Universitat Politècnica de Catalunya
Department of Civil and Environmental Engineering
Hydrogeology Group

GPKDE: Fortran code for Grid Projected Kernel Density Estimation of Discrete Particles Distributions

Documentation of Input-Output

Rodrigo Pérez-Illanes
Daniel Fernàndez-Garcia

Barcelona
April 16, 2023

Contents

Chapter 1

Introduction

This report contains information about the configuration of input files and output structures for the program GPKDE. The software performs Grid Projected Kernel Density Estimation of a discrete distribution of points in one, two or three dimensions, based on the methodology presented in Sole-Mari *et al.* (2019). This version is a new implementation of the method in the Fortran programming language, also parallelized with the OpenMP library.

Main module can be made available for use into external software by employing the files:

- `GridProjectedKDE.f90`: provides the reconstruction methodology, optimization for bandwidth selection, interfaces for computing density of a given dataset and writing output files.
- `Histogram.f90`: Computation of histograms with uniform and non-uniform weights.
- `KernelMultiGaussian.f90`: Kernel functions employed in the program.
- `GridCell.f90`: Utils for storing specific parameters for a grid cell while computing kernel convolutions.

Besides these, the code is delivered with an interface (`GPKDE.f90`) that allows users to input an external file storing the data coordinates and configure the optimization process for their specific application. In this regard, this document provides the configuration instructions for this specific interface.

Code repository is available via Github[1] and example applications are provided illustrating different use cases of the program. Users are encouraged to follow this channel for bug reports and code updates.

[1] https://github.com/upc-ghs/gpkde

Chapter 2

Input

The main program requires a single simulation file, in which information necessary for the configuration of the reconstruction process is provided. Parameters defining the reconstruction grid, controls over the optimization process and kernel specifications are indicated in this file.

2.1 Simulation file

Main entry point for a GPKDE simulation. The program is executed following the simplified structure

```
gpkde simfile
```

The structure of the simulation file and illustrative parameters is shown in Figure (2.1). Some of the input values are optional and eventually the interpretation of others is controlled by some of the values specified by the user. Specific meaning and possible options for these variables are shown in Table (2.1).

The simulation file begins with the file name of the input data. The user needs to specify an input format which might be written only as coordinates `(x y z)` or also including an specific weight `(x y z w)`. Following the format specification, the user can indicate the number of lines in the data file, which if given will lead to faster loading of the data points. If not, the program will infer how many particles are provided by first counting the number of lines in the input file. In the case the format considers only coordinates, the user can still specify a scaling parameter representing a uniform weight, which in practice can be useful to transform the particle density to other interpretations (for example mass concentration). The simulation file continues with the specification of the output file name, where density will be written.

The program will compute the bin or cell to which a point belongs based on the relevant dimensions specified by the user. In this regard, the domain origin, domain size and the bin sizes are required for creating a grid representation. An optional parameter allow users to specify if the reconstruction and other grids for internal use shall be allocated following the domain dimensions or adapted to the given points distribution. In this sense, the latter is usually more efficient in terms of memory as the grids are allocated based on the minimum and maximum coordinates of the distribution. The total number of points to be considered in the reconstruction process are those within the boundaries delimited by the domain specification.

After the grid specification, the program continues with necessary information for the optimization loop. More specifically, the maximum number of loops is required and a flag is provided to indicate whether the variables of a given loop should be exported or not. This can be useful for debugging purposes and essentially the quantities employed for determining the optimal kernel sizes are written to loop specific files, with names that concatenate the output file name and the number of loop. Also as a control of the optimization process, the user can specify whether or not the error convergence checks should be applied or not, meaning that the program will perform in any case the maximum number of loops. In case the error checks are preserved, the user may also specify a value for the relative error convergence parameter, which essentially defines the threshold for deciding whether convergence has been achieved or not.

