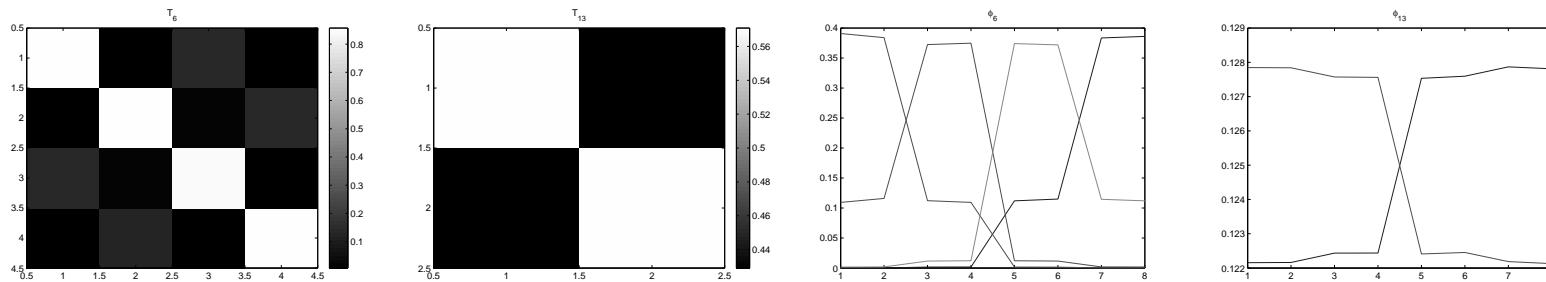
We now consider a simple example of a Markov chain on a graph with 8 states.

$$T = \begin{pmatrix} 0.80 & 0.20 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.20 & 0.79 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.01 & 0.49 & 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.499 & 0.001 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.001 & 0.499 & 0.50 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.50 & 0.49 & 0.01 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.01 & 0.49 & 0.50 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.50 & 0.50 \end{pmatrix}$$

From the matrix it is clear that the states are grouped into four pairs  $\{\nu_1, \nu_2\}$ ,  $\{\nu_3, \nu_4\}$ ,  $\{\nu_5, \nu_6\}$ , and  $\{\nu_7, \nu_8\}$ , with weak interactions between the the pairs.



Some dyadic powers of the Markov chain  $T$ .



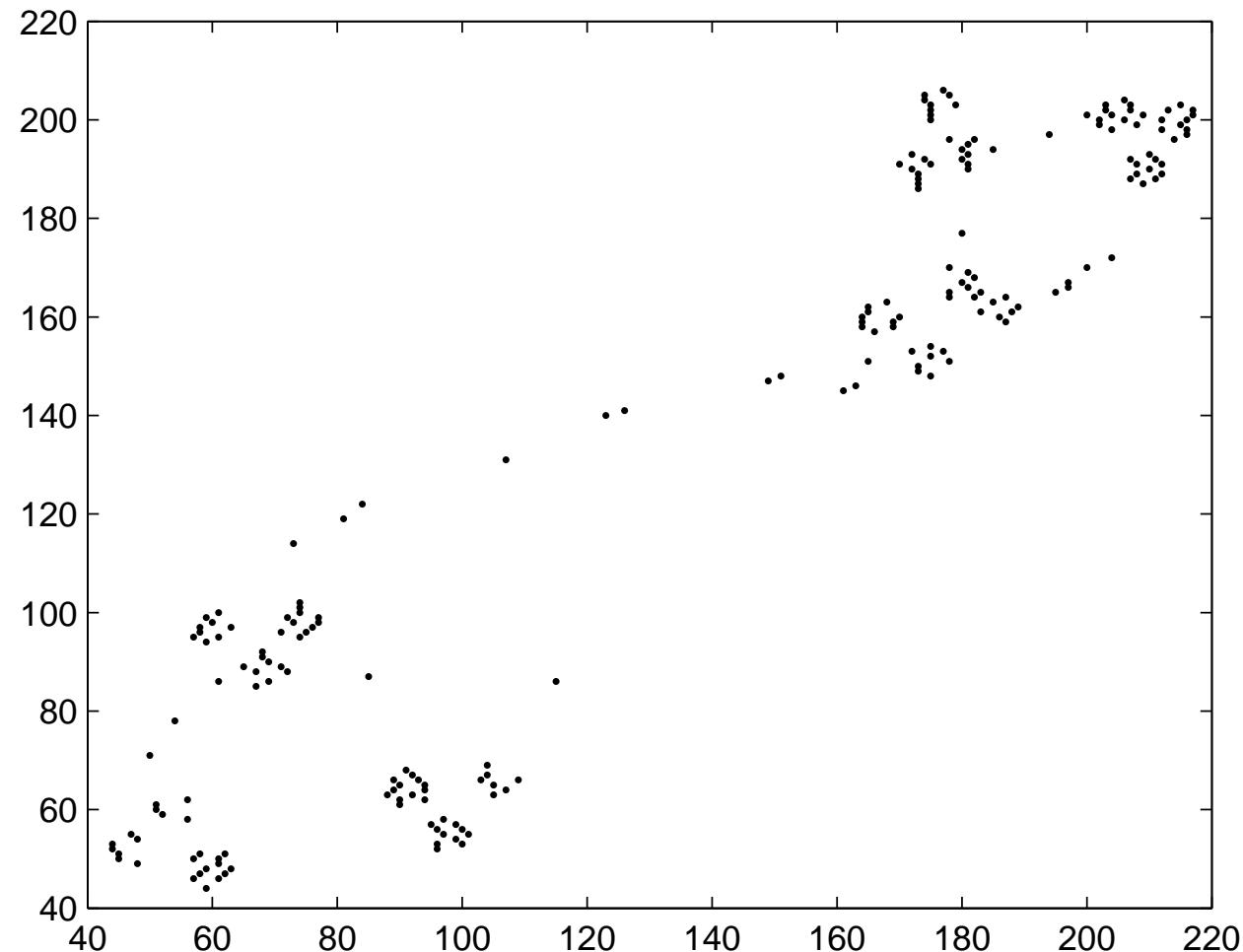
$T^{2^6}$  compressed

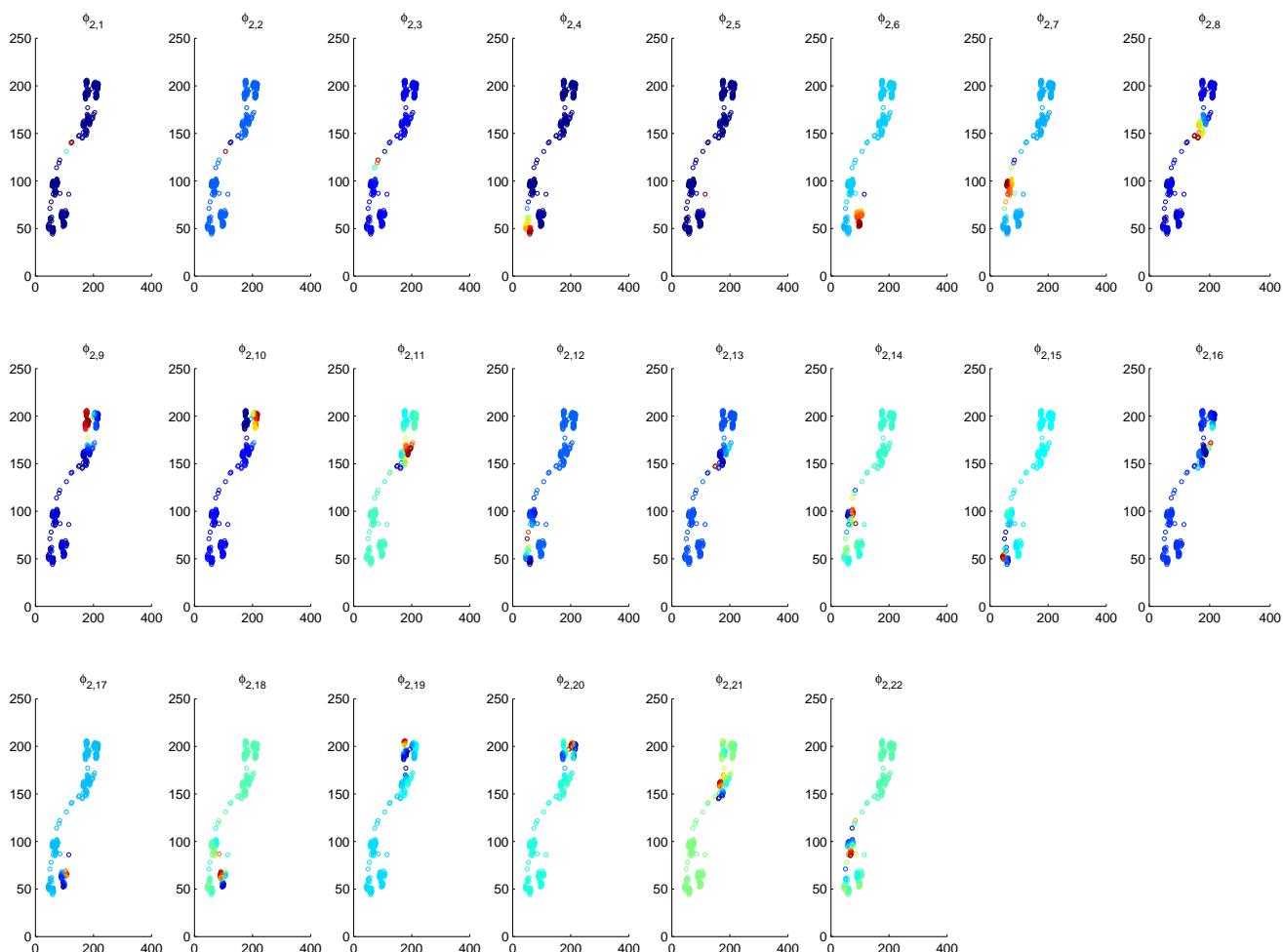
$T^{2^{13}}$  compressed

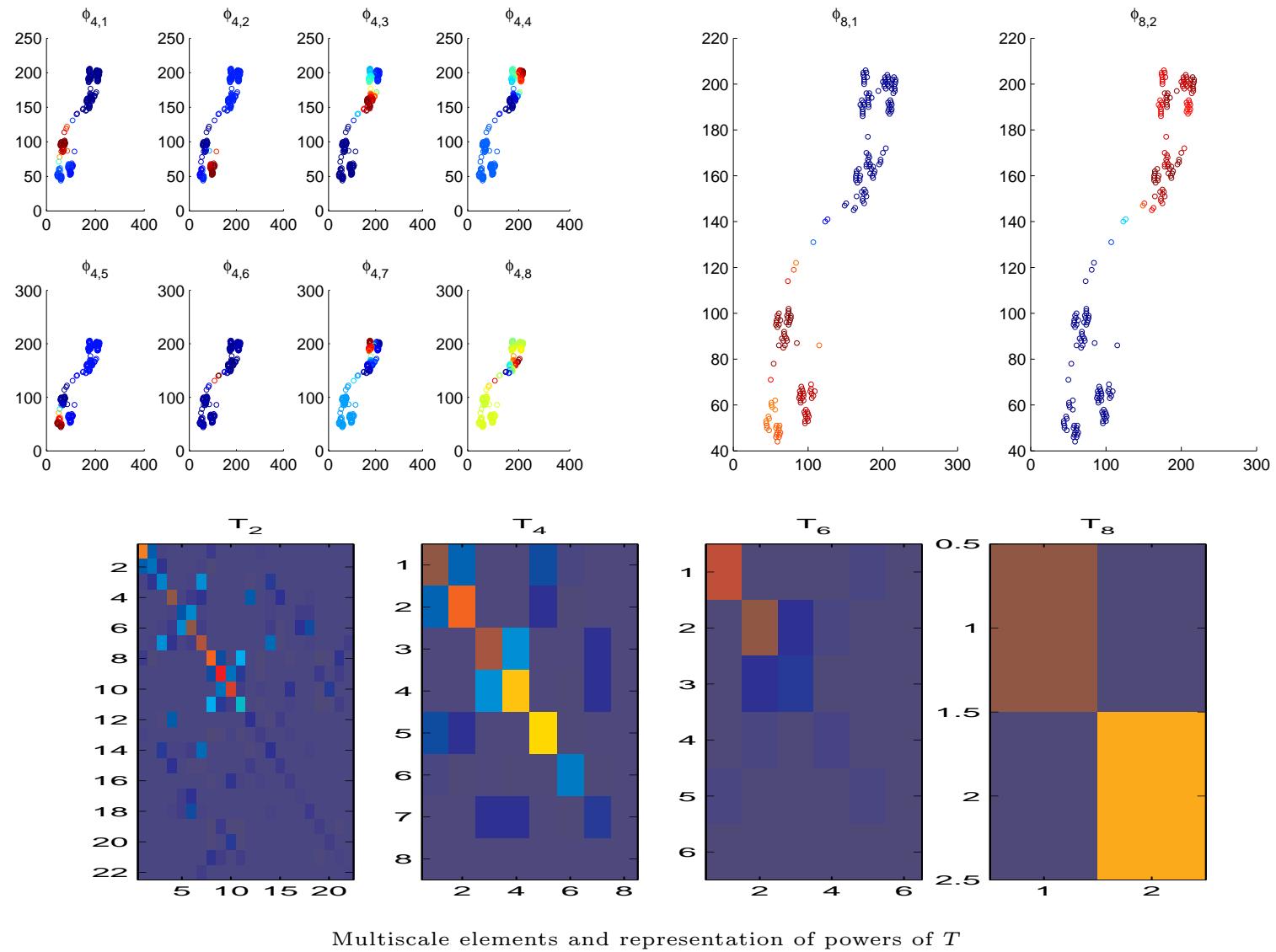
$\phi_6, \dots$

$\phi_{13}, \dots$

## A multiscale “network”







## Multiscale Analysis, I

We construct multiscale bases on manifolds, graphs, point clouds.

Classical constructions of wavelets are based on geometric transformations (such as dilations, translations) of the space, transformed into actions (e.g. via representations) on functions. There are plenty of such transformations on  $\mathbb{R}^n$ , certain classes of Lie groups and homogeneous spaces (with automorphisms that resemble “anisotropic dilations”), and manifolds with large groups of transformations.

Here the space is in general highly *non-symmetric*, not invariant under ”natural” geometric transformation, and moreover it is “*noisy*”.

Idea: use *diffusion and the heat kernel as dilations*, acting on functions on the space, to generate multiple scales.

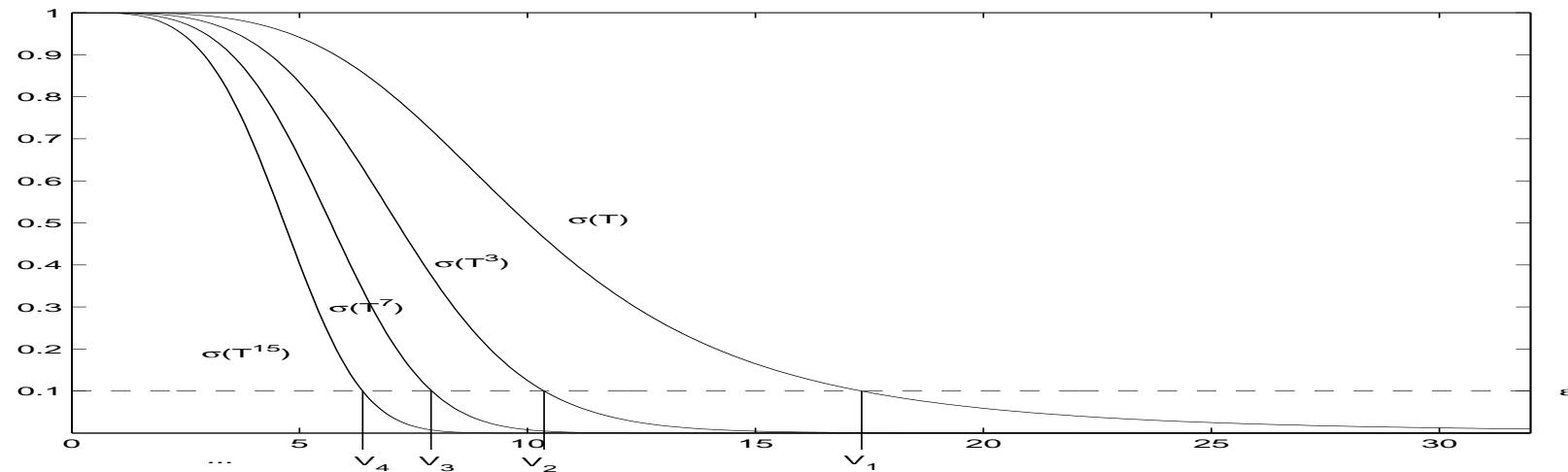
This is connected with the work on diffusion or Markov semigroups, and Littlewood-Paley theory of such semigroups (a la Stein).

We would like to have *constructive* methods for efficiently computing the multiscale decompositions and the wavelet bases.

