Sparse Grids for Stochastic Integrals

John Burkardt,
Department of Scientific Computing
Florida State University

.....

Workshop on Efficient Numerical Methods for PDE's,
Hosted by Professor Hyung-Chun Lee,
Ajou University, Suwon, Korea
24 September 2011.

.....

https://people.sc.fsu.edu/~jburkardt/presentations/sparse_2011_ajou.pdf

Sparse Grids for Stochastic Integrals

STOCHASTIC: Background

Stochastic problems are mathematical models that attempt to include an aspect that is nondeterministic or uncertain.

In a mathematical model, even uncertainty has to be described precisely.

So when an uncertain process produces a new state or result, we imagine this as coming about because that state was selected from an ensemble of possible states, with each state having a particular probability of being selected.

If our stochastic problem involved only a single such choice, then it should be clear that we can "understand" our problem by considering every possible outcome, and that we can compute an expected value by multiplying every possible outcome by its probability.

STOCHASTIC: Background

Partial differential equations allow us to describe the behavior of of a quantity u($\vec{x}$) and often to solve for or approximate its value.

A classic example is the diffusion equation:

$$-\nabla \cdot (a(\vec{x})\nabla u(\vec{x})) = f(\vec{x})$$

a($\vec{x}$) is a spatially varying diffusion coefficient;
f($\vec{x}$) represents a source term.

STOCHASTIC: Background

A finite element approach would produce an approximate solution by integrating the equation against various test functions $v_i(\vec{x})$:

$$\int_D a(\vec{x})\,\nabla u(\vec{x}) \cdot \nabla v_i(\vec{x})\,d\vec{x} = \int_D f(\vec{x}) v_i(\vec{x})\,d\vec{x}$$

By assuming $u(\vec{x}) = \sum_{j=1}^{N} c_j v_j(\vec{x})$, this becomes a set of N linear algebraic equations $A * c = f$.

Evaluating A and f requires approximate integration over D.

STOCHASTIC: Background

A stochastic version of this equation might allow uncertainties in the diffusion coefficient:

$$-\nabla \cdot (a(\vec{x};\omega)\nabla u(\vec{x};\omega)) = f(\vec{x})$$

Here, ω represents the stochastic component, and we must even write u with an implicit dependence on ω, through a.

We could, if we liked, also consider uncertainties in the source term f, or the initial or boundary conditions.

STOCHASTIC: Background

Since ω is an unknown and undetermined quantity, it might seem that the solution process is hopeless.

Actually, if we could specify a particular set of values for the stochastic component ω, then presumably we could solve for u, so our problem is really that our classical solution has now become a family of solutions with parameter ω.

Moreoever, if, with each value of ω we can associate a probability density function $\rho(\omega)$, then we can now reformulate our problem to solve for the finite element coefficients of the expected value function $\bar{u}(\vec{x})$.

STOCHASTIC: Background

We want to know how stochasticity will affect the classical solution $u(\vec{x})$; comparing $u(\vec{x})$ to $\bar{u}(\vec{x})$ will indicate the expected magnitude of the deviations caused by the stochastic component.

To compute $\bar{u}(\vec{x})$ or other statistical moments, we add stochasticity, weight it, and then integrate it out. The integration processes collapses the infinite family of perturbed solutions into one object which represents a bulk average perturbed solution, and which can be compared to a classical solution.

You can plot $\bar{u}(\vec{x})$ for instance, or evaluate it, or find its maximum.

STOCHASTIC: Background

If we simply integrate over the whole probability space, we get:

$$\int_\Omega \int_D a(\vec{x};\omega)\,\nabla u(\vec{x};\omega)\cdot\nabla v_i(\vec{x})\,d\vec{x}\,\rho(\omega)\,d\omega =$$
$$\int_\Omega \int_D f(\vec{x})v_i(\vec{x};\omega)\,d\vec{x}\,\rho(\omega)\,d\omega$$

This is an algebraic system $A * c = f$ for the finite element coefficients of $\bar{u}(\vec{x})$, but evaluating A and f requires approximate integration over D AND over the probability space Ω^M.

We must approximate this integral, where Ω is a product region whose dimension **M** may be high, and in which the integrands are probably smooth.

Sparse Grids for Stochastic Integrals

1. Integrals Arising from Stochastic Problems
2. **Quadrature Rules**
3. Smolyak Quadrature
4. Quadrature Test Example
5. Stochastic PDE Example
6. Conclusion

QUADRATURE: A 1D Rule

A 1D quadrature rule is a set of **N** points **X** and weights **W** intended to approximate integrals over a particular interval **[a,b]**, perhaps including a weight function:

$$\int_a^b f(x)\,\rho(x)\,dx \approx \sum_{i=1}^{N} w_i\,f(x_i)$$

For the integrals we are interested in, the interval is either [-1,+1], $[0,+\infty)$ or $(-\infty,+\infty)$ and ρ(x) will be a smooth function.

QUADRATURE: Precision, Order

If a 1D quadrature rule can compute the exact integral when f(x) is a polynomial of degree **P** or less, the rule has **precision P**.

The precision of common quadrature families can be given in terms of the **order** (number of points) **N**:

- Interpolatory rules: **P = N-1** or **N**
- Gauss rules **P = 2 * N - 1** ;
- Monte Carlo and Quasi-Monte Carlo rules, **P = 0**;
- "transform rules": tanh, tanh-sinh, erf rules **P = 1**.

