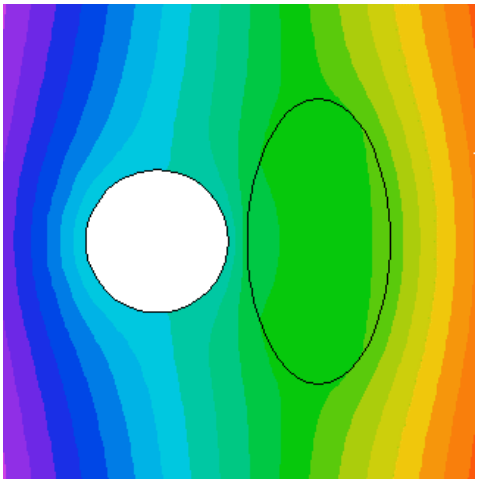
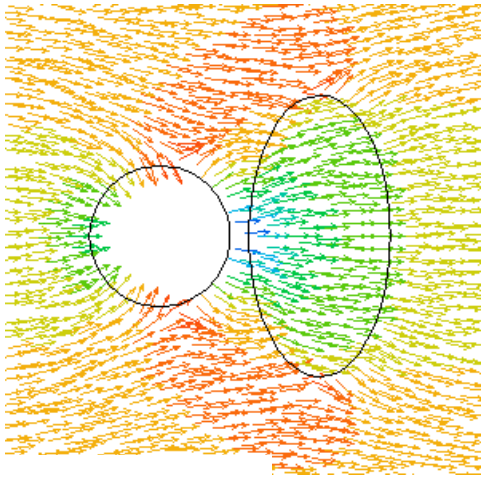
Fields of Physics

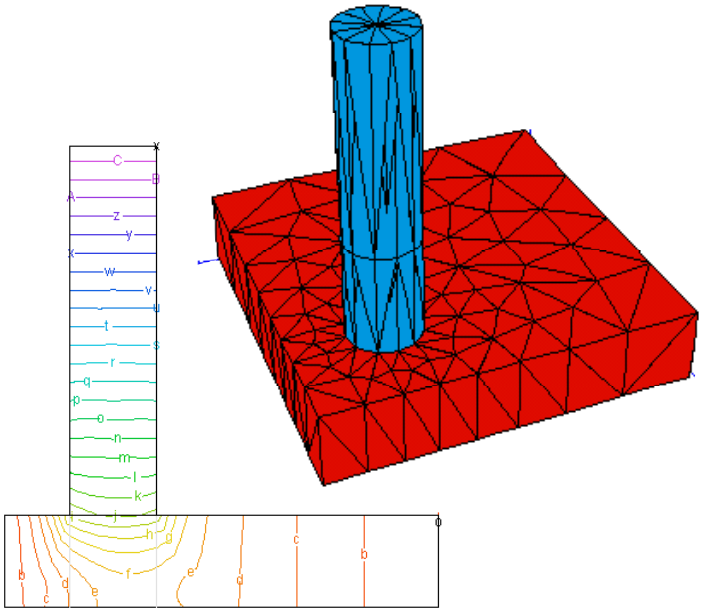
by Finite Element Analysis

Electricity, Magnetism, and Heat in 1D, 2D, and 3D

Using FlexPDE® Version 5







GB Publishing

Gunnar Backstrom

Seaching for Words

The *Acrobat* program lets you search for words and even word combinations. After selecting *Edit*, *Find* (or pushing the keys *Ctrl+f*) it suffices to enter the item of interest. This feature makes an *index* unnecessary.

Navigating by Bookmarks

The table of contents is always available and may be brought up to the left of the text by pushing *F5*. A simple click on a subtitle opens that section immediately. Push *F5* once more, and the table of contents becomes hidden.

Preface

During my early years as a student at Uppsala University, Sweden, I took a special interest in partial differential equations. To my mind, these equations seemed to be important keys to the predictions in space and time that physics is all about. Nevertheless, undergraduates still regard them as esoteric objects that are part of the curriculum but sadly useless.

Partial differential equations are indeed cumbersome, since few analytical methods of solution exist, and numerical tools of reasonable generality have been lacking. During a visit in the San Francisco Bay area in 1992 I happened to see an advertisement about a "Solver for Partial Differential Equations", which caught my attention since I was currently planning a course in applied finite element analysis (FEA). When I started to work with this software I had the feeling that a curtain was drawn, and that I could suddenly see the fields of physics through the graphics on the computer screen. I was absolutely stunned to find that almost any problem could be solved, even if it involved several complicated, simultaneous equations and often even if they were not linear. I simply could not keep this extraordinary experience to myself.

My first book, *Fields of Physics on the PC by Finite Element Analysis* (1994), was based on a DOS version and covered electro- and magnetostatics, heat conduction, elasticity, and liquid flow. The 2nd edition (1996) exploited the additional software features that had appeared during the intervening period. It also comprised applications in the areas of vibration in fluids and solids, electromagnetic waves, and quantum mechanics.

The software has since evolved to a significant degree. The pioneer created a new program, FlexPDE®, written in C language for Windows 95/98/NT. This program was the basis of a third book, *Fields of Physics by Finite Element Analysis, An Introduction* (1998), with applications to electromagnetism and heat, exploiting the

convenient new features that had become available. It was intended as a guide for students in individual or supervised laboratory work, illustrating the corresponding physics courses.

The present textbook exploits version 5 of FlexPDE, which is marketed both as a Professional Version and a Student Version. The latter is limited to five simultaneous PDEs and also restricts the number of node points in space. This version of FlexPDE may be used with Linux and MacIntosh, as well as under MS Windows.

I am deeply indebted to my late friend Dr. Russell Ross, formerly of the University of East Anglia, UK, for reading and criticizing earlier typescripts, as well as for working through the examples. His intelligence and vigilance made him the ideal pilot user.

Finally I gratefully acknowledge the support of Mr. Robert Nelson, PDE Solutions, Inc., the demon programmer behind FlexPDE®. His prompt attention to my innumerable queries about the operation of his software was essential to the success of this work.

Gunnar Backstrom

The finite-element software package used for this book (FlexPDE®) is marketed by

PDE Solutions Inc
PO Box 4217, Antioch, CA 94531-4217, USA
Phone: +1 925 776 2407
Fax: +1 925 776 2406
Email: sales@pdesolutions.com
http://www.pdesolutions.com

Contents

Magnetism

Heat

Metal and Semiconductor Physics

Three-Dimensional Problems

Appendix

1 Introduction

The classical fields of physics are governed by partial differential equations. Some exact solutions are available, but not commonly in cases of practical interest. Even when solutions exist, they tend to be complicated and must be evaluated numerically for graphical presentation in digestible form.

Finite element analysis (FEA) is a numerical approach in which the field domain under study is divided into a multitude of regions, each giving rise to one or more equations. The main task is to solve all these hundreds of simultaneous equations, which was impracticable before the days of transistorized computers.

An enormous number of textbooks on finite element analysis exist today, and they all present the various mathematical procedures in detail. They assume that the reader will make calculations on simple systems by hand and then type FORTRAN code or other, provided in an appendix, in order to handle more complex situations.

Some thirty years ago, the choice was between writing one's own programs or abstaining completely from FEA. Today, expert software saves students from worrying about programming strategy, formatting, indices and graphics. The tools of FEA are now on anyone's desk, since the average personal computer is adequate for solving a large number of such problems.

Now a program exists that permits you to enter the equations and the boundary conditions required by your mathematical model, solve them automatically, and present the results graphically in a variety of ways.

The software that achieves these wonders is FlexPDE® from PDE Solutions Inc. One of the attractive applications of this program is in university education. An extremely large range of problems involving classical fields may easily be studied in detail, and with realistic boundary conditions. Non-linear equations and boundary conditions as well as space-dependent materials properties are no longer serious

limitations. The number of cases for study is virtually unlimited, and by solving problems under various conditions the student may develop an intuitive feeling for how fields behave.

No one would even consider taking courses in Object Oriented Programming, with applications to advanced string, windows and mouse handling, before using a modern word processor. We all leave such chores to specialists, who spend years learning the tricks of their profession and then devote more years to produce the software we use every day.

Numerical algorithms and programming really have little to do with the concepts of physical fields and if months, or even years, may be gained by skipping the details of these items, then there is no reason to hesitate. Every scientist has to make a choice about what to learn and what to leave alone. No one can master all of physics, not even all of classical physics. The real choice is between additional ignorance within the discipline of physics or within some adjacent field.

On the other hand, no scientist would be content to use a numerical toolbox, such as FlexPDE, without knowing at least the principles of operation. Some details are even essential for the formulation of boundary conditions. In an attempt to include a general description of the method and at the same time de-emphasize its importance to the reader, I have included an appendix on this subject.

The main purpose of this volume is to illustrate how fields change when you modify the geometry, introduce different materials, and change the boundary conditions. It should be regarded as a companion to ordinary physics textbooks. Most of the analytic solutions in such texts may henceforth be replaced by FEA calculations, which yield better understanding of the physical phenomena.

Page References

As a reader, I have found that books generally contain much unnecessary numbering, which is not only redundant but makes the text tiresome to read. This made me choose a minimalist system for internal references. Strictly speaking, the page number is the only coordinate necessary. Hence, I have refrained from numbering figures

and equations. Most often, there is only one figure on a page, and in any case the risk of ambiguity is exceedingly small. Equations of special importance have been marked by the bullet symbol ● for later reference. Hence, p.63●2 would point at the 2nd bullet on page 63. Another simplification is that figures are discussed in the current text, just before or after the figure.

The external reference[2p37] is to be understood as "source 2, page 37".

How to use this Book

This volume aspires to confer understanding of the fields of electromagnetism, heat transport, and electronic conduction by numerical solutions of elementary problems. In a few, particularly simple, cases we shall compare to known analytic solutions, but in most cases only numerical methods are practicable.

The FEA method offers the advantage that similar problems may be solved by the same procedure, mostly by mere modification of the input data. Analytic methods, on the other hand, may require a radically different formalism for handling a slightly different situation. In the case of non-linear equations, numeric solution normally is the only option available.

To the ordinary physicist or engineer there is little point in studying the analytic techniques for solving PDEs. This toil is for specialized mathematicians, who devote their lives to extending the range of problems tractable by exact means. For anyone interested in applications, the analytic approach to PDEs is extremely time-consuming, and the mathematical apparatus tends to obscure the essential physical principles involved.

A better use of the student's time and effort is to consider various aspects of FEA solutions that illustrate the role of boundary conditions and materials properties. The syntactic rules of FlexPDE are simple and direct and may be learnt while solving elementary problems. The investment made in getting acquainted with FlexPDE is profitable, since PDEs occur in virtually all areas of physics and engineering.

FlexPDE allows us to treat finite-element analysis as a mathematical tool, which means that the user does not have to know about the numerous algorithms that constitute this powerful program package. Basically, the latter is similar to other, more expensive, products on the market. The insight gained by using FlexPDE will thus prove to be valuable for future work with other programs.

Here are a few principles to keep in mind while studying this book.

- Do not be afraid of making errors. The computer does not go up in smoke, and by correcting mistakes you learn the syntax.
- It is not enough to *read* the book. It shows a large number of plotted results, but running the files yields additional figures that are essential to understanding.
- Typing descriptor files is a way of learning. These lists illustrate the practice of finite-element calculations, and the command words constitute the language of FlexPDE.
- As you work with descriptors, take the opportunity of displaying additional plots.
- Be inquisitive! Do not accept FlexPDE commands until you understand what they achieve. Make a small test on your own by a short descriptor file. Occasionally, modify examples in the book to watch the effect.
- Allot time for the exercises. Some of them are simple variations of examples in the same chapter. Others expect you to be more independent. The important thing is to be active.
- Learn by doing!

2 Graphical Facilities

When a solution to a partial differential equation has been obtained, one would normally want to plot the results. Even if this is the last stage of the process, it is the easiest one to learn. For this reason we shall begin by studying the various modes of graphical presentation available in FlexPDE.

Downloading the Free FlexPDE Program

There are two main modifications of the FlexPDE program, the *Professional* Version and the (free) *Student* Version. These are identical, except that in the latter the number of node points in space and the number of equations are limited. This limitation is not serious, however, since it permits almost all of the examples given in this book to be run. In order to obtain a key file to open the Student Version you just have to supply your name and address.

Both versions plus a number of examples are available on the Internet under the address

www.pdesolutions.com

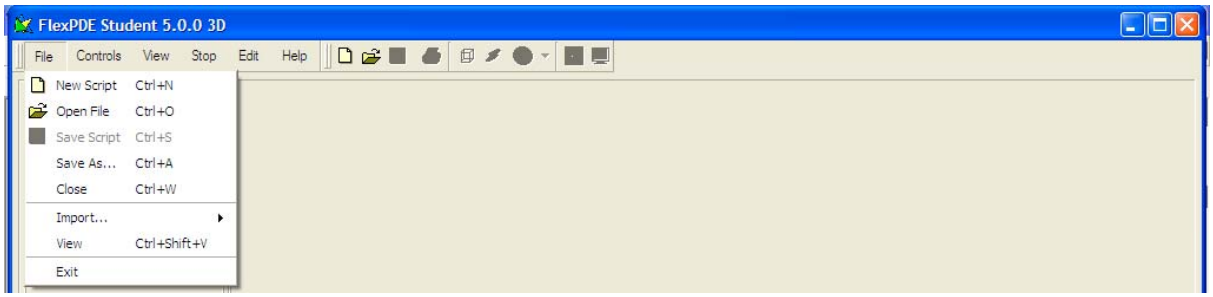
At this web site you are invited to *download* the latest version of the program chosen. You should store the incoming file in a suitable folder, such as *flexpde5*. The Student Version should be kept in a separate folder, such as *flexpde5s*. After the data transmission, which could take several minutes, you double-click on the file icon to *install* the program. This final stage only takes seconds to complete.

Immediately after installation, FlexPDE is ready to be used. You can now run the sample files included, and you may even modify the numerical input values and obtain completely new results. A number of the descriptor files found in this e-book will also be obtained on downloading and may be run and modified for demonstration purposes.

Before starting to work with the program you should create a special folder for your FlexPDE applications. Even if you could run them from any folder, it is wise to create a special one, say *flexpde_exa*, for your own problem descriptor files (scripts). First open *My Computer*, select (C:) and then *File, New, Folder* and type *flexpde_exa* over *NewFolder*.

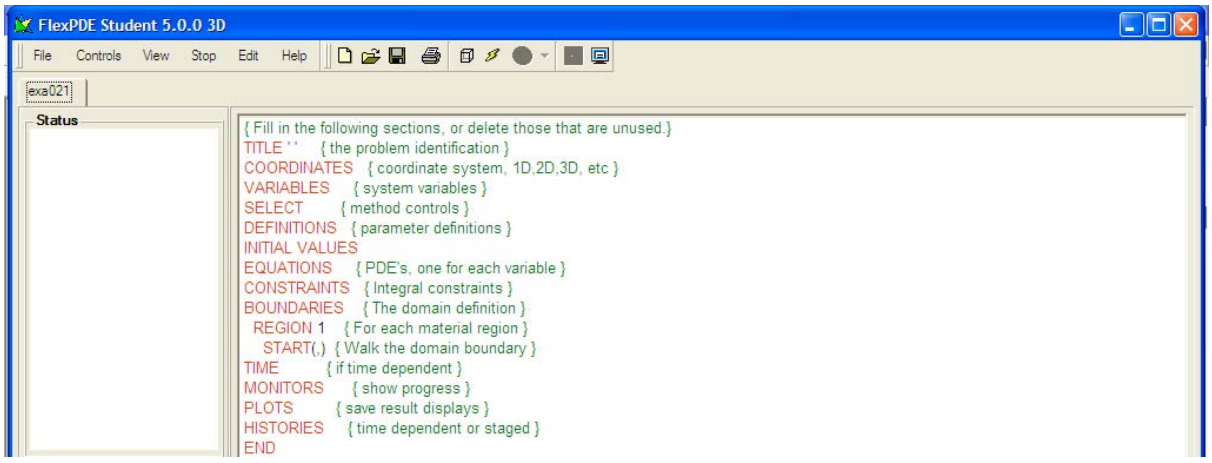
The FlexPDE Editor

Double-clicking on the FlexPDE icon starts the program, and after clicking on *File* in the left corner you will see the following menu.



Clicking on *New Script* makes a window appear where you browse for the path (*flexpde_exa*) and then name your first problem descriptor file, e.g. *exa021*. The extension (family name) is understood to be *.pde*.

Naming gives rise to a new window, as illustrated below. The one to the right is the *editor*, where you type the script using the headings (red) already present as a guide.



The lower left area is the *status* window, recording the operations performed when an application is executed.

Plotting Functions of One Variable

We shall now create a file that defines your first task: just plotting a function of one variable and its derivatives. We use the *editor* to enter the following *descriptor* (*script*) file.

The upper-case words are reserved for the program and will be used again and again in all descriptors. They mark the beginning of a segment where you enter instructions and data pertinent to the problem at hand. In this book, keywords will be printed in **bold** when used for the first time, or else discussed in the text.

The editor automatically adds colors to different kinds of entries in order to enhance readability.

In this book we use the *Student* Version 5, with few exceptions.

```
TITLE                                                    { exa021.pde }
  'sin(x)+ x* cos(x)'
COORDINATES                 { Student Version 5 }
  cartesian1                { This is a one-dimensional application }
DEFINITIONS
  Lx=10
  f=sin(x)+ x* cos(x)       { Function to be plotted }
  fx=dx(f)                  { First derivative }
  fxx=dxx(f)                { Second derivative }
BOUNDARIES
region 'domain'             { Region for plot }
  start(-Lx) line to (Lx)
PLOTS
  elevation(f, fx, fxx) from (-Lx) to (Lx)
END
```

Notice that a string of characters, such as the *title*, must be delimited by quotes. The curly brackets { } enclose *comments* which helps us to remember the purpose of various lines, and these portions are not processed by the program. For instance, it is wise to include the intended name of the descriptor file on the first line. The names in this book use the chapter number for the first digits.

