Making high order methods effective

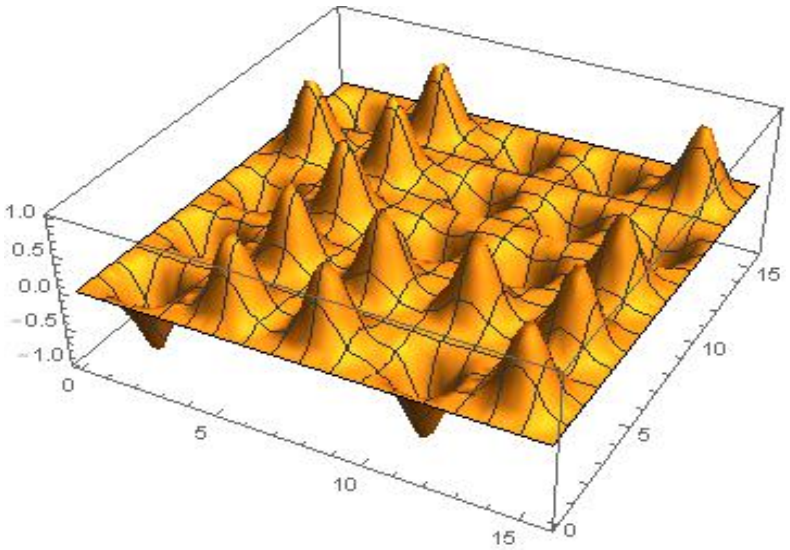
Terry Lyons[13] joint work with Maria Tchernychova[13], Wei Pan[1], Hao Ni[13] and Horatio Boedihardjo[23]

[1]Mathematical Institute and OMI, University of Oxford
[2]Department of Mathematics, Reading University
[3]Supported by ERC advanced grant

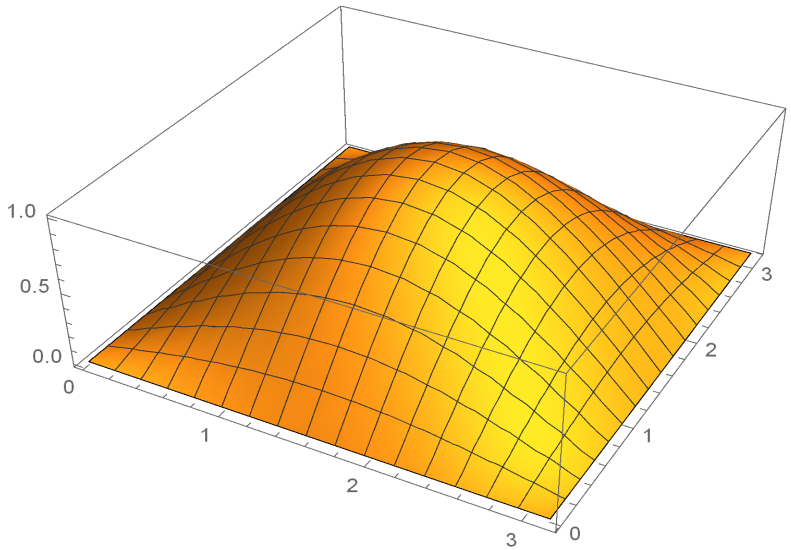
Monte Carlo techniques, Paris, July 5-8, 2016

Outline

PDEs are about propagation of knowledge and relationships



(a) An egg carton



(b) A dome

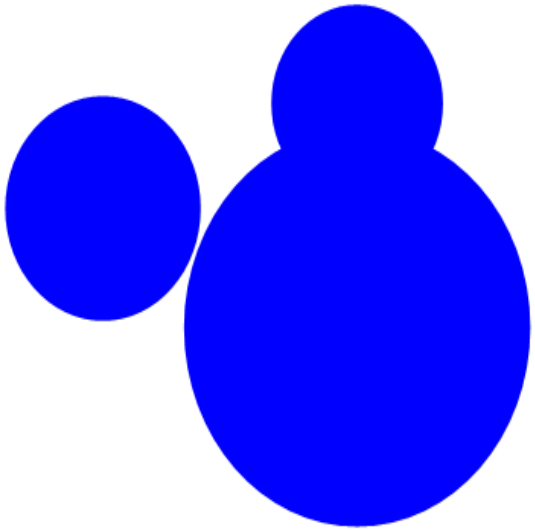
Figure: The curse of Information not the curse of Dimension