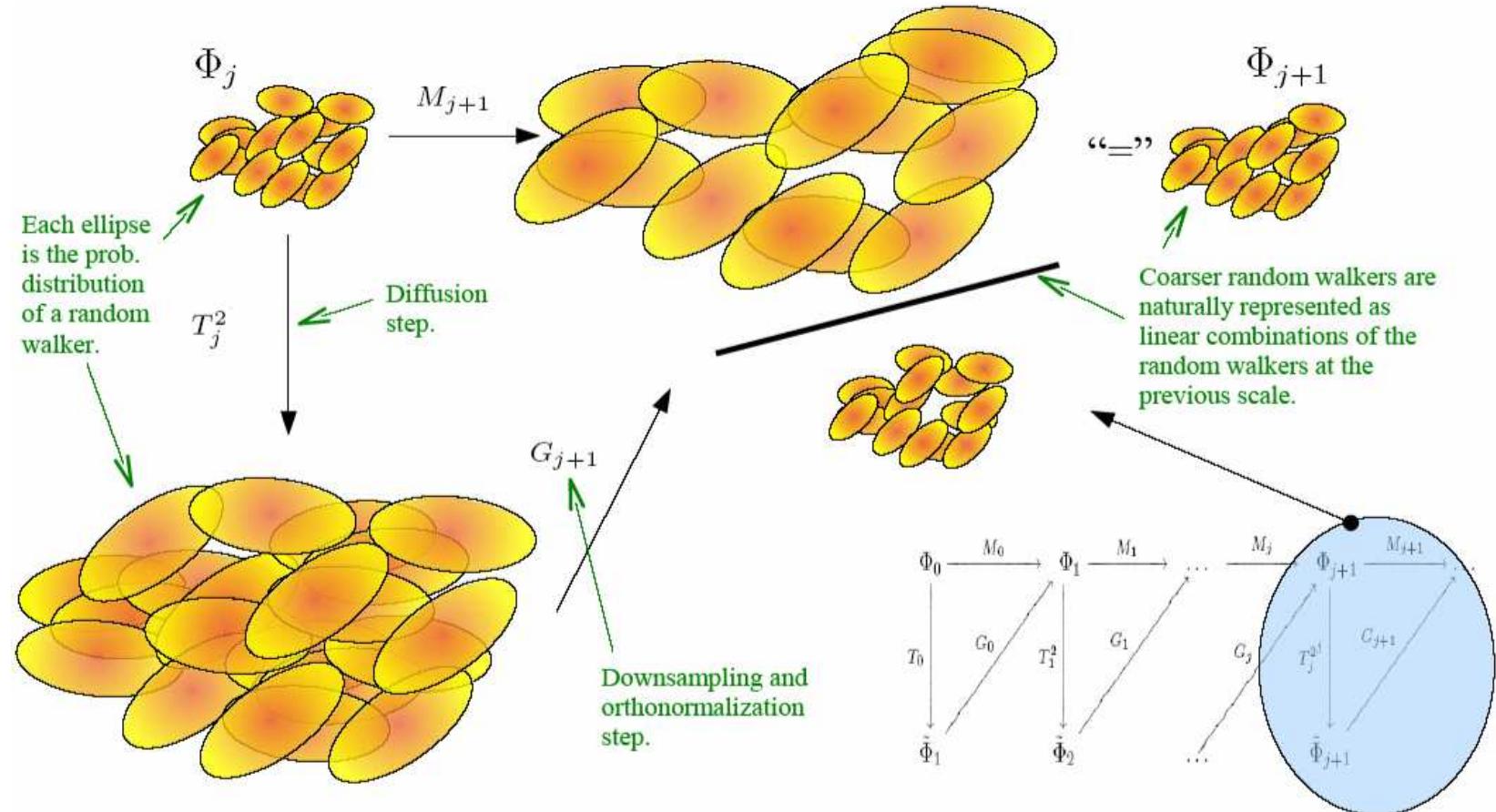
## Multiscale Analysis, II

Suppose for simplicity we have a weighted graph  $(G, E, W)$ , with corresponding Laplacian  $\mathcal{L}$  and random walk  $P$ . Let us renormalize, if necessary,  $P$  so it has norm 1 as an operator on  $L^2$ : let  $T$  be this operator. Assume for simplicity that  $T$  is self-adjoint, and high powers of  $T$  are low-rank:  $T$  is a diffusion, so range of  $T^t$  is spanned by smooth functions of increasingly (in  $t$ ) smaller gradient.

A “typical” spectrum for the powers of  $T$  would look like this:



## Construction of Diffusion Wavelets



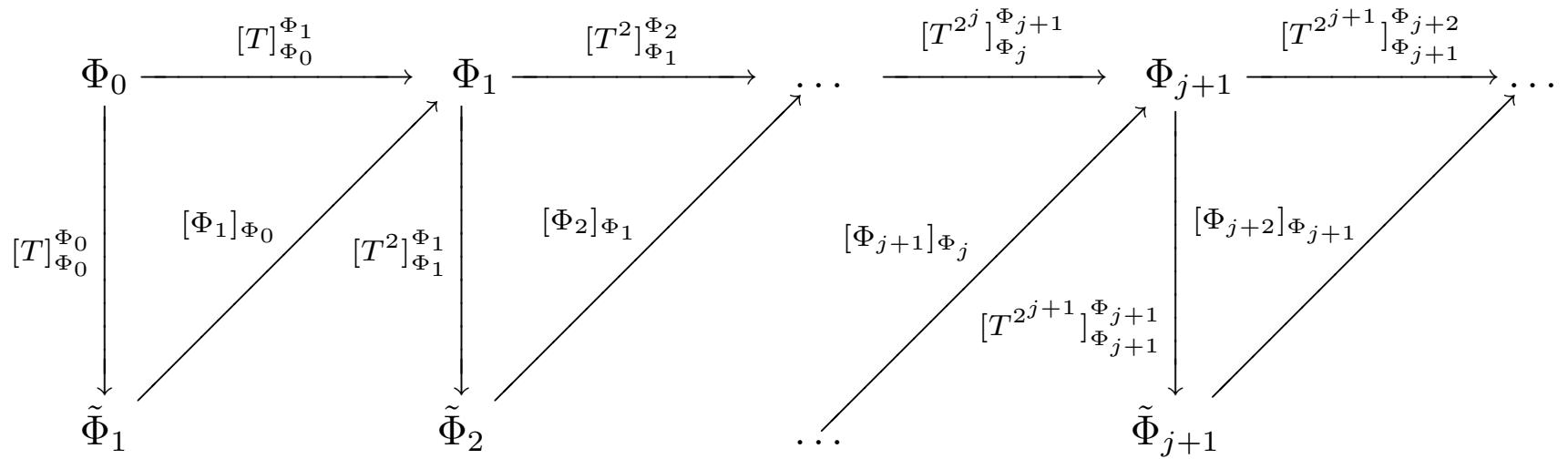


Diagram for downsampling, orthogonalization and operator compression. (All triangles are commutative by construction)

```

 $\{\Phi_j\}_{j=0}^J, \{\Psi_j\}_{j=0}^{J-1}, \{[T^{2^j}]_{\Phi_j}^{\Phi_j}\}_{j=1}^J \leftarrow \text{DiffusionWaveletTree } ([T]_{\Phi_0}^{\Phi_0}, \Phi_0, J, \text{SpQR}, \epsilon)$ 
//  $[T]_{\Phi_0}^{\Phi_0}$  : a diffusion operator, written on the o.n. basis  $\Phi_0$ 
//  $\Phi_0$  : an orthonormal basis which  $\epsilon$ -spans  $V_0$ 
//  $J$  : number of levels to compute
// SpQR : a function compute a sparse  $QR$  decomposition, template below.
//  $\epsilon$ : precision
// Output: The orthonormal bases of scaling functions,  $\Phi_j$ , wavelets,  $\Psi_j$ , and
// compressed representation of  $T^{2^j}$  on  $\Phi_j$ , for  $j$  in the requested range.

```

**for**  $j = 0$  **to**  $J - 1$  **do**

$$\begin{aligned}
 & [\Phi_{j+1}]_{\Phi_j}, [T]_{\Phi_0}^{\Phi_1} \leftarrow \text{SpQR}([T^{2^j}]_{\Phi_j}^{\Phi_j}, \epsilon) \\
 & T_{j+1} := [T^{2^{j+1}}]_{\Phi_{j+1}}^{\Phi_{j+1}} \leftarrow [\Phi_{j+1}]_{\Phi_j} [T^{2^j}]_{\Phi_j}^{\Phi_j} [\Phi_{j+1}]_{\Phi_j}^* \\
 & [\Psi_j]_{\Phi_j} \leftarrow \text{SpQR}(I_{\langle \Phi_j \rangle} - [\Phi_{j+1}]_{\Phi_j} [\Phi_{j+1}]_{\Phi_j}^*, \epsilon)
 \end{aligned}$$

**end**

```

 $Q, R \leftarrow \text{SpQR } (A, \epsilon)$  //  $A$ : sparse  $n \times n$  matrix,  $\epsilon$ : precision
// Output:  $Q, R$  matrices, hopefully sparse, such that  $A =_\epsilon QR$ ,  $Q$  is  $n \times m$  and orthogonal,
//  $R$  is  $m \times n$ , and upper triangular up to a permutation,
// the columns of  $Q$   $\epsilon$ -span the space spanned by the columns of  $A$ .

```

## A Diffusion Multiresolution on the Circle (cont'd)

For pedagogical purposes, we illustrate the construction in a very simple example. We consider the Laplacian on the circle  $\mathbb{T}$ , since even in this case the multiresolution analysis we introduce is new.

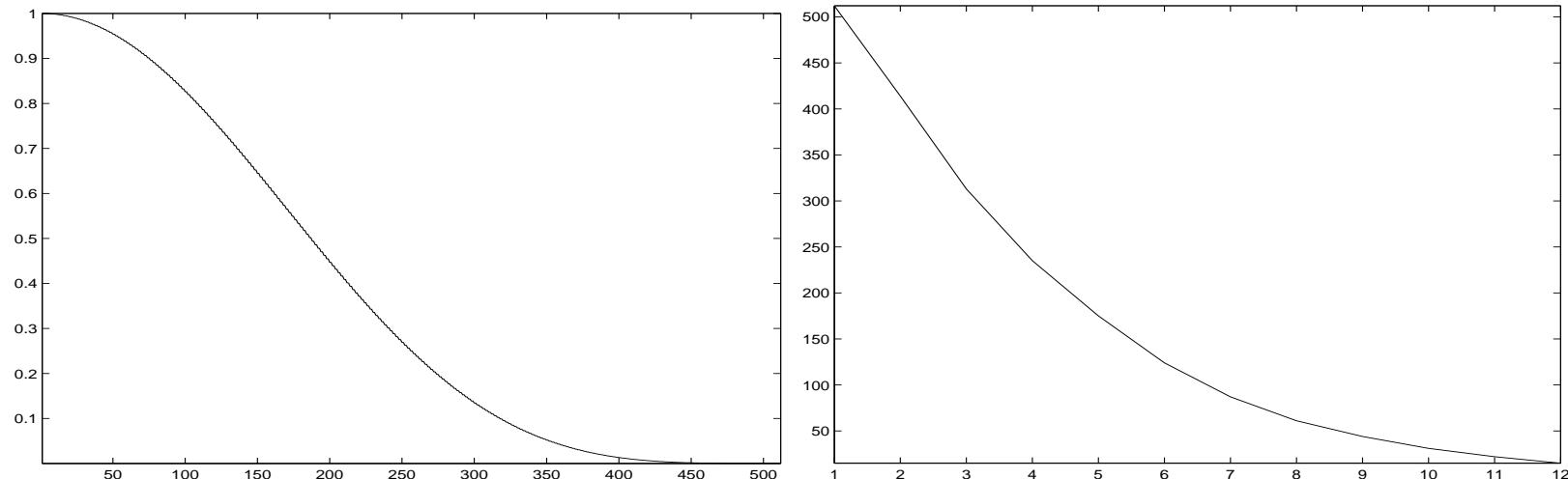


Figure 1: Left: the spectrum of  $T$ . Right: the dimension of  $V_j$  as a function  $j$ .

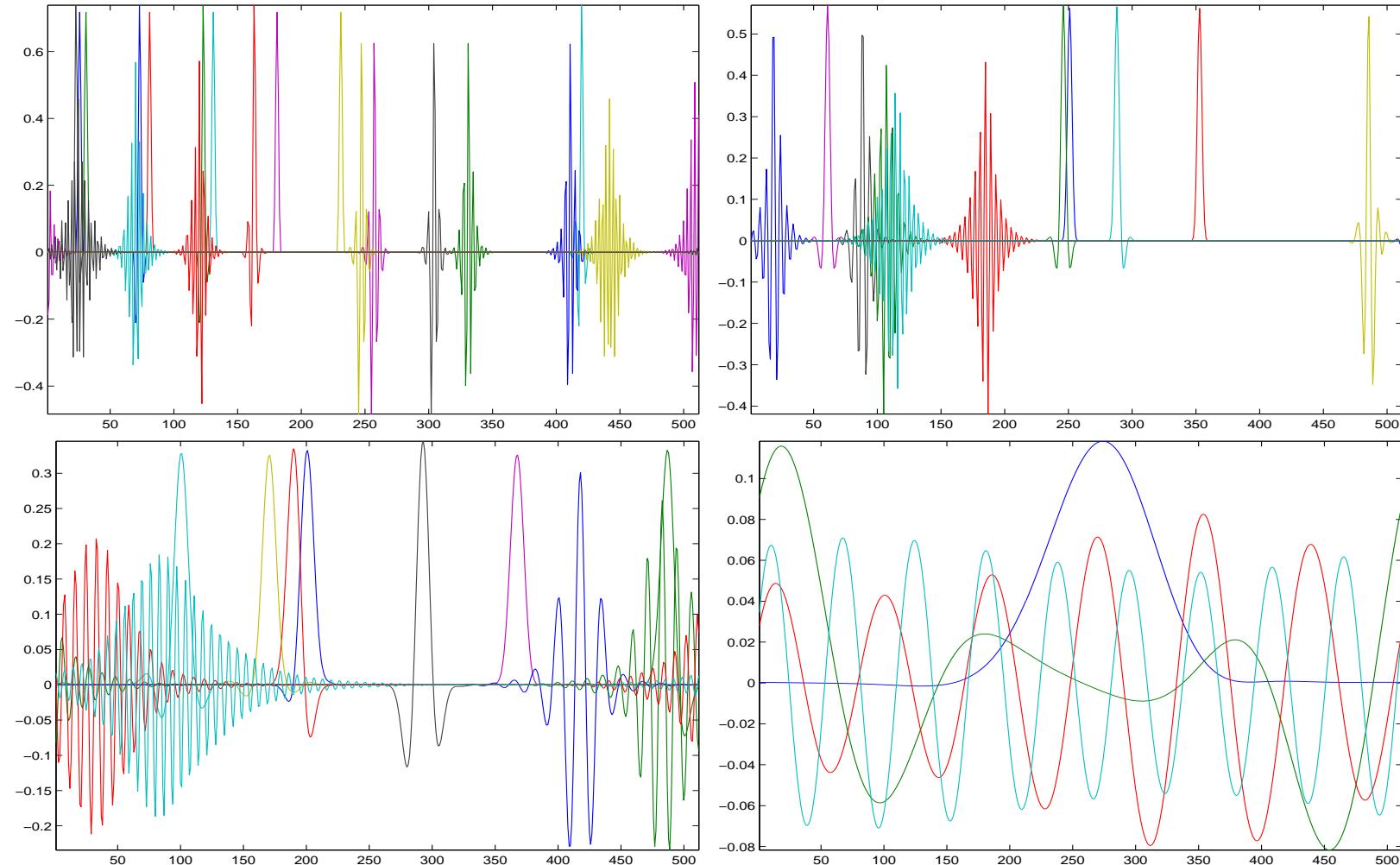


Figure 2: Some scaling functions in  $V_1$  (top left), in  $V_3$  (top right),  $V_6$  (bottom left) and  $V_{12}$  (bottom right).

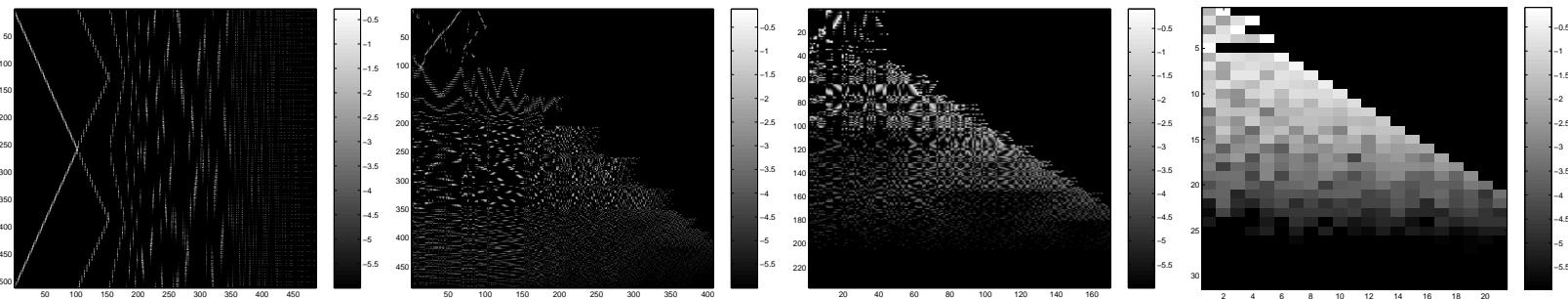


Figure 3: The scaling function filters  $M_1$  (top left),  $M_2$  (top right),  $M_5$  (bottom left) and  $M_{11}$  (bottom right). The images are in logarithmic scale to show entries larger than  $10^{-6}$ .