The mathematical notation used under *definitions* is much like that of most programming languages. The sign * means "multiply" and ^ means "raise to a power". FlexPDE is case-insensitive, which means that f and F are treated as *identical* variables.

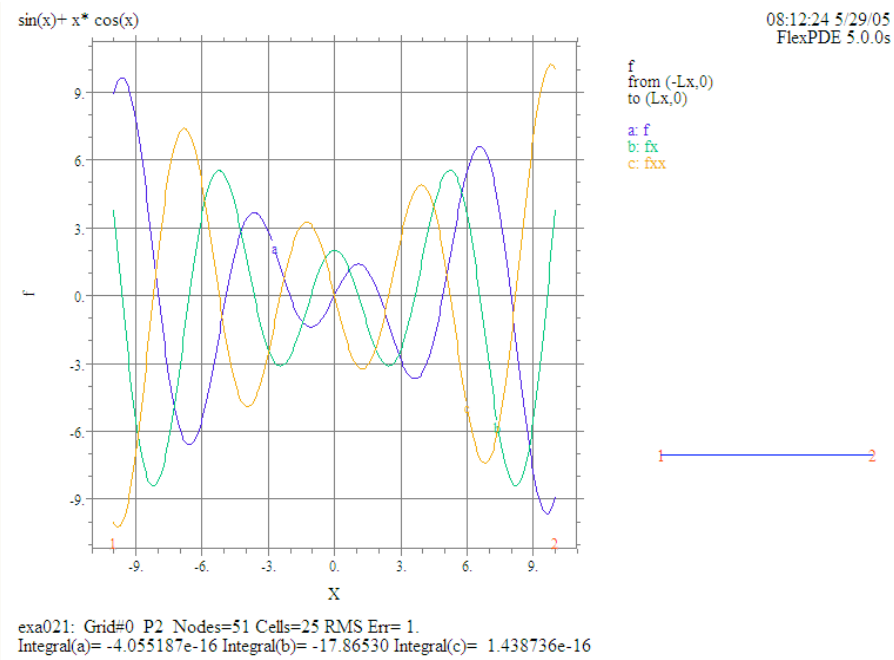
Standard functions are also available, such as sin(x) above. In addition, *partial derivatives* are allowed operators. The derivative $\partial f/\partial x$ is simply denoted dx(f), and so on. A higher derivative may be written dx(dx(f)), or simply dxx(f).

The last section specifies an *elevation* plot, which effectively is a set of curves. You may plot several functions in the same figure, but that is useful only if the function values are of comparable magnitude.

In order to make the script file more readable it is helpful to indent lines containing your statements. You may design the page to suit your personal taste, and the above style is only a suggestion.

Running the Problem Descriptor File

A click on the yellow lightning symbol at the top of the editor window starts execution of the program and immediately produces the figure below.



In this plot, the three curves *a*, *b* and *c* may be identified by means of their colors and the table on the right. The curves permit you, for instance, to relate zero crossings to minimum and maximum points.

After this you may want to explore other functions $f(x)$ in a similar manner. If an expression is more complicated than in the above model descriptor, divide it into groups of terms and let it run over two lines if necessary. The program interprets the contents of each line in an intelligent manner, and no continuation signs are needed. If this would make an expression more clear, you may separate terms by extra spaces.

Plotting Functions of Two Variables

Next we shall study a function of two variables, which is what this software is primarily designed to handle. When typing the new descriptor file you have two options. Either you click on *File*, *New Script* and type all the lines required, or you *Open* a previous file (*exa021*), click on *SaveAs* to make a copy under a new name, and then modify lines as needed. If the new file should be rather similar to the previous one, the latter alternative is of course preferable.

As shown in the *definitions* segment, a line may contain two or more assignments, without any marks (except spaces) to separate the statements.

Since the present function depends on two variables, the program requires that we reserve a *region* in (x, y) space, here of size 1 by 1. In the *boundaries* segment we define the shape and size of this region by a set of (x, y) coordinates, connected by straight lines. We always draw the boundary of a region in the positive sense (*counter-clockwise*), so that the interior of the region remains on our left as we go around.

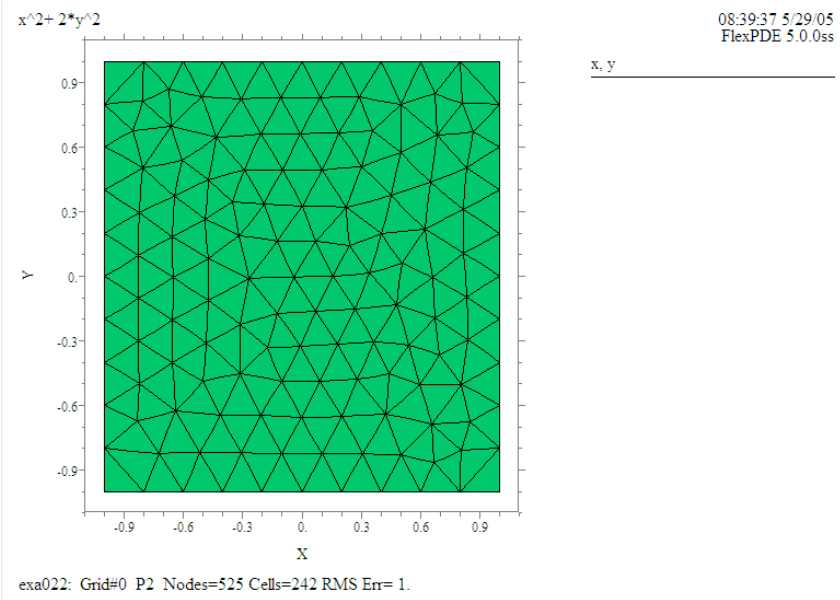
To obtain a *region* we must close the boundary by going back to the starting point. The command close achieves this.

An *elevation* plot of a function $f(x, y)$ always refers to a line in the (x, y) plane, specified by the from...to statement. This type of plot displays the height of the function surface above the given line.

```
TITLE                                                    { exa022.pde }
  'x^2+ 2*y^2'
SELECT
  spectral_colors                          { Values from red to violet }
DEFINITIONS
  Lx=1     Ly=1     f=x^2+ 2*y^2
  grad_f=vector( dx(f), dy(f))     laplace_f=dxx(f)+ dyy(f)
BOUNDARIES
region 'domain'                            { Closed by return to start }
  start (-Lx,-Ly) line to (Lx,-Ly) to (Lx,Ly)  to (-Lx,Ly) close
PLOTS
  grid( x, y)                              { Triangular mesh }
  surface( f)                              { Surface in 3 dimensions }
  elevation( f) from (-Lx,-Ly) to (Lx,Ly)  { Height of f(x,y) above line }
  contour( f)                              { Contour plot of function }
  contour( f) painted                      { Color coded in plane }
  vector( grad_f) as 'Gradient'            { Arrow plot with a title }
  contour( laplace_f)                      { Test if f(x) is harmonic }
END
```

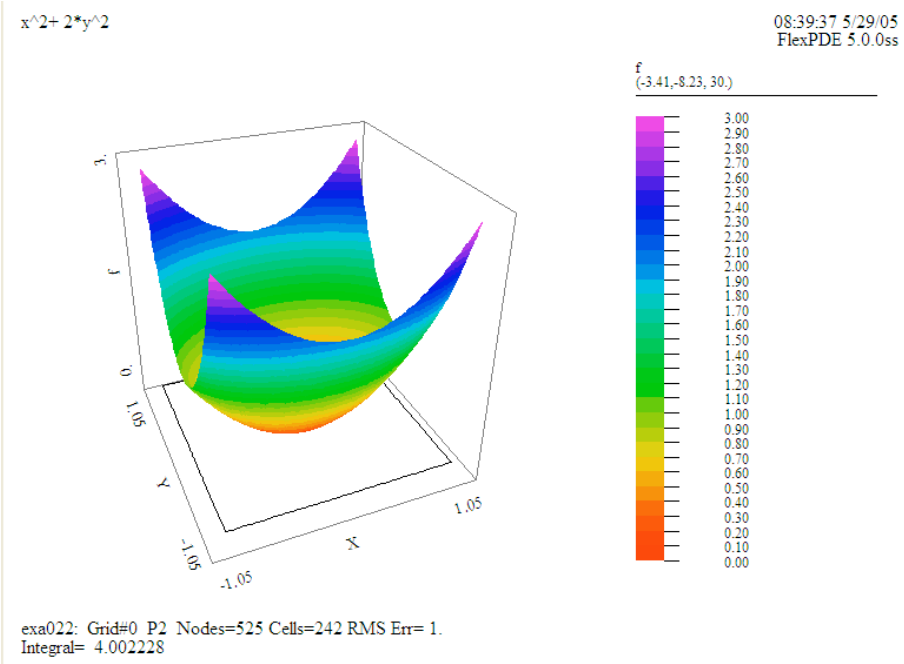
When we run this file, the program presents all the plots as miniatures on the screen. A double-click anywhere inside a selected figure will enlarge it. Another double-click restores the mosaic of miniatures.

A figure may be copied on paper by right-clicking and selecting *Print.* Alternatively, you can use the *PrtScr* (print screen) key on your PC and paste the plot to a word processor.



In the final section of the script we begin by plotting the *grid* (above), which shows how the domain is divided into cells. FlexPDE computes one function value for each of the corners of a triangular cell, and one value at each midpoint of a side. These are the reference data, and in order to obtain values at intermediate points, and to form derivatives, the program interpolates by fitting polynomials $P(x,y)$ to the known values. Normally, we do not need to request a grid plot, because such a plot automatically appears below the status window.

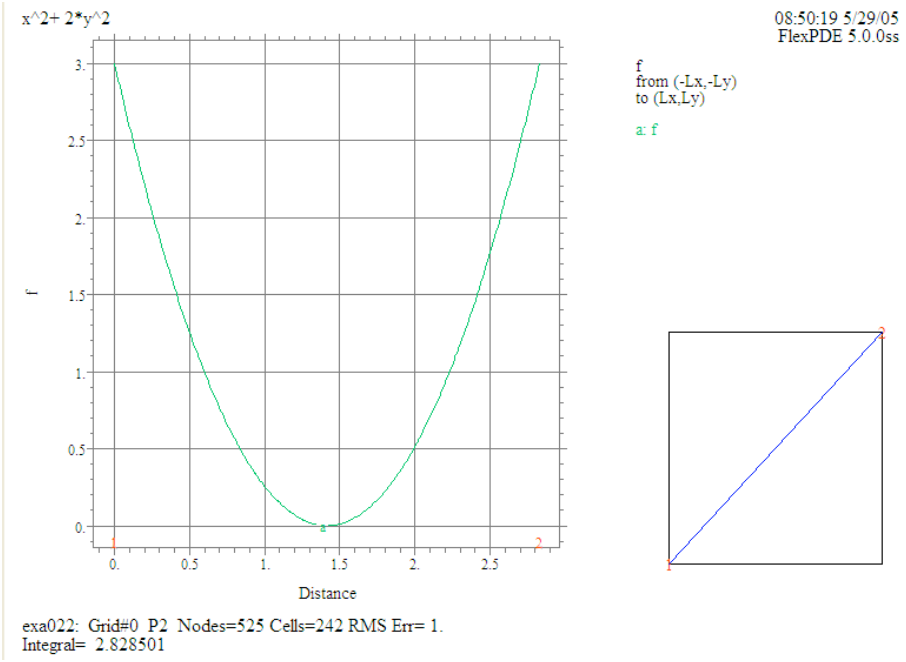
The second figure (below) produced by this file is a plot of a *surface* that presents function values as heights over the (x,y) plane. This figure also presents $f(x,y)$ by the colors of the rainbow in the order of the frequency (or energy) of light, i.e. the lowest values by red and the highest ones by violet. This also associates with temperature, in the sense that a metal first becomes red then bluish-white when heated. Under *select* we specify *spectral_colors*, since the reverse order is the default.



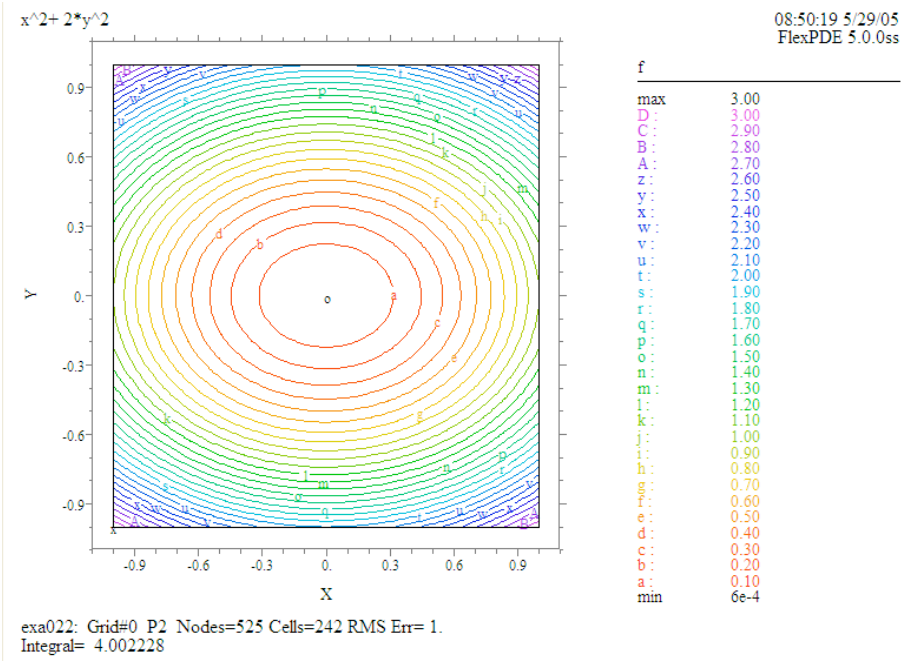
All points on the surface having values between 2.00 and 2.10, say, are in the same color. This means that the borderline between two colors corresponds to a value of $f(x,y)$ that may be read off the table to the right of the plot.

The above surface is viewed at a certain angle with respect to the coordinate axes, but you can move the "camera" by right-clicking on the plot, and then click on *Rotate*. The three values given just above the list to the right indicate the current position of the viewpoint, i.e. the camera coordinates (x, y) and the elevation angle in degrees.

The next figure (below) is an *elevation* plot on a diagonal line in the (x, y) plane.

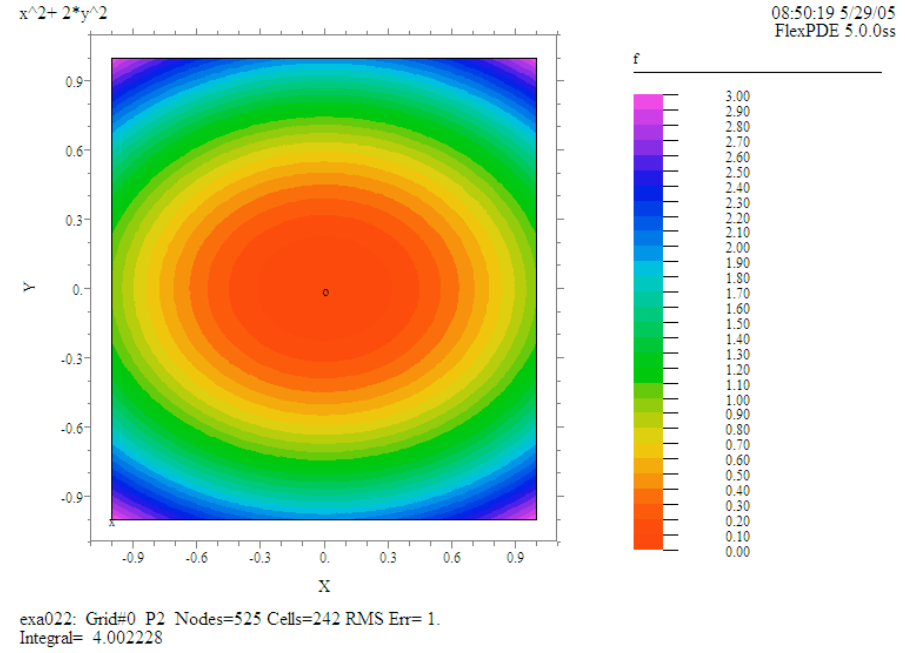


The fourth figure is a *contour* plot (below) consisting of a set of curves in the (x, y) plane, each curve corresponding to a constant value of f. These function values may be read off the adjoining table, which also indicates the minimum and maximum values over the domain. This is similar to a surface plot viewed from above, the borderlines between the different colors forming the contours.



The *painted* contour plot below is simply a plot with colored spaces between the contours.

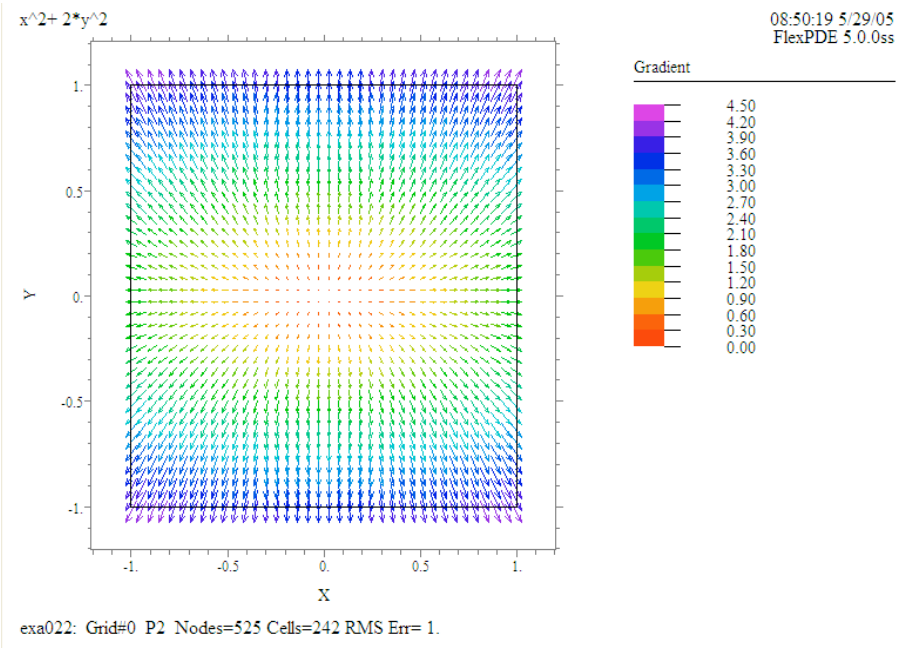
Alternatively, this plot may be regarded as a picture of a surface plot, taken straight from above. The color code yields a quick overview of the surface representing the function.



