Accuracy, Precision and Efficiency in Sparse Grids

John Burkardt,
Information Technology Department,
Virginia Tech.
....
Computer Science Research Institute,
Sandia National Laboratory,
23 July 2009
....
https://people.sc.fsu.edu/∼jburkardt/presentations/
sparse_2009_sandia.pdf

1. **Accuracy, Precision, Efficiency**
2. Families of Quadrature Rules in 1D
3. Product Rules for Higher Dimensions
4. Smolyak Quadrature
5. Covering Pascal's Triangle
6. How Grids Combine
7. Sparse Grids in Action
8. Smoothness is Necessary
9. A Stochastic Diffusion Equation
10. Conclusion

A+P+E: Accuracy, Precision, Efficiency

In this talk, we consider the problem of constructing interpolatory quadrature rules for high dimensional regions.

For smooth integrands, rule **precision** implies **accuracy**.

But the natural method of creating precise rules, using products, incurs a cost that is exponential in the spatial dimension.

We show that this explosion is not a necessary feature of interpolation, and we investigate **efficient** methods of achieving precision, and hence accuracy, for smooth integrands in high dimensional spaces.

A+P+E: Precision

If a quadrature rule is exact when applied to any polynomials of degree **P** or less, the rule has **precision P**.

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

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

High precision is a property of interpolatory and Gauss rules.

In multi-dimensional case, precision is defined with respect to all polynomials whose *total degree* is **P** or less.

A+P+E: Precision Can Mean Accuracy

Using a rule with **P = N** on a smooth function, low order terms get integrated, leaving an error that is $O(\frac{1}{N}^{N+1})$,

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

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

However, keep in mind that precision:

- *is not necessary* - after all, Monte Carlo rules work.
- *is not a guarantee* - Newton Cotes rules are unstable;
- *can be harmful* - f(x) = step or piecewise or singular!

A+P+E: Accuracy is the Goal

If we have a particular integrand in mind, **accuracy** simply measures the error in our estimate of the integral.

Often it is possible to make a general accuracy estimate for an entire class of integrands. In the most common case, we may be able to show that for integrands that are sufficiently smooth, the error is a function of h whose leading term is dominates as $h \to 0$.

In such a case, we speak of the **asymptotic accuracy** of the rule for the family of integrands.

For this talk, we will usually mean *asymptotic accuracy* when we say "accuracy".

A+P+E: Efficiency is the Number of Abscissas

Efficiency measures the amount of work expended for the result.

For quadrature, we measure our work in terms of the number of function evaluations, which in turn is the number of abscissas.

Since it is common to use a sequence of rules, it is important, for efficiency, to take advantage of nestedness, that is, to choose a family of rule for which the function values at one level can be reused on the next.

1. Accuracy, Precision, Efficiency
2. **Families of Quadrature Rules in 1D**
3. Product Rules for Higher Dimensions
4. Smolyak Quadrature
5. Covering Pascal's Triangle
6. How Grids Combine
7. Sparse Grids in Action
8. Smoothness is Necessary
9. A Stochastic Diffusion Equation
10. Conclusion

QUAD1D: Approximation of Integrals

Integrals are numerically approximated by **quadrature rules**.

In 1D, this is a “mature” (dead?) area.

$$\int_{\Omega} f(x)\,dx \quad \approx \quad \sum_{i=1}^{N} w_i\, f(x_i)$$

- Interpolatory rules: Newton-Cotes, Chebyshev points;
- Semi-interpolatory rules: Gauss rules;
- Sampling rules: Monte Carlo and Quasi-Monte Carlo;
- Transform rules: tanh, tanh-sinh, erf rules.

QUAD1D: Families of Rules

Most quadrature rules are available in any order **N**.

Generally, increasing **N** produces a more accurate result
(more about this in a minute!)

A calculation desiring a specific level of accuracy must be able to generate elements of a family of quadrature rules of increasing order.

An efficient calculation may seek a family of rules in which some or all abscissas are reused. This is called **nesting**.

QUAD1D: Order, Level, Growth Rule

The **order** of a rule, **N**, is the number of points or abscissas.

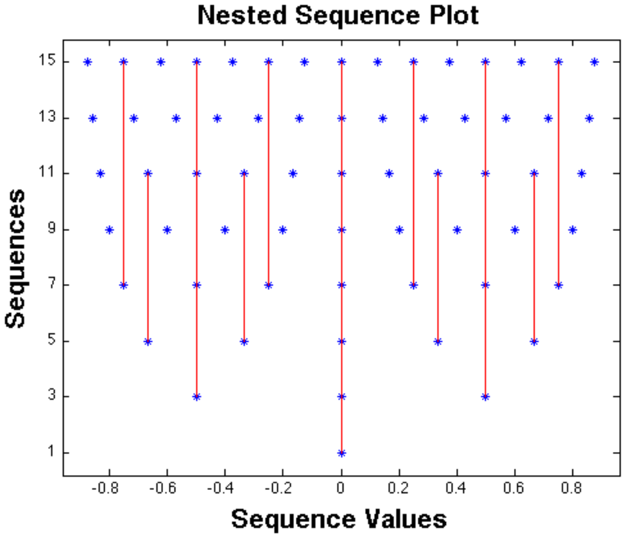
The **level** of a rule, **L**, is its index in a family.