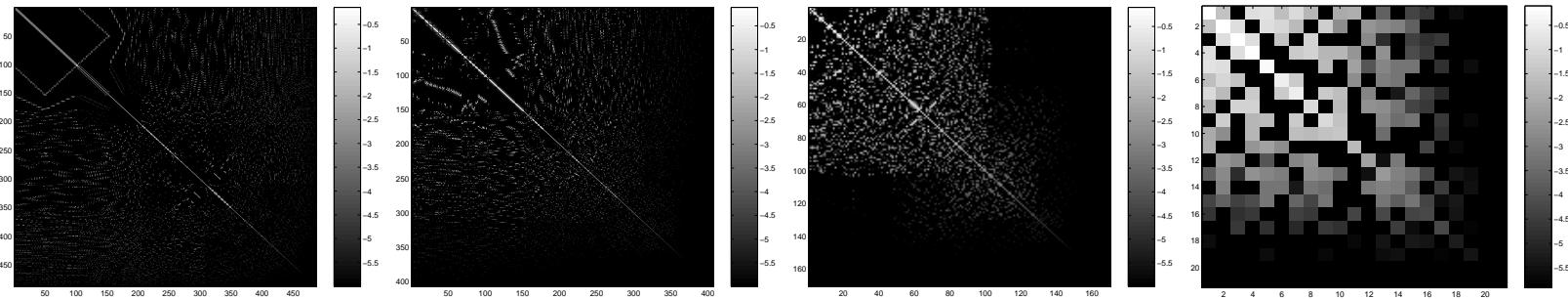


Figure 4: The compressed powers of  $T$  at scale  $j = 1, 2, 5, 11$ . The images are in logarithmic scale to show entries larger than  $10^{-6}$ .

## Thinking multiscale on graphs...

Investigating other constructions:

- Biorthogonal diffusion wavelets, in which scaling functions are probability densities (useful for multiscale Markov chains)
- Top-bottom constructions: recursive subdivision
- Both...

Applications besides Markov Decision Processes:

- Document organization and classification
- Nonlinear Analysis of Images
- Semi-supervised learning through diffusion processes on data

# Outline

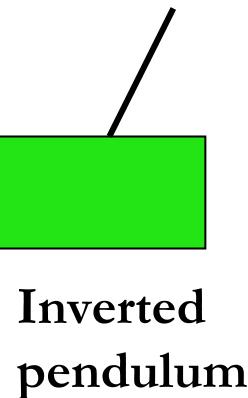
- **Part One**
  - History and Motivation (9:00-9:15)
  - Overview of framework (9:15-9:30)
  - Technical Background (9:30-10:30)
  - Questions: (10:30-10:45)
- **Part Two**
  - Algorithms and implementation (11:15-11:45)
  - Experimental Results (11:45-12:30)
  - Discussion and Future Work (12:30-12:45)
  - Questions (12:45-1:00)

# Outline

- **Part One**
  - History and Motivation (9:00-9:15)
  - Overview of framework (9:15-9:30)
  - Technical Background (9:30-10:30)
  - Questions: (10:30-10:45)
- Part Two
  - Algorithms and implementation (11:15-11:45)
  - Experimental Results (11:45-12:30)
  - Discussion and Future Work (12:30-12:45)
  - Questions (12:45-1:00)

# Overview of Framework

(Mahadevan, AAAI, ICML, UAI 2005; Mahadevan & Maggioni, NIPS 2005;  
Maggioni and Mahadevan, ICML 2006)



Samples from random walk  
on manifold in state space

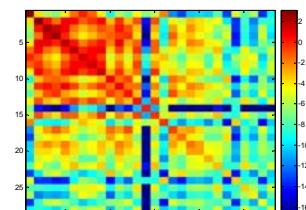
Build  
Graph

Fourier

Wavelets

$$\Delta\phi = \lambda\phi$$

Eigenfunctions of  
Laplacian on state  
space: **global** basis  
functions



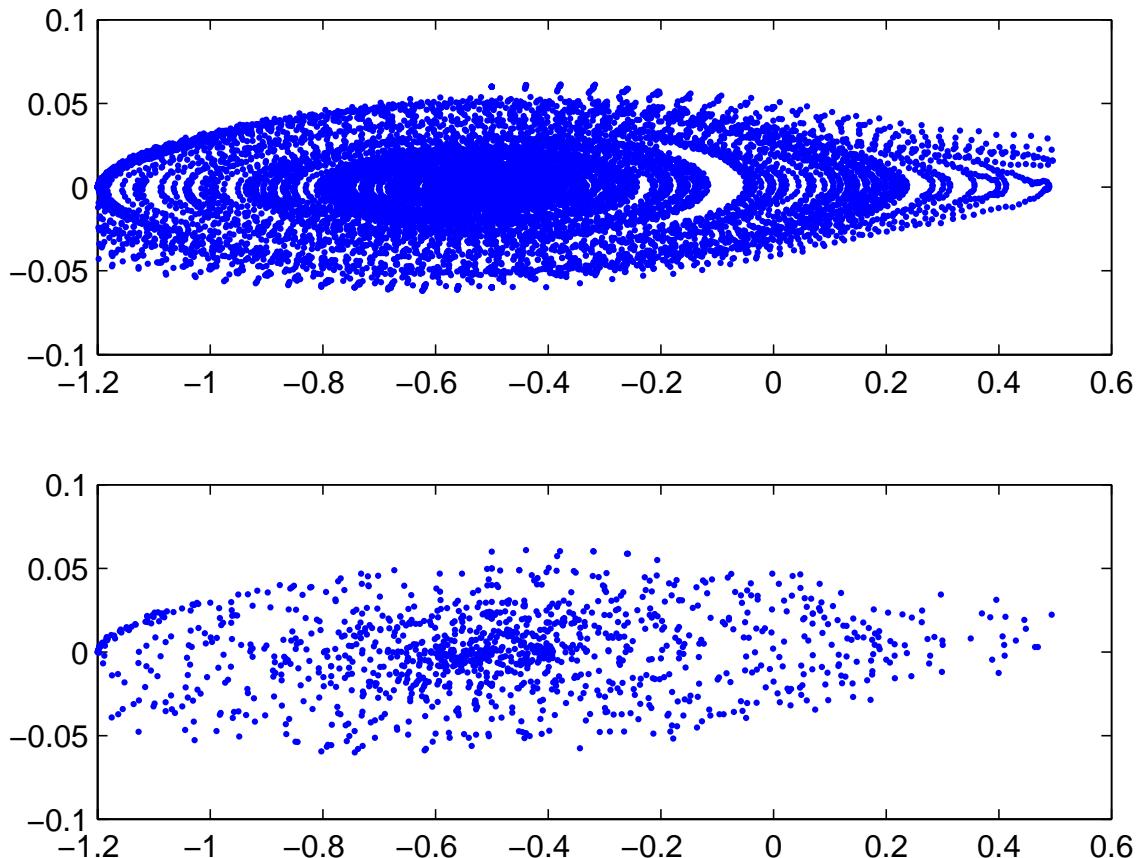
Dilations of  
diffusion operator:  
**local multiscale** basis  
functions

# Algorithmic Issues

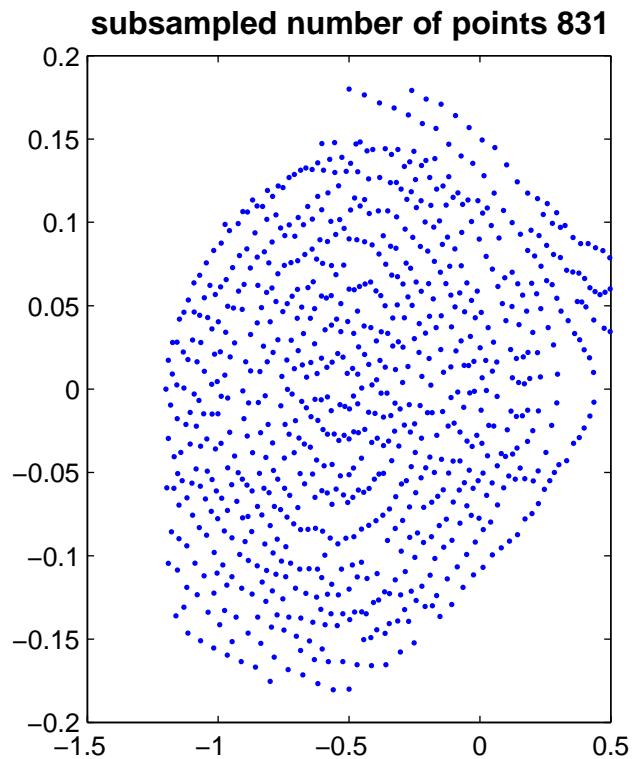
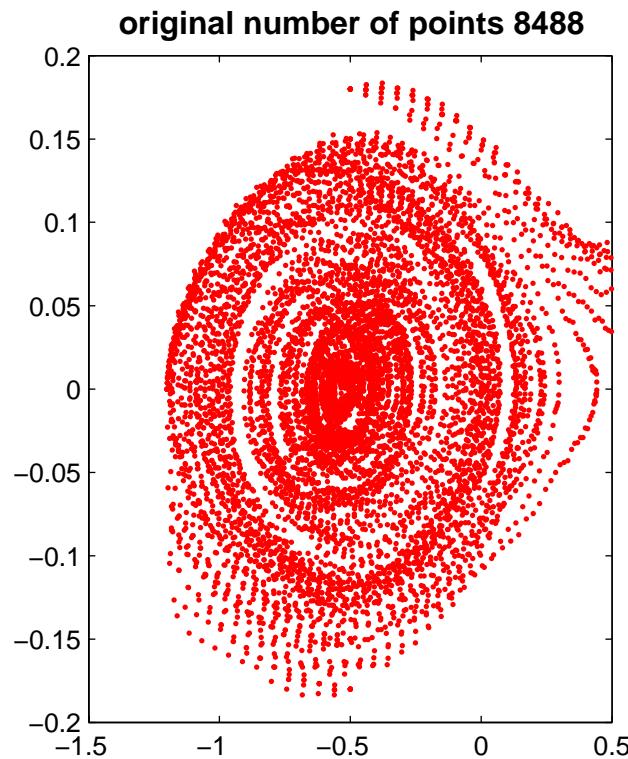
- How to sample the manifold?
  - Random walk or guided exploration
  - How to select a subset of states to construct the bases?
- What sort of graph to build?
  - Similarity metric (action-based or Euclidean)
  - State graph (undirected or directed)
  - State action graph (ditto)
- What graph operator to use?
  - Combinatorial or normalized Laplacian
  - Beltrami normalization
- How to compactly store the eigenvectors?
  - Nystrom and other low-rank approximations
  - Tensor products of basis functions

# Random Sampling from a Continuous Manifold

Mountain  
car domain



# Trajectory Sampling from a Continuous Manifold

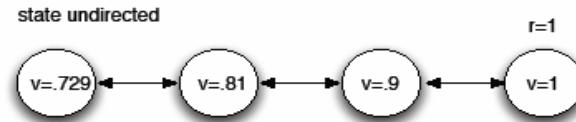


# Type of Graphs

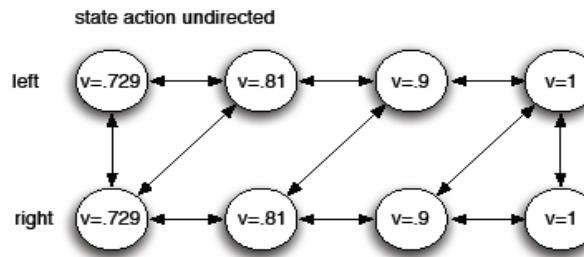
- Graphs on the state space
  - Undirected graph (weighted or unweighted)
  - Directed graph (assume strongly connected)
- Graphs on state action space [Osentoski, 2006]
  - Basis functions directly represent  $\phi(s,a)$
  - Graph grows larger
- Other types of graph
  - Graphs on state controller space (hierarchical RL)
  - Hypergraphs (multivalued relations)

# Types of Graphs

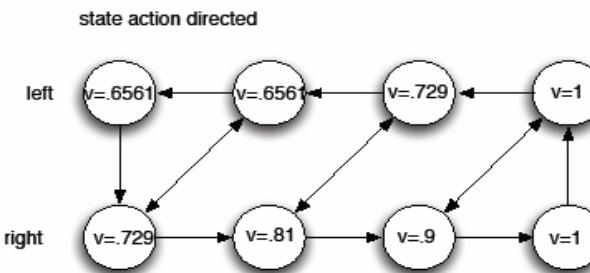
Undirected state:



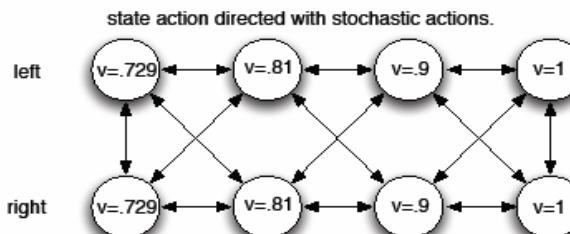
Undirected state action:



Directed state action:



Directed stochastic state action:



# Similarity Metrics

- The graph is constructed using a similarity metric
  - In discrete spaces, connect state  $s$  to  $s'$  if an action led the agent from  $s \rightarrow s'$
  - Action respecting embedding [Bowling, ICML 2005]
- Local distance metrics:
  - Nearest neighbor: connect an edge from  $s$  to  $s'$  if  $s'$  is one of  $k$  nearest neighbors of  $s$
  - Heat kernel: connect  $s$  to  $s'$  if  $|s - s'|^2 < \varepsilon$  with weight  $w(s, s') = e^{-|s - s'|^2/2} < \varepsilon$

# Graph Operator

- Random walk:  $R = D^{-1} W$
- Discrete Laplacian:  $L_d = I - D^{-1}W$
- Combinatorial Laplacian:  $L = D - W$
- Normalized Laplacian:  $\mathcal{L} = D^{-1/2} (D - W) D^{-1/2}$
- Directed Laplacian, Beltrami, ....

# Laplacian on Directed Graphs

(Chung, 2005)

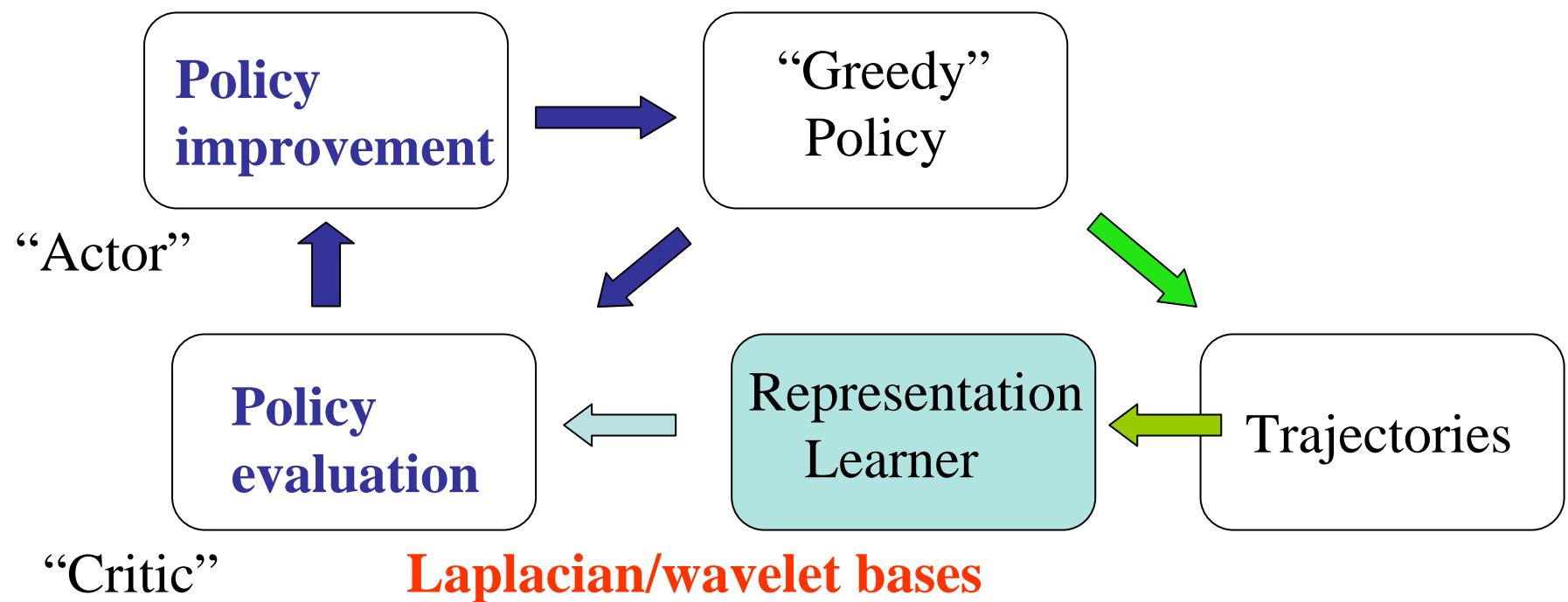
- Given a **strongly connected** directed graph  $G = (V, E, W)$ , the directed Laplacian is defined as
  - $-L = \Phi - (\Phi P + P^T \Phi)/2$  (combinatorial)
  - $\mathcal{L} = I - (\Phi^{1/2} P \Phi^{-1/2} + \Phi^{-1/2} P \Phi^{1/2})/2$
- The diagonal matrix  $\Phi$  is the matrix formed by placing the **Perron vector**  $\phi$  on its main diagonal
- The Perron vector  $\phi$  is the eigenvector associated with the largest eigenvalue of  $P$  (spectral radius)
- The **Perron-Frobenius theorem**: all strongly connected graphs define ergodic irreducible transition matrices, whose largest eigenvector  $\phi$  has all real entries  $> 0$

# Two Specific Algorithms

- **Representation Policy Iteration** (Mahadevan, UAI 2005)
  - Approximate policy iteration on adaptive basis
  - Interleave policy and representation learning
- **Diffusion policy evaluation** (Maggioni and Mahadevan, ICML 2006)
  - New approach to policy evaluation
  - $O(|S|)$  in many problems of interest
  - Unlike incremental methods (e.g, TD), computation is not reward-specific
  - Compute  $(I - \gamma P)^{-1}$  by building basis functions!

