Multidimensional linear diffusion for image enhancement on a type II quantum computer

By Mark W. Coffey* and Gabriel G. Colburn

Department of Physics, Colorado School of Mines, Golden, CO 80401, USA

We describe how multidimensional linear diffusion with application to image processing could be carried out on a hybrid classical–quantum computer. We present the quantum-lattice-gas-based algorithms, their effective finite difference approximations and representative simulation results. The methods for two-dimensional diffusion processing are particularly relevant to image enhancement tasks. We additionally demonstrate an extension to constrained linear diffusion that provides for non-uniform image smoothing. The simulation results and accompanying analysis support the utility of both classical and classical–quantum lattice gases for image enhancement. The diffusion modelling has applicability to many other fields including biological and physical science.

Keywords: quantum lattice gas algorithm; type II quantum computer; linear diffusion; anisotropic diffusion; constrained linear diffusion; image enhancement

1. Introduction

Diffusion processing has proved to be very useful for practical image enhancement, wherein the visual quality of an image is improved. We present an investigation of a method of carrying out such processing in a combined classical–quantum computing environment. While quantum Fourier and some quantum wavelet transforms are known (Nielsen & Chuang 2000), their direct application to image or signal processing is unclear. Motivated by real-world applications, we offer an alternative based upon the simulation of partial differential equations (PDEs).

Key to the efficient simulation of PDEs on a hybrid (or type II) processor is the execution of a quantum lattice gas algorithm (QLGA). In a type II architecture, nodes of phase-coherent, entangled qubits are linked by classical communication to nearest neighbours in a discrete lattice (Yepez 1998, 2001a,b, 2002). Generally, this type II approach has the advantage of requiring both less spatial and temporal entanglement than the usual (type I) quantum computing (QC) methods. Indeed, if the coherence time of the qubits should be sufficiently long, quantum error correction (Shor 1995; Steane 1996) would not be required. However, such error correction could be implemented within the nodes if needed or as a test bed of an interim QC technology.

* Author for correspondence (mcoffey@mines.edu).
A QLGA includes the sequential repetition of four main steps (Yepez 1998, 2001a,b, 2002; Berman et al. 2002; Vahala et al. 2003; Yepez et al. 2006). Firstly, initialization creates the quantum-mechanical initial state that corresponds to the initial probability distribution for a PDE to be solved. Secondly, a unitary transformation is applied in parallel to all the local Hilbert spaces in the lattice. Thirdly, in the measurement step, the quantum states of all the nodes are read out. Finally, these results are shifted or ‘streamed’ to neighbouring lattice sites, providing reinitialization of the lattice in the state which corresponds to the updated probability distribution.

In some QLGAs (e.g. Boghosian & Taylor 1996; Yepez & Boghosian 2002), the measurement step is omitted and the generally entangled quantum states are streamed. This places much stronger requirements on the QC hardware but can give an exponential speedup over classical simulation. It is possible to envision QLGAs with occasional measurements, with reduced speedup but relaxed hardware requirements.

The type II approach offers a means to effectively solve complex gas and fluid dynamics problems (Yepez 1998, 2001a, 2002; Vahala et al. 2003). Quantum lattice gas algorithms have been shown to solve the diffusion, Burgers, Boltzmann, Schrödinger and Dirac equations (Boghosian & Taylor 1996; Yepez 1998, 2001a,b, 2002; Yepez & Boghosian 2002; Vahala et al. 2003; Yepez et al. 2006). Implementation of these algorithms has mainly been carried out for one spatial dimension only. Very recently, multidimensional simulation for the Schrödinger equation has been performed (Sakai et al. 2005; Yepez et al. 2006).

Classical lattice Boltzmann simulations have proved highly valuable for fluid phenomena, including the presence of complex geometry (e.g. Harting et al. 2005). Quantum lattice gas algorithms may be written in an analogous discretized lattice Boltzmann equation form.

There exist quantum lattice gas algorithms that do not correspond to the average over some underlying lattice gas model in the Boltzmann approximation and their complexity analysis remains an open problem (Love & Boghosian 2006). The algorithms we consider do have a classical counterpart, and these provide an efficient implementation on parallel architectures. Therefore, our work strongly suggests the usefulness of both classical and hybrid classical–quantum lattice gases for image enhancement. Our methods are well suited to implementation on parallel computers, and, beyond this, to computational grids (e.g. Harting et al. 2005).