QUADRATURE: Accuracy

Using a rule with **P = N** on a function with **N** derivatives, the low degree terms get integrated exactly, leaving error $\sim O(\frac{1}{N}^{N+1})$.

(Taking the typical spacing between abscissas to be $h = \frac{1}{N}$.)

The integrands encountered in high dimensional problems are typically smooth, and suitable for high precision rules.

An interpolation rule with order N will typically have precision $P = N - 1$; we'll assume the rule and interval are symmetric, and that N is odd, in which case we get $P = N$.

QUADRATURE: Family of 1D Rules

A **family** of 1D quadrature rules are an indexed set of rules of increasing order.

The **level** of a rule, **L**, is its index in a family. This is simply a way to select a subset of the rules for our iterative scheme.

A family typically starts at level 0 with an order 1 rule.

The **growth rule** for a family relates level **L** and order **N**:

Linear growth allows a limited amount of nesting:

$$N = 2L + 1 : 1, 3, 5, 7, ...$$

Exponential growth has perfect nesting, but rapid order growth:

$$N = 2^L + 1 : 1, 3, 5, 9, 17, 33 :$$

QUADRATURE: Product Rules

Let $\mathcal{Q}_L$ be the L-th member of a family of 1D quadrature rules, with order N_L and precision P_L.

We can construct a corresponding family of 2D product rules as $\mathcal{Q}_L \otimes \mathcal{Q}_L$, with order N_L^2 and precision P_L.

This rule is based on interpolating data on the product grid; the analysis of precision and accuracy is similar to the 1D case.

Similarly, we can construct an M-dimensional product rule. Notice that the order of such a rule is N_L^M, but the precision stays at P_L. Thus, even for a relatively low precision requirement, the order of a product rule will grow very rapidly if M increases.

How many points would you use in a product rule for M=*100?*

QUADRATURE: Do We Get Our Money's Worth?

Suppose we form a 2D quadrature rule by "squaring" a 1D rule which is precise for monomials 1 through x^4.

Our 2D product rule will be precise for any monomial in x and y with individual degrees no greater than 4.

The number of monomials we will be able to integrate exactly matches the number of abscissas the rule requires.

Our expense, function evaluations at the abscissa, seems to buy us a corresponding great deal of monomial exactness.

But for interpolatory quadrature, many of the monomial results we "buy" are actually **nearly worthless** in the sense that they don't improve our asymptotic accuracy.

QUADRATURE: Pascal's Precision Triangle

Here are the monomials of total degree exactly 5. A rule has precision 5 if it can integrate these and all monomials below that diagonal.

7	!	y^7	xy^7	x^2y^7	x^3y^7	x^4y^7	x^5y^7	x^6y^7	x^7y^7
6	!	y^6	xy^6	x^2y^6	x^3y^6	x^4y^6	x^5y^6	x^6y^6	x^7y^6
5	!	y^5	xy^5	x^2y^5	x^3y^5	x^4y^5	x^5y^5	x^6y^5	x^7y^5
4	!	y^4	xy^4	x^2y^4	x^3y^4	x^4y^4	x^5y^4	x^6y^4	x^7y^4
3	!	y^3	xy^3	x^2y^3	x^3y^3	x^4y^3	x^5y^3	x^6y^3	x^7y^3
2	!	y^2	xy^2	x^2y^2	x^3y^2	x^4y^2	x^5y^2	x^6y^2	x^7y^2
1	!	y	xy	x^2y	x^3y	x^4	x^5y	x^6y	x^7y
0	!	1	x	x^2	x^3	x^4	x^5	x^6	x^7
P	!	0	1	2	3	4	5	6	7

QUADRATURE: Pascal's Precision Triangle

A product rule results in a rectangle of precision, not a triangle. The monomials above the diagonal of that rectangle represent a cost that does not correspond to increased overall asymptotic accuracy.

7	!	y^7	xy^7	x^2y^7	x^3y^7	x^4y^7	x^5y^7	x^6y^7	x^7y^7
6	!	y^6	xy^6	x^2y^6	x^3y^6	x^4y^6	x^5y^6	x^6y^6	x^7y^6
5	!	y^5	xy^5	x^2y^5	x^3y^5	x^4y^5	x^5y^5	x^6y^5	x^7y^5
4	!	y^4	xy^4	x^2y^4	x^3y^4	x^4y^4	x^5y^4	x^6y^4	x^7y^4
3	!	y^3	xy^3	x^2y^3	x^3y^3	x^4y^3	x^5y^3	x^6y^3	x^7y^3
2	!	y^2	xy^2	x^2y^2	x^3y^2	x^4y^2	x^5y^2	x^6y^2	x^7y^2
1	!	y	xy	x^2y	x^3y	x^4	x^5y	x^6y	x^7y
0	!	1	x	x^2	x^3	x^4	x^5	x^6	x^7
P	!	0	1	2	3	4	5	6	7

QUADRATURE: It Gets Worse in Higher Dimensions

Consider products of a 10 point rule with precision up to x^9.

We only need to get to diagonal 9 of Pascal's precision triangle. The monomials up to that row can be computed as a multinomial coefficient. Compare the number of abscissas to monomials!

