The Journal of Open Source Software

SparseGridsKit.jl: Adaptive single- and multi-fidelity sparse grid approximation in Julia

Benjamin M. Kent [1¶]

1 CNR-IMATI, Pavia, Italy ¶ Corresponding author

DOI: 10.21105/joss.08300

Software

- Review
- Repository
- Archive

Editor: Mehmet Hakan Satman

Reviewers:

- @dannys4
- @baxmittens
- @fverdugo

Submitted: 23 May 2025
Published: 24 September 2025

Summary

Approximating functions with high-dimensional domains is crucial in modern scientific and engineering problems. An example of this is constructing surrogate models for quantities of interest in high-dimensional, parameterized PDE problems. These surrogate models are constructed to provide computationally inexpensive yet accurate approximations that can be used in applications such as uncertainty quantification, optimization, and parameter estimation (Ghanem et al., 2017). For suitably smooth functions, accurate surrogates can be constructed using global polynomial approximation techniques over the parameter domain, and a common approach is the use of sparse grid polynomial approximation. In particular, sparse grid polynomial interpolation techniques enable practitioners to approximate solutions to parametric problems in a non-intrusive manner using existing numerical solvers.

`SparseGridsKit.jl` provides a Julia (Bezanson et al., 2017) toolbox to manually and adaptively construct sparse grid polynomial approximations. Interpolation and quadrature routines allow evaluation and integration of the surrogate models. Multi-fidelity approximation via the multi-index stochastic collocation algorithm is also possible (Haji-Ali et al., 2016; John D. Jakeman et al., 2019; Piazzola et al., 2022). Approximations can be represented either in a basis of global Lagrange interpolation polynomials or in a basis of domain appropriate spectral-type global polynomials (e.g. Legendre, Chebyshev, Hermite, etc.).

Statement of need

Sparse grid approximation is a well-developed methodology and is featured in many survey articles and textbook chapters (Bungartz & Griebel, 2004; Cohen & DeVore, 2015; Le Maître & Knio, 2010; Schwab & Gittelson, 2011; Sullivan, 2015). The need for sparse grid surrogate modelling is demonstrated by its use in many applications, from simpler elliptic and parabolic PDEs to complex practical engineering problems (Li et al., 2024; Piazzola et al., 2021, 2022). The `SparseGridsKit.jl` implementation offers a rich set of features to enable this.

Specifically, `SparseGridsKit.jl` is a Julia implementation of adaptive sparse grid global polynomial approximation methods. Development was motivated by the desire for approximation code that closely resembles the mathematical literature to aid algorithm development and analysis.

The functionality includes:

- One-dimensional knots and quadrature rules.
- Multi-index set construction and manipulation.
- Combination technique sparse grid approximations, and functionality for interpolation, integration and derivatives of the surrogate model.
- Adaptive sparse grid approximation construction based on the ubiquitous Gerstner-Griebel dimensional adaptive algorithm (Gerstner & Griebel, 2003). This implementation uses

profit indicators as described in Nobile et al. (2016).
- Adaptive multi-fidelity approximation via the Multi-Index Stochastic Collocation (MISC) algorithm (Haji-Ali et al., 2016; John D. Jakeman et al., 2019; Piazzola et al., 2022).
- Conversion to and from Polynomial Chaos / spectral polynomial series representation.
- Limited support for surrogate model differentiation via automatic differentiation.

The functionality described above is tested and documented with examples included in the repository.

Other sparse grid approximation packages in Julia include:

- `DistributedSparseGrids.jl` (Bittens & Gates, 2023): A Julia package providing adaptive sparse grid approximation using a local hierarchical basis and distributed computing functionality.
- `Tasmanian.jl`: A Julia interface to the C++ Tasmanian library.
- `AdaptiveSparseGrids.jl`: A Julia package offering sparse grid approximation using a local hierarchical basis.

As described above, `SparseGridsKit.jl` instead offers functionality based upon global polynomial approximation targeting problems in which the function is assumed to be suitably smooth. The adaptive approximation algorithm is also split cleanly into `SOLVE-ESTIMATE-MARK-REFINE` steps to aid adaptive algorithm development. For non-smooth functions, global polynomial approximation is a poor choice and `DistributedSparseGrids.jl` or `AdaptiveSparseGrids.jl` may offer a better approximation strategy.