The appeal of our methods for classical computers is especially important as the computational advantages of purely quantum processors continue to be elusive. This is illustrated by the recent and extremely significant discovery that the quantum Fourier transform may be efficiently classically simulated (Aharonov et al. 2006; Browne 2007). With this finding, there appear to be new opportunities for novel algorithms for classical digital and/or digital–analogue computers. In at least some sense, it appears that the complexity characterization for the quantum Fourier transform had been otherwise anticipated (Alicki 2003).

We are interested in developing diffusion processing as a tool for image processing in future combined classical–quantum computing contexts. The diffusion of image intensities can be used for multiscale image enhancement,
selective smoothing, edge detection and as an aid to feature extraction. In this paper, we present simulation examples of multidimensional linear diffusion on a type II quantum computer. For the sake of definiteness, we mainly concentrate on two-dimensional diffusion which is mapped onto a rectangular lattice of nodes in a type II QC. However, our algorithm carries over to higher dimensions and we briefly describe this. We may stress that diffusion processes are common throughout engineering and scientific fields. Therefore, diffusion modelling is valuable among a very large number of disciplines, extending beyond signal and image processing.

After these sections of the paper dealing with two-, three- and n-dimensional linear diffusion, we present an important innovation for image applications. Namely, we demonstrate constrained linear diffusion that provides a method for non-uniform image smoothing. This extended algorithm fits naturally into our type II computing approach, combining a constraint condition on pixel value differences with the measurement step. In the concluding section, we discuss possible future generalizations, including nonlinear diffusion processing.

We mention the close connection of linear diffusion with the fundamental concept of linear scale space for images. A scale-space representation of an image is a special type of multiscale representation that comprises a continuous scale parameter and preserves the same spatial sampling at all scales. Linear scale space is an embedding of a given image \( I \) into a one-parameter family of images obtained by convolution with Gaussian kernels \( g \) of increasing width, \( I(\cdot; t) = g(\cdot; t) * I(\cdot; 0) \), with \( t \) the scale parameter. Equivalently, performing linear diffusion on an image gives linear scale space, as the Gaussian is the Green function for the diffusion equation on an infinite domain. As desired, no new local extrema are created in the scale-space family of images, which can also be seen to result from imposing conditions of causality, isotropy and homogeneity (Koenderink 1984). Since any \( n \)th-order derivative of \( I \) also satisfies the linear diffusion equation \( I_t = \nabla^2 I \), it follows that new zero-crossing curves in \( I_x \) cannot be created with increasing scale, and hence no new extrema.

We have implemented a variety of QLGAs for linear diffusion in one to four spatial dimensions in both MATLAB and MATHEMATICA. Of these, we report the MATHEMATICA implementations for two- and three-dimensional simulations. The constrained two-dimensional diffusion algorithm has been implemented in MATHEMATICA.

A preliminary implementation of a type II QC has been made in a liquid-state nuclear magnetic resonance (NMR) system (Pravia et al. 2002). These experiments used a chloroform \((^{13}\text{CHCl}_3\) solution, with hydrogen and a particular carbon nucleus serving as 2 qubits per node. Sixteen nodes of a one-dimensional lattice were created by the magnetic field of a gradient coil. Specific radio frequency (rf) pulses were applied for the unitary operations on qubits. It was possible to carry out 12 time steps of the algorithm for the linear one-dimensional diffusion equation. Improved system control should permit the execution of the algorithm for longer times with more fidelity.

Many further avenues for physical implementations of type II QCs should exist. These include other spin-based systems, coupled ion or neutral atom traps and flux-, charge- or phase-based superconducting qubit systems (Berns & Orlando 2005). An attraction of the type II approach is the possibility of realization significantly prior to large-scale type I quantum computers.
2. Type II QC for simulation of two-dimensional diffusion equation: basics

We consider a rectangular lattice of \( L \times M \) nodes, with 2 qubits per node, 
\[
|q_a(x, y, t)\rangle = \sqrt{f_a(x, y, t)}|1\rangle + \sqrt{1-f_a(x, y, t)}|0\rangle, \quad a=1, 2,
\]
where the occupancy probability \( f_a \) is defined below. The local wave function at each node is given as the tensor product 
\[
|\psi(x, y, t)\rangle = |q_1(x, y, t)\rangle \otimes |q_2(x, y, t)\rangle.
\]
The algorithm we describe may be thought of as a multidimensional extension of the ‘improved algorithm’ of Yepez (2001b).