Integration and Differentiation

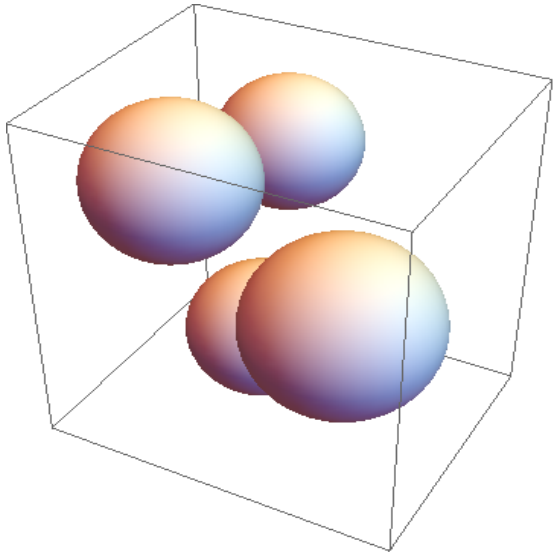
In moderate dimension even tracking is still a challenge

- Singular diffusions in 6 dimensions (position and momentum);
- The weather;
- Solve a parabolic PDE with very narrowly supported boundary data;
- There are techniques - hybrid and exploiting many ideas. They sometimes work!
- Wonjung Lee : Adaptive patched particle filter[3].
- Specific High Dimensional Data Techniques essential - asymptotics largely irrelevant

Cubature - well thought scenarios - a basis for the functions



(a) Nested Cubature



(b) Nested Cubature

Figure: Summarising Lebesgue measure on a cube

Functions

Smooth functions - local bases

$$y = f(x) \approx \sum_{i=1\ldots n} \phi_1(x), x \in \mathbb{R}^d.$$

1. polynomial or wavelet basis: $x^0, x^1, x^2, \ldots, x^n$;
2. Fix degree, f, accuracy ϵ and get a scale R;
3. Computation trades between order and scale - usually avoids the extremes.

Regression and the learning of functions)

1. From a few values $f(x_j)$, identify an approximation to f to error ϵ over the region of radius R.
2. Use a basis and linear methods and add anti-overfitting techniques like lasso.

Many approaches to integration and differentiation

Exploration and integration

1. Monte carlo - slow but broadly applicable;
2. Grids - still low order - curse of dimension;
3. Cubature - not well known for good reason.
4. Sparse grids - unstable;

High order methods much more effective

Harder to implement

1. Litterer: High order recombination and an application to cubature on Wiener space, [4];
2. Best speed from adaptive and truly high order approaches.
3. Diffusion 32 order, patching 11th Order
4. to time $<$ 1s. $10^{-10} <$ accuracy $< 10^{-12}$

The competition: an Oxford colleague

I've now tried a few approaches to solving your problem but can't get results even close to yours in terms of accuracy achieved in such a small amount of CPU time. In all cases I've solved the heat equation on the spatial interval $-9.9 < x < 10.1$ (so that with a coarse uniform mesh the point $x = 0$; was not a node). Then to look at the error I have computed the solution at time 1 and for x integer between -5 and 5 as you suggest.