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

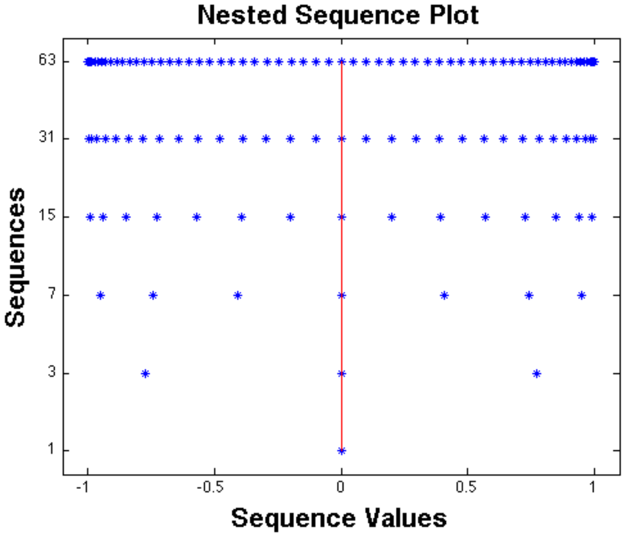
The **growth rule** for a family relates level **L** and order **N**:

$$
\begin{aligned}
N =& 2L + 1 \quad \text{Linear} \\
N =& 2^L + 1 \quad \text{Exponential, Clenshaw} - \text{Curtis} \\
N =& 2^{L+1} - 1 \quad \text{Exponential, Gauss} - \text{Legendre}
\end{aligned}
$$

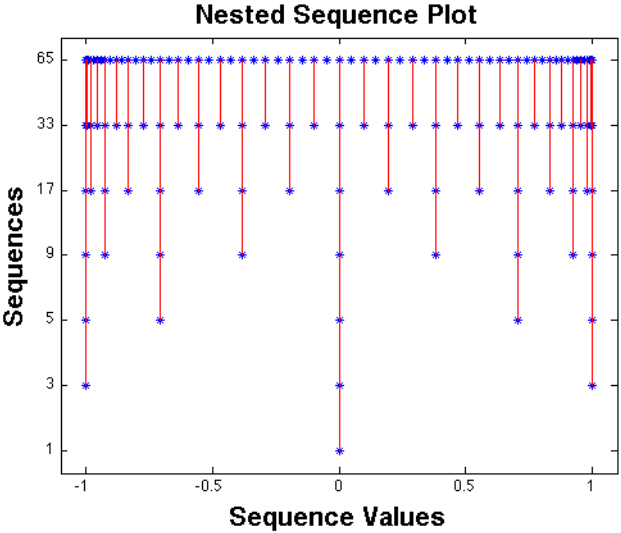
QUAD1D: Newton Cotes Open, Slightly Nested, Linear Growth



QUAD1D: Gauss Legendre, Weakly Nested, Exponential Growth



QUAD1D: Clenshaw Curtis, Nested, Exponential Growth



Accuracy, Precision and Efficiency in Sparse Grids

1. Accuracy, Precision, Efficiency
2. Families of Quadrature Rules in 1D
3. **Product Rules for Higher Dimensions**
4. Smolyak Quadrature
5. Covering Pascal's Triangle
6. How Grids Combine
7. Sparse Grids in Action
8. Smoothness is Necessary
9. A Stochastic Diffusion Equation
10. Conclusion

PRODUCT RULES: Formed from 1D Rules

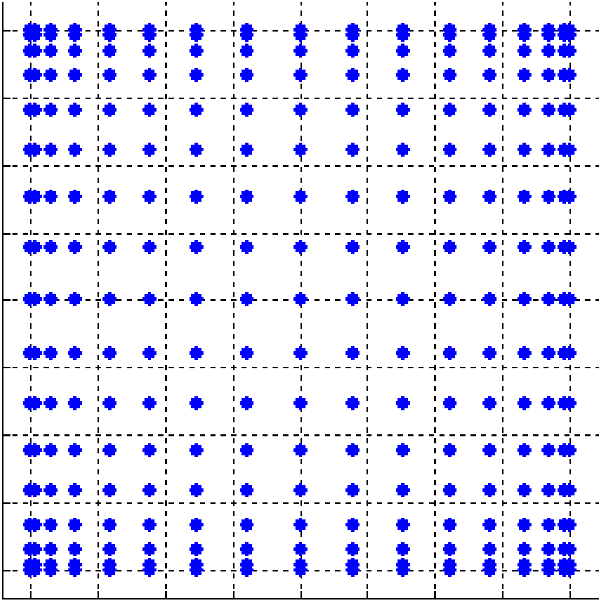
Let $\mathcal{Q}_L$ be the L-th member of a family of nested quadrature rules, with order N_L and precision P_L.

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.

Everything extends to the general M-dimensional case... except that the order growth N_L^M is unacceptable!

PRODUCT RULES: 17x17 Clenshaw-Curtis



PRODUCT RULES: 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**!.

PRODUCT RULES: Pascal's Precision Triangle

We can seek accuracy by requiring that our quadrature rule have a given precision. To say our quadrature rule has precision 5 is to say that it can correctly integrate every polynomial of degree 5 or less. This corresponds to integrating all the monomials below the 5-th diagonal in a sort of Pascal's triangle.

A given rule may integrate some monomials above its highest diagonal; but these extra monomials don't improve the overall asymptotic accuracy of the rule.