Dim	Abscissas	Monomials	Wasted	Percentage
1D	10	10	0	0%
2D	100	55	45	45%
3D	1,000	220	780	78%
4D	10,000	715	9,285	92%
5D	100,000	2,002	97,998	97%
6D	1,000,000	5,005	994,995	99%

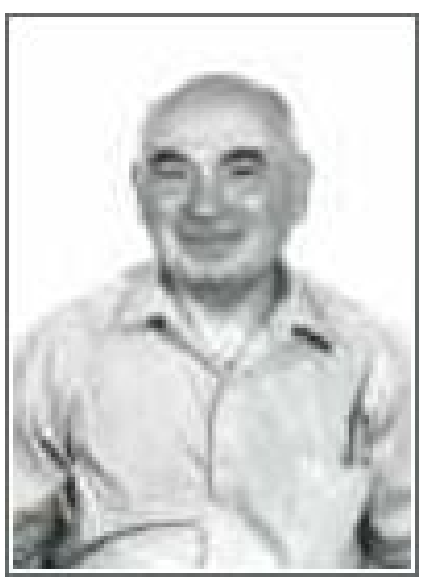
In 5D, there are only 2,002 items to search for.
Can't we find a quadrature rule of roughly that order?

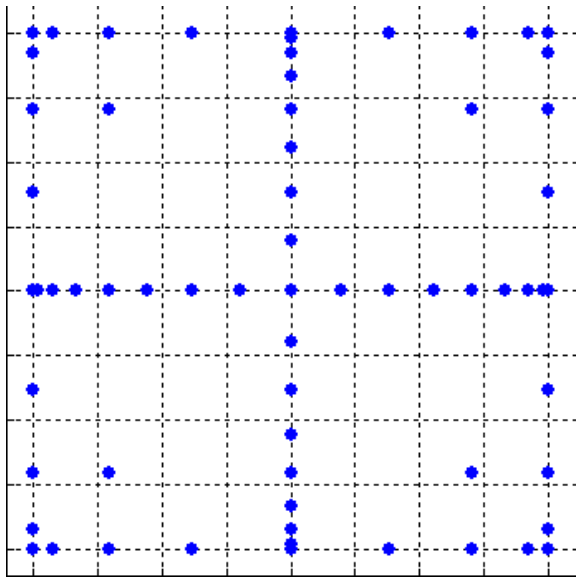
Sparse Grids for Stochastic Integrals

1. Integrals Arising from Stochastic Problems
2. Quadrature Rules
3. **Smolyak Quadrature**
4. Quadrature Test Example
5. Stochastic PDE Example
6. Conclusion

SMOLYAK

Sergey Smolyak (1963) suggested **sparse grids**:





- an algebraic combination of low order product grids;
- Pascal's precision diagonals achieved with far fewer points;

Smooth $f(x)$ + rule precision $\Rightarrow$ accuracy + efficiency.

SMOLYAK: Construction

Starting from an indexed family of 1D quadrature rules $\mathcal{Q}^L$,
we form rules for dimension **M**, indexed by level **L**.
Here $\mathrm{i} = i_1 + \cdots + i_M$, where i_j is the "level" of the j-th 1D rule.

$$\mathcal{A}(L,M) = \sum_{L-M+1 \leq |\mathrm{i}| \leq L} (-1)^{L+M-|\mathrm{i}|} \left(\begin{array}{c} L+M \\ L+M-|\mathrm{i}| \end{array} \right) (\mathcal{Q}^{i_1} \otimes \cdots \otimes \mathcal{Q}^{i_M})$$

The rule $\mathcal{A}(L,M)$ is a weighted sum of product rules, indexed by **L**, with an **M**-dimensional point count of **N**.

As **L** (and **N**) increase, we expect the precision **P** to increase.

Theorem

The Clenshaw-Curtis Smolyak formula of level **L** *is precise for all polynomials of degree* $P = 2 * L + 1$ *or less.*

SMOLYAK: Precision Table for Clenshaw Curtis

Level L	Precision P	N(1D)	N(5D)	N(10D)
0	1	1	1	1
1	3	3	11	21
2	5	5	61	221
3	7	9	241	1581
4	9	17	801	8801
5	11	33	2433	41265
6	13	65	6993	171425

Compare a 5D product rule, which would require $\mathbf{N} = 100{,}000$ abscissas to integrate 2,002 entries in Pascal's precision triangle (precision $\mathbf{P} = 9$).

SMOLYAK: Asymptotic Accuracy

Let N be the order in the rule $A(L, M)$.

and let I be the integral of $f(x)$ where:

$f(x) : [-1, 1]^M \rightarrow R | D^\alpha$ continuous if $\alpha_i \leq r$ for all i;

Then the accuracy of a Smolyak rule based on a nested 1D quadrature family satisfies:

$$||I - A(L, M)|| = O(N^{\frac{-r}{\log(2M)}})$$

This behavior is near optimal; no family of rules could do better than $O(N^{-r})$ for this general class of integrands.

SMOLYAK: Efficiency

The space of **M**-dimensional polynomials of degree **P** or less has dimension $\left(\begin{array}{c} P+M \\ M \end{array} \right) \approx \frac{M^P}{P!}$.

