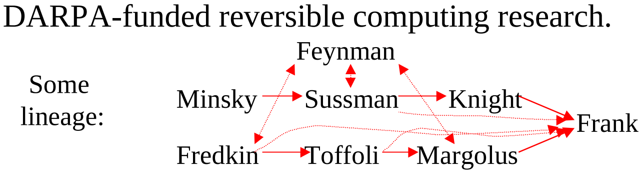
# Modeling Quantum Physics & Computation

Michael Frank (UF) & DoRon Motter (U.Mich.)

September 16, 2001

## Who We Are

- Dr. Michael Frank
  - MIT Ph.D. stud. & postdoc, 1996-97 & 1999.
    - Area exam studies on quantum computing.
    - DARPA-funded reversible computing research.




  - 1999-now: Head of Reversible & Quantum Computing group at UF's CISE dept.
    - http://www.cise.ufl.edu/research/revcomp

## Who We Are, cont.

- DoRon Motter
  - Undergrad in UF CISE dept., 1997-2000.
    - Coursework in CS + quantum mechanics.
    - Sr. highest honors thesis w. Dr. Frank, 2000.
  - Now a Masters student at U. Mich.
    - Advisor: Igor Markov, U. Mich.
    - DARPA-funded project on quantum logic systhesis

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

## A Grab-Bag of Topics

- Stable, reversible numerical simulations of wave mechanics.
- Visualization techniques for quantum algs.
- Linear-space classical simulations of quantum systems.
- Complexity models, classical + quantum parallelism.
- Models for systems engineering of scalable quantum computers.

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

## Simulating Wave Mechanics

- The basic problem situation:
  - Given:
    - A (possibly complex) initial wavefunction $\Psi_0 = \Psi(\vec{x}, t_0)$ in an $N$-dimensional position basis, and
    - a (possibly complex and time-varying) potential energy function $V(\vec{x}, t)$,
    - a time $t$ after (or before) $t_0$,
  - Compute:
    - $\Psi(\vec{x}, t)$
- Many applications...

## The Problem with the Problem

- An efficient technique (when possible):
  - Convert $V$ to a Hamiltonian $H$.
  - Find the energy eigenstates of $H$.
  - Project $\Psi$ onto eigenstate basis.
  - Multiply each component by $e^{iH(t-t_0)}$.
  - Project back onto position basis.
- Problem:
  - It may be intractable to find the eigenstates!
- We resort to numerical methods...

## History of Reversible Schrödinger Sim.

- Technique discovered by Ed Fredkin and William Barton in 1975.
- Subsequently proved by Feynman to conserve a certain probability measure.
- 1-D simulations in C/Xlib written by Frank at MIT in 1996.
- 1 & 2-D sims in Java, and proof of stability by Motter at UF in 2000.

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

## Overview

- Discrete update technique discovered in 1975 by Fredkin and Barton
- Known to give good simulations empirically
- Shown here that there is a mathematical basis for this
- Sample simulations shown in HSV

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

# Introduction

- Example of a reversible sequence of statements
  - $A \leftarrow A + f(B)$
  - $B \leftarrow B + f(A)$
- At each step either (A or B) changes
- This change depends only on the other variable (held constant)

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

# Introduction

- Undoing the computation
  - $A \leftarrow A + f(B)$
  - $B \leftarrow B + f(A)$
- Exactly reversible
  - Even after n steps of computation
  - Even if f cannot be computed exactly
  - Even if A, B are approximate values (finite precision)

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

# Introduction

- Undoing the computation
  - A ← A + f(B)
  - B ← B + f(A)
  - B ← B - f(A)
  - A ← A - f(B)
- Exactly reversible
  - Even after n steps of computation
  - Even if f cannot be computed exactly
  - Even if A, B are approximate values (finite precision)



