Macsyma Introduction

Steps for PDE solving by Macsyma

- Obtaining solutions with PDEase2D is a simple four-step process:

  Create a mathematical model of the system.

  Translate the mathematical model into a PDEase2D problem descriptor file.

  Open MFE (Macsyma Front End) to run PDEase2D to produce a solution.

  Review, animate, and/or print the solution produced by PDEase2D.

MFE and PDEase2D

- You can use MFE to load the .pde problem descriptor file into a PDEase2D Interaction section. Or you can Edit-Insert a new PDEase Interaction section and enter the problem description. You can see over 140 sample .pde files and MFE notebooks containing problems in the PDEase2D₩demos directory.

- See the Macsyma and PDEase2D Front End Reference for more information about MFE.

- You can execute the PDEase2D commands by pressing (the PDEase2D button).

- PDEase2D uses MFE to make graphics. You can use the graphical and text features of MFE to examine any plot, change its attributes, make annotations, animate the plot, or print it. You can save the note with the graphics to review later.

- You can use the P button to pause the calculation. The P button then turns into an R for Resume calculation. You can erase the PDEase2D Output sections with the Eraser button

Language-Based Problem Specification

You can prepare a PDEase2D command script using any standard ASCII text editor (DOS EDIT.COM, Windows Notebook, or SUN OpenWindows TEXTEDIT).

You can then present PDEase2D with a simple, natural, and easy-to-learn language-based specification system.

Poisson's equation appears as div(grad(T)) + h=0, its normal mathematical form, and polynomials appear as A*u^2+B*u*v+C*v^2.

Powerful Galerkin Finite Element Spatial Dependence Solver

Space discretization

- Finite element modeling has been the preferred method for converting the spatial components of complex sets of continuous partial differential equations (with well defined boundary values) to a set of discrete nodal equations for numerical solving. In this method the spatial area of interest is gridded into small patches called finite elements over which the variables are represented by simple polynomials. PDEase2D uses the Galerkin Finite Element Method of weighted residuals with quadratic basis to convert continuous partial differential equations into discrete nodal equations.

- Fully Automated Adaptive Grid Refinement

  One difficult decision to make, when accuracy is required, in implementing the finite element method, is how to divide the area of interest into patches. Small patches require excessive computer time and memory. PDEase2D solves this problem for the user by starting with a coarse grid of triangular patches (the most universal patch geometry that can be selected) and then using an iterative process to refine the grid to suit the problem. After subdividing, recalculation is fast because of the good starting estimate provided by the previous iteration. In this way PDEase2D uses fine gridding only in those areas where sharp curvatures and tight geometries exist, providing near optimum speed and memory utilization.

Powerful Evolution Time Dependence Solver

- Evolution solvers have been the preferred method of breaking continuos time dependent boundary/initial value problems into discrete time slices which can then be solved by the Galerkin Finite Element Method of weighted residuals.
- PDEase2D uses a proprietary self adjusting evolution solver for time stepped problems. This method insures highly accurate results capable of following extremely rapid changes in time.

Initial and Boundary Conditions

- Fully Automated Time Step Refinement
- The PDEase2D evolution solver automatically and continuously adjusts its time step interval to reduce the time interval when PDEase2D encounters a region of rapid change and to increase the time step when PDEase2D encounters a region of slow change. This automatic refining of the time step interval relieves the user of the responsibility of predetermining a time step and provides maximum accuracy for a minimum number of time steps.

- Support for Both Value and Natural Boundary Conditions
- PDEase2D supports both value and natural boundary conditions. PDEase2D correctly interprets the natural boundary condition for equations written as the divergence of a vector, that is DIV(D)-=0, (the normal form for most heat and electrostatic problems) as the dot product of the outward directed normal and the vector D. PDEase2D correctly interprets the natural boundary condition for equations written as the curl of a vector, that is CURL(H)-J=0

- Automatic Handling of Internal Boundary Conditions in Multi-Region Problems
- When solving multi-region problems, the finite element method used by PDEase2D guarantees that when problems are expressed in either divergence form or curl form that at the region interfaces both the normal component of flux density and the tangential component of the field intensity are continuous.

Other features

- Eigenvalue "Modal" Analysis
- Using the "subspace iteration" method to reduce the number of degrees of freedom, PDEase2D, when requested, calculates and lists a user selected number of the smallest eigenvalues of specified linear systems.

- Non-Analytic Data Import and Export
- PDEase2D's unique table function allows it to import and export non-analytic (numerical) data. This feature can be used to import numerical data from programs such as experimental data gathering programs or to export data for special post processing.

- Automatic or User-Controlled Solution Flow
- Most linear and non-linear problems can be solved automatically by PDEase2D, using its built-in default selectors to control the internal flow of the solutions. For those especially difficult problems which do not converge well with the built-in defaults, any of the over twenty built-in default selectors normally used by PDEase2D to control the internal flow of the solution can be overridden by the user to improve problem convergence.

Continued

- Graphic Output Viewing and Recall
- PDEase2D produces a series of output graphic as MFE Graphics section. You can view them by scrolling through the notebook, print them. Or, if you used the animate directive, make an animation of any plot.