# Representation Policy Iteration

(Mahadevan, UAI 2005)



# Least-Squares Policy Iteration

(Lagoudakis and Parr, JMLR 2003)

Random walk generates transitions  $\mathcal{D} = (s_t, a_t, r, s'_t), \dots$

$$\begin{aligned}\tilde{A}^{t+1} &= \tilde{A}^t + \phi(s_t, a_t) (\phi(s_t, a_t) - \gamma \phi(s'_t, \pi(s'_t)))^T \\ \tilde{b}^{t+1} &= \tilde{b}^t + \phi(s_t, a_t) r_t\end{aligned}$$

Solve the equation:  $\tilde{A} w^\pi = \tilde{b}$

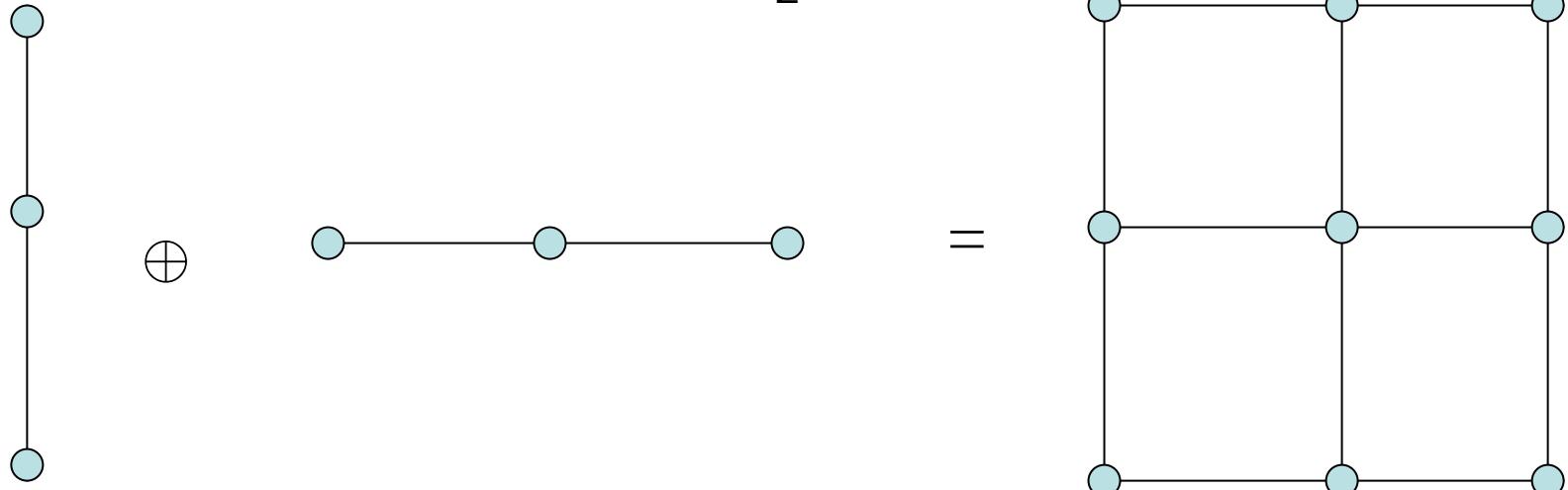
$$Q^\pi(x, a) \approx \sum_{i=1}^k \phi(s, a) w_i^\pi$$

# Scaling Fourier and Wavelet Bases

- Factored MDPs generate **product (tensor) spaces**
  - It is possible to represent spectral bases compactly for large factored MDPs
  - Basis functions can be represented in space *independent* of the size of the state space
  - Fourier analysis on groups: compact representations
- Continuous spaces can be handled by sampling the underlying manifold and constructing a graph
  - **Nystrom interpolation method** for extension of eigenfunctions
  - Low-rank approximations of diffusion matrices

# Kronecker Sum Graphs

- The **Kronecker sum** of two graphs  $G = G_1 \oplus G_2$  is the graph with vertex set  $V = V_1 \times V_2$  and adjacency matrix  $A = A_1 \otimes I_2 + I_2 \otimes A_1$ 
  - Alternative definition: The Kronecker sum graph  $G$  has an edge between vertices  $(u,v)$  and  $(u',v')$  if and only if  $(u,u') \in E_1$  and  $v=v'$  or  $(u=u')$  and  $(v,v') \in E_2$



# Spectral Theory of Tensor Products

- Let  $A_{r \times r}$  and  $B_{s \times s}$  be two matrices of full rank
- Let  $(\lambda_i, u_i)$  and  $(\mu_j, v_j)$  be the  $i^{\text{th}}$  eigenvalue and eigenvector of graph A and B, respectively
- Spectra of tensor sum and products:
  - $(A \otimes B) (u_i \otimes v_j) = \lambda_i \mu_j (u_i \otimes v_j)$
  - $(A \otimes I_s + I_r \otimes B) (u_i \otimes v_j) = (\lambda_i + \mu_j) (u_i \otimes v_j)$
- This result is based on the following identity
  - $(A C) \otimes (B D) = (A \otimes B) (C \otimes D)$  (if AC and BD are well-defined)

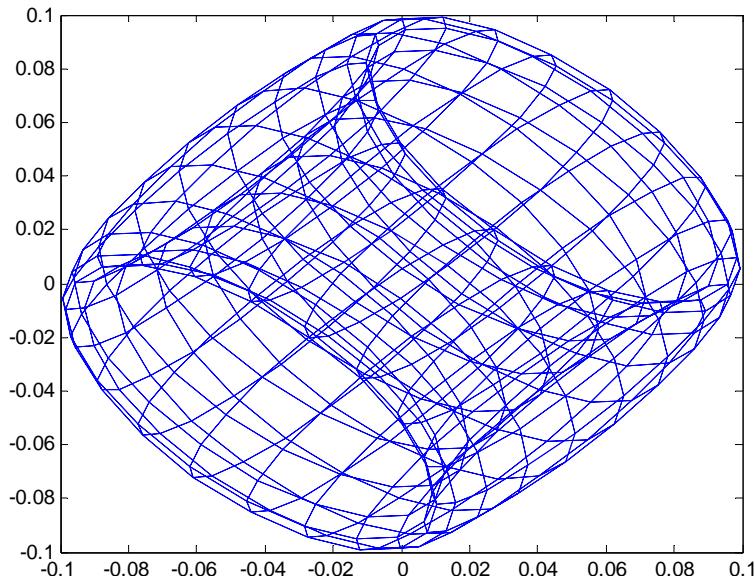
# Laplacian of Kronecker Graphs

- If  $L_1, L_2$  be the combinatorial Laplacians of graphs  $G_1, G_2$ , then the spectral structure of the combinatorial Laplacian of the Kronecker sum of these graphs  $G = G_1 \oplus G_2$  is specified as

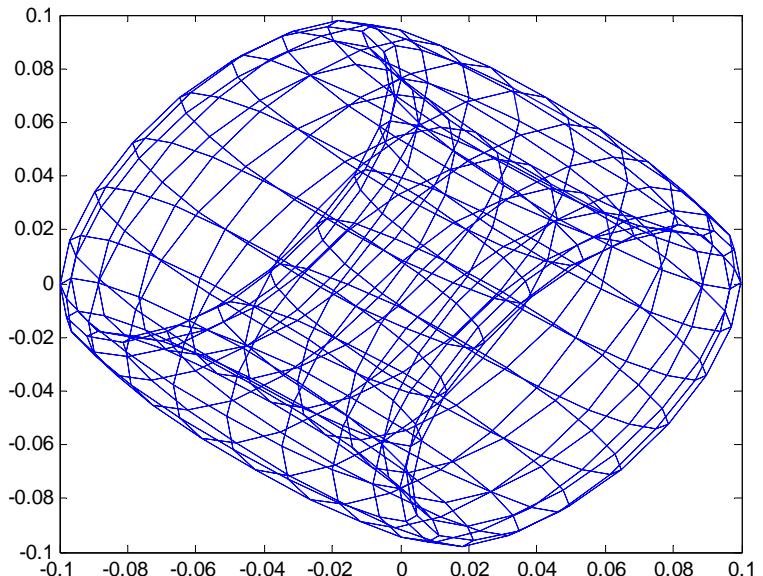
$$\sigma(L), \chi(L) = \{\lambda_i + \mu_j, l_i \otimes k_j\}$$

- where  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue of  $L(G_1)$  with associated eigenvector  $l_i$  and  $\mu_j$  is the  $j^{\text{th}}$  eigenvalue of  $L(G_2)$  with associated eigenvector  $k_j$ .

# Embedding of Structured Spaces



Torus



2<sup>nd</sup> and 3<sup>rd</sup> eigenvectors  
of combinatorial Laplacian

# RPI in Continuous State Spaces

(Mahadevan, Maggioni, Ferguson, Osentoski, AAAI 2006)

- RPI in continuous state spaces
  - The Nystrom extension interpolates eigenfunctions from sample points to new points
- Many practical issues are involved
  - How many samples to use to build the graph?
  - Local distance metric: Gaussian distance, k-NN
  - Graph operator: Normalized Laplacian, Combinatorial Laplacian, Random Walk, ...
  - Type of graph: Undirected, directed, state-action graph

# The Nystrom method

(Williams and Seeger, NIPS 2001)

- The Nystrom approximation was developed in the context of solving integral equations

$$\int_D K(t,s) \Phi(s) ds = \lambda \Phi(t), t \in D$$

- A quadrature approximation of the integral:

$$\int_D K(t,s) \Phi(s) ds = \sum_j w_j k(x,s) \phi(s_j)$$

leads to the following equation

$$\sum_j w_j k(x,s) \phi(s_j) = \lambda \phi(x)$$

- which rewritten gives the Nystrom extension

$$\phi_m(x) = 1/\lambda_m \sum_j w_j k(x,s) \phi_m(s_j)$$

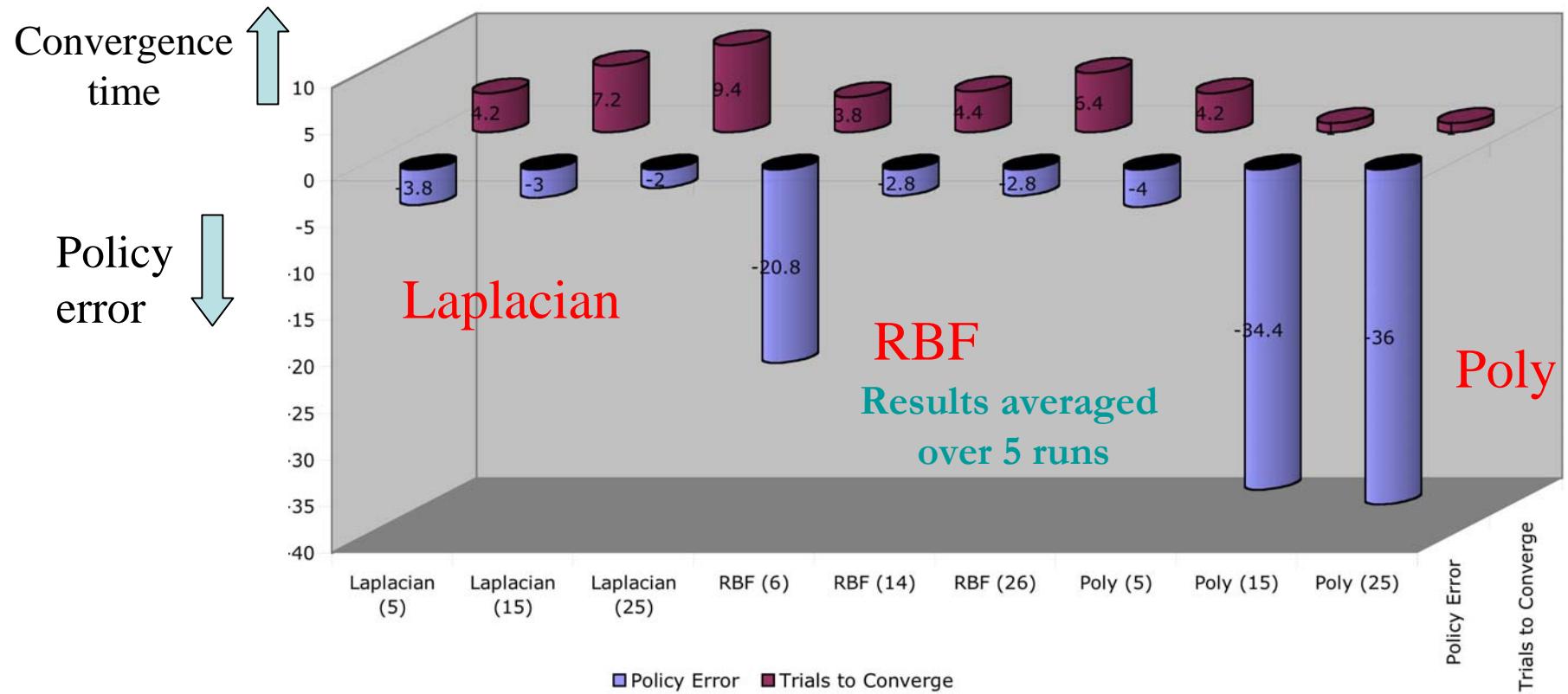
# Outline

- Part One
  - History and Motivation (9:00-9:15)
  - Overview of framework (9:15-9:30)
  - Technical Background (9:30-10:30)
  - Questions: (10:30-10:45)
- Part Two
  - Algorithms and implementation (11:15-11:45)
  - Experimental Results (11:45-12:30)
  - Discussion and Future Work (12:30-12:45)
  - Questions (12:45-1:00)

# Experimental Testbeds

- Discrete MDPs:
  - One dimensional chains [Koller and Parr, UAI 2001]
  - Two dimensional “room” environments [Mannor, McGovern, Simsek et al]
  - Factored MDPs (Sallans and Hinton, JMLR 2003)
- Continuous MDPs:
  - Inverted pendulum and mountain car

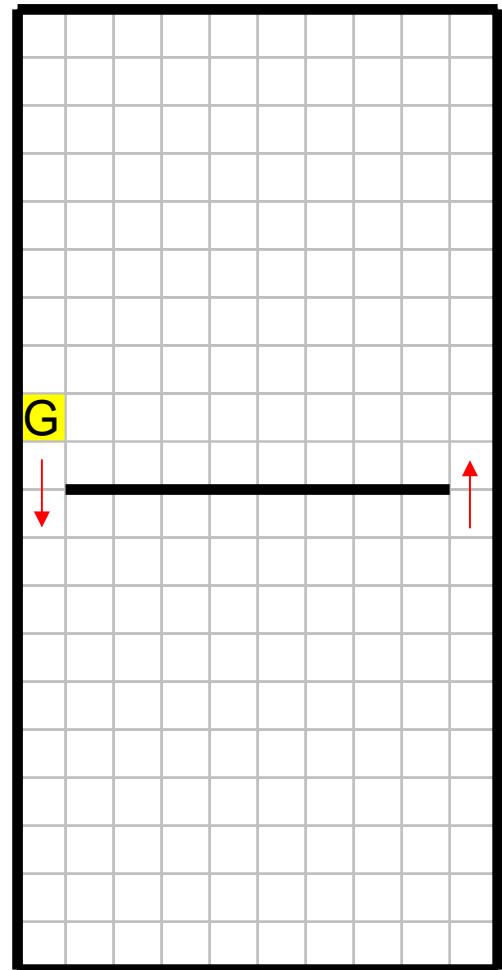
# Learned vs. Handcoded Representations: Chain MDP



# Directed Two-Room Environment

(Johns, 2006)

