10.4.3 Gaussian elimination implementation

MO2.3 | MO2.8 | MO2.9

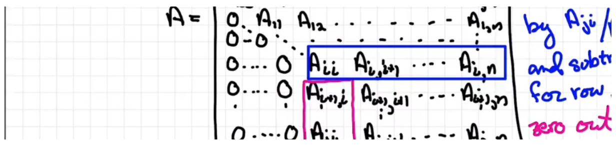
This construction of Gaussain elimination can be generalized to N-by-N matrices, and results in the code below. Note that the code below is not utilizing `NumPy`'s vectorization capability. As a result, it will be significantly slower than it could, especially in comparison to `np.linalg.solver`.

```
"""
Solve Ku = f by Gaussian elimination (no
pivoting)
"""
import numpy as np

def mysolve(K, f):
    """
    Args:
        K (numpy ndarray): square matrix
        f (numpy ndarray): right-hand side
vector

    Returns:
        numpy ndarray: u, solution to Ku=f
    """
    m,n = K.shape
    assert m == n, "Non-square matrix"

    u = np.zeros(n)

    # Extended matrix with the right-hand side
as last column
    A = np.zeros((n,n+1))
    A[:,0:n] = K
    A[:,n] = f

    # Elimination of nonzero elements below
the diagonal
    for i in range(n):
        assert A[i,i] != 0.0, "Zero pivot
detected"

        for j in range(i+1,n):
            Lji = A[j,i] / A[i,i]

            for k in range(i+1,n+1):
                A[j,k] = A[j,k] - Lji * A[i,k]

    # Back substitution
    u[n-1] = A[n-1,n]/A[n-1,n-1]

    for i in range(n-2,-1,-1):
        u[i] = A[i,n]

        for j in range(i+1,n):
            u[i] = u[i] -  A[i,j]*u[j]
```

Discussions

All posts sorted by recent activity

Previous
Next



eliminated all of the columns below the