- Hardcopy using Windows
- You can use the Windows printer as its hardcopy device by printing notebook sections from MFE.

- Graphics with the Macsyma Front End
- You can animate any graphical output using MFE's powerful animation facilities.

- Data Import of Region Geometries and Boundary Conditions in DXF Format

Macsyma Programs

- Diffusion
- Macsyma demo

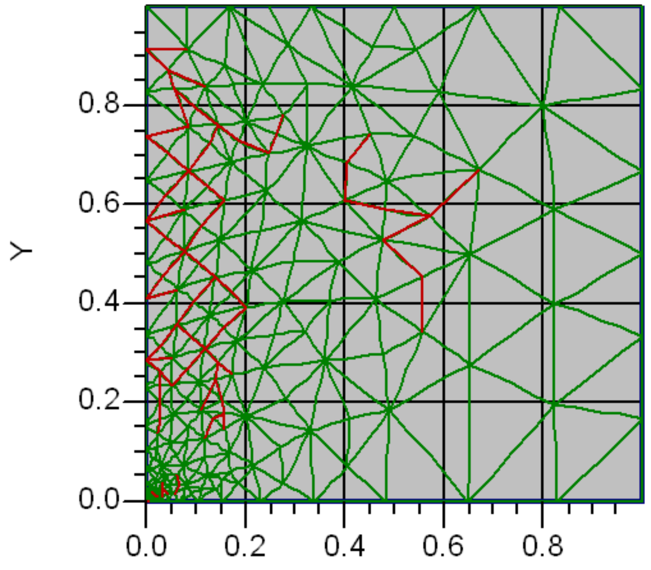
```
{Diffuse.pde
   In this example, we have recast a 1D initial-value problem as a
  2D boundary-value problem, with the scaled time variable
  represented by "Y".  Notice that this technique requires gridding
  and  simultaneous solution over the entire time-space domain.
   For the true initial-value treatment of this problem in 2 space
  dimensions, see DIFFUSET.PDE.


   This problem considers the thermally driven diffusion of a
  dopant into a solid from a constant source.  Parameters have
  been chosen to be those typically encountered in semiconduct
  diffusion.
  surface concentration = 1.8e20 atoms/cm**2
  diffusion coefficient = 3.0e-15 cm**2/sec


   The boundary condition natural(u) = 0 chosen for all distance at
  time=1 and for all time at distance=1 implies that these
  boundaries are sufficiently remote to have no effect on the
  process.     The analytic solution for this problem (the
  complementary error function) is compared with the PDEase2
  solution.
  A "gridding feature" allows us to short-cut several iterations of
  grid refinement by focussing the gridder on the area of interest. }
```

```
Title
"Constant Surface Source Diffusion (Pseudo-2D)“

Select
  gridlimit = 15
  alias(y) = "time"
  alias(x) = "distance“

Variables
  u

Definitions
  concs = 1.8e8   {surface concentration atom/micron**3}
  D = 1.1e-2      {diffusion coefficient micron**2/hr}
  conc = concs*u
  cexact = concs*erfc(x/(2*sqrt(D*y)))  {analytic solution}
  uexact = erfc(x/(2*sqrt(D*y)))

Initial values
  u = 0
```

```
Equations
  dx(D*dx(u)) - dy(u) = 0

Boundaries
  region 1
    start(0,0)
    value(u) = 0.0001
    line to (1,0)
    natural(u) = 0
    line to (1,1)
    natural(u) = 0
    line to (0,1)
    value(u) = 1
    line finish

  feature
    { a "Gridding Feature" for gridding control }
    start(0.01,0) line to (0.01,0.01) to (0,0.01)
```

```
Monitors
  contour(u)

Plots
  contour(u) as "C/Cs"
  contour(conc)
  contour(conc) zoom(0,0,0.2,0.2)
  contour(u-uexact) as "Relative Error"
  contour(u-uexact) zoom(0.05,0.05,0.9,0.9) as
   "Relative Error"
  elevation(conc) from (0,.2) to (.4,.2)
  elevation(u-uexact) from (0,.2) to (.4,.2)
        as "Relative Error"
  surface(u) interactive as "C/Cs"

End
```

Constant Surface Source Diffusion (Pseudo-2D)