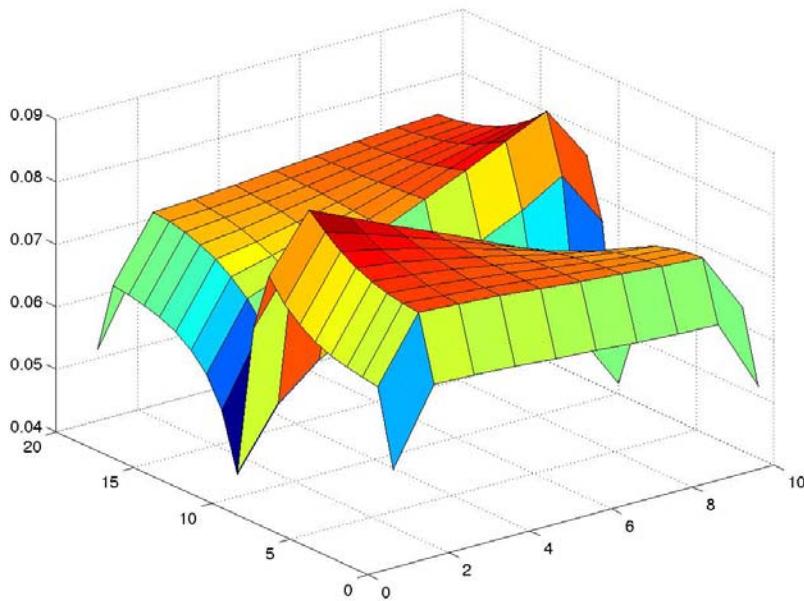
- This domain was used to compare the bases functions from the undirected Laplacian vs. the directed Laplacian
  - Two 10x10 rooms with two directed edges (all other edges are undirected)
  - Four stochastic actions, zero reward unless in goal state (+100)
  - Discount factor of 0.9



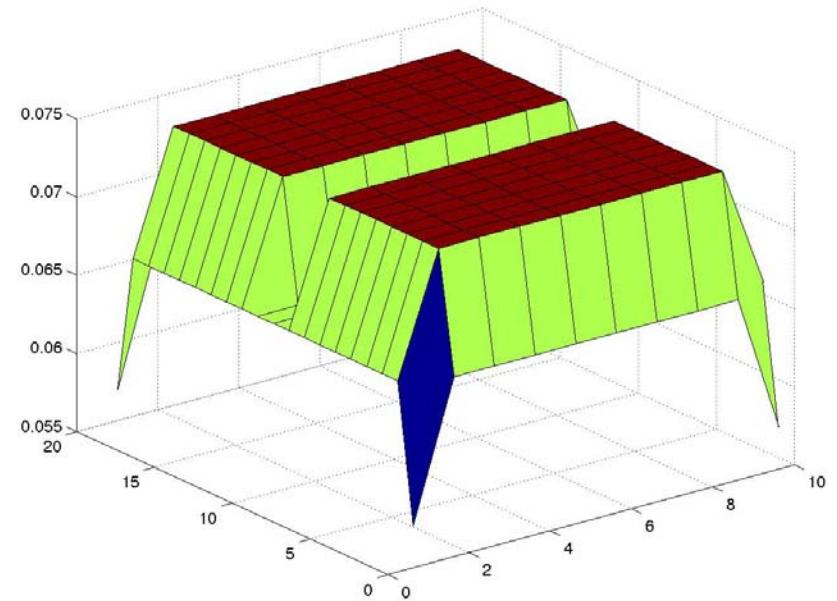
# Directed vs. Undirected Laplacian

- The first eigenvector of the normalized Laplacian shows the difference directionality makes on the steady-state distribution

Directed



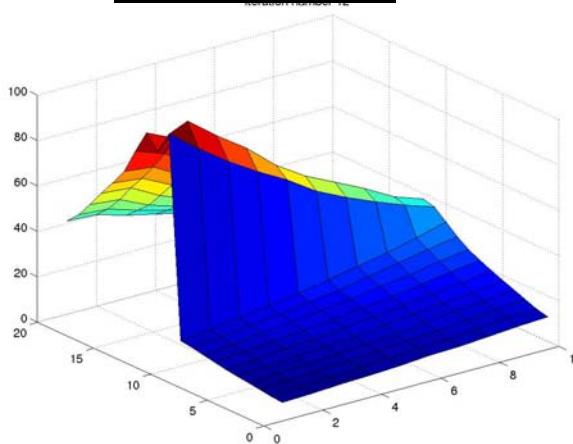
Undirected



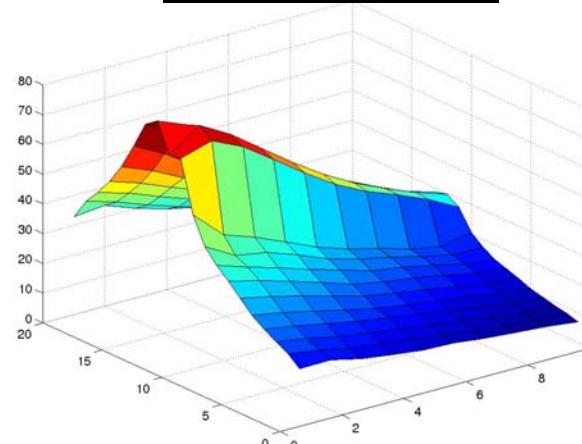
# Results: Directed vs. Unidirected Laplacian (Johns, 2006)

- The undirected Laplacian results in a poorer approximation because it ignores directionality

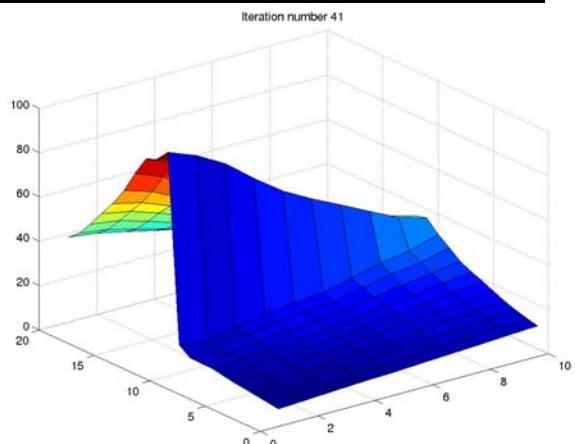
Exact VF



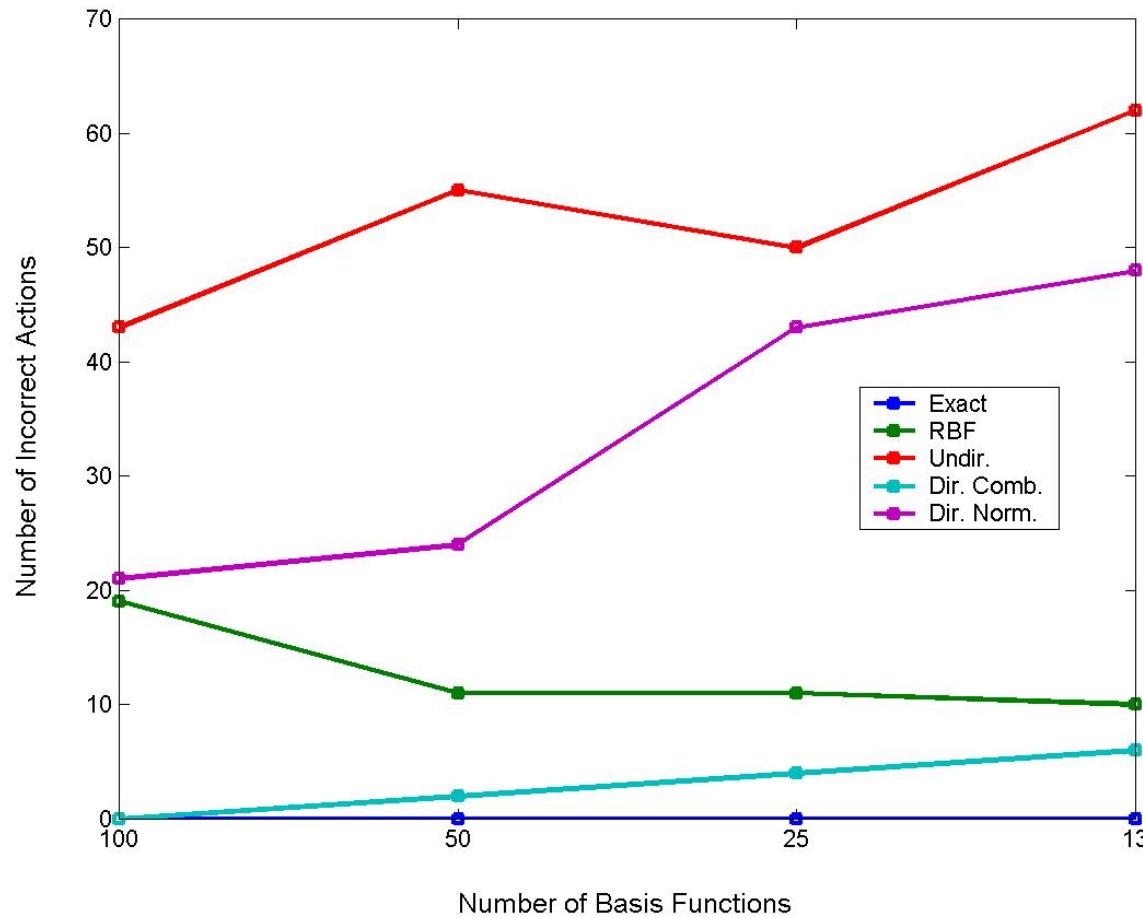
Undirected



Dir. Combinatorial



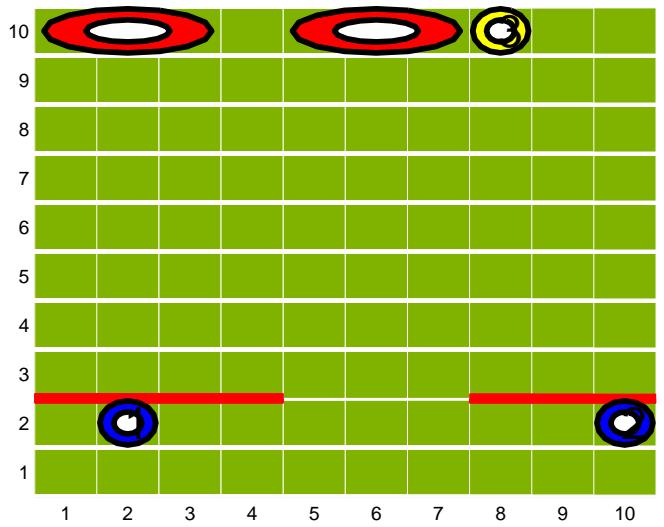
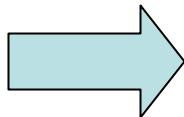
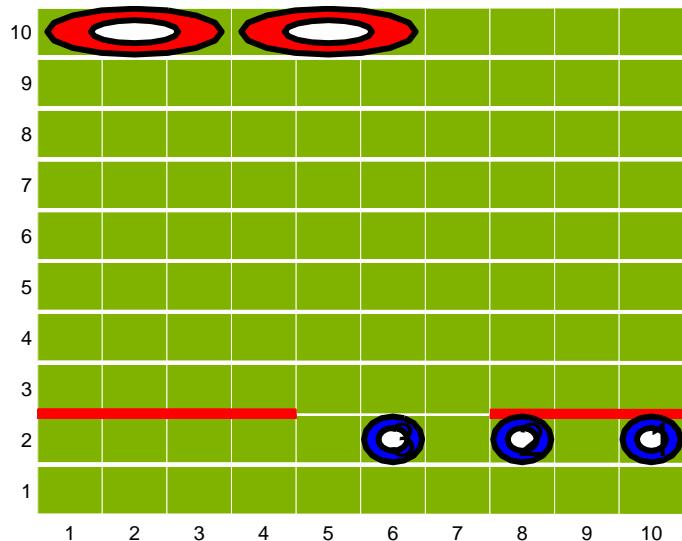
# Comparison of Undirected vs. Directed Laplacians



# Blockers Domain

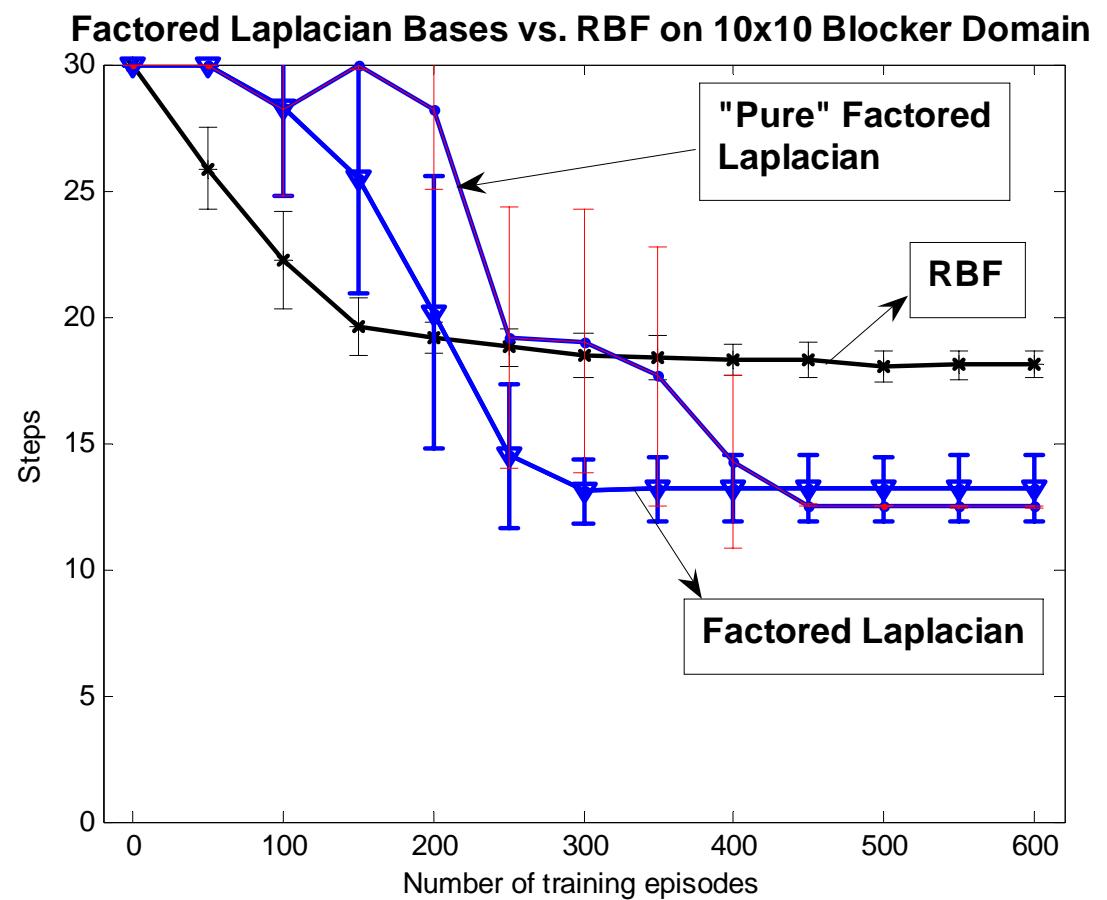
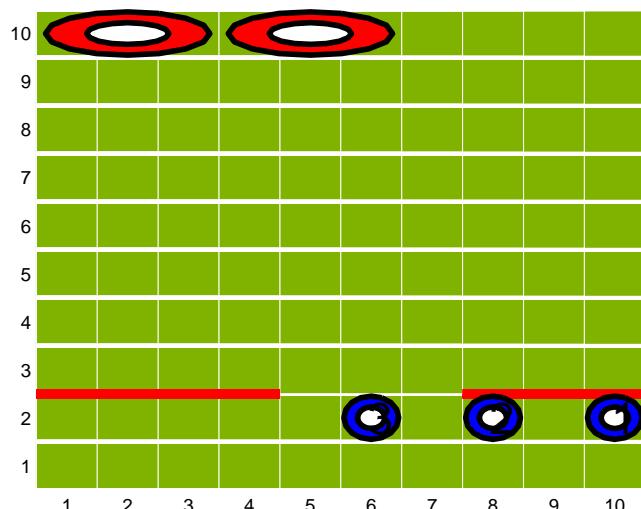
(Sallans and Hinton, JMLR 2003)