A new feature under *definitions* is the *vector* grad_f, formally defined by

$$\text{grad}(f) \equiv \nabla f \equiv \mathbf{i}\frac{\partial f}{\partial x} + \mathbf{j}\frac{\partial f}{\partial y} \qquad \bullet$$

where **i** and **j** are vectors of unit length along the *x* and *y* axes. We express this vector simply in terms of its components dx(f) and dy(f). This permits us to represent the *gradient* field by arrows.



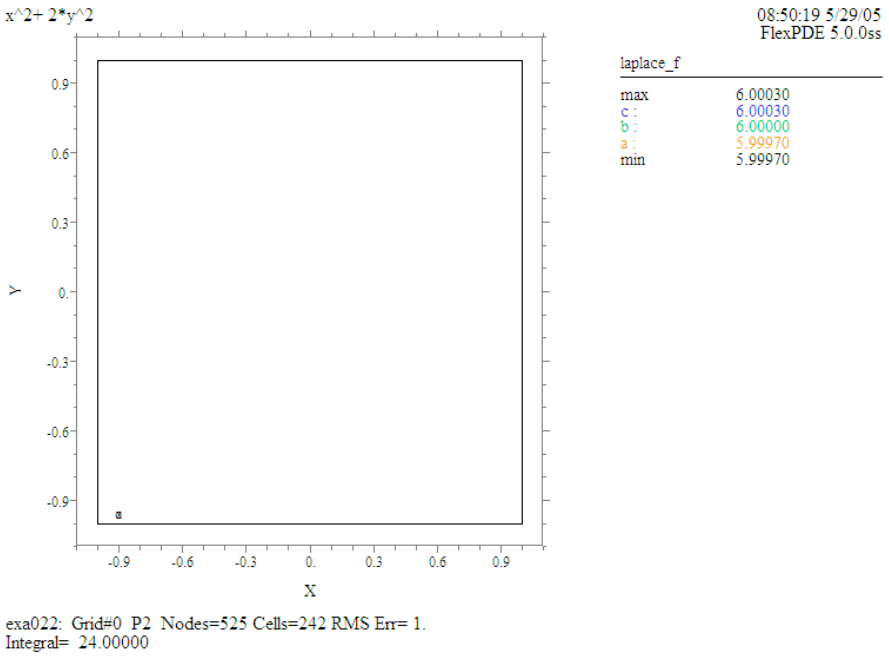
Each arrow shows the direction of maximum slope and (by its length) the corresponding magnitude, also indicated by colors as shown below.

In this particular case all arrows point away from the center, which shows that a minimum is located at the origin (0,0).

The last figure on the screen shows the result of applying the ∇-operator twice to obtain laplace_f, as follows.

$$\nabla^2 f = \left(\mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y}\right)\cdot\left(\mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y}\right) f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad \bullet$$

The corresponding *contour* plot (below) exhibits no curves at all, and the maximum and minimum values are both closely equal to 6.0, differing only in the 4th decimal. This means that the function laplace_f is constant. The fact that $\nabla^2(f)$ is different from zero also indicates that $f(x)$ is not a harmonic function[1p477].



We notice that there are now two tabs on top of the editor, corresponding to the scripts. Clicking on tabs, it becomes easy to return to earlier scripts and plots from the same session.

Help and Manual

FlexPDE offers a *Help* button, which provides information about the various commands, including the syntax of their use. Refer to this source whenever you encounter a command that does not seem familiar. The program also comes with a printed manual.

Exercises

Explore the following functions, perhaps with modified values of Lx and Ly.

- ❑ $f = x^2 y$
- ❑ $f = x^2 y^2$
- ❑ $f = \sin(5x^2 + 5y^2)$

3 “Curly” Velocity Fields

In the preceding chapter we demonstrated various aspects of more or less arbitrary functions. We are now about to study more realistic fields, similar to those occurring in the mechanics of fluids, where a velocity vector may be defined at every point of the field.

We shall consider a few simple cases where a fluid rotates around an axis. This means that the velocity **v** at every point is perpendicular to the radius vector and that its magnitude is $|\mathbf{v}| = \omega r$. These facts may be summarized by a determinant expression, yielding the velocity vector:

$$\mathbf{v} = \mathbf{w} \times \mathbf{r} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \omega_x & \omega_y & \omega_z \\ x & y & z \end{vmatrix} = \begin{Bmatrix} \omega_y z - \omega_z y \\ \omega_z x - \omega_x z \\ \omega_x y - \omega_y x \end{Bmatrix} \quad \bullet$$

We restrict our study to the (x, y) plane, i.e. v_z in the third row must be zero. This generally requires $\omega_x = \omega_y = 0$, leaving us with only the *z*-component, which we shall simply call ω.

Liquid, Rotating as a Disk

In this case, ω will be the same over the entire domain. The following descriptor is what we need to explore the various features of this motion. In order to define a circular domain we need the new command *arc*, which permits us to draw a sequence of quarter-circles. In the *definitions* segment we have included the special differentials div(**v**) and $(\nabla \times \mathbf{v})_z$, the latter being the *z*-component of curl(**v**). The definitions of these operations in (x, y) coordinates are

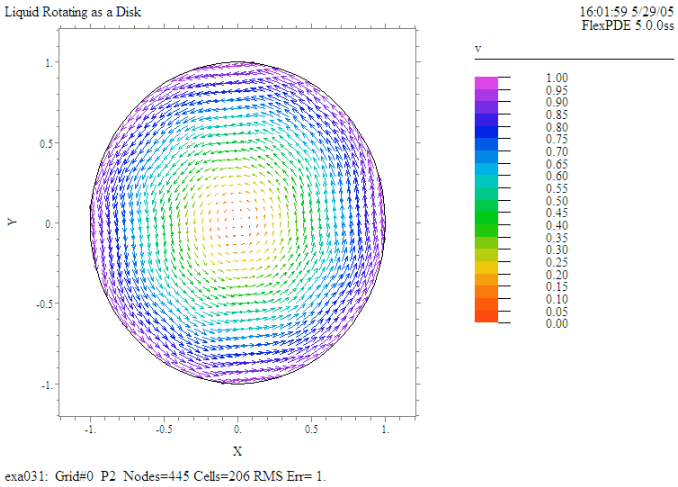
$$\nabla \cdot \mathbf{v} \equiv \text{div}(\mathbf{v}) = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \quad \bullet$$

and

$$\nabla \times \mathbf{v} \equiv \text{curl}(\mathbf{v}) = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ v_x & v_y & v_z \end{vmatrix} = \begin{Bmatrix} \frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \\ \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \\ \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \end{Bmatrix}$$

where we now use only the third (z) component.

```
TITLE                                                    { exa031.pde }
  'Liquid Rotating as a Disk'
SELECT                                                   { Student Version }
  spectral_colors
DEFINITIONS                                              { SI units }
  r1=1.0     rad=sqrt(x^2+y^2)                           { Radius=square root }
  omega=1.0                                              { Angular velocity }
  vx=-omega*y     vy=omega*x                             { Velocity components}
  v=vector( vx, vy)                                      { Velocity vector }
  vm=sqrt(vx^2+vy^2)                                     { Magnitude of v }
  div_v=dx(vx)+ dy(vy)     curl_z=dx(vy)- dy(vx)
BOUNDARIES
region 'domain'  start(r1,0)
  arc to (0,r1) to (-r1,0) to (0,-r1) close              { Circular arc }
PLOTS
  contour( vx)     contour( vy)     contour( vm)
  contour( div_v)     contour( curl_z)     vector( v)
END
```



The above figure shows the vector plot of the velocity. The divergence div_v turns out to be constant, with the value zero. The contour plot of curl_z is also trivial, since the function has the almost constant value 2.000. Of course, we could easily calculate these functions exactly by hand in this simple case, but we shall find similar plots convenient with other, more complicated velocity fields.

Non-Constant ω

We next explore a situation where the angular velocity ω varies with the radius R. Such a variation could exist in a liquid, but we need not discuss in detail what would be required to make it move the way we specify. In the descriptor below, modeled after *exa031*, we choose the function $\omega = 1/R^2$.

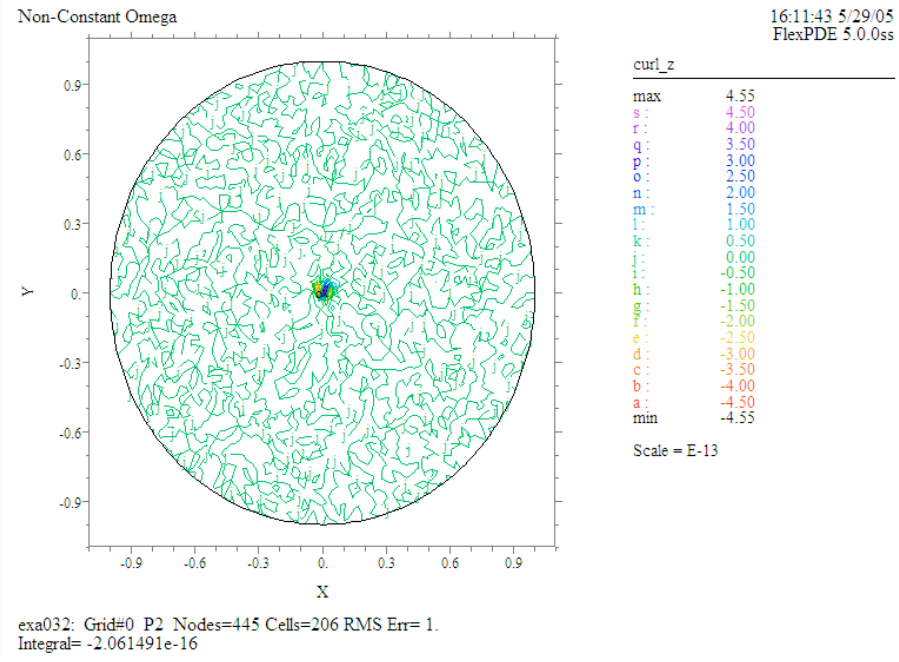
```
TITLE                                                    { exa032.pde }
  'Non-Constant Omega'
SELECT
  spectral_colors
DEFINITIONS                                              { SI units }
  r1=1.0
  rad=sqrt(x^2+y^2)    omega=1/rad^2                     { rad=R }
  vx=-omega*y    vy=omega*x                              { Velocity }
  v=vector( vx, vy)    vm=sqrt( vx^2+vy^2)               { Magnitude }
  div_v=dx(vx)+ dy(vy)    curl_z=dx(vy)- dy(vx)
BOUNDARIES
region 'domain'
  start(r1,0) arc to (0,r1) to (-r1,0) to (0,-r1) close
PLOTS
  contour(vx)    contour(vy)
  contour( abs(vx)) log                          { Log10 of absolute value }
  contour( abs(vy)) log
  contour( div_v)    contour( curl_z)
  contour( vm) log    vector( v/vm)                        { Unit magnitude }
END
```

Evidently, the contour plots of vx and vy are trivial, because most of the variation occurs close to the origin.

In order to follow the variation over a large range of function values, we use the modifier log to request a contour plot with a

10-logarithmic scale. The logarithmic plots of the magnitudes of vx and vy now appear to be similar, except for a rotation through 90^0.

The plot of curl_z may seem peculiar. It actually consists of a number of irregularly shaped zero contours. Other contours listed in the table correspond to magnitudes below 10^{-12}, whereas we obtained 2.0 in *exa031*. In view of the round-off errors always present in numerical calculations we take this to mean that the value of curl_z is zero over the whole region, which we can readily verify by exact calculus. The divergence div_v also seems to be zero.



By a contour plot of vm and a normalized vector plot (**v**/vm) we present both the speed and the direction at various points. Thus these two graphs together completely characterize the vector field.

We now proceed to a similar case with $\omega = \sin(R)$, which only requires the following changes of the descriptor *exa032*.

```
TITLE                                                { exa032a.pde }
  'Non-Constant Omega'
…
  rad=sqrt( x^2+y^2)     omega=sin( rad)
...
```

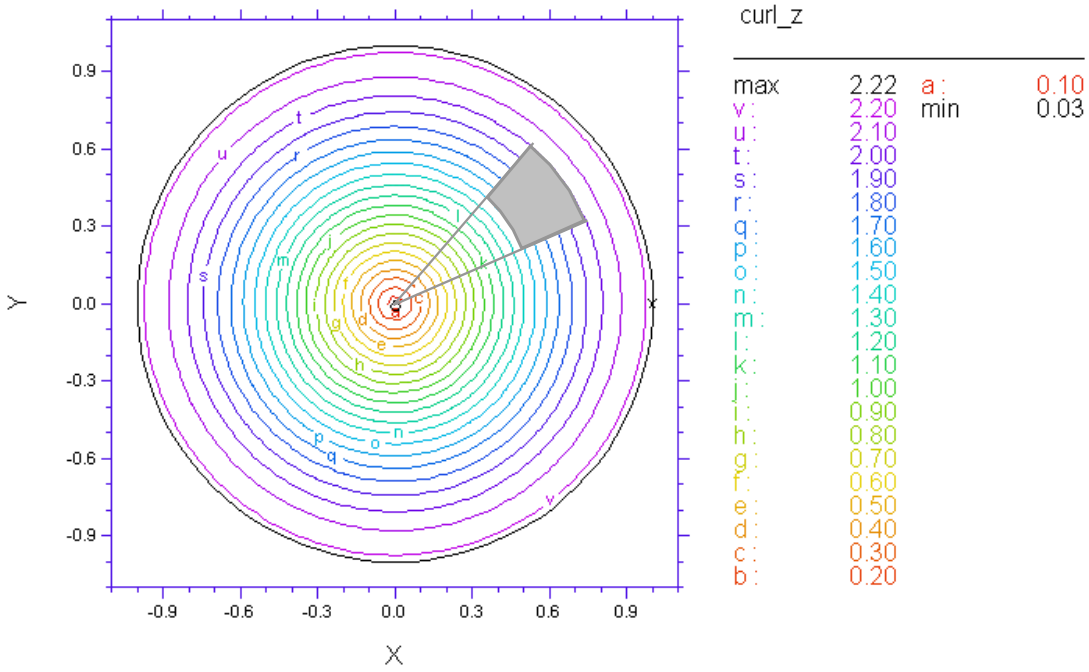
Running this file, we find that curl_z becomes clearly different from zero, although the liquid appears to circulate much the same

way as it did before. The value of the divergence div_v, however, still vanishes.

What kinds of rotary motion yield non-zero values of curl_z? In textbooks on vector analysis we find the definition

$$\mathrm{curl}(\mathbf{v})_z \equiv (\nabla \times \mathbf{v})_z \equiv \lim_{S\to 0} \frac{1}{S} \oint_C \mathbf{v} \cdot d\mathbf{l} \qquad \bullet$$

where the line integral is taken over a closed curve C of area S in the (x, y) plane. The situation is depicted in the next figure, where we have drawn an area element (gray) that will eventually shrink to negligible size. This element is limited by two radial and two circular segments. The velocity **v** is directed along the circular paths; hence the radial parts contribute nothing to the integral. The contribution from a circular segment is just the speed vm multiplied by the length of the arc, and since these two terms are of opposite sign they might accidentally cancel.



In *exa031* the outer path was longer and, in addition, the speed was larger there. Each of these enhancing factors was proportional to the radius and the result was a nonzero value of curl_z.

In the case of the liquid with $\omega = 1/R^2$, the smaller velocity on the outer path compensates exactly for the increase of the path length,

which makes curl_z equal to zero. Generally, one might say that a vanishing *curl* is a rare accident.

In the following file, which is based on *exa031*, we first make an elevation plot on the curve 'inner', which is a circle and may be eccentric (due to r0). We define it by a *feature*, a curve that is closed but does not form a *region*. Here, we use the convenient command *angle* to draw the second circle.

