Title: Toward a quantum Fourier transform on SU(2).

Date: Jun 04, 2007 02:30 PM

URL: http://pirsa.org/07060023

Abstract: <span>Almost all known superpolynomial quantum speedups over classical algorithms have used the quantum Fourier transform (QFT). Most known applications of the QFT make use of the QFT over abelian groups, including Shorâ€™s well known factoring algorithm [1]. However, the QFT can be generalised to act on non-abelian groups allowing different applications. For example, Kuperberg solves the dihedral hidden subgroup problem in subexponential time using the QFT on the dihedral group. The aim of this research is to construct an efficient QFT on SU(2). Most of the progress in constructing QFTs has come from applying ideas from classical algorithms such as subgroup adapted bases. For example, Moore et al. &nbsp;have applied classical ideas from e.g. &nbsp;to build non-abelian QFTs. Applying these ideas to infinite groups such as SU(2) requires new tools. The function must be sampled or discretised in a way so as to minimise the error. I will present some ideas based on classical algorithms which may lead to a QFT over the group SU(2) for band limited functions. There are problems with making these algorithms unitary that must be addressed and efficient methods for calculating coefficients (cf. the controlled phase gates in the abelian case) must be found.</span>

Towards a QFT on SU(2)

Richard Low

Department of Computer Science
University of Bristol
UK

Joint work with Aram Harrow

Outline

1. Introduction
2. Basics of representation theory
3. Generalisation of FT to non-Abelian groups
4. Other FFT algorithms
5. Possible approaches for SU(2)
6. Summary

Introduction

- Almost all known superpolynomial speedups use the QFT
- Most use Abelian
- However, non-Abelian QFT has been useful too e.g. hidden subgroup problem on the dihedral group [1]
- Maybe the SU(2) QFT will help solve the HSP on SU(2).
- Might help to find a polynomial circuit for QFT on GL(n, q) solving open problem in [2]
- QFT can be used to construct Clebsch-Gordan transform

[1] G. Kuperberg, A subexponential-time quantum algorithm for the dihedral hidden subgroup problem, arXiv:quant-ph/0302112
[2] A. Ben-Aroya and A. Ta-Shma, Quantum expanders and the entropy difference problem, quant-ph/0702129

Basics of Representation Theory

- A representation can be thought of as a homomorphic map from the group to a set of matrices

$$\rho_\lambda : G \rightarrow V_\lambda \otimes V_\lambda^*$$

- E.g. defining representation

$$\rho(g_1)|g_2 >= |g_1 g_2 >$$

- A finite group has finitely many irreducible representations (irreps) with dimensions satisfying

$$\sum_{\lambda \in \hat{G}} d_\lambda^2 = |G|$$

with

$$d_\lambda = \dim V_\lambda$$

FT on Cyclic Group

The familiar cyclic group FT

$$\tilde{f}(y_k) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f(x_j) e^{2\pi i jk/N}$$

FT on Cyclic Group

The familiar cyclic group FT

$$\tilde{f}(y_k) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f(x_j) e^{2\pi i jk/N}$$

Can be thought of as a basis transformation

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle$$

$$U = \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} e^{2\pi i jk/N} |k\rangle\langle j|$$