Each node is connected by classical communication channels to each of its four nearest neighbours and the Hilbert space for one node has four basis states, 
\[
|00\rangle = |0\rangle \otimes |0\rangle, \ldots, |11\rangle = |1\rangle \otimes |1\rangle.
\]
The number operators \( n_1 \) and \( n_2 \) for the 2 qubits are given simply by
\[
n_1 = \text{diag}(1, 1, 0, 0),
n_2 = \text{diag}(1, 0, 1, 0).
\]
The occupancy probability of the \( j \)-th qubit at site \((x, y)\) at time \( t \) is defined as
\[
f_j(x, y, t) = \langle \psi(x, y, t)|n_j|\psi(x, y, t)\rangle,
\]
where \( j=1, 2 \). A density at the node at \((x, y, t)\) is defined as the sum of the occupancy probabilities,
\[
\rho(x, y, t) = f_1(x, y, t) + f_2(x, y, t).
\]
We next describe an appropriate sequence of quantum gate and classical shift operations applied across the lattice so that the function \( \rho \) is made to evolve in time as a solution of the linear diffusion equation
\[
\frac{\partial u}{\partial t} = \frac{1}{4\tau} \left( \Delta x^2 \frac{\partial^2 u}{\partial x^2} + \Delta y^2 \frac{\partial^2 u}{\partial y^2} \right).
\]
Although the diffusion is linear, it can be made anisotropic if desired by choosing the lattice spacings \( \Delta x \) and \( \Delta y \) in the \( x \) and \( y \) directions, respectively, to be unequal.

We divide an algorithmic time step of \( \tau \) into four substeps of length \( \tau/4 \). Each substep is a combination of a collision, measurement and streaming operation. The local collision operator \( U \) can be based upon the \( \sqrt{\text{SWAP}} \) gate (Yepez 1998, 2001a,b, 2002; Berman et al. 2002),
\[
U_{\sqrt{\text{SWAP}}} =
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \frac{1}{2}(1-i) & \frac{1}{2}(1+i) & 0 \\
0 & \frac{1}{2}(1+i) & \frac{1}{2}(1-i) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]
In equation (2.5), there is a \( U(2) \) subblock entangling states \( |01\rangle \) and \( |10\rangle \). With \( U \) as in equation (2.5) to perform the local collision \( |\psi'(x, y, t)\rangle = U|\psi(x, y, t)\rangle \), we then carry out the following substeps, wherein the state of the first qubit is not streamed.
We collide, measure the qubit states and stream qubit \( q_2 \) to the left one lattice unit, keeping qubit \( q_1 \) the same. We then collide, measure and stream \( q_2 \) to the right one lattice unit. This is followed by collision, measurement and streaming of \( q_2 \) up one place and then collision, measurement and streaming of \( q_2 \) down one place in the lattice.

The above sequence suffices for our current purposes. We could also think of a more symmetric algorithm with eight substeps, in which the second four substeps shift only the state of the first qubit.

By assuming that the occupation probabilities do not deviate too much from the equilibrium values \( f_1^{eq} = f_2^{eq} = \rho/2 \), we may develop a finite difference approximation for the evolution of the density. We write the finite differences for each of the four substeps and then add these, giving

\[
\rho(x, y, t + \tau) - \rho(x, y, t) = -\frac{1}{2} \rho(x, y, t) + \rho(x, y, t + \tau/4) + \rho(x, y, t + \tau/2) + \rho(x, y, t + 3\tau/4) + \frac{1}{2} \rho(x + \Delta x, y, t) + \rho(x - \Delta x, y, t + \tau/4) + \rho(x, y + \Delta y, t + \tau/2) + \rho(x, y - \Delta y, t + 3\tau/4). \tag{2.6}
\]