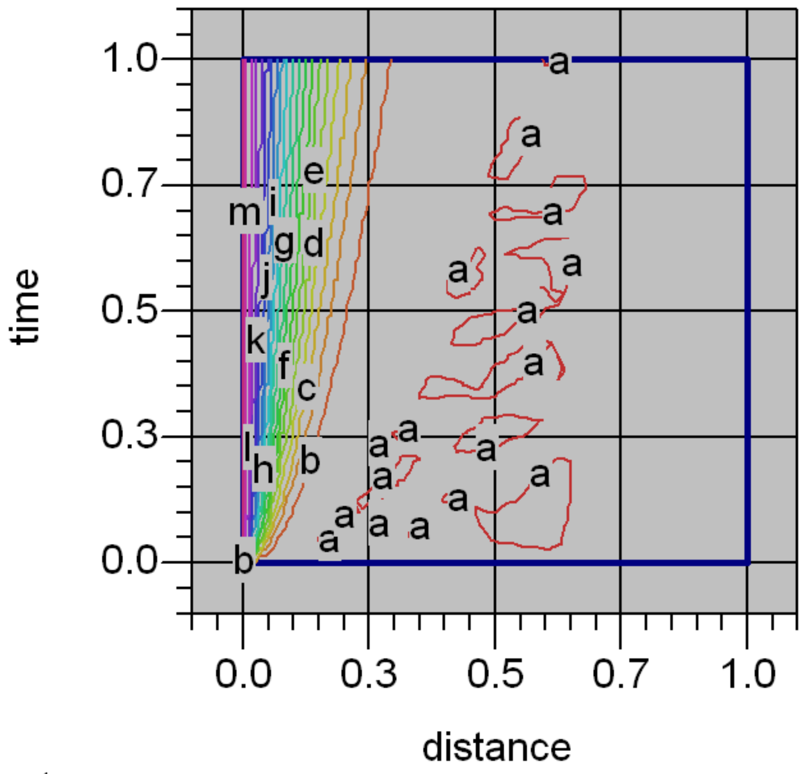


Default Grid Monitor

09-29-09 - 14:29:51

Constant Surface Source Diffusion (Pseudo-2D)

u



Contours:
----
min: -0.0002472
a: 0
b: 0.05
c: 0.15
d: 0.25
e: 0.3
f: 0.4
g: 0.5
h: 0.55
i: 0.65
j: 0.7
k: 0.8
l: 0.9
m: 1
max: 1

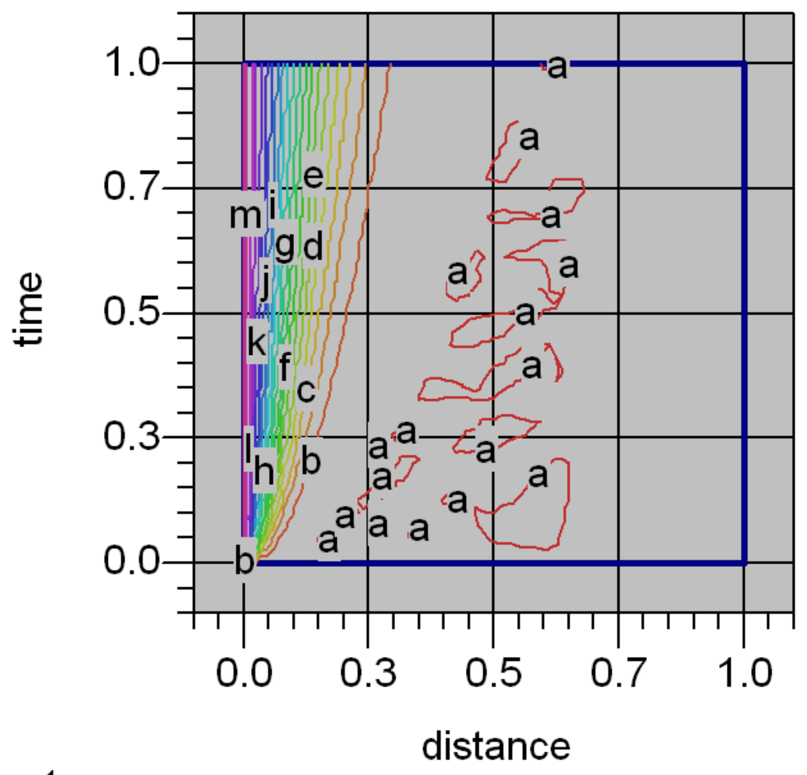
0<=X<=1
0<=Y<=1
-0.0002472<=c(X,Y)<=1
Vol= 0.07915

PdzMon01Cont

09-29-09 - 14:29:52

Constant Surface Source Diffusion (Pseudo-2D)

u



Contours:

----

min: -0.0002472

a: 0

b: 0.05

c: 0.15

d: 0.25

e: 0.3

f: 0.4

g: 0.5

h: 0.55

i: 0.65

j: 0.7

k: 0.8

l: 0.9

m: 1

max: 1

0<=X<=1

0<=Y<=1

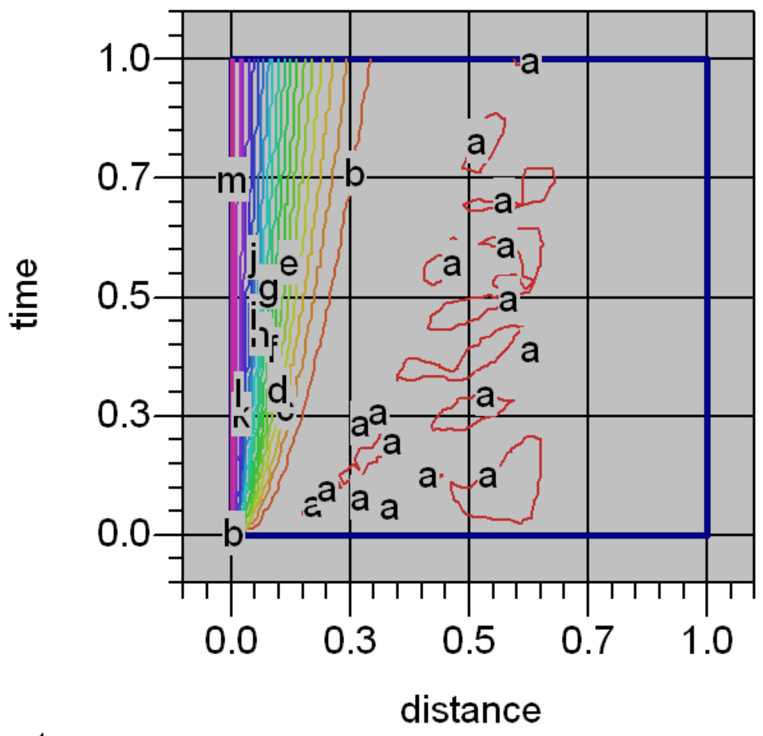
-0.0002472<=c(X,Y)<=1

Vol= 0.07915

PdzMon01Cont

09-29-09 - 14:29:52

Constant Surface Source Diffusion (Pseudo-2D)
C/Cs



Contours:
----
min: -0.0002472
a: 0
b: 0.05
c: 0.15
d: 0.25
e: 0.3
f: 0.4
g: 0.5
h: 0.55
i: 0.65
j: 0.7
k: 0.8
l: 0.9
m: 1
max: 1

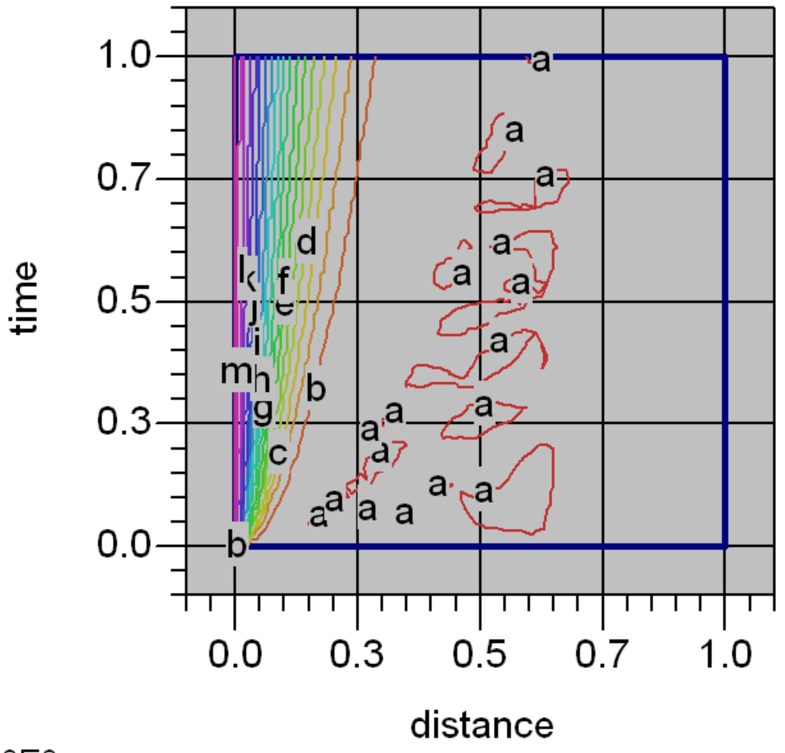
0<=X<=1
0<=Y<=1
-0.0002472<=c(X,Y)<=1
Vol= 0.07915

PdzPlt02Cont: Gr=4 err=0.0003246

09-29-09 - 14:29:52

Constant Surface Source Diffusion (Pseudo-2D)

conc



Contours:

----

min: -4.449E4

a: 0

b: 1E7

c: 3E7

d: 4E7

e: 6E7

f: 7E7

g: 9E7

h: 1E8

i: 1.2E8

j: 1.3E8

k: 1.5E8

l: 1.6E8

m: 1.8E8

max: 1.8E8

0<=X<=1

0<=Y<=1

-4.449E4<=c(X,Y)<=1.8E8

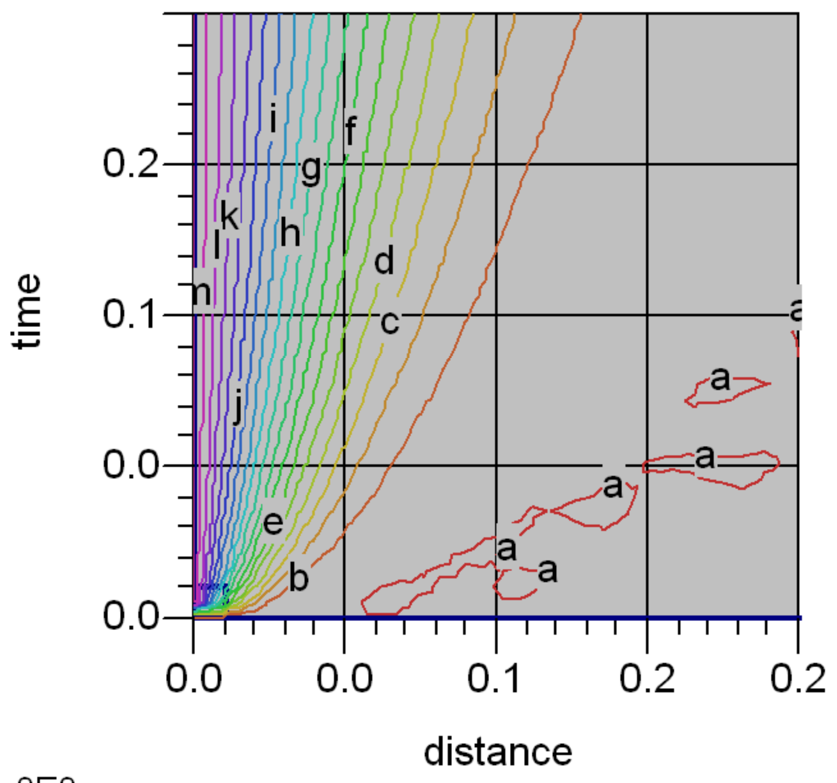
Vol= 1.425E7

PdzPlt03Cont: Gr=4 err=0.0003246

09-29-09 - 14:29:52

Constant Surface Source Diffusion (Pseudo-2D)

conc



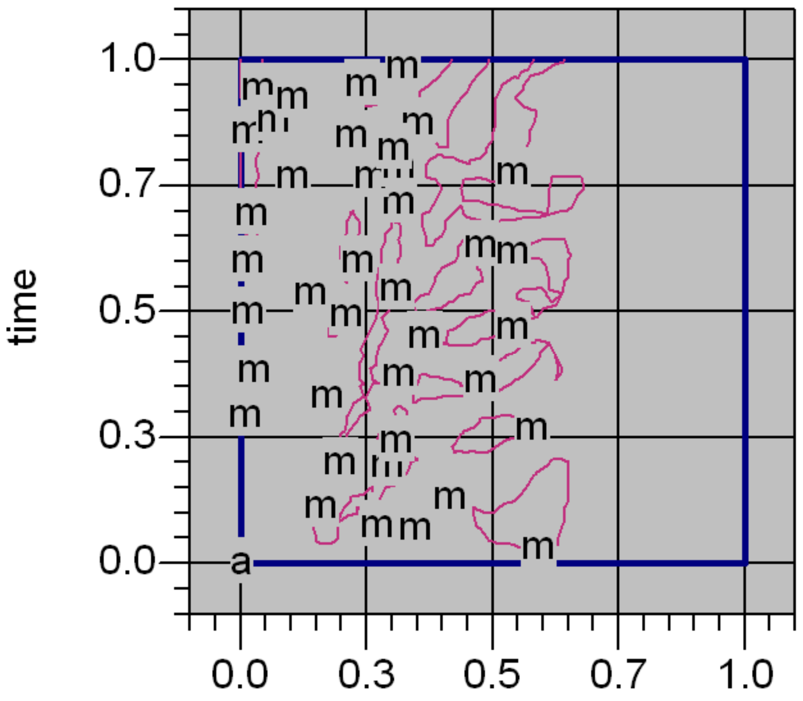
Contours:
----
min: -1.135E5
a: 0
b: 1E7
c: 3E7
d: 4E7
e: 6E7
f: 7E7
g: 9E7
h: 1E8
i: 1.2E8
j: 1.3E8
k: 1.5E8
l: 1.6E8
m: 1.8E8
max: 1.8E8