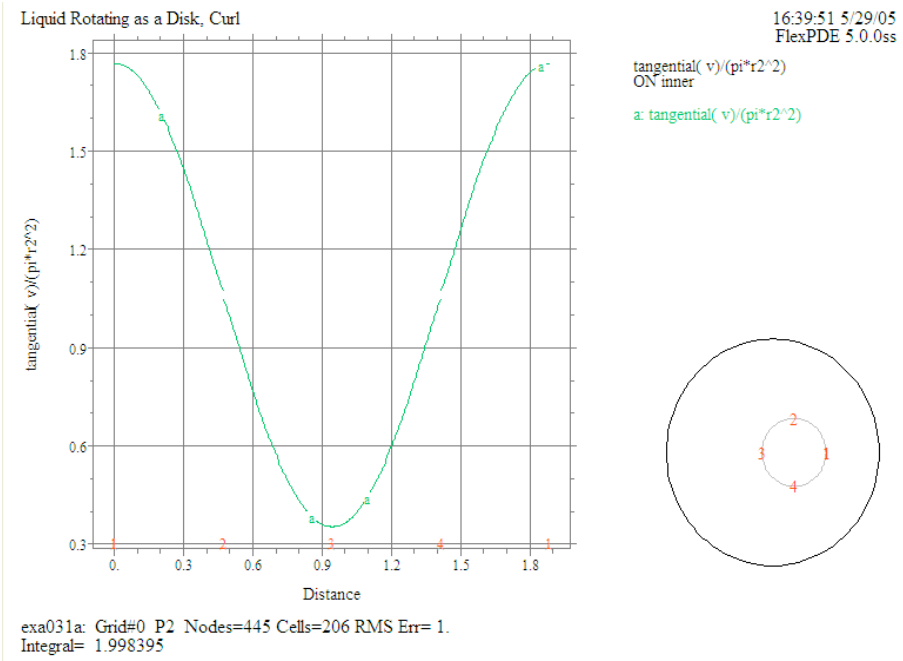
```
TITLE                                                    { exa031a.pde }
  'Liquid Rotating as a Disk, Curl'
SELECT
  spectral_colors
DEFINITIONS
  r1=1.0     r2=0.3     r0=0.2
…
region 'domain'
  start(r1,0)  arc to (0,r1) to (-r1,0) to (0,-r1) close
feature
  start 'inner' (r0+r2, 0) arc( center=r0,0) angle=360
PLOTS
  elevation( tangential( v)/(pi*r2^2) ) on 'inner'        { ⇨ Curl_z }
  contour( vm/(pi*r1^2))                                  { ⇨ Average }
END
```

The software enables us to estimate curl_z by means of a line integral. We need to calculate the integral of

$$\mathbf{v} \cdot d\mathbf{l} \equiv v_t dl$$

along a closed curve, v_t being the component of **v** tangential to that curve. The command *tangential* selects the vector component v_t. The program automatically calculates the integral of the function plotted, and we divide that by the area *S* of the smaller circle. The result is displayed on the bottom line of the following plot.

The line integral for $(\nabla \times \mathbf{v})_z$ should really be taken to the limit of vanishing r2, but in this simple case that would not be necessary. If we were to decrease the radius of the curve by a factor of 10, say, there would be too few triangular cells to guarantee reasonable accuracy.



In a contour plot of vm we automatically obtain the area integral over (x, y) space. Dividing by the area of the domain we generate the *average* value.

In the past examples we noticed that the *divergence* of the velocity field was zero. In order to understand why this happens we consider the integral definition

$$\mathrm{div}(\mathbf{v}) \equiv \nabla \cdot \mathbf{v} \equiv \lim_{V \to 0} \frac{1}{V} \oiint_S \mathbf{v} \cdot \mathbf{n}\, da, \qquad \bullet$$

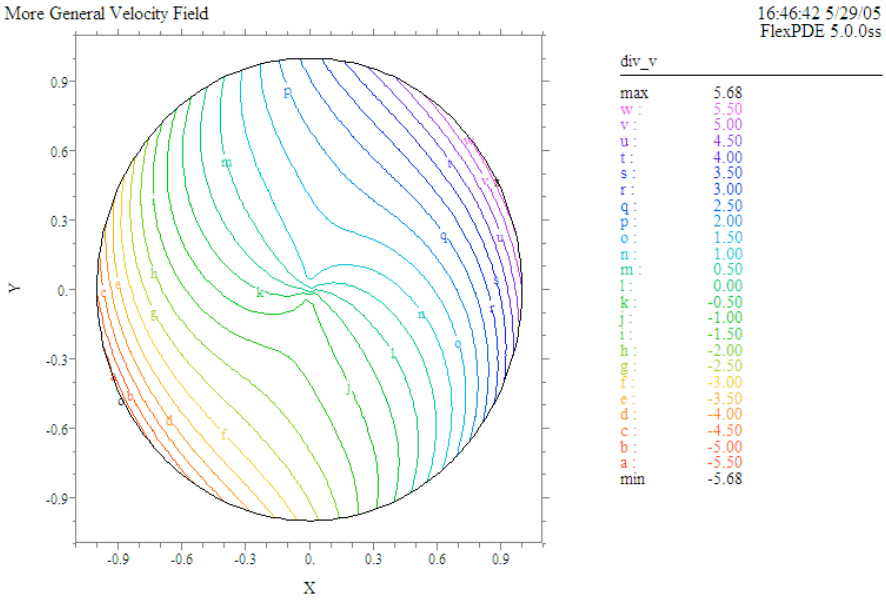
where the integral is now taken over the surface S of a volume element V as shown in an earlier figure (p.20). The cylindrical surfaces of the box give no contributions to the integral, since **v** is perpendicular to the normal **n** on these surfaces. Two of the limiting planes are parallel to the (x, y) plane. The integral over these planes also adds nothing, because the velocity is always confined in the (x, y) plane. As regards the radial planes, they give equal contributions, but of opposite signs, since the velocity depends on the radius R in the same way on both of them. Thus, whenever ω depends on R only, the divergence must be zero.

More General Velocity Fields

In previous examples we were concerned with rotation about an axis, and the angular velocity depended only on the radius. That was clearly a special case of fluid motion. We shall now look at a more general field, where the velocity components are independent functions of *R*. Let us introduce the following (arbitrary) definitions into a copy of *exa032*, produced by *SaveAs*.

```
TITLE                                                    { exa033.pde }
  'More General Velocity Field'
SELECT
  spectral_colors
DEFINITIONS
  r1=1.0   rad=sqrt(x^2+y^2)
  vx=rad*tan( rad)   vy=exp( rad)
…
```

The resulting contour plot of the divergence shows non-zero values, as seen in the figure below.



The contour plot of curl_z also exhibits non-zero values. The *vector* plot of the velocity, however, reveals no signs of any circulating motion that could lead us to suspect non-zero curl_z.

The lesson to be drawn from the examples of this chapter is that the appearance of a velocity field, as given by a vector plot, is not sufficient to assess whether *curl* or *div* is zero or not. We always need to apply the differential or integral definitions to make this distinction.

Exercises

- ❑ Try $\omega = 1/R$ in one of the above descriptors.
- ❑ Try $\omega = R^2$ and estimate the curl by a line integral. Then do it again with a different center for the smaller circle.
- ❑ Plot the vector fields curl(grad(x^2+y^2)) and curl(grad(x^2+y^3)).
- ❑ Invent your own functions of (x, y) for vx and vy and calculate the curl and the divergence.

4 Fields of Gravitation

Massive objects create a force field, acting on any other mass. This force may be expressed in terms of a potential function U. For a point mass at (x_0, y_0), the potential may be written

$$U = -G\frac{m}{R} = -G\frac{m}{\sqrt{(x-x_0)^2 + (y-y_0)^2}}$$

where G is the gravitational constant and R the radial distance from the point mass to the field point.

Earth and Moon as Point Masses

Let us choose the Earth-Moon system as a concrete example, using the following descriptor. Throughout this volume we shall use *SI units* (kg, m, s, etc.) for all physical quantities.

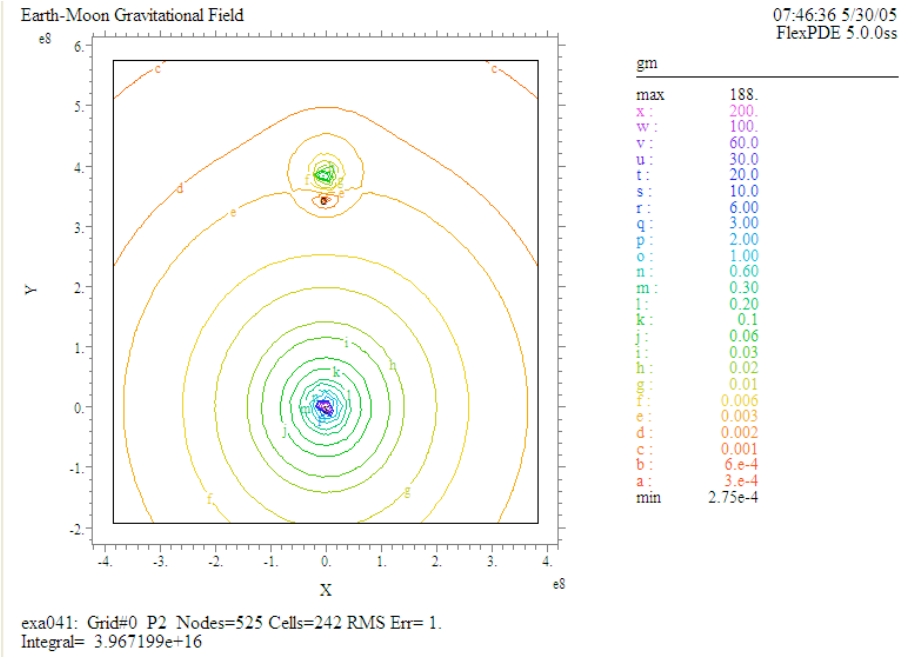
```
TITLE                                                    { exa041.pde }
  'Earth-Moon Gravitational Field'
SELECT                                  { Student Version }
  spectral_colors                       { From red to violet }
DEFINITIONS                             { SI units }
  d=3.84e8    Lx=d    Ly=d
  m1=5.98e24    m2=7.35e22              { Masses of Earth and Moon }
  G=6.67e-11                            { Graviational constant }
  U=-G*m1/sqrt(x^2+y^2)- G*m2/sqrt(x^2+(y-d)^2)
  gx=-dx(U)    gy=-dy(U)                { Field components }
  gv=vector( gx, gy)    gm=sqrt( gx^2+ gy^2)  { Vector and magnitude }
BOUNDARIES
region 'domain' start(-Lx,-Ly/2)
  line to (Lx,-Ly/2) to (Lx,3/2*Ly)  to (-Lx,3/2*Ly) close
PLOTS
  contour( U)    contour(-U) log    surface(-U) log
  contour( abs( gx)) log    contour( abs( gy)) log
  contour( gm) log
```

```
  vector( gv/gm) as 'Field directions'
END
```

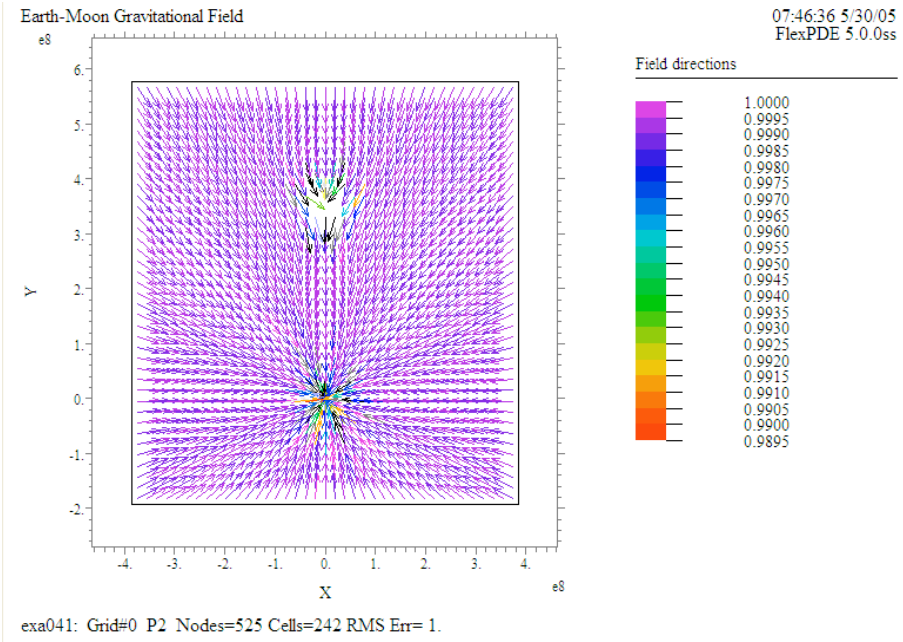
For the potential U we simply write a sum of terms of the form $-Gm_i / r_i$, where G is the gravitational constant, m_i the masses of the objects and r_i the distances to the field point. The relation $\mathbf{g} = -\nabla U$ then yields the gravitational field strength. We place the origin at the center of the Earth, and the Moon at a distance d on the y-axis.

The contour plot of U yields curves that all lie close to the Earth. In order to study the function over the full domain we request a *logarithmic* plot of -U, rather than the direct value.

The logarithmic plot of the magnitude of **g** (below) demonstrates how the Moon perturbs the Earth's gravitational field. In fact, there is a region (red) close to the Moon where the field seems to be zero.



The point where the magnitude of the field vanishes is also evident from the vector plot below, where the field direction at that point seems to be undetermined.



Planets of Finite Size

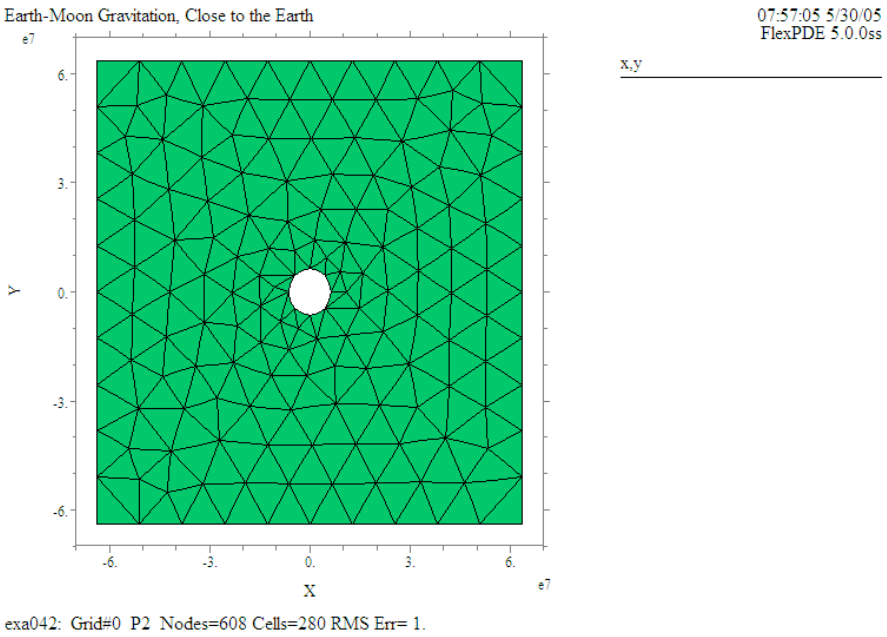
Using the next descriptor we shall look more closely at the field around each planet. In order to indicate these objects we cut out the areas occupied by the Earth and the Moon from the domain. This is also a better presentation in the sense that our potential strictly applies only to the region outside the massive objects.

In order to *exclude* a region occupied by a planet we simply define its contour. For drawing circles we use the command *arc* in the simple way demonstrated on p.21. We just specify the center coordinates and the angle for one full turn. The descriptor has much in common with *exa041*.

```
TITLE                                                    { exa042.pde }
  'Earth-Moon Gravitation, Close to the Earth'
SELECT
  spectral_colors
DEFINITIONS
  d=3.84e8     r1=6.37e6     L=10*r1
  m1=5.98e24     m2=7.35e22              { Masses of Earth and Moon }
  G=6.67e-11                             { Graviational constant }
  U=-G*m1/sqrt(x^2+y^2)- G*m2/sqrt(x^2+(y-d)^2)
  gx=-dx(U)     gy=-dy(U)                { Field components }
  gv=vector( gx, gy)     gm=sqrt( gx^2+ gy^2)   { Vector and magnitude }
```

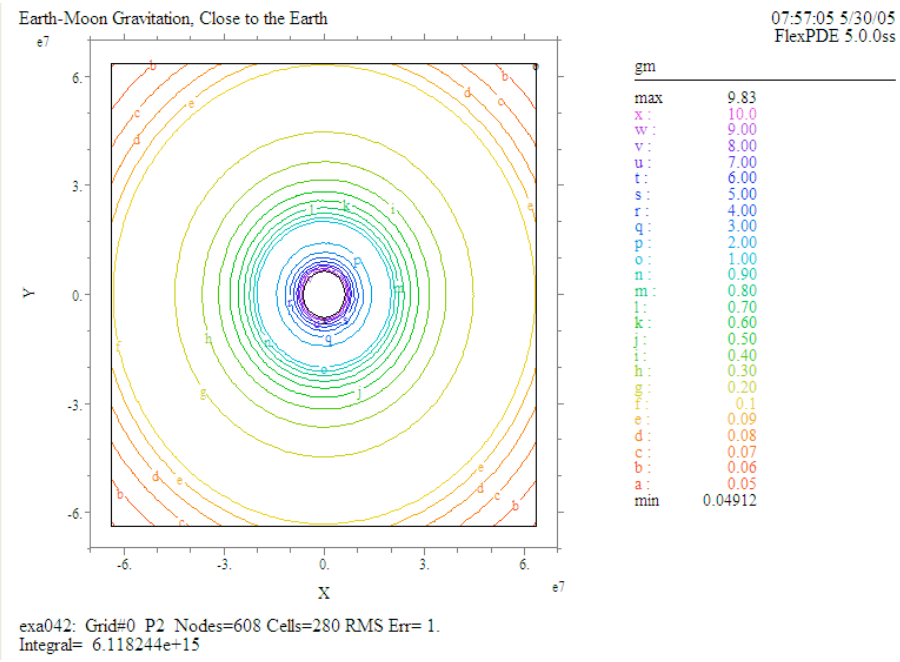
```
  gv_angle=sign(gy)*arccos( gx/gm)/pi*180
BOUNDARIES
region 'domain'
  start(-L,-L) line to (L,-L) to (L,L) to (-L,L) close
  start(r1,0) arc( center=0,0) angle=360                { Exclude Earth }
PLOTS
  grid(x,y)
  contour( gm) as 'Magnitude'     contour( gm) log
  surface( gm) log
  vector( gv/gm)     contour( gv_angle)
END
```

The following grid plot shows that the program automatically divides the area into smaller triangles as you approach an object, in this case a planet. This is useful since we are going to inspect the behavior of the gravitational field at close range.

