**An Introductory Comparison of Matlab and R**
**Applications to Ordinary and Partial Differential Equations**

**W. E. Schiesser**
**Lehigh University**
**Bethlehem, USA**[1]

[1]Iacocca B312, Lehigh University, 111 Research Drive, Bethlehem, PA 18015 USA; wes1@lehigh.edu; 610-758-4264

## Features

**Matlab**

- **Convenient graphics**
  - **2D, 3D (in-line)**
  - **Color**
- **Quality utilities**
  - **ODE integrators (e.g., ode15s)**
  - **Extensive linear algebra routines**

## Features

**Matlab**

- **Convenient graphics**
  - **2D, 3D (in-line)**
  - **Color**
- **Quality utilities**
  - **ODE integrators (e.g., ode15s)**
  - **Extensive linear algebra routines**

**R**

- **Convenient graphics**
  - **2D, 3D (in-line)**
  - **Color**
  - **Many options**
- **Quality utilities**
  - **ODE integrators (e.g., ode, lsoda, lsodes)**
  - **Extensive linear algebra routines**

## Features (cont'd)

**Matlab**

- **Complicated, expensive annual licenses**
- **Limits use**

**R**

- **Open source (no cost)**
- **Available to anyone with an Internet connection**
- **Easily downloaded**

# ODEs

## Single ODE for tumor growth

$$\frac{dV}{dt} = \lambda e^{-\alpha t} V;\ V(t_0) = V_0$$

| | |
|---|---|
| $V$ | tumor volume |
| $t$ | time |
| $\lambda, \alpha, t_0, V_0$ | constants |

## ODEs

### Single ODE for tumor growth

$$\frac{dV}{dt} = \lambda e^{-\alpha t} V; \; V(t_0) = V_0$$

| | |
|---|---|
| $V$ | tumor volume |
| $t$ | time |
| $\lambda, \alpha, t_0, V_0$ | constants |

### Analytical solution

$$V(t) = V_0 e^{(\lambda/\alpha)(1-e^{-\alpha t})}$$

### Limiting solution

$$\lambda = \alpha > 0; \; V_0 = 1$$

$$V(t \to \infty) = e$$

Braun, M. (1993), *Differential Equations and Their Applications*, 4th ed., Springer-Verlag, New York, 52-53

Two interpretations:

$$\frac{dV}{dt} = (\lambda e^{-\alpha t})V; \quad \frac{dV}{dt} = \lambda(e^{-\alpha t})V;$$

## ODE Routines

Matlab

```
  function yt=tumor_1(t,y)
%
% Function tumor_1 computes the derivative for the
% tumor growth ODE
%
  global alpha lambda icase ncall
%
% Calls to tumor_1
  ncall=ncall+1;
%
% ODE
  yt=lambda(icase)*exp(-alpha*t)*y;
```

R

```
  tumor_1=function(t,y,parms){
#
# Function tumor_1 computes the derivative for the
# tumor growth ODE
#
# Calls to tumor_1
  ncall<<-ncall+1;
#
# ODE
  yt=lambda[icase]*exp(-alpha*t)*y;
#
# Return derivative
  return(list(c(yt)));
  }
```

## ODE Main Program - Matlab

```
%
% Clear previous files
  clear all
  clc
%
% Parameters, variable shared by other routines
  global alpha lambda icase ncall
%
% Initial condition
  for(icase=1:3)
    if(icase==1)lambda(1)=1   ;end
    if(icase==2)lambda(2)=0.75;end
    if(icase==3)lambda(3)=1.25;end
    V0=1;alpha=1;ncall=0;
%
%   t scale
    tout=[0:0.5:10]';nout=21;
%
%   ODE integration
    [t,V]=ode15s(@tumor_1,tout,V0);
%
%   Display heading
    fprintf('\n icase = %2d',icase);
%
%   Display calls to tumor_1
    fprintf('\n\n Calls to tumor_1 = %3d\n',ncall);
    fprintf('\n       t     V(t)(num)   V(t)(anal)    diff\n');
%
%   Display numerical output
    for(it=1:nout)
       Va(it)=V0*exp(lambda(icase)/alpha*(1-exp(-alpha*tout(it))));
      Vplot(it,icase)=V(it,1);
     Vaplot(it,icase)= Va(it);
```

```
     diff(it,icase)=(Vplot(it,icase)-Vaplot(it,icase))...
                    /Vplot(it,icase);
    fprintf('%10.2f%12.4f%12.4f%12.6f\n',...
    tout(it),Vplot(it,icase),Vaplot(it,icase),diff(it,icase));
  end
 end
%
% Plot numerical, analytical solutions
 figure(1)
 plot(tout,Vplot,'-',tout,Vaplot,'o');
 xlabel('t');ylabel('V(t)');
 title('V(t), solid - num, points - anal');
%
% Plot difference of solutions
 figure(2)
 plot(tout,diff,'-');
 xlabel('t');ylabel('diff');
 title('diff = num - anal (rel), V(t)');
```

## ODE Main Program - R

```
#
# Delete previous workspaces
  rm(list=ls(all=TRUE))
#
# Access ODE integrator
  library("deSolve");
#
# Access ODE file
  setwd("c:/R/odepde_intro/ode");
  source("tumor_1.R");
#
# Matrices for plotted output
  nout=21;
   Vplot=matrix(0,nrow=nout,ncol=3);
  Vaplot=matrix(0,nrow=nout,ncol=3);
    diff=matrix(0,nrow=nout,ncol=3);
#
# Initial condition
  lambda=rep(0,3);
  for(icase in 1:3){
    if(icase==1){lambda[1]=1   ;}
    if(icase==2){lambda[2]=0.75;}
    if(icase==3){lambda[3]=1.25;}
    V0=1;alpha=1;ncall=0;
#
#   t scale
    tout=seq(from=0,to=10,by=0.5);
#
#   ODE integration
    out=ode(func=tumor_1,times=tout,y=V0,parms);
#
#   Display heading
    V=rep(0,nout);Va=rep(0,nout);
```

```
    V=out[,-1];
    cat(sprintf("\n\n icase = %2d\n",icase));
#
#   Display calls to ODE routine
    cat(sprintf("\n ncall = %3d\n",ncall));
#
#   Display numerical output
    cat(sprintf("\n         t     V(t)(num)   V(t)(anal)    diff\n"));
    for(it in 1:nout){
       Va[it]=V0*exp(lambda[icase]/alpha*(1-exp(-alpha*tout[it])));
      Vplot[it,icase]= V[it];
     Vaplot[it,icase]=Va[it];
       diff[it,icase]=(Vplot[it,icase]-Vaplot[it,icase])/
                       Vplot[it,icase];
    cat(sprintf("%10.2f%12.4f%12.4f%12.6f\n",
        tout[it],Vplot[it,icase],Vaplot[it,icase],diff[it,icase]));
    }
  }
#
# Plot numerical, analytical solutions
  par(mfrow=c(1,1))
  plot(tout,Vplot[,1],xlab="t",ylab="V(t)",
       xlim=c(0,10),ylim=c(1,5),
       main="V(t), solid - num, points - anal",type="l",lwd=2);
  points(tout,Vaplot[,1],pch="1");
   lines(tout, Vplot[,2],type="l",lwd=2);
  points(tout,Vaplot[,2],pch="2");
   lines(tout, Vplot[,3],type="l",lwd=2);
  points(tout,Vaplot[,3],pch="3");
#
# Plot difference of solutions
  par(mfrow=c(1,1))
  matplot(tout,diff,type="l",xlab="t",ylab="diff",
    col="black",lwd=2,lty=1,main="diff = num - anal (rel)");
```

## Numerical Solution - Matlab

```
icase =  1

Calls to tumor_1 =  53

      t      V(t)(num)   V(t)(anal)     diff
     0.00      1.0000      1.0000    0.000000
     0.50      1.4770      1.4821   -0.003475
     1.00      1.8742      1.8816   -0.003954
     1.50      2.1674      2.1747   -0.003365
     2.00      2.3680      2.3742   -0.002617
     2.50      2.4980      2.5041   -0.002440
     3.00      2.5799      2.5863   -0.002473
     3.50      2.6297      2.6374   -0.002923
     4.00      2.6598      2.6689   -0.003449
     4.50      2.6779      2.6883   -0.003849
     5.00      2.6890      2.7000   -0.004104
     5.50      2.6955      2.7072   -0.004327
     6.00      2.6996      2.7116   -0.004422
     6.50      2.7020      2.7142   -0.004522
     7.00      2.7036      2.7158   -0.004513
     7.50      2.7046      2.7168   -0.004486
     8.00      2.7053      2.7174   -0.004475
     8.50      2.7057      2.7177   -0.004462
     9.00      2.7059      2.7179   -0.004450
     9.50      2.7061      2.7181   -0.004437
    10.00      2.7061      2.7182   -0.004441


icase =  2

Calls to tumor_1 =  51

      t      V(t)(num)   V(t)(anal)     diff
```

