SPARSE_GRID_HERMITE
Sparse Grids Based on Gauss-Hermite Rules

SPARSE_GRID_HERMITE is a FORTRAN90 library which constructs sparse grids based on 1D Gauss-Hermite rules.

Sparse grids are more naturally constructed from a nested family of quadrature rules. Gauss-Hermite rules are not nested, but have higher accuracy. Thus, there can be a tradeoff. If we compare two sparse grids of the same "level", one using Gauss-Hermite rules and the other a nested rule, then the Gauss-Hermite sparse grid will have higher accuracy...but also a significantly greater number of points. When measuring efficiency, we really need to balance the cost in quadrature points against the accuracy, and so it is not immediately obvious which choice is best!

To slightly complicate matters, Gauss-Hermite rules are very weakly nested, in that the rules of odd order all include the abscissa value X=0.0. A sparse grid is constructed by summing a set of product rules. The weak nesting of the Gauss-Hermite rules means that a few of the points will be repeated across several sparse grids, and so an efficient code will try to group those points together and sum their weights.

Here is a table showing the number of points in a sparse grid based on Gauss-Hermite rules, indexed by the spatial dimension, and by the "level", which is simply an index for the family of sparse grids.

DIM:	1	2	3	4	5
LEVEL_MAX					
0	1	1	1	1	1
1	3	5	7	9	11
2	7	21	37	57	81
3	15	73	159	289	471
4	31	221	597	1265	2341
5	63	609	2031	4969	10363
6	127	1573	6397	17945	41913

Licensing:

The code described and made available on this web page is distributed under the GNU LGPL license.

Languages

SPARSE_GRID_HERMITE is available in a C++ version and a FORTRAN90 version and a MATLAB version.

Related Data and Programs:

QUADRATURE_RULES, a dataset directory which defines quadrature rules; a number of examples of sparse grid quadrature rules are included.

QUADRULE, a FORTRAN90 library which defines quadrature rules for various intervals and weight functions.

SGMGA, a FORTRAN90 library which creates sparse grids based on a mixture of 1D quadrature rules, allowing anisotropic weights for each dimension.

SMOLPACK, a C library which implements Novak and Ritter's method for estimating the integral of a function over a multidimensional hypercube using sparse grids.

SPARSE_GRID_CC, a FORTRAN90 library which creates sparse grids based on Clenshaw-Curtis rules.

SPARSE_GRID_F2, a dataset directory which contains the abscissas of sparse grids based on a Fejer Type 2 rule.

SPARSE_GRID_GL, a FORTRAN90 library which computes a sparse grid based on 1D Gauss-Legendre rules.

SPARSE_GRID_GP, a dataset directory which contains the abscissas of sparse grids based on a Gauss Patterson rule.

SPARSE_GRID_HERMITE, a dataset directory which contains the abscissas of sparse grids based on a Gauss Hermite rule.

SPARSE_GRID_HW, a FORTRAN90 library which creates sparse grids based on Gauss-Legendre, Gauss-Hermite, Gauss-Patterson, or a nested variation of Gauss-Hermite rules, by Florian Heiss and Viktor Winschel.

SPARSE_GRID_LAGUERRE, a FORTRAN90 library which creates sparse grids based on Gauss-Laguerre rules.

SPARSE_GRID_MIXED, a FORTRAN90 library which constructs a sparse grid using different rules in each spatial dimension.

SPARSE_GRID_NCC, a dataset directory which contains the abscissas of sparse grids based on a Newton Cotes Closed rule.

SPARSE_GRID_NCO, a dataset directory which contains the abscissas of sparse grids based on a Newton Cotes Open rule.

SPARSE_GRID_OPEN, a FORTRAN90 library which defines sparse grids based on open nested quadrature rules.

TOMS847, a MATLAB program which uses sparse grids to carry out multilinear hierarchical interpolation. It is commonly known as SPINTERP, and is by Andreas Klimke.

Reference:

1. Volker Barthelmann, Erich Novak, Klaus Ritter,

High Dimensional Polynomial Interpolation on Sparse Grids,
Advances in Computational Mathematics,
Volume 12, Number 4, 2000, pages 273-288.

2. Thomas Gerstner, Michael Griebel,
Numerical Integration Using Sparse Grids,
Numerical Algorithms,
Volume 18, Number 3-4, 1998, pages 209-232.
3. Albert Nijenhuis, Herbert Wilf,
Combinatorial Algorithms for Computers and Calculators,
Second Edition,
Academic Press, 1978,
ISBN: 0-12-519260-6,
LC: QA164.N54.
4. Fabio Nobile, Raul Tempone, Clayton Webster,
A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data,
SIAM Journal on Numerical Analysis,
Volume 46, Number 5, 2008, pages 2309-2345.
5. Sergey Smolyak,
Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions,
Doklady Akademii Nauk SSSR,
Volume 4, 1963, pages 240-243.
6. Dennis Stanton, Dennis White,
Constructive Combinatorics,
Springer, 1986,
ISBN: 0387963472,
LC: QA164.S79.

