Montestigliano Workshop | **Problem 7 Quadrature on the Sphere** | **Level 1 & 2 Grady Wright**

Problems 4–6 have dealt with approximating surface derivatives of a function from "scattered" samples. This problem deals with the related topic of approximating the surface integral of a function from "scattered" samples (refer to [1] for more details). The goal is to compute a set of quadrature (or cubature) weights $\{w_j\}_{j=1}^N$ for a set of nodes $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{S}^2$ such that for $f : \mathbb{S}^2 \longrightarrow \mathbb{R}$

$$\int_{\mathbb{S}^2} f(\mathbf{x}) d\mu(\mathbf{x}) \approx \sum_{j=1}^N w_j f(\mathbf{x}_j), \tag{1}$$

where $d\mu(\mathbf{x})$ is the measure for $\mathbb{S}^2$. As discussed in the lecture slides, one solution is to compute these weights from an RBF interpolant of f sampled at X. In the case that one uses a basic RBF interpolant with radial kernel ϕ for this problem, then it is explained in the lecture slides that the weights can be computed from solving the linear system:

$$\underbrace{\begin{bmatrix} & & \\ & \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) & \\ & & \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}}_{\underline{w}} = \underbrace{\begin{bmatrix} \rho_0 \\ \vdots \\ \rho_0 \end{bmatrix}}_{\rho_0 \underline{1}}, \tag{2}$$

where ρ_0 is the surface integral of the radial kernel ϕ shifted at any point on the sphere, e.g. $\rho_0 = \int_{\mathbb{S}^2} \phi(\|\mathbf{x}_i - \mathbf{x}_1\|) d\mu(\mathbf{x})$. This is a pretty neat results since it says the quadrature weights are just the coefficients of the RBF interpolant to the constant function scaled by ρ_0, which only requires solving *one* linear system. Note that the resulting weights give the exact answer to (1) when f is in the finite-dimensional space

$$V = \text{span}\left\{\phi(\|\mathbf{x} - \mathbf{x}_1\|), \phi(\|\mathbf{x} - \mathbf{x}_2\|), \ldots, \phi(\|\mathbf{x} - \mathbf{x}_N\|)\right\}, \quad \mathbf{x} \in \mathbb{S}^2.$$

This is in contrast to quadrature weights that are typically derived by making them exact for the finite-dimensional space consisting of all spherical harmonics up to a certain degree related to N. Numerical results for quadrature weights given by (2) and values of ρ_0 for different kernels can be found in the technical report [2].

In this problem we consider computing quadrature weights using the *general* form of the RBF interpolant from problem 1:

$$s(\mathbf{x}) = \sum_{j=1}^N c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{\ell=1}^{k^2} d_\ell p_\ell(\mathbf{x}), \tag{3}$$

where $s(\mathbf{x}_i) = f_i$, $i = 1, \ldots, N$ and $\sum_{j=1}^N c_j p_\ell(\mathbf{x}_j) = 0$, $\ell = 1, \ldots, k^2$. Here, $p_\ell(\mathbf{x})$, $\ell = 1, \ldots, k^2$ will be assumed to correspond to the standard spherical harmonics, i.e.

$$p_1(\mathbf{x}) = Y_0^0(\mathbf{x}),\ p_2(\mathbf{x}) = Y_1^{-1}(\mathbf{x}),\ p_3(\mathbf{x}) = Y_1^0(\mathbf{x}),\ p_4(\mathbf{x}) = Y_1^1(\mathbf{x}), \ldots$$

Note that the `sphHarm` function in the `rbfsphere` package can be used to compute spherical harmonics of a given degree and order. As discussed in the introductory lecture slides, the interpolation coefficients can be determined from the linear system

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{c} \\ \underline{d} \end{bmatrix} = \begin{bmatrix} \underline{f} \\ \underline{0} \end{bmatrix} \tag{4}$$

where A is given in (2) above, P is the N-by-k^2 matrix with $P_{j,\ell} = p_\ell(\mathbf{x}_j)$, and $\underline{0}$ is the zero vector of size k^2. Here we assume that P has full-column rank so that a unique solution to (4) is guaranteed. A delightful result from using this interpolant is that for any $k \geq 1$ the quadrature weights can be computed without needing to know ρ_0, the surface integral or ϕ. In the case that $k = 1$, this can be shown rather straightforwardly (try it!), but for $k > 1$ the argument is a little more subtle (see [1]).

Following a similar procedure to how (2) is derived in the lecture slides, but with a little more effort, the following system of equations can be derived for computing the quadrature weights associated with the interpolant (3):

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{w} \\ \underline{d} \end{bmatrix} = \begin{bmatrix} \rho_0 \underline{1} \\ \underline{\rho} \end{bmatrix}, \tag{5}$$

where $\underline{1}$ is the vector of all ones of length N, and $\underline{\rho}$ is a vector of length k^2 with entry $\rho_1 = \int_{\mathbb{S}^2} Y_0^0(\mathbf{x}) d\mu(\mathbf{x})$ and all other entries equal to zero. Note that $Y_0^0(\mathbf{x})$ is a constant so that $\rho_1 = 4\pi Y_0^0(\boldsymbol{\xi})$, for any $\boldsymbol{\xi} \in \mathbb{S}^2$. As mentioned above, the weights are independent of ρ_0, so you can simply set it equal to 1 in (5).