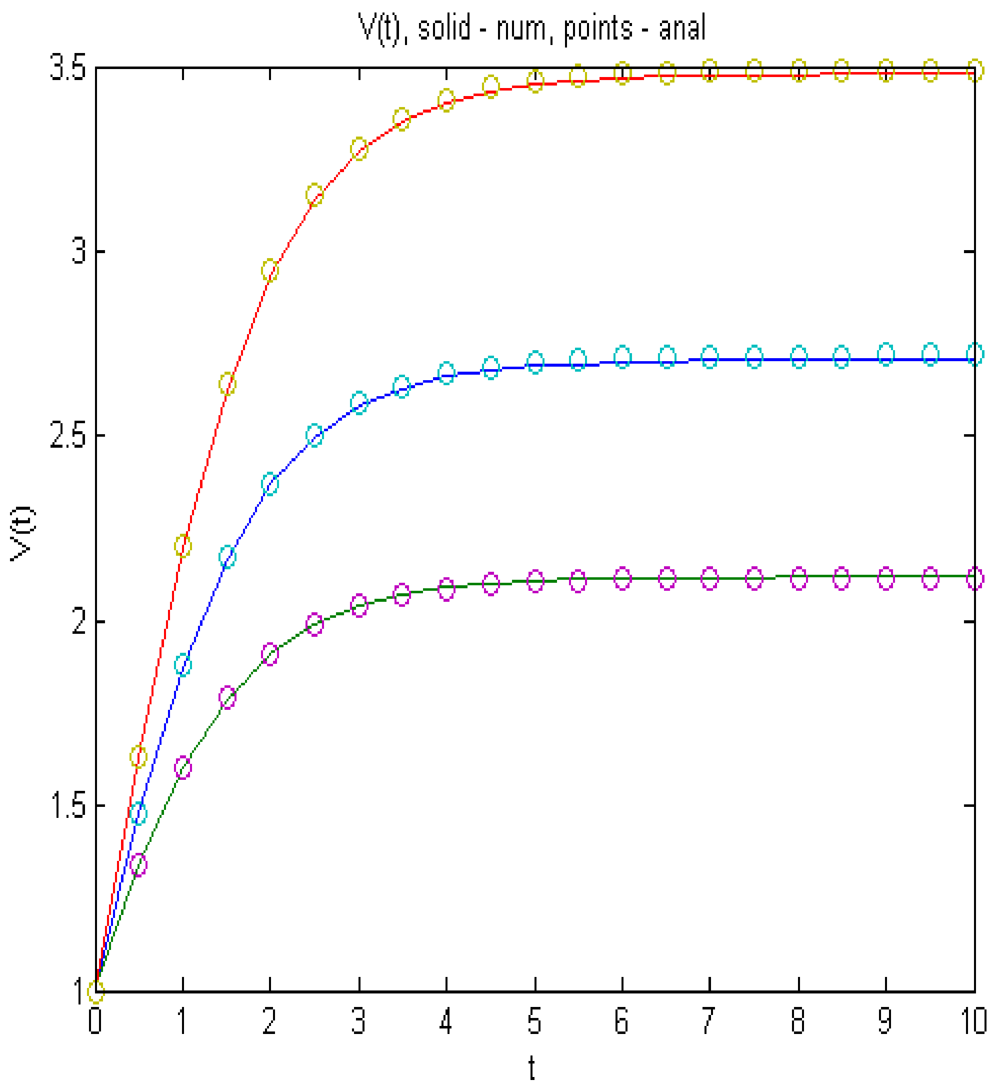
```
    0.00      1.0000      1.0000    0.000000
    0.50      1.3401      1.3433   -0.002388
    1.00      1.6027      1.6066   -0.002410
                 .                    .
                 .                    .
       Solution for t = 1.50 to 8.50 removed
                 .                    .
                 .                    .
    9.00      2.1192      2.1168    0.001129
    9.50      2.1193      2.1169    0.001157
   10.00      2.1194      2.1169    0.001172


icase =  3

Calls to tumor_1 =  59

     t     V(t)(num)   V(t)(anal)    diff
    0.00      1.0000      1.0000    0.000000
    0.50      1.6328      1.6353   -0.001522
    1.00      2.1922      2.2037   -0.005250
                 .                    .
                 .                    .
       Solution for t = 1.50 to 8.50 removed
                 .                    .
                 .                    .
    9.00      3.4788      3.4898   -0.003166
    9.50      3.4790      3.4900   -0.003173
   10.00      3.4791      3.4901   -0.003168
```

## Numerical Solution - R

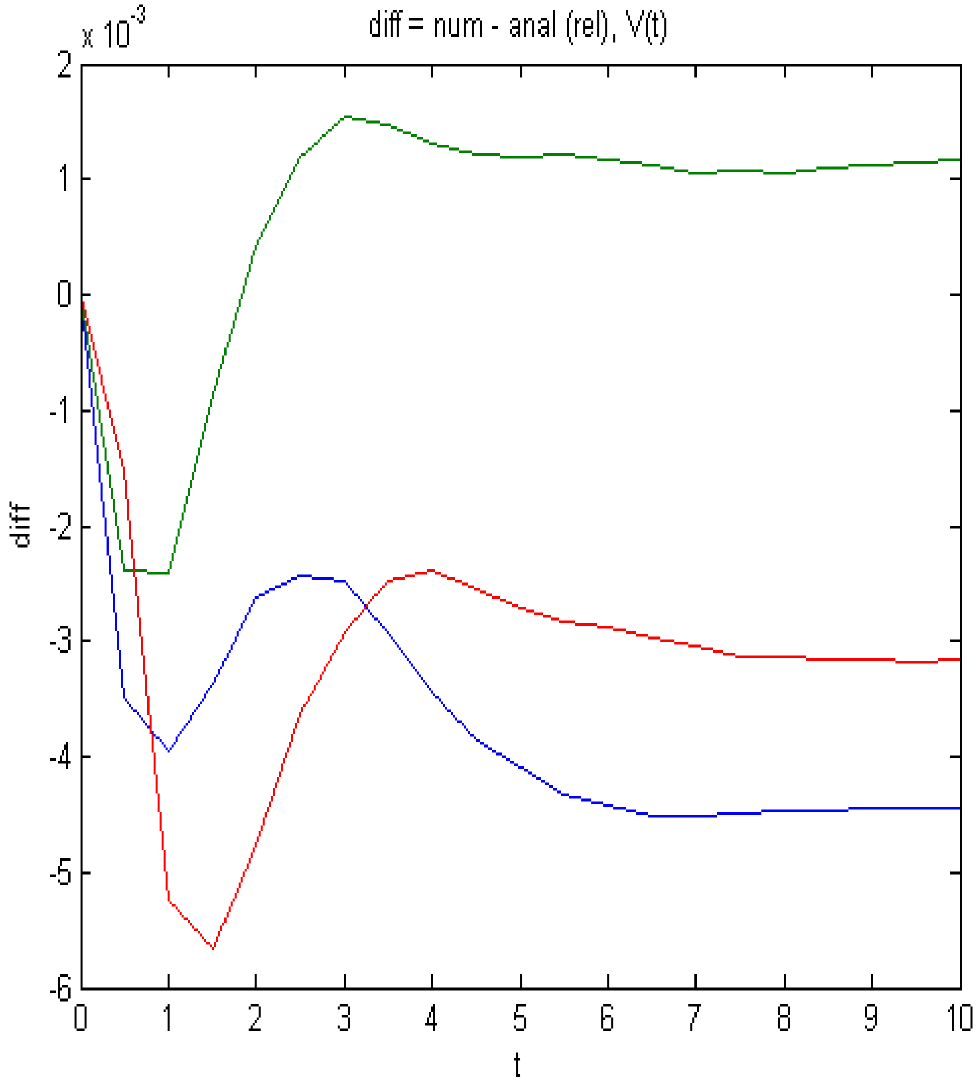
```
icase =  1

ncall = 100

       t     V(t)(num)   V(t)(anal)    diff
     0.00      1.0000      1.0000    0.000000
     0.50      1.4821      1.4821    0.000001
     1.00      1.8816      1.8816   -0.000001
     1.50      2.1747      2.1747   -0.000000
     2.00      2.3742      2.3742    0.000002
     2.50      2.5041      2.5041    0.000001
     3.00      2.5863      2.5863    0.000001
     3.50      2.6374      2.6374    0.000001
     4.00      2.6690      2.6689    0.000001
     4.50      2.6883      2.6883    0.000000
     5.00      2.7000      2.7000   -0.000001
     5.50      2.7072      2.7072   -0.000001
     6.00      2.7115      2.7116   -0.000002
     6.50      2.7142      2.7142   -0.000002
     7.00      2.7158      2.7158   -0.000002
     7.50      2.7168      2.7168   -0.000002
     8.00      2.7174      2.7174   -0.000003
     8.50      2.7177      2.7177   -0.000003
     9.00      2.7179      2.7179   -0.000003
     9.50      2.7181      2.7181   -0.000003
    10.00      2.7182      2.7182   -0.000003


icase =  2

ncall =  88

       t     V(t)(num)   V(t)(anal)    diff
```

```
     0.00      1.0000      1.0000    0.000000
     0.50      1.3433      1.3433    0.000000
     1.00      1.6065      1.6066   -0.000002
                 .                    .
                 .                    .
        Solution for t = 1.50 to 8.50 removed
                 .                    .
                 .                    .
     9.00      2.1168      2.1168   -0.000004
     9.50      2.1169      2.1169   -0.000004
    10.00      2.1169      2.1169   -0.000004


icase =  3

ncall = 102

       t     V(t)(num)   V(t)(anal)    diff
     0.00      1.0000      1.0000    0.000000
     0.50      1.6353      1.6353    0.000002
     1.00      2.2037      2.2037    0.000000
                 .                    .
                 .                    .
        Solution for t = 1.50 to 8.50 removed
                 .                    .
                 .                    .
     9.00      3.4898      3.4898   -0.000006
     9.50      3.4900      3.4900   -0.000006
    10.00      3.4901      3.4901   -0.000006
```

