UNIVERSITY PAUL SABATIER, IRIT LABORATORY

Hermite radial basis function (HRBF). Gradient and matrix developments

Rodolphe Vaillant

January 20, 2015

1 NOTATIONS

- $\mathbf{v}$: vectors are in bold
- $\mathbf{M}$: matrices are in bold capital.
- $\mathbf{v}_0 * \mathbf{v}_1$: scalar product between $\mathbf{v}_0$ and $\mathbf{v}_1$.
- $\mathbf{v}_0 \cdot \mathbf{v}_1$: componant wise product between $\mathbf{v}_0$ and $\mathbf{v}_1$.
- $\mathbf{v}_0 \otimes \mathbf{v}_1$: tensor product between $\mathbf{v}_0$ and $\mathbf{v}_1$ (outer product).
- $f_{x_0}(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial x_0}$: is the partial derivative of the first variable x_0 of f.

2 HRBF, GENERAL EXPRESSION

An HRBF is written:

$$f(\mathbf{x}) = \sum_{i}^{N} \alpha_i \phi(\|\mathbf{x}-\mathbf{p}_i\|) + \boldsymbol{\beta}_i * \nabla\left[\phi(\|\mathbf{x}-\mathbf{p}_i\|)\right] \tag{2.1}$$

With N the number of samples $\mathbf{p}_i$, then α_i a scalar and $\boldsymbol{\beta}_i$ a vector which are weights to be found to interpolate the data (pairs of points and normals). The expression $\nabla[\phi(\|\mathbf{x}-\mathbf{p}_i\|)]$ can be developed by applying the formula :

$$\nabla\left[g(f(\mathbf{x}))\right] = g'(f(\mathbf{x})).\begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_0} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{pmatrix} = g'(f(\mathbf{x}))\nabla f(\mathbf{x}) \tag{2.2}$$

with $f:\mathbb{R}^n \mapsto \mathbb{R}$ and $g:\mathbb{R}\mapsto\mathbb{R}$. If you're curious, you can find back this equality by using rules of *differentiabilities* for *multivariate* functions. [1]. These rules are rarely taught in secondary schools, I invite you to take a look at the multivariate chain rule. After this reading you should be able to understand why $[g(f(\mathbf{x}))]_{x_0} = g'(\mathbf{x}).f_{x_0}(\mathbf{x})$ and apply it to 2.2. So we have :

$$\nabla[\phi(\|\mathbf{x}-\mathbf{p}_i\|)] \quad = \quad \phi'(\|\mathbf{x}-\mathbf{p}_i\|).\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]$$

[1] Function with multiple variables $f(x_0,\ldots,x_n)$ as opposed to univariate functions with a single variable $f(x)$

$$= \phi'(\|\mathbf{x}-\mathbf{p}_i\|).\begin{pmatrix} \frac{(x_0-p_{i,x_0})}{\|\mathbf{x}-\mathbf{p}_i\|} \\ \vdots \\ \frac{(x_n-p_{i,x_n})}{\|\mathbf{x}-\mathbf{p}_i\|} \end{pmatrix}$$

$$= \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\frac{\mathbf{x}-\mathbf{p}_i}{\|\mathbf{x}-\mathbf{p}_i\|}$$

Which enable us to write the equation 2.1 :

$$f(\mathbf{x}) = \sum_i^N \alpha_i \phi(\|\mathbf{x}-\mathbf{p}_i\|) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\boldsymbol{\beta}_i * \frac{\mathbf{x}-\mathbf{p}_i}{\|\mathbf{x}-\mathbf{p}_i\|}$$

3 Computing the gradient of f

To Find the expression of the gradient ∇f you'll need to know the arithmetic rules of the differential operator ∇. Fortunately these are similar to the derivation rules of univariate functions:

$$\begin{aligned} \nabla[f+g] &= \nabla f + \nabla g \\ \nabla[f.g] &= \nabla f.g + f.\nabla g \\ \nabla\left[\frac{f}{g}\right] &= \frac{\nabla f.g - f.\nabla g}{g^2} \\ \nabla[\alpha.f] &= \alpha.\nabla f \end{aligned}$$