Following, the simulation file requires information for kernels configuration. The user can indicate whether kernels should be computed in real time or precalculated and stored on a kernels database. For multidimensional reconstruction, it is generally recommended to preallocate the kernel database as it provides improvements in computational times with respect to real time calculations, sacrificing some accuracy on kernel values, which in practice is also controlled by the level of discretization of the database. The latter is specified in terms of the non-dimensional smoothing, that is, the ratio between the kernel bandwidth and the bin size. In case of using the database, the user needs to provided the minimum non-dimensional smoothing, a step, and the maximum value (`MinHL DeltaHL MaxHL`). For one dimensional problems, it has been observed that the program is able to provide fast reconstruction while using real time calculation of the kernels. The user may also indicate whether in multidimensional problems kernels shall be considered as isotropic or anisotropic, being the latter the default.

Main input file continues with the specification for selecting the initial kernel bandwidth. In this regard, the user can specify whether an initial bandwidth should be estimated from the expression for Gaussian distributions of Silverman (1986), or as a factor a multiplying the cell sizes, or by providing an array with the bandwidths for each direction.

```
0 : Optional comment line     | # GPKDE configuration file    |
1 : DataFileName              | particles.csv                 |
2 : i.   InputDataFormat      | 0      10000     1.0          |
    ii.  NPoints              | density.out                   | 3: OutputFileName
    iii. UniformWeight        | 0.0    0.0   0.0              | 4: DomainOrigin (x y z)
5 : i.  DomainSize (x y z)    | 100.0  50.0  10.0      1      |
    ii. GridAllocationFormat  | 1.0    1.0   1.0              | 6: BinSize (x y z)
7 : i.  NOptLoops             | 10     0                      |
    ii. ExportOptVars         | 0      1E-03                  | 8: i.  SkipErrorConvergence
9 : i.  KernelDatabase        | 1      0                      |    ii. RelativeConvergence
    ii. IsotropicKernels      | 1.0    0.1    10.0            |10: KernelDatabaseParams (MinHL DeltaHL MaxHL)
11: i.  InitialSmoothingFormat| 1      5.0                    |
    ii. BinSizeFactor         | 5.0    1.0    1.0             |12: InitialSmoothingArray (x y z)
13: AdvancedOptions           | 1                             |
14: i.   BoundKernelFormat    | 1      0.5    10.0            |
    ii.  MinHL    MaxHL       | 1      1E-04   1.0            |15: i.   MinRoughnessFormat
16: IsotropicThreshold        | 0.9                           |    ii.  MinRefRoughness
17: EffectiveWeightFormat     | 1                             |    iii. MinRoughnessLScale
18: UseGlobalSmoothing        | 0                             |
```

Figure 2.1: Illustrative GPKDE simulation file. Example parameters are enclosed between vertical dividers and their names are given in left/right columns.

Until this point, the previous parameters are expected to be provided, but an additional set of options is also given to control some advanced aspects of the reconstruction process. If the input file is given until a certain parameter, the subsequent interpretation will not continue, and parameters succsessfully read will be given to the program and the rest remain as default.

Once the user enabled the advanced options, it is possible to specify bounding values for the kernel sizes, with three different alternatives. The first bound kernel sizes from domain restrictions, which is the default format. In the upper bound, kernels cannot grow larger than a fraction of the domain size. This is a good approach to avoid issues when correcting kernels by boundary reflection, in order to avoid multiple reflections. The lower bound is established in terms to a minimum number of bins, by forcing kernels to have at least two cells from the center, which avoids extremelly small kernels. In the second bounding format, the users specifies values for `MinHL` and `MaxHL`, that will override previous specification if given in the context of the kernel database. In the third case, kernels are unbounded.