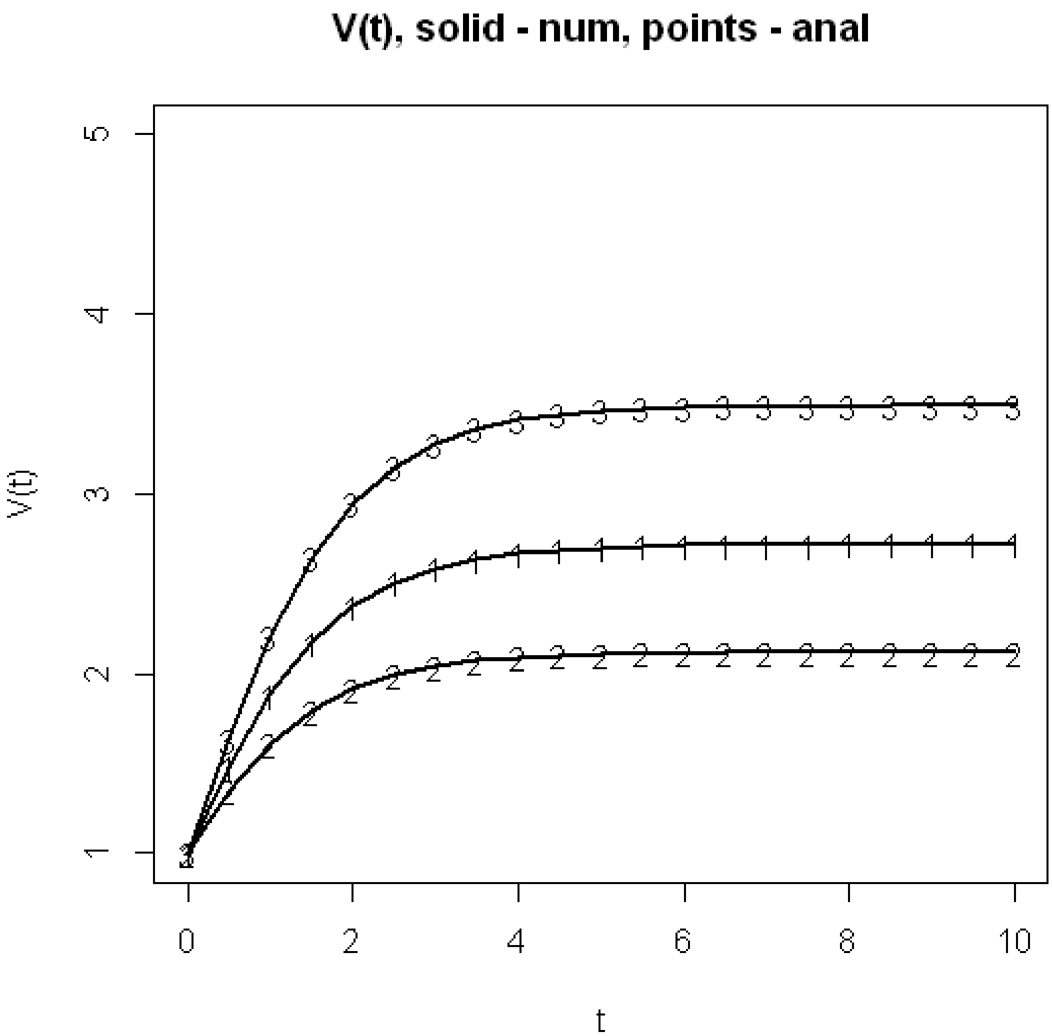
## Matlab output



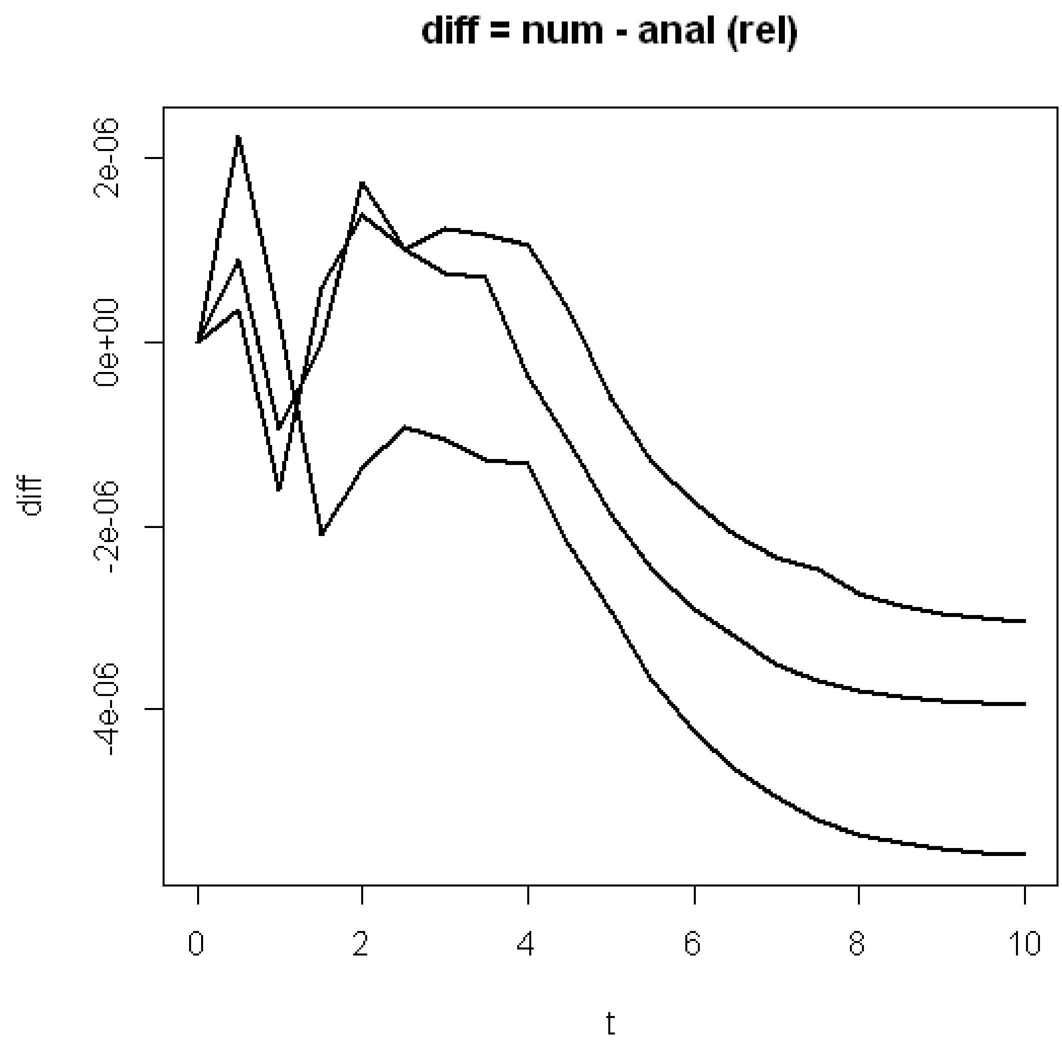
## Matlab errors



## R output



## R errors



# PDEs

## Fick's second law

$$\frac{\partial u}{\partial t} = \frac{\partial \left[ D \frac{\partial u}{\partial x} \right]}{\partial x}$$

| Variables parameters | Interpretation |
|---|---|
| $u$ | dependent variable |
| $x$ | boundary value independent variable |
| $t$ | initial value independent variable |
| $D$ | diffusivity (a parameter to be specified) |

| | |
|---|---|
| (1) $D = \text{constant}$ | linear |
| (2) $D = u$ | nonlinear |
| (3) $D = e^u$ | nonlinear |

# PDEs

## Fick's second law

$$\frac{\partial u}{\partial t} = \frac{\partial \left[ D \frac{\partial u}{\partial x} \right]}{\partial x}$$

$$u(x, t = 0) = 0$$

$$u(x = 0, t) = 1; \ \frac{\partial u(x = L, t)}{\partial x} = 0$$

BC 1: Dirichlet; BC 2: Neumann

## Analytical solution, D=1

$$u(x, t) = 1 - 2 \sum_{i=1}^{\infty} \frac{1}{\lambda_i} e^{-(\lambda_i^2/L^2)t} \sin(\lambda_i x/L)$$

with the eigenvalues

$$\lambda_i = (i - 1/2)\pi, \ i = 1, 2, ...$$

## ODE Routine - Matlab

```
  function ut=pde_1(t,u)
%
% Parameters
  global xl xu nx ncall ndss ncase
%
% BC at x = 0 (Dirichlet)
  u(1)=1;
%
% ux
  if    (ndss== 2) ux=dss002(xl,xu,nx,u); % second order
  elseif(ndss== 4) ux=dss004(xl,xu,nx,u); % fourth order
  elseif(ndss== 6) ux=dss006(xl,xu,nx,u); % sixth order
  elseif(ndss== 8) ux=dss008(xl,xu,nx,u); % eighth order
  elseif(ndss==10) ux=dss010(xl,xu,nx,u); % tenth order
  end
%
% BC at x = 1 (Neumann)
  ux(nx)=0;
%
% Diffusivity
  for i=1:nx
    if(ncase==1)D(i)=1          ;end
    if(ncase==2)D(i)=u(i)       ;end
    if(ncase==3)D(i)=exp(u(i));end
    ux(i)=D(i)*ux(i);
  end
%
% uxx
  if    (ndss== 2) uxx=dss002(xl,xu,nx,ux); % second order
  elseif(ndss== 4) uxx=dss004(xl,xu,nx,ux); % fourth order
  elseif(ndss== 6) uxx=dss006(xl,xu,nx,ux); % sixth order
  elseif(ndss== 8) uxx=dss008(xl,xu,nx,ux); % eighth order
```

```
  elseif(ndss==10) uxx=dss010(xl,xu,nx,ux); % tenth order
  end
%
% PDE
  ut=uxx';
  ut(1)=0;
%
% Increment calls to pde_1
  ncall=ncall+1;
```