Unless needed we won't developed the expression of the gradient of the norml:

$$\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right] = \frac{\mathbf{x}-\mathbf{p}_i}{\|\mathbf{x}-\mathbf{p}_i\|}$$

We are finally ready to compute the gradient of f. Don't be afraid of the length of the development, because each line represents only a single simplification. The changes only happen in the blue expressions:

$$\begin{aligned}
\nabla f(\mathbf{x}) &= \nabla\left[\sum_i^N \alpha_i\phi(\|\mathbf{x}-\mathbf{p}_i\|) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right] \\
&= \sum_i^N \nabla\left[\alpha_i\phi(\|\mathbf{x}-\mathbf{p}_i\|)\right] + \nabla\left[\phi'(\|\mathbf{x}-\mathbf{p}_i\|)\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right] \\
&= \sum_i^N \alpha_i\nabla\left[\phi(\|\mathbf{x}-\mathbf{p}_i\|)\right] + \nabla\left[\phi'(\|\mathbf{x}-\mathbf{p}_i\|)\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right] \\
&= \sum_i^N \alpha_i \underbrace{\phi'(\|\mathbf{x}-\mathbf{p}_i\|)\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]}_{\mathbf{a}_i(\mathbf{x})} + \nabla\left[\phi'(\|\mathbf{x}-\mathbf{p}_i\|)\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right] \\
&= \sum_i^N \alpha_i\mathbf{a}_i(\mathbf{x}) + \nabla\left[\phi'(\|\mathbf{x}-\mathbf{p}_i\|)\left(\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right)\right] \\
&= \sum_i^N \alpha_i\mathbf{a}_i(\mathbf{x}) + \left(\nabla\left[\phi'(\|\mathbf{x}-\mathbf{p}_i\|)\right]\cdot\left(\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\nabla\left[\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right]\right) \\
&= \sum_i^N \alpha_i\mathbf{a}_i(\mathbf{x}) + \underbrace{\phi''(\|\mathbf{x}-\mathbf{p}_i\|)\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\cdot\left(\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right)}_{\mathbf{z}_i(\mathbf{x})} + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\nabla\left[\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right] \\
&= \sum_i^N \alpha_i\mathbf{a}_i(\mathbf{x}) + \mathbf{z}_i(\mathbf{x}) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\nabla\left[\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right] \\
&= \sum_i^N \alpha_i\mathbf{a}_i(\mathbf{x}) + \mathbf{z}_i(\mathbf{x}) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\nabla\left[\frac{\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)}{\|\mathbf{x}-\mathbf{p}_i\|}\right] \\
&= \sum_i^N \alpha_i\mathbf{a}_i(\mathbf{x}) + \mathbf{z}_i(\mathbf{x}) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\left(\frac{\nabla\left[\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)\right]\|\mathbf{x}-\mathbf{p}_i\| - \boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)\cdot\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]}{\|\mathbf{x}-\mathbf{p}_i\|^2}\right) \\
&= \sum_i^N \alpha_i\mathbf{a}_i(\mathbf{x}) + \mathbf{z}_i(\mathbf{x}) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\left(\nabla\left[\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)\right]\frac{\|\mathbf{x}-\mathbf{p}_i\|}{\|\mathbf{x}-\mathbf{p}_i\|^2} - \frac{\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)}{\|\mathbf{x}-\mathbf{p}_i\|^2}\cdot\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right)
\end{aligned}$$

$$= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \mathbf{z}_i(\mathbf{x}) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\left(\frac{\nabla\left[\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)\right]}{\|\mathbf{x}-\mathbf{p}_i\|} - \frac{\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)}{\|\mathbf{x}-\mathbf{p}_i\|^2}\cdot\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right)$$

