Sparse Grids: Mixed Families, Growth Rules, Anisotropy

John Burkardt
Virginia Tech

....

http://people.sc.fsu.edu/~jburkardt/presentations/icms_2010.pdf

....

Funded in part by a grant from Sandia National Laboratories.

28 May 2010

The Whole Talk in One Page

I am involved in a software project developing sparse grid codes which allow the user to choose, for each spatial component, the quadrature family, rate of growth, and anisotropy weight.

The software library is called **SGMGA**:

1. **S**parse **G**rids
2. **M**ixed families
3. **G**rowth rules
4. **A**nisotropic weighting

Nested Families of Quadrature Rules

I will consider sparse grids as used for high dimensional quadrature.

The **cost** of a quadrature rule is measured in the number of function evaluations,

If we are considering smooth integrands, the **benefit** may be described in terms of the degree of precision: the rule can exactly integrate all monomials of that degree or less.

We seek families of rules, indexed by increasing precision.

As the spatial dimension increases, we want to ensure that the cost of a given precision does not grow exponentially.

Monomials of Total Degree 5 or Less

A reasonable goal for a family of quadrature rules is that each member exactly integrate all monomials up to a given total order. In 2D, a rule that could integrate up to 5th order would need to capture the monomials shown in blue:

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

A Product Rule Overshoots the Goal

The cost of a product rule of precision P in dimension M is P^M. In 2D, we get a square of precision instead of a triangle. In higher dimensions, most of the cost of a product rule is used to integrate "red" monomials that don't actually improve the achieved precision. (They exponentially outweigh the blue monomials).

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

Compare Product and Smolyak Precisions

Smolyak's sparse grid procedure combines lower order product rules to more carefully achieve the desired precision level.

If we can afford 1,000,000 evaluations in M-space, a non-Gaussian product rule has precision roughly $1,000,000^{(1/M)}$.

M	P(Product Rule)	P(Sparse Grid)
10	4.0	15
20	2.0	9 or 11
30	1.6	9 or 11
50	1.3	7 or 9
100	1.1	7

After dimension 20, you can't afford a 2-point product rule!

Construction of an isotropic Sparse Grid

The level **L** sparse grid in **M** dimensions is formed from low order product rules, each defined by a level vector $\mathbf{i} = i_1 + \cdots + i_M$, where i_j is the "level" or index of the j-th 1D rule.

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

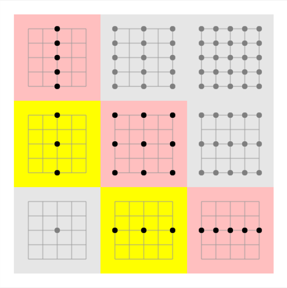
While the individual product rules may be anisotropic, the indexing on the sum allows the resulting grid to be isotropic.

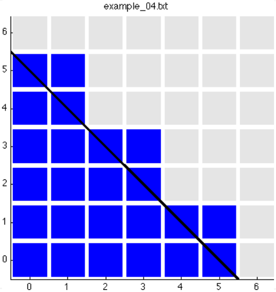
This formula leaves us free to choose the domains, weights, families and growth rates of the 1D factors $\mathcal{Q}^i$.

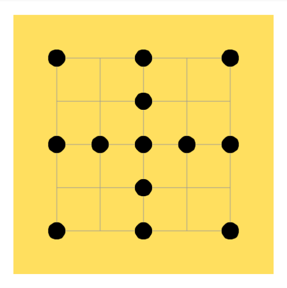
VirginiaTech

Sparse Grid = Sum of Selected Product Rules

Combination Precision Grid

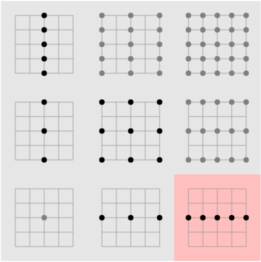


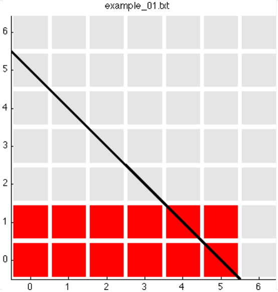


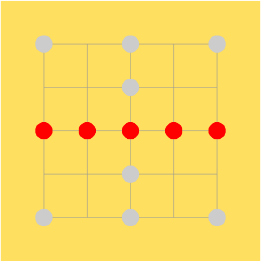


Add 5x1 Rule:

Combination Precision Grid

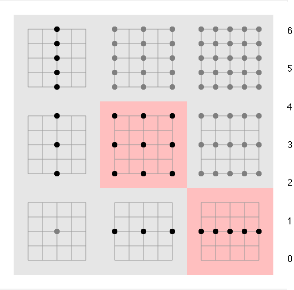


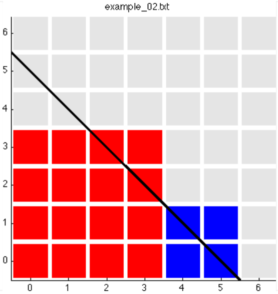


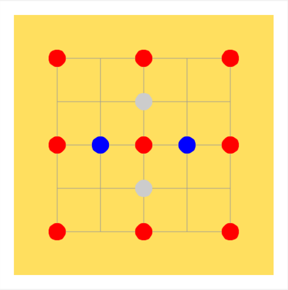


Add 3x3 Rule:

Combination   Precision   Grid





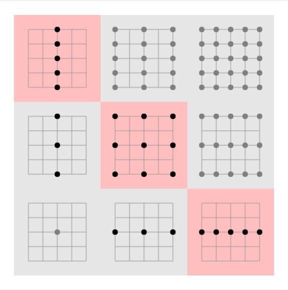


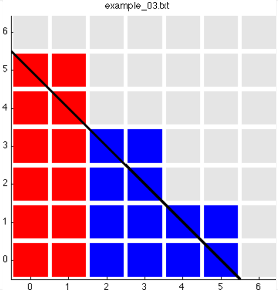
Add 1x5 Rule:

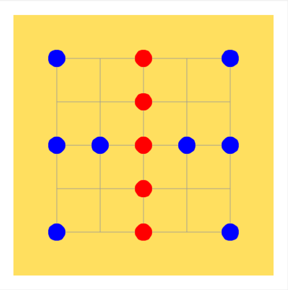
Combination

Precision

Grid







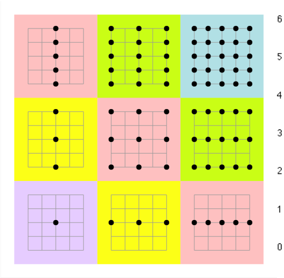
The Results

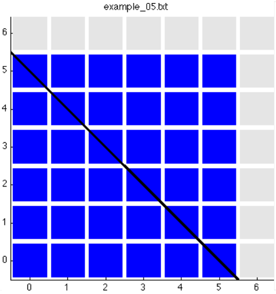
To complete this 2D sparse grid, we include contributions from the two lower order rules, 3x1 and 1x3. Here, we are using a nested family of rules, so the resulting grid does not change, although these lower order rules do affect the computed weights applied to the grid points.

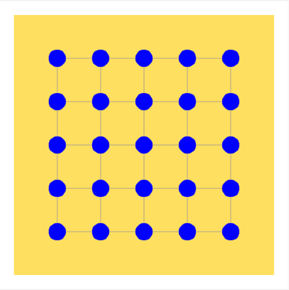
From the precision plot, we can see that the sparse grid claims to be precise for all monomials of total degree 5. It has achieved this precision goal using fewer points than a simple 5x5 product rule.