Large state space of  $> 10^6$  states

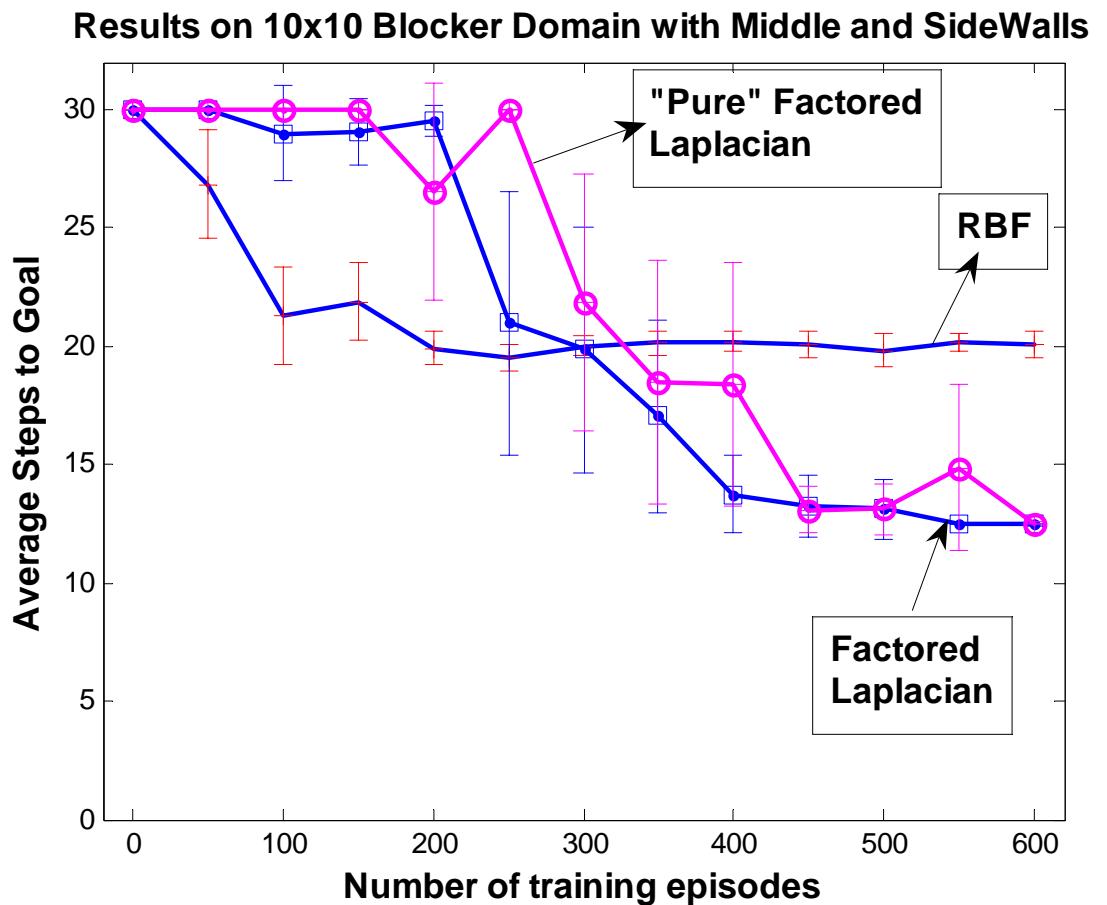
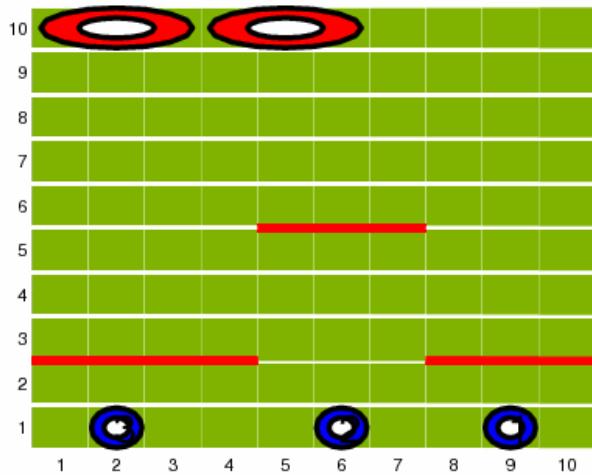


# Large Factored MDP: Blockers Domain

Topologically, this space is the tensor product of three “irregular” cylinders

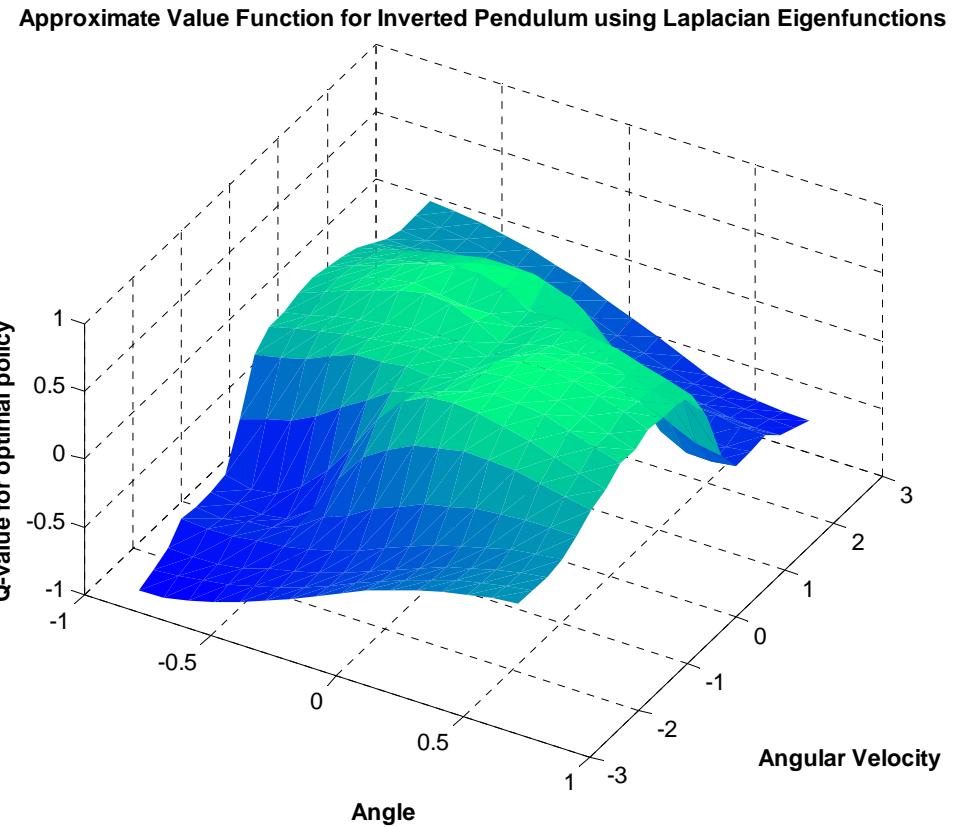
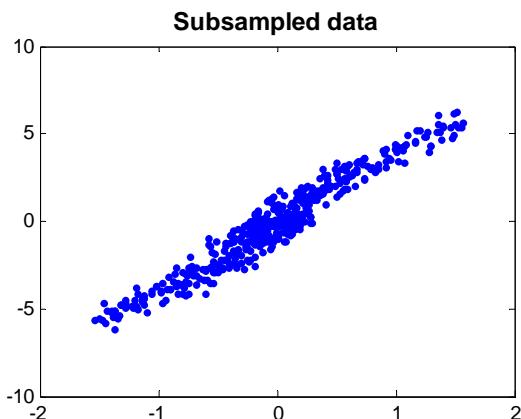
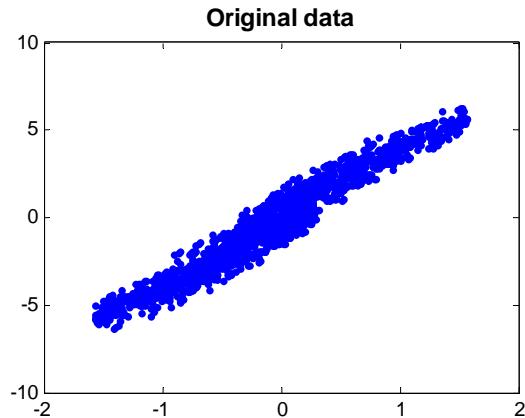


# Irregular Blockers Domain

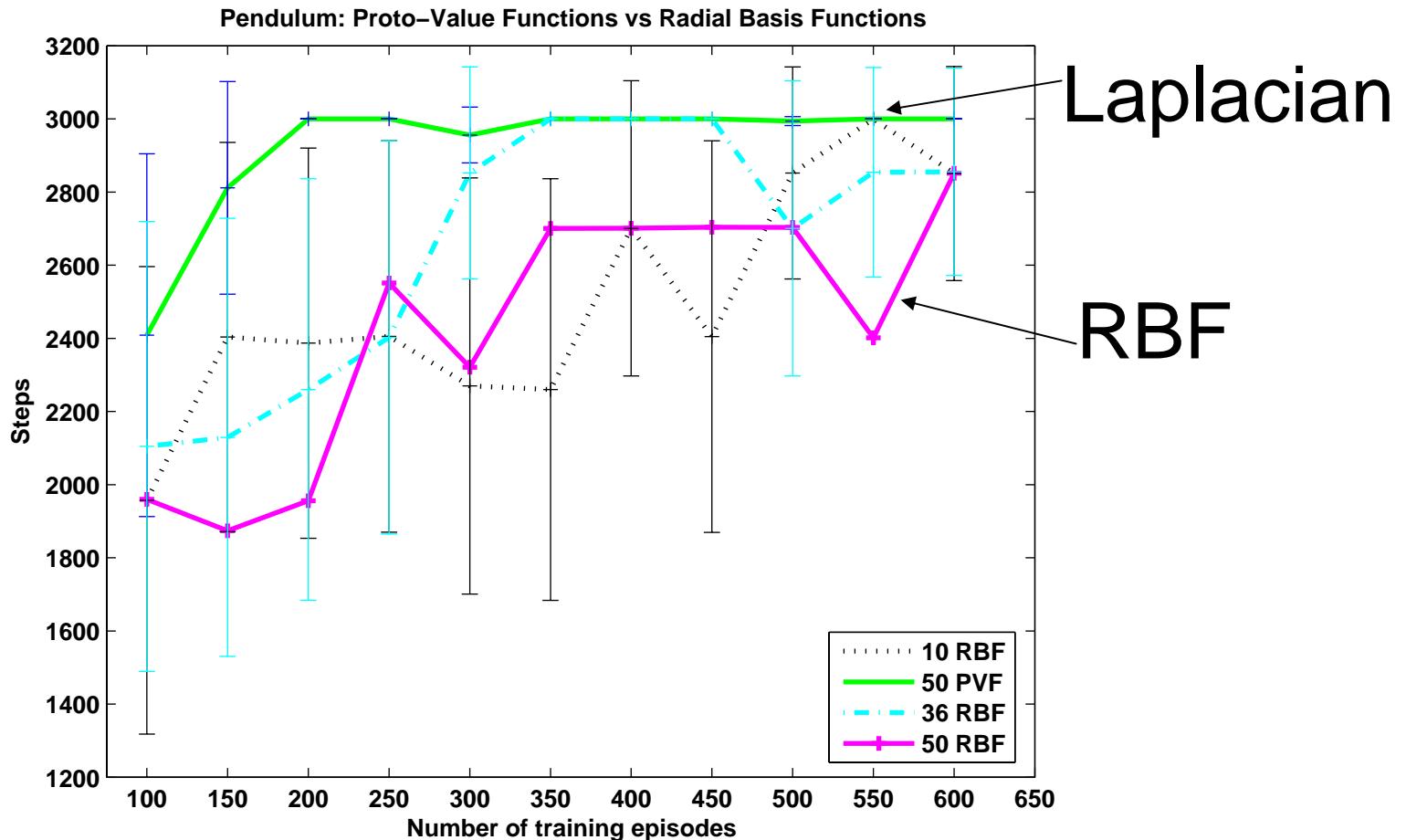


# RPI on Inverted Pendulum

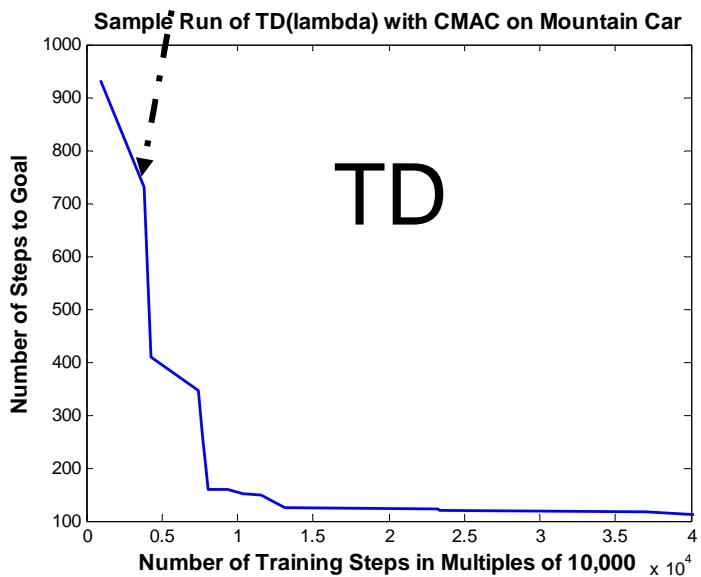
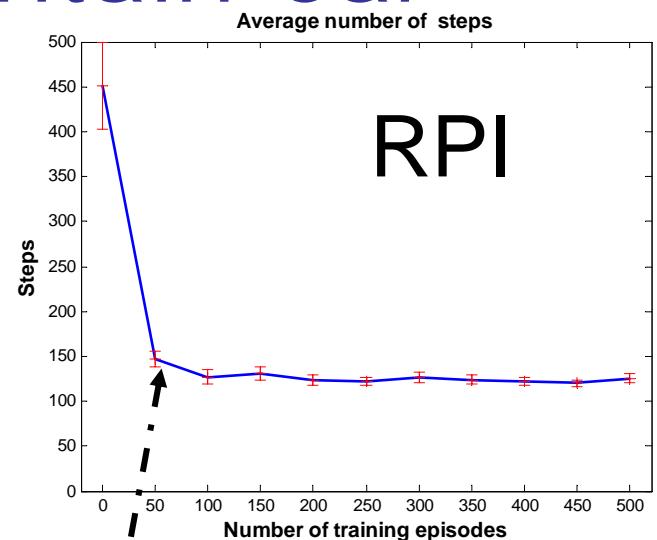
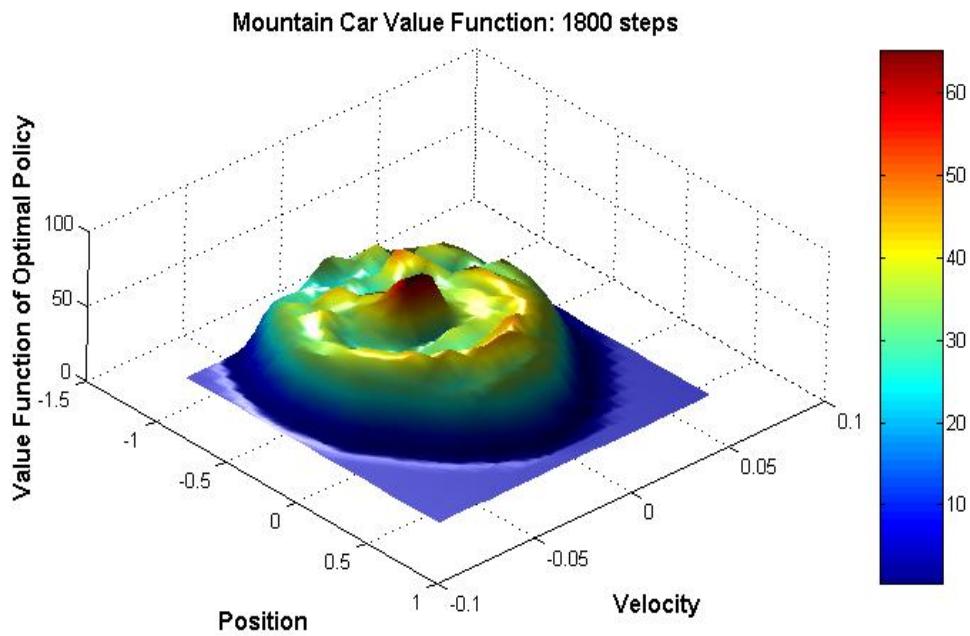
(Mahadevan, Maggioni, Ferguson, Osentoski, AAAI 2006)