Other popular software packages implementing sparse grid approximation include:

- `Sparse Grids MATLAB Kit`: A MATLAB package on which the `SparseGridsKit.jl` is loosely based (Piazzola & Tamellini, 2024).
- `spinterp`: A MATLAB toolbox for sparse grid interpolation (Klimke & Wohlmuth, 2005) (no longer maintained).
- `UQLab`: A broad MATLAB uncertainty quantification toolkit (Marelli & Sudret, 2014).
- `PyApprox`: A Python package for high-dimensional approximation (J. D. Jakeman, 2023).
- `Dakota`: A C++ library for optimisation and surrogate modelling (Adams et al., 2024).
- `UQTk`: A collection of C++/Python uncertainty quantification tools including sparse grid quadrature (Debusschere et al., 2017).
- `Tasmanian`,`SG++`: C++ sparse grid approximation implementations with wrappers for many popular software languages (Pflüger, 2010; Stoyanov, 2015).

`SparseGridsKit.jl` specifically offers a Julia toolkit sharing the ethos of the `Sparse Grids MATLAB Kit`: to be user-friendly and aid fast algorithm prototyping. Notably, `SparseGridsKit.jl` also provides an implementation of the multi-index stochastic collocation algorithm which is currently only available in `PyApprox`.

Example Adaptive Sparse Grid Approximation

To briefly demonstrate the package, consider the *Gaussian Peak* test function from the Genz test suite (Genz, 1984),

$$f : [-1,1]^2 \to \mathbb{R}, \quad f(\vec{y}) = \exp\left(-\sum_{i=1}^{2}(c_i^2(y_i - W)^2)\right).$$

The problem is set up as follows:

```
using SparseGridsKit, Plots, LaTeXStrings
n = 2
C = 1.0
W = 0.0
```

```
T = "quarticdecay"
N = "gaussianpeak"
f = genz(n, C, W, T, N)
```

where `"quarticdecay"` defines coefficients $c_i = C(i+1)^{-4}$ and the constructed `f` is a function. An adaptive approximation is constructed as follows using the default configuration and a user specified profit tolerance `proftol`.

```
(sg, f_on_Z) = adaptive_sparsegrid(f, n; proftol=1e-10)
```

The adaptively constructed multi-index set and sparse grid are shown in Figure 1. These reflect the anisotropic nature of the function f due to the decaying coefficients c_i.

```
miset = get_mi_set(sg)
x     = getindex.(get_mi(miset), 1)
y     = getindex.(get_mi(miset), 2)
plot(
  scatter(x,y, xlabel=L"\beta_1", ylabel=L"\beta_2", legend=:none,
            title="Multi-Index Set", aspect_ratio=:equal),
  plot(sg; legend=:none, xlabel=L"y_1", ylabel=L"y_2", aspect_ratio=:equal)
)
```

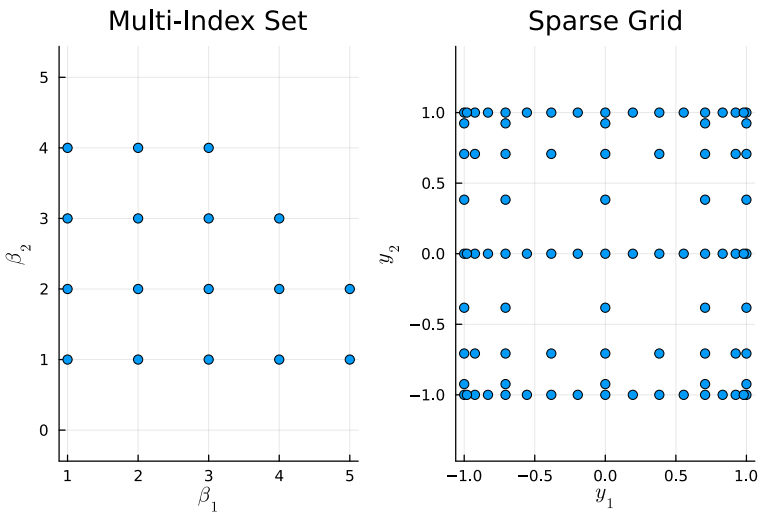

Figure 1: Multi-index set and sparse grid for adaptive approximation of Gaussian Peak

