Montestigliano Workshop | **Problem 8: Vector field interpolation and decomposition** | **Level 1: Grady Wright**

In this problem you will look at interpolating and decomposing a vector field $\mathbf{u}$ that is tangent to the sphere using the RBF method described in the lecture notes (which comes from the paper [1]). The decomposition is known as the Helmholtz or Hodge decomposition and states that any vector field $\mathbf{u}$ that is tangent to the sphere can be *uniquely* decomposed as $\mathbf{u} = \mathbf{u}_{\text{div}} + \mathbf{u}_{\text{curl}}$, where $\mathbf{u}_{\text{div}}$ has zero surface divergence and $\mathbf{u}_{\text{curl}}$ has zero surface curl on the sphere. This decomposition can often provide useful diagnostic information in applications. For example, in the atmosphere the divergence-free (or rotational) part of the horizontal wind field gives details about cyclonic storms, while the curl-free (or irrotational) part gives details on high and low pressure systems. Similarly, in the ocean, these respective components of the horizontal ocean currents give information on gyres and overturning flow. The RBF method from [1] can be used to approximate these two components using only samples of the field at (possibly scattered) locations and solving a single interpolation problem. Additionally, a stream function and velocity potential can be constructed for the divergence-free and curl-free fields, respectively. An implementation of this method is provided in the `rbfsphere` package with the function `rbfvectdecomp`, although presently it is only available for the Gaussian radial kernel.

8.a) The following MATLAB anonymous functions define the components of a divergence-free vector field $\mathbf{u}_{\text{div}} = [u_{\text{div}}\ v_{\text{div}}\ w_{\text{div}}]$ tangent to the sphere (when expressed with respect to Cartesian coordinates):

```
udiv= @(x,y,z) 0.5*y.*(x.^4+y.^4-4*y.^2.*z.^2-6*x.^2.*(y.^2-2*z.^2)-1);
vdiv=@(x,y,z) -0.5*x.*(x.^4+y.^4-4*x.^2.*z.^2-6*y.^2.*(x.^2-2*z.^2)-1);
wdiv=@(x,y,z) 8*x.*y.*z.*(y.^2-x.^2);
```

Similarly, the following functions define the components of a curl-free field $\mathbf{u}_{\text{curl}} = [u_{\text{curl}}\ v_{\text{curl}}\ w_{\text{curl}}]$ tangent to the sphere:

```
ucurl=@(x,y,z) 5*x/(2*sqrt(2)).*(sqrt(2)*z.^2.*(3-7*z.^2)-y.^2.*(3+y.^2)+x.^2.*(1+6*y.^2)-x.^4);
vcurl=@(x,y,z) 5*y/(2*sqrt(2)).*(sqrt(2)*z.^2.*(3-7*z.^2)-x.^2.*(3+x.^2)+y.^2.*(1+6*x.^2)-y.^4);
wcurl=@(x,y,z) -5*z/4.*(6+sqrt(2)*(x.^4-6*x.^2.*y.^2+y.^4)-2*z.^2.*(10-7*z.^2));
```

$\mathbf{u}_{\text{div}}$ was computed from the surface curl of a scalar potential function so that it is guaranteed to be divergence-free. Similarly, $\mathbf{u}_{\text{curl}}$ was computed from the surface gradient of a scalar potential function, guaranteeing it is surface curl-free.

Now consider the combined field $\mathbf{u} = \mathbf{u}_{\text{div}} + \mathbf{u}_{\text{curl}}$:

```
u=@(x,y,z) udiv(x,y,z) + ucurl(x,y,z);
v=@(x,y,z) vdiv(x,y,z) + vcurl(x,y,z);
w=@(x,y,z) udiv(x,y,z) + ucurl(x,y,z);
```

(i) Sample the components of the field $\mathbf{u}$ at $N = 900$ MD points. Use these samples in the `rbfvectdecomp` function with a shape parameter of $\varepsilon = 2.5$ to compute approximations to $\mathbf{u}_{\text{div}}$ and $\mathbf{u}_{\text{curl}}$ at some other set of nodes (say $N = 3600$ ME nodes). Compute the errors in the approximations this function gives to $\mathbf{u}_{\text{div}}$ and $\mathbf{u}_{\text{curl}}$. How good is the decomposition? Also compute the errors in the approximation of the whole filed $\mathbf{u}$. How good is the interpolant of the field?

(ii) Make a quiver plot of the approximated $\mathbf{u}_{\text{div}}$ and $\mathbf{u}_{\text{curl}}$ fields.

(iii) Use the `rbfvectdecomp` function to make a contour plot of a stream function for div-free part of the field and a contour plot of a velocity potential for the curl-free part. See the tutorial for plotting on the sphere for examples on how this can be done.

8.b) The file `FlowOverMountainN1849.mat` in the data directory of the `rbfsphere` package has velocity data from a numerical simulation at day 15 of a shallow water code for the standard test problem of flow over an isolated mountain. This data can be loaded into the MATLAB workspace using:

```
load FlowOverMountainN1849;
```

The node locations are stored in the variable `x` and the velocity field data (which is expressed with respect to Cartesian coordinates) is stored as individual components in the variables `u`, `v`, and `w`. Use the `rbfvectdecomp` function to decompose the field. Make quiver plots of the decomposed fields on a much finer set of points. Plot contour of a stream function for the divergence-free field and contours for a velocity potential for the curl-free field. Can you locate the approximate position of the mountain based on these plots?

References

[1] E. J. Fuselier and G. B. Wright, *Stability and error estimates for vector field interpolation and decomposition on the sphere with RBFs*, SIAM J. Num. Anal., 47 (2009), pp. 3213–3239.