WendlandXool: Simplified C++ code to compute Wendland functions

Carlos Argáez[*], Peter Giesl[**] and Sigurdur Freyr Hafstein [*]
[*]*Science Institute, University of Iceland, Tæknigarður, Dunhagi 5, 107 Reykjavík, Iceland (e-mail: {carlos,shafstein}@hi.is).*
[**]*Department of Mathematics, University of Sussex, Falmer, BN1 9QH, UK (e-mail: p.a.giesl@sussex.ac.uk)*

Abstract. Radial basis functions (RBFs) are certain real-valued functions Ψ that depend only on the distance of their argument to a fixed point, i.e. $\boldsymbol{\Psi}(\|\mathbf{x}-\mathbf{x}_0\|)$. Among them, one can find Gaussians, multiquadrics and, the subject of this paper, Wendland functions that are compactly supported and positive definite functions [1], constructed as polynomials on their compact support. They find a great amount of applications, in particular, in algorithms to construct Complete Lyapunov functions. In this paper, we present a new code to construct Wendland functions of any order, as well as their derived auxiliary functions. This new algorithm optimises, simplifies and shortens the code firstly presented in [2]. It optimises the routines to evaluate the function and presents a new feature: a compilable LaTeX report with all the instructions and steps to construct them.

Introduction

The Wendland functions $\boldsymbol{\Psi}^0_{l,k}$, where $l \in \mathbb{N}$ and $k \in \mathbb{N}_0$ are defined recursively by [1],

$$\left.\boldsymbol{\Psi}^0_{l,0}(r) = (1-r)^l_+ \text{ and } \boldsymbol{\Psi}^0_{l,k+1}(r) = \int_r^1 t\boldsymbol{\Psi}^0_{l,k}(t)\,\mathrm{d}t \text{ for } k=0,1,\ldots\right\} \text{ where } x_+ = \begin{cases} x \text{ for } x>0 \\ 0 \text{ for } x \le 0\end{cases} .$$

So, if we want to evaluate a Wendland function for given k and l at the point cr, where $c>0$ is a fixed constant, we obtain, $\boldsymbol{\Psi}^0_{l,k}(cr) = \underbrace{\int_{cr}^1 t_k \int_{t_k}^1 t_{k-1}\ldots\int_{t_2}^1}_{k\text{ iterations}} t_1\boldsymbol{\Psi}^0_{l,0}(t_1)\ \overbrace{\mathrm{d}t_1\ldots\mathrm{d}t_k}^{k\text{ differentials}}$. For a given Wendland function $\boldsymbol{\Psi}^0_{l,k}(cr)$ the auxiliary functions $\boldsymbol{\Psi}^1_{l,k}(cr)$ and $\boldsymbol{\Psi}^2_{l,k}(cr)$ are defined as follows for $r>0$, $\boldsymbol{\Psi}^1_{l,k}(cr) = r^{-1}\frac{d}{dr}\boldsymbol{\Psi}^0_{l,k}(cr)$ and $\boldsymbol{\Psi}^2_{l,k}(cr) = r^{-1}\frac{d}{dr}\boldsymbol{\Psi}^1_{l,k}(cr)$.

Results and discussion

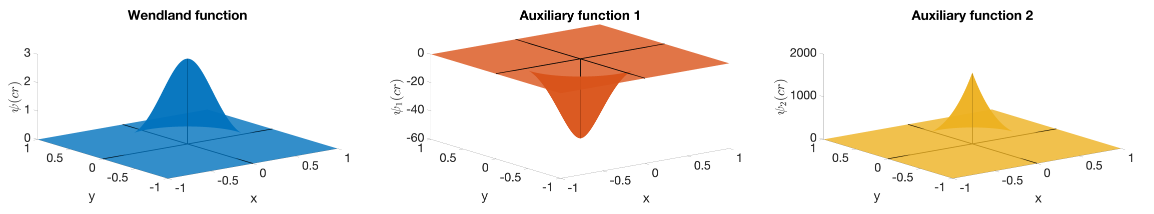

Figure 1: Left: Wendland function. Middle and right: Auxiliary functions 1 and 2.

The algorithm is written in C++ and uses the Armadillo library. The idea is simple and follows the natural increase of the exponents' values for a given base r when multiplying by the same base r or when integrating. Therefore, it also follows the coefficients change during the integration process, Table 1.

	Wendland function $\Psi_{3,1}(r)$						Function $\Psi^1_{3,1}(r)$				Function $\Psi^2_{3,1}(r)$		
Exponent k of r^k	0	1	2	3	4	5	0	1	2	3	-1	0	1
Coefficient c_k of c_kr^k	1	0	-10	20	-15	4	-20	60	-60	20	60	-120	60

Table 1: Matrices presenting the results and their storage.

The polynomial is stored as exponent and corresponding coefficient in a matrix of two rows and dynamically increasing columns. Therefore, evaluating these functions can optimally be done under the Horner's scheme. Since the Wendland functions and their derived auxiliary functions are essentially only defined up to a multiplicative nonzero constant, the algorithm keeps track of common factors to compute and display them using rational coefficients in the LaTeX report. To sum up, our optimised software makes it very simple to use Wendland functions for computations in C++. Additionally, it delivers a LaTeX file with the generated functions for reports associated to the computations.

References

[1] Wendland, H. (1995) Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* 389–396

[2] Argáez, C., and Giesl, P., Hafstein, S.F. (2017). Wendland Functions - A C++ Code to Compute Them. In *Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2017)*. 323–330. ISBN: 978-989-758-265-3.