Following, options for bounding the net roughness are provided, which provides users some additional alternatives for controlling kernel sizes. In the first case, a minimum roughness is established assuming a Gaussian distribution and predefined value for the relative roughness analyzed from the Gaussian distribution problem. The program internally employs the standard deviation of the distribution and the maximum density, combined with a default minimum relative roughness to define the lower bound of the roughness. In the second case, the user provides a minimum relative roughness and a characteristic length scale. The program will use these values and the maximum density to obtain a lower bound. In the third case, the program reads a given value of roughness which is used as the lower bound and in the last case, the roughness is not bounded. In this case however, the program will not compute a bandwidth for cases with zero roughness (if any) and instead will assign the maximum already computed value. The next advanced specification allows user to indicate a threshold value to determine a relative fraction of a directional roughness with the sum of the main directions, to apply a correction to the estimated net roughness while using anisotropic kernels. This is useful for cases where the particle distribution presents a strong symmetry in one direction, possible leading to very small values of net roughness, but with a clearly dominant direction. In essence, if any of the directional roughnesses explains more than the `IsotropicThreshold` of the sum of the direction roughnesses, then net roughness is corrected.

Next, the user can specify the format for computing the effective number of particles for weighted histograms. The default format follows the method for weighted distributions of Kish (1965, 1992), where an effective number of particles is computed from the squared individual weights, and then an effective particle weight is obtained. A second format is left available for evaluation purposes and the effective weight is the simple average over all particle weights, with the effective number of particles equal to the total. The last parameter is a flag to indicate whether global smoothing expressions should be used for the reconstruction. This means that the net roughness is not a local integral anymore and instead is replaced by a domain integral and the optimal smoothing is isotropic and homogeneous for the whole domain, obtained with a global expression (see Sole-Mari & Fernàndez-Garcia, 2018). This option overrides the locality principles upon which most of the program is built and it can be useful in rare cases of extremelly sharp gradients.

Table 2.1: Simulation parameters for the GPKDE configuration file.

Id	Parameter	Type	Values
1	DataFileName	string	The name of the input data file.
2.1	InputDataFormat	int	0: Data file read as (x,y,z). 1: Data file with weights (x,y,z,w).
2.ii	NPoints	int	The number of data points. If NPoints=0 infers from file.
2.iii	UniformWeight	float	Uniform weight if InputDataFormat=0.
3	OutputFileName	string	The output file where density is written.
4	DomainOrigin	float	Coordinate x y z for the origin.
5.i	DomainSize	float	Dimensions x y z of the domain.
5.ii	GridAllocationFormat	int	0: Grids allocated according to domain size. 1: Grids allocated according to min/max coordinates.
6	BinSize	float	Dimensions x y z of the cells.
7.i	NOptLoops	int	The maximum number of optimization loops.
7.ii	ExportOptVars	int	0: Do not export optimization variables. 1: Write a file per optimization loop with variables.
8.i	SkipErrorConvergence	int	0: Verifies convergence criteria and break. 1: Skip convergence verification.
8.ii	RelativeConvergence	float	Threshold of relative change between subsequent loops.
9.i	KernelDatabase	int	0: Kernels are computed in real time. 1: Kernels are precalculated and stored in database.
9.ii	IsotropicKernels	int	0: Kernels are anisotropic. 1: Kernels are isotropic.
10	KernelDatabaseParams	float	Range of non-dimensional smoothing for the database. Only read if KernelDatabase=1.
11.i	InitialSmoothingFormat	int	0: First bandwidth obtained from Silverman expression. 1: First bandwidth as a factor multiplying bin size. 2: First bandwidth given by user as x y z array.
11.ii	BinSizeFactor	float	For initial bandwidth if InitialSmoothingFormat=1.
12	InitialSmoothingArray	float	Initial bandwidth if InitialSmoothingFormat=2.
13	AdvancedOptions	int	If 1, enables the interpretation of advanced parameters.
14.i	BoundKernelFormat	int	0: Bound kernels based on domain restrictions. 1: Read parameters 14.ii MinHL and MaxHL. Overrides previously given if using KernelDatabase. 2: Unbounded kernels.
15.i	MinRoughnessFormat	int	0: Minimum roughness assuming Gaussian distribution. 1: Limit determined using 15.ii and 15.iii. 2: Uses 15.ii as minimum value. 3: Minimum roughness is zero.
16	IsotropicThreshold	float	If the fraction between a directional roughness and isotropic net roughness is above this threshold, then then net roughness is corrected by isotropic estimate.
17	EffectiveWeightFormat	int	0: Kish's effective sample size to obtain effective weight. 1: Effective weight as simple average over all points.
18	UseGlobalSmoothing	int	If 1, optimal smoothing as global isotropic.