PRODUCT RULES: 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

PRODUCT RULES: 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

PRODUCT RULES: 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%

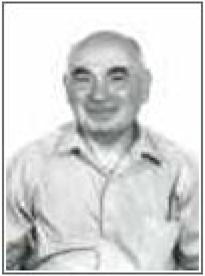
In 5D, there are only 2,002 items to search for.
Can't we find a quadrature rule of roughly that order?

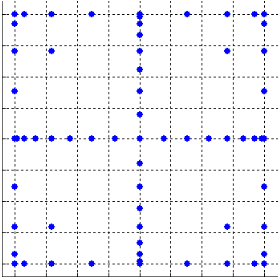
Accuracy, Precision and Efficiency in Sparse Grids

1. Accuracy, Precision, Efficiency
2. Families of Quadrature Rules in 1D
3. Product Rules for Higher Dimensions
4. **Smolyak Quadrature**
5. Covering Pascal's Triangle
6. How Grids Combine
7. Sparse Grids in Action
8. Smoothness is Necessary
9. A Stochastic Diffusion Equation
10. Conclusion

SMOLYAK QUADRATURE

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





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

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

SMOLYAK QUADRATURE: Construction

We have 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.

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

Thus, the rule $\mathcal{A}(L,M)$ is a weighted sum of product rules.

SMOLYAK QUADRATURE: A sum of rules/a rule of sums

The Smolyak construction rule can be interpreted to say:

Compute the integral estimate for each rule,
then compute the algebraic sum of these estimates.

but it can also be interpreted as:

Combine the component rules into a single quadrature rule,
the new abscissas are the set of the component abscissas;
the new weights are the component weights multiplied by the sparse grid coefficient.

SMOLYAK QUADRATURE: Efficiency from Nesting

Under the second interpretation, we can see that in cases where an abscissa is duplicated in the component rules, the combined rule can use a single copy of the abscissa, with the sum of the weights associated with the duplicates.

Duplication is a property inherited from the 1D rules.

Duplication is useful when computing a single sparse grid rule, but also when computing a sequence of sparse grids of increasing level. In some cases, all the values from the previous level can be reused.

SMOLYAK QUADRATURE: Using Clenshaw-Curtis

A common choice is 1D Clenshaw-Curtis rules.

We can make a nested family by choosing successive orders of 1, 3, 5, 9, 17, ...

We wrote $\mathcal{Q}^i$ to indicate the 1D quadrature rules indexed by a **level** running 0, 1, 2, 3, and so on.

We will use a plain Q_n to mean the 1D quadrature rules of **order** 1, 3, 5, 9 and so on.

We will find it helpful to count abscissas.

SMOLYAK QUADRATURE: Using Clenshaw-Curtis

Theorem

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

Thus, although our construction of sparse grids seems complicated, we still know the level of precision we can expect at each level.

SMOLYAK QUADRATURE: Precision

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

Recall 5D product rule required 100,000 abscissas to integrate 2,002 entries in Pascal's precision triangle (precision 9).

SMOLYAK QUADRATURE: Asymptotic Accuracy

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

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

The accuracy for a Smolyak rule based on a nested 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 QUADRATURE: Efficiency

The space of **M**-dimensional polynomials of degree **P** or less has dimension $\begin{pmatrix} P+M \\ M \end{pmatrix} \approx \frac{M^P}{P!}$.

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**).

And, of course, notice there is no exponent of **M** in the point growth.

Accuracy, Precision and Efficiency in Sparse Grids

1. Accuracy, Precision, Efficiency
2. Families of Quadrature Rules in 1D
3. Product Rules for Higher Dimensions
4. Smolyak Quadrature
5. **Covering Pascal's Triangle**
6. How Grids Combine
7. Sparse Grids in Action
8. Smoothness is Necessary
9. A Stochastic Diffusion Equation
10. Conclusion

COVERING PASCAL'S TRIANGLE

A family of precise interpolatory rules must cover successive rows of Pascal's precision triangle in a regular way.

In higher dimensions, the triangle is a tetrahedron or a simplex.

The product rule does this by "overkill".

Smolyak's construction covers the rows, but does so much more economically, using lower order product rules.

COVERING PASCAL'S TRIANGLE

Let's watch how this works for a family of 2D rules.

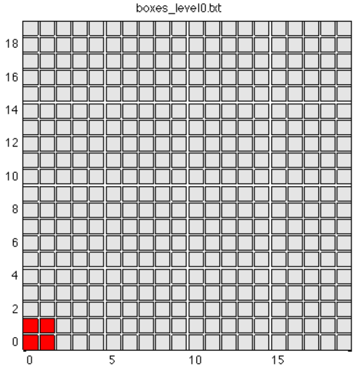
I've had to turn Pascal's triangle sideways, to an XY grid. If we count from 0, then box (I,J) represents $x^i y^j$.

Thus a row of Pascal's triangle is now a **diagonal** of this plot.

The important thing to notice is the maximum diagonal that is completely covered. This is the precision of the rule.

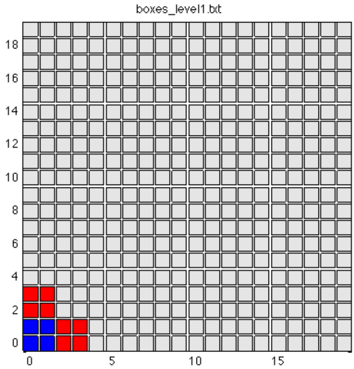
We will see levels 0 through 4 and expect precisions 1 through 11 by 2's.