The competition: an Oxford colleague

The first approach: I took was to do an adaptive finite element solution with the adaptivity geared towards getting an accurate solution at time $t = 1$. The mesh can change at every timestep which is obviously less than ideal as you then need to keep recomputing the matrices. The code is taking about 30 seconds and giving accuracy of between 10^-4 and 10^-7 depending on which integer you look at. It actually turns out to be more efficient to do something a bit more naive, namely to adapt the mesh to resolve the initial condition well and then use that mesh for the rest of the computation. As expected, this clusters the nodes around $x = 0$ and the mesh is fairly coarse elsewhere. The advantage of this is that you just solve the same matrix problem at every timestep. This speeds things up a lot without degrading the accuracy for this problem. So here I'm getting accuracy of between 10^-4 and 10^-6 in about 1 second.

The competition: an Oxford colleague

Then, finally: I gave Nick Trefethen et al's Matlab package Chebfun a go. In order to solve the heat equation which exploits the fact that the problem is linear so you can write the solution at a given time t as $exp(tL)u0$ where L is the spatial operator (including boundary conditions) and $u0$ is the initial condition. It seems that Chebfun struggles when $u0$ is not smooth and it actually turns out to be more efficient to compute the solution at time $t = 1$ in two stages... can solve the same type of problem in 6.5s giving errors of $5 * 10^{-}6$.

Parabolic PDEs as integrals

Parabolic PDEs as expectations

Solve the parabolic PDE whose second order term is $L = \sum(V^i V^i)$ via

$$dY_t = V^i(Y_t)dW_t^i$$
$$u(x,0) = \mathbb{E}\left(u(Y_T, T)\right)$$

How should we approximate this integration?

Linearising SDEs

Definition (The signature of a path)

Let $X : J \to E := \mathbb{R}^m$ be a continuous path with finite length. For every $[s,t] \subset [0,T]$, the signature of $S(X_{[s,t]})$ in $T((E))$ is defined as follows

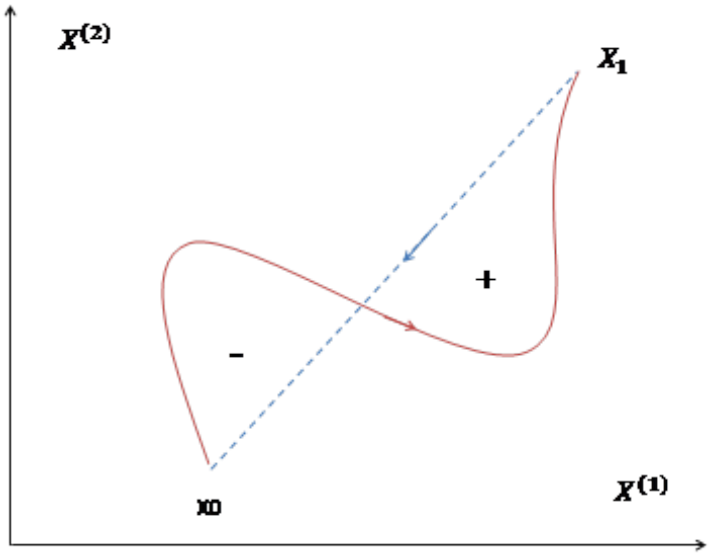
$$S(X_{[s,t]}) = (1, X^1_{[s,t]}, X^2_{[s,t]}, \ldots),$$

where, for each $n \geq 1$,

$$X^n_{[s,t]} = \int_s^t \int_0^{u_n} \cdots \int_0^{u_2} dX_{u_1} \otimes \ldots \otimes dX_{u_n}.$$

Signature - A top-down description on the path

- Level 1 - increment of a path; Level 2 - area of a path;
- Higher degree- a local structure of a path.
- Uniqueness of the signature ([2], [1]).



Linear Differential Controlled Equation

Let $Y : [0, T] \to \mathbb{R}$ satisfy

$$dY_t = AY_t dX_t, \; Y_0 = y_0,$$

where $X : [0, T] \to \mathbb{R}^d$ be a continuous path of bounded variation, and $A : \mathbb{R} \to L(\mathbb{R}^d, \mathbb{R})$ be a bounded linear map.