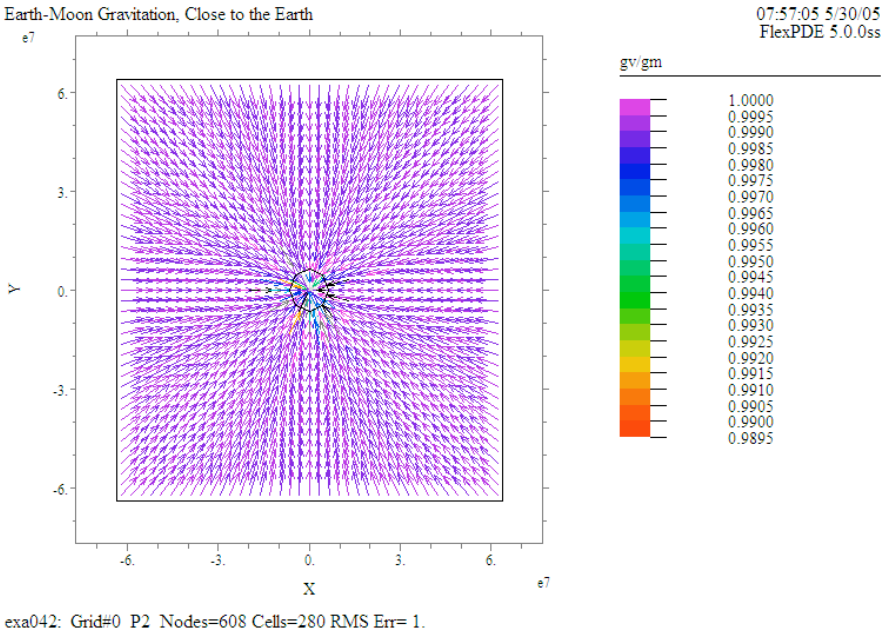


The shrinking of cells size close to an object is the first example of the *adaptive gridding* employed by the program. Inspecting the grid plot we understand why it would not practicable to show both planets in the same figure. Using the Student Version, this example requires 608 nodes, while the maximum allowed is about 800. For this reason, we shall present the field close to the Moon separately.

From the next plot we find that the maximum magnitude of the gravitational field occurs on the surface of the Earth and has the value 9.83.



In the vector plot below, you will notice that some arrows seem to penetrate into the region reserved for the Earth. This occurs because an arrow presents the field existing at its *base*, not at the tip.

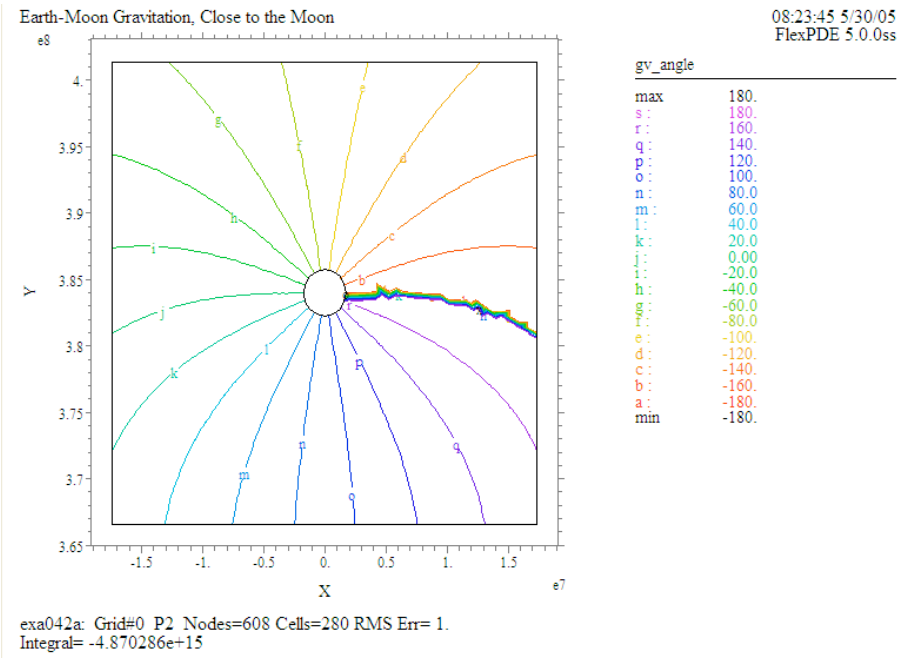


Since a field normally has a direction at each point in space, we may also request a contour plot of the angle of direction. This we do in the last plot (not shown), just inverting the equation $\cos\theta = g_x / g_m$ and then converting from radians to degrees. This leaves an ambiguity concerning the sign of the direction, which we resolve by means of the *sign* function.

The next task will be to plot the field in the neighbourhood of the Moon. To do this we only need to modify the geometry in *exa042* as follows.

```
TITLE                                                    { exa042a.pde }
  'Earth-Moon Gravitation, Close to the Moon'
SELECT
  spectral_colors
DEFINITIONS
  d=3.84e8     r2=1.74e6     L=10*r2
…
region 'domain'
  start(-L,d-L) line to (L,d-L) to (L,d+L) to (-L,d+L) close
  start(r2,d) arc( center=0,d) angle=360                 { Exclude Moon }
PLOTS
…
```

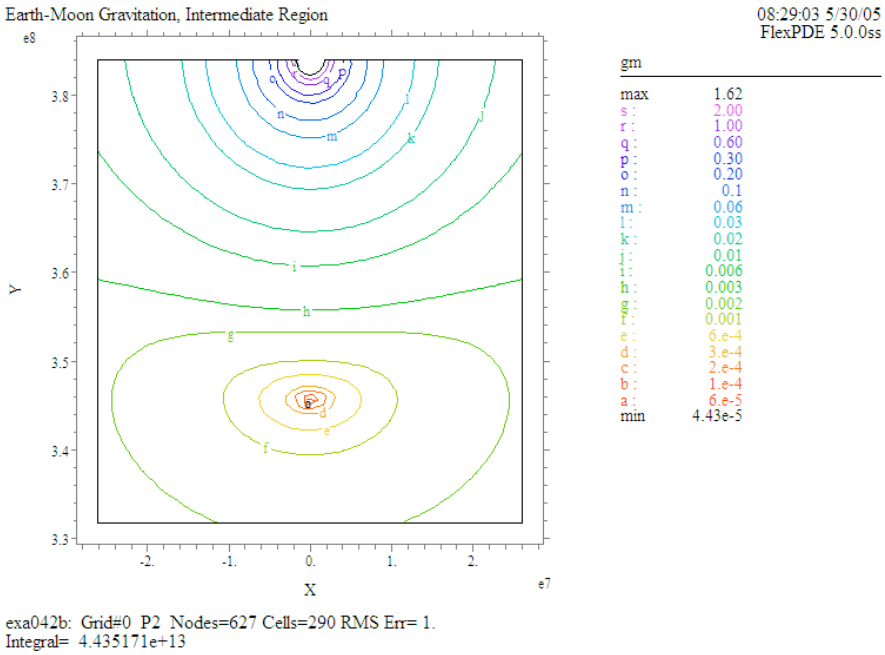
The angle plot below shows that the field now is strongly asymmetric, because of the influence of the Earth's gravitation.



Next, we explore the field in a region between the Earth and the Moon. The following amendments to *exa042* are required.

```
TITLE                                                    { exa042b.pde }
  'Earth-Moon Gravitation, Intermediate Region'
SELECT
  spectral_colors
DEFINITIONS
  d=3.84e8     r2=1.74e6     L=15*r2
…
region 'domain'
  start(-L,d-2*L) line to (L,d-2*L) to (L,d) to (-L,d) close
  start(-r2,d) arc( center=0,d) angle=180 line to (-r2,d)      { Moon }
PLOTS
…
```

The following is a logarithmic contour plot of $|\mathbf{g}|$ over a region that includes the neighborhood of the Moon and extends somewhat in the direction of the Earth. Here, a deep minimum indicates the point where the gravitational field vanishes. The gradual deformation of the circular contours around the Moon, caused by the Earth's attraction, is clearly visible.

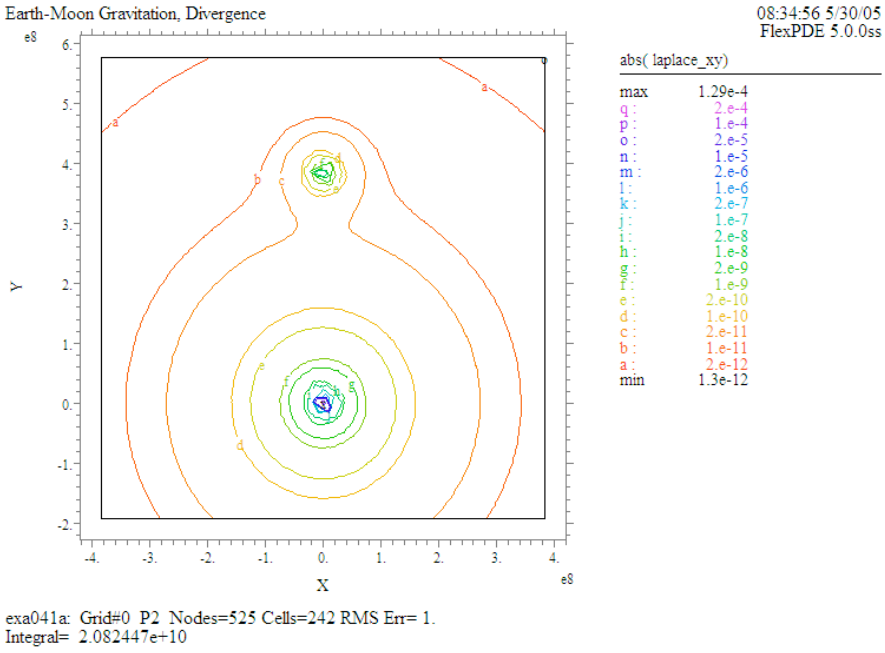


Divergence of ***g***

Before leaving the gravitational field we should investigate whether the divergence, $\nabla \cdot \mathbf{g} \equiv -\nabla^2 U$, is equal to zero. For this purpose we add the following definition before *boundaries* to a copy of *exa041* and make the corresponding contour plot.

```
TITLE                                                    { exa041a.pde }
  'Earth-Moon Gravitation, Divergence'
…
  laplace_xy=dxx(U)+dyy(U)     { Divergence of gv in (x,y) coordinates }
BOUNDARIES
...
PLOTS
  contour( abs( laplace_xy)) log
END
```

To the reader who knows that the divergence of a gravitational field in free space vanishes, the following plot of laplace_xy may come as a shock.



We are in three dimensions, however, although the plots are in a plane through the two planetary objects. If we had included the term dzz(U), which is not allowed in the 2D version of the software, we

would in fact have obtained the irregular contours suggesting zero divergence.

Exercises

- ❑ Modify *exa041* by adding a contour plot of curl(**g**)$_z$.
- ❑ Change *exa042* by making an elevation plot of gm along the line y=0.
- ❑ Convert the contour plots in *exa042b* to *painted*. Also try a logarithmic elevation plot along the line of symmetry.

5 Fields around Electric Charges

Using basic knowledge of the electrostatic potential[3p72] we may easily explore the field **E** around a set of point charges. The starting point is Coulomb's law, which leads to the following expression for the electrostatic potential of a point charge q at (x_0, y_0).

$$U = \frac{q}{4\pi\varepsilon_0 R} = \frac{q}{4\pi\varepsilon_0\sqrt{(x-x_0)^2+(y-y_0)^2}} \quad \bullet$$

Here, ε_0 is the permittivity of free space.

Field around Two Positive Point Charges

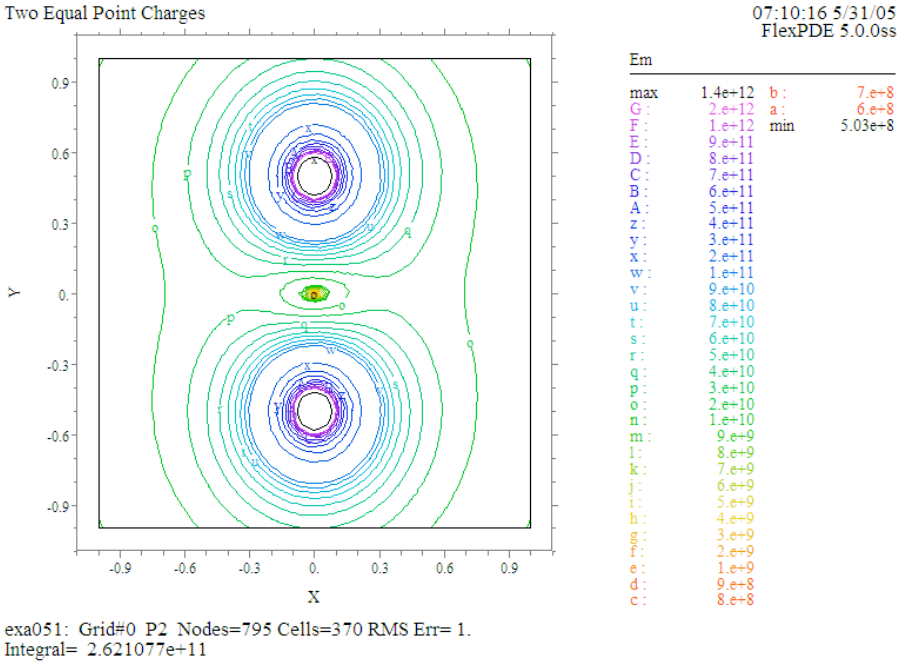
Under *definitions* in the following descriptor, we enter an expression for the potential *U*. It is just the sum of two terms of the above type, one for each of the charges. Here, we introduce short forms for the *gradient* and *magnitude* of a vector, which the program expands exactly as we did explicitly in earlier descriptors.

```
TITLE                                                    { exa051.pde }
  'Two Equal Point Charges'
SELECT                                                   { Student Version}
  spectral_colors                                        { From red to violet }
DEFINITIONS                                              { SI units throughout }
  Lx=1    Ly=1    d0=Ly/2    r0=8e-2    q=1    { Both charges positive }
  eps0=8.85e-12    c=1/(4*pi*eps0)
  U=q*c/sqrt(x^2+(y+d0)^2)+ q*c/sqrt(x^2+(y-d0)^2)
  Ex=-dx(U)    Ey=-dy(U)                                  { Field components }
  E=-grad(U)    Em=magnitude(E)
  E_angle=sign(Ey)*arccos(Ex/Em)/pi*180      { Degrees }
BOUNDARIES
region 'domain'
  start(-Lx,-Ly)  line to (Lx,-Ly) to (Lx,Ly)  to (-Lx,Ly) close
  start(r0,-d0) arc( center=0,-d0) angle=360    { Exclude circle }
```

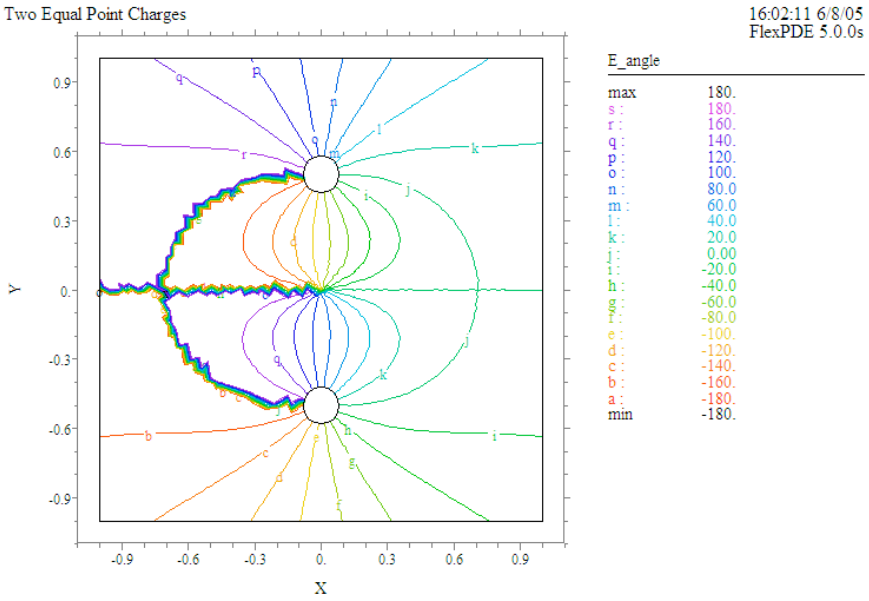
```
  start(r0,d0) arc( center=0,d0) angle=360      { Exclude circle }
PLOTS
  contour( U)   contour( Ex)   contour( Ey)
  contour( abs(Ex)) log   contour( abs(Ey)) log
  contour( Em) log   vector( E/Em)   contour( E_angle)
END
```

Under *boundaries* we exclude small circular regions containing the singularities at the point charges, as we already did with the Earth (p.28). In this way, we avoid very large potential and field values, which would dominate the plots.

We see that the plot of the magnitude Em below is much simpler than the logarithmic plots for the magnitudes of E_x and E_y.



The field directions are illustrated by the vector plot, or alternatively by the contour plot of the *angle*, shown next. In the latter figure, the contours corresponding to 180^0 have degenerated into bands, which may be an unintended feature of the software. Unfortunately some curve identifications are missing in this plot, but the contours may still be assigned to the angles given in the table since we know that the field lines must leave a point charge in a radial direction. This fact and the alphabetic order permit us to assign angles to all the curves.

