# Contents

# Chapter 1

# Derivatives for RHS of Weight System

We start with:

$$\psi(x, y) = e^{-\epsilon^2(x^2+y^2)} \cdot G_k(2\epsilon^2(xx_i + yy_i)) \tag{1.1}$$

from Fornberg et al. [1], Equation 7.1 (Appendix A).

The authors provide that derivatives of the basis functions requires:

$$\frac{\partial^p G_k(z)}{\partial z^p} = G_{max(0,k-p)}(z), \qquad p = 0, 1, 2 \cdots \tag{1.2}$$

and derivatives of Equation 1.1 can be obtained via the *product rule*.

For RBF-GA, we have:

$$z = 2\epsilon^2(xx_i + yy_i) \tag{1.3}$$

Remember: the $x, y$ are the *stencil nodes*. The $x_i, y_i$ indicate a *stencil center*. That implies that the $\psi(x, y)$ function only requires the center information when evaluating $G_k$.

## 1.1 $\frac{\partial}{\partial x}$

First, we test derivation for $\frac{\partial}{\partial x}$. In Equation 1.2, $p = 1$.

Apply the product rule:

$$\begin{aligned}
\frac{\partial \psi(x, y)}{\partial x} &= \frac{\partial}{\partial x}\left(e^{-\epsilon^2(x^2+y^2)} \cdot G_k(z)\right) \\
&= \frac{\partial}{\partial x}\left(e^{-\epsilon^2(x^2+y^2)}\right) G_k(z) + e^{-\epsilon^2(x^2+y^2)}\frac{\partial G_k(z)}{\partial x}
\end{aligned} \tag{1.4}$$

Use the chain rule as necessary:

$$\begin{aligned}
\frac{\partial}{\partial x}\left(e^{-\epsilon^2(x^2+y^2)}\right) &= e^{-\epsilon^2(x^2+y^2)} \cdot \frac{\partial}{\partial x}\left(\epsilon^2(x^2 + y^2)\right) \\
&= e^{-\epsilon^2(x^2+y^2)}\left(2x\epsilon^2\right)
\end{aligned} \tag{1.5}$$

$$\frac{\partial G_k(z)}{\partial x} = \frac{\partial z}{\partial x}\frac{\partial}{\partial z}G_k(z)$$

$$= \frac{\partial}{\partial x}\left(2\epsilon^2(xx_i + yy_i)\right)G_{max(0,k-1)}(z)$$

$$= 2\epsilon^2 x_i G_{max(0,k-1)}(z) \tag{1.6}$$

Note that in the derivative of $z$ we assume the center coordinates $x_i$ and $y_i$ are considered scalars.

Finally, we substitute Equation 1.5 and 1.6 into Equation 1.4 to get:

$$\frac{\partial}{\partial x}\psi(x,y) = \left(2x\epsilon^2 G_k(z) + \frac{\partial G_k(z)}{\partial x}\right)e^{-\epsilon^2(x^2+y^2)}$$

$$= \left(2x\epsilon^2 G_k(z) + 2\epsilon^2 x_i G_{max(0,k-1)}(z)\right)e^{-\epsilon^2(x^2+y^2)}$$

$$= \left(xG_k(z) + x_i G_{max(0,k-1)}(z)\right)2\epsilon^2 e^{-\epsilon^2(x^2+y^2)} \tag{1.7}$$

We repeat the process for $\frac{\partial}{\partial y}$, but due to similarity to $\frac{\partial}{\partial x}$ for the general form:

$$\frac{\partial\psi(\mathbf{x})}{\partial x} = \left(xG_k(z) + x_i G_{max(0,k-1)}(z)\right)2\epsilon^2 e^{-\epsilon^2 r(\mathbf{x})^2} \tag{1.8}$$

## 1.2   Second Derivatives

$$\frac{\partial^2}{\partial x^2}\psi(x,y) = \frac{\partial}{\partial x}\left(\left(xG_k(z) + x_i G_{max(0,k-1)}(z)\right)2\epsilon^2 e^{-\epsilon^2(x^2+y^2)}\right)$$

$$= \frac{\partial}{\partial x}\left(2\epsilon^2 e^{-\epsilon^2(x^2+y^2)}\right)\left(xG_k(z) + x_i G_{max(0,k-1)}(z)\right)$$

$$+ \left(2\epsilon^2 e^{-\epsilon^2(x^2+y^2)}\right)\frac{\partial}{\partial x}\left(xG_k(z) + x_i G_{max(0,k-1)}(z)\right)$$

$$= 2\epsilon^2\left(e^{-\epsilon^2(x^2+y^2)}\left(2x\epsilon^2\right)\right)\left(xG_k(z) + x_i G_{max(0,k-1)}(z)\right)$$

$$+ \left(2\epsilon^2 e^{-\epsilon^2(x^2+y^2)}\right)\left(\left(G_k(z) + xG_{max(0,k-1)}(z)\right) + x_i\left(2\epsilon^2 x_i G_{max(0,k-2)}(z)\right)\right)$$