FT on Cyclic Group

Evaluate f at the sample points

$$|f\rangle = \sum_{j=0}^{N-1} f(x_j)|j\rangle$$

FT on Cyclic Group

Evaluate f at the sample points

$$|f\rangle = \sum_{j=0}^{N-1} f(x_j)|j\rangle$$

FT is then

$$|\tilde{f}\rangle = \sum_{k=0}^{N-1} \tilde{f}(y_k)|k\rangle = U|f\rangle$$

FT on Cyclic Group

Evaluate f at the sample points

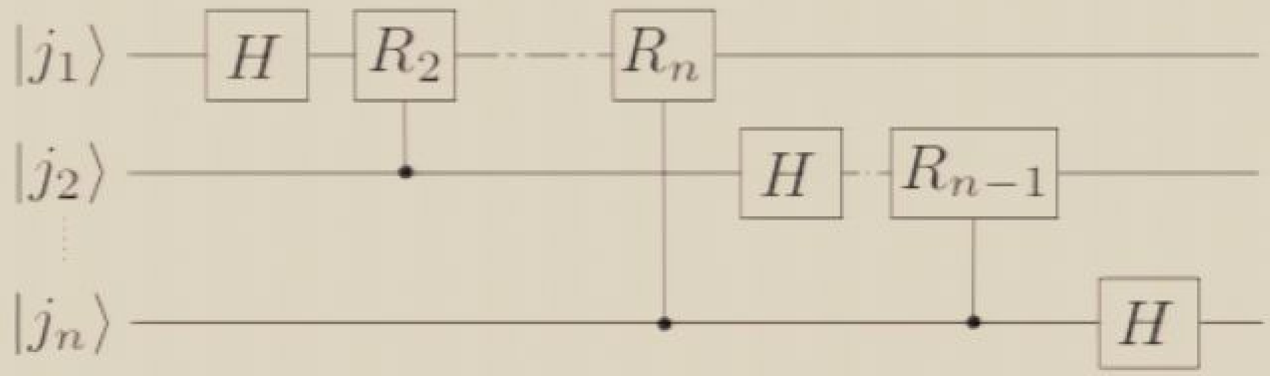
$$|f\rangle = \sum_{j=0}^{N-1} f(x_j)|j\rangle$$

FT is then

$$|\tilde{f}\rangle = \sum_{k=0}^{N-1} \tilde{f}(y_k)|k\rangle = U|f\rangle$$

Circuit:

$\Theta(\log^2 N)$



Non-Abelian FT

Can generalise to any finite group through isomorphism

$$\mathbb{C}[G] \cong \bigoplus_{\lambda \in \hat{G}} V_\lambda \otimes V_\lambda^*$$

Non-Abelian FT

Can generalise to any finite group through isomorphism

$$\mathbb{C}[G] \cong \bigoplus_{\lambda \in \hat{G}} V_\lambda \otimes V_\lambda^*$$

With corresponding change of basis

$$U = \sum_{g\in G} \sum_{\lambda \in \hat{G}} \sum_{i,j=1}^{\dim V_\lambda} \sqrt{\frac{\dim V_\lambda}{|G|}} \rho_\lambda(g)_{ij} |\lambda, i, j\rangle\langle g|$$

$$|g\rangle \rightarrow \begin{pmatrix} \left(\rho_{\lambda_0}(g)\right) & & \\ & \left(\rho_{\lambda_1}(g)\right) & \\ & & \ddots \end{pmatrix}$$

What is known so far

Group	Quantum	Classical
Cyclic	$O(\log^2 N)$	$O(N \log N)$
Symmetric	$O(\mathrm{polylog}(\lvert\mathrm{G}\rvert))$ [1]	$O(\lvert G\rvert \mathrm{polylog}(\lvert\mathrm{G}\rvert))$[2]
Dihedral	$O(\mathrm{polylog}(\lvert\mathrm{G}\rvert))$ [3]	$O(\lvert G\rvert \mathrm{polylog}(\lvert\mathrm{G}\rvert))$
$GL(k, \mathbb{F}_q)$	$\exp(O(\sqrt{\log \lvert G\rvert}))$[4]	$\lvert G\rvert^{1+\Theta(1/k)}$ [5]

[1] R. Beals, Quantum computation of Fourier transforms over symmetric groups, STOC, 1997
[2] P. Diaconis and D. Rockmore, Efficient computation of the Fourier Transform on finite groups, J AMS, 1990
[3] P. Høyer, Efficient Quantum Transforms, quant-ph/9702028
[4] C. Moore, D. Rockmore and A. Russel, Generic Quantum Fourier Transforms, quant-ph/0304064
[5] D. Maslen and D. Rockmore, Adapted diameters and the efficient computation of Fourier transforms on finite groups, ACM_SIAM Symposium on Discrete Algorithms, 1995

Infinite groups

- Can extend to infinite but compact groups
- Sum over group elements becomes integration w.r.t. some group invariant measure (e.g. the Haar measure).
- To implement, must *discretise* or sample

SU(2)