Product Grid:

Combination Precision Grid







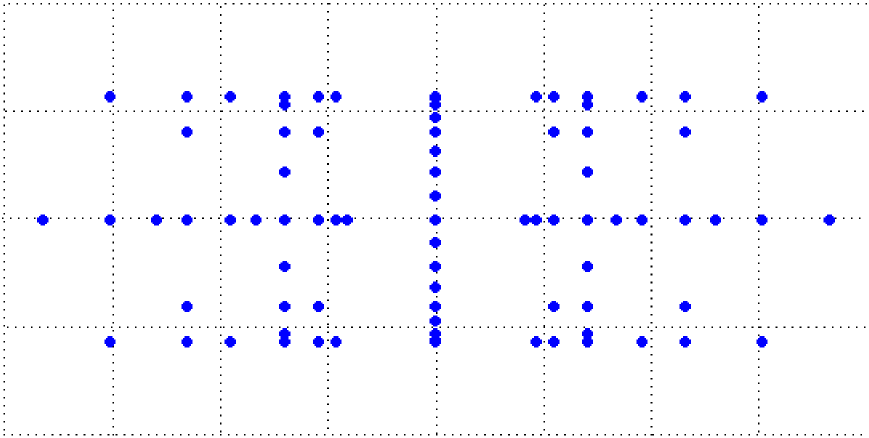
Mixed Families

SGMGA allows each dimension to choose its quadrature family, including: Clenshaw-Curtis, Gauss-Patterson, Legendre, generalized Hermite, generalized Laguerre, Jacobi or a user-created family.

A parameterized family can have different parameters in each dimension.

This makes it possible to correctly treat cases involving a mixture of uniform and normally distributed quantities, for instance, or cases in which a single distribution is used, but the parameters defining that distribution vary from one dimension to the next.

Gauss-Hermite X Clenshaw Curtis, Level 4



Growth-Rules

The indexing of the 1D quadrature family defines the growth rule. For Gauss Patterson family this is typically:

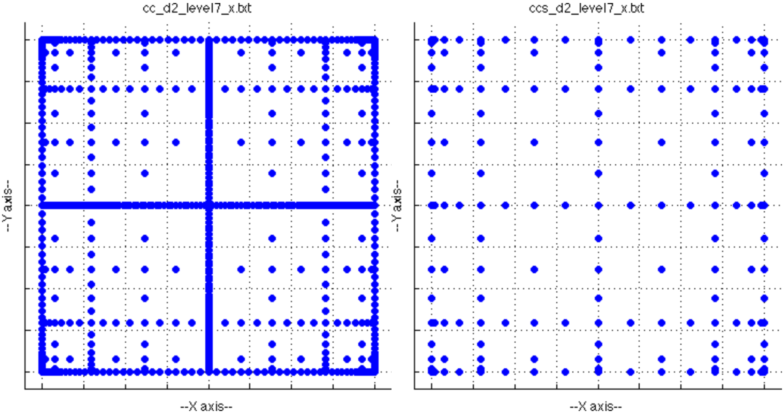
$$O = 2^{L+1} - 1$$

It is common to have allow this exponential growth to preserve the benefits of nesting.

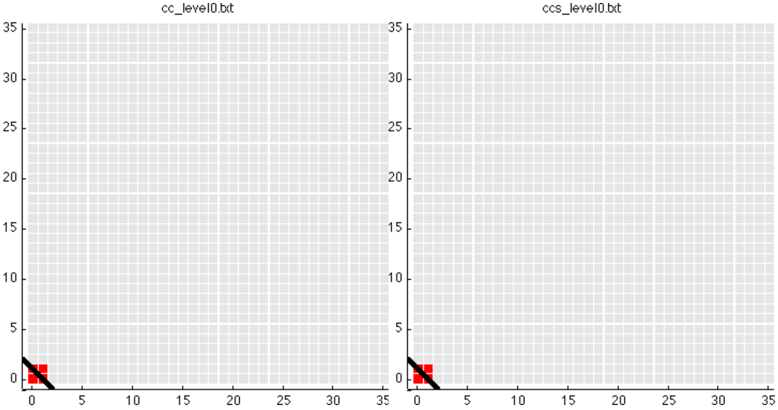
We have implemented a **slow growth** version of such rules; at each level, we use the lowest order nested rule that will achieve the necessary precision.

The advantages of the slow growth approach are most apparent in low dimensions, or for high order.