Picard's iteration

$$\begin{aligned} Y_T &= y_0 + \sum_{n=1}^{\infty} A^{\otimes n} y_0 \int_0^T \int_0^{u_n} \cdots \int_0^{u_2} dX_{u_1} \otimes \ldots \otimes dX_{u_n}. \\ (1d) &= y_0 + \sum_{n=1}^{\infty} A^n y_0 \frac{(X_T - X_0)^n}{n!} = y_0 \exp(A(X_T - X_0)). \end{aligned}$$

Hao Ni

Learning Stochastic Differential Equation

Suppose Y_t satisfies the following Stochastic Differential Equation:

$$dY_t = a(1 - Y_t)dX_t^{(1)} + bY_t^2 dX_t^{(2)}, Y_0 = 0.$$

where $X_t = (X_t^{(1)}, X_t^{(2)}) = (t, W_t)$, and the integral is in the Stratonovich sense, and (a, b) is chosen to $(1, 2)$.

Dataset

We generate 800 independent samples of pairs $(X_{[0,T]}, Y_T)$ using Milstein's method with discretization step 0.001. Half of the samples are used for the training set, and the rest is for the back-testing set.

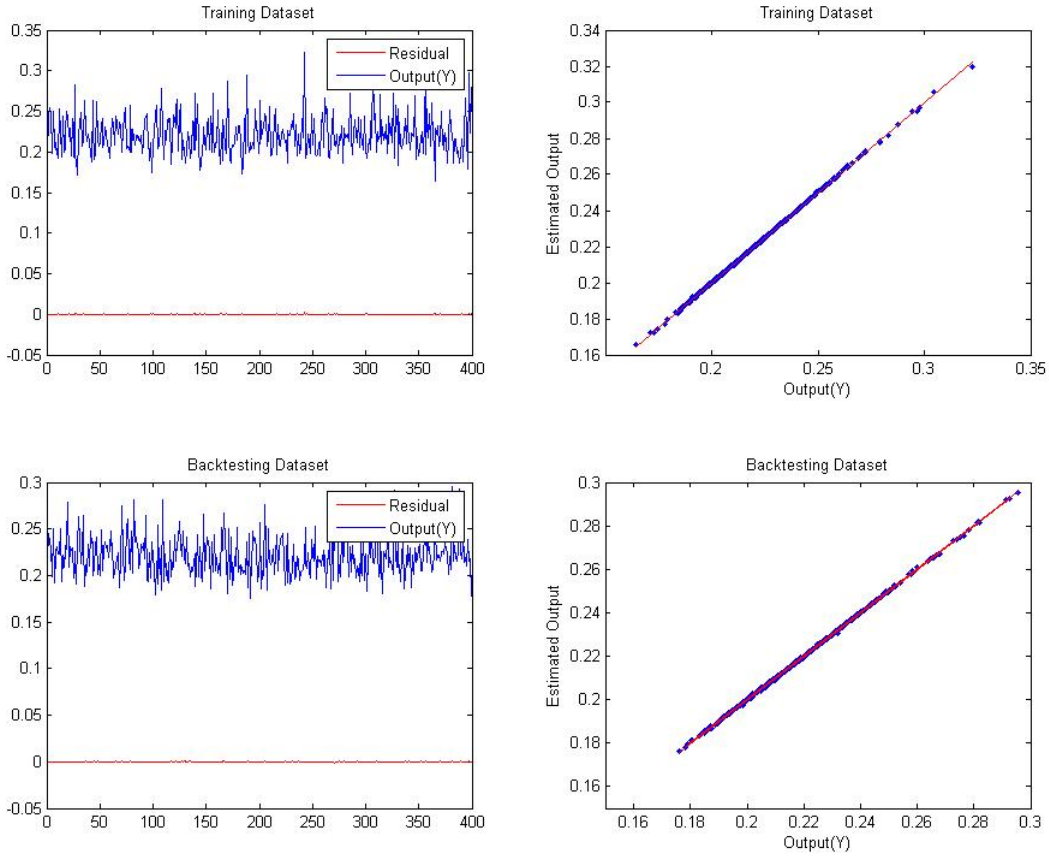


Figure: $T = 0.25$, Degree of signatures = 4

Cubature on Wiener Space

High order sampling of paths

Integration over path space can be well approximated if we can find paths and weights so that the integrals of (expectations of) the first few iterated integrals are correct. Cubature on Wiener space looks for a discrete parsimonious set of paths so that the expected value of the first few terms in the signature for the discrete measure and the wiener measure agree. Only require a finite set of paths.