0<=X<=0.2
0<=Y<=0.2
-1.135E5<=c(X,Y)<=1.8E8
Vol= 1.287E6

PdzPlt04Cont: Gr=4 err=0.0003246

09-29-09 - 14:29:52

Constant Surface Source Diffusion (Pseudo-2D)
Relative Error

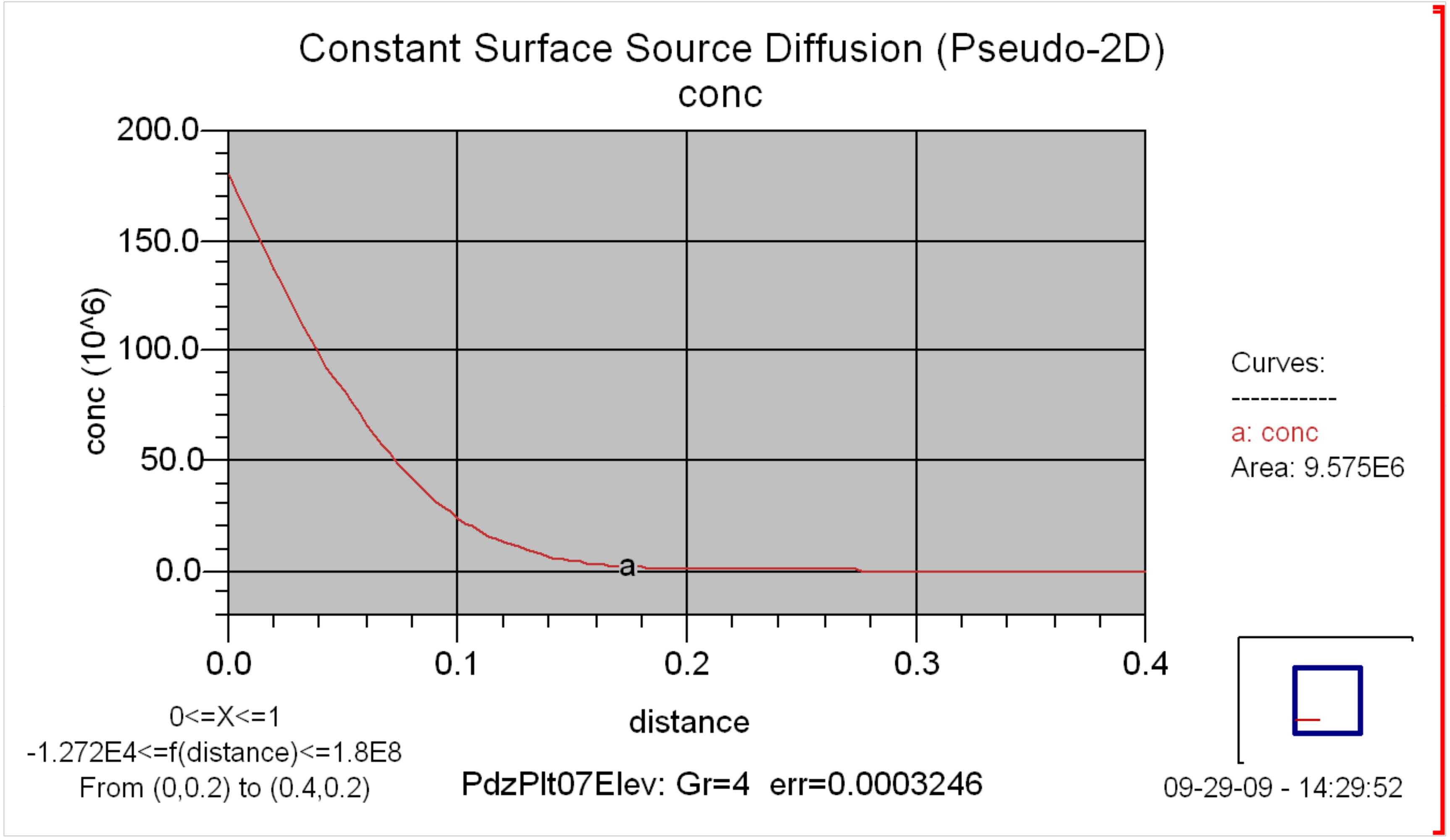


Contours:
----
min: -0.9999
a: -0.95
b: -0.9
c: -0.8
d: -0.75
e: -0.65
f: -0.6
g: -0.5
h: -0.4
i: -0.35
j: -0.25
k: -0.2
l: -0.1
m: 0
max: 0.03972

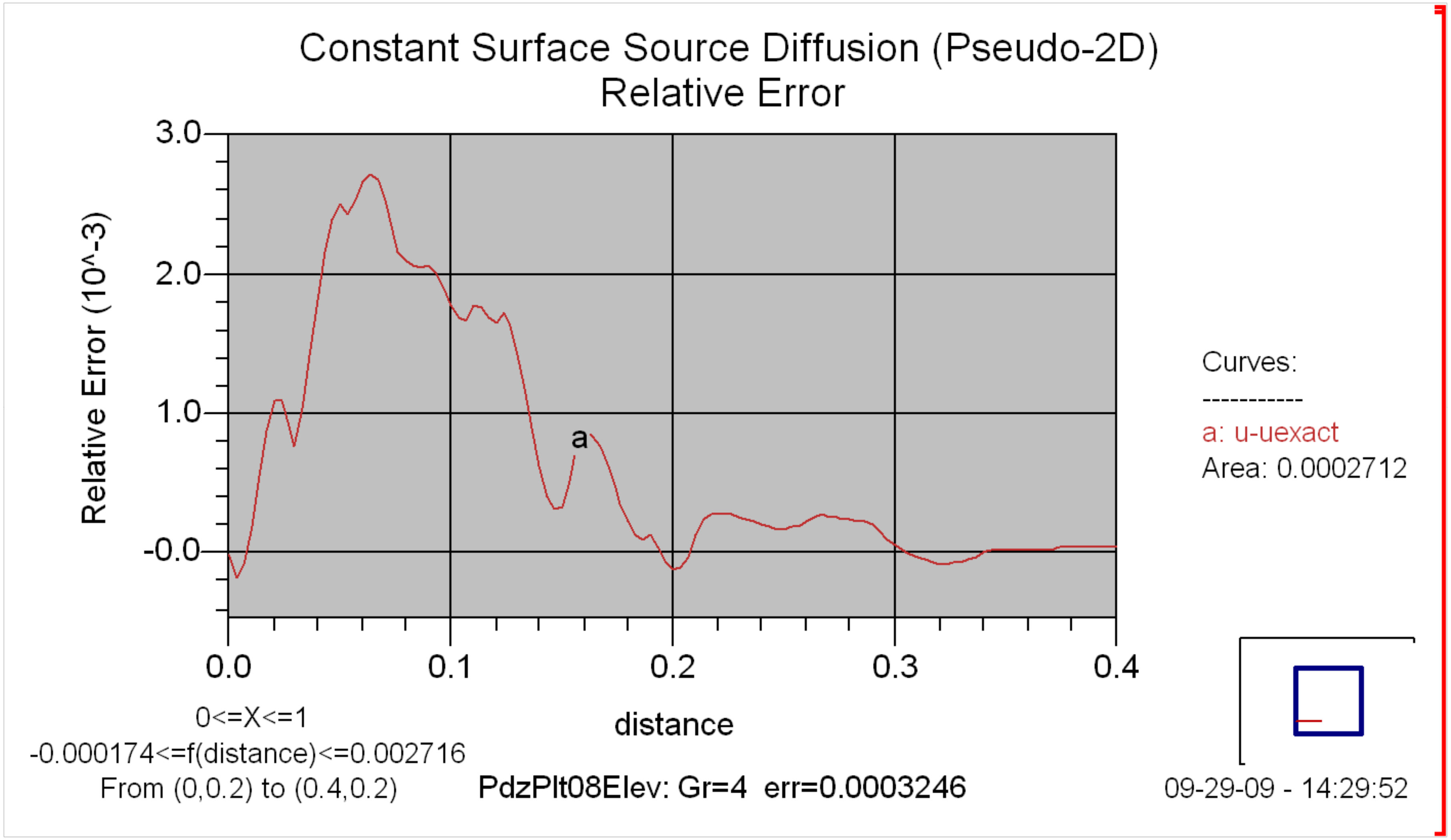
$0<=X<=1$
$0<=Y<=1$
$-0.9999<=c(X,Y)<=0.03972$
Vol=0.0002508