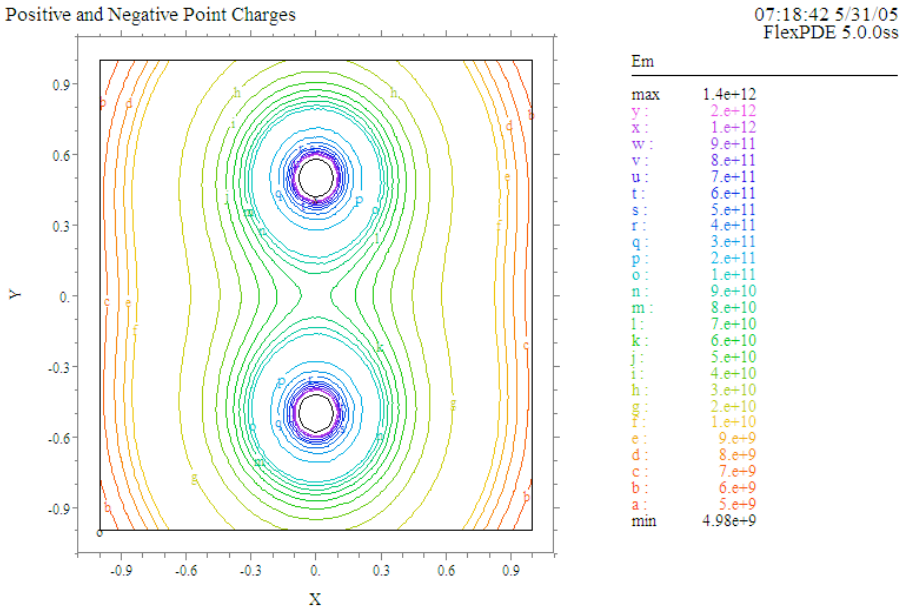


Positive and Negative Point Charges

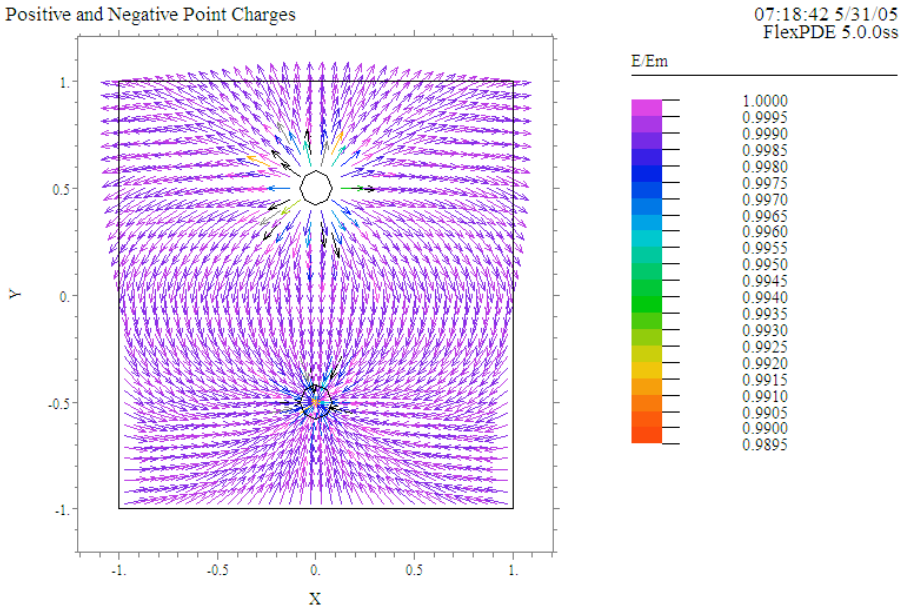
It is interesting to study a slight variation of the preceding descriptor. Let us change the sign of the first charge to negative in the expression for U shown below.

```
TITLE                                                    { exa052.pde }
  'Positive and Negative Point Charges'
SELECT
  spectral_colors
DEFINITIONS
  Lx=1    Ly=1    d0=Ly/2    r0=8e-2    q=1
  eps0=8.85e-12    c=1/(4*pi*eps0)
  U= -q*c/sqrt(x^2+(y+d0)^2)+ q*c/sqrt(x^2+(y-d0)^2)      { Signs: -, + }
…
```

Signs are of course hidden in the plots involving absolute values. The plot of the magnitude $|\mathbf{E}|$, however, does differ from what we saw in the preceding case, especially as regards the field halfway between the charges.



The corresponding vector plot illustrates how the field lines proceed from the upper to the lower charge.



The Dipole Field

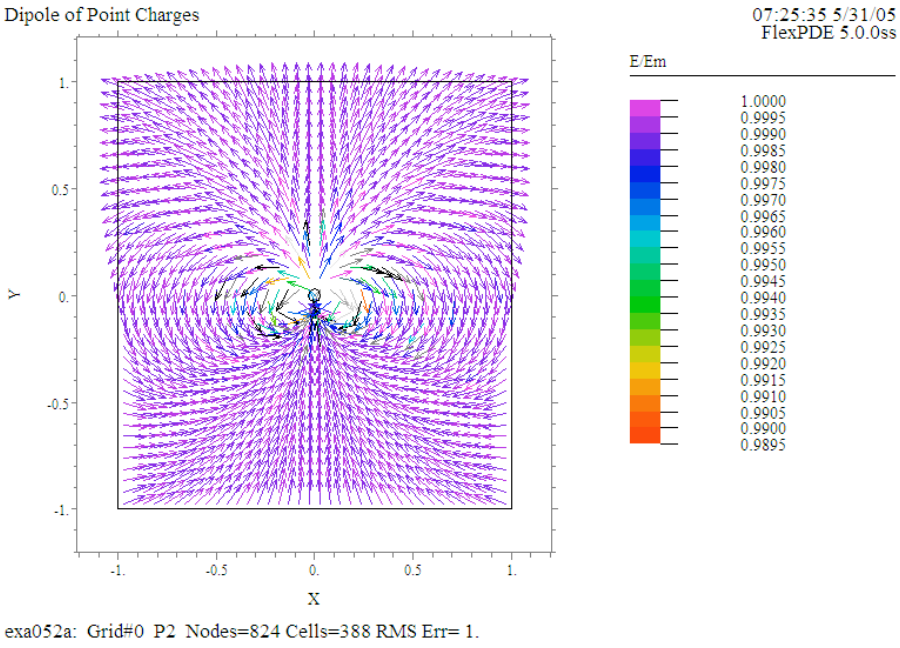
We now proceed to an extreme case of the preceding example: the electric dipole. This object is a combination of a positive and a

negative charge ($\pm q$), at a distance of $2d_0$ from each other. The product $2d_0q$ is known as the dipole moment, and at distances much larger than d_0 the resulting field is similar to that of an infinitesimally small dipole having the same moment. The electric dipoles occurring in the real world are of course of finite extension, but the expression for the field[3p83] becomes simpler in the limit of vanishing d_0.

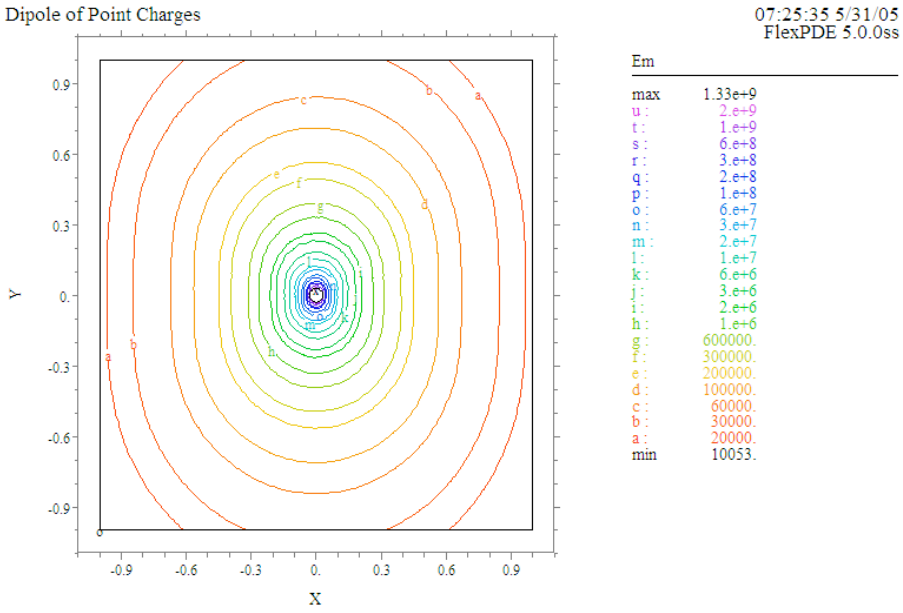
The high precision of FlexPDE permits us to plot the dipole field directly from the point charge potentials we used in *exa052*. We change the name to *exa052a* and reduce the charge distance to a very small value. We also prefer to exclude a *single* region centered on the dipole.

```
TITLE                                                    { exa052a.pde }
  'Dipole of Point Charges'
SELECT
  spectral_colors
DEFINITIONS
  Lx=1    Ly=1    d0=Ly*1e-6    r0=3e-2    q=1
…
  start(r0,0) arc( center=0,0) angle=360          { Replaces two circles }
PLOTS
…
```

The first of the following figures shows the characteristic vector plot of **E** for the dipole field.



The second figure (the contour plot of Em) is remarkably simple. Together with the vector plot it uniquely describes the field. Looking back at the corresponding figure of the previous configuration we may imagine how the contours gradually transform as we go from the near field to the far field.



Field around Charged Wires

The three-dimensional field due to two equal point charges has a counterpart where the field is caused by charged wires perpendicular to the (x, y) plane. In that configuration, the fields in all planes parallel to the base plane are identical, and the field may thus be called two-dimensional (2D).

Let q_w denote the charge per unit length of the wire and R the radial distance. The expression for the electrostatic potential of a wire through the origin is known to be[3p89]

$$U_w = -\frac{q_w \ln(R)}{2\pi\varepsilon_0} = -\frac{q_w \ln(\sqrt{x^2+y^2})}{2\pi\varepsilon_0} = -\frac{q_w \ln(x^2+y^2)}{4\pi\varepsilon_0}$$

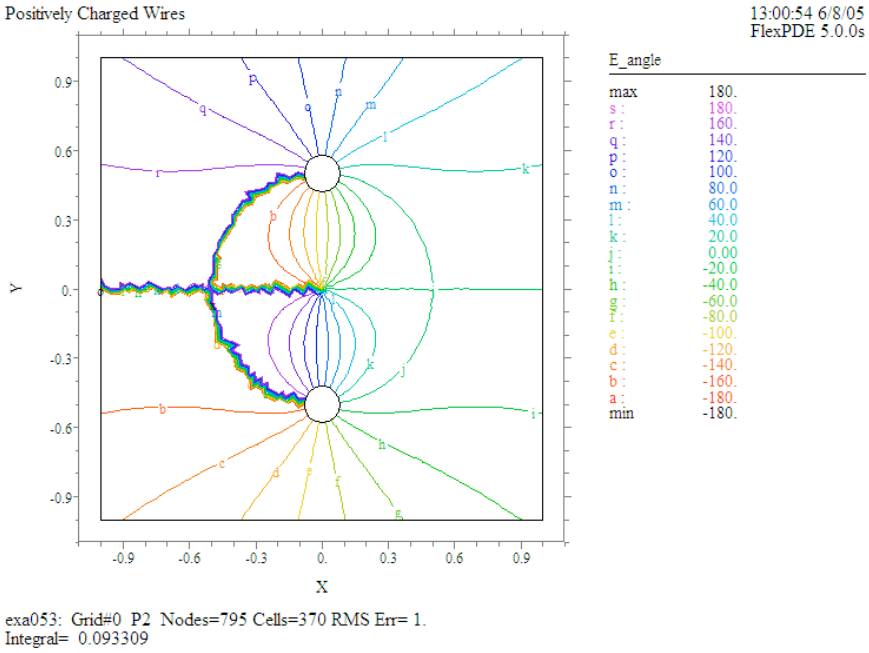
In order to study this new field you only need to change a few lines in the *definitions* segment of *exa051* and add two plots. The most

convenient way is to save that file as *exa053* and modify the latter as follows.

```
TITLE                                                    { exa053.pde }
  'Positively Charged Wires'
SELECT
  spectral_colors
DEFINITIONS
  Lx=1    Ly=1    d0=Ly/2    r0=8e-2    qw=1 { Both charges positive }
  eps0=8.85e-12    c=1/(4*pi*eps0)
  U=-qw*c*ln(x^2+(y+d0)^2)- qw*c*ln(x^2+(y-d0)^2) { Natural logarithm }
…
  contour( del2(U))                                  { Plot of Laplacian }
  contour( dx(Ey)- dy(Ex))                           { Plot of curl_z }
END
```

The 2D field will look much the same as the cross-section of the 3D field through the point charges, but a closer inspection reveals certain differences. Comparing the plots of the angle, for instance, we find that the contour for the angle zero now looks circular, whereas it was definitely oval in the case of point charges (*exa051*).

In this file we use the short form del2(U) for $\nabla^2 U$, which the program automatically transforms into derivatives. The plot of this expression indicates a vanishing Laplacian, and the same is true of the z-component of curl(**E**).

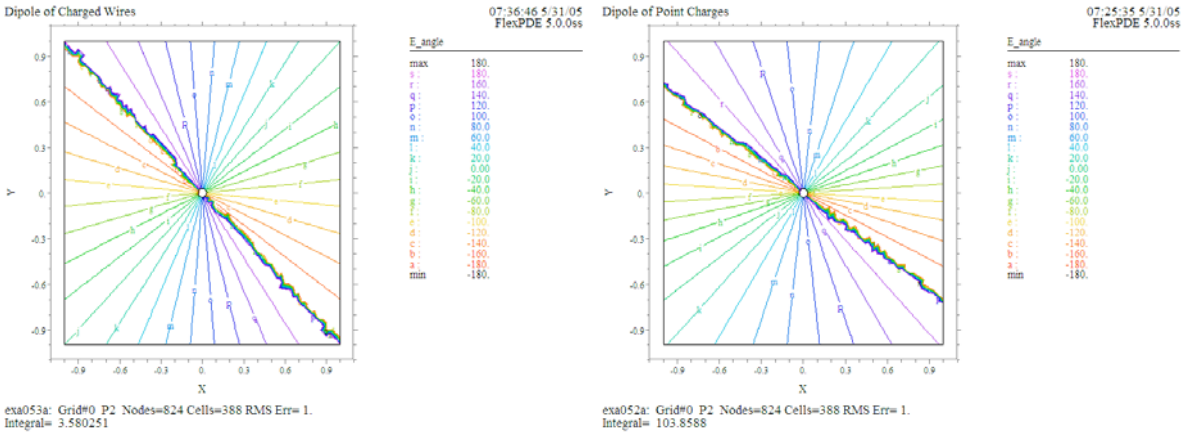


Dipole of Charged Wires

The 3D dipole field has its 2D counterpart generated by two adjacent, charged wires. The changes to *exa053* required to study this case are shown below.

```
TITLE                                                    { exa053a.pde }
  'Dipole of Charged Wires'
SELECT
  spectral_colors
DEFINITIONS
  Lx=1    Ly=1    d0=Ly*1e-6    r0=3e-2    qw=1
  eps0=8.85e-12    c=1/(4*pi*eps0)
  U=qw*c*ln(x^2+(y+d0)^2)- qw*c*ln(x^2+(y-d0)^2)
…
  start(r0,0) arc( center=0,0) angle=360    { Replaces two circles }
PLOTS
…
```

The resulting plots are hard to distinguish from those due to point charges, but there are definite differences, as illustrated by the following figures. Surprisingly, the field direction is constant along any radial line. It appears that the line corresponding to 180^o is diagonal for the wire dipole but rotated anti-clockwise for the point dipole.



The Gauss Integral

We may relate an integral of the field to the total charge $Q = \sum q_i$ by the Gauss relation[3p324]

$$\oiint_S \mathbf{D} \cdot \mathrm{d}\mathbf{s} = Q$$

where $\mathbf{D} = \varepsilon_0 \mathbf{E}$ and S is the surface enclosing the set of charges Q. Here, the vector $d\mathbf{s}$ is directed along the outward normal to the surface S and its magnitude is equal to the area element.

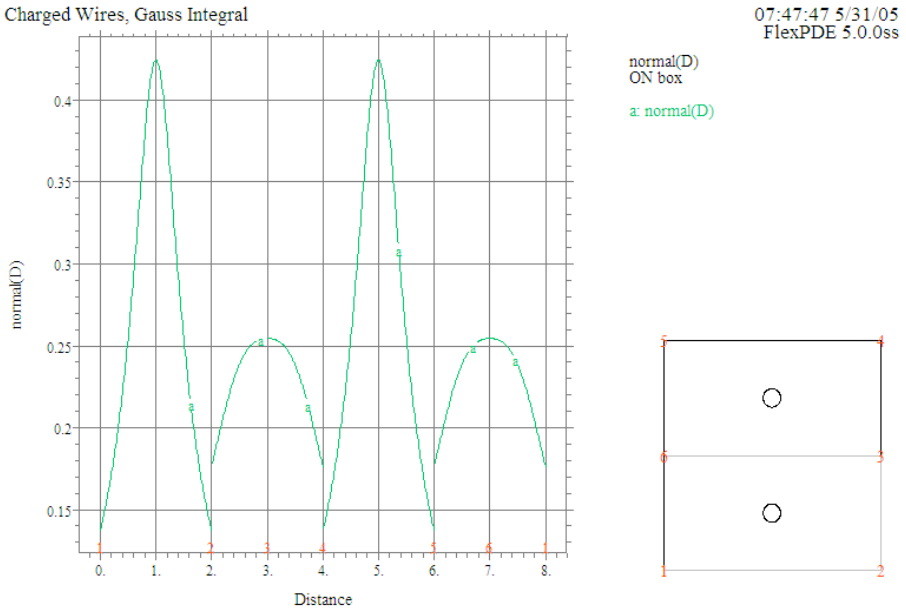
A convenient feature of FlexPDE is that an elevation plot automatically presents the integral of a function along a given line in the plane. This information is displayed at the bottom of such a plot.

We now extend *exa053* to calculate the Gauss integral over two different, closed surfaces, both parallel to the z-axis. We obtain the integral *per unit length* by an elevation plot over the intersection of a surface with the (x, y) plane.