COVERING PASCAL'S TRIANGLE: 2D Level 0



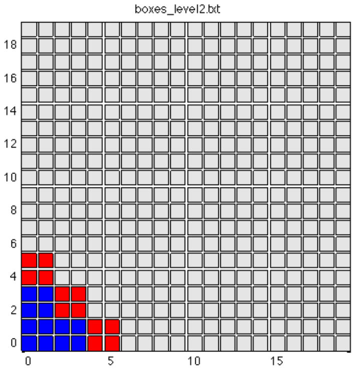
$Q_1 \otimes Q_1$

COVERING PASCAL'S TRIANGLE: 2D Level 1



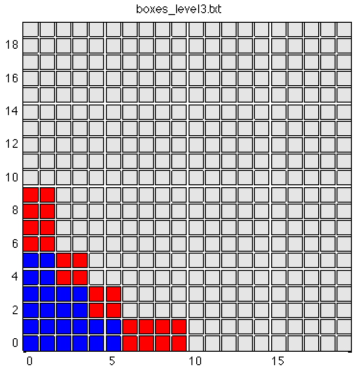
$Q_3 \otimes Q_1 + Q_1 \otimes Q_3$ - old

COVERING PASCAL'S TRIANGLE: 2D Level 2



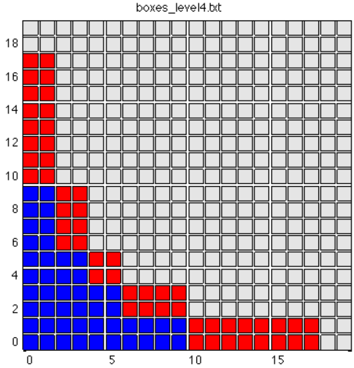
$Q_5 \otimes Q_1 + Q_3 \otimes Q_3 + Q_1 \otimes Q_5$ - old.

COVERING PASCAL'S TRIANGLE: 2D Level 3



$Q_9 \otimes Q_1 + Q_5 \otimes Q_3 + Q_3 \otimes Q_5 + Q_1 \otimes Q_9$ - old;

COVERING PASCAL'S TRIANGLE: 2D Level 4



$Q_{17} \otimes Q_1 + Q_9 \otimes Q_3 + Q_5 \otimes Q_5 + Q_3 \otimes Q_9 + Q_1 \otimes Q_{17}$ - old;

COVERING PASCAL'S TRIANGLE: 2D Level 4

When based on an exponential growth rule like $N = 2^L + 1$, each new level of a Smolyak family:

- covers 2 more diagonals, needed for precision;
- avoids filling in the heavy "half" of the hypercube that the product rule fills;
- adds long but thin regions of excess accuracy along the axes;

Accuracy, Precision and Efficiency in Sparse Grids

1. Accuracy, Precision, Efficiency
2. Families of Quadrature Rules in 1D
3. Product Rules for Higher Dimensions
4. Smolyak Quadrature
5. Covering Pascal's Triangle
6. **How Grids Combine**
7. Sparse Grids in Action
8. Smoothness is Necessary
9. A Stochastic Diffusion Equation
10. Conclusion

HOW GRIDS COMBINE

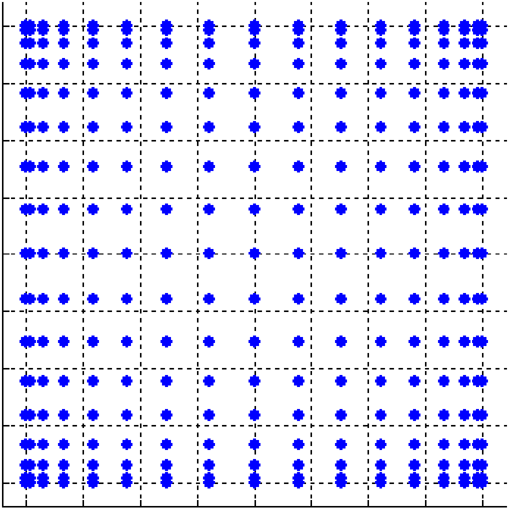
We said that the Smolyak construction combines low order product rules, and that the result can be regarded as a single rule.

Let's look at the construction of the Smolyak grid of level **L**=4 and hence precision **P**=7 in 2D.

Our construction will involve 1D rules of orders 1, 3, 5, 9 and 17, and product rules formed of these factors.

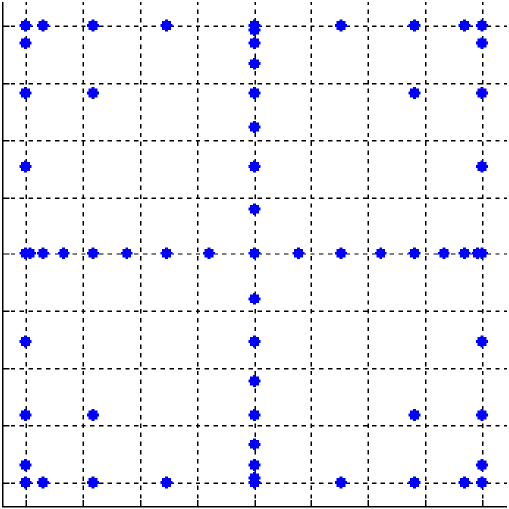
Because of nesting, every product rule we form will be a subset of the 17x17 full product grid.