For large **M**, a Clenshaw-Curtis Smolyak rule that achieves precision **P** uses $N \approx \frac{(2M)^P}{P!}$ points.

Thus, if we are seeking exact integration of polynomials, the Clenshaw-Curtis Smolyak rule uses an optimal number of points (to within a factor 2^P that is independent of **M**).

Notice there is no exponent of **M** in the point growth.

QUADRATURE EXAMPLE: A Test Case of Genz

Let's take a problem that's reasonable but not trivial.

We'll work in a space with dimension $\mathbf{M} = 6$.

We'll try to integrate the **Genz Product Peak**:

$$f(X) = \frac{1}{\prod_{i=1}^{M}(C_i^2 + (X_i - Z_i)^2)}$$

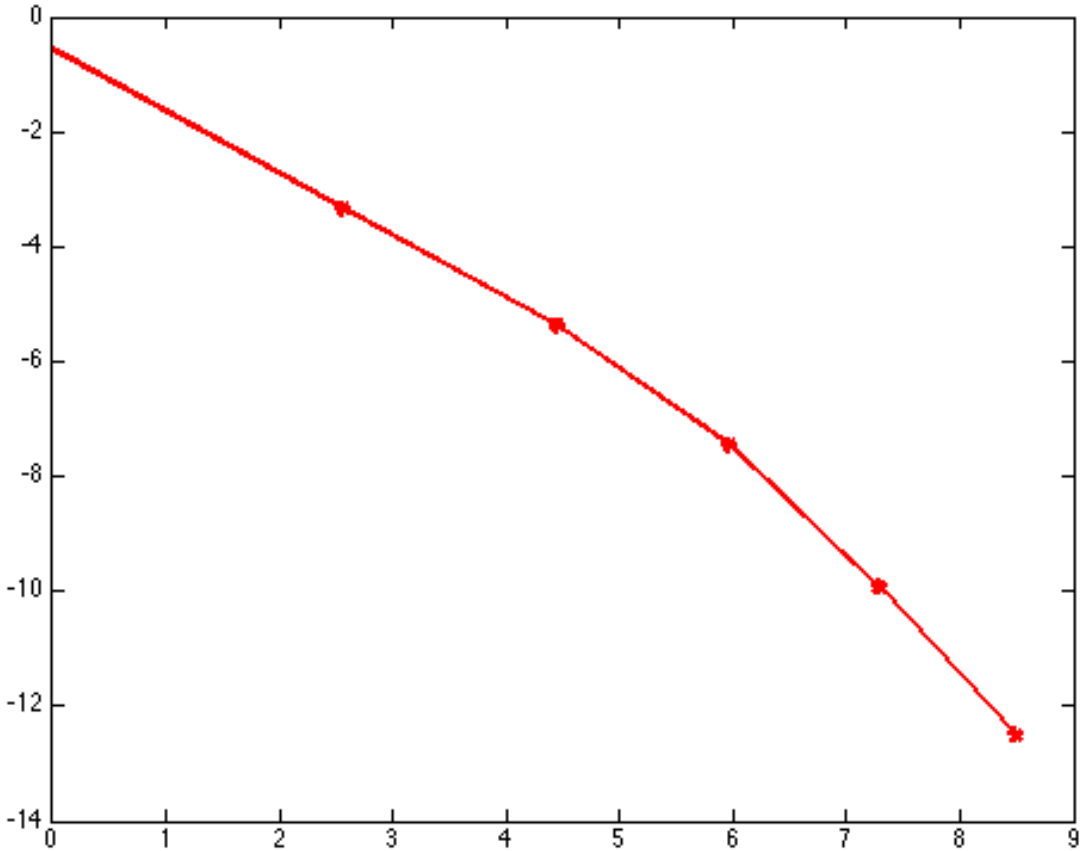
where C_i and Z_i are given M-dimensional points.

QUADRATURE EXAMPLE: 6D Smolyak

Quadrature estimates using a Smolyak sparse grid based on Clenshaw-Curtis points.

Level	Order	Estimate	Error
0	1	0.062500	0.573282
1	13	0.600000	0.0357818
2	85	0.631111	0.00467073
3	389	0.636364	0.000582152
4	1457	0.635831	0.0000492033
5	4865	0.635778	0.00000375410
∞	∞	0.635782	0.0000

QUADRATURE EXAMPLE:6D Smolyak



QUADRATURE EXAMPLE: 6D Gauss-Legendre

Quadrature estimates using a product grid based on Gauss-Legendre rule.