- Group of 2x2 unitary matrices with unit determinant
- $SO(3) \cong SU(2)/\{I, -I\}$
- Parameterise elements by *Euler angles*

$$g = R_x(\phi_2)R_z(\theta)R_x(\phi_1)$$

- So sample at fixed angles. Most algorithms use equiangular sampling

$$\theta_j = \pi(2j+1)/2N$$

- To convert integrals to sums, need *weights*

Weights

- Suppose f_1 and f_2 are polynomials of degree $\leq n$ and w is a weight function. Then

$$\int_a^b f_1(x) f_2(x) w(x) dx = \sum_{k=0}^{n-1} f_1(x_k) f_2(x_k) w_k$$

- The x_k are the *sample points* and the w_k are the *weights*
- If chosen correctly, the above integral is evaluated *exactly* by the sum (Gaussian quadrature)
- We can also *oversample* to make algorithms faster

Back to SU(2)

- Use a functional representation where elements act by rotation

$$T^l_{m_1,m_2}(\theta,\phi_1,\phi_2) = Q^l_{m_1,m_2}(\theta) e^{im_1\phi_1} e^{im_2\phi_2}$$

$$l \geq 0, |m_1|, |m_2| \leq l, \dim V_l = 2l+1 \qquad [1]$$

- Function Q is a sum of Jacobi polynomials, which are generalisations of the Legendre polynomials

- FT is then

$$\underbrace{\tilde{f}(l)}_{\text{rrep label}}{}_{\underbrace{m_1,m_2}_{\text{matrix indices}}} = \frac{1}{8\pi^2} \int f(\phi_2,\theta,\phi_1) T^l_{m_1,m_2}(\theta,\phi_1,\phi_2) \sin\theta d\phi_1 d\phi_2 d\theta$$

- Which we calculate by the sum

$$\tilde{f}(l)_{m_1,m_2} = \frac{1}{N} \sum_k w_k Q^l_{m_1,m_2}(\theta_k) \sum_{j_2} e^{im_2\phi_{2,j_2}} \sum_{j_1} e^{im_1\phi_{1,j_1}} f(\phi_{2,j_2},\theta_k,\phi_{2,j_2})$$

[1] D. K. Maslen & D. N. Rockmore, Generalized FFTs - A Survey of Some Recent Results, 1996

FT on SU(2)

- Note the sums over ϕ_1 and ϕ_2 are just the cyclic FT so can implement efficiently both classically and quantumly.

FT on SU(2)

- Note the sums over ϕ_1 and ϕ_2 are just the cyclic FT so can implement efficiently both classically and quantumly.
- Leaves us with the θ sum

$$\sum_k w_k Q^l_{m_1,m_2}(\theta_k) f_2(\theta_k)$$

- For $m_1 = m_2 = 0$ this is the Legendre transform. Will just consider this case since contains all essential details.

FT on SU(2)

- Note the sums over ϕ_1 and ϕ_2 are just the cyclic FT so can implement efficiently both classically and quantumly.
- Leaves us with the θ sum

$$\sum_k w_k Q^l_{m_1,m_2}(\theta_k) f_2(\theta_k)$$

- For $m_1 = m_2 = 0$ this is the Legendre transform. Will just consider this case since contains all essential details.
- Suppose the function f is *band limited*, that is

$$\forall l > b, \tilde{f}(l) = 0$$

- Then the sum is exact if we have at least $b+1$ sample points

Legendre Transform

- Legendre polynomials are the set of orthogonal polynomials on the interval [-1, 1]

$$P_0(x) = 1, P_1(x) = x, P_2(x) = \frac{1}{2}(3x^2 - 1)$$

- Any polynomial on this interval can be expanded in terms of them

$$\hat{f}(l) = \int_0^\pi f(\cos\theta) P_l(\cos\theta) \sin\theta d\theta$$

- If band limited this can be calculated exactly

$$\hat{f}(l) = \sum_k f(\cos\theta_k) P_l(\cos\theta_k) w_k$$

Legendre Transform (Ideal Version)