HOW GRIDS COMBINE: 2D Order 17 Product Rule



A 17x17 Clenshaw-Curtis product grid (289 points).

HOW GRIDS COMBINE: 2D Level4 Smolyak Grid

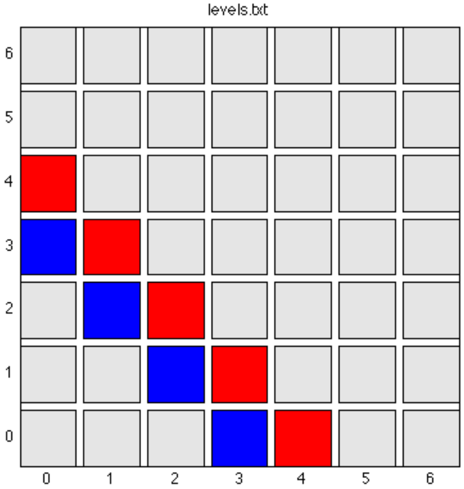


The sparse grid is a subset of the 17x17 product grid (65 points).

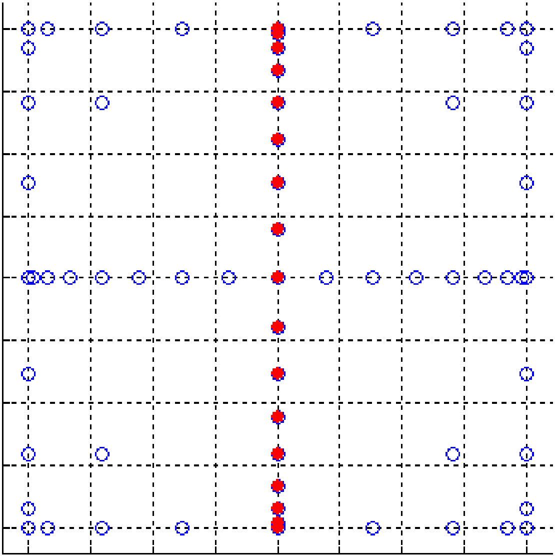
HOW GRIDS COMBINE

$$
\begin{aligned}
\mathcal{A}(4,2) &= \sum_{3 \le |\mathrm{i}| \le 4} (-1)^{4-|\mathrm{i}|} \begin{pmatrix} -1 \\ 4-|\mathrm{i}| \end{pmatrix} (\mathcal{Q}^{i_1} \otimes \mathcal{Q}^{i_2}) \\
&= + \mathcal{Q}^0 \otimes \mathcal{Q}^4 \quad (Q_1 \otimes Q_{17}) \\
&\quad + \mathcal{Q}^1 \otimes \mathcal{Q}^3 \quad (Q_3 \otimes Q_9) \\
&\quad + \mathcal{Q}^2 \otimes \mathcal{Q}^2 \quad (Q_5 \otimes Q_5) \\
&\quad + \mathcal{Q}^3 \otimes \mathcal{Q}^1 \quad (Q_9 \otimes Q_3) \\
&\quad + \mathcal{Q}^4 \otimes \mathcal{Q}^0 \quad (Q_{17} \otimes Q_1) \\
&\quad - \mathcal{Q}^0 \otimes \mathcal{Q}^3 \quad (Q_1 \otimes Q_9) \\
&\quad - \mathcal{Q}^1 \otimes \mathcal{Q}^2 \quad (Q_3 \otimes Q_5) \\
&\quad - \mathcal{Q}^2 \otimes \mathcal{Q}^1 \quad (Q_5 \otimes Q_3) \\
&\quad - \mathcal{Q}^3 \otimes \mathcal{Q}^0 \quad (Q_9 \otimes Q_1)
\end{aligned}
$$

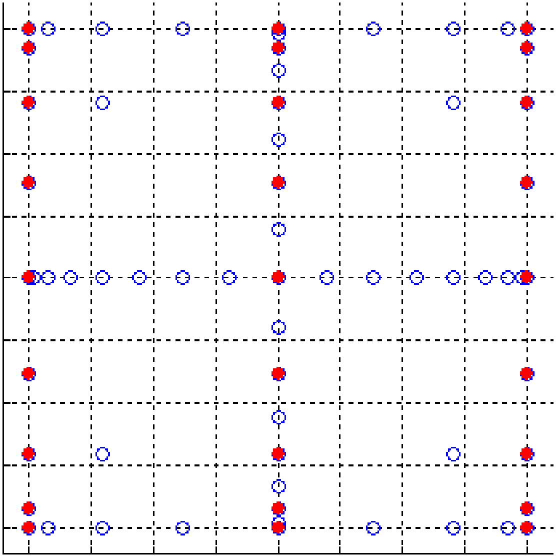
HOW GRIDS COMBINE: Red Rules - Blue Rules