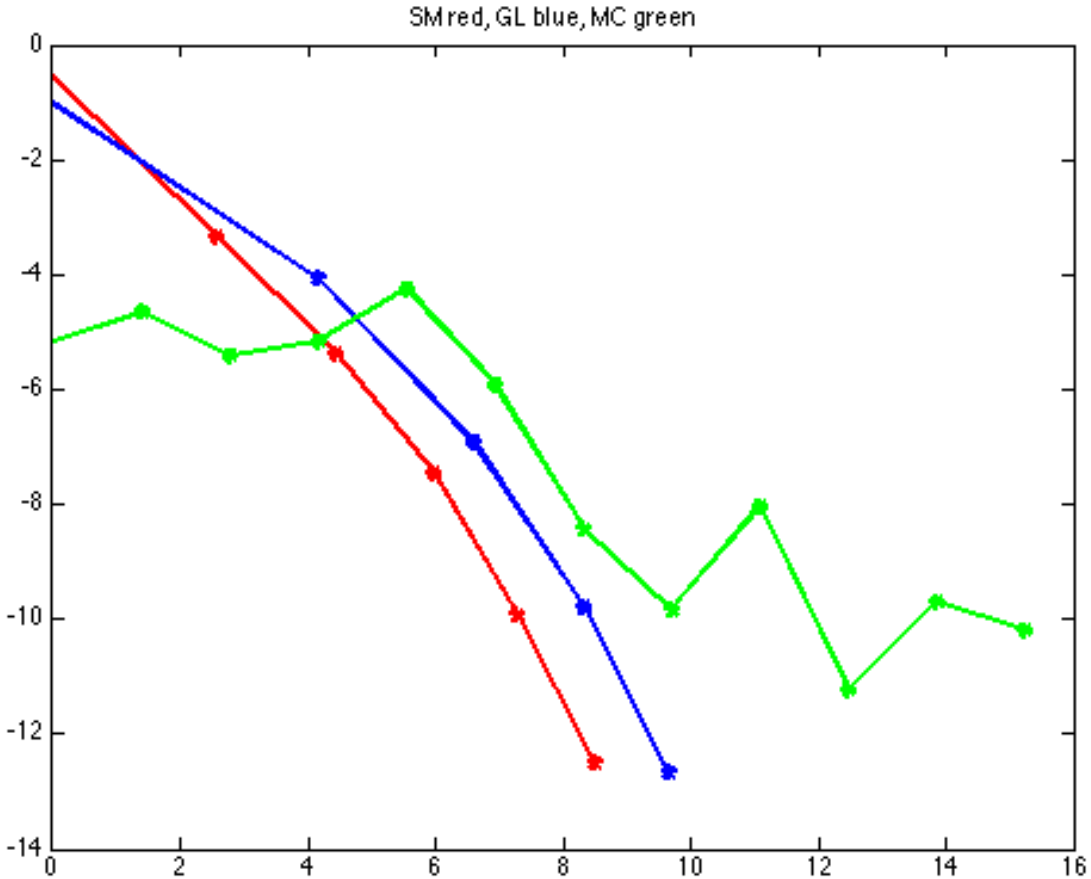
1D Order	6D Order	Estimate	Error
1	1	1.00000	0.364218
2	64	0.618625	0.0171570
3	729	0.636774	0.000992123
4	4096	0.635726	0.0000560162
5	15625	0.635785	0.00000314963
∞	∞	0.635782	0.0000

QUADRATURE EXAMPLE: 6D Monte Carlo

Quadrature estimates using a Monte Carlo approximation with N samples.

log2 (N)	N	Estimate	Error
0	1	0.641468	0.00568631
4	16	0.640218	0.00443594
8	256	0.650114	0.0143321
16	4096	0.636000	0.000218054
24	65536	0.636105	0.000323117
32	1048576	0.635843	0.0000612090
∞	∞	0.635782	0.0

QUADRATURE EXAMPLE: 6D Smolyak/GL/MC



Sparse Grids for Stochastic Integrals

1. Integrals Arising from Stochastic Problems
2. Quadrature Rules
3. Smolyak Quadrature
4. Quadrature Test Example
5. **Stochastic PDE Example**
6. Conclusion

STOCHASTIC EXAMPLE: A Deterministic PDE

$$-\nabla \cdot (a(\vec{x}) \nabla u(\vec{x})) = f(\vec{x})$$

$u(\vec{x})$ is an unknown quantity, like temperature;

$a(\vec{x})$ is a **known** physical property, the conductivity, which controls how quickly hot or cold spots average out.

- heat conduction;
- slow subsurface flow of water;
- particle diffusion;
- Black-Scholes equation (flow of money!).

STOCHASTIC EXAMPLE: Modeling Uncertain Conductivity

Using a fixed value for $a(\vec{x})$ might be unrealistic.

Without variations in $a(\vec{x})$, we might never see the bumps and swirls typical of real physical problems.

We might think of $a(\vec{x})$ as a *random field* $a(\vec{x};\omega)$.

The ω represents the unknown variation from the average.

STOCHASTIC EXAMPLE: Uncertain Solution

If $a(\vec{x};\omega)$ has an "unknown" component, then so does our solution, which we write $u(\vec{x};\omega)$.

$$-\nabla \cdot (a(\vec{x};\omega)\,\nabla u(\vec{x};\omega)) = f(\vec{x})$$

Now if we don't know what the equation is, we can't solve it!

Can we still extract information from the equation?

STOCHASTIC EXAMPLE: Expected Values

Each variation ω determines a solution u.

If we added up every variation, we'd get an average or expected value for the solution.

The expected value is an important first piece of information about a problem with a random component.

$$E(u(\vec{x})) = \int_{\Omega} u(\vec{x};\omega)\,\rho(\omega)\,d\omega$$

It's like using weather records to estimate the *climate*.