By substituting the four substep equations at \( x \pm \Delta x \) and \( y \pm \Delta y \), we can simplify equation (2.6) to be independent of the intermediate times

\[
\rho(x, y, t + \tau) - \rho(x, y, t) = \frac{1}{16} \left[ -12 \rho(x, y, t) + 2 \rho(x, y - \Delta y, t) + 2 \rho(x, y + \Delta y, t) + 2 \rho(x - \Delta x, y, t) + \rho(x - \Delta x, y + \Delta y, t) + \rho(x + \Delta x, y, t) + \rho(x + \Delta x, y - \Delta y, t) + \rho(x + \Delta x, y + \Delta y, t) \right]. \tag{2.7}
\]

We then Taylor expand the right-hand side of this equation up to second order in \( \Delta x \) and \( \Delta y \) and obtain

\[
\rho(x, y, t + \tau) - \rho(x, y, t) = \frac{1}{4} \left( \frac{\partial^2 \rho}{\partial x^2} \Delta x^2 + \frac{\partial^2 \rho}{\partial y^2} \Delta y^2 \right) + O(\Delta x^4) + O(\Delta y^4) + O(\Delta x^2 \Delta y^2). \tag{2.8}
\]

Similarly expanding the left-hand side of this equation to lowest order in \( \tau \) gives the linear diffusion equation (2.4). The algorithm is unconditionally stable since the collision operator \( U \) is unitary by construction. The error terms in equation (2.8) show that the algorithm is at least second-order convergent.

### 3. Two-dimensional simulation examples

We now illustrate the above algorithm for two-dimensional linear diffusion with various initial conditions. Throughout, periodic boundary conditions are employed. Unless described otherwise, we take a square lattice of stated size with unit lattice spacing, \( \Delta x = \Delta y = \ell = 1 \).
As a first example, we consider a Gaussian distribution centred over the square lattice
\[ g(x, y, t) = \frac{1}{2} + \frac{1}{4} \frac{\sigma_0}{\sigma^2(t)} \exp\left( -\left[ \frac{(x - L/2)^2 + (y - L/2)^2}{\sigma^2(t)} \right] \right). \] (3.1)
where \( \sigma(t) \equiv \sqrt{\sigma_0^2 + 4Dt} \). We take \( L = 64 \), the width parameter \( \sigma_0 = 0.1L \), and the diffusion constant \( D = \ell^2 / 4\tau \). The two-dimensional diffusion algorithm is initialized with the values \( g(x, y, 0)/2 \) for each of the occupancy probabilities \( f_1 \) and \( f_2 \). Figure 1 shows a sequence of superposed plots at various times \( 10\tau, 20\tau, 30\tau, \) and \( 40\tau \) of both the numerical and exact solution along the midline \((L/2, y)\) and the numerical solution is seen to properly track the latter very well.

A two-dimensional Dirac delta distribution initial condition provides a rather stringent test of a lattice gas algorithm. This distribution is in fact a certain \( t \to 0, \sigma_0 \to 0 \) limit of the distribution \( g(x, y, t) - 1/2 \) of equation (3.1). Owing to its simple and rotationally symmetric structure, the Dirac delta distribution is very convenient for examining the effects of collision and streaming operations.

In this example, there is initially only a unit impulse at \((L/2, L/2)\) so that initially \( f_1 = f_2 = 1/2 \) only at \((x_0, y_0) = (L/2, L/2)\) and \( f_1 = f_2 = 0 \) everywhere else in the lattice. The Green function for the linear diffusion equation in Euclidean space \( \mathbb{R}^n \) is given by
\[ G(\mathbf{x} - \mathbf{x}_0, t) = \frac{1}{(4\pi Dt)^{n/2}} \exp\left( -\sum_{i=1}^{n} \frac{(x_i - x_{0i})^2}{4Dt} \right). \] (3.2)
For a finite domain, the Green function may be constructed using, for instance, Fourier series to incorporate periodic or other boundary conditions. By taking equation (3.2) with \( n = 2 \) as the exact solution for the square lattice, we commit only an exponentially small error near the boundary. The corresponding reference and numerical solutions along the midline \((L/2, y)\) are given in figure 2. Again \( L = 64, \ell = 1 \) and \( \tau = 1 \). Only at the earliest times do the numerical solution and equation (3.2) differ by any more than the scale of the graphical resolution.
An example well suited to periodic boundary conditions is the sinusoidal function
\[ r(x, y, t) = \frac{1}{2} + \frac{1}{4} e^{-\Gamma t} \sin \left( \frac{2\pi}{L} (x + y) \right), \] (3.3)
with attenuation constant \( \Gamma \equiv D (2\pi / L)^2 \). Equation (3.3) is an instance of a separated Fourier series solution with a single wavenumber \( k = 2\pi / L \). Since \( \Gamma \propto k^2 \), this solution is sensitive to the lattice size \( L \). The numerical and exact solutions for \( L = 32 \) and \( D = 1/4 \) are plotted jointly at successive times \( 8\tau, 32\tau, 64\tau \) and \( 128\tau \) versus \( y \) for \( x = L/2 \) in figure 3 and show very good agreement. Note that the periodic boundary condition on \( \rho \) enforces the numerical solution to also have periodic derivatives on the edge of the square. The particular exact solution (3.3) shares this property. Otherwise, a discrepancy between the exact and numerical solutions would exist at the edges that would grow with increasing time.