Source Code:

- sparse_grid_hermite.f90, the source code.
- sparse_grid_hermite.sh, commands to compile the source code.

Examples and Tests:

- sparse_grid_hermite_prb.f90, a sample calling program.
- sparse_grid_hermite_prb.sh, commands to compile and run the sample program.
- sparse_grid_hermite_prb_output.txt, the output from a run of the sample program.

The sample program creates three files, which are an example of how a sparse grid quadrature rule for spatial dimension 2 and level 3 can be defined using a Hermite quadrature rule:

- gh_d2_level3_r.txt, the "R" or "region" file.
- gh_d2_level3_w.txt, the "W" or "weight" file.
- gh_d2_level3_x.txt, the "X" or "abscissa" file.

List of Routines:

- COMP_NEXT computes the compositions of the integer N into K parts.
- GET_UNIT returns a free FORTRAN unit number.

- HERMITE_COMPUTE computes a Gauss-Hermite quadrature rule.
- HERMITE_COMPUTE_POINTS computes points of a Hermite quadrature rule.
- HERMITE_COMPUTE_WEIGHTS computes weights of a Hermite quadrature rule.
- IMTQLX diagonalizes a symmetric tridiagonal matrix.
- LEVEL_TO_ORDER_HERMITE: default growth for Hermite sparse grids.
- MONOMIAL_INTEGRAL_HERMITE integrates a Hermite monomial.
- MONOMIAL_QUADRATURE_HERMITE applies a quadrature rule to a monomial.
- MONOMIAL_VALUE evaluates a monomial.
- PRODUCT_HERMITE_WEIGHT computes the weights of a Hermite product rule.
- R8_CHOOSE computes the binomial coefficient C(N,K) as an R8.
- R8_FACTORIAL2 computes the double factorial function.
- R8_GAMMA evaluates Gamma(X) for a real argument.
- R8_HUGE returns a very large R8.
- R8_MOP returns the I-th power of -1 as an R8.
- R8COL_COMPARE compares columns in an R8COL.
- R8COL_SORT_HEAP_A ascending heapsorts an R8COL.
- R8COL_SORT_HEAP_INDEX_A does an indexed heap ascending sort of an R8COL.
- R8COL_SORTED_UNIQUE_COUNT counts unique elements in a sorted R8COL.
- R8COL_SWAP swaps columns I and J of an R8COL.
- R8COL_TOL_UNIQUE_COUNT counts tolerably unique entries in an R8COL.
- R8COL_UNDEX returns unique sorted indexes for an R8COL.
- R8MAT_PRINT prints an R8MAT.
- R8MAT_PRINT_SOME prints some of an R8MAT.
- R8MAT_TRANSPOSE_PRINT prints an R8MAT, transposed.
- R8MAT_TRANSPOSE_PRINT_SOME prints some of an R8MAT, transposed.
- R8MAT_WRITE writes an R8MAT file.
- R8VEC_COMPARE compares two R8VEC's.
- R8VEC_DIRECT_PRODUCT2 creates a direct product of R8VEC's.
- S_BLANK_DELETE removes blanks from a string, left justifying the remainder.
- SORT_HEAP_EXTERNAL externally sorts a list of items into ascending order.
- SPARSE_GRID_HERMITE computes a Hermite sparse grid.
- SPARSE_GRID_HERMITE_INDEX indexes a Hermite sparse grid.
- SPARSE_GRID_HERMITE_POINT computes the points of a Hermite sparse grid.
- SPARSE_GRID_HERMITE_SIZE sizes a Hermite sparse grid, discounting duplicates.
- SPARSE_GRID_HERMITE_SIZE_TOTAL Hermite sparse grid size counting duplicates.
- SPARSE_GRID_HERMITE_UNIQUE_INDEX maps nonunique points to unique points.
- SPARSE_GRID_HERMITE_WEIGHT computes Hermite sparse grid weights.
- SPARSE_GRID_HERMITE_WRITE writes a Hermite sparse grid rule to files.
- TIMESTAMP prints the current YMDHMS date as a time stamp.
- VEC_COLEX_NEXT3 generates vectors in colex order.

You can go up one level to the FORTRAN90 source codes.

Last revised on 15 May 2012.