Leads to high order methods of integration. A deterministic method with deterministic a-priori error bounds.

No Grid, Iteration leads to an explosion!

(a) Clenshaw Curtis 7 Points (b) After one iteration - 49 points!

Figure: If there are D paths in the cubature, then n-multistep leads to D^n.

Points, Dynamic Cubature and Recombination

Recombination

It is possible to redistribute the weights and so reduce the number of points to the minimum. With a d dimensional cloud of N weighted points, then computation to reduce the cloud to D points having the same moments ($\leq n$) requires $ND + D^3 \log(N/D)$ units of time and needs storage of $N + D^2$ storage. $D(n,d) = (n+d)!/(n!d!)$.

Core Ideas

Approximating a cloud

1. Stability - use the same points.
2. Accuracy - integrate basic functions correctly
3. Use the basic functions to translate the problem to an example of Caratheodory's theorem.

Dynamic Caratheodory theorem [6]

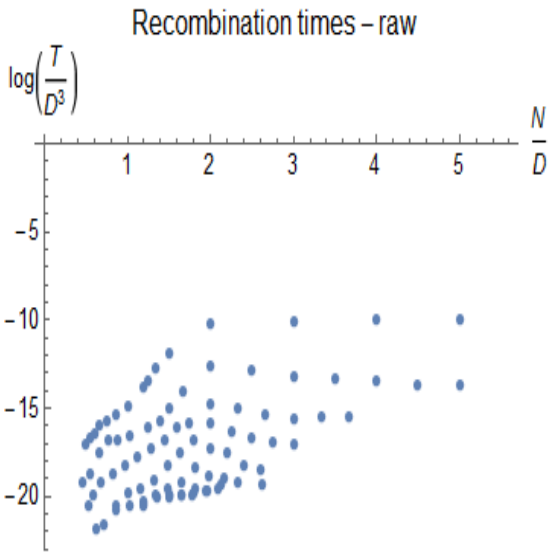
a divide and conquer approach

1. Cluster weighted points into $2D + 1$ subclusters
2. Eliminate D clusters. Repeat

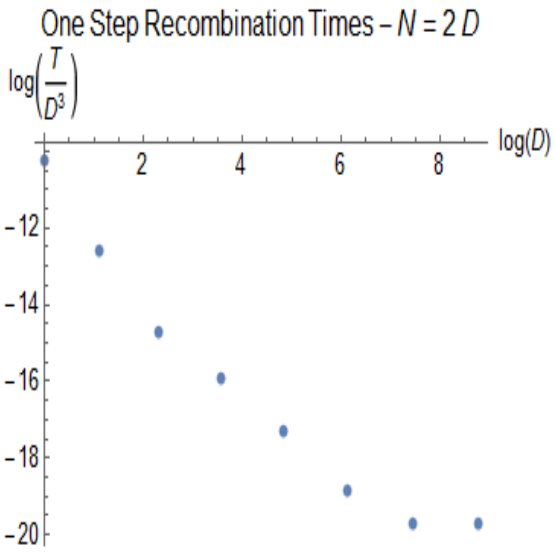
The workhorse: $2D \to D$

1. Use SVD (+random matrix theory for numerical kernel) to find the full kernel of weight changes on $2D$ points that do not change the COG
2. Use pivoted Gaussian elimination with these vectors to remove all mass from D of the points.