As a final example, we show a solution that illustrates both separation of variables and superposition to satisfy a zero and periodic boundary condition on
where erf is the error function. This solution results from a piecewise constant initial condition. (When the height of this initial condition is taken to infinity while its area shrinks to zero such that the enclosed volume remains constant, the two-dimensional Dirac delta distribution may be obtained.) The periodicity of solution (3.4) may be immediately verified by putting \( x/xb \) and \( y/ya \). Then a shift of the summation indices \( jx/jx1 \) and \( jy/jy1 \) gives back \( r(x, y, t) \).

Solution (3.4) may be obtained by convolving the \( n=2 \) Green function of equation (3.2) with the periodic initial condition \( r(x, y, 0) = 1 \) on \( 0 \leq x, y \leq a \) and \( r(x, y, 0) = 0 \) for \( a < x, y < b \). A change of variable is made in the two-dimensional convolution and the definition for the integral representation of the error function, \( \text{erf}(x) = (2/\sqrt{\pi}) \int_0^x e^{-t^2} dt \), is applied. For figures 4 and 5, we took \( a=2, b=4, \ell = b/(L-1) \) with \( L=64 \) and \( \tau=1 \). Surface plots of the numerical solution are given in the sequence in figure 4, and a plot comparing the numerical and exact solution along \( x=1 \) at a later time \( t=256\tau \) is given in figure 5. We found that just taking the \( jx, jy = -1, 0 \) and 1 terms of equation (3.4) provided an accurate approximation of the exact solution for comparison purposes. The numerical solution tracks very well the exact solution on the square domain.

**Figure 4.** Sequence of surface plots at times \( \tau, 16\tau, 32\tau, 48\tau \) and \( 64\tau \) of the numerical solution of two-dimensional diffusion with the initial condition \( r(x, y, 0) = 1 \) on \( 0 \leq x, y \leq 2 \) and \( r(x, y, 0) = 0 \) for \( 2 < x, y < 4 \), and periodically repeated. The lattice size is \( L=64 \) and spacing \( \ell = 4/63 \).
Asymptotic approximations of solution (3.4) can be made in various regimes using known analytic forms (e.g. Gradshteyn & Ryzhik 1980). Owing to the small value of the diffusion constant $D \approx 10^{-3}$ in the example, the large argument form of the error function, $\text{erf}(x) = 1 - (1/\sqrt{\pi}x)e^{-x^2}[1 + O(1/x^2)]$ as $x \to \infty$, can be approximately used. For illustration using only the $j_x = j_y = 0$ term of equation (3.4), this results in

$$\rho(x, y, z, t) \approx \frac{D}{\pi} \left[ e^{-(a-x)^2/4Dt} - \frac{e^{-x^2/4Dt}}{x} \right] \left[ e^{-(a-y)^2/4Dt} - \frac{e^{-y^2/4Dt}}{y} \right]. \quad (3.5)$$

The later time results of the example are not strictly asymptotic, but equation (3.5) helps to partly explain the evolution to a premultiplied two-dimensional Gaussian-like density. As expected for a linear parabolic PDE, sharp changes in the initial condition, in this case discontinuities, are rapidly mollified.

