TIME DEPENDENT DIFFUSION BY MONTE CARLO

M. Ragheb
3/16/2013

INTRODUCTION

We consider the application of the Monte Carlo method to time dependent non stationary problems. As an example we consider the diffusion equation:

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} \tag{1}$$

which can represent the time dependence of a variety of heat and mass transport problems with α as the diffusivity coefficient. The solution of this equation is a function u, which depends on both the spatial coordinates and on time. Consequently its solution depends on the identification of appropriate initial and boundary conditions.

Let us consider a lattice that is inscribed in the domain D in which it is required to find the solution u(P, t) at point P, which satisfies the Dirichlet boundary condition on the boundary C:

$$u\big|_B = f(C) \tag{2}$$

or the Neumann boundary condition on C:

$$\left.\frac{\partial u}{\partial n}\right|_B = 0 \tag{3}$$

or a mixed type of boundary condition:

$$\left.\frac{\partial u}{\partial n}\right|_B = k\,u\big|_B + f(C) \tag{4}$$

as well as the initial condition:

$$u\big|_{t=0} = g(P) \tag{5}$$

TIME DEPENDENT RANDOM WALK

We consider a two dimensional form of Eqn. 1 as:

$$\nabla^2 u(x,y,t) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{1}{\alpha}\frac{\partial u}{\partial t} \tag{6}$$

We write the finite difference approximation of the Laplacian ∇^2 operator in the x direction:

$$\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) \approx \frac{\left(\frac{\Delta u}{\Delta x}\right)_+ - \left(\frac{\Delta u}{\Delta x}\right)_-}{\Delta x} \approx \frac{\left(u_{i+1,j,k} - u_{i,j,k}\right) - \left(u_{i,j,k} - u_{i-1,j,k}\right)}{\left(\Delta x\right)^2} \tag{7}$$

Similarly, in the y direction:

$$\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y}\right) \approx \frac{\left(\frac{\Delta u}{\Delta y}\right)_+ - \left(\frac{\Delta u}{\Delta y}\right)_-}{\Delta y} \approx \frac{\left(u_{i,j+1,k} - u_{i,j,k}\right) - \left(u_{i,j,k} - u_{i,j-1,k}\right)}{\left(\Delta y\right)^2} \tag{8}$$

and for the time derivative:

$$\frac{1}{\alpha}\frac{\partial u}{\partial t} \approx \frac{1}{\alpha}\frac{\Delta u}{\Delta t} \approx \frac{1}{\alpha}\frac{\left(u_{i,j,k+1} - u_{i,j,k}\right)}{\left(\Delta t\right)} \tag{9}$$

where the i subscript represents the x coordinate, the j subscript represents the y coordinate, and the subscript k represents the time coordinate.

Let us choose:

$$\Delta x \equiv \Delta y \equiv h\,,$$

thus we can rewrite Eqns. 7, 8 as:

$$\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) \approx \frac{\left(u_{i+1,j,k} + u_{i-1,j,k} - 2u_{i\,j,k}\right)}{h^2} \tag{10}$$

$$\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y}\right) \approx \frac{\left(u_{i,j+1,k} + u_{i,j-1,k} - 2u_{i,j,k}\right)}{h^2} \tag{11}$$

In two-dimensional problems we can add Eqns. 10 and 11 and use Eqn. 9 to yield the finite difference form for the diffusion equation as:

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=\frac{(u_{i+1,j,k}+u_{i-1,j,k}-2u_{i\ j,k})}{h^2}+\frac{\left(u_{i,j+1,k}+u_{i,j-1,k}-2u_{i,j,k}\right)}{h^2}$$
$$=\frac{(u_{i+1,j,k}+u_{i-1,j,k}+u_{i,j+1,k}+u_{i,j-1,k}-4u_{i\ j,k})}{h^2} \quad (12)$$
$$=\frac{1}{\alpha}\frac{\left(u_{i,j,k+1}-u_{i,j,k}\right)}{(\Delta t)}$$

