

Quantum States

```
In[*]:= Quit[]
```

```
In[*]:= << Q3`
```

Basis States

You always start with choosing a symbol to refer to your quantum register (a collection of qubits).

```
In[*]:= Let[Qubit, S]
```

Here is the *computation basis* for qubit $S[1, \$]$.

```
In[*]:= bs = Basis[S[1, $]]
```

```
Out[*]=  
 $\{ |0_{S_1}\rangle, |1_{S_1}\rangle \}$ 
```

Construct a basis state manually.

```
In[*]:= v1 = Ket[S[1] -> 1]
```

```
Out[*]=  
 $|1_{S_1}\rangle$ 
```

This specification ends up with a seemingly different state.

```
In[*]:= v0 = Ket[S[1] -> 0]
```

```
Out[*]=  
 $|0_{S_1}\rangle$ 
```

Q3 automatically converts the form you enter to a more accurate form.

```
In[*]:= v0 // InputForm
```

```
Out[*]//InputForm=  
Ket[<|S[1, $] -> 0|>]
```

Superposition States

Consider again the computational basis for qubit $S[1, \$]$.

```
In[*]:= bs = Basis[S[1]]
```

```
Out[*]=  
 $\{ |0_{S_1}\rangle, |1_{S_1}\rangle \}$ 
```

Construct a superposition state by summing all elements in the computational basis.

```
In[*]:= new = Total@bs
```

```
Out[*]=
```

$$|0_{S_1}\rangle + |1_{S_1}\rangle$$

Of course, you can construct an arbitrary superposition using any complex numbers.

```
In[*]:= vec = 3 * Ket[S[1] -> 0] + (2 - I) * Ket[S[1] -> 1]
```

```
Out[*]=
```

$$3 |0_{S_1}\rangle + (2 - i) |1_{S_1}\rangle$$

Matrix representation: column-vector form

A state vector is often represented by a column vector of coefficients

```
In[*]:= vv = Matrix[vec, S[1]];
```

```
vv // MatrixForm
```

```
Out[*]//MatrixForm=
```

$$\begin{pmatrix} 3 \\ 2 - i \end{pmatrix}$$

Recover the vector expression by using `ExpressionFor`.

```
In[*]:= ExpressionFor[vv, S[1]]
```

```
Out[*]=
```

$$3 |0_{S_1}\rangle + (2 - i) |1_{S_1}\rangle$$

Bloch sphere: graphical representation

Just a list of *complex numbers* is difficult to understand the underlying physical meaning. Geometrical representation is often useful in this respect.

Convert a *single-qubit* state to a three-dimensional vector, which is called a *Bloch vector*.

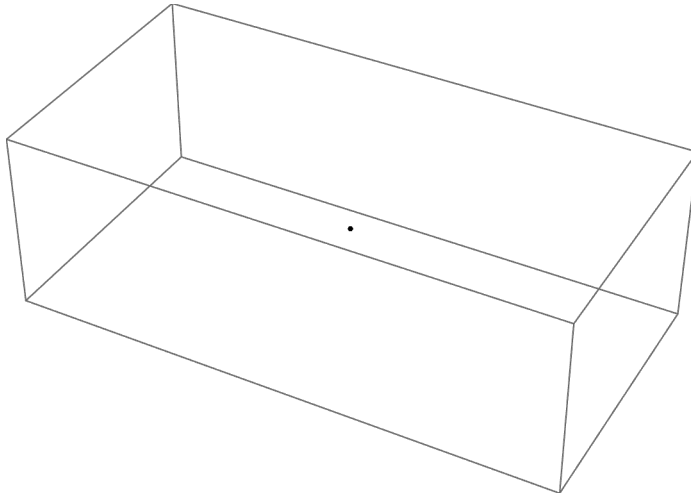
```
In[*]:= bb = BlochVector[vec]
```

```
Out[*]=
```

$$\left\{ \frac{6}{7}, -\frac{3}{7}, \frac{2}{7} \right\}$$

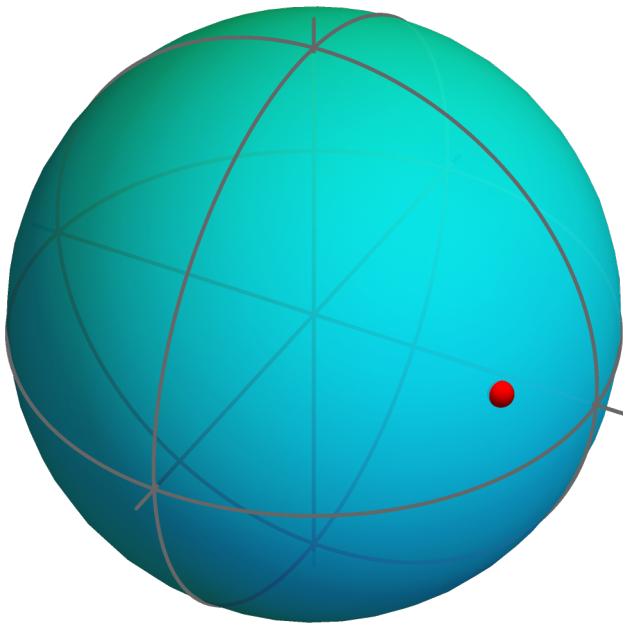
Now you can plot the vector in the three-dimensional (Euclidean) space.

```
In[*]:= Graphics3D[Point[bb], ImageSize → Medium]
Out[*]=
```



In many cases, it is simpler to plot the Bloch vector by using **BlochSphere**.

```
In[*]:= BlochSphere[{Red, Bead[bb]}, ImageSize → Medium]
Out[*]=
```



Separable State

When there are more qubits, the superposition states are richer.

Consider the computational basis for two qubits, $S[1, \$]$ and $S[2, \$]$.

```
In[*]:= bs = Basis[S[{1, 2}, $]]
Out[*]=
```

$$\{ |0_{S_1}0_{S_2}\rangle, |0_{S_1}1_{S_2}\rangle, |1_{S_1}0_{S_2}\rangle, |1_{S_1}1_{S_2}\rangle \}$$

```
In[*]:= vec = Total[bs]
Out[*]=

$$|0_{S_1}0_{S_2}\rangle + |0_{S_1}1_{S_2}\rangle + |1_{S_1}0_{S_2}\rangle + |1_{S_1}1_{S_2}\rangle$$

In[*]:= KetFactor[vec]
Out[*]=

$$(|0_{S_1}\rangle + |1_{S_1}\rangle) \otimes (|0_{S_2}\rangle + |1_{S_2}\rangle)$$

```

Entangled State

An exotic superposition state that even classical wave cannot explain.

```
In[*]:= new = Ket[] + Ket[S@{1, 2} -> 1] // KetRegulate
Out[*]=

$$|0_{S_1}0_{S_2}\rangle + |1_{S_1}1_{S_2}\rangle$$

In[*]:= out = KetFactor[new]
Out[*]=

$$|0_{S_1}0_{S_2}\rangle + |1_{S_1}1_{S_2}\rangle$$

```

Advanced Topic: Schmidt Decomposition

A *systematic* method to tell if a given state is separable or entangled.

Summary

Functions

- Ket
- Basis
- KetRegulate, DefaultForm
- Matrix, ExpressionFor
- BlochVector, BlochSphere, Bead
- KetFactor

Related Links

- Chapter 1 of the Quantum Workbook (2022, 2023).
- Tutorial: “Quantum States”
- Tutorial: “Quick Quantum Computing with Q3”