The resulting adaptive approximation is shown below in Figure 2.

```
plot(SparseGridApproximation(sg,f_on_Z); seriestype=:surface,
      title="", xlabel=L"y_1", ylabel=L"y_2", zlabel=L"u(\vec{y})")
```

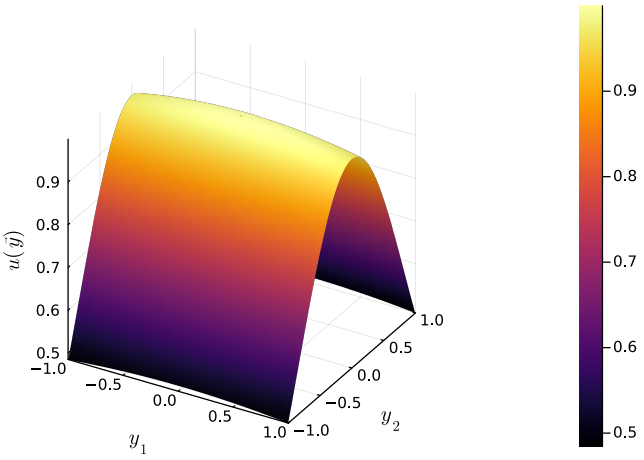

Figure 2: Adaptive approximation of the Gaussian Peak function f

Interpolation and integration of the resulting approximation is straightforward.

```
f_0       = interpolate_on_sparsegrid(sg, f_on_Z, [[0,0]])
integral  = integrate_on_sparsegrid(sg,f_on_Z)
println("f([0,0]) = $f_0, Integral(f) = $integral")
```

```
f([0,0]) = [1.0], Integral(f) = 0.8343194685053069
```

The sparse grid and the evaluations can then be wrapped up as a `SparseGridApproximation` or a `SpectralSparseGrid`. These can both be evaluated as functions.

```
f_sga         = SparseGridApproximation(sg,f_on_Z)
f_spectral    = convert_to_spectral_approximation(sg, f_on_Z)
f_sga_0       = f_sga([0,0])
f_spectral_0  = f_spectral([0,0])
println("f_sga([0,0]) = $f_sga_0, f_spectral(f) = $f_spectral_0")
```

```
f_sga([0,0])  = 1.0, f_spectral(f) = 1.0
```

For further examples please consult the package documentation.

Acknowledgements

The author has been supported by the project 202222PACR “Numerical approximation of uncertainty quantification problems for PDEs by multi-fidelity methods (UQ-FLY)”, funded by European Union – NextGenerationEU.

References

Adams, B. M., Bohnhoff, W. J., Dalbey, K. R., Ebeida, M. S., Eddy, J. P., Eldred, M. S., Hooper, R. W., Hough, P. D., Hu, K. T., Jakeman, J. D., Khalil, M., Maupin, K. A., Monschke, J. A., Prudencio, E. E., Ridgway, E. M., Robbe, P., Rushdi, A. A., Seidl, D. T., Stephens, J. A., … Winokur, J. G. (2024). *Dakota 6.21.0 documentation. Technical report SAND2024-15492O.* Sandia National Laboratories, Albuquerque, NM, November 2024. http://snl-dakota.github.io

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Bittens, M., & Gates, R. L. (2023). DistributedSparseGrids.jl: A Julia library implementing an adaptive sparse grid collocation method. *Journal of Open Source Software*, *8*(83), 5003. https://doi.org/10.21105/joss.05003

Bungartz, H.-J., & Griebel, M. (2004). Sparse grids. *Acta Numerica*, *13*, 147–269. https://doi.org/10.1017/s0962492904000182

Cohen, A., & DeVore, R. (2015). Approximation of high-dimensional parametric PDEs. *Acta Numerica*, *24*, 1–159. https://doi.org/10.1017/s0962492915000033

Debusschere, B., Sargsyan, K., Safta, C., & Chowdhary, K. (2017). Uncertainty quantification toolkit (UQTk). In *Handbook of uncertainty quantification* (pp. 1807–1827). Springer International Publishing. https://doi.org/10.1007/978-3-319-12385-1_56

Genz, A. C. (1984). *Testing multidimensional integration routines* (J. C. R. B. Ford & F. Thomasset, Eds.; pp. 81–94).