Solving for $u_{i,j,k+1}$, we get:

$$u_{i,j,k+1}=u_{i,j,k}+\frac{\alpha.\Delta t}{h^2}(u_{i+1,j,k}+u_{i-1,j,k}+u_{i,j+1,k}+u_{i,j-1,k})-\frac{4\alpha.\Delta t}{h^2}u_{i,j,k} \quad (13)$$

Combining terms yields:

$$u_{i,j,k+1}=(1-\frac{4\alpha.\Delta t}{h^2})u_{i,j,k}+\frac{4\alpha.\Delta t}{h^2}.\frac{(u_{i+1,j,k}+u_{i-1,j,k}+u_{i,j+1,k}+u_{i,j-1,k})}{4} \quad (14)$$

This implies that each internal point at the time step (k+1) is a weighted average over the surrounding node points at the previous time step k in addition to a contribution initially from the initial condition and thereafter from each previous increment in time.

This equation can be simplified by choosing the following relation between the time scale and the lattice constant:

$$\Delta t=\frac{h^2}{4\alpha} \quad (15)$$

or:

$$h^2=4\alpha.\Delta t \quad (16)$$

Making that choice, simplifies Eqn. 14 into:

$$u_{i,j,k+1}=\frac{(u_{i+1,j,k}+u_{i-1,j,k}+u_{i,j+1,k}+u_{i,j-1,k})}{4} \quad (17$$

RANDOM WALK MODEL

The averaging process in Eqn. 17 is a characteristic of potential problems in general. This implies a random with equal probabilities of moving from a mesh point to its four immediate neighbors in two dimensions, and to its six immediate neighbors in three dimensions with the initial condition added at each node point in the walk.

We consider that the random walk takes exactly one unit of time to move between two neighboring sites. We assume that the random walk starts from a cross road point P,

reaches one of the neighboring points with a probability of ¼ and continues this way until it reaches the boundary where it stops there. The process is allowed to continue for only k steps. If after k steps the random walk does not reach the boundary, but instead reaches an internal point, then a score for this point is scored as the initial condition at that internal point given by Eqn. 5:

$$\eta = g(P) \tag{18}$$

If, on the other hand the random walk does reach the boundary, then he accumulates a score given by the boundary condition:

$$\eta = f(C) \tag{19}$$

As shown in the procedure of Fig. 1, n random walks are generated of length k each giving the time dependence of the solution to the diffusion process.

```
!       Time_dependent_diffusion_equation for
        program Time_dependent_diffusion_equation
!       Two-dimensional Diffusion Equation Solver, with profile generation
!       Diffusion Equation in two Dimensions, Cartesian Coordinates
!               du/dt = d2u/dx2 + d2u/dy2
!       Solution by Monte Carlo random walk on a rectangular surface
!       with Dirichlet boundary conditions along the boundaries
!       and initial condition within the boundary
!       M. Ragheb
!       Random walk with equal step sizes
        dimension score(31,31), temp(31,31)
        dimension  initial_condition(30,30)
        real(8) elapsed_time
        character*1 tab
        elapsed_time=timef()
        tab=char(9)
!       Store output matrix for visualization using Excel
!       open (unit=10, file='temp_profile.xls', status='unknown')
!       Store output matrix for visualization using the Array visualizer
        open (unit=10, file='temp_profile.agl', status='unknown')
!       m1 = number of mesh points in x-direction
        m1=31
!       n1 = number of mesh points in y-direction
        n1=31
!       step probabilities in x+:p1, y+:p2, x-:p3 and y-:p4 directions
        p1=0.25
        p2=0.25
        p3=0.25
        p4=0.25
!       Construct cumulative distribution function for random walk
        p12=p1+p2
        p123=p1+p2+p3
!       number of random walks: nsamp
        nsamp=1000
!       Boundary conditions on the rectangle
!       left t1=t(0,y), bottom t2=t(x,0), right t3=t(m1,y), upper t4=t(x,n1)
        t1=100.
        t2=0.
        t3=0.
        t4=100.
```

```
!       Initial conditions within boundary
        do i=2,30,1
                do j=2,30,1
                        initial_condition(i,j)=0.0
                end do
        end do
        xsamp=nsamp
!       Allowed number of time steps
        ktotal=10
        do 30 m=2,30
        do 30 n=2,30
!       Score counter
        score(m,n)=0.0
        do 77 ncount=1,nsamp
!       Start random walk here
!       Initialize counters
        i=m
        j=n
!       History counter
!       Move particle ktotal steps
        do 66 kk=1,ktotal
!       Initiate random walk
!       i=m
!       j=n
        call random(r)
!       Sample cumulative distribution function for random walk
!       Move one step to the right
                if (r le.p1) then
                i=i+1
        goto 11
!       Move one step up
        else if (r le.p12) then
        j=j+1
        goto 11
!       Move one step left
        else if (r le.p123) then
        i=i-1
        goto 11
        else
!       Move one step down
        j=j-1
        goto 11
        end if
!       Check for random walk reaching boundary
!       Check whether lower boundary is reached
11      if (j.eq.1) then
        score(m,n) = score(m,n) + t2
        goto 88
!       Check whether right boundary is reached
        else if (i.eq m1) then
        score(m,n) = score(m,n) + t3
        goto 88
!       Check whether upper boundary is reached
        else if (j.eq.n1) then
        score(m,n) = score(m,n) + t4
        goto 88
!       Check whether left boundary is reached
        else if (i.eq.1)  then
        score(m,n) = score(m,n) + t1
        goto 88
        else
        go to 66
```

```
          end if
88        temp(m,n)=score(m,n)/xsamp
          go to 77
66        continue
!         write(*,*)m,n,i,j
          score(m,n)=score(m,n)+initial_condition(i,j)
          temp(m,n)=score(m,n)/xsamp

77        continue
!         Calculate solution at points of interest
!         Print results
30        continue
          write(*,*)'number of random walks=',nsamp
!         Boundary values
          do 40 i=1,31
                    do 40 j=1,31
!                   bottom boundary
                    temp(i,1)=0.0
!                   top boundary
                    temp(i,31)=100.0
!                   left boundary
                    temp(1,j)=100.0
!                   right boundary
                    temp(31,j)=0.0
40        continue
          do 20 n=1,31
          write(10,300) (temp(m,n),tab, m=1,31)
!         write(*,*) (temp(m,n),tab, m=1,31)
20        continue
300       format(31(e14.8,a1))
!         elapsed_time=timef()
          write(*,*) elapsed_time
          stop
          end
```

Figure1. Procedure for a random walk to solve the time dependent diffusion equation with initial and boundary conditions.

TIME DEPENDENT DIFFUSION EQUATION BENCHMARK

The procedure is tested using the geometry for the diffusion equation given in Fig. 2. As a problem whose solution is heuristically known, we consider that the boundary conditions are:

$$u_1 = u_2 = u_3 = u_4 = 100\,,$$

with an initial condition at all the internal non boundary points as:

$$u_{t=0} = 5\,.$$

The solution to this problem should start at the initial value then as time evolves should reach an equilibrium situation with the boundary values covering the whole domain. This is chosen as a suitable benchmark to test the Monte Carlo procedure. The solution for the benchmark is shown for different numbers of time steps in Figs. 3-6 for k

= 10, 100 1,000 and 10,000 time steps, or random chain lengths. The approach of the solution from the initial condition to reaching the values at the boundaries at equilibrium after a large number of time steps can be readily observed.