STOCHASTIC EXAMPLE: Approximate Integral

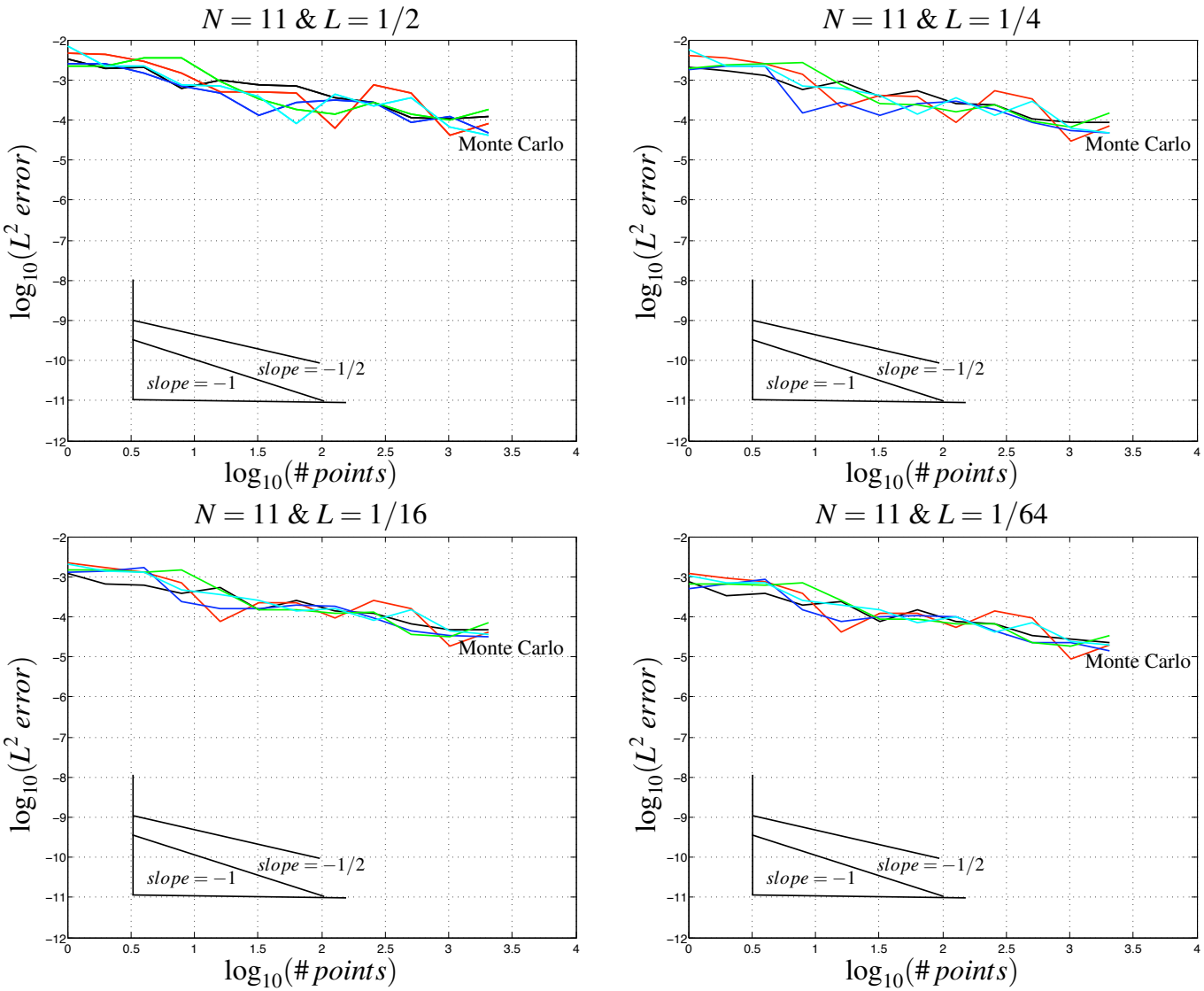
We approximate the function space Ω by an M-dimensional space Ω^M, of linear sums of perturbations ω.

We now estimate the integral of $u(\vec{x};\omega)$ in Ω^M.