The first intersection ('box') includes the full domain, and the second one ('half_box') contains half of the domain. In the following descriptor, these curves are defined under *boundaries*.

```
TITLE                                                    { exa054.pde }
  'Charged Wires, Gauss Integral'
SELECT
  spectral_colors
DEFINITIONS
  Lx=1    Ly=1    d0=Ly/2    r0=8e-2    qw=1 { Both charges positive }
  eps0=8.85e-12    c=1/(4*pi*eps0)
  U=-qw*c*ln(x^2+(y+d0)^2)- qw*c*ln(x^2+(y-d0)^2)
  Ex=-dx(U)    Ey=-dy(U)    E=-grad(U)    Em=magnitude(E)
  D=eps0*E                                        { Vector D }
BOUNDARIES
region 'domain'
  start 'box' (-Lx,-Ly)  line to (Lx,-Ly) to (Lx,Ly)  to (-Lx,Ly) close
  start(r0,-d0) arc( center=0,-d0) angle=360    { Exclude circle }
  start(r0,d0) arc( center=0,d0) angle=360      { Exclude circle }
feature  start 'half_box' (-Lx,-Ly)             { For line integral }
  line to (Lx,-Ly) to (Lx,0) to (-Lx,0) to (-Lx,-Ly)
PLOTS
  elevation( normal(D)) on 'box'    elevation( normal(D)) on 'half_box'
END
```

In the first elevation plot we use *normal* to extract the component D_n and take the integral over the 'box' already created. The following curve is the result.



exa054: Grid#0 P2 Nodes=799 Cells=372 RMS Err= 1.
Integral= 1.999985

The value of the integral becomes closely equal to 2 per unit length in the z direction, which is to be expected since the surface includes two wires with qw=1. The second integral yields one-half of that value.

Positive and Negative Charges in (ρ,z)

In the preceding section and in the whole chapter on gravitation (pp.25-33) we displayed fields in (x, y) space, although they are in fact three-dimensional. What we illustrated there was a cross-section of the field in a plane going through the objects. That plane is unique, in the sense that the field vectors are confined in that plane. If we had chosen a parallel plane at a certain distance from the objects, we would have missed the field component perpendicular to that plane.

In the case of point charges located on a straight line, the field will be axially symmetric. Consequently, if we make plots in a plane going from the common axis outwards, we may completely describe the

field. In the next example we shall thus use *cylindrical* coordinates (ρ, z) for the graphical presentation.

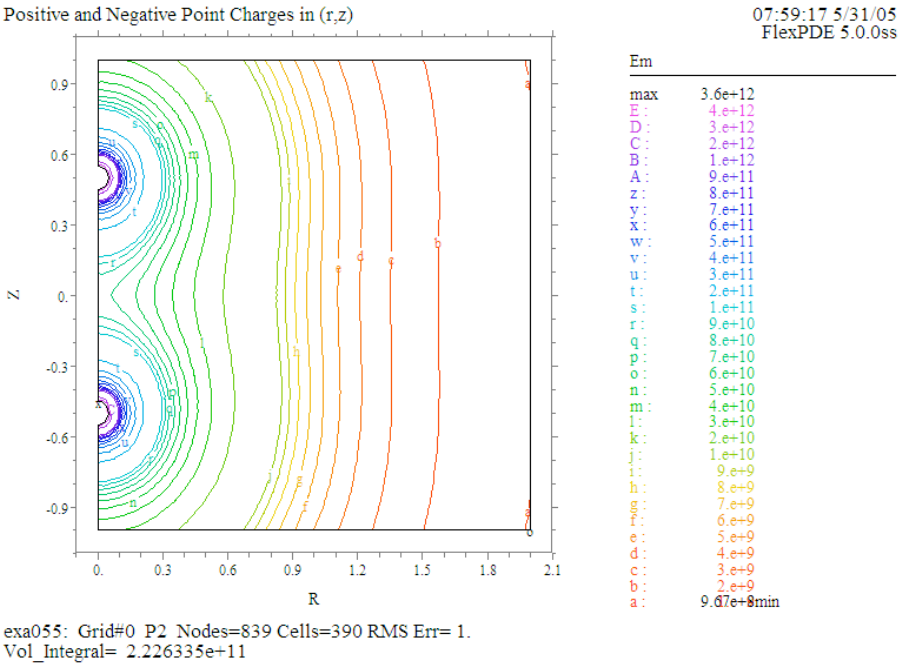
To study the field around opposite charges in this manner you only need to save *exa052* as *exa054* and modify the latter as follows. The first charge will now be of negative sign. Here, we introduce the new segment *coordinates*, which specifies axial symmetry by *ycylinder*. The components of **E** take the same form as in an (x, y) system.

```
TITLE                                                    { exa055.pde }
  'Positive and Negative Point Charges in (r,z)'
SELECT
  spectral_colors
COORDINATES
  ycylinder( 'r', 'z')
DEFINITIONS
  Lr=2    Lz=1    d0=Lz/2    r0=5e-2    q=1
  eps0=8.85e-12    c=1/(4*pi*eps0)
  U=-q*c/sqrt(r^2+(z+d0)^2)+ q*c/sqrt(r^2+(z-d0)^2)
  Er=-dr(U)    Ez=-dz(U)                          { Field components }
  E=vector( Er, Ez)   Em=magnitude(E)
  E_angle=sign( Ez)*arccos( Er/Em)/pi*180    { Degrees }
  D=eps0*E                                           { Vector D }
BOUNDARIES
region 'domain'                          { With half-circular indentations }
  start (0,-Lz) line to (Lr,-Lz) to (Lr,Lz) to (0,Lz) to (0,d0+r0)
  arc(center=0,d0) angle=-180 to (0,d0-r0) line to (0,-d0+r0)
  arc(center=0,-d0) angle=-180 to (0,-d0-r0) line to close
feature                                                { For line integrals }
  start 'can' (0,-Lz)  line to (Lr,-Lz) to (Lr,Lz)  to (0,Lz)
  start 'half_can' (0,-Lz) line to (Lr,-Lz) to (Lr,0) to (0,0)
PLOTS
  contour( abs(U)) log
  contour( abs(Er)) log   contour( abs(Ez)) log
  contour( Em) log   vector( E/Em)   contour( E_angle)
  elevation( normal(D)) on 'half_can'
  elevation( normal(D)*2*pi*r) area_integrate on 'half_can'
  elevation( normal(D)) on 'can'
END
```

Here, have excluded half-circular regions around the charges. Since these are at the very boundary, we simply create two indentations.

Even before we run this file, we may click on *Controls, Domain Review* to verify that the boundary looks as we intended.

The following plot of Em confirms that we have excluded the singularities. It also gives a rough idea about the field distribution, when combined with a direction plot.



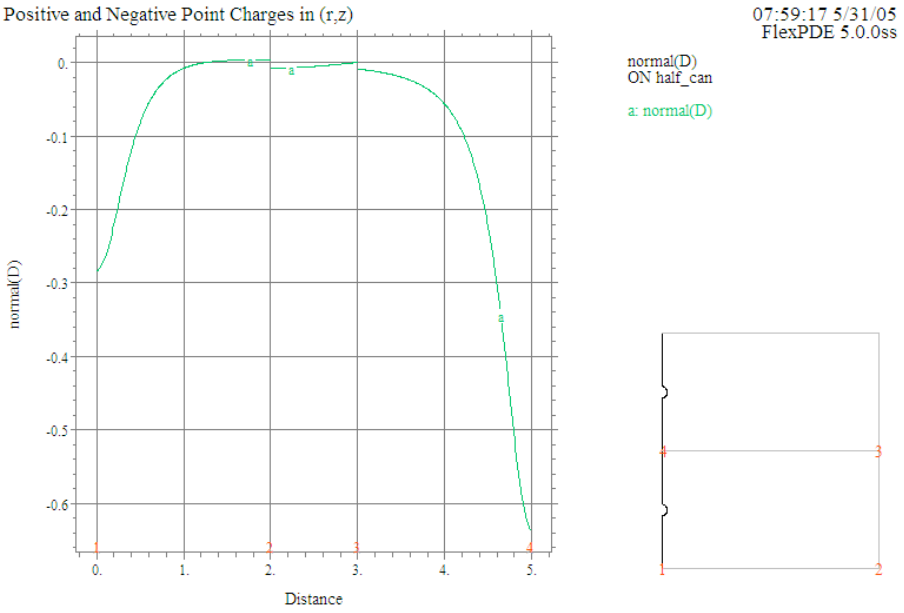
The above plot reports a value for *vol_integral* (which we shall not use). This means that FlexPDE automatically integrates Em over 3D space whenever cylindrical coordinates are declared.

In this descriptor we have defined two rectangular features named 'can' and 'half_can'. We generate a Gauss enclosure by defining three lines: for the bottom, top and cylindrical surfaces. The enclosure ('can') may be thought of as generated by rotating this rectangle around the axis.

We now need to integrate over a 3D surface, rather than along a line. This means that we would have to multiply the integrand D_n by $ds = 2\pi\rho\, dl$ (dl being the length of the line element) before summation to obtain the integral over cylindrical surface elements.

When (ρ, z) coordinates have been declared, however, the program automatically includes the factor $2\pi\rho$, as is evident from the integral of normal(D). As demonstrated, we may also supply the factor 2*pi*r explicitly and attach the modifier area_integrate to eliminate the

internal $2\pi\rho$ factor. The two integrals agree very well, but of course the actual plots differ. The first alternative gives us the actual variation of normal(D) along the *feature*.



This figure reports *surf_integral* to remind us that the integral is automatically taken over a surface of rotation. The elevation plot of D_n is for the 'half-can'. The discontinuities of the integrand at the corners are caused by the sudden changes of the normal direction. The integral value is evidently what we should expect.

The last elevation plot yields the charge included by 'can', which should be zero. The numeric value obtained by integration does not vanish exactly, but it is very small compared to the value obtained for 'half_can'.

Exercises

Exploit *SaveAs* and the *Copy*, *Paste* facilities to simplify the following tasks.

❑ Add a contour and a surface plot of Ex to *exa051* and choose a suitable viewing angle before printing.

❑ Modify *exa052* by putting d0 equal to 0.1, to explore the transition to a dipole field.

❑ Using *exa052a* as a guide, plot a dipole field with the two charges on a horizontal, rather than a vertical, line. How is the resulting field different? Then plot the combined field due to both dipoles and study the results.

❑ Add contour plots of div(**E**) and $\text{curl}_z(\mathbf{E})$ to *exa052a*. Explain the different character of the results.

❑ Add contour plots of div(**E**) and $\text{curl}_z(\mathbf{E})$ to *exa053a*. Compare the results to those of the preceding exercise.

❑ Type the expressions for the dipole field components[3p83], into the *definitions* segment of *exa052a* (notice that `d=2*d0`). Add logarithmic contour plots of the analytic expressions for `abs(Ex)` and `abs(Ey)` and compare to the previous results.

6 Laplace and Poisson Equations

We are finally coming to the core of this book: calculating a field from the governing partial differential equation (PDE) and certain knowledge about conditions on the boundary.

Known Values on Boundaries

The first example is the simplest one possible. We shall solve the Laplace equation

$$\nabla^2 U \equiv \nabla \cdot \nabla U \equiv \text{div}(\text{grad}(U)) = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \qquad \bullet$$

using known values of $U(x,y)$ on a rectangular boundary. FlexPDE accepts both div and grad as operators and transforms them internally into the above 2nd-order PDE.

The following descriptor for this problem makes use of a few new commands, given in bold characters. First we specify the maximum *relative error* of the solution by the optional statement errlim=1e-5. We can request this high accuracy, because we suspect that the problem will be an easy one.

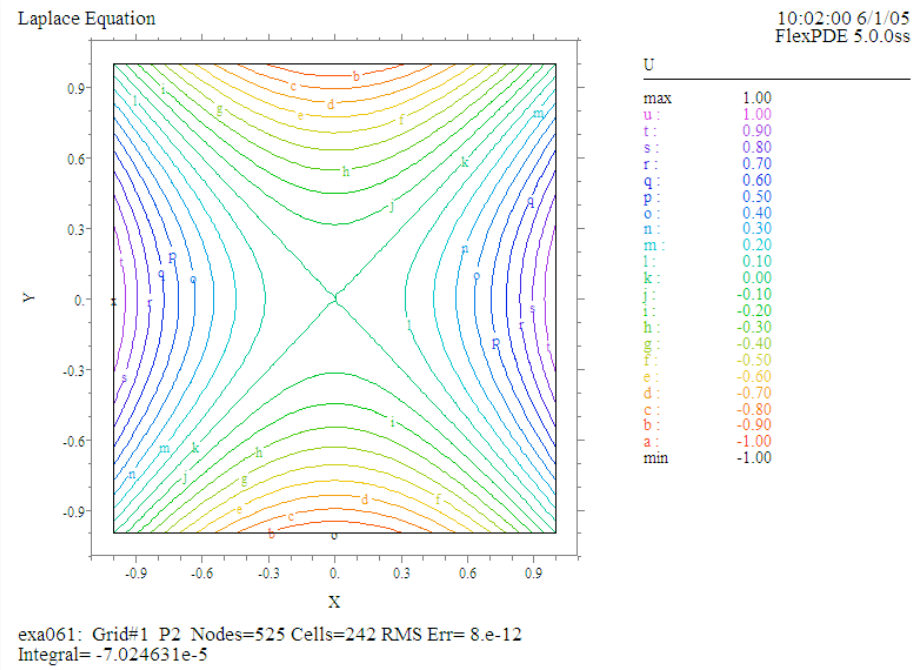
The next new keyword is *variables*, which heads the segment for the dependent variable: the one to solve for. Then comes the *equations* segment with the Laplace PDE.

The solution area limited by the boundary is called the *domain*. In the *boundaries* segment, we specify values of the dependent variable U on each of the four sides of the rectangle by *value* statements.

We could have supplied almost any function for the boundary values, but the skeptical reader might want to convince himself that the solution emerging is correct. The best way of checking the numerical results is to use boundary values taken from an analytic solution. It is known[1p477] that the real as well as the imaginary part of

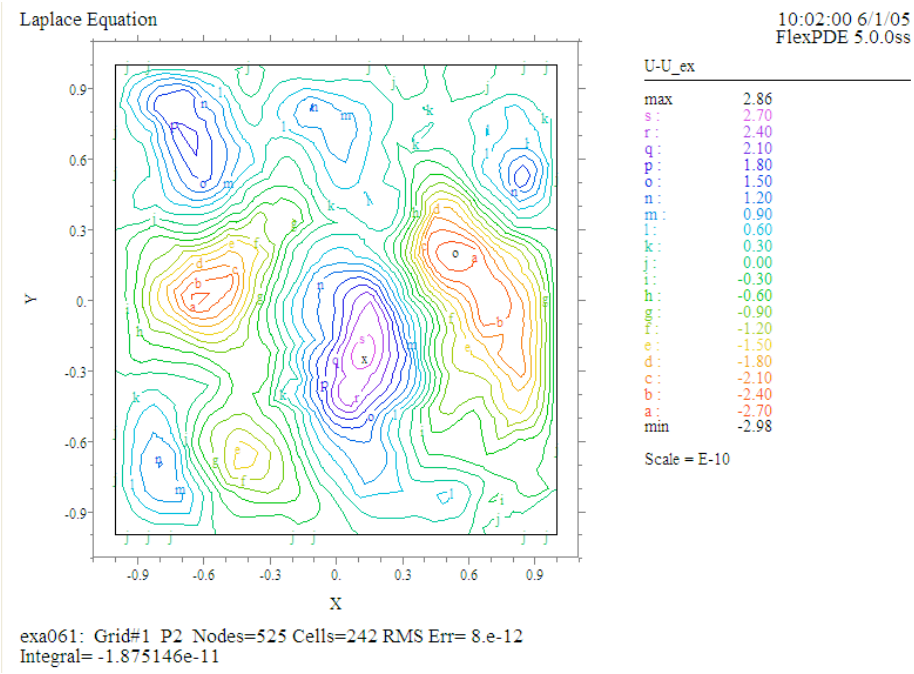
any simple complex function $f(z)$ satisfies the Laplace equation. Thus we specify $U = \mathrm{Re}(z^2) = x^2 - y^2$ for every point on the boundary. The program cannot "know" that U_ex actually *is* the solution; it only uses the values on the boundary and the PDE. This means that our test is non-trivial.

```
TITLE                                                    { exa061.pde }
  'Laplace Equation'
SELECT                                          { Student Version }
  spectral_colors     errlim=1e-5               { Requested relative accuracy }
VARIABLES
  U                                             { Unknown }
DEFINITIONS
  Lx=1     Ly=1     U_ex=x^2- y^2              { Exact solution }
EQUATIONS
  div( grad( U))=0                              { Laplace PDE }
BOUNDARIES
region 'domain' start(-Lx,-Ly)
  value(U)=U_ex   line to (Lx,-Ly)              { U known on boundary }
  value(U)=U_ex   line to (Lx,Ly)
  value(U)=U_ex   line to (-Lx,Ly)
  value(U)=U_ex   line to close
PLOTS
  contour( U)     surface( U)     contour( U_ex)     contour( U-U_ex)
END
```



In seconds, the program produces the above contour plot of the solution.

The plots of U and U_ex are indistinguishable to the naked eye. In anticipation we added a plot of U-U_ex, which presents the error of the numerical solution. As you see from the next plot the error is less than 10^{-9} of the maximum value of U for this number of nodes.



In this example we specified values on a rectangular boundary, but we could just as well have made it triangular, circular or half-circular. We could even have cut out areas inside the domain, supplying values on the inner boundaries too. Rigorous literature on PDEs proves that the Laplace equation has exactly one solution, if the values on the boundaries are known.