DIFFUSION EQUATION SOLUTION BY MONTE CARLO

Having tested the procedure of what can be considered as a benchmark the procedure is now applied to another case involving different boundary conditions with the results shown in Figs. 7-10 for the time steps or random chain lengths of k = 10, 100, 1,000 and 10,000 time steps. The boundary conditions are in this case:

$$u_1 = u_4 = 100$$
$$u_2 = u_3 = 0$$

with an initial condition at all the internal non boundary points as:

$$u_{t=0} = 5.$$

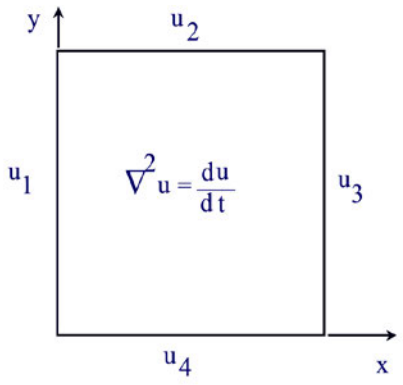


Figure 2. Geometry for the solution of the diffusion equation.

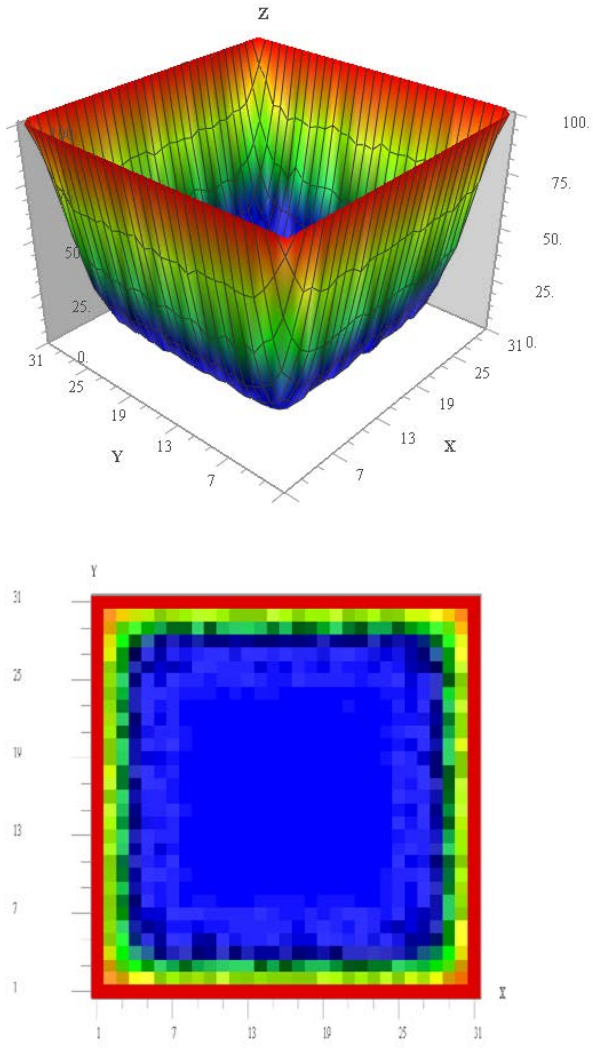
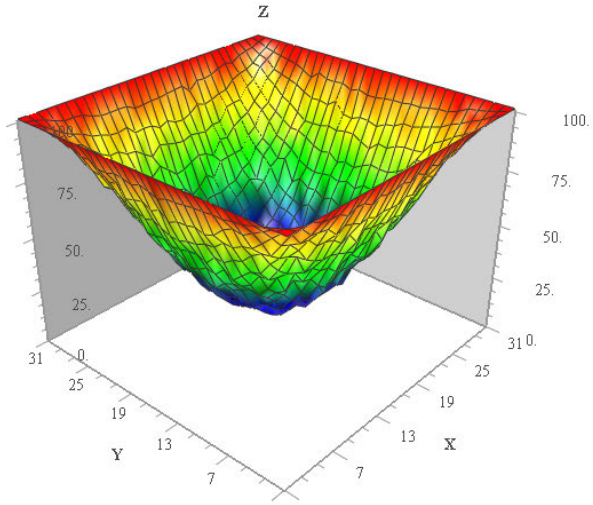


Figure 3. Diffusion Equation benchmark, number of time steps k=10.