## ODE Routine - R

```
  pde_1=function(t,u,parms){
#
# BC at x = 0 (Dirichlet)
  u[1]=1;
#
# ux
  if      (ndss== 2){ux=dss002(xl,xu,nx,u); # second order
  }else if(ndss== 4){ux=dss004(xl,xu,nx,u); # fourth order
  }else if(ndss== 6){ux=dss006(xl,xu,nx,u); # sixth order
  }else if(ndss== 8){ux=dss008(xl,xu,nx,u); # eighth order
  }else if(ndss==10){ux=dss010(xl,xu,nx,u); # tenth order
  }
#
# BC at x = 1 (Neumann)
  ux[nx]=0;
#
# Diffusivity
  D=rep(0,nx);
  for(i in 1:nx){
    if(ncase==1){D[i]=1        ;}
    if(ncase==2){D[i]=u[i]     ;}
    if(ncase==3){D[i]=exp(u[i]);}
    ux[i]=D[i]*ux[i];
  }
#
# uxx
  if      (ndss== 2){uxx=dss002(xl,xu,nx,ux); # second order
  }else if(ndss== 4){uxx=dss004(xl,xu,nx,ux); # fourth order
  }else if(ndss== 6){uxx=dss006(xl,xu,nx,ux); # sixth order
  }else if(ndss== 8){uxx=dss008(xl,xu,nx,ux); # eighth order
  }else if(ndss==10){uxx=dss010(xl,xu,nx,ux); # tenth order
  }
```

```
#
# PDE
  ut=rep(0,nx);
  ut=uxx;
  ut[1]=0;
#
# Increment calls to pde_1
  ncall <<- ncall+1;
#
# Return derivative vector
  return(list(c(ut)));
}
```

## Main Program - Matlab

```
%
% Clear previous files
  clear all
  clc
%
% Parameters shared with the ODE routine
  global xl xu nx ncall ndss ncase
%
% Grid in x, initial condition
  xl=0;xu=1;nx=26;
  for ix=1:nx
    xg(ix)=(xu-xl)*(ix-1)/(nx-1);
    u0(ix)=0;
  end
%
% Independent variable for ODEs
  t0=0.0;tf=1.5;nout=11;
  tout=linspace(t0,tf,nout);
  ncall=0;
%
% ODE integration
  ncase=1;ndss=4;
  [t,u]=ode15s(@pde_1,tout,u0);
%
% Display selected output
  for it=1:nout
    if(it==1);u(it,1)=0;end
    if(it> 1);u(it,1)=1;end
    fprintf('\n      t       x      u(x,t)\n');
    for ix=1:5:nx
      fprintf('%7.2f%8.2f%12.5f\n',t(it),xg(ix),u(it,ix));
    end
  end
```

```
  fprintf('\n ncall = %4d\n',ncall);
%
% Plot numerical solution
  figure(1);
  plot(xg,u);
  xlabel('x');ylabel('u(x,t)');
  title('u(x,t) vs x, t=0,0.15,...,1.5');
```

## Main Program - R

```
#
# Remove previous workspaces
  rm(list=ls(all= TRUE));
#
# Access ODE integrator
  library("deSolve");
#
# Access functions
  setwd("c:/R/pde_intro");
  source("pde_1.R");
  source("dss004.R");
#
# Grid in x, initial condition
  xl=0;xu=1;nx=26;
  xg=rep(0,nx);u0=rep(0,nx);
  for(ix in 1:nx){
    xg[ix]=(xu-xl)*(ix-1)/(nx-1);
    u0[ix]=0;
  }
#
# Independent variable for ODEs
  t0=0;tf=1.5;nout=11;
  tout=seq(from=0,to=tf,by=(tf-t0)/(nout-1));
  ncall=0;
#
# ODE integration
  ncase=1;ndss=4;
  out=lsodes(y=u0,times=tout,func=pde_1,parms=NULL)
#
# Display selected output
  for(it in 1:nout){
    if(it==1){out[it,2]=0;}
    if(it> 1){out[it,2]=1;}
```

```
    cat(sprintf("\n      t       x      u(x,t)\n"));
    for(ix in 1:nx){
      if((ix-1)*(ix- 6)*(ix-11)*(ix-16)
               *(ix-21)*(ix-26)==0){
        cat(sprintf("%7.2f%8.2f%12.5f\n",
        out[it,1],xg[ix],out[it,ix+1]));
      }
    }
  }
  cat(sprintf("\n ncall = %4d\n",ncall));
#
# Plot numerical solution
  par(mfrow=c(1,1));
  matplot(x=xg,y=t(out[,-1]),type="l",xlab="x",
    ylab="u(x,t) vs x, t=0,0.15,...,1.5",xlim=c(xl,xu),
    main="u(x,t) vs x, t=0,0.15,...,1.5;",lty=1,lwd=2);
```

**Matlab output, D = 1 (ncase = 1)**

```
     t        x       u(x,t)
   0.00     0.00     0.00000
   0.00     0.20     0.00000
   0.00     0.40     0.00000
   0.00     0.60     0.00000
   0.00     0.80     0.00000
   0.00     1.00     0.00000
           .
  (output for t = 0.15
    to 0.30 removed)
           .
     t        x       u(x,t)
   0.45     0.00     1.00000
   0.45     0.20     0.87040
   0.45     0.40     0.75347
   0.45     0.60     0.66065
   0.45     0.80     0.60105
   0.45     1.00     0.58052
          .
  (output for t = 0.60
    to 1.35 removed)
           .
     t        x       u(x,t)
   1.50     0.00     1.00000
   1.50     0.20     0.99033
   1.50     0.40     0.98160
   1.50     0.60     0.97467
   1.50     0.80     0.97022
   1.50     1.00     0.96869

ncall =  154
```

**R output, D = 1 (ncase = 1)**

```
    t        x       u(x,t)
 0.00     0.00      0.00000
 0.00     0.20      0.00000
 0.00     0.40      0.00000
 0.00     0.60      0.00000
 0.00     0.80      0.00000
 0.00     1.00      0.00000
          .
(output for t = 0.15
  to 0.30 removed)
          .
    t        x       u(x,t)
 0.45     0.00      1.00000
 0.45     0.20      0.87036
 0.45     0.40      0.75342
 0.45     0.60      0.66064
 0.45     0.80      0.60107
 0.45     1.00      0.58055
          .
(output for t = 0.60
  to 1.35 removed)
          .
    t        x       u(x,t)
 1.50     0.00      1.00000
 1.50     0.20      0.99028
 1.50     0.40      0.98152
 1.50     0.60      0.97456
 1.50     0.80      0.97009
 1.50     1.00      0.96855

 ncall =  230
```

## Comparison of Matlab, R, analytical solutions
## D = 1 (ncase = 1)

Matlab

| t | x | u(x,t) |
|---|---|---|
| 0.45 | 0.00 | 1.00000 |
| 0.45 | 0.20 | 0.87040 |
| 0.45 | 0.40 | 0.75347 |
| 0.45 | 0.60 | 0.66065 |
| 0.45 | 0.80 | 0.60105 |
| 0.45 | 1.00 | 0.58052 |