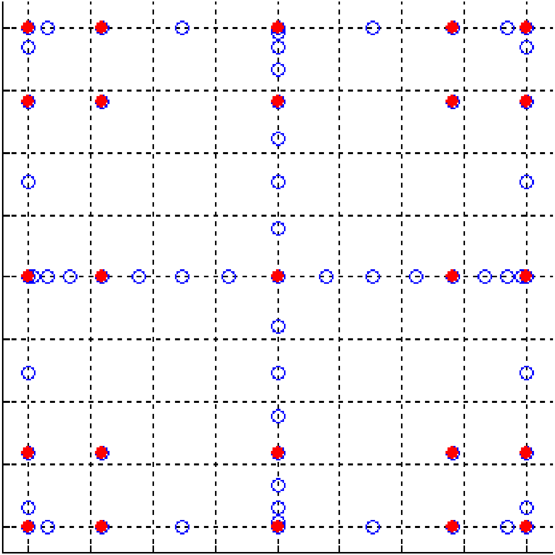
HOW GRIDS COMBINE: 2D Level4 1x17 component



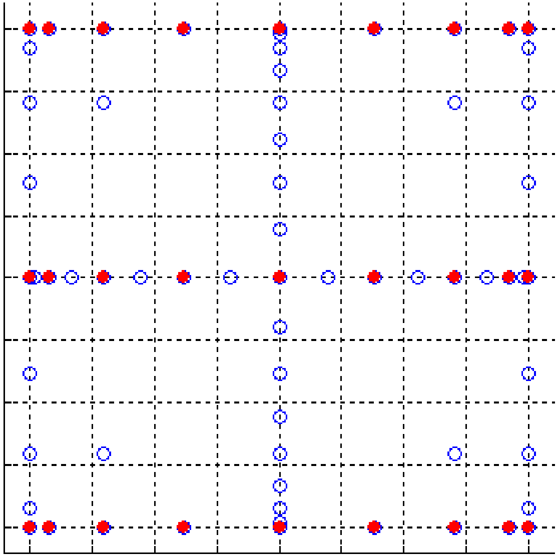
HOW GRIDS COMBINE: 2D Level4 3x9 component



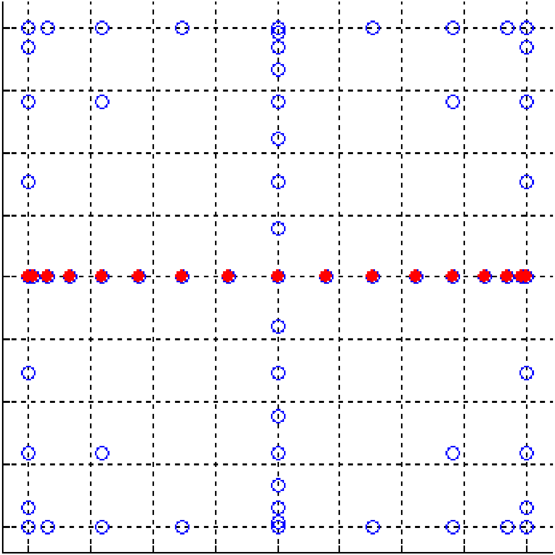
HOW GRIDS COMBINE: 2D Level4 5x5 component



HOW GRIDS COMBINE: 2D Level4 9x3 component



HOW GRIDS COMBINE: 2D Level4 17x1 component



HOW GRIDS COMBINE: Red Rules - Blue Rules

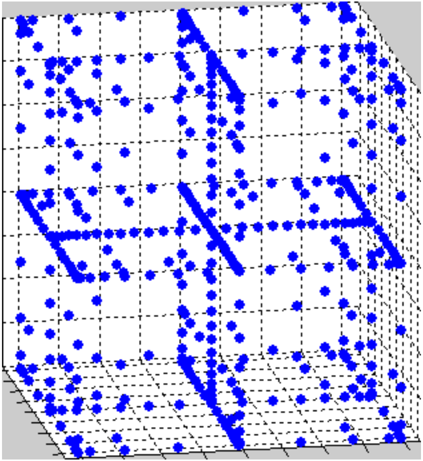
We've shown the component "red" rules, which show up in the sum with a positive sign.

The 'blue" rules are similar, though at a lower level:

- The first of the 5 red rules has order 1x17;
- The first of the 4 blue rules has order 1x9.

Notice that this rule is "symmetric" in all dimensions. If we have a 65x3 rule, we are also guaranteed a 3x65 rule. The Smolyak formula is **isotropic**.

HOW GRIDS COMBINE: 3D Level5 Smolyak Grid



3D sparse grid, level 5, precision 11 uses 441 abscissas;

3D product grid of precision 11 uses 1,331 abscissas.

Accuracy, Precision and Efficiency in Sparse Grids

1. Accuracy, Precision, Efficiency
2. Families of Quadrature Rules in 1D
3. Product Rules for Higher Dimensions
4. Smolyak Quadrature
5. Covering Pascal's Triangle
6. How Grids Combine
7. **Sparse Grids in Action**
8. Smoothness is Necessary
9. A Stochastic Diffusion Equation
10. Conclusion

SPARSE GRIDS IN ACTION

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**:

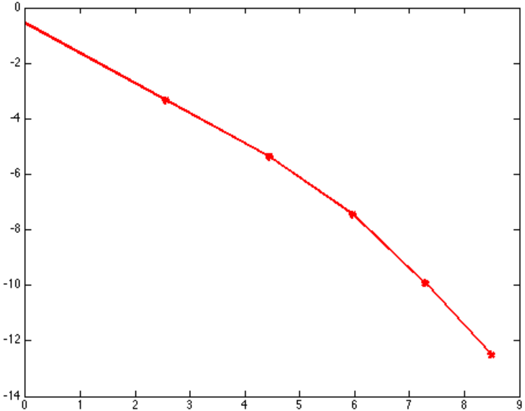
$$f(X) = \frac{1}{\prod_{i=1}^{M}(C_i^2 + (X_i - Z_i)^2)}$$

where C_i and Z_i are prescribed.