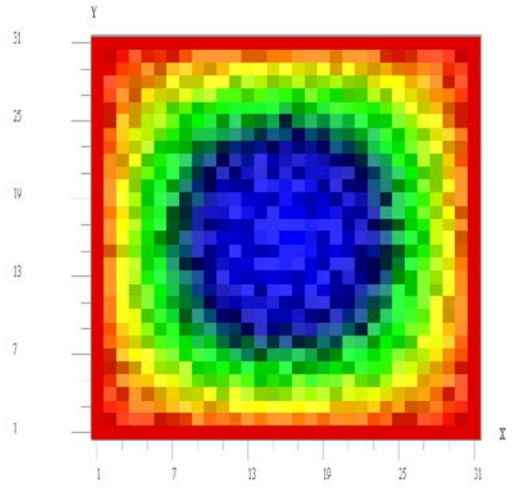


Figure 4. Diffusion Equation benchmark, number of time steps k=100.

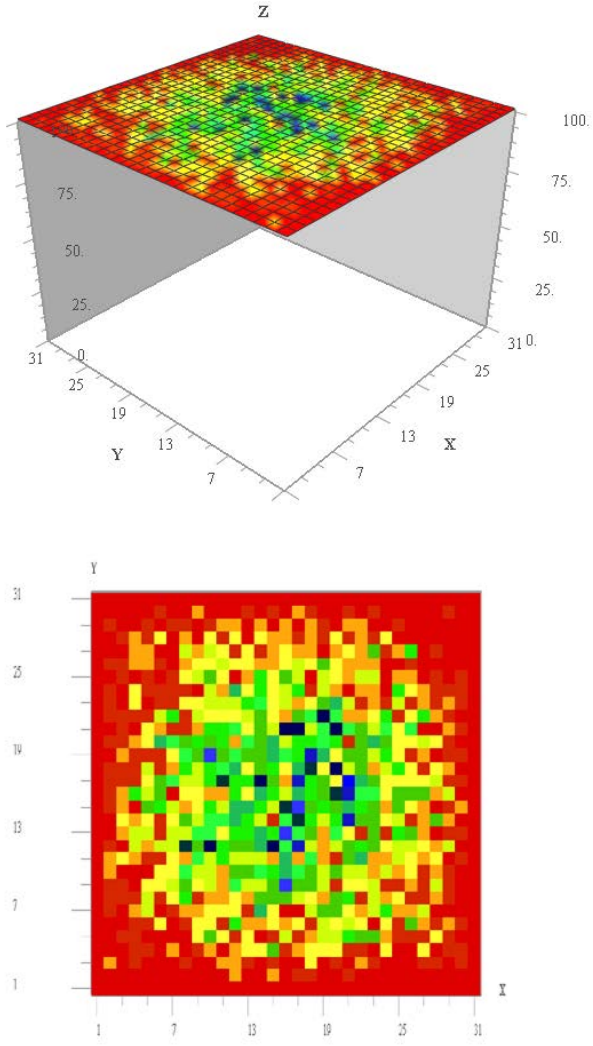


Figure 5. Diffusion Equation benchmark, number of time steps k=1,000.