4. Dimension $n \geq 3$ simulation

The algorithm for two-dimensional linear anisotropic diffusion that we have described and illustrated may be extended to any number of Cartesian dimensions with cubic or hypercubic lattices. For each additional dimension, we apply two more steps of collision, measurement and streaming of the second qubit, maintaining the state of the first qubit. For instance, in three dimensions, we split an overall time step of $\tau$ into six substeps, with the last two substeps performing streaming of the second qubit in the $\pm z$ direction. Writing out the successive effective finite difference approximations, the net result extending equation (2.8) with the leading order error terms is

$$\rho(x, y, z, t + \tau) - \rho(x, y, z, t) = \frac{1}{4} \left( \frac{\partial^2 \rho}{\partial x^2} \Delta x^2 + \frac{\partial^2 \rho}{\partial y^2} \Delta y^2 + \frac{\partial^2 \rho}{\partial z^2} \Delta z^2 \right) + O(\Delta x^4) + O(\Delta y^4)$$

$$+ O(\Delta z^4) + O(\Delta x^2 \Delta y^2) + O(\Delta x \Delta z^2) + O(\Delta y^2 \Delta z^2). \quad (4.1)$$

Figure 5. Comparison of the exact (solid curve) and numerical (triangles) solutions for the two-dimensional diffusion problem with the exact solution given in equation (3.4) in terms of error functions versus $y$ at $x = 1$ at time $t = 256\tau$. 

Asymptotic approximations of solution (3.4) can be made in various regimes using known analytic forms (e.g. Gradshteyn & Ryzhik 1980). Owing to the small value of the diffusion constant $D \approx 10^{-3}$ in the example, the large argument form of the error function, $\text{erf}(x) = 1 - (1/\sqrt{\pi}x)e^{-x^2}[1 + O(1/x^2)]$ as $x \to \infty$, can be approximately used. For illustration using only the $j_x = j_y = 0$ term of equation (3.4), this results in

$$\rho(x, y, z, t) \approx \frac{D}{\pi} \left[ e^{-(a-x)^2/4Dt} - \frac{e^{-x^2/4Dt}}{x} \right] \left[ e^{-(a-y)^2/4Dt} - \frac{e^{-y^2/4Dt}}{y} \right]. \quad (3.5)$$

The later time results of the example are not strictly asymptotic, but equation (3.5) helps to partly explain the evolution to a premultiplied two-dimensional Gaussian-like density. As expected for a linear parabolic PDE, sharp changes in the initial condition, in this case discontinuities, are rapidly mollified.

4. Dimension $n \geq 3$ simulation

The algorithm for two-dimensional linear anisotropic diffusion that we have described and illustrated may be extended to any number of Cartesian dimensions with cubic or hypercubic lattices. For each additional dimension, we apply two more steps of collision, measurement and streaming of the second qubit, maintaining the state of the first qubit. For instance, in three dimensions, we split an overall time step of $\tau$ into six substeps, with the last two substeps performing streaming of the second qubit in the $\pm z$ direction. Writing out the successive effective finite difference approximations, the net result extending equation (2.8) with the leading order error terms is

$$\rho(x, y, z, t + \tau) - \rho(x, y, z, t) = \frac{1}{4} \left( \frac{\partial^2 \rho}{\partial x^2} \Delta x^2 + \frac{\partial^2 \rho}{\partial y^2} \Delta y^2 + \frac{\partial^2 \rho}{\partial z^2} \Delta z^2 \right) + O(\Delta x^4) + O(\Delta y^4)$$

$$+ O(\Delta z^4) + O(\Delta x^2 \Delta y^2) + O(\Delta x \Delta z^2) + O(\Delta y^2 \Delta z^2). \quad (4.1)$$

Hence, in the continuum limit, we obtain
\[
\frac{\partial u}{\partial t} = \frac{1}{4\tau} \left( \Delta x^2 \frac{\partial^2 u}{\partial x^2} + \Delta y^2 \frac{\partial^2 u}{\partial y^2} + \Delta z^2 \frac{\partial^2 u}{\partial z^2} \right),
\]
the three-dimensional linear anisotropic diffusion equation. Again, the algorithm is at least second-order convergent.