The workhorse: Dynamic Caratheodory $2D \to D$

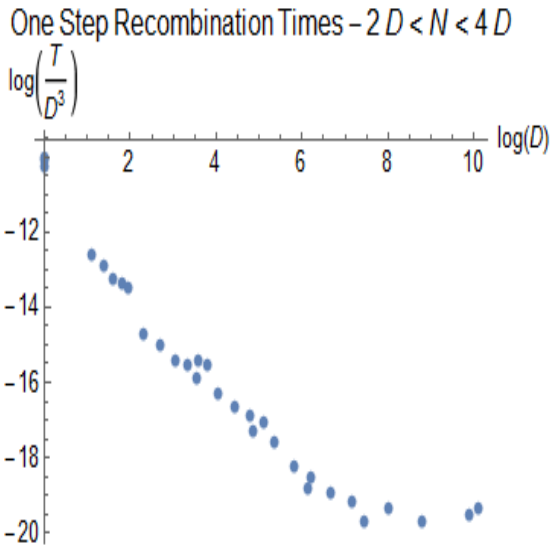


(a) Raw Times

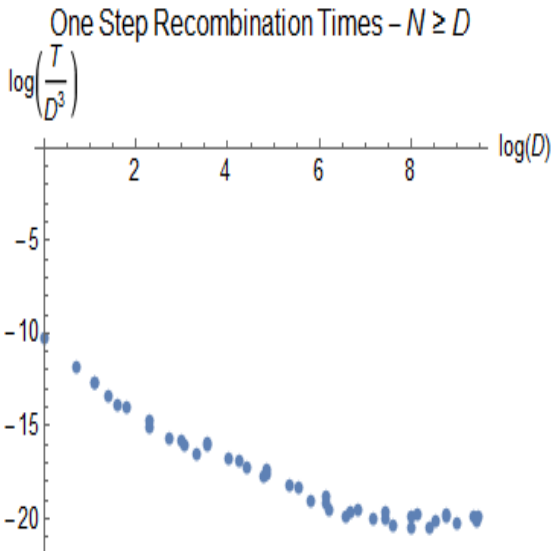


(b) exactly $2D \to D$

Figure: ((1, 2, 0.0000353381), (3, 6, 0.0000898817), (10, 20, 0.00040562), (35, 70, 0.00539367), (126, 252, 0.0604481), (462, 918, 0.65147), (1716, 3376, 14.4207), (6435, 12510, 726.204))



(a) Eliminating more than $2D$



(b) Eliminating less than $2D$

Figure: No difference in computation time if one eliminates 1, D or 2D points

A dense case of the crossover problem in linear programming

The case $N = D^2$ is the most typical. Work of Christian Litterer, and more recently Maria Tchernynova's thesis. Although not a light approach, real time recombination can transform the numerical approaches to solving PDEs and make totally feasible problems that were previously out of reach.
It can be also regarded as a rather special problem in Linear Programming, however our methods empirically outperform the state of the art LP packages by a small order. Recombination is a powerful, and completely non-naive, data reduction technique.