The expression $\nabla\left[\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)\right]$ can be easily developed componant by componant:

$$\nabla\left[\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)\right] = \begin{pmatrix} \frac{\partial(\beta_{i,x_0}.(x_0-p_{i,x_0})+\ldots+\beta_{i,x_n}.(x_n-p_{i,x_n}))}{\partial x_0} \\ \vdots \\ \frac{\partial(\beta_{i,x_0}.(x_0-p_{i,x_0})+\ldots+\beta_{i,x_n}.(x_n-p_{i,x_n}))}{\partial x_n} \end{pmatrix} = \boldsymbol{\beta}_i$$

$$\begin{aligned}
\nabla f(\mathbf{x}) &= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \mathbf{z}_i(\mathbf{x}) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\left(\frac{\boldsymbol{\beta}_i}{\|\mathbf{x}-\mathbf{p}_i\|} - \frac{\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i)}{\|\mathbf{x}-\mathbf{p}_i\|^2}\cdot\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right) \\
&= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \mathbf{z}_i(\mathbf{x}) + \phi'(\|\mathbf{x}-\mathbf{p}_i\|)\left(\frac{\boldsymbol{\beta}_i}{\|\mathbf{x}-\mathbf{p}_i\|} - \frac{\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]}{\|\mathbf{x}-\mathbf{p}_i\|}\cdot\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right)
\end{aligned}$$

To get more compact expression we rename the norm as follow:

$$l_i(\mathbf{x}) = \|\mathbf{x}-\mathbf{p}_i\|$$

If we want to express the gradient under a matrix product form $\nabla f = A.\begin{pmatrix}\alpha_i \\ \boldsymbol{\beta}_i\end{pmatrix}$ we need to factorize by $\boldsymbol{\beta}_i$:

$$\begin{aligned}
\nabla f(\mathbf{x}) &= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \phi''(l_i(\mathbf{x}))\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\cdot(\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]) + \phi'(l_i(\mathbf{x}))\left(\frac{\boldsymbol{\beta}_i}{l_i(\mathbf{x})} - \frac{\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]}{l_i(\mathbf{x})}\cdot\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\right) \\
&= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \phi''(l_i(\mathbf{x}))\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]\cdot(\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]) + \phi'(l_i(\mathbf{x}))\frac{\boldsymbol{\beta}_i}{l_i(\mathbf{x})} - \phi'(l_i(\mathbf{x}))\frac{\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]}{l_i(\mathbf{x})}\cdot\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right] \\
&= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \phi'(l_i(\mathbf{x}))\frac{\boldsymbol{\beta}_i}{l_i(\mathbf{x})} + (\boldsymbol{\beta}_i * \nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right])\left(\phi''(l_i(\mathbf{x}))\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right] - \phi'(l_i(\mathbf{x}))\cdot\frac{\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]}{l_i(\mathbf{x})}\right) \\
&= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \underbrace{\frac{\phi'(l_i(\mathbf{x}))}{l_i(\mathbf{x})}}_{b_i(\mathbf{x})}\boldsymbol{\beta}_i + (\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i))\left(\phi''(l_i(\mathbf{x}))\frac{\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]}{l_i(\mathbf{x})} - \phi'(l_i(\mathbf{x}))\cdot\frac{\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]}{(l_i(\mathbf{x}))^2}\right) \\
&= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + b_i(\mathbf{x})\boldsymbol{\beta}_i + (\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i))\cdot\frac{\nabla\left[\|\mathbf{x}-\mathbf{p}_i\|\right]}{l_i(\mathbf{x})}\left(\phi''(l_i(\mathbf{x})) - \frac{\phi'(l_i(\mathbf{x}))}{l_i(\mathbf{x})}\right) \\
&= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + b_i(\mathbf{x})\boldsymbol{\beta}_i + (\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i))\cdot(\mathbf{x}-\mathbf{p}_i)\underbrace{\frac{1}{(l_i(\mathbf{x}))^2}\left(\phi''(l_i(\mathbf{x})) - \frac{\phi'(l_i(\mathbf{x}))}{l_i(\mathbf{x})}\right)}_{c_i(\mathbf{x})} \\
&= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \boldsymbol{\beta}_i b_i(\mathbf{x}) + (\boldsymbol{\beta}_i * (\mathbf{x}-\mathbf{p}_i))\cdot(\mathbf{x}-\mathbf{p}_i)c_i(\mathbf{x})
\end{aligned}$$