Similarly carrying out the algorithm for \( R^2 \) with streaming of the second qubit only, we obtain the \( n \)-dimensional anisotropic linear PDE
\[
\frac{\partial u}{\partial t} = \frac{1}{4\tau} \left( \Delta x_1^2 \frac{\partial^2 u}{\partial x_1^2} + \Delta x_2^2 \frac{\partial^2 u}{\partial x_2^2} + \cdots + \Delta x_n^2 \frac{\partial^2 u}{\partial x_n^2} \right).
\]

We have implemented our algorithm for isotropic linear diffusion in three and four dimensions. Our test cases in three dimensions include the three-dimensional Dirac delta distribution, whose corresponding exact solution for an infinite domain is given by equation (3.2) with \( n=3 \). We have also developed and used the \( n \)-dimensional generalization of the Gaussian distribution, extending equation (3.1). For \( n \) dimensions, we have
\[
g(x - x_0, t) = \frac{\sigma_0}{\sigma^n(t)} \exp \left( -\sum_{i=1}^{n} \frac{(x_i - x_{0i})^2}{\sigma^2(t)} \right).
\]

This function also solves the PDE \( u_t = D \nabla^2 u \). We have performed simulations on cubic lattices with smaller values of \( L \) (typically \( L=32 \)), \( x_0 = L/2 \), and verified the numerical solution on the central plane \( x=L/2 \). In the case of the initial condition \( g(x - x_0, 0) \) from equation (4.4), we again took the width parameter \( \sigma_0 = 0.1L \).

We may also note that the \( n \)-dimensional extensions of equations (3.3) and (3.4) are easily written. In the former case, the argument of the sine function becomes simply the sum \( \sum_{i=1}^{n} x_i \), and for the latter we take an \( n \)-fold product over sums of differences of error functions.

5. Constrained linear diffusion

To be of most use to image-processing tasks, we would like to have the ability to perform selective image smoothing, and linear diffusion alone will not provide this. In this section, we describe and demonstrate a technique that gives non-uniform image smoothing within our type II computing approach. We combine a linear diffusion step with a constraint condition step. Since measurement of qubit states is part of a quantum lattice gas algorithm, the incorporation of the constraint step fits well as part of the overall algorithm. An example of the use of constrained linear diffusion for colour image dequantization with classical computing is given in Keysers et al. (2006).

Suppose that our original image is \( I_0 \). We now perform at iteration \( j, j=1, 2, \ldots \) on image \( I_j \) (i) two-dimensional linear diffusion, giving the image \( I_j' \), followed by (ii) the pixelwise constraint
\[
I_{j+1}(x, y) = \max[I_0(x, y) - \delta, \min(I_0(x, y) + \delta, I_j'(x, y))].
\]

Here, \( \delta \) is a processing parameter. It can be chosen according to the distribution of pixel values or other properties of an image. For instance, \( \delta \) can be taken as proportional to the quantization level for pixel intensities. For greyscale images with pixel values in the range 0 to \( 2^8 - 1 \), we could choose \( \delta \approx 2^{-8} \).
Note that the measurement and constraint step in the above algorithm effectively gives us a form of nonlinear operation within the processing. This is an advantage of the type II computing approach. With a pure quantum processor, this would not be possible. With the constraint in place, image regions with similar pixel intensities are diffused while regions with large differences such as edges remain hardly touched. This is highly attractive for image processing as the edge information is preferentially preserved.

Trial experience with this constrained linear diffusion algorithm indicates that 10–100 iterations are effective. This fairly modest number of iterations means that practical image sizes (of a few hundred by a few hundred pixels) could be routinely processed.

Our implementation of constrained two-dimensional diffusion has been carried out in MATHEMATICA for images with 8-bit intensity values. Each of the occupancy probabilities $f_j(x, y, 0)$ for the 2-qubit algorithm is initialized as one-half of the pixel intensity divided by 255. When displaying intermediate or final image processing results, the image values are multiplied by 255 in order to use the MATHEMATICA graphics functions.

In order to have a benchmark for our constrained diffusion algorithm, we have taken a bitmap input image of the histogram-equalized image of fig. 2b of Keysers et al. (2006). This input image contains a rectangular region of smooth greyscale variation, rectangular striped regions and a sample alphabetic text region. On this input, we have run 50 steps of the above algorithm with $\delta = 0.025$. The left column of output images at successive iterations in figure 6 shows that the features of the input are preserved over time. This is in stark contrast to running linear diffusion alone, where the image is quickly uniformly blurred. In order to bring out the pixel changes during the constrained diffusion processing in figure 6a,c,e, we show the difference between the original image and the result of the constrained processing at the corresponding iteration. According to the sequence of images in figure 6b,d,f, we see that the desirable property of intraregion smoothing has occurred. For instance, in the largest single rectangular region in the top of the image, the original greyscale stepping has been smoothed. This has occurred while the striping and textual information in the rest of the image has been maintained. Therefore, this algorithm should be a viable candidate for image pre- or post-processing or image enhancement on a type II quantum computer.

