Software Implementation of the Partition of Unity Method

Roberto Cavoretto[a], Alessandra De Rossi[a], **Sandro Lancellotti**[a], Emma Perracchione[b]

[a] Department of Mathematics "Giuseppe Peano", University of Torino (Italy)
[b] Department of Mathematical Sciences "Giuseppe Luigi Lagrange", Politecnico di Torino (Italy)
roberto.cavoretto@unito.it, alessandra.derossi@unito.it,
sandro.lancellotti@unito.it, emma.perracchione@polito.it

The Partition of Unity (PU) scheme is a well-established and efficient kernel-based interpolation method. First introduced in the mid 1990s in [1], the PU method produces a global approximant by combining, via the use of compactly supported weights, many local fits [4]. Such a scheme is also rather popular for researchers working on local collocation schemes for PDEs. The PU method organizes the initial set of scattered data, that lay on a multivariate domain, into several patches. Then, for each of those patches it solves a small interpolation problem. A key step in its implementation is thus the one of efficiently distributing the scattered data into the different patches. A Matlab implementation of the PU scheme, based on the so-called kd-tree partitioning data structures (see [3]), already exists but it is not exploitable on recent Matlab releases. With this motivation, we propose an effective implementation of the PU scheme based on what we call the integer-based routines. The aim of this talk is to discuss the detailed implementation of the algorithm whose description was briefly treated in [2]. Moreover, motivated by the growing interest of the kernel community towards Python packages for machine learning, we also developed a Python implementation of the PU scheme. Finally, some experiments and comparisons between the two software implementations will be presented.

References

[1] I. Babuška, J.M. Melenk, *The partition of unity method*, Int. J. Numer. Meth. Eng., 40 (1997), pp. 727–758.

[2] R. Cavoretto, A. De Rossi, *A trivariate interpolation algorithm using a cube-partition searching procedure*, SIAM J. Sci. Comput., 37 (2015), pp. A1891–A1908.

[3] G.E. Fasshauer, *Meshfree Approximations Methods with Matlab*, World Scientific, Singapore, 2007.

[4] H. Wendland, *Fast evaluation of radial basis functions: Methods based on partition of unity*, in: Approximation Theory X: Wavelets, Splines, and Applications, C.K. Chui et al. (Eds.), Vanderbilt Univ. Press, Nashville, 2002, pp. 473–483.