Growth Rules: Level 7 CC, Default versus Slow Growth

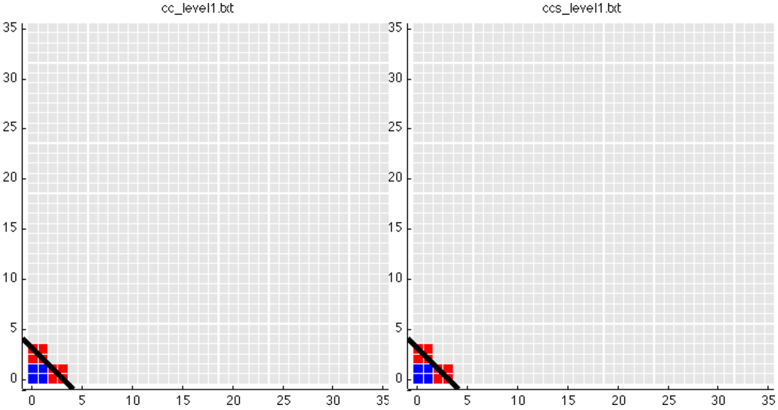


Default vs Slow Clenshaw Curtis, Level 0

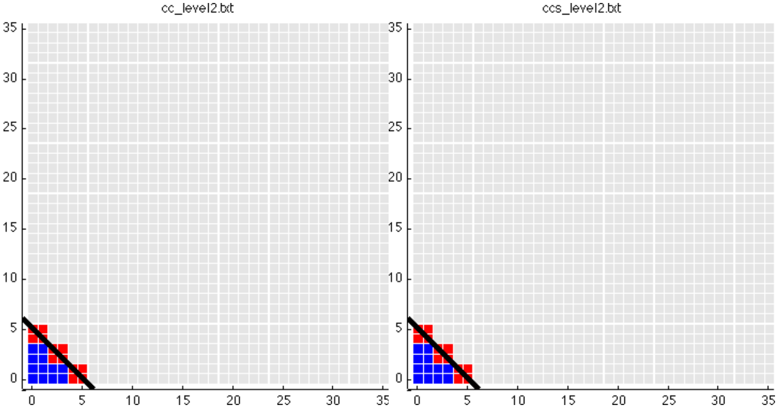


VirginiaTech

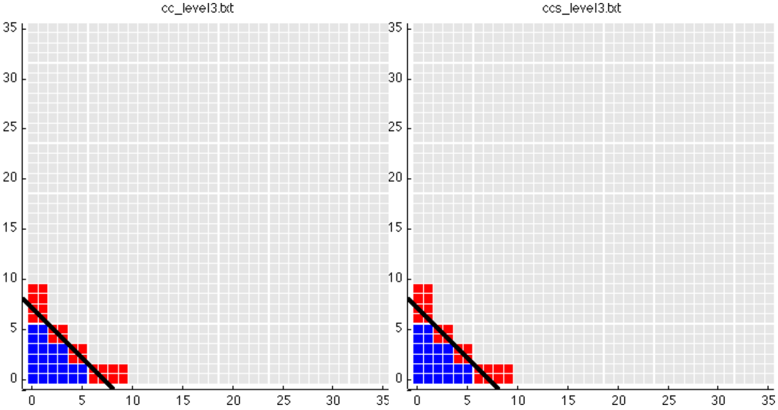
Default vs Slow Clenshaw Curtis, Level 1



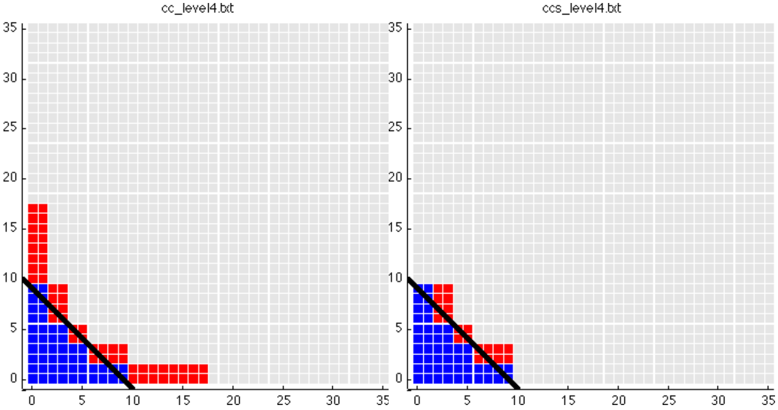
Default vs Slow Clenshaw Curtis, Level 2