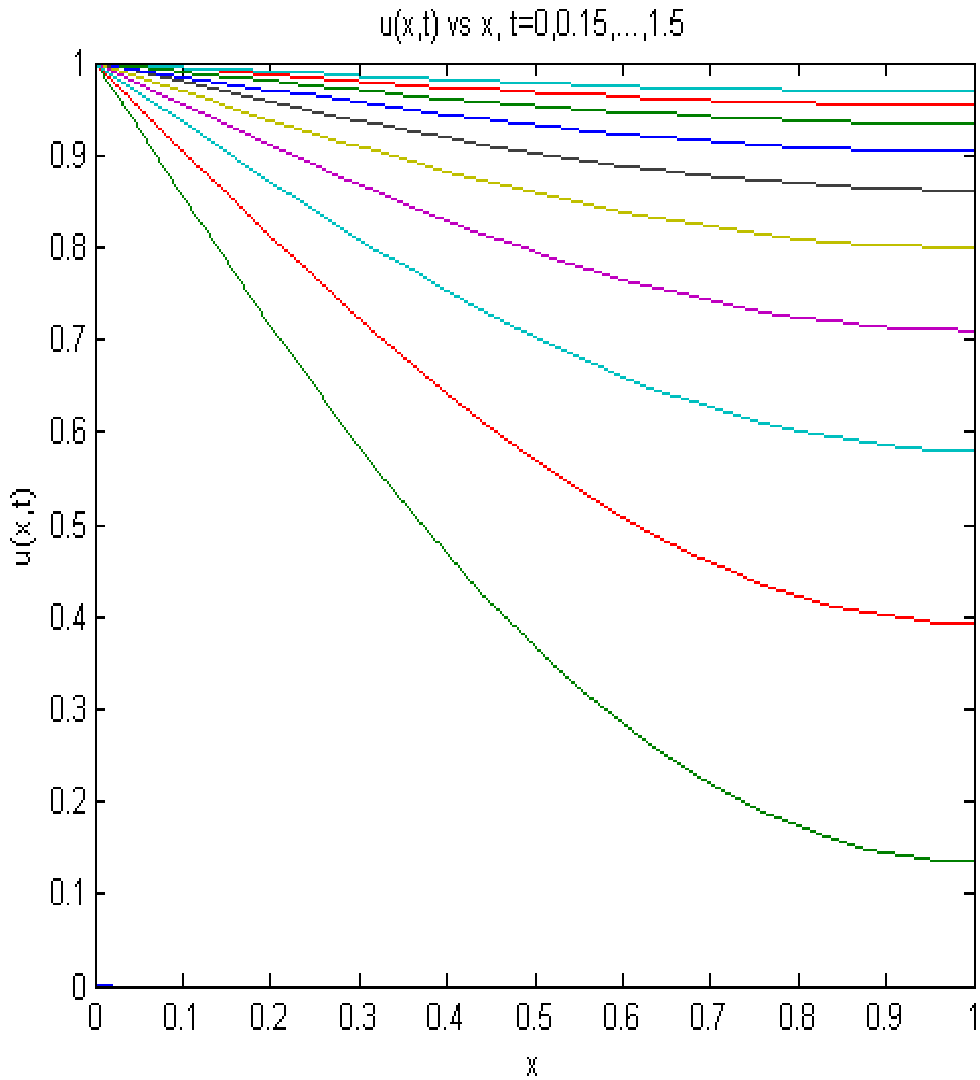
R

| t | x | u(x,t) |
|---|---|---|
| 0.45 | 0.00 | 1.00000 |
| 0.45 | 0.20 | 0.87036 |
| 0.45 | 0.40 | 0.75342 |
| 0.45 | 0.60 | 0.66064 |
| 0.45 | 0.80 | 0.60107 |
| 0.45 | 1.00 | 0.58055 |

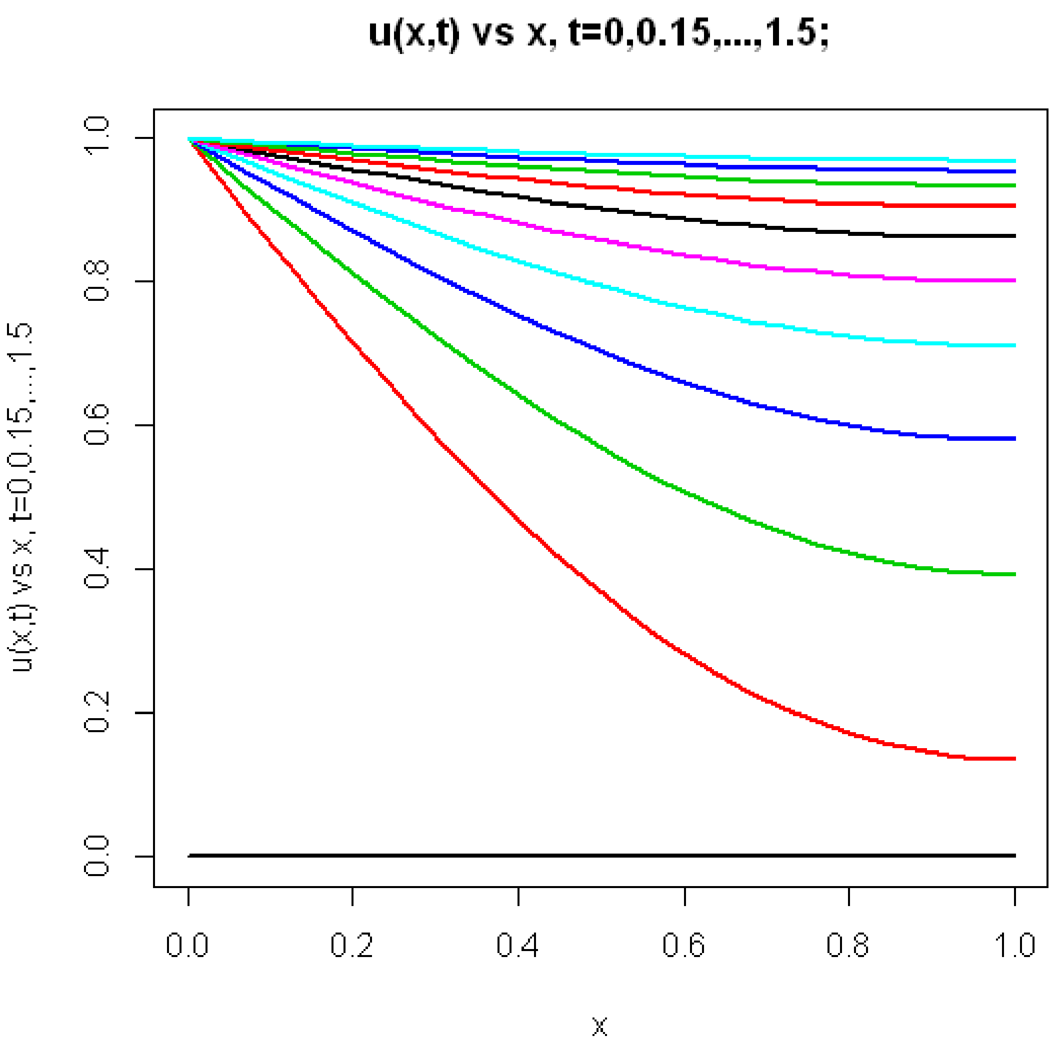
Analytical

| t | x | u(x,t) |
|---|---|---|
| 0.45 | 0.00 | 1.00000 |
| 0.45 | 0.20 | 0.87036 |
| 0.45 | 0.40 | 0.75342 |
| 0.45 | 0.60 | 0.66064 |
| 0.45 | 0.80 | 0.60107 |
| 0.45 | 1.00 | 0.58055 |

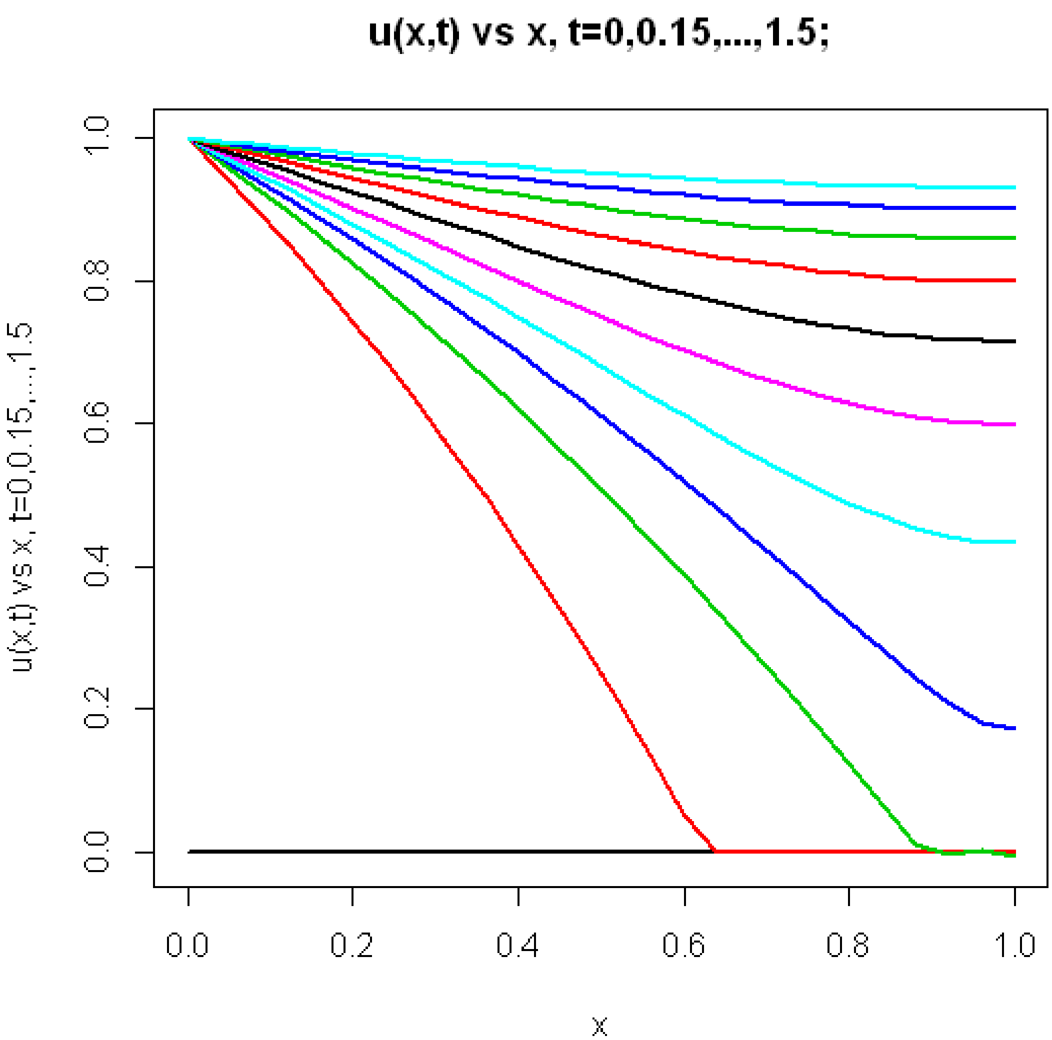
$u(x,t)$ **vs** $x$ **with** $t$ **as a parameter (Matlab) for** `ncase=1`