We have tried the constrained diffusion algorithm on other input images with similar evidence of intraregion smoothing.

6. Discussion and summary

We have presented algorithms for, and demonstrations of, multidimensional diffusion processing on a type II (or hybrid) quantum computer with 2 qubits per node. We gave the effective finite difference approximations satisfied by the density formed from the sum of qubit occupancy probabilities. We used rectangular lattices or their higher-dimensional analogues, but the lattice spacing in each Cartesian direction need not be the same.

Various types of anisotropic diffusion have proven very useful for image processing, partly motivating this investigation. The basic idea of anisotropic diffusion is to diffuse intensities along edges of objects that appear within an
image while not diffusing (or even enhancing the contrast) along directions that are perpendicular to edges.

The analysis and processing of image data are an important and ubiquitous industry. In addition, diffusion, reaction–diffusion and advection–diffusion processes are common throughout science and engineering fields.

Of direct use for image enhancement purposes is the ability to selectively smooth regions of an image. We presented and illustrated an algorithm for constrained linear diffusion in two spatial dimensions that already demonstrates this capability on a type II computing architecture. Another direction for future research is to explore nonlinear diffusion (Perona & Malik 1987, 1990; Whitaker & Pizer 1991; Alvarez et al. 1992; Catte et al. 1992; Cottet & Germain 1993; Whitaker 1993; El-Fallah & Ford 1994). This is also attractive for QLFGAs because they are able to simulate nonlinear phenomena, whereas purely quantum methods are not.

Figure 6. (a,c,e) A sequence of images resulting at iterations 1, 25 and 50 from the constrained diffusion algorithm with the constraint condition given in equation (5.1) with parameter $\delta=0.025$. (b,d,f) The difference between the original image and the processed image at (a,c,e).
We have also begun analysis and numerical experimentation with quantum lattice gas algorithms for simulation of the backward heat equation $I_t = -\nabla^2 I$ in one and two spatial dimensions. Despite the initial value problem associated with this PDE being ill posed, it has nonetheless found application to image sharpening and restoration. Image sharpening is an inverse problem, with the typical instability associated with backward diffusion. With proper stabilization or regularization, backward diffusion can be a very useful technique (e.g. Lee & Kang 2004). We again have an instance in which type II QC would be advantageous for image processing.

For two or higher dimensions, there are much greater numbers of diffusion processes than in one dimension, in which the boundary conditions are simply given at points. For two dimensions, in particular, one could consider other type II lattices. For example, it may be possible to employ a lattice built from hexagons and with nodes containing 2, 3 or 6 qubits. The collision operator would need to be extended to such situations and need not give a symmetric mixing of occupancy probabilities.

A concatenated time sequence of images provides a three-dimensional space–time in which to consider image processing. By discriminating changing features in such data, one could detect moving objects. The various diffusion attributes of dimension, isotropy or anisotropy, forward or backward, and linear or nonlinear processes provide opportunities for exploring aspects of image processing on a type II architecture.

The early NMR experiments which qualitatively demonstrate the feasibility of implementing one-dimensional diffusion (Pravia et al. 2002) and the Burgers equation (Chen et al. 2006) are encouraging. They provide some evidence that the path to large hybrid classical–quantum computers may be realizable well before large-scale type I quantum computers. In addition, liquid-state NMR realizations of an ensemble of sufficient size would exceed the computational capacity of classical digital computers. Indeed, type II NMR-based computing provides an intriguing combination of QC with classical molecular computing. If one could sufficiently well output the diagonal elements of the density matrix by sampling from the computing ensemble, then Markov chain problems could be attacked that are outside the capability of classical digital computers. With current liquid-state NMR computing using $10^{18}$ molecules, Markov chain problems with up to $2^{60}$ states could in principle be addressed. This, in turn, implies that there is a range of NMR-computable problems of size $2^{30} - 2^{60}$ that lie beyond the capabilities of conventional digital computers (Love & Boghosian 2006).

This work was supported by Air Force contract number FA8750-04-1-0298. A US patent is pending on various aspects of the processing described herein.

References