SPARSE GRIDS IN ACTION: 6D Smolyak

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

SPARSE GRIDS IN ACTION:6D Smolyak



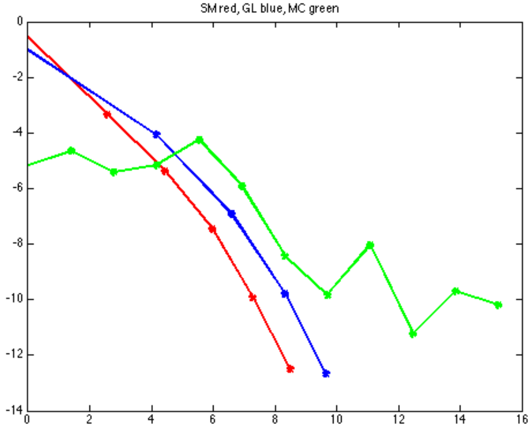
SPARSE GRIDS IN ACTION: 6D Gauss-Legendre

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

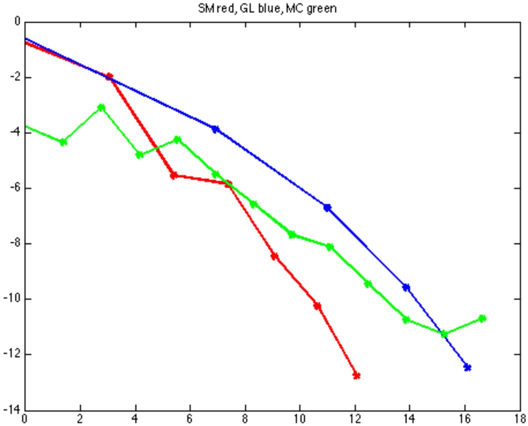
SPARSE GRIDS IN ACTION: 6D Monte Carlo

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

SPARSE GRIDS IN ACTION: 6D Smolyak/GL/MC



SPARSE GRIDS IN ACTION: 10D Smolyak/GL/MC



SPARSE GRIDS IN ACTION: Thoughts

The graphs suggests that the accuracy behavior of the sparse grid rule is similar to the Gauss-Legendre rule, at least for this kind of integrand.

For 6 dimensions, the sparse grid rule is roughly 3 times as efficient as Gauss-Legendre, (4,865 abscissas versus 15,625 abscissas).

Moving from 6 to 10 dimensions, the efficiency advantage is 60: (170,000 abscissas versus 9,700,000 abscissas).

The Gauss-Legendre product rule is beginning the explosive growth in abscissa count.

Accuracy, Precision and Efficiency in Sparse Grids

1. Accuracy, Precision, Efficiency
2. Families of Quadrature Rules in 1D
3. Product Rules for Higher Dimensions
4. Smolyak Quadrature
5. Covering Pascal's Triangle
6. How Grids Combine
7. Sparse Grids in Action
8. **Smoothness is Necessary**
9. A Stochastic Diffusion Equation
10. Conclusion

Smoothness: A Few Words of Wisdom

A sparse grid approach is the right choice when the function to be integrated is known to be smooth or to have bounded derivatives up to the order of the rule we are applying.

In those cases, the precision of a sparse grid extracts extra information from the function values, to provide accurate answers with efficiency.

But if the smoothness assumption is not true, the sparse grid approach will fail.

Smoothness: Characteristic Function of 6D Sphere

In the region $[-1,+1]^6$, define

$$f(x) = \begin{cases} 1, & \text{if } \|x\| \leq 1; \\ 0, & \text{if } \|x\| > 1. \end{cases}$$

This function is not even continuous, let alone differentiable. We will try to apply a series of Clenshaw Curtis sparse grids to this integrand.

The hypercube volume is 64;
the hypersphere volume is $\frac{\pi^3}{6} \approx 5.16771$.

Smoothness: Sparse Grid Quadrature

N	SG Estimate	SG Error	:	MC Estimate	MC Error
1	4.000	1.167	:	...	...
13	64.000	58.832	:	...	...
85	-42.667	-47.834	:	...	...
389	-118.519	-123.686	:	...	...
1457	148.250	143.082	:	...	...
4865	-24.682	-29.850	:	...	...

Can you see why negative estimates are possible even though the integrand is never negative?

Smoothness: MC Quadrature

N	SG Estimate	SG Error	:	MC Estimate	MC Error
1	4.000	1.167	:	0.00000	5.16771
13	64.000	58.832	:	0.00000	5.16771
85	-42.667	-47.834	:	3.01176	2.15595
389	-118.519	-123.686	:	4.77121	0.39650
1457	148.250	143.082	:	5.16771	0.01555
4865	-24.682	-29.850	:	5.41994	0.25226

Here, we make the Monte Carlo method look like a quadrature rule with equal weights.

Smoothness: MC Quadrature

So how far do we have to go to get 3 digits correct?