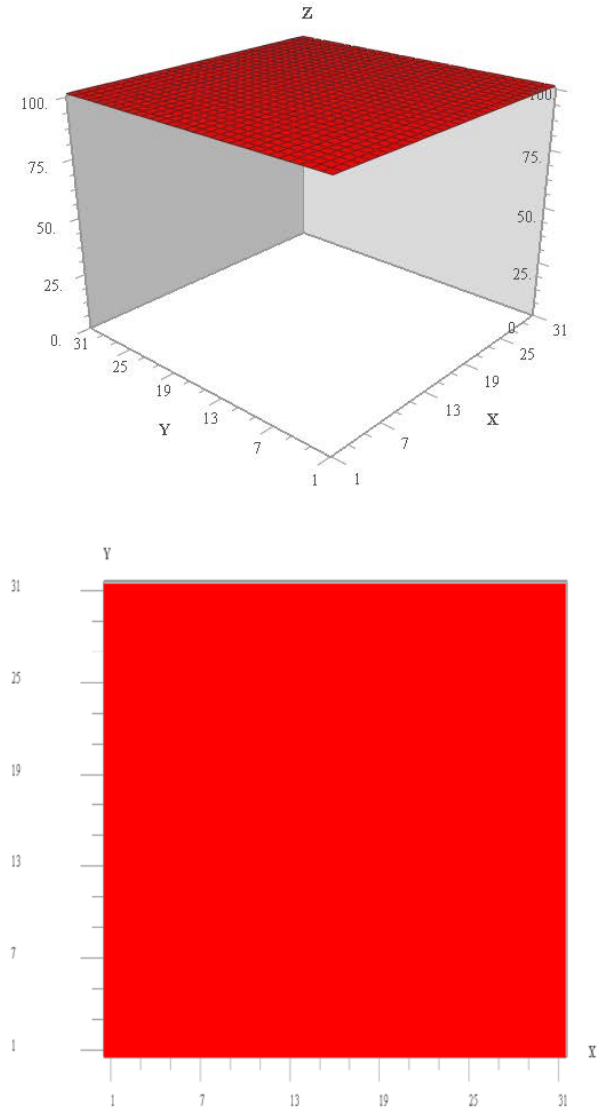


Figure 6. Diffusion Equation benchmark, number of time steps k=10,000. Steady state equilibrium and exact solution are attained.

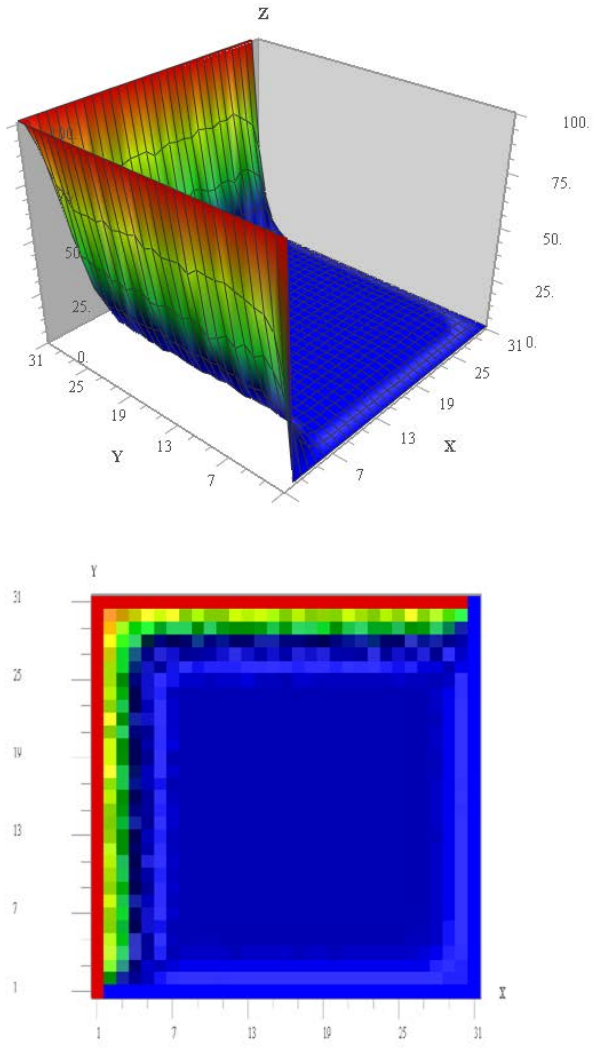
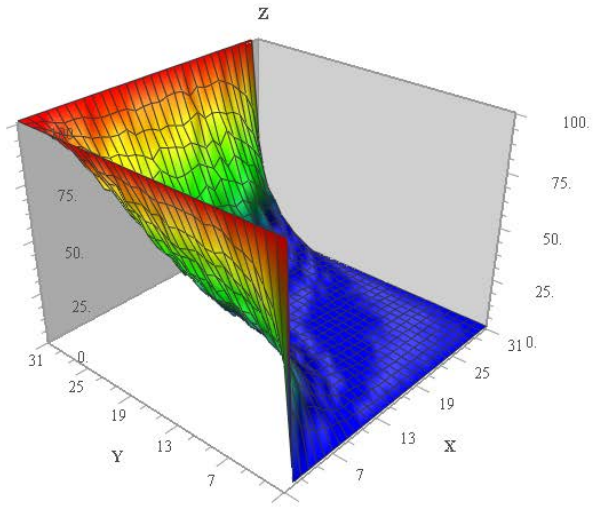