# Introduction

- Centered approximation schemes

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2\Delta x}$$

$$f''(x_i) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{2(\Delta x)^2}$$

UNIVERSITY OF FLORIDA
UNIVERSITY OF MICHIGAN

## Introduction

- Centered approximation schemes

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2\Delta x}$$

$$f''(x_i) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{2(\Delta x)^2}$$

- Schrödinger Equation (1D)

$$i\hbar\frac{d}{dt}\psi(x,t) = -\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi(x,t) + V(x,t)\psi(x,t)$$

UNIVERSITY OF FLORIDA UNIVERSITY OF MICHIGAN

## Discrete Schrödinger Update

- Substituting centered approximation formulas gives

$$i\hbar\frac{\psi_m^{n+1} - \psi_m^{n-1}}{2k} \approx -\frac{\hbar^2}{2m}\frac{\psi_{m+1}^n - 2\psi_m^n + \psi_{m-1}^n}{2h^2} + V_m^n\psi_m^n$$

where $\psi_m^n \equiv \psi(x_m, t_n)$

UNIVERSITY OF FLORIDA UNIVERSITY OF MICHIGAN

## Discrete Schrödinger Update

- Substituting centered approximation formulas gives

$$i\hbar\frac{\psi_m^{n+1}-\psi_m^{n-1}}{2k} \approx -\frac{\hbar^2}{2m}\frac{\psi_{m+1}^n - 2\psi_m^n + \psi_{m-1}^n}{2h^2} + V_m^n\psi_m^n$$

where $\psi_m^n \equiv \psi(x_m, t_n)$

- "Central-time, central-space" scheme


## Discrete Schrödinger Update

- Substituting centered approximation formulas gives

$$i\hbar\frac{\psi_m^{n+1}-\psi_m^{n-1}}{2k} \approx -\frac{\hbar^2}{2m}\frac{\psi_{m+1}^n - 2\psi_m^n + \psi_{m-1}^n}{2h^2} + V_m^n\psi_m^n$$

where $\psi_m^n \equiv \psi(x_m, t_n)$

- "Central-time, central-space" scheme


$$\psi_m^{n+1} = \psi_m^{n-1} + i\left[\alpha\frac{k}{h^2}\left(\psi_{m+1}^n - 2\psi_m^n + \psi_{m-1}^n\right) + \beta k V_m^n \psi_m^n\right]$$

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

# Reversibility

- Real component at time n+1 depends on imaginary component at time n
- Similar to:
  - A = A + f(B)
  - B = B + f(A)

# Reversibility

- Real component at time n+1 depends on imaginary component at time n
- Similar to:
  - A = A + f(B)
  - B = B + f(A)
- Let $\psi = X + iY$, then

## Reversibility

- Real component at time n+1 depends on imaginary component at time n
- Similar to:
  - A = A + f(B)
  - B = B + f(A)
- Let $\psi = X + iY$, then

$$X_m^{n+1} = X_m^{n-1} - \left[\alpha \frac{k}{h^2}\left(Y_{m+1}^n - 2Y_m^n + Y_{m-1}^n\right) + \beta k V_m^n Y_m^n\right]$$

$$Y_m^{n+1} = Y_m^{n-1} + \left[\alpha \frac{k}{h^2}\left(X_{m+1}^n - 2X_m^n + X_{m-1}^n\right) + \beta k V_m^n X_m^n\right]$$



