Turning e^{itA} into arbitrary rotation gates (trotterization)

July 29, 2022

For 2×2 real matrix that is Hermitian (symmetric), $A = \begin{pmatrix} a & b \\ b & d \end{pmatrix}$, we want to apply a controlled unitary gate e^{iAt2^j}, where j is the position of the corresponding control qubit. We can break down A into a sum of I, X, Z matrices.

$$A = \begin{pmatrix} a & b \\ b & d \end{pmatrix} = \begin{pmatrix} 0 & b \\ b & 0 \end{pmatrix} + \begin{pmatrix} (a+d/2) & 0 \\ 0 & (a+d)/2 \end{pmatrix} + \begin{pmatrix} (a-d)/2 & 0 \\ 0 & (a-d)/2 \end{pmatrix} = bX + \frac{a+d}{2} I + \frac{a-d}{2} Z = \alpha I + \beta X + \gamma Z.$$

We can use the following matrix identities

$$e^{i\theta X} = R_x(-2\theta), \quad e^{i\theta Z} = R_z(-2\theta), \quad e^{i\theta I} = e^{i\theta} I$$

where the last gate just applies a global phase and thus does nothing.

One approximation (that uses more arbitrary rotation gates) can be found here:

$$e^{itA} \approx (e^{(it\beta/r)X} e^{(it\gamma/r)Z})^r = (R_x\left(\frac{-2t\beta}{r}\right) R_z\left(\frac{-2t\gamma}{r}\right))^r = R_x\left(\frac{-2t\beta}{r}\right) R_z\left(\frac{-2t\gamma}{r}\right) \dots R_x\left(\frac{-2t\beta}{r}\right) R_z\left(\frac{-2t\gamma}{r}\right).$$

For some integer r. So if $r = 2$, you would need to apply 4 arbitrary rotation gates for every bit in the input control register. Example with $r = 2$: e^{itA} can be simulated as