PdzPlt05Cont: Gr=4 err=0.0003246

09-29-09 - 14:29:52

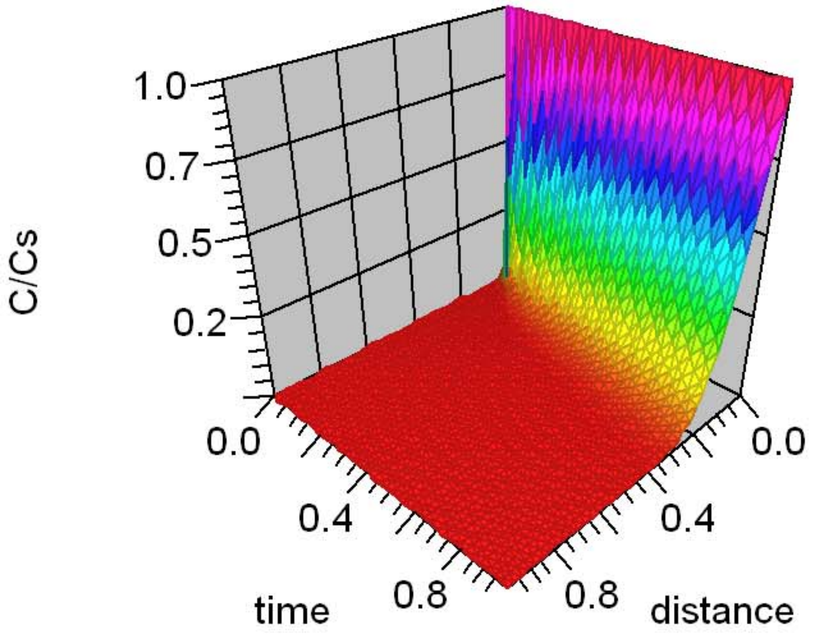


Constant Surface Source Diffusion (Pseudo-2D)
conc
conc (10^6)
200.0
150.0
100.0
50.0
0.0
a
0.0
0.1
0.2
0.3
0.4
distance
Curves:
----
a: conc
Area: 9.575E6
0<=X<=1
-1.272E4<=f(distance)<=1.8E8
From (0,0.2) to (0.4,0.2)
PdzPlt07Elev: Gr=4 err=0.0003246
09-29-09 - 14:29:52


Constant Surface Source Diffusion (Pseudo-2D)
Relative Error
Relative Error (10^-3)
3.0
2.0
1.0
-0.0
0.0
0.1
0.2
0.3
0.4
distance
a
Curves:
----
a: u-uexact
Area: 0.0002712
0<=X<=1
-0.000174<=f(distance)<=0.002716
From (0,0.2) to (0.4,0.2)
PdzPlt08Elev: Gr=4 err=0.0003246
09-29-09 - 14:29:52

Constant Surface Source Diffusion (Pseudo-2D)

