State Vector Extraction

```python
Evaluate q(x) for a string in K (page 10, equation 42)
 def q(self, x):
     # If affine space has dimension zero then phase does not matter
     if (self.k == 0): return 0

     # x is a length n vector in basis of F_2^n
     # vecx is a length k vector in basis of L(K)

     # B is n*k 'basis matrix' with each row a length n basis vector
     # Let vecx and x be row vectors. Then solve equation B vecx = x+h
     B = self.G[:self.k].T
     vecx = np.linalg.lstsq(B, x + self.h)[0].astype(int) % 2

     # check result: should succeed if x in K
     if not np.allclose(np.dot(B, vecx) % 2, (x + self.h) % 2):
         raise LookupError("Input vector is not the affine space.")

     # Evaluate equation 42
     qx = self.Q
     qx += np.inner(self.D, vecx)

     for a in range(self.k):
         for b in range(a):
             qx += self.J[a, b]*vecx[a]*vecx[b]

     return qx % 8
```

```python
 # Coefficient for x in the superposition
 def coeff(self, x):
     # compute coefficient according to page 10, equation 46
     try: return np.power(2, -0.5*self.k) * np.exp(self.q(x) * 1j * np.pi/4)
     except LookupError: return 0  # if vector is not in affine space
```