QUANTUM RANDOM WALK MAPLE SCRIPTS USER GUIDE

RAJARSHI DAS AND TORIN GREENWOOD

1. One Dimensional QRW for $k \times k$ unitary matrix with k chiralities

General User Inputs:

1. U is a $k \times k$ unitary matrix as determined by the user.

2. *chiralities* is a $1 \times k$ integer array containing the step sizes as determined by the user. For example, the j^{th} entry would contain the step size for chirality j.

3. *state* is a $n \times k$ matrix containing the most up-to-date amplitudes of the QRW. In this matrix, the (i, j) entry represents the amplitude at a generic location i and chirality j. The initial $1 \times k$ matrix is entered by the user and contains a 1 in the starting chirality column and 0 in other columns.

4. After the above matrices are entered the key procedure `TakeNSteps` (which uses many other procedures) can be run to generate the final amplitude matrix. This can then be converted to the probability matrix using the `AmpToProb` procedure. Finally, upon calling the `pointplot` Maple procedure the final image is generated.

The user defined matrices generate k, a static variable which represents the number of chiralities. Also, n is a dynamic variable which represents the number of possible locations for the particle and therefore grows as the number of steps increases.

Time Trials for executing entire sheet: (will vary depending on system)
10 steps: 1.05 s
50 steps: 1.36 s
100 steps: 2.75 s
200 steps: 11.28 s
500 steps: 127.61 s

Date: June 1, 2007.

The first procedure, `QCoinFlip`, produces an array that uses the unitary matrix to calculate the new amplitudes.

Inputs:
1. *state* (matrix with the current state of the QRW), $n \times k$ matrix where k equals the number of chiralities and n equals the number of possible locations
2. U, $k \times k$ unitary matrix

Output: A new $n \times k$ matrix.

Example:

$$state := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}$$

$$U := \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

$QCoinFlip(state, U)$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

Explanation:

Since the amplitudes are changed based on the unitary matrix, each element of the state matrix is multiplied by a respective element from the unitary matrix. This is done by multiplying an element in say row i and column j of the state matrix by column j of the unitary matrix. Then the k products are placed in the new matrix in row i, in order from column 1 to j.

The `Move` procedure enlarges the matrix and moves elements around to their new location.

Inputs:
1. *state* (matrix with the current state of the QRW), $n \times k$ matrix where k equals the number of chiralities and n equals the number of possible locations
2. *chiralities*, $1 \times k$ chiralities array where each chirality is named by its step size

Output: A new state matrix that is larger than the old one by `RangeSize` (This procedure finds the largest and smallest chiralities and uses those numbers to calculate the spread of the quantum random walk). The newly generated matrix has all its elements in their new and correct locations. The (i, j) entry of the old state matrix is the ($i+$ chirality of column j + minimum chirality, j) entry of the new matrix.

Example:

$state := \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

$chiralities := Array([1, -1, 0])$

$$\begin{bmatrix} 1 & -1 & 0 \end{bmatrix}$$

$Move(state, chiralities)$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 \end{bmatrix}$$

Explanation:

A new larger matrix is created and the elements in each column are moved up or down depending on the step size of its chirality. For example, the elements in the k^{th} column of the *state* matrix will be moved as many spaces as the integer in the k^{th} column of the chiralities matrix. **It is important to note that a positive step size means a downward movement and vice versa. Also, take into acount that the center, location 0, is shifted down based on the minimum chirality.**

The `TakeNSteps` Maple procedure puts together the above three procedures and repeats them for the number of steps.

Inputs:
1. U, $k \times k$ unitary matrix
2. *state* (matrix with the current state of the QRW), $n \times k$ matrix where k equals the number of chiralities and n equals the number of possible locations
3. *chiralities*, $1 \times k$ chiralities array where each chirality is named by its step size
4. N, integer number of steps

Output: The minimum and maximum possible locations as well as the final matrix that is transformed by all above procedures N number of times.

Example:

$U := \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

$chiralities := Array([1, -1, 0])$

$$\begin{bmatrix} 1 & -1 & 0 \end{bmatrix}$$

$state := Matrix([[1, 0, 0]])$

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$TakeNSteps(U, state, chiralities, 2)$

The minimum location is, -2
The maximum location is, 2

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 \end{bmatrix}$$

Explanation:
Earlier procedures are called and proper inputs are given to generate the final matrix. This is then repeated as many times as the number of steps.

The `AmpToProb` procedure produces an matrix with 2 columns: the left containing the location and the right containing the probability the particle may be at that location.

Inputs:
1. *state* (matrix with the current state of the QRW), $n \times k$ matrix where k equals the number of chiralities and n equals the number of possible locations

Output: The resultant matrix will have two columns. The first column will list the possible locations and the second column will display the probabilities. These probabilities should add up to 1.

Example:

$$state := \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 \end{bmatrix}$$

$AmpToProb(state)$

$$\begin{bmatrix} -2 & 0 \\ -1 & \frac{1}{2} \\ 0 & 0 \\ 1 & \frac{1}{4} \\ 2 & \frac{1}{4} \end{bmatrix}$$

$Explanation:$

The procedure begins by creating an empty 2-columned matrix. The two columns are then filled almost simultaneously. The left column is filled with the possible locations starting from smallest and going to largest and the right column is filled by adding the squares of the amplitudes in each row of the *state* matrix.

Finally, the Maple procedure `pointplot` is called to generate a plot using the probability matrix. Therefore it goes through all aforementioned procedures to generate an image as can be seen below in Figure 1.

Example:

$U := \begin{bmatrix} \frac{9}{11} & \frac{-6}{11} & \frac{2}{11} \\ \frac{-2}{11} & \frac{-6}{11} & \frac{-9}{11} \\ \frac{-6}{11} & \frac{-7}{11} & \frac{6}{11} \end{bmatrix}$

$$\begin{bmatrix} \frac{9}{11} & \frac{-6}{11} & \frac{2}{11} \\ \frac{-2}{11} & \frac{-6}{11} & \frac{-9}{11} \\ \frac{-6}{11} & \frac{-7}{11} & \frac{6}{11} \end{bmatrix}$$

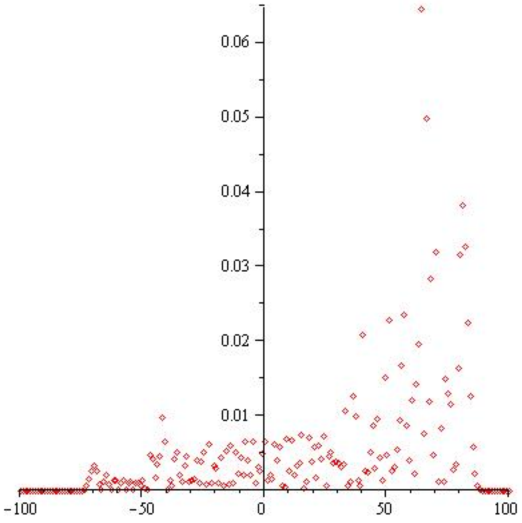
$chiralities := Array([1, -1, 0])$

$$\begin{bmatrix} 1 & -1 & 0 \end{bmatrix}$$

$state := Matrix([[1, 0, 0]])$

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$pointplot(AmpToProb(TakeNSteps(U, state, chiralities, 100)), color = red)$



Department of Mathematics, University of Pennsylvania, Philadelphia, PA 19104–6395
E-mail address: `rajdas@seas.upenn.edu` and `toringr@sas.upenn.edu`