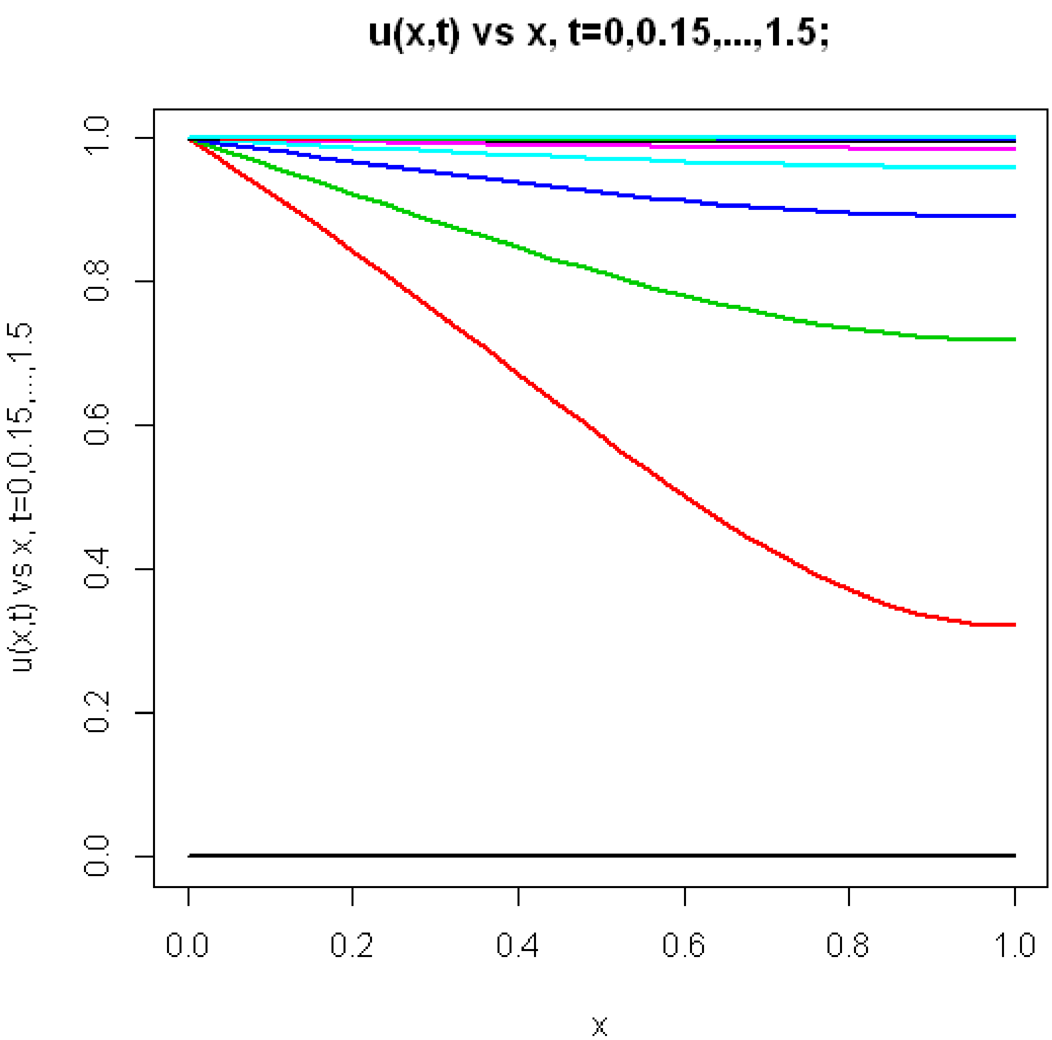
$u(x,t)$ **vs** $x$ **with** $t$ **as a parameter (R) for** `ncase=1`



$u(x,t)$ **vs** $x$ **with** $t$ **as a parameter (R) for** `ncase=2`

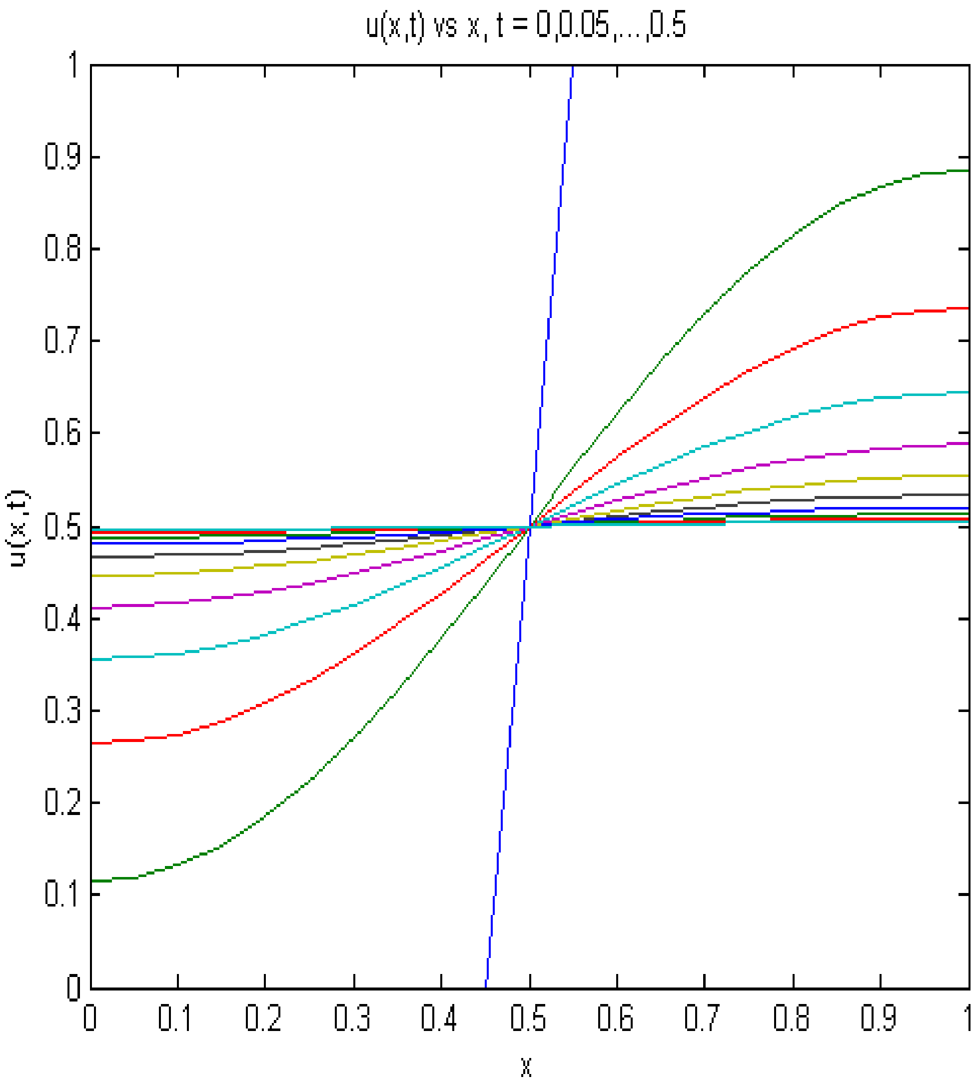


$u(x,t)$ **vs** $x$ **with** $t$ **as a parameter (R) for** `ncase=3`



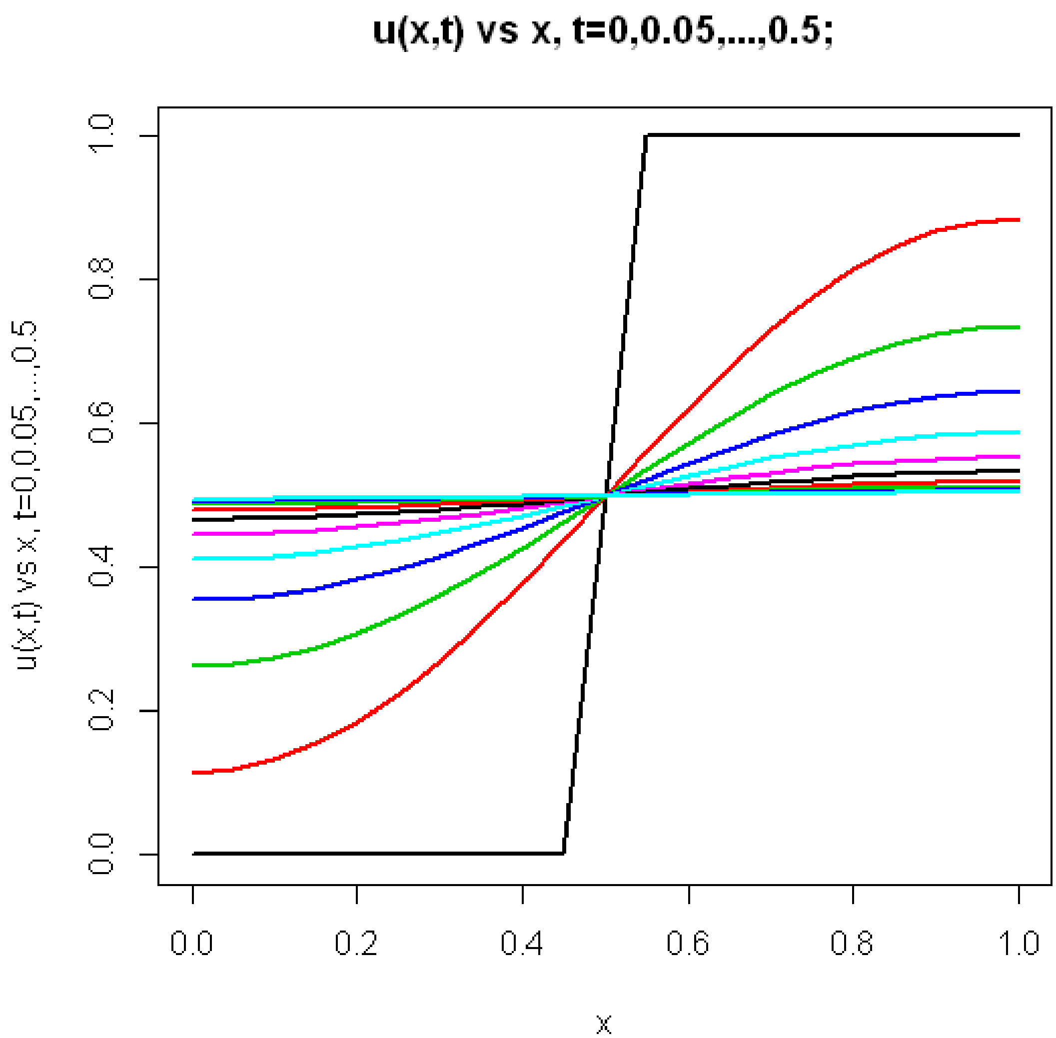
**$u(x,t)$ vs $x$ with $t$ as a parameter (Matlab) for `ncase=1`**