7.a) *Level 1.* Write a function in MATLAB for computing quadrature weights for a given set of nodes $X = \{\mathbf{x}\}_{j=1}^N$ on the unit sphere sphere using (5). Implement the code so that it accepts as input an array containing the nodes X, a value $k \geq 1$ for the additional spherical harmonic terms, and function handle for the radial kernel that should. Try to vectorize the code so that at no loops are used. Use Gaussian elimination (the `mldivide` or "backslash" function in MATLAB) to solve the linear system for the coefficient vector. Your function should return a vector containing the weights. As in problem 1.c), if you don't feel up to writing code for a general k, just implement the case of $k = 1$ and $k = 2$ correctly.

(i) Test your code for the following target functions, given as anonymous function in MATLAB :

```
f1 = @(x) 0.75*exp(-(9*x(:,1)-2).^2/4 - (9*x(:,2)-2).^2/4 - (9*x(:,3)-2).^2/4)...
    + 0.75*exp(-(9*x(:,1)+1).^2/49 - (9*x(:,2)+1)/10 - (9*x(:,3)+1)/10)...
    + 0.5*exp(-(9*x(:,1)-7).^2/4 - (9*x(:,2)-3).^2/4 - (9*x(:,3)-5).^2/4)...
    - 0.2*exp(-(9*x(:,1)-4).^2 - (9*x(:,2)-7).^2 - (9*x(:,3)-5).^2);
f2 = @(x) (1 + sign(-9*x(:,1) - 9*x(:,2) + 9*x(:,3)))/9;
```

`f1` is smooth and its surface integral is 6.6961822200736179523..., while `f2` is discontinuous with a surface integral of $4\pi/9$. Use the IMQ and Gaussian radial kernels and the thin plate spline kernel all with $k = 2$ for your tests and use the $N = 400$ MD nodes.

(ii) Compute the errors in approximations of the integrals from part (a) for $N = 10^2, 20^2, \ldots, 60^2$ MD nodes and plot the relative errors vs. $\sqrt{N}$ to get a feeling for the convergence rates of the approximations. Are there differences between in the convergence rates for the different target functions? How about for the different kernels?

(iii) Repeat (ii) for the minimum energy (ME) nodes. How do these results compare to the MD nodes?

(iv) Repeat (ii) for the Fibonacci nodes using $N = 10^2+1, 20^2+1, \ldots, 60^2+1$. How do these results compare to the MD and ME nodes?

(v) Make a plot of the weights from (ii)–(iv) for some N and verify they are all positive. Make the figure by plotting the nodes on the sphere and coloring each node with a color representative of the value of weight for that node. The —scatter3— function might be useful here.

7.b) *Level 2.* The system (5) does not quite have the same form as the one in (4) because of the non-zero term $\underline{\rho}$ in the right hand side. So, the coefficients $\underline{w}$ and $\underline{d}$ cannot be interpreted as interpolation coefficients for an interpolant of the form (3). Thus, if one has a "fast method" for computing interpolants of the form (3) with the constraints in (4) (as we will develop in a later problem), the code would not be immediately adaptable to solving the quadrature problem. However, it is possible to adapt the quadrature weight system (5) to one that is immediately equivalent to the system for interpolation (4) as shown in [1].

The idea is to express the weights as a combination of two orthogonal vectors

$$\underline{w} = \underline{w}_{\|} + \underline{w}_{\perp}, \tag{6}$$

where $\underline{w}_{\|}$ is the orthogonal projection of $\underline{w}$ onto the range of P and $\underline{w}_{\perp}$ is in the left null-space of P, i.e. $P^T \underline{w}_{\perp} = 0$. The standard normal equations give $\underline{w}_{\|}$ as

$$\underline{w}_{\|} = P(P^T P)^{-1}\underline{\rho}. \tag{7}$$

Once $\underline{w}_{\|}$ is known, (5) can be re-written in terms of $\underline{w}_{\perp}$ as follows:

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{w}_{\perp} \\ \underline{d} \end{bmatrix} = \begin{bmatrix} \rho_0 \underline{1} - A\underline{w}_{\|} \\ 0 \end{bmatrix}. \tag{8}$$

This system now mimics the system for interpolation in (3). As mentioned above, any non-zero value of ρ_0 can be used here since the weights do not depend on this value. Also, note that the proper way to compute $\underline{w}_{\|}$ in (7) is through the QR decomposition or singular value decomposition (SVD), not the normal equations.

(i) Write a function in MATLAB similar to 7.a) for computing quadrature weights by means of Equations (6)–(8). Use the QR decomposition to compute $\underline{w}_{\|}$. Test your function using the suggested tests in 7.a) (i).

References

[1] E. Fuselier, T. Hangelbroek, F. Narcowich, J. Ward, and G. Wright, *Kernel based quadrature on spheres and other homogeneous spaces*, Numerische Mathematik, Online (2014), pp. 1–36.

[2] A. Sommariva and R. S. Womersley, *Integration by RBF over the sphere*, Applied Mathematics Report AMR 05/17, University of New South Wales, Sydney, 2005.