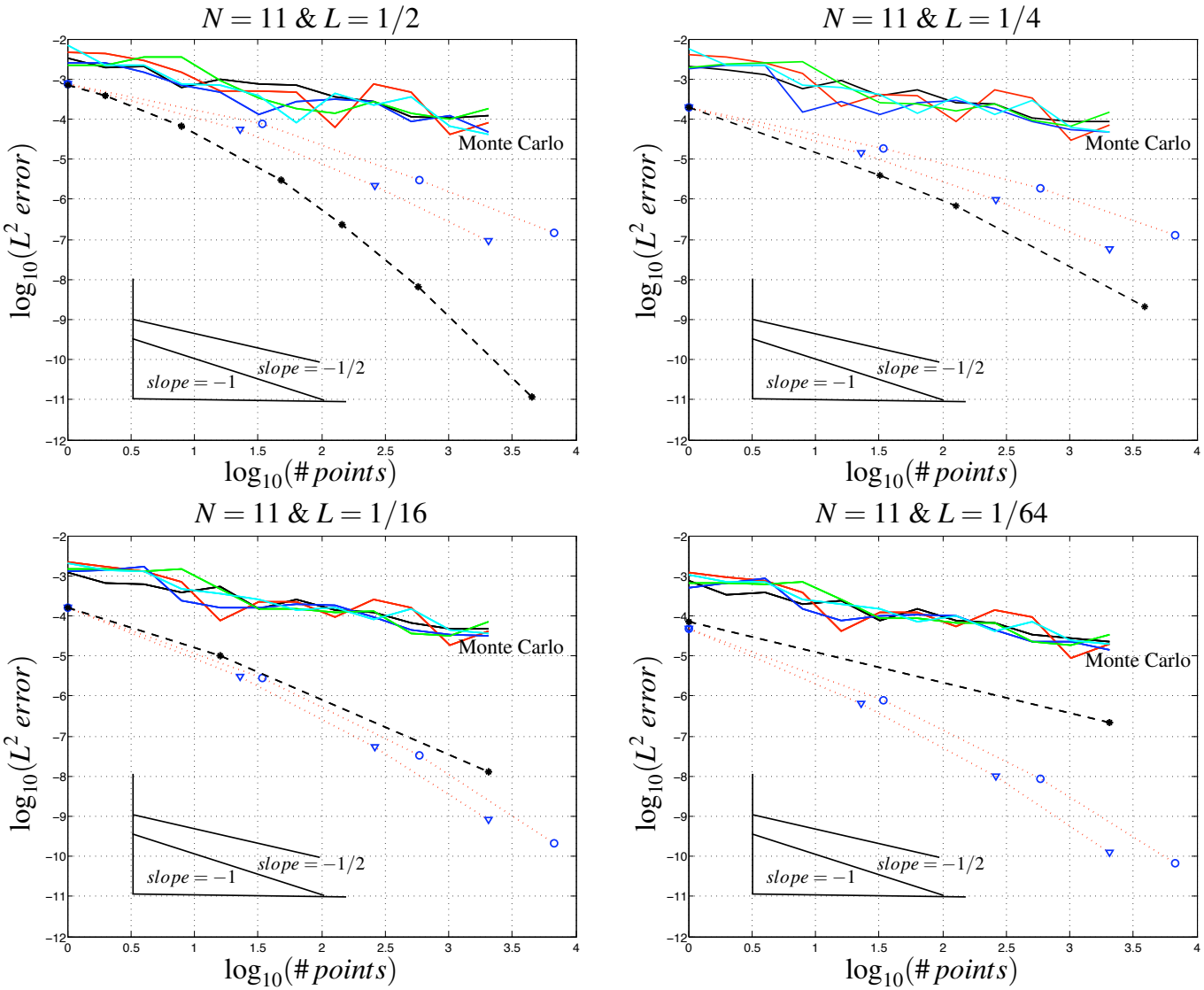
Monte Carlo: select random sets of parameters ω, (weighted by $\rho(\omega)$, solve for $u(\vec{x};\omega)$ and average.

Sparse grid: choose a level, defining a grid of ω values in Ω^M. For each ω grid value, evaluate the spatial integrals to get a contribution to A and f; sum to get A and f, solve $A * x = f$ for the finite element coefficients for $E(u(\vec{x});\omega))$.

STOCHASTIC EXAMPLE: Monte Carlo Approximation Error



STOCHASTIC EXAMPLE: Smolyak Approximation Error



CONCLUSION: Software

Sparse grid software available online:

- **SPINTERP** (in MATLAB, by Andreas Klimke) does interpolation, quadrature, optimization, based on data values on a sparse grid; ACM TOMS Algorithm 847.
- **SMOLPACK** (in C, by Knut Petras) quadrature using sparse grids.
- **SPARSE_GRID_HW** (in MATLAB, by Florian Heiss and Viktor Winschel) sparse grid quadrature rules based on Gauss-Legendre, Gauss-Hermite, Gauss-Patterson, or a nested variation of Gauss-Hermite rules.

CONCLUSION: Software

I have written C++, F90 and MATLAB versions of libraries and programs for sparse grid integration using specific 1D rules:

- **sparse_grid_cc** Clenshaw Curtis;
- **sparse_grid_gl** Gauss-Legendre;
- **sparse_grid_hermite** Hermite;
- **sparse_grid_laguerre** Laguerre;
- **sparse_grid_mixed** Mixture of 1D rules;
- **sparse_grid_mixed_growth** Mixture + growth;
- **sgmga** mixture + growth + anisotropic;

For example, look at
http://people.sc.fsu.edu/~jburkardt/m_src/sparse_grid_cc/sparse_grid_cc.html

CONCLUSION: A few observations

Sparse grids are based on combinations of product rules.

The combinations seek specific **precision** levels.

For integrands with bounded derivatives, precision produces **accuracy**.

By discarding some of the unneeded precision of product rules, sparse grids have a higher **efficiency**.

Abstract probability integrals, stochastic collocation and polynomial chaos expansions are examples of settings in which sparse grids may be useful.

CONCLUSION: A few observations

The underlying 1D quadrature rules could just as well be Jacobi, Laguerre, Hermite or their generalizations.

We can choose different quadrature rules for each dimension.

The rule family for a particular dimension could be a **piecewise** polynomial or some kind of **composite** rule. This makes it possible to handle functions with limited differentiability.