**Discontinuous IC**

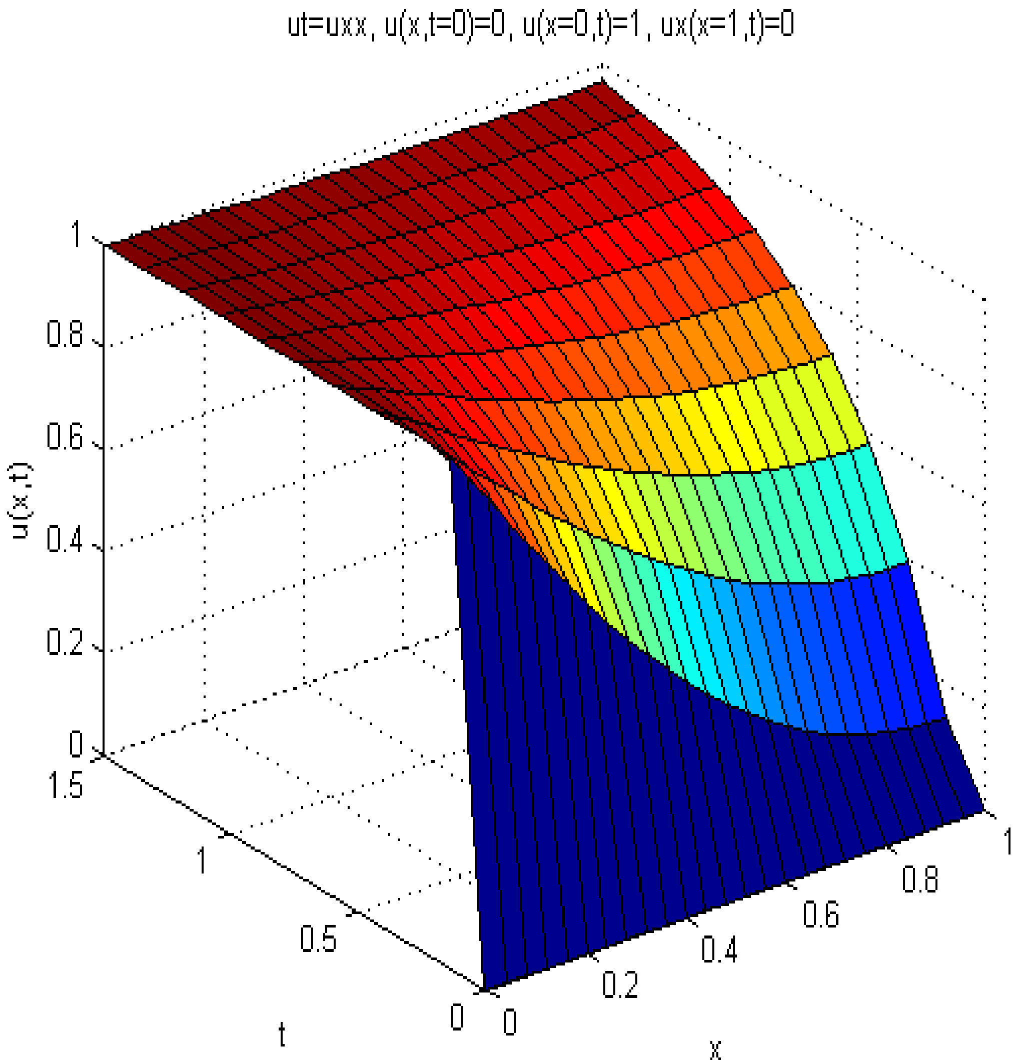


$u(x,t)$ **vs** $x$ **with** $t$ **as a parameter (R) for** `ncase=1`

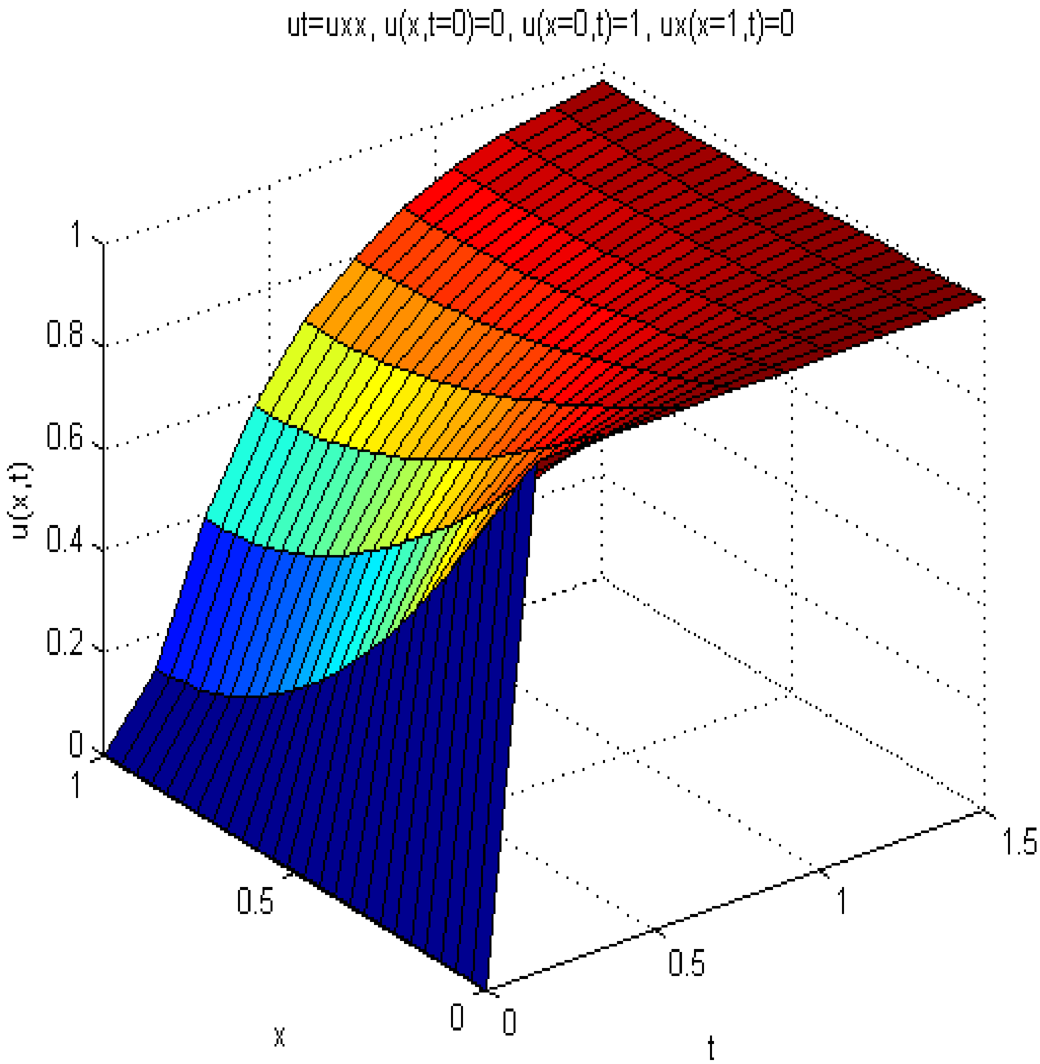
**Discontinuous IC**



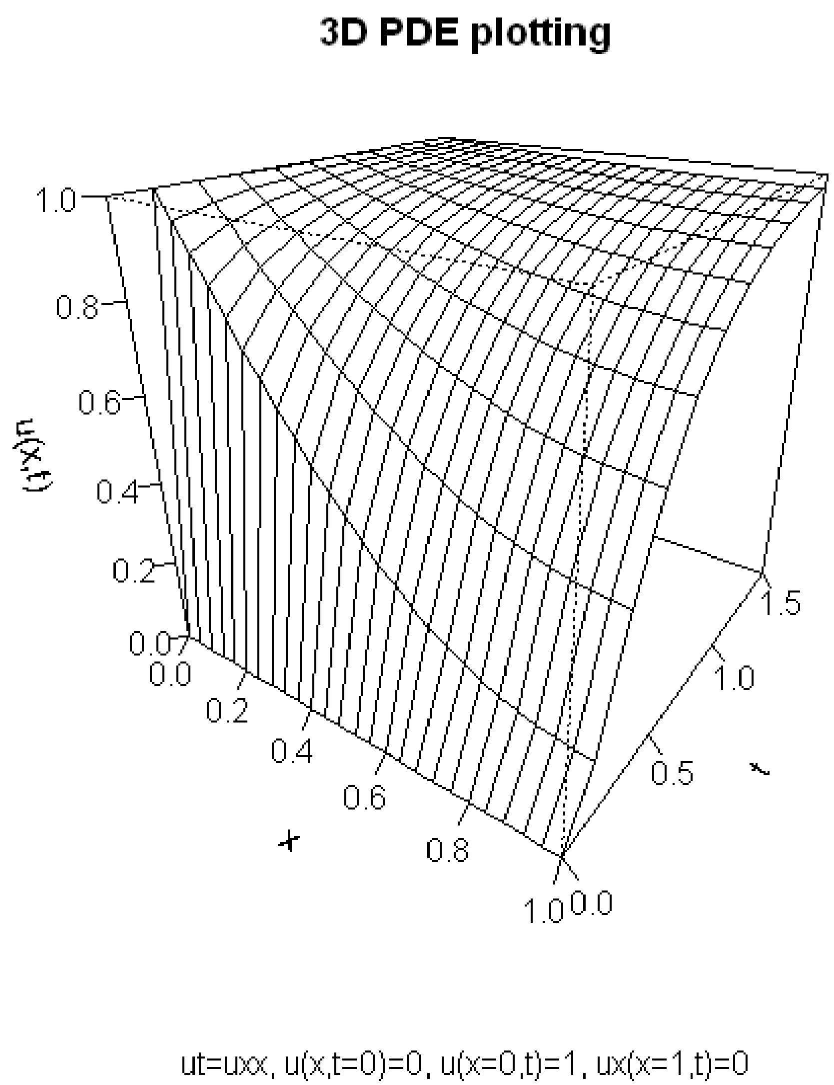
**3D perspective of $u(x,t)$ vs $x$ and $t$ from `surf` (Matlab)**



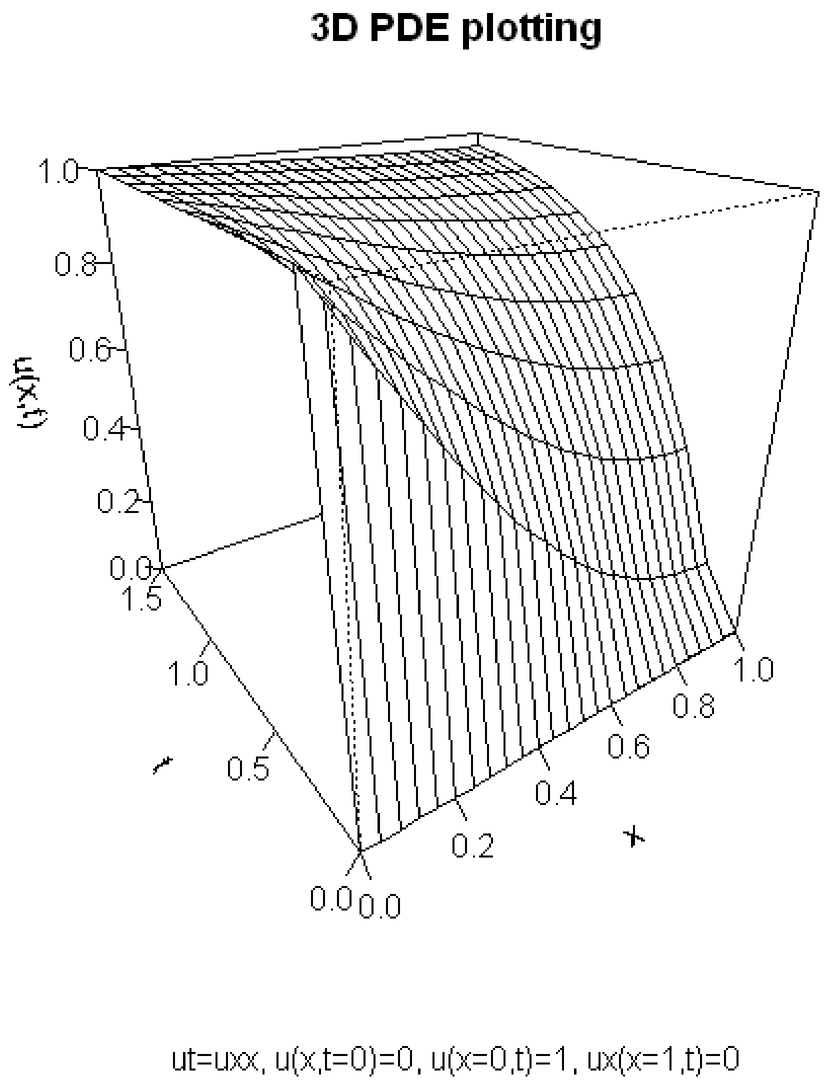
**3D perspective of $u(x, t)$ vs $x$ and $t$ from `surf` (Matlab)**



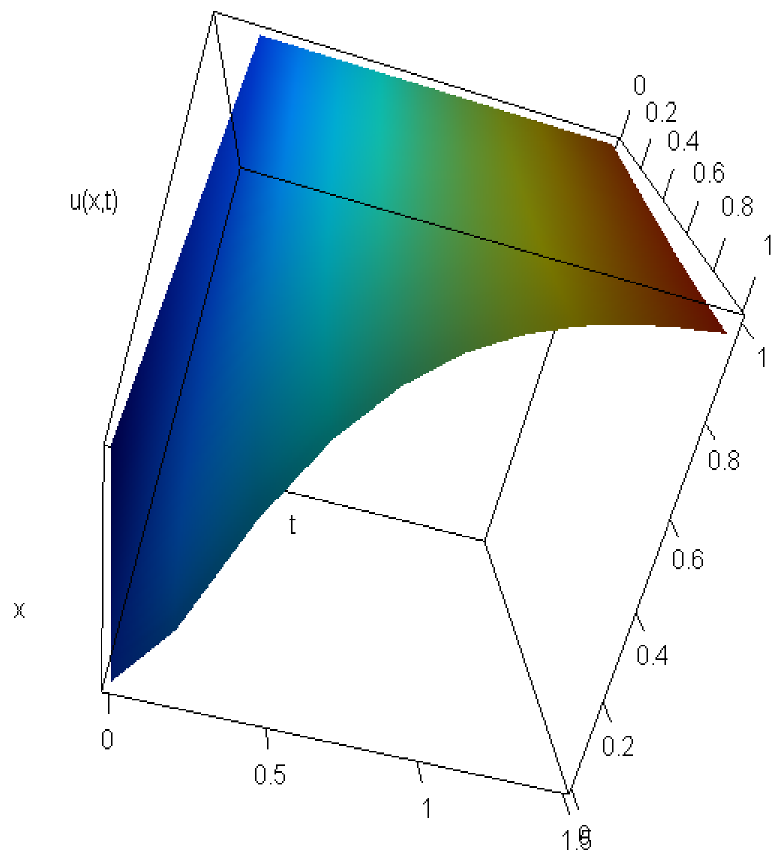
**3D perspective of $u(x,t)$ vs $x$ and $t$ from `persp` (R), $\theta = 35$**



**3D perspective of $u(x,t)$ vs $x$ and $t$ from `persp` (R), $\theta = -35$**



## 3D perspective of $u(x,t)$ vs $x$ and $t$ from `persp3d` (R)



**In summary, R is:**

- **A quality scientific computing system**
  - **An extensive library of utilities and applications (e.g., deSolve, persp3d, Rstudio)**
  - **A fully functional alternative to Matlab**

**In summary, R is:**

- **A quality scientific computing system**
  - **An extensive library of utilities and applications (e.g., deSolve, persp3d, Rstudio)**
  - **A fully functional alternative to Matlab**
- **Open source (no cost)**

**In summary, R is:**

- **A quality scientific computing system**
  - **An extensive library of utilities and applications (e.g., deSolve, persp3d, Rstudio)**
  - **A fully functional alternative to Matlab**
- **Open source (no cost)**
- **Available to anyone with an Internet connection**

**In summary, R is:**

- **A quality scientific computing system**
  - **An extensive library of utilities and applications (e.g., deSolve, persp3d, Rstudio)**
  - **A fully functional alternative to Matlab**
- **Open source (no cost)**
- **Available to anyone with an Internet connection**
- **Easily downloaded and used (e.g., by faculty and students); WES glad to help**

**R is available from the Internet in Linux, Mac, Windows formats: http://cran.fhcrc.org/. Also, the ODE/DAE library deSolve[1] can be downloaded for use with R.**

**The editor Rstudio is recommended for use in working with R and for facilitating graphical output: http://rstudio.org/. deSolve can be conveniently downloaded via Rstudio.**

**References**

[1] **Soetaert, K., J. Cash and F. Mazzia (2012),** *Solving Differential Equations in R*, **Springer-Verlag, Heidelberg, Germany**

**Thank you.**

**Questions?**