Gerstner, T., & Griebel, M. (2003). Dimension–adaptive tensor–product quadrature. *Computing*, *71*(1), 65–87. https://doi.org/10.1007/s00607-003-0015-5

Ghanem, R., Higdon, D., & Owhadi, H. (2017). *Handbook of uncertainty quantification*. Springer International Publishing. https://doi.org/10.1007/978-3-319-12385-1

Haji-Ali, A.-L., Nobile, F., Tamellini, L., & Tempone, R. (2016). Multi-index stochastic collocation for random PDEs. *Computer Methods in Applied Mechanics and Engineering*, *306*, 95–122. https://doi.org/10.1016/j.cma.2016.03.029

Jakeman, J. D. (2023). PyApprox: A software package for sensitivity analysis, bayesian inference, optimal experimental design, and multi-fidelity uncertainty quantification and surrogate modeling. *Environmental Modelling & Software*, *170*, 105825. https://doi.org/10.1016/j.envsoft.2023.105825

Jakeman, John D., Eldred, M. S., Geraci, G., & Gorodetsky, A. (2019). Adaptive multi-index collocation for uncertainty quantification and sensitivity analysis. *International Journal for Numerical Methods in Engineering*, *121*(6), 1314–1343. https://doi.org/10.1002/nme.6268

Klimke, A., & Wohlmuth, B. (2005). Algorithm 847: Spinterp: Piecewise multilinear hierarchical sparse grid interpolation in MATLAB. *ACM Transactions on Mathematical Software*, *31*(4), 561–579. https://doi.org/10.1145/1114268.1114275

Le Maître, O. P., & Knio, O. M. (2010). Spectral methods for uncertainty quantification: With applications to computational fluid dynamics. In *Scientific Computation*. Springer Netherlands. https://doi.org/10.1007/978-90-481-3520-2

Li, Y., Zoccarato, C., Piazzola, C., Bru, G., Tamellini, L., Guardiola-Albert, C., & Teatini, P. (2024). *Characterizing aquifer properties through a sparse grid-based Bayesian framework and InSAR measurements: A basin-scale application to Alto Guadalentín, Spain*. https://doi.org/10.22541/essoar.172373105.53381390/v1

Marelli, S., & Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. *Vulnerability, Uncertainty, and Risk*, 2554–2563. https://doi.org/10.1061/9780784413609.257

Nobile, F., Tamellini, L., Tesei, F., & Tempone, R. (2016). An adaptive sparse grid algorithm for elliptic PDEs with lognormal diffusion coefficient. In *Sparse grids and applications - stuttgart 2014* (pp. 191–220). Springer International Publishing. https://doi.org/10.1007/978-3-319-28262-6_8

Pflüger, D. (2010). *Spatially adaptive sparse grids for high-dimensional problems*. Institut für Informatik, Technische Universität München; Verlag Dr. Hut. ISBN: 9783868535556

Piazzola, C., & Tamellini, L. (2024). Algorithm 1040: The sparse grids MATLAB kit - a MATLAB implementation of sparse grids for high-dimensional function approximation and

uncertainty quantification. *ACM Transactions on Mathematical Software*, *50*(1), 1–22. https://doi.org/10.1145/3630023

Piazzola, C., Tamellini, L., Pellegrini, R., Broglia, R., Serani, A., & Diez, M. (2022). Comparing multi-index stochastic collocation and multi-fidelity stochastic radial basis functions for forward uncertainty quantification of ship resistance. *Engineering with Computers*, *39*(3), 2209–2237. https://doi.org/10.1007/s00366-021-01588-0

Piazzola, C., Tamellini, L., & Tempone, R. (2021). A note on tools for prediction under uncertainty and identifiability of SIR-like dynamical systems for epidemiology. *Mathematical Biosciences*, *332*, 108514. https://doi.org/10.1016/j.mbs.2020.108514

Schwab, C., & Gittelson, C. J. (2011). Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs. *Acta Numerica*, *20*, 291–467. https://doi.org/10.1017/s0962492911000055

Stoyanov, M. (2015). *User manual: TASMANIAN sparse grids* (ORNL/TM-2015/596). Oak Ridge National Laboratory.

Sullivan, T. J. (2015). Introduction to uncertainty quantification. In *Texts in Applied Mathematics*. Springer International Publishing. https://doi.org/10.1007/978-3-319-23395-6