Reduced Basis Milestones

Task 1:
Assume that your training set are real 3D vectors expressed in a Cartesian basis. Verify that G is the identity matrix. Show that any vector in that training set can then be expressed as

$$\vec{a} \sim \sum_{i=1}^{m} \langle \vec{a} | e_i \rangle e_i$$

where the inner product is simply the Euclidean norm.

Task 2:

- Write Python code to build an orthonormal basis using the Gram-Schmidt
- Apply your code to 1000 monomials x^n with x in [-1,1] and n in [1,10].
- To orthonormalise the basis, use the Gram-Schmidt procedure, see e.g. https://en.wikipedia.org/wiki/Gram%E2%80%93Schmidt_process
- You will need to specify an integration rule for the normalisation. Does your result depend on the integration you specify, e.g. Scipy's “trapezoidal” or “simpson”?
- How large is your orthonormal basis?
- Verify that your basis is indeed orthonormal.
- Plot the first few basis elements.

Task 3:

- Implement the greedy algorithm as described in Appendix A in https://journals.aps.org/prd/pdf/10.1103/PhysRevD.87.124005
- Build a reduced basis for (separately):
    - Bessel function of the first kind (from scipy.special import jv as BesselJ)
    - Spherical Bessel functions (from scipy.special import spherical_jn)