CONCLUSION: A few observations

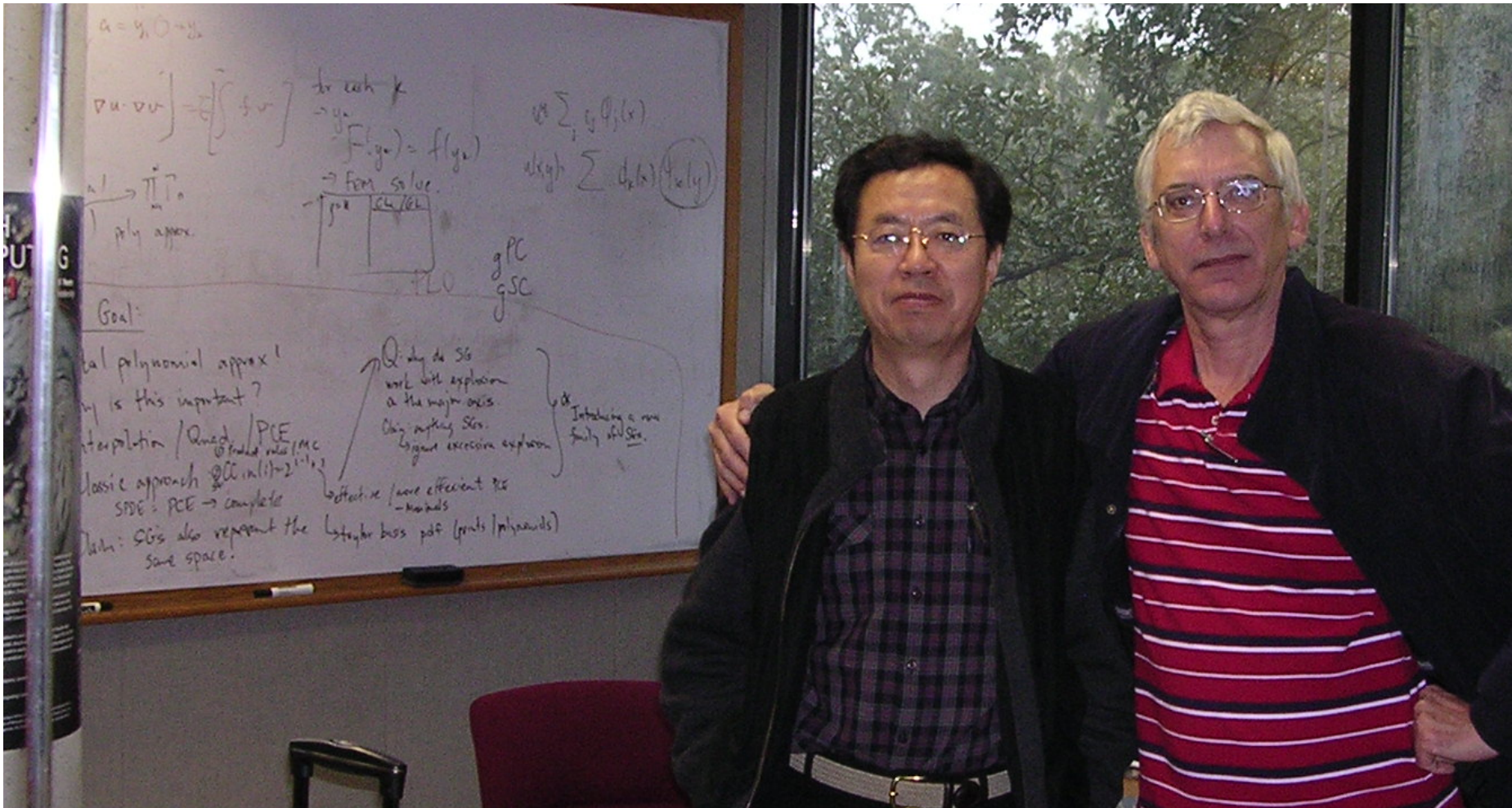
The approach we have outline here is **isotropic**. It treats each spatial dimension with the same degree of importance.

But many very high dimensional problems that people work on are solvable in part because most of the dimensions have very little variation.

An **anisotropic** sparse grid rule can be developed which chooses the maximum order in each dimension based on weights.

These weights can be supplied in advance by the user, or determined adaptively.

CONCLUSION: Thanks to my Host!



CONCLUSION: References

Volker **Barthelmann**, Erich **Novak**, Klaus **Ritter**,
High Dimensional Polynomial Interpolation on Sparse Grids,
Advances in Computational Mathematics, 12(4) 2000, p273-288.

John **Burkardt**, Max **Gunzburger**, Clayton **Webster**,
Reduced Order Modeling of Some Nonlinear Stochastic PDE's,
IJNAM, 4(3-4) 2007, p368-391.

Thomas **Gerstner**, Michael **Griebel**,
Numerical Integration Using Sparse Grids,
Numerical Algorithms, 18(3-4), 1998, p209-232.

Andreas **Klimke**, Barbara **Wohlmuth**,
Algorithm 847: SPINTERP: Piecewise Multilinear Hierarchical Sparse Grid Interpolation in MATLAB,
ACM Transactions on Mathematical Software, 31,(4), 2005, p561-579.

CONCLUSION: References

Sergey **Smolyak**,
Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions,
Doklady Akademii Nauk SSSR, 4, 1963, p240-243.