- If we use the optimal sampling we can represent the first b Legendre polynomials exactly using b sample points
- Since they are orthogonal the transformation is unitary (if we rescale)

$$U = \begin{pmatrix} \sqrt{w_0}P_0(x_0) & \sqrt{w_1}P_0(x_1) & \cdots & \sqrt{w_{b-1}}P_0(x_{b-1}) \\ \sqrt{w_0}P_1(x_0) & \sqrt{w_1}P_1(x_1) & \cdots & \sqrt{w_{b-1}}P_1(x_{b-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{w_0}P_{b-1}(x_0) & \sqrt{w_1}P_{b-1}(x_1) & \cdots & \sqrt{w_{b-1}}P_{b-1}(x_{b-1}) \end{pmatrix}$$

$$|f>= \begin{pmatrix} \sqrt{w_0}f(x_0) \\ \sqrt{w_1}f(x_1) \\ \vdots \\ \sqrt{w_{N-1}}f(x_{N-1}) \end{pmatrix}$$

Legendre Transform [1]

- Most classical algorithms oversample by a factor of 2
- So we now have b x 2b matrix, but we have freedom to choose the sample points
- Choose points

$$\theta_j = \pi(2j+1)/4b$$

- This means we can use the *cosine transform*

$$f(\cos\theta_j) = \sum_{n=0}^{2b-1} \sigma_n \cos(n\theta_j)$$

[1] D. Healy Jr., D. Rockmore, P. Kostelec and S. Moore, "FFTs for the 2-Sphere - Improvements and Variations", 2003

Legendre Transform

- Since

$$\cos^k \theta = \sum_{n=0}^{k} \sigma_n \cos(n\theta)$$

- The DCT of $P_l(\cos\theta_k)$ only has non-zero entries in the first $l+1$ points
- So we can make a *lowpass filter*

$$\mathcal{L}_p^N = \mathcal{C}_p^{-1} \mathcal{T}_p^N \mathcal{C}_N$$

- So we can *divide and conquer*: evaluate high and low degree coefficients separately

Legendre Transform

$$LT_M^L = \begin{pmatrix} \boxed{LT_{M/2}^L} & \\ & \boxed{LT_{M/2}^{L+M/2}} \end{pmatrix} \begin{pmatrix} \boxed{S_{L,M}^0} \\ \boxed{S_{L,M}^1} \end{pmatrix}$$

Legendre Transform

$$LT^L_M = \begin{pmatrix} \boxed{LT^L_{M/2}} & \\ & \boxed{LT^{L+M/2}_{M/2}} \end{pmatrix} \begin{pmatrix} \boxed{S^0_{L,M}} \\ \boxed{S^1_{L,M}} \end{pmatrix}$$

- Problem: $S^1_{L,M}$ is not unitary and can't even be made unitary by some weights

Legendre Transform

- Problem: does not generate weights

$$LT = \begin{pmatrix} P_0(x_0) & P_0(x_1) & \cdots & P_0(x_{2b-1}) \\ P_1(x_0) & P_1(x_1) & \cdots & P_1(x_{2b-1}) \\ \vdots & \vdots & \ddots & \vdots \\ P_{b-1}(x_0) & P_{b-1}(x_1) & \cdots & P_{b-1}(x_{2b-1}) \end{pmatrix}$$

- And the cosine transform does not work on weighted input

$$LT = \begin{pmatrix} \mathcal{C}P_0(x_0) & 0 & \cdots & 0 \\ \mathcal{C}P_1(x_0) & \mathcal{C}P_1(x_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}P_{b-1}(x_0) & \mathcal{C}P_{b-1}(x_1) & \cdots & \mathcal{C}P_{b-1}(x_{2b-1}) \end{pmatrix} \mathcal{C}$$

Summary

- Introduced generalised Fourier transforms
- Shown how QFT on SU(2) can be reduced to Legendre transform
- Approach for SU(2) uses cosine transform and divide and conquer by degree. Does not create weights.
- Can we make this quantum?
- Or do we need an entirely new approach?
- Where can we approximate?

Legendre Transform

- Since

$$\cos^k \theta = \sum_{n=0}^{k} \sigma_n \cos(n\theta)$$

- The DCT of $P_l(\cos\theta_k)$ only has non-zero entries in the first $l+1$ points
- So we can make a *lowpass filter*

$$\mathcal{L}_p^N = \mathcal{C}_p^{-1} \mathcal{T}_p^N \mathcal{C}_N$$

- So we can *divide and conquer*: evaluate high and low degree coefficients separately