Now the problem is how do we factorize $\boldsymbol{\beta}_i$ while it is bound with a dot product with $((\mathbf{x}-\mathbf{p}_i)\cdot c_i(\mathbf{x}))$ and with a componant wise product with $b_i(\mathbf{x})$ at the same time. The trick here, is to express the two sort of product with a more general matrix product of corresponding dimension $(n \times n)$ in order to factorize it:

$$\begin{aligned}
\nabla f(\mathbf{x}) &= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \underbrace{\begin{pmatrix} b_i(\mathbf{x}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & b_i(\mathbf{x}) \end{pmatrix}}_{\mathbf{B}_i(\mathbf{x})}\boldsymbol{\beta}_i + (\mathbf{x}-\mathbf{p}_i)\otimes(\mathbf{x}-\mathbf{p}_i)c_i(\mathbf{x})\boldsymbol{\beta}_i \\
&= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + \mathbf{B}_i(\mathbf{x})\boldsymbol{\beta}_i + \underbrace{\begin{pmatrix} (x_0-p_{i,x_0})(x_0-p_{i,x_0})c_i(\mathbf{x}) & \ldots & (x_n-p_{i,x_n})(x_0-p_{i,x_0})c_i(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ (x_0-p_{i,x_0})(x_n-p_{i,x_n})c_i(\mathbf{x}) & \ldots & (x_n-p_{i,x_n})(x_n-p_{i,x_n})c_i(\mathbf{x}) \end{pmatrix}}_{\mathbf{C}_i(\mathbf{x})}\boldsymbol{\beta}_i \\
&= \sum_i^N \alpha_i \mathbf{a}_i(\mathbf{x}) + (\mathbf{B}_i(\mathbf{x}) + \mathbf{C}_i(\mathbf{x}))\boldsymbol{\beta}_i
\end{aligned}$$

4 Matrix representation

To be easier to read, developments will be done for $n = 3$. Generalizing to a $\mathbb{R}^n$ space is done easily because so far we kept f and ∇f as general as possible without setting n.
Finding the HRBF weights is done by developing the following system with a LU decomposition:

$$\begin{pmatrix} f(\mathbf{p}_i) \\ \nabla f(\mathbf{p}_i) \end{pmatrix} = \begin{pmatrix} c \\ \mathbf{n}_i \end{pmatrix}$$

If we develop according to the interpolated points:

$$\begin{pmatrix} f(\mathbf{p}_0) \\ \vdots \\ f(\mathbf{p}_N) \\ \nabla f(\mathbf{p}_0) \\ \vdots \\ \nabla f(\mathbf{p}_N) \end{pmatrix} = \begin{pmatrix} c \\ \vdots \\ c \\ \mathbf{n}_0 \\ \vdots \\ \mathbf{n}_N \end{pmatrix}$$

We have to expand the gradient to see the $n + 3n$ equations explicitly:

$$\begin{pmatrix} f(\mathbf{p}_1) \\ \vdots \\ f(\mathbf{p}_N) \\ f_x(\mathbf{p}_1) \\ f_y(\mathbf{p}_1) \\ f_z(\mathbf{p}_1) \\ f_x(\mathbf{p}_2) \\ \vdots \\ f_z(\mathbf{p}_N) \end{pmatrix} = \begin{pmatrix} c \\ \vdots \\ c \\ \mathbf{n}_{1,x} \\ \mathbf{n}_{1,y} \\ \mathbf{n}_{1,z} \\ \mathbf{n}_{2,x} \\ \vdots \\ \mathbf{n}_{N,z} \end{pmatrix}$$

