Problem 1: Where do you belong, Krylov?

Problem Statement

a

b

c

Solution

a

$$Aq^i = \frac{q^{i+1}}{\|q^{i+1}\|} + \sum_k^i \langle Aq^i, q^k \rangle q^k \in \mathcal{K}_{i+1}$$

b

$$\begin{bmatrix} Aq_1 & | & \dots & | & Aq_n \end{bmatrix} = \begin{bmatrix} Qh_1 & | & \dots & | & Qh_n \end{bmatrix}$$
$$\implies Aq_1 = Qh_1$$
$$Aq_1 \in \mathcal{K}_2 \implies Qh_1 \in \mathcal{K}_2$$
$$Aq_1 = \begin{bmatrix} * \\ * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},\ Aq_2 = \begin{bmatrix} * \\ * \\ * \\ 0 \\ \vdots \\ 0 \end{bmatrix},\ \dots Aq_n = \begin{bmatrix} * \\ * \\ * \\ * \\ \vdots \\ * \end{bmatrix}$$

c

Problem 2: Ask Gram, Schmidt, or Givens for QR

Problem Statement

a

b

c

Solution

a

Taken from "Linear Algebra Done Right" by Sheldon Axler.

Suppose $v_1, \ldots, v_m$ is a lin. ind. list of vectors in V. Let $e_1 = \frac{v_1}{\|v_1\|}$. For $j = 2, \ldots, m$, define e_j inductively by:

$$e_j = \frac{v_j - \langle v_j, e_1 \rangle e_1 - \cdots - \langle v_j, e_{j-1} \rangle e_{j-1}}{\|v_j - \langle v_j, e_1 \rangle e_1 - \cdots - \langle v_j, e_{j-1} \rangle e_{j-1}\|}$$

(i.e. the Gram-Schmidt process definition). Then, $e_1, \ldots, e_m$ is an orthonormal list of vectors in V s.t.:

$$\text{span}(v_1, \ldots, v_j) = \text{span}(e_1, \ldots, e_j) \forall j = 1, \ldots, m$$

Note that for $j = 1$, $\text{span}(v_1) = \text{span}(e_1)$ because v_1 is a positive mulitple of e_1.

Suppose $1 < j < m$ and it has been verified that:

$$\text{span}(v_1, \ldots, v_{j-1}) = \text{span}(e_1, \ldots, e_{j-1})$$

Note that $v_j \notin \text{span}(v_1, \ldots, v_{j-1})$ because $v_1 \ldots, v_m$ are linearly independent. Thus, $v_j \notin \text{span}(e_1, \ldots, e_{j-1})$. As such, we are not dividing by zero in the definition of e_j. We can also see that $\|e_j\| = 1$ by its definition.

Let $k \in [1, j)$. Then:

$$\begin{aligned} \langle e_j, e_k \rangle &= \left\langle \frac{v_j - \langle v_j, e_1 \rangle e_1 - \ldots - \langle v_j, e_{j-1} \rangle e_{j-1}}{v_j - \langle v_j, e_1 \rangle e_1 - \ldots - \langle v_j, e_{j-1} \rangle e_{j-1}}, e_k \right\rangle \\ &= \frac{\langle v_j, e_k \rangle - \langle v_j, e_k \rangle}{v_j - \langle v_j, e_1 \rangle e_1 - \ldots - \langle v_j, e_{j-1} \rangle e_{j-1}} \\ &= 0 \end{aligned}$$

Thus $e_1, \ldots, e_j$ is an orthonormal list.

From the definition of e_j, we see that $v_j \in \text{span}(e_1, \ldots, e_j)$. Combined with the equivalency of the spans provided above, we know:

$$\text{span}(v_1, \ldots, v_j) \subset \text{span}(e_1, \ldots, e_j)$$

Both these lists are lin. ind., thus both subspaces have dimension j and are equal. $\square$

b

for $j = 1, \ldots, n$ **do**
  $v \leftarrow a^j$
  for $i = 1, \ldots, j-1$ **do**
    $R_{ij} \leftarrow \langle v, q^i \rangle$ {m multiplications, $(m-1)$ additions $\implies 2m-1$}
    $v \leftarrow v - R_{ij} q^i$ {m multiplications, m subtractions $\implies 2m$}
  end for{Total cost is }
  $q^j \leftarrow \frac{v}{\|v\|}$ {m multiplications, $m-1$ additions, 1 for sqrt $\implies 2m$}
  $R_{jj} \leftarrow \langle a^j, q^j \rangle$ {m multiplications, $(m-1)$ additions $\implies 2m-1$}
end for
$Q = (q^1|\ldots|q^n)$

The inner loop requires:

$$(4m-1)(1-1)+(4m-1)+(2-1)+\ldots+(4m-1)(n-1) = (4m-1)(0+1+\ldots+n-1) = (4m-1)(\frac{1}{2}n(n+1)-1)$$

The outer portion requires $(4m-1) * n$ operations. Summing:

$$= (4m+1)(\frac{1}{2}n(n+1)-1) + (4m+1)(n) = (4m+1)(\frac{1}{2}n(n)-1+n)$$

Simplifying and discarding lower order terms we find:

$$= 2mn^2$$

c

An upper Hessenberg matrix of size $n \times n$ will require $n-1$ operations to zero out the non-zero elements below the diagonal, which will result in a QR decomposition. Givens matrices take the form:

$$G_n = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

Code

Problem 1

```python
import sys
import numpy as np

float_formatter = "{:.2f}".format

np.set_printoptions(formatter={"float_kind": float_formatter})


# Taken from: https://en.wikipedia.org/wiki/Arnoldi_iteration#The_Arnoldi_iteration
def arnoldi_iteration(A, b, n: int):
    """Compute a basis of the (n + 1)-Krylov subspace of the matrix A.

    This is the space spanned by the vectors {b, Ab, ..., A^n b}.

    Parameters
    ----------
    A : array_like
        An m x m array.
    b : array_like
        Initial vector (length m).
    n : int
        One less than the dimension of the Krylov subspace, or equivalently the *degree* of the Krylov
     space. Must be >= 1.

    Returns
    -------
    Q : numpy.array
        An m x (n + 1) array, where the columns are an orthonormal basis of the Krylov subspace.
    h : numpy.array
        An (n + 1) x n array. A on basis Q. It is upper Hessenberg.
    """
    eps = 1e-12
    h = np.zeros((n + 1, n))
    Q = np.zeros((A.shape[0], n + 1))
    # Normalize the input vector
    Q[:, 0] = b / np.linalg.norm(b, 2)  # Use it as the first Krylov vector
    for k in range(1, n + 1):
        v = np.dot(A, Q[:, k - 1])  # Generate a new candidate vector
        for j in range(k):  # Subtract the projections on previous vectors
            h[j, k - 1] = np.dot(Q[:, j].conj(), v)
            v = v - h[j, k - 1] * Q[:, j]
        h[k, k - 1] = np.linalg.norm(v, 2)
        if h[k, k - 1] > eps:  # Add the produced vector to the list, unless
            Q[:, k] = v / h[k, k - 1]
        else:  # If that happens, stop iterating.
            return Q, h
    return Q, h


if __name__ == "__main__":
    A = np.array([[1, 2, 3], [2, 2, 4], [3, 4, 4]])
    b = np.array([1, -1, 1])

    q, h = arnoldi_iteration(A, b, 3)

    print("Q:")
    print(q)
    print("H:")
    print(h)
```