N	MC Estimate	MC Error
1	0.00000	5.16771
32	6.00000	0.83228
1,024	4.81250	0.35521
32,768	5.39063	0.22291
1,048,576	5.18042	0.01271
33,554,432	5.16849	0.00077
∞	5.16771	0.00000

The function values are only 0 or 1
the spatial dimension is “only” 6D...

...but 3 digit accuracy requires 33 million evaluations!

Smoothness: Use Sparse Composite Rule

In fact, a sparse grid **could** be used for this problem, as long as it used rules that never required more smoothness than the integrand has.

Since the integrand doesn't have any derivates, we can get by with a composite rule made up of one point rules.

The ability to detect and react to this kind of nonsmoothness is a proposed area of future work.

Accuracy, Precision and Efficiency in Sparse Grids

1. Accuracy, Precision, Efficiency
2. Families of Quadrature Rules in 1D
3. Product Rules for Higher Dimensions
4. Smolyak Quadrature
5. Covering Pascal's Triangle
6. How Grids Combine
7. Sparse Grids in Action
8. Smoothness is Necessary
9. **A Stochastic Diffusion Equation**
10. Conclusion

Stochastic Diffusion

$$-\nabla \cdot (a(x,y)\, \nabla u(x,y)) = f(x,y)$$

u is an unknown quantity, like temperature;

a 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 Diffusion: Uncertain Conductivity

Using a fixed value for $a(x, y)$ might be unrealistic.

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

We might think of $a(x, y)$ as a *random field* $a(\omega; x, y)$.

The ω represents the unknown variation from the average.

Stochastic Diffusion: Uncertain Solution

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

$$-\nabla \cdot (a(\omega; x, y) \nabla u(\omega; x, y)) = f(x, y)$$

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

Can we still extract information from the equation?

Stochastic Diffusion: 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.

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

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

Stochastic Diffusion: Approximate Integral

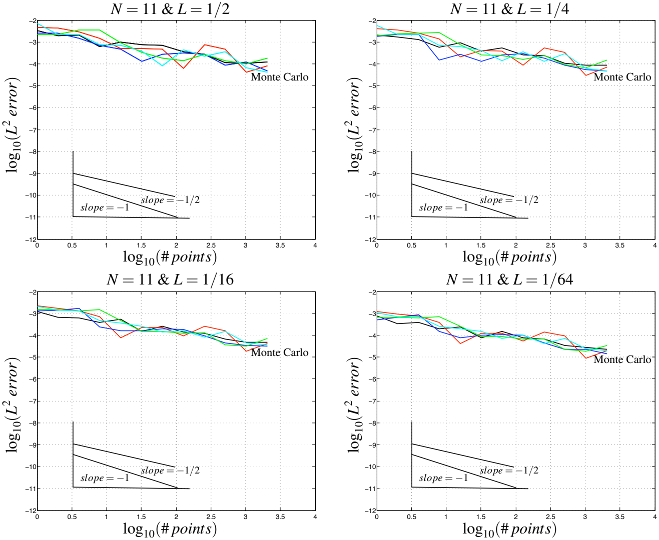
We approximate the function space Ω by an M-dimensional space Ω_M, of linear sums of perturbations ω_M.

We now estimate the integral of $u(\omega_M; x, y)$ in Ω_M.

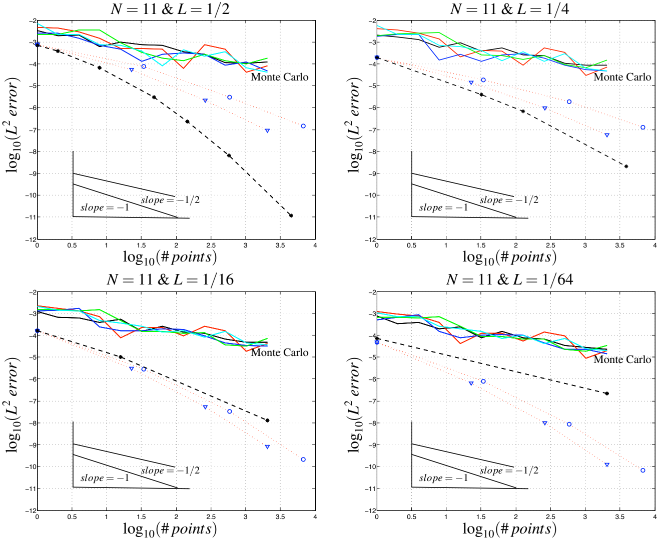
Monte Carlo: select a random set of parameters ω_M, solve for u, multiply by the probability, and average.

Sparse grid: choose a level, defining a grid of ω_M values in Ω_M, solve for each u, multiply by the probability, and take a weighted average.

Stochastic Diffusion: Monte Carlo



Stochastic Diffusion: Smolyak



Accuracy, Precision and Efficiency in Sparse Grids

1. Accuracy, Precision, Efficiency
2. Families of Quadrature Rules in 1D
3. Product Rules for Higher Dimensions
4. Smolyak Quadrature
5. Covering Pascal's Triangle
6. How Grids Combine
7. Sparse Grids in Action
8. Smoothness is Necessary
9. A Stochastic Diffusion Equation
10. **Conclusion**

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 **composite** rule. This approach would fix the characteristic function of the 6D sphere.

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: 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.

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