Figure 7. Diffusion Equation benchmark, number of time steps k=10.



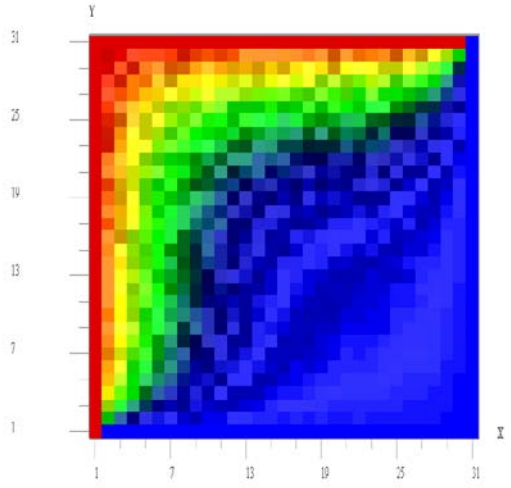
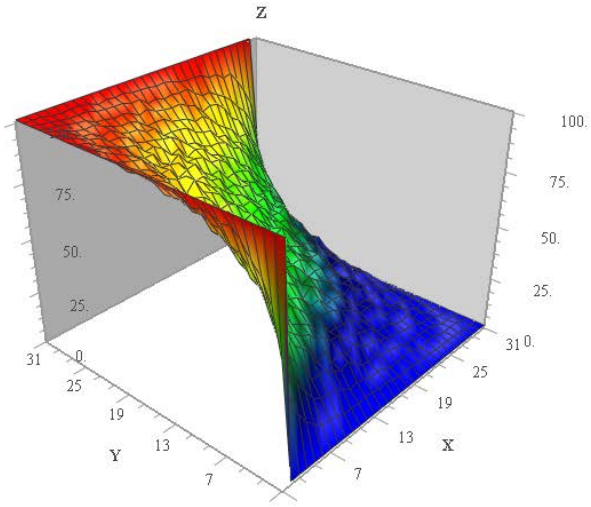


Figure 8. Diffusion Equation benchmark, number of time steps k=100.



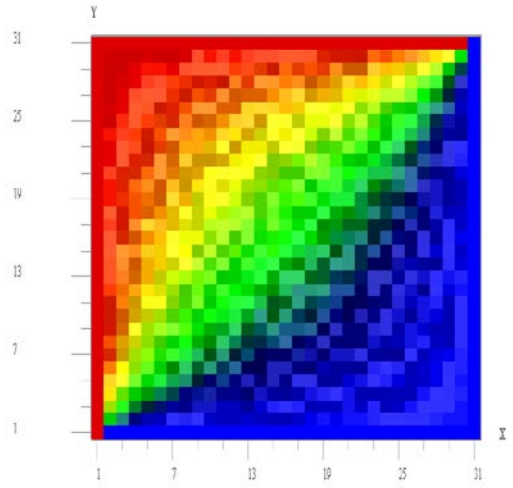


Figure 9. Diffusion Equation benchmark, number of time steps k=1,000.