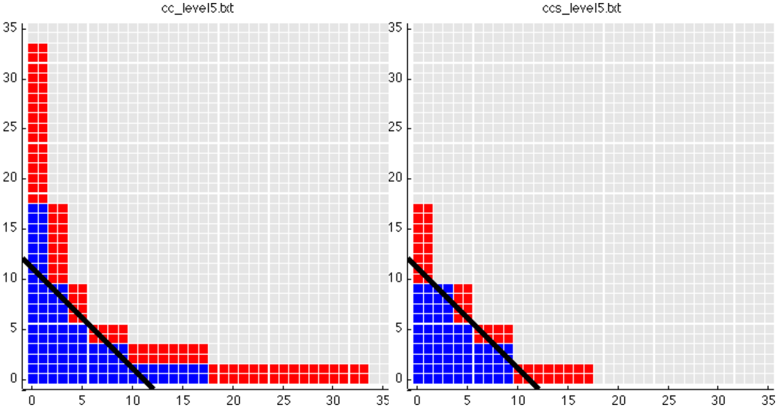
Default vs Slow Clenshaw Curtis, Level 3



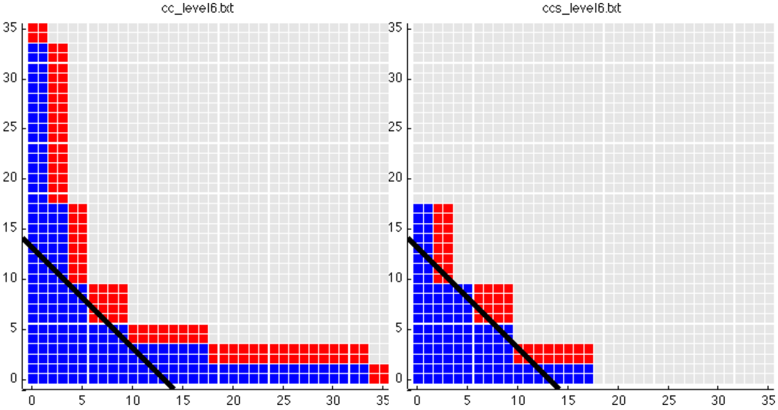
Default vs Slow Clenshaw Curtis, Level 4



Default vs Slow Clenshaw Curtis, Level 5



Default vs Slow Clenshaw Curtis, Level 6



The Anisotropic Smolyak Formula

We formed isotropic sparse grids using this constraint on indices:

$$L - M < \sum_j i_j \leq L$$

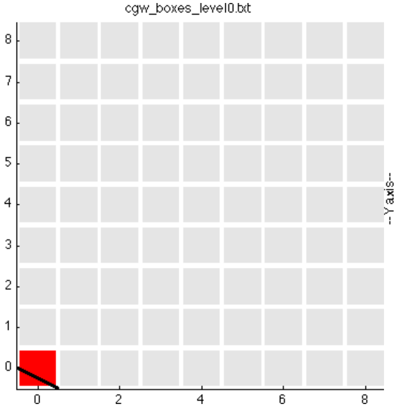
An anisotropic grid modifies this constraint with a weight vector α:

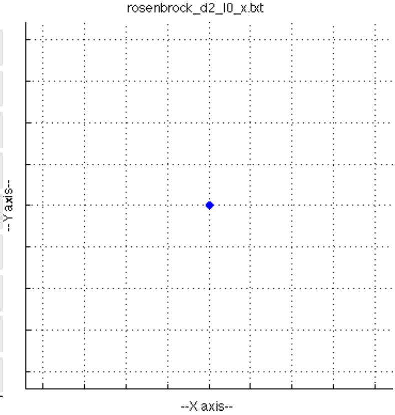
$$L \cdot \min(\alpha) - \sum_j \alpha_j < \sum_j \alpha_j \cdot i_j \leq L \cdot \min(\alpha)$$

The combining cofficient is now defined by:

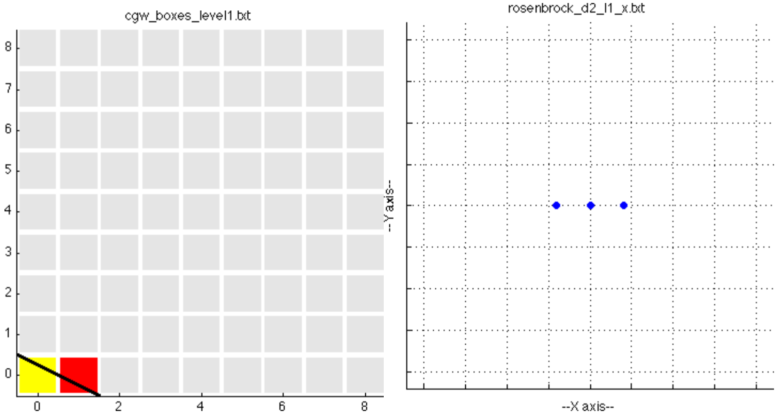
$$c_\alpha(\mathbf{i}) = \sum_{\substack{\mathbf{j} \in \{0,1\}^d \\ \mathbf{i}+\mathbf{j} \text{ satisfies constraint}}} (-1)^{|\mathbf{j}|}$$