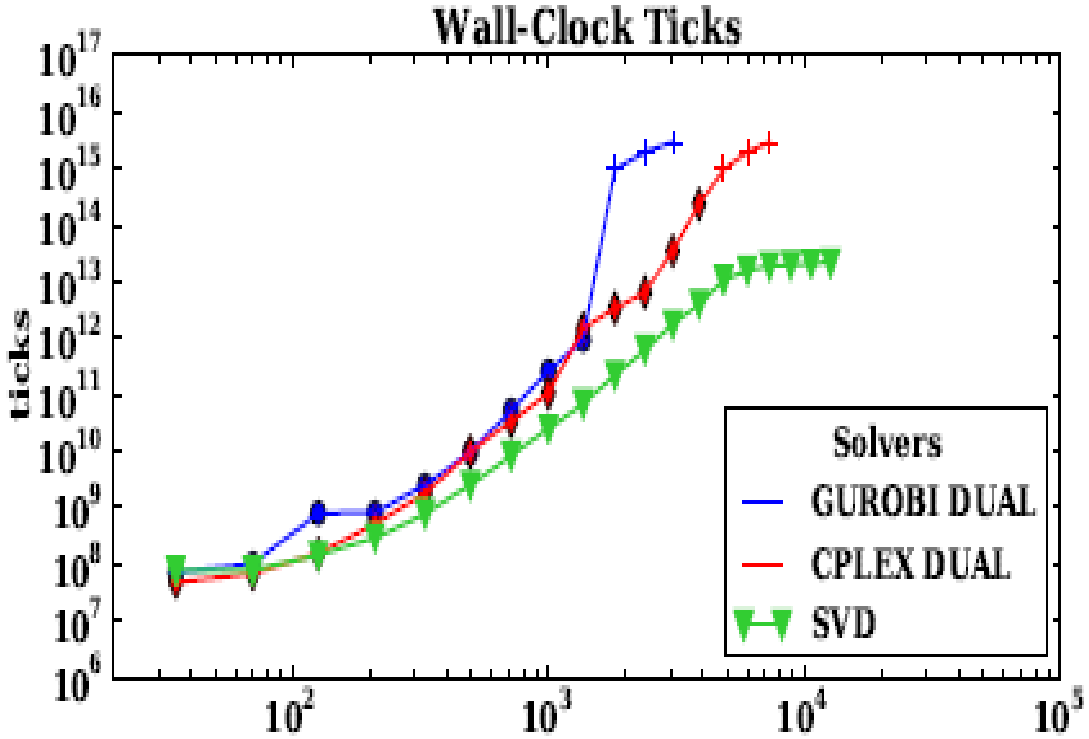


Figure: A comparison

dim$= d$	$M = \binom{m+d}{d}$	dim$= d$	$M = \binom{m+d}{d}$
3	35	17	5985
4	70	18	7315
5	126	19	8855
6	210	20	10626
7	330	21	12650
8	495	22	14950
9	715	23	17550
10	1001	24	20475
11	1365	25	23751
12	1820	26	27405
13	2380	27	31465
14	3060	28	35960
15	3876	29	40920
16	4845	30	46376

Figure: Dimensions of spaces of polynomials of degree 4

Wei Pan: Memoization and Space filling curves

Memoization for backward equations?

High order methods require that one evaluates functions where one chooses and accurately! Nonlinearity then forces evaluation of other functions that would evaluate other functions ... . Memoization is a computer science technique that recognises when you have seen a case before (log NoCases) and then uses the previously computed value. It has huge potential for accelerating nonlinear numerics - backward optimisation problems etc.

Wei Pan: Space Filling Paths

Just in time evaluation

If u is piecewise smooth, locally well approximated by a polynomial of fixed degree. Only evaluate when you need to. If you have a big enough demand for the function in a neighbourhood, then use evaluations to extrapolate. The challenge is to find nearby evaluations [5].

Simplest

Take the square to the line $\{.a1a2..., .b1b2...\} \to .a1b1a2b2...$ using the Morton order which is easily computed. Store points in 1d order. Logarithmic algorithm to find all entries in a dyadic box. Binary search trees have same effect.

References I


H Boedihardjo, X Geng, T Lyons, and D Yang.
The signature of a rough path: Uniqueness.
Advances in Mathematics, 293:720—737, 2016.


B.M. Hambly and Terry Lyons.
Uniqueness for the signature of a path of bounded variation and the reduced path group.
Annals of Mathematics, 171(1):109–167, 2010.


Wonjung Lee and Terry Lyons.
The adaptive patched particle filter and its implementation.
Communications in Mathematical Sciences, to appear, http://www.math.wisc.edu/~jcms/, preprint arXiv:1509.04239, 2016.

References II

Christian Litterer and Terry Lyons.
High order recombination and an application to cubature on wiener space.
The Annals of Applied Probability, pages 1301–1327, 2012.

Wei Pan.
Dphil thesis.
To appear: Thesis, University of Oxford, 2016.

Maria Tchernychova.
Caratheodory cubature measures.
Thesis, University of Oxford, 2015.