The equation system $\mathbf{A}.\mathbf{x} = \mathbf{b}$ entirely developed look like this:

$$\mathbf{A} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \\ \beta_{1,x} \\ \beta_{1,y} \\ \beta_{1,z} \\ \vdots \\ \beta_{N,z} \end{pmatrix} = \begin{pmatrix} c \\ \vdots \\ c \\ n_{1,x} \\ n_{1,y} \\ n_{1,z} \\ \vdots \\ n_{N,z} \end{pmatrix}$$

With the matrix $\mathbf{A}$:

$$\begin{pmatrix}
\phi(l_1(\mathbf{x}_1)) & \dots & \phi(l_N(\mathbf{x}_1)) & e_{1,x}(\mathbf{x}_1) & e_{1,y}(\mathbf{x}_1) & e_{1,z}(\mathbf{x}_1) & \dots & e_{N,z}(\mathbf{x}_1) \\
\phi(l_1(\mathbf{x}_2)) & \dots & \phi(l_N(\mathbf{x}_2)) & e_{1,x}(\mathbf{x}_2) & e_{1,y}(\mathbf{x}_2) & e_{1,z}(\mathbf{x}_2) & \dots & e_{N,z}(\mathbf{x}_2) \\
\vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\phi(l_1(\mathbf{x}_N)) & \dots & \phi(l_N(\mathbf{x}_N)) & e_{1,x}(\mathbf{x}_N) & e_{1,y}(\mathbf{x}_N) & e_{1,z}(\mathbf{x}_N) & \dots & e_{N,z}(\mathbf{x}_N) \\
a_1(\mathbf{x_1}) & \dots & a_N(\mathbf{x_1}) & (b_1 + d_{1,x}d_{1,x}.c_1) & d_{1,y}d_{1,x}.c_1 & d_{1,z}d_{1,x}.c_1 & \dots & (b_N + d_{N,z}d_{1,x}.c_N) \\
a_1(\mathbf{x_1}) & \dots & a_N(\mathbf{x_1}) & d_{1,x}d_{1,y}.c_1 & (b_1 + d_{1,y}d_{1,y}.c_1) & d_{1,z}d_{1,y}.c_1 & \dots & (b_N + d_{N,z}d_{1,y}.c_N) \\
a_1(\mathbf{x_1}) & \dots & a_N(\mathbf{x_1}) & d_{1,x}d_{1,z}.c_1 & d_{1,y}d_{1,z}.c_1 & (b_1 + d_{1,z}d_{1,z}.c_1) & \dots & (b_N + d_{N,z}d_{1,z}.c_N) \\
\vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
a_1(\mathbf{x_N}) & \dots & a_N(\mathbf{x_N}) & d_{1,x}d_{1,z}.c_1 & d_{1,y}d_{1,z}.c_1 & (b_1 + d_{1,z}d_{1,z}.c_1) & \dots & (b_N + d_{N,z}d_{1,z}.c_N)
\end{pmatrix}$$

To be more compact the parameter $\mathbf{x}_i$ is implicit. Moreover we renamed some expressions:

$$b_i(\mathbf{x}) = b_i$$

$$
\begin{aligned}
c_i(\mathbf{x}) &= c_i \\
\mathbf{d}_i(\mathbf{x}) &= \begin{pmatrix} d_{i,x}(\mathbf{x}) \\ d_{i,y}(\mathbf{x}) \\ d_{i,z}(\mathbf{x}) \end{pmatrix} = \mathbf{x} - \mathbf{p}_i \\
\mathbf{e}_i(\mathbf{x}) &= \begin{pmatrix} e_{i,x}(\mathbf{x}) \\ e_{i,y}(\mathbf{x}) \\ e_{i,z}(\mathbf{x}) \end{pmatrix} = \phi'(\|\mathbf{x} - \mathbf{p}_i\|)\frac{\mathbf{x} - \mathbf{p}_i}{\|\mathbf{x} - \mathbf{p}_i\|} = \phi'(l_i(\mathbf{x})).\frac{\mathbf{d}_i(\mathbf{x})}{l_i(\mathbf{x})}
\end{aligned}
$$

In practice when filling the matrix, every coefficients of **A** with null division must be set to zero to be ignored.