C/Cs



0<=X<=1

0<=Y<=1

-0.0002472<=s(X,Y)<=1

Vol= 0.07915

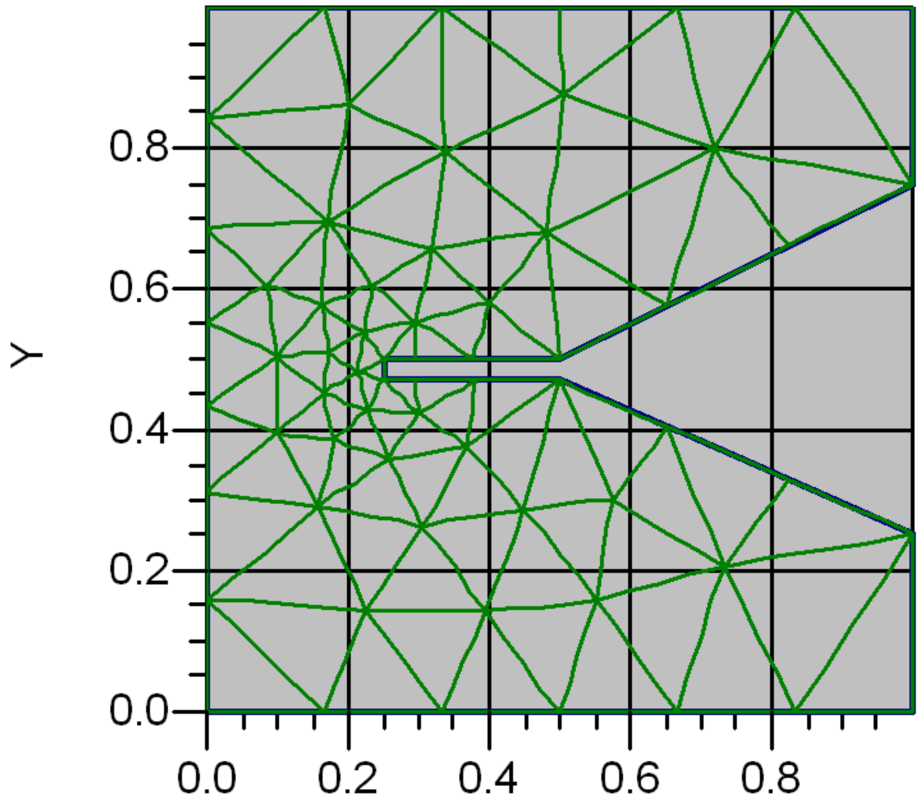
PdzPlt09Surf: Gr=4 err=0.0003246

09-29-09 - 14:29:52

```
{ Macsdemo.pde }

Title
"Macsyma Equation Test"

Select
  macsyma
  errlim = 0.1

Coordinates
  cartesian(x,y)

Definitions
  b = 1.0

Variables
  u

Initial values
  u = 1-(x-0.5)**2-(y-0.5)**2
```

```
Equations
  'DIFF(b*'DIFF(u,x,1),x,1)
   + 'DIFF(b*'DIFF(u,y),y)
   + 4*b = 0;

Boundaries
  region 1
    start(0,0)
    value(u)=0
    line to (1,0)
    to (1,0.25)
    to (0.5,0.47)
    to (0.25,0.47)
    to (0.25,0.50)
    to (0.5,0.50)
    to (1,0.75)
    to (1,1)
    to (0,1)
    finish

Plots
  surface(u)

End
```

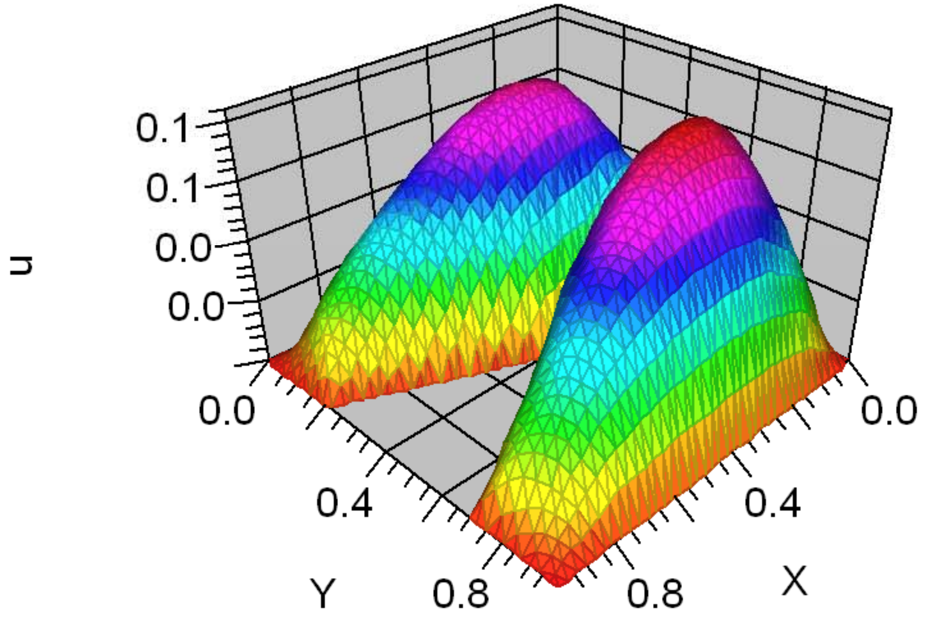
Macsyma Equation Test



Default Grid Monitor

09-29-09 - 14:36:29

Macsyma Equation Test
u



0<=X<=1
0<=Y<=1
-2.151E-8<=s(X,Y)<=0.1055
Vol= 0.04334

PdzPlt01Surf: Gr=1 err=0.009384

09-29-09 - 14:36:29