[2,1] Hermite, Level 0

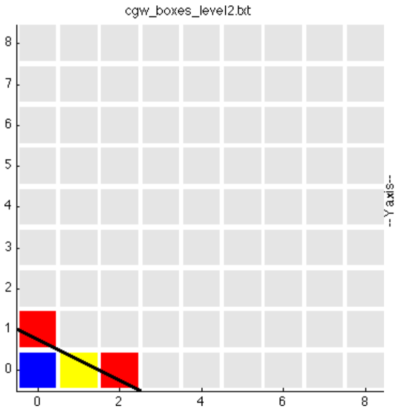


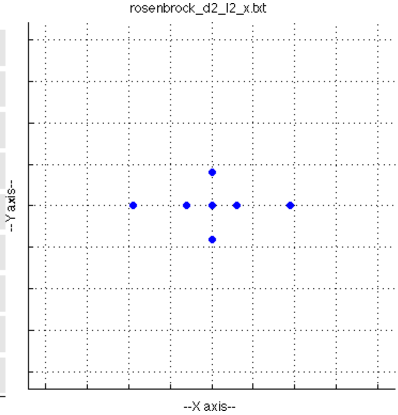


[2,1] Hermite, Level 1

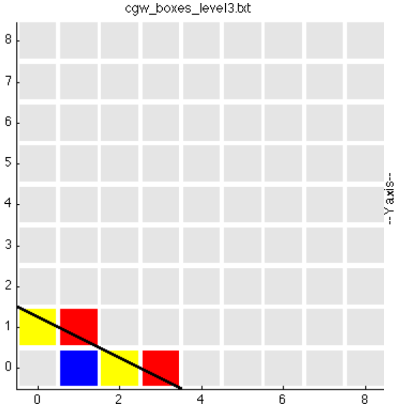


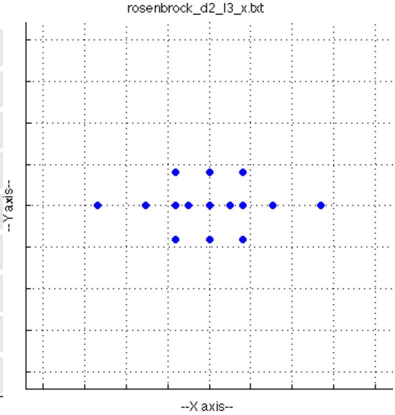
[2,1] Hermite, Level 2



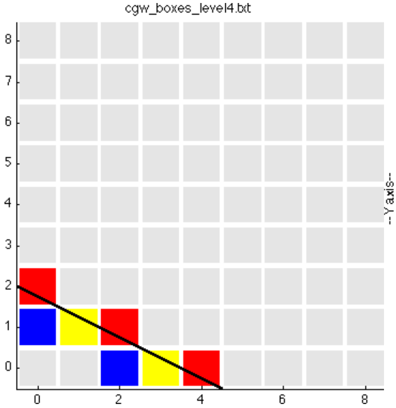


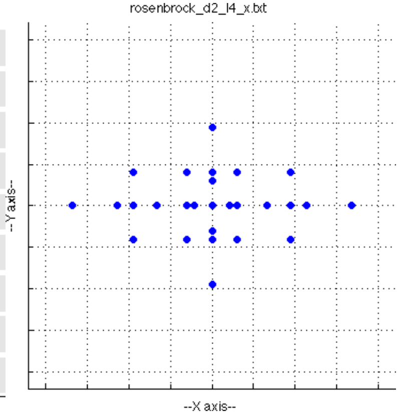
[2,1] Hermite, Level 3



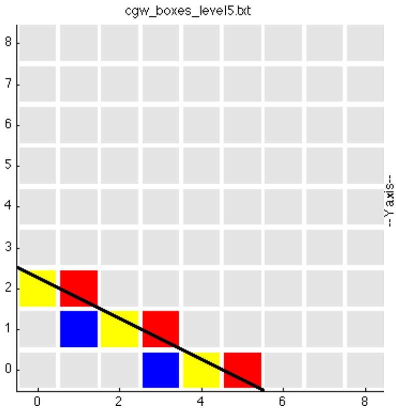


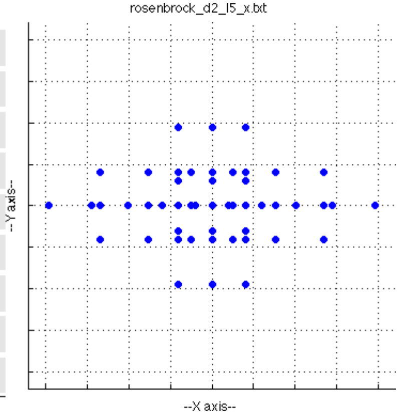
[2,1] Hermite, Level 4





[2,1] Hermite, Level 5





VirginiaTech

CONCLUSION: Adaptive Anisotropy

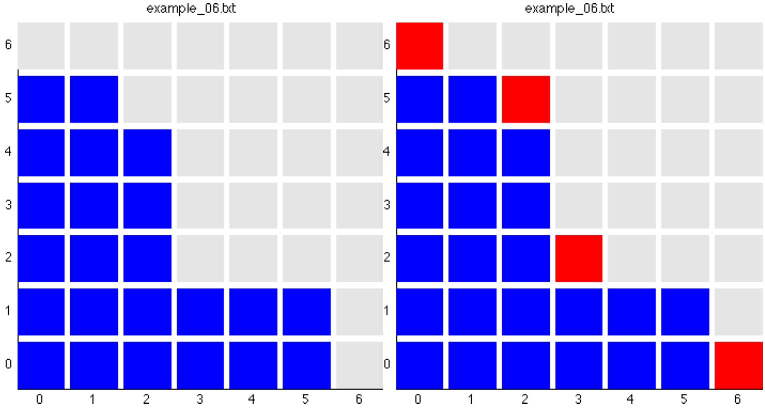
The anisotropy allowed by the SGMGA program must be prescribed beforehand, in terms of relative weights for each dimension, which define a linear constraint on level vectors.

The sparse grid is then formed from product grids whose level vectors lie between corresponding sets of parallel lines (in 2D) or hyperplanes.

The anisotropic formula can be applied to more general groupings of level vectors.

An approach that would make sense would be to adaptively seek level vectors that could be added to the current rule.

Consider adding product rules at “corners” of the diagram.



CONCLUSION: Current Concerns

A naive computation of the combining coefficient would required generating 2^M checks; we are investigating ways to avoid falling back into the exponential trap!

It would be useful to have analogues of the nested Gauss-Patterson family for problems involving Hermite or Laguerre weight functions.

For functions of limited smoothness, we are interested in the Klimke/Wohlmuth approach using hierarchical piecewise linear interpolants.

VirginiaTech

CONCLUSION: References

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

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

Fabio **Nobile**, Raul **Tempone**, Clayton **Webster**,
An Anisotropic Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data,
SINUM, 46(5), 2008, p2411-2442.

VirginiaTech