$$= 2\epsilon^2 e^{-\epsilon^2(x^2+y^2)}\left(2\epsilon^2 x^2 G_k(z) + 2\epsilon^2 x x_i G_{max(0,k-1)}(z)\right)$$

$$+ 2\epsilon^2 e^{-\epsilon^2(x^2+y^2)}\left(G_k(z) + xG_{max(0,k-1)}(z) + 2\epsilon^2 x_i^2 G_{max(0,k-2)}(z)\right)$$

$$\frac{\partial^2}{\partial x^2}\psi(x,y) = 2\epsilon^2 e^{-\epsilon^2(x^2+y^2)}\left(G_k(z)\left(2\epsilon^2 x^2 + 1\right) + G_{max(0,k-1)}(z)\left(2\epsilon^2 x x_i + x\right)\right.$$

$$\left. + G_{max(0,k-2)}(z)\left(2\epsilon^2 x_i^2\right)\right) \tag{1.9}$$

NOTE: this assumes $x_i$ is a constant.

## 1.3 Laplacian $\nabla^2$

The Laplacian, $\nabla^2 = \sum_{d=1}^{D} \frac{\partial^2}{\partial x_d^2}$ (where $D$ is the problem dimension) can be obtained following [1] to rewrite Equation 1.1 as the product of two functions in one variable. Author's Note: Have not mastered this yet

### 1.3.1 2D

### 1.3.2 3D

## 1.4 Projected Operators

For the Stokes problem we need projected operators on the sphere. Derivation of these operators can be done via the chain rule, but I suspect that we may be able to compose the projected operators using only the basic operators for $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, and $\frac{\partial}{\partial z}$ as demonstrated in one of my dissertation Appendices—I was able to compose differentiation matrices for projected operators based on DMs for the unprojected operators with minimal loss of precision.

In theory the composition/indirect approach to obtaining the projected operators will suffice. If not, we shall return to derive the RHS forms.

## 1.5 Problems with Paper

- In Equations 12-14 in Bengt's paper, the growth rate for the power on the $\frac{1}{\epsilon^*}$ is not obvious. Through trial and error I was able to deduce that the power should scale as: $\frac{1}{\epsilon^{2k}}$ where $k$ is the same $k$ on $G_k$.

  My trial and error was naive, but I found that without the proper scaling, any changes to $\epsilon$ would impact the conditioning and magnitude of elements in the matrix. I did a simple solve of $Ax = ones$ where $A$ was the LHS produced with RBF-GA. When $\epsilon$ would change, the values of $x$ would change scale as well. By fixing the power on the scaling to $2k$ I am able to input any value for $\epsilon$ (e.g. $1e-2 \rightarrow 10^{-15}$ and the values of $x$ change by a minuscule amount ($< 10^{-9}$).

- When $z = 2\epsilon^2 x \cdot x_i$ is negative (i.e., the cases where stencil nodes are outside of first quadrant), the gammainc function in matlab returns NaN values. How do we handle the situation?

  - Actually, the problem is not when they are outside the first quadrant. Instead, it appears when the nodes span multiple quadrants (e.g., when in $[-1,1]^d$ instead of $[0,1]^d$).
  - Answer: The $imag(gammainc(z,k)) < 10^{-15}$ for all cases tested. . Since it is so near to 0, we simply exclude it by requesting the $real(gammainc(z,k))$ when producing the bases.

- I also found that for $k = 0$ occasionally I get NaN values out of matlab's gammainc(z,k). Consulting the documentation for this routine I found the statement that, for small values of z and $k = 0$, "For small $x$ and $a$, $gammainc(x,a) \sim x^a$, so
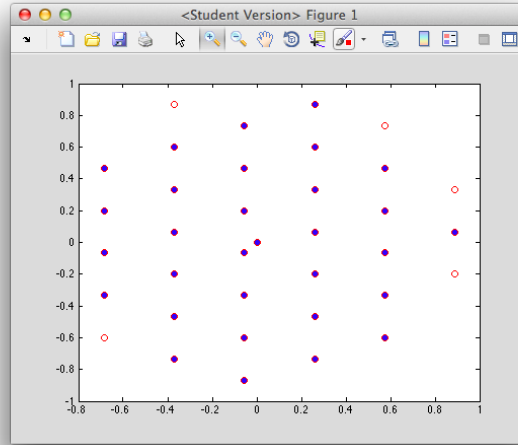
Figure 1.1: Red: A hexagonal grid within the unit disk ($r <= 1$). Blue: a stencil of size $n = 31$ nodes selected from the grid.

> $gammainc(0,0) = 1$". In fact, the occurrences of NaN correspond to $a = 0$ and $x < -1e - 5$. Working so near to 0, I simply added a check on the output from gammainc that catches NaN and substitutes the value 1. For $k > 0$ NaNs result in errors.
>
> The MATLAB snippet I use to calculate the $G_k$'s is:

- I am attempting to generate figures similar to Figure 3. I am able to reproduce the basis functions for $\Psi_1(x), \Psi_2(x), \Psi_3(x)$, but the remainder do not reproduce the figures in the paper.