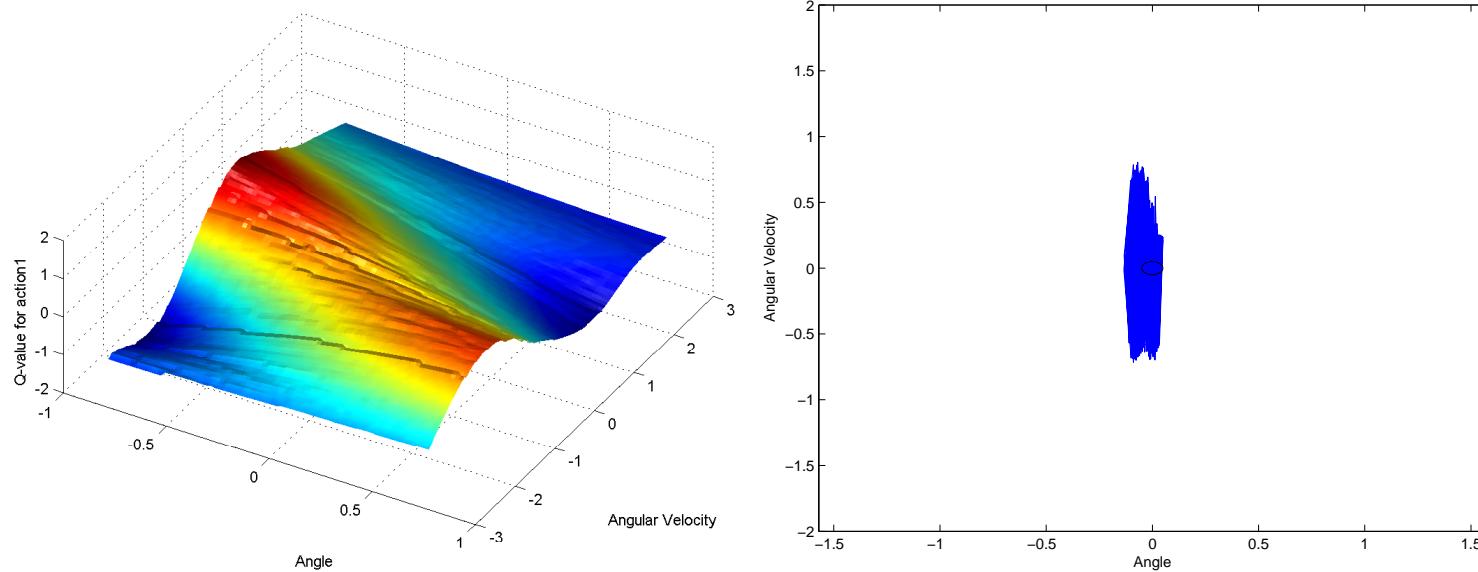
# RPI on Inverted Pendulum



# RPI with Laplacian Bases vs. TD with CMAC on Mountain Car

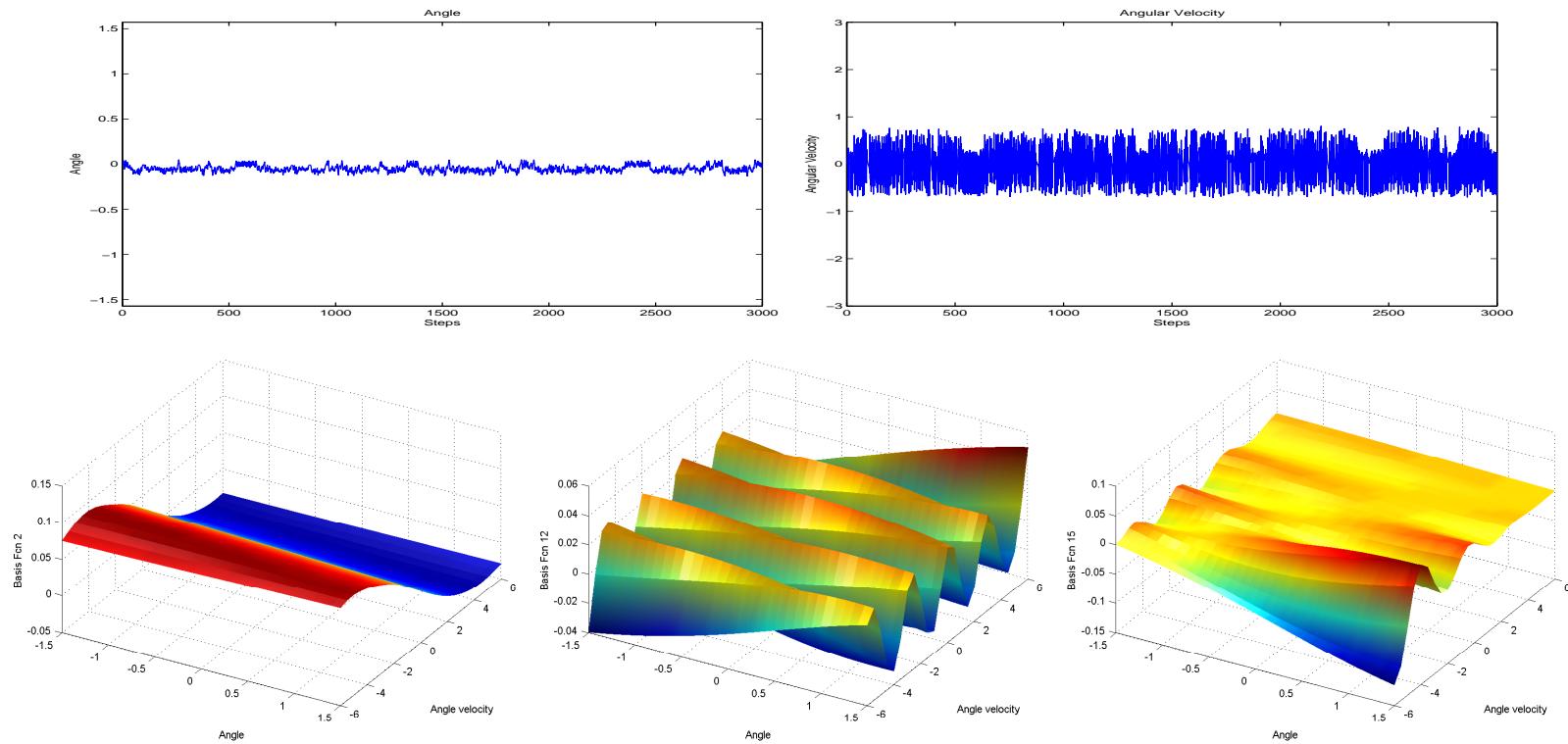


## Inverted Pendulum



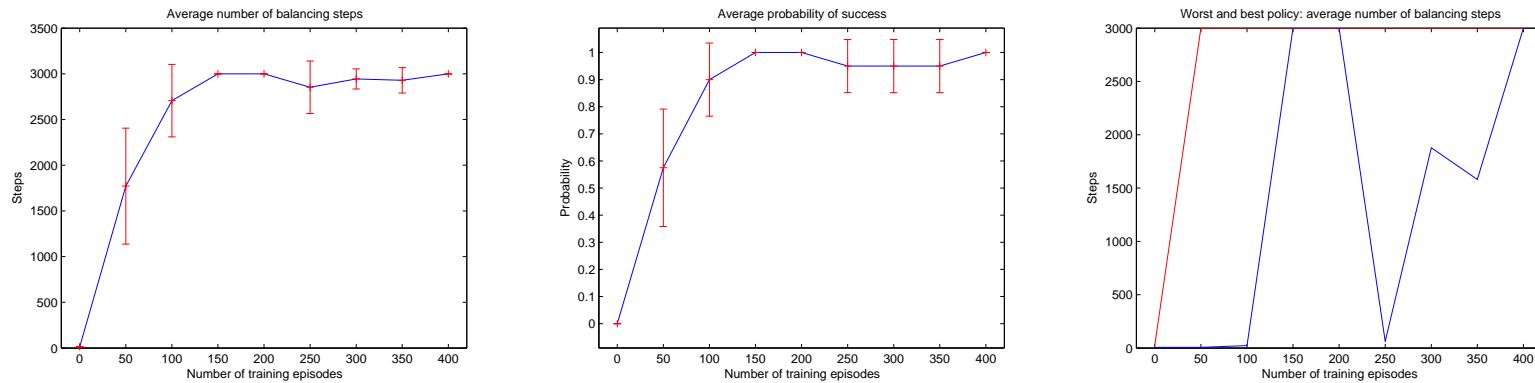
Left:  $Q$ -value function for the action “left”, reconstructed from its representation of the diffusion wavelet basis. Right: trajectory of the pendulum in phase space according to the policy learnt.

## Inverted Pendulum (cont'd)



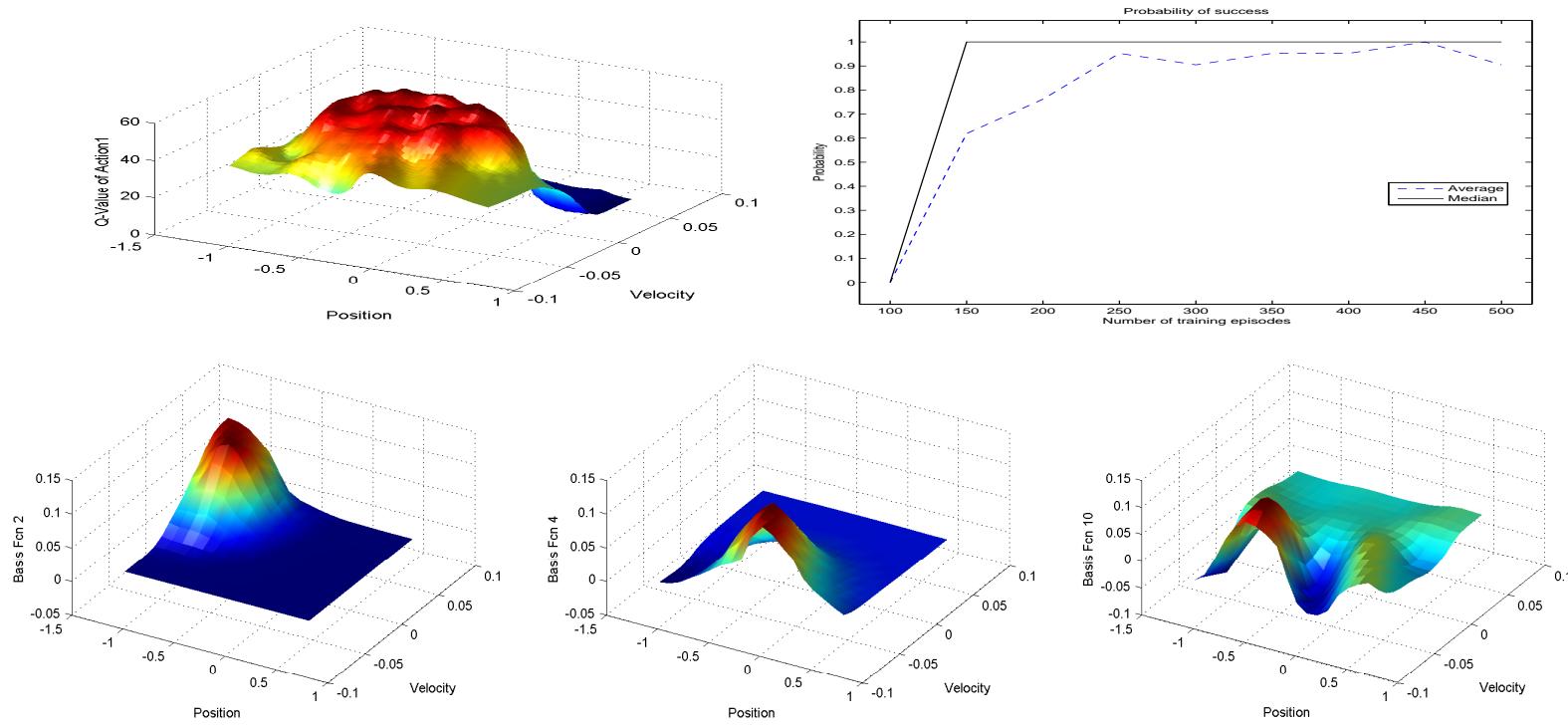
Top row: trajectory of angle and angle velocity variables. Bottom row: some diffusion wavelets used as basis functions for representation during the learning phase.

## Inverted Pendulum (cont'd)



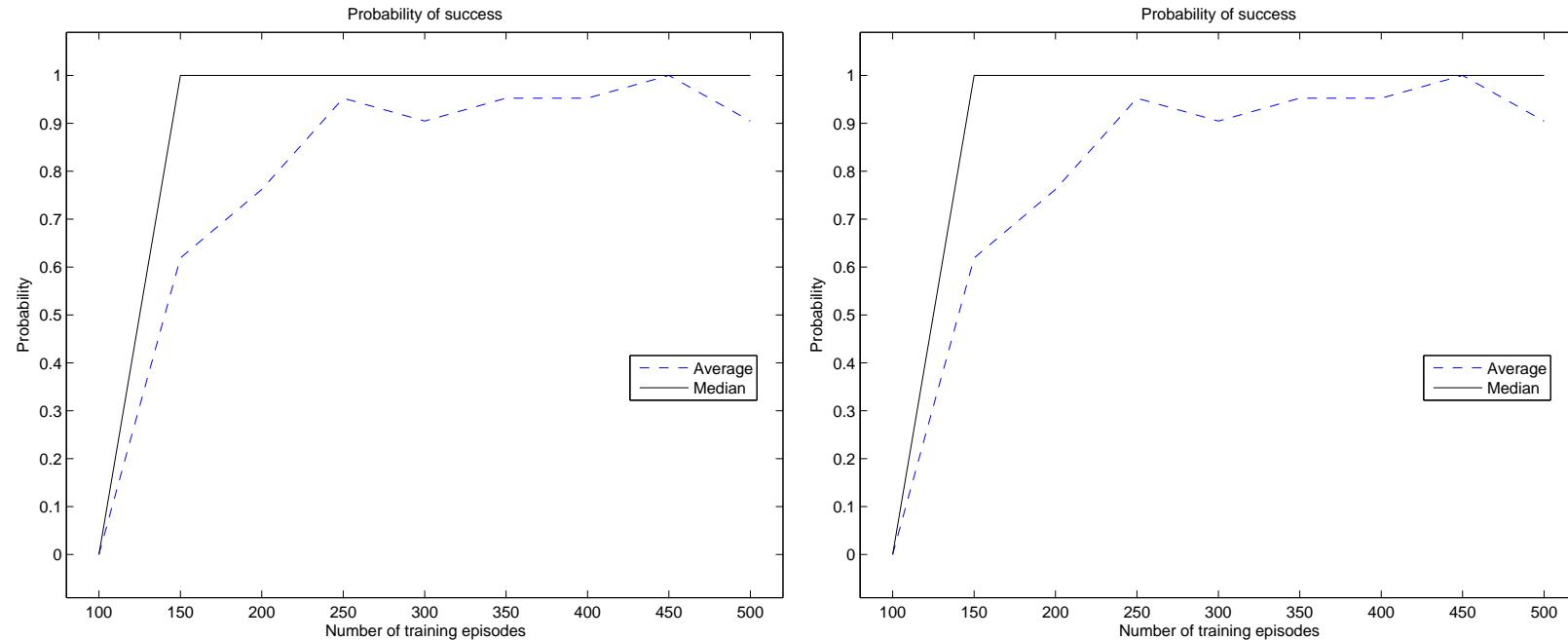
Measures of performance based on 20 experiments, as a function of number of training runs (each of which of length at most 100). From left to right: average number of successful steps of inverted pendulum balancing, average probability of succeeding in balancing for at least 3000 steps, and worst and best number of balancing steps. Each simulation was stopped and considered successful after 3000 steps, which biases the first and third graphs downwards.

# Mountain Car



Top left:  $Q$ -value function for the action "left", reconstructed from its representation of the diffusion wavelet basis. Top right: trajectory of the mountain car in phase space according to the policy learnt (107 steps). Bottom row: some diffusion wavelets used as basis functions for representation during the learning phase.

## Mountain Car (cont'd)



Measures of performance based on 23 experiments, as a function of number of training runs (each of which of length at most 100). Left: average and median number of successful steps for reaching the goal; right: average and median probability of succeeding in reaching the goal in less than 800 steps. The best policy actually finds a path in 103 steps.

## Multiscale inversion

The multiscale construction enables a direct solution of Bellman's equation. The algorithm consists of two parts:

- (i) a *pre-computation* step, that depends on the structure of the state space and on the policy, and yields the multiscale analysis described above.
- (ii) an *inversion* step which uses the multiscale structure built in the pre-computation step to efficiently compute the solution of Bellman's equations for a given reward function.

## Multiscale inversion (cont'd)

The starting point are the identities

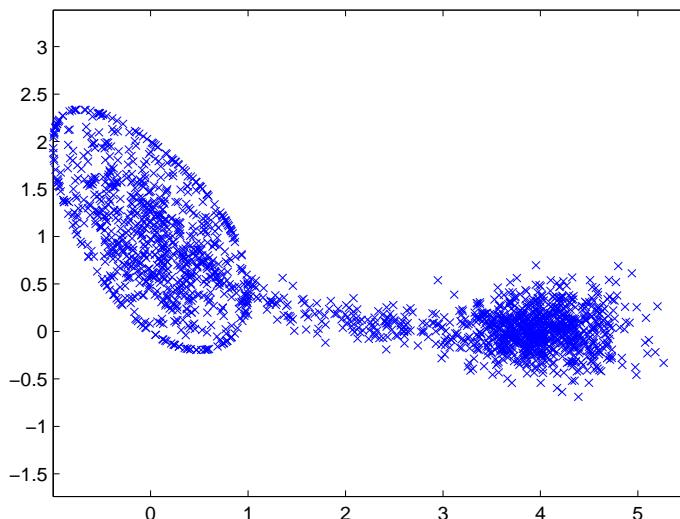
$$V^\pi = (I - \gamma P^\pi)^{-1} R = \sum_{k \geq 0} (\gamma \Pi^{-\frac{1}{2}} T^\pi \Pi^{\frac{1}{2}})^k R = \prod_{k \geq 0} (I + \gamma^{2^k} \Pi^{-\frac{1}{2}} (T^\pi)^{2^k} \Pi^{\frac{1}{2}}) R,$$

where  $P^\pi = \Pi^{-\frac{1}{2}} T^\pi \Pi^{\frac{1}{2}}$ ,  $\Pi$  is the matrix whose diagonal is the asymptotic distribution of  $P$ , and  $R$  is the reward vector. The formulas hold for  $\gamma \leq 1$  and  $R$  has no component in the kernel of  $(I - \gamma P^\pi)$ .