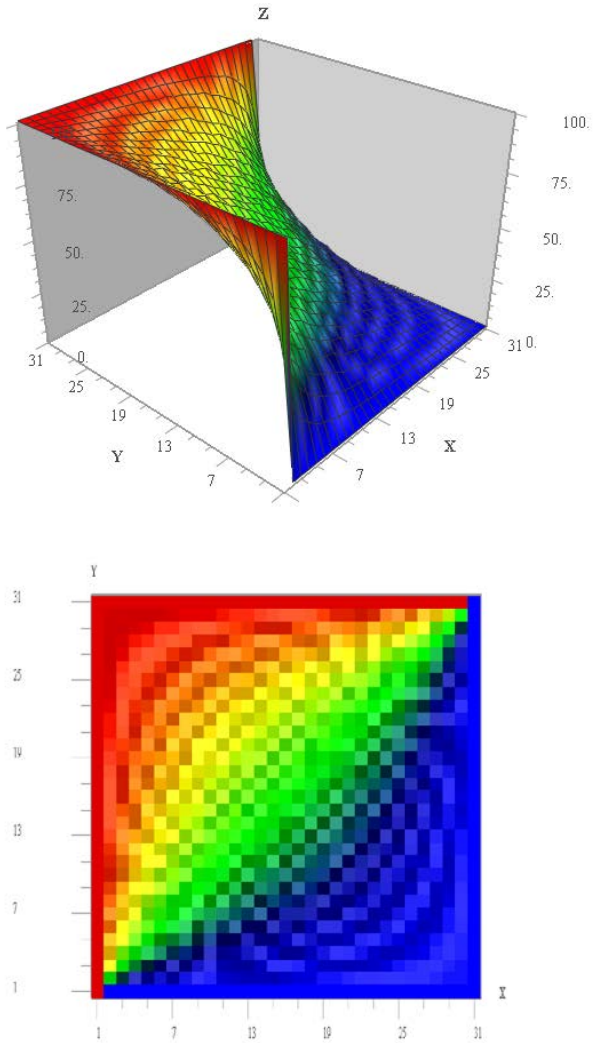


Figure 10. Diffusion Equation benchmark, number of time steps k=10,000. Equilibrium steady state solution.

DISCUSSION

Random walk procedures can be used to solve a variety of problem involving mass and heat transport, where each step of the generated random walk corresponds to a single time step in the simulation. The procedure described above can be generalized to more complex problems involving nonhomogeneous media, media with in internal sources and sinks and to mixed types of boundary conditions. The areas of applications range from the study of purification processes and crystal formation involving adsorption and desorption to time dependent heat transport problems including varying medium properties and mixed types of boundary conditions.

EXERCISES

1. Modify the procedure for the solution of the Diffusion Equation to deal with the situation where an internal or external source is present corresponding to the time dependent Poisson's equation with a mixed type of boundary condition:

$$k\nabla^2 u + q_0 = \frac{1}{\alpha}\frac{\partial u}{\partial t} \quad on \;\; domain \;\; D,$$

$$u = (af + b\frac{df}{dr}) \quad on \; boundary \; C,$$

$$u\Big|_{t=0} = g(D)$$

2. Solve the case of a diffusion process in a nonuniform medium where;

$$\nabla.[k(x,y)\nabla u(x,y,t)] + q_0 = \frac{1}{\alpha}\frac{\partial u(x,y,t)}{\partial t} \quad on \;\; domain \;\; D.$$