2.2 Command line interface

Some basic operations can also be managed from the command line interface implemented for the main program. The basic instructions for using the command line can be requested with the command `gpkde --help` or simply `gpkde -h`, which will show in console the following message:

```
GPKDE version 0.0.1
Program compiled Apr 12 2023 19:44:24 with GFORTRAN compiler (ver. *.*.*)

Fortran code for Grid Projected Kernel Density Estimation of discrete particles distributions

usage:

  gpkde [options] simfile

options:

  -h          --help                Show this message
  -l  <str>   --logname    <str>    Write program logs to <str>
  -nl         --nolog               Do not write log file
  -np <int>   --nprocs     <int>    Run with <int> processes
  -p          --parallel            Run in parallel
  -v          --version             Show program version

For bug reports and updates, follow:
  https://github.com/upc-ghs/gpkde
```

Figure 2.2: Help message from the GPKDE program.

Chapter 3

Output

The current version of the program may generate up to three different output files, which are detailed in the following.

3.1 Density

This is the main output from the program. It contains the reconstructed density and associated grid indexes. The file contains information only for those cells with a non-zero density. Notice that the latter implies that cells without particles, but with an estimated density, will also be written in this file. Besides density, the histogram density is by default also included in the last column for comparison purposes. The structure of this file is as follows:

```
1 : BinX
2 : BinY
3 : BinZ
4 : Density
5 : Histogram
```

3.2 Log file

This file is written during the interpretation of parameters and reconstruction process, containing a summary of the most relevant information related to the program execution and parameters. It reports the program workflow based on user input parameters and relevant metrics of the optimization process. This file is in general a good source of information for post-processing purposes. By default, every time the program is executed the file `gpkde.log` is written. Users can indicate that this file should not be generated or change its name by means of the command line parameters, as indicated in the previous chapter.

3.3 Optimization variables

File generated when users specify that optimization variables should be exported. One file is generated for each optimization loop following the naming convention `OutputFileName`+`loopId`. In contrast to the density file, optimization variables are only written for those cells that contain particles. The structure of this file is as follows:

```
1 : BinX                8 : ShapeFactorX             15: CurvatureBandwidthZ
2 : BinY                9 : ShapeFactorY             16: AveragedDensity
3 : BinZ                10: ShapeFactorZ             17: RoughnessXX
4 : Density             11: KernelBandwidthScale     18: RoughnessYY
5 : KernelBandwidthX    12: KernelSupportScale       19: RoughnessZZ
6 : KernelBandwidthY    13: CurvatureBandwidthX      20: NetRoughness
7 : KernelBandwidthZ    14: CurvatureBandwidthY
```

Bibliography

Kish, L. 1965. *Survey sampling.* John Wiley & Sons.

Kish, L. 1992. Weighting for unequal P_i. *Journal of Official Statistics*, **8**(2), 183–200.

Silverman, B. W. 1986. *Density estimation for statistics and data analysis.* Vol. 26. CRC press.

Sole-Mari, G., & Fernàndez-Garcia, D. 2018. Lagrangian modeling of reactive transport in heterogeneous porous media with an automatic locally adaptive particle support volume. *Water Resources Research*, **54**(10), 8309–8331.

Sole-Mari, G., Bolster, D., Fernàndez-Garcia, D., & Sanchez-Vila, X. 2019. Particle density estimation with grid-projected and boundary-corrected adaptive kernels. *Advances in Water Resources*, **131**(Sept.), 103382.