## Convergence and Stability

- Outline of the proof depends on Parseval's relation

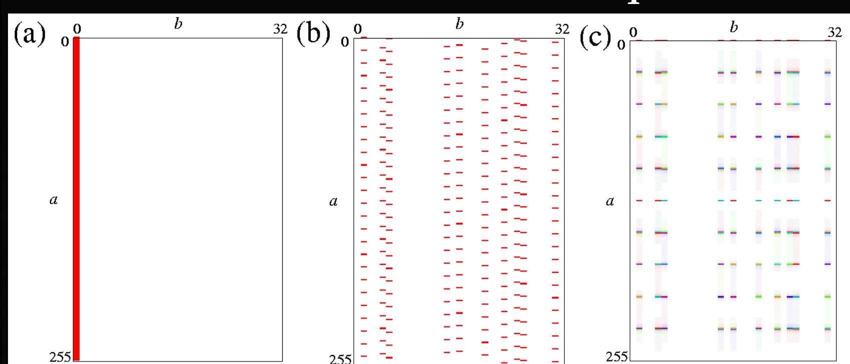
# Simulation of QC Algorithms

- Visualization:
  - Project states onto 2-D/3-D spaces
    - Corresponding to register pairs/triplets.
  - Use HSV color space to represent amplitudes.
  - Visualize gate ops with continuous color change.
- Simulation Efficiency:
  - Optimizations:
    - Track only states having non-zero amplitude.
  - Linear-space simulations of *n*-qubit systems.

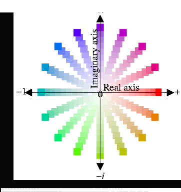
UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

# Visualization Technique




- Illustration: 3 stages of Shor's algorithm
- Register value → spatial position
- Phase angle → pixel color hue.
- Magnitude → pixel color saturation.




UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

## Linear-space quantum simulation

- A popular myth:
  - "Simulating an *n*-qubit (or *n*-particle) quantum system takes $e^{\Theta(n)}$ space (as well as time)."
- The usual justification:
  - It takes $e^{\Theta(n)}$ numbers even to *represent* a single $\Theta(n)$-dimensional state vector, in general.
- The hole in that argument:
  - Can simulate the statistical behavior of a quantum system w/o ever storing a state vector.

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

## The Basic Idea

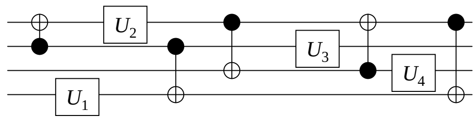
- Inspiration:
  - Feynman's *path integral* formulation of QED.
  - Gives the amplitude of a given final configuration by accumulating amplitude over all paths from initial to final configurations.
  - Each path consists of only a single $\Theta(n)$-coordinate configuration at each time, not a full wavefunction over the configuration space.
  - Can enumerate all paths, while only ever representing one path at a time.

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN

# Simulating Quantum Computations

- Given:
  - An $n$-qubit quantum computation, expressed as a sequence of 1-qubit gates and CNOT gates.




  - An initial state $s_0$ which is just a basis state in the classical bitwise basis, *e.g.* $|00000\rangle$.
- Goal:
  - Generate a final basis state with the same distribution as the quantum computer.

# Matrix Representation

- Consider each gate as rank-$2^n$ unitary matrix:
  - Each CNOT application is a 0-1 (permutation) matrix - a classical reversible bit-operation.
  - With appropriate row ordering, each $U_i$ gate application is block-diagonal, w. each 2×2 block equal to $U_i$.
  - We need never represent these full matrices!
  - The 1 or 2 nonzero entries in a given row can be located & computed immediately given the row id (bit string) and $U_i$.

# The Linear-Space Algorithm

- Generate a random coin $c \in [0,1]$. Let $p \leftarrow 0$.
- For each final $n$-bit string $y$ at time $t$,
  - Compute its amplitude $\Psi(y)$ as follows:
    - Generate its possible 1 or 2 predecessor strings $x_1$ (and maybe $x_2$) given the gate-op preceding $t$.
    - For each predecessor, compute its amplitude at time $t-1$ recursively using this same algorithm,
      - unless $t=0$, in which case $\Psi=1$ if $|x\rangle = s_0$, 0 otherwise.
    - Add predecessor amplitudes, weighted by entries.
  - Accumlate Pr[$y$]: $p \leftarrow p + ||\Psi(y)||^2$
  - Output $y$ and halt if $p>c$.

UNIVERSITY OF FLORIDA

UNIVERSITY OF MICHIGAN