  - Note: I was playing with different node distributions (regular, hexagonal, halton), and I realized that the visualized bases depend on the stencil and node locations. Bengt's paper does not explicitly state what node distribution was used to produce that figure. But at least I am confident that the results in Figure 1.2 (and the X-Y axes view in Figure 1.3) are correct for my node distributions.
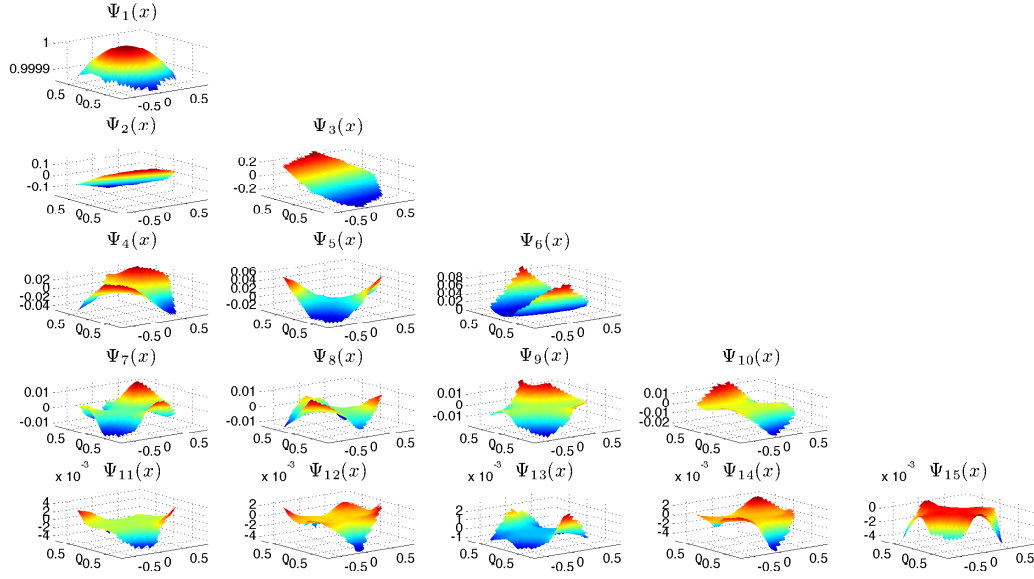
Figure 1.2: 3D view of the new basis functions from RBF-GA for the hexagonal grid and stencil shown in Figure 1.1 and stencil size $n = 31$.
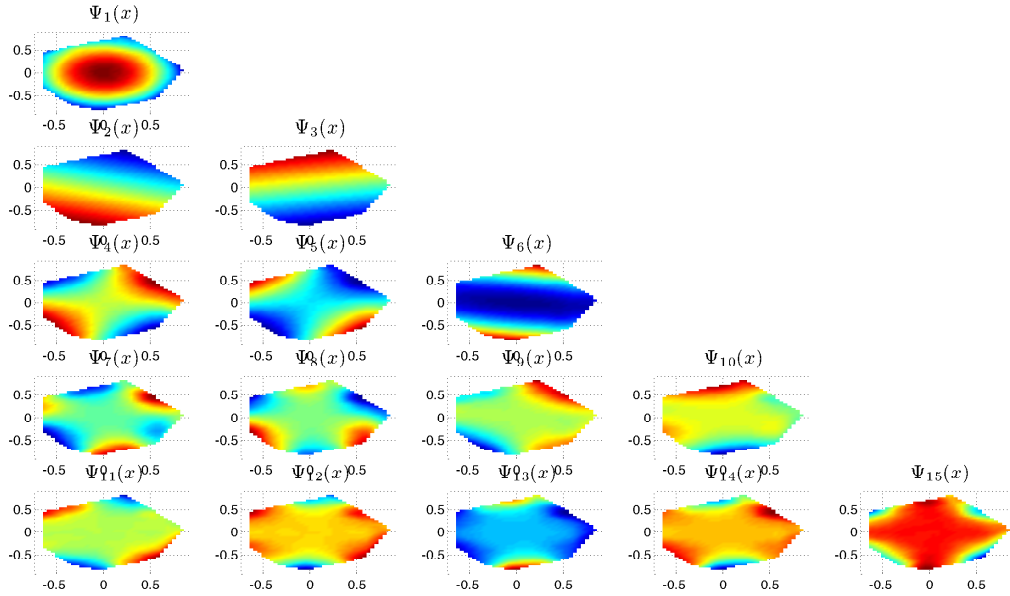


Figure 1.3: 2D view of the new basis functions from RBF-GA for the hexagonal grid and stencil shown in Figure 1.1 and stencil size $n = 31$.

# Bibliography

[1] Bengt Fornberg, Erik Lehto, and Collin Powell. Stable calculation of gaussian-based rbf-fd stencils. Technical Report 2012-018, Uppsala University, Numerical Analysis, 2012. 2, 4