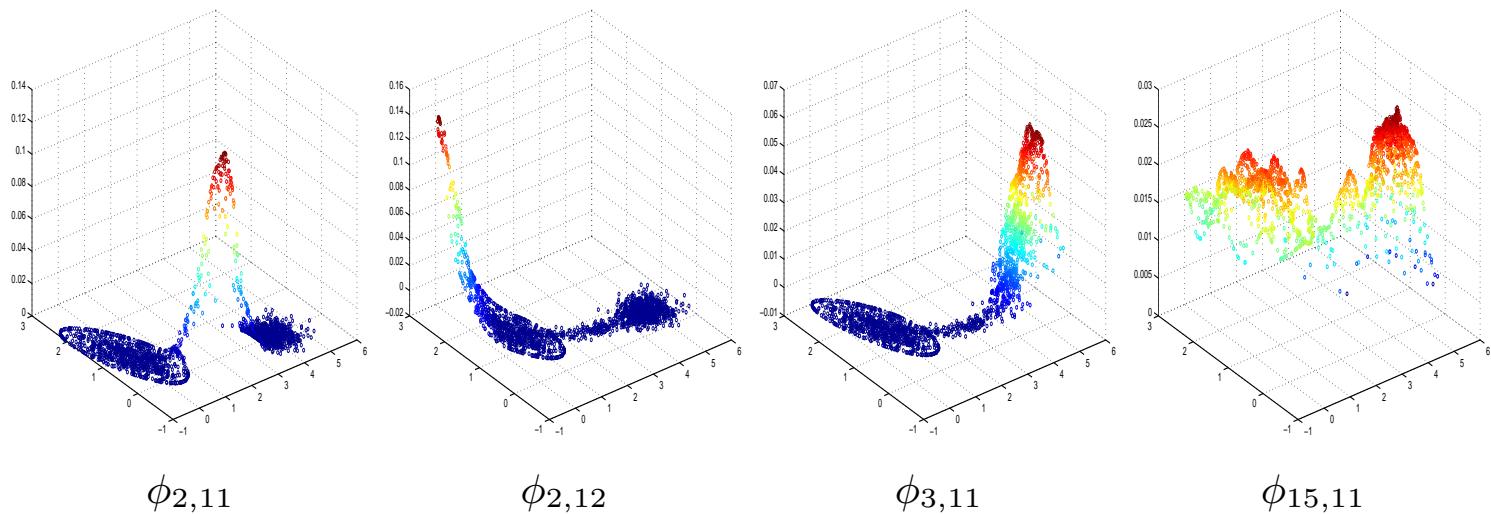
We have compressed in a multiscale fashion the (quasi-)dyadic powers of the operator  $T^\pi$ .

In many cases of interest, both the construction of the multiscale structure and the inversion take only  $\mathcal{O}(|S|)$  operations!

## Example: Two-room environment

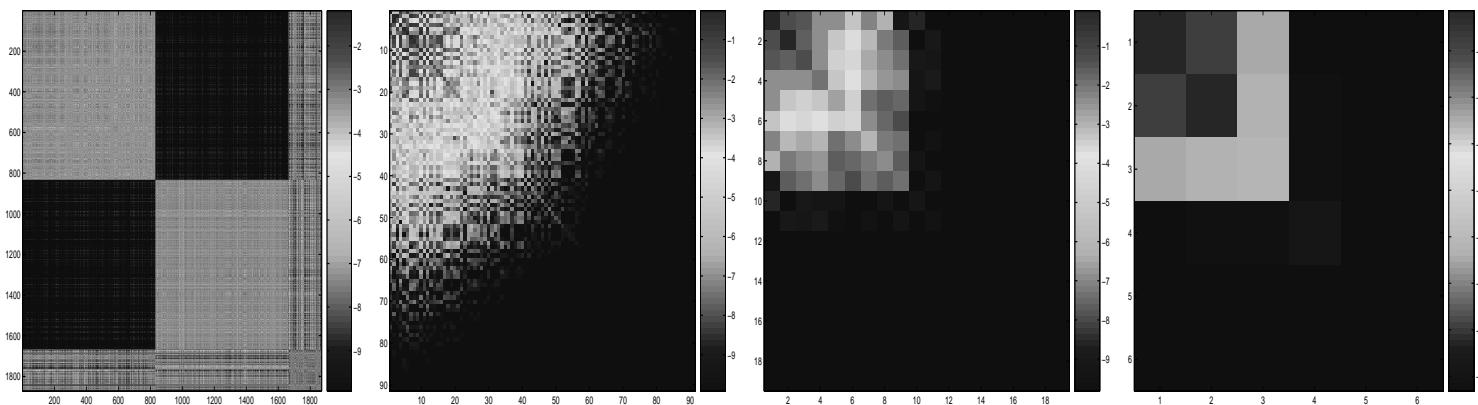


## Example: Two-room environment, II



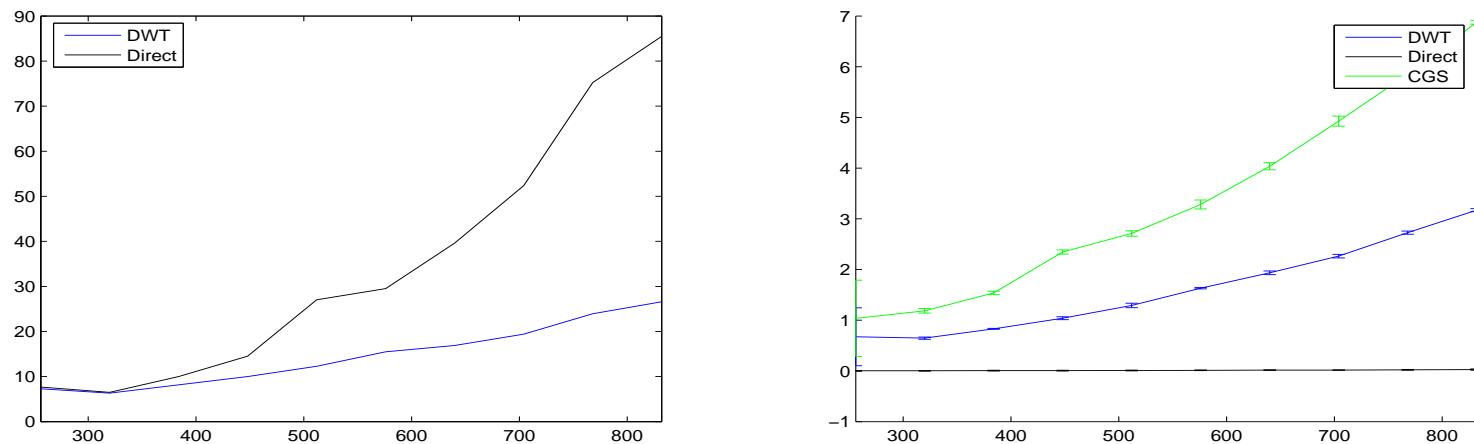
Four diffusion scaling functions built on the set, at increasing scale. Note the localization at the finer scales, and the global support at coarser scales.

### Example: Two-room environment, III

 $T^1$  $T^2$  $T^{2^5}$  $T^{2^7}$ 

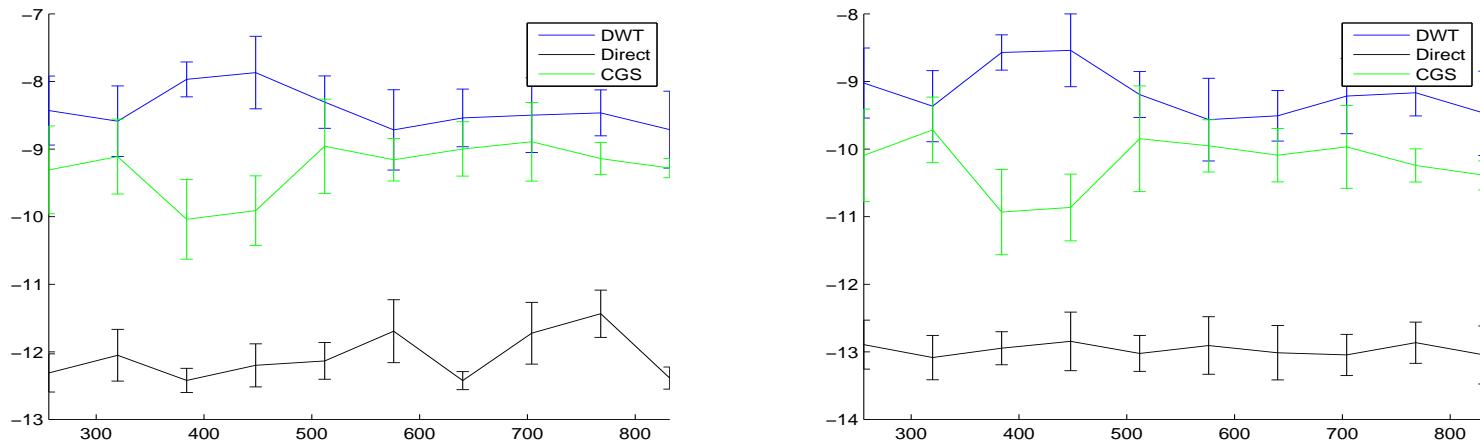
Compression of the powers of the symmetrized random walk  $T$  in a continuous two-room environment.  $T_0$  is sorted to show the two-room and corridor structures.  $T_6$  is very small, and essentially represents only the transition between two states (the two rooms).

## Comparison with standard Direct and Iterative techniques



Solving a Bellman equation on a random walk in the two-room environment, as a function of the number of states explored ( $x$ -axis): DWT inversion, iterative Conjugate Gradient Squared method (Matlab implementation) and direct inversion. Left: pre-processing time, comparing computation of the full inverse and construction diffusion wavelet tree. Right: computation time of applying the inversion scheme.

## Comparison with standard Direct and Iterative techniques, II



Precision, defined as  $\log_{10}$  of the Bellman residual error  $\| (I - \gamma P^\pi) \tilde{V}^\pi - R \|_p$ , where  $\tilde{V}^\pi$  is the computed solution, achieved by the different methods. The precision requested was  $1e-10$ . We show the results for  $p = 2$  (left) and  $p = \infty$  (right).

# Outline

- **Part One**
  - History and Motivation (9:00-9:15)
  - Overview of framework (9:15-9:30)
  - Technical Background (9:30-10:30)
  - Questions: (10:30-10:45)
- **Part Two**
  - Algorithms and implementation (11:15-11:45)
  - Experimental Results (11:45-12:30)
  - Discussion and Future Work (12:30-12:45)
  - Questions (12:45-1:00)

# Challenges and Future Directions

- Computational tractability
  - Fourier and wavelet bases for high-dimensional continuous control tasks (e.g. humanoid robots)
- Convergence and theoretical analysis
  - Can these bases be shown to be “optimal” in some interesting sense?
- Application of this approach to related problems
  - POMDPs: value function is highly compressible!
  - PSRs: low-rank approximation of dynamical systems
- The approach is general, and provides a way to do multiscale data analysis on a graph

# Factored and Relational MDPs

- Much work on factored and relational MDPs and RL
  - [Koller and Parr, UAI 2000; Guestrin et al, IJCAI 2003; JAIR 2003 ]
  - [Fern, Yoon, and Givans, NIPS 2003]
  - ICML 2004 workshop on relational RL
- How to construct Fourier and wavelet bases over relational representations?
  - Symmetries and group automorphisms

# Exploiting Symmetries to Reduce Basis Size

- A **graph automorphism**  $h$  is a mapping from the vertex set  $V \rightarrow V$ , such that
  - $w(u, v) > 0 \leftrightarrow w(h(u), h(v)) > 0$
- The automorphisms of a graph can generate compact bases
- Let  $P$  be a permutation matrix such that

$$A P = P A$$

- If  $x$  is an eigenvalue of  $A$ , then so is  $Px$

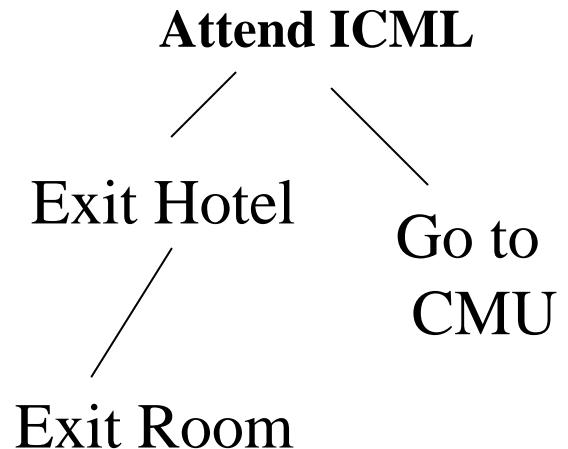
$$A P x = P A x = \lambda P x$$

# Modeling Temporally Extended Actions

(Barto and Mahadevan, Discrete-Event Systems, 2003)

- Semi-Markov decision process
  - $S$ : set of states
  - $A$ : set of *activities* (or behaviors)
  - $P: S \times A \times N \times S \rightarrow (0,1)$  multi-step transition probability
  - $R: S \times A \times N \rightarrow R$  expected reward over duration of activity

[Kaelbling, ICML 1993]  
[Parr and Russell, NIPS 1998]  
[Sutton, Precup, and Singh, AIJ 1999]  
[Dietterich, JAIR 2000]



# How to Discover Temporal Abstractions?

- Much recent work
  - Find **bottlenecks** and **symmetries** in state spaces
  - [McGovern, U.Mass PhD, 2002; Balaraman, U.Mass, PhD 2004]
  - Rank state variables by **rate of change**
  - [Hengst, ICML 2002]
  - **Graph-based** approaches
  - [Menache et al, ECML 2002; Simsek, Wolfe, and Barto, ICML 2005]
- Lacks formal framework that generalizes to arbitrary (continuous or discrete) spaces
- Does not yield compact representations of temporally extended actions

# Applications to Hierarchical RL

- Learning temporally extended actions
  - Laplacian eigenfunctions can be used to partition the graph (Cheeger constant)
  - Diffusion wavelets can be used to learn multiscale option models
- Task transfer
  - Laplacian or diffusion bases are reward independent
  - Proto-transfer: map representations from one task to another [Ferguson and Mahadevan, ICML Workshop on Transfer Learning, 2006]

## Current & Future work

- Relate approximation rates with guarantees on convergence of the algorithm;
- Construct bases even better adapted to approximation and learning;
- Study sensitivity of basis construction with respect to sampling and other deformations;
- Explore other ways for performing multiscale analysis, tuned to the policy iteration algorithm and its target optimum;
- Transferring learning, by mapping the manifolds, and transport basis functions from one problem to the other;

Material (Matlab code, tutorial talks) available at [www.math.yale.edu/~mmm82](http://www.math.yale.edu/~mmm82).

Thank you!

# Outline

- **Part One**
  - History and Motivation (9:00-9:15)
  - Overview of framework (9:15-9:30)
  - Technical Background (9:30-10:30)
  - Questions: (10:30-10:45)
- **Part Two**
  - Algorithms and implementation (11:15-11:45)
  - Experimental Results (11:45-12:30)
  - Discussion and Future Work (12:30-12:45)
  - Questions (12:45-1:00)

# Further Reading

- Fourier bases (Laplacian eigenfunctions)
  - Sridhar Mahadevan, "[Samuel Meets Amarel: Automating Value Function Approximation using Global State Space Analysis](#)", Proceedings of the National Conference on Artificial Intelligence (AAAI-2005), Pittsburgh, PA, July 9-13, 2005.
  - Sridhar Mahadevan, "[Representation Policy Iteration](#)", Proceedings of the 21st Conference on Uncertainty in AI (UAI-2005), Edinburgh, Scotland, July 26-29, 2005.
  - Sridhar Mahadevan, "[Proto-Value Functions: Developmental Reinforcement Learning](#)", Proceedings of the International Conference on Machine Learning (ICML-2005), Bonn, Germany, August 7-13, 2005.
- Wavelet bases
  - Sridhar Mahadevan and Mauro Maggioni, "[Value Function Approximation using Diffusion Wavelets and Laplacian Eigenfunctions](#)", Neural Information Processing Systems (NIPS) conference, Vancouver, December, 2005.
- Fast policy evaluation
  - Mauro Maggioni and Sridhar Mahadevan, "[Fast Direct Policy Evaluation Using Multiscale Markov Diffusion Processes](#)", University of Massachusetts, Department of Computer Science Technical Report TR-2005-39, 2005 (also accepted to ICML 2006)

# Acknowledgements

- National Science Foundation
- Autonomous Learning Lab, U.Mass
  - Kimberly Ferguson, Jeff Johns, Sarah Osentoski, Chang Wang
- Yale Applied Math Department
  - Ronald Coifman