Values and Derivatives on Boundaries

As an alternative to function values we could have specified values of the normal derivative, $\nabla U \cdot \mathbf{n} \equiv \partial U/\partial n$, taken in a direction **n** perpendicular to the boundary, and outwards from the solution domain. Let us use such a condition in a new descriptor, based on *exa061*, where the *region* segment reads as follows.

```
TITLE                                                    { exa062.pde }
  'Laplace Equation'
...
region 'domain' start(-Lx,-Ly)
  value(U)=U_ex   line to (Lx,-Ly)
  natural(U)=2*x  line to (Lx,Ly)          { Outward derivative, dx(U) }
  value(U)=U_ex   line to (-Lx,Ly)
  value(U)=U_ex   line to close
…
```

The FlexPDE notation for $\partial U/\partial n$ is natural(U), for obscure traditional reasons. Running this descriptor we again find excellent agreement with the exact solution.

Multiplying through the PDE

There is an important point to notice about *natural* boundary conditions. One would think that multiplying all the terms of a PDE by the same factor would not change the solution, but this is not so. In fact, we need to multiply the natural boundary conditions by the same factor. This occurs because the program integrates the PDE by parts, making this factor appear in the result. Let us illustrate this by a modification of *exa062*.

```
TITLE                                                   { exa062a.pde }
  'Laplace Equation, Multiplied'
...
  div( 5*grad( U))=0                          { Laplace PDE }

  natural(U)=5*2*x  line to (Lx,Ly) { Outward derivative, dx(U) }
…
```

Running this file we find the same small error as before. If we erase the multiplier in the natural boundary condition, however, the error becomes very large.

We shall encounter several examples in this book, where a multiplier occurs in this position inside the PDE. Usually, the factor is not an integral number but a materials property.

Derivative Boundary Conditions Only

Now let us proceed to the extreme case, supplying *only* derivative conditions on all of the four boundary lines, using the modifications to *exa061* indicated below. For the line at the bottom of the rectangle we must enter the *downward* derivative, $\partial U/\partial(-y) = 2y$.

```
TITLE                                                    { exa063.pde }
  'Laplace Equation'
...
region 'domain' start(-Lx,-Ly)
  natural(U)=2*y   line to (Lx,-Ly)              { Outward derivatives }
  natural (U)=2*x   line to (Lx,Ly)
  natural (U)=-2*y   line to (-Lx,Ly)
  natural (U)=-2*x   line to close
...
```

The deviation of the numeric solution from the exact one is no longer small, but it takes an almost constant value. This may be no surprise, if you thought (correctly) that a function could not be uniquely determined by its derivatives.

You may now guess that derivative boundary conditions are sufficient except for a constant, and fortunately the program also allows us to assign a value to a single point. Using this feature we modify *exa063* as follows.

```
TITLE                                                   { exa063a.pde }
  'Laplace Equation'
...
region 'domain'   start(-Lx,-Ly)   point value(U)=U_ex
...
```

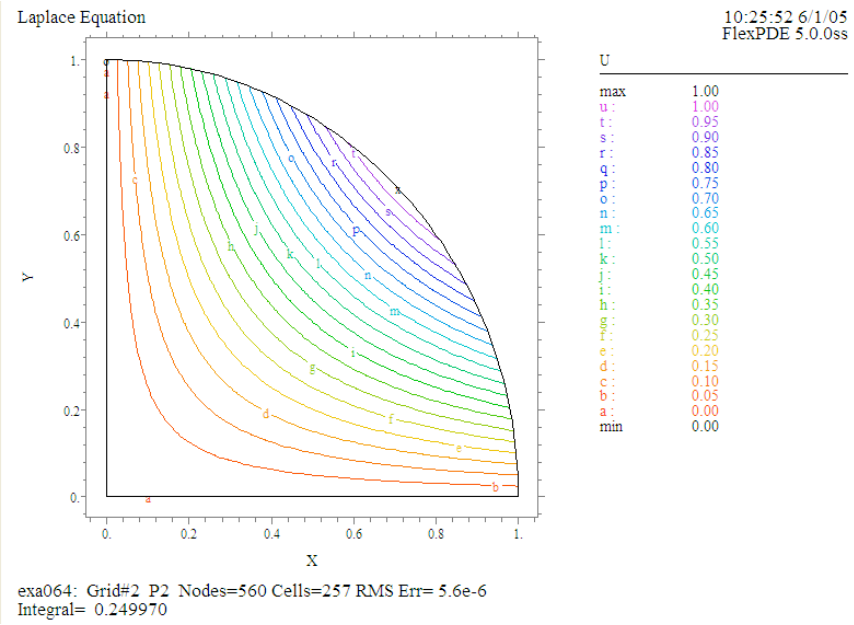
The *point value* statement must be typed immediately *after* the parentheses containing the point coordinates. The value applies only to that point and takes no effect later in the descriptor. Running this descriptor file we obtain an error almost as low as for *exa061*.

Solution over a Quarter-Circle

We shall now solve the Laplace equation over the first quadrant of a circle, using $\mathrm{Im}(z^2)$ for the boundary values, as explained by the partial descriptor below, based on *exa061*. Here we must supply coordinates for the *center* of the *arc* in order to specify uniquely the boundary curve we want.

```
TITLE                                                    { exa064.pde }
  'Laplace Equation'
...
DEFINITIONS
  r0=1     U_ex=2*x*y                                  { Exact solution }
  ...
region 'domain'  start(0,0)
  value(U)=U_ex   line to (r0,0)
  value(U)=U_ex   arc(center=0,0) to (0,r0)
  value(U)=U_ex   line to close
...
```

Running this new descriptor you obtain the plot below for U. The last plot (not shown here) indicates that the actual error in U is larger than that estimated (RMS Error, MAX Error) at a number of spots close to the curved boundary. The error is largest inside the cells having a curved side, but it remains smaller than errlim over the rest of the domain.



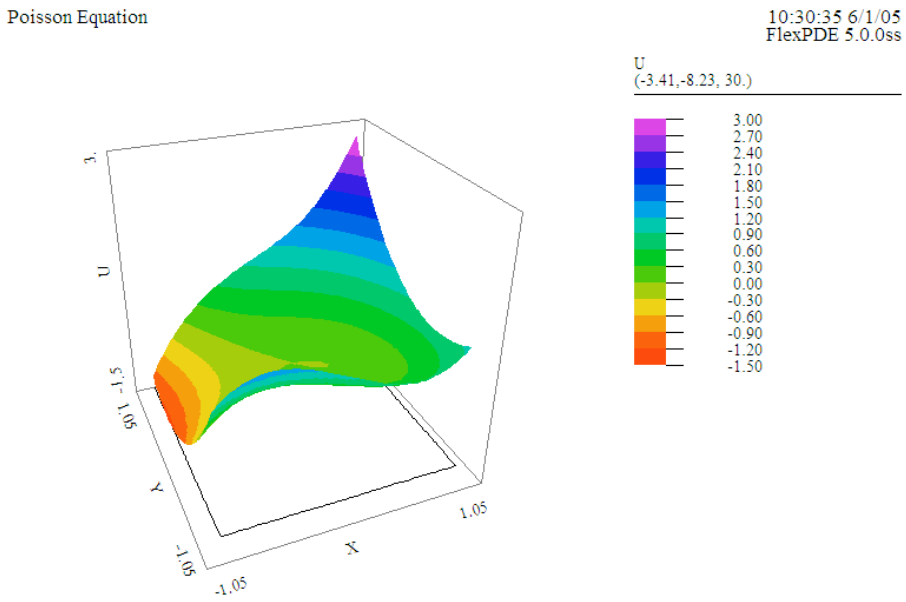
The Poisson Equation

The Poisson equation is similar to the Laplace PDE, the only difference being that the right member is a function, rather than zero. It is difficult to find exact solutions to such problems, but it is easy to find a PDE that corresponds to a given exact solution. We need only apply ∇^2 to a function of our choice to obtain the corresponding right member of the equation. To that solution we could then add any known solution to the Laplace equation (e.g. x*y), since that would contribute zero to the right side of the PDE.

```
TITLE                                                          { exa065.pde }
  'Poisson Equation'
SELECT
  errlim=1e-3     spectral_colors
VARIABLES
  U
DEFINITIONS
  Lx=1     Ly=1     U_ex=(x^3+y^2)+ x*y                { Exact solution }
EQUATIONS
  div( grad( U))=6*x+2
BOUNDARIES
region 'domain'  start(-Lx,-Ly)   value(U)=U_ex
  line to (Lx,-Ly)  to (Lx,Ly)   to (-Lx,Ly)  close
PLOTS
  contour( U)     surface( U)     contour( U_ex)     contour( U-U_ex)
END
```

Here, the *value* boundary condition takes the same form for all four sides of the square. In such a case, the program permits us to type the expression for the value only once, making it automatically valid for the following segments of the boundary.

The next figure shows the surface plot of the solution. The last figure (not shown here) indicates that the solution is inside the error limit requested, except at one point.



exa065: Grid#1 P2 Nodes=525 Cells=242 RMS Err= 6.7e-5
Integral= 1.334098

Exercises

❑ In *exa061*, try using the function U_ex=x^2+y^2 (which is *not* an analytic solution to the PDE). Do you still obtain a numeric solution?

❑ Solve *exa061* over a rectangular region of the same size using value boundary conditions from U_ex=x/(x^2+y^2) (which is the real part of 1/z). Locate your domain so that it does not include the singular point at the origin.

❑ Solve the Laplace equation in *exa064* over the first quadrant, but specify normal derivatives on the two straight boundaries and values on the arc.

❑ Modify *exa065* with respect to the boundary conditions. Let the values be known on the horizontal boundaries, and introduce derivative boundary conditions on the vertical lines. Remember to use the *outward* derivative.

❑ Modify *exa065* again, specifying derivative boundary conditions on all boundaries and adding a *point value* at one of the corners.

❑ Change the boundary values in *exa065* to U_ex=sin(x)+y^2 and modify the Poisson equation accordingly.

7 Electrostatic Fields in (*x*,*y*) Space

We are now ready to treat some simple problems related to the real world. The electric potential U is the key to all the examples to follow. For a static (time-constant) electric field in the vacuum[3p153] we have

$$\mathbf{E} = -\nabla U \qquad \bullet$$

and

$$\nabla \cdot \mathbf{E} = 0 \qquad \bullet$$

The last of these equations expresses the trivial fact that charge is neither created nor destroyed in any volume element. These two relations combine to yield the Laplace equation

$$\nabla \cdot \mathbf{E} = \nabla \cdot (-\nabla U) \equiv -\nabla \cdot \nabla U = 0 \qquad \bullet$$

which we have already solved. Once the solution is known we may apply the gradient operator to obtain the field **E**.

Metal Rod in a Metal Box

We first consider a rectangular box of metallic material, extending far both ways in the z direction, in which a metal rod is suspended by insulating wires at the ends. The descriptor below applies to a cross-section at the middle of this system.

In the *boundaries* segment we define the geometry of the box, which is at constant potential. The actual value is arbitrary, and we choose to put $U = 0$. The boundary of this box consists of four line segments, but it is not necessary to assign a potential value to each of these, since the program remembers the first assignment and automatically applies it to the following ones.

The second part of this segment defines the geometry of the rod. The space containing the field will be that between the square and the circle, which means that the circular area does not belong to the

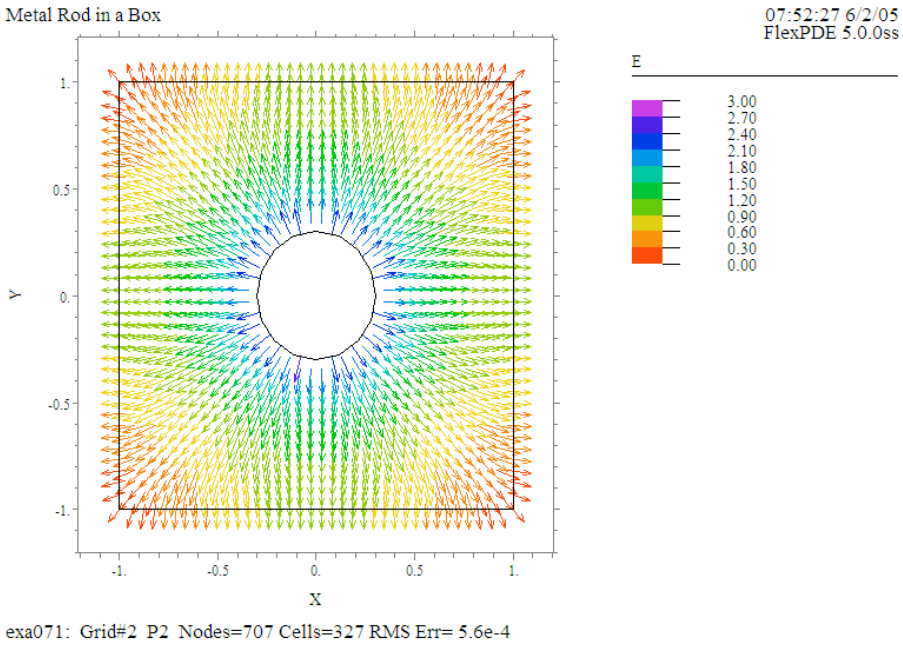
solution domain. We therefore *exclude* a circular region. The potential is constant (U_rod) for the surface of the rod.

```
TITLE                                                          { exa071.pde }
  'Metal Rod in a Box'
SELECT                                                    { Student Version }
  errlim=1e-3     spectral_colors
VARIABLES
  U
DEFINITIONS
  L=1    r0=0.3     U_rod=1.0
  Ex=-dx(U)     Ey=-dy(U)     E=-grad(U)     Em=magnitude(E)
EQUATIONS
  div( grad( U))=0
BOUNDARIES
region 'domain'
  start 'box' (-L,-L)  value( U)= 0                            { Metal box }
  line to (L,-L)  to (L,L)  to (-L,L)  close
  start 'rod' (r0,0)                                           { Cutout for rod }
  value(U)=U_rod   arc( center=0,0) angle=360
PLOTS
  contour( U)     surface( U)
  elevation( U) on 'box' range(-0.0001,0.0001)   { Plot range for U }
  elevation( U) on 'rod' range(0.999,1.001)
  contour( Em)     elevation( Em) on 'box'     vector( E) norm
END
```

The first two plots represent the potential U. It is clear already from these figures that the boundary values are those specified in the file. The two corresponding elevation plots confirm this in more detail. The variation of *U* is so small that we need to specify a plot range.

The resulting field is presented graphically by three plots pertaining to **E**. The contour plot of Em and the related elevation plot indicate that the field near the box is strongest halfway between the corners.

The following vector plot shows the direction of the field. The trailing modifier *norm* makes the lengths of all vectors equal, but the colors roughly indicate the actual magnitude. Here, we notice that the vectors are radial close to the rod and gradually redirect to meet the square enclosure at right angles.



Evidently, the MAX Error given in the *monitor* window is about as small as we requested, and the RMS Error is much smaller. The error would become further reduced if we used the Professional Version.

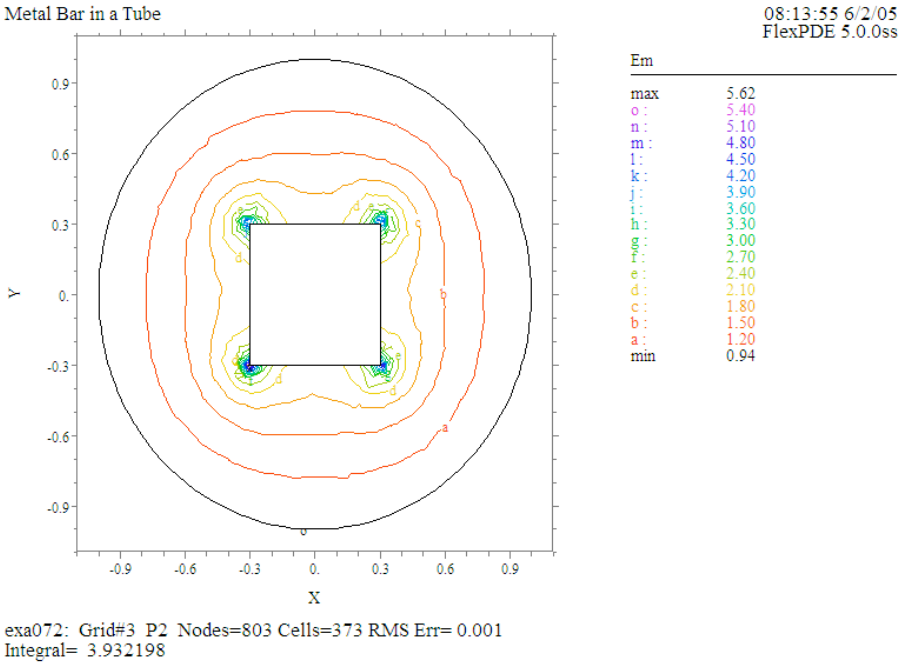
Metal Bar in a Tube

The next example is a variation of *exa071*, and you can easily use *SaveAs* and then rearrange the list to create the following file. The circular boundary is now on the outside.

```
TITLE                                                    { exa072.pde }
  'Metal Bar in a Tube'
SELECT
  errlim=1e-3     spectral_colors
VARIABLES
  U
DEFINITIONS
  r0=1.0     b=0.3     U_bar=1.0
  Ex=-dx(U)     Ey=-dy(U)
  E=-grad(U)     Em=magnitude(E)
EQUATIONS
  div( grad( U))=0
BOUNDARIES
region 'domain'
  start 'tube' (r0,0)  value(U)=0  arc( center=0,0) angle=360
```

```
  start 'bar' (-b,-b) value( U)=U_bar                    { Cutout for bar }
  line to (b,-b) to (b,b) to (-b,b)  close
PLOTS
  contour( U)     surface( U)
  elevation( U) on 'tube'     elevation( U) on 'bar'
  contour( Em)     elevation( Em) on 'tube'     elevation( Em) on 'bar'
  vector( E) norm
END
```

The results of this example are qualitatively different from what we just saw. The contour plot below shows that Em is strongly peaked at the corners of the bar. This field concentration at outer corners is of great practical importance in electrical engineering. Sharp metal tips in a field provoke discharge, for instance, and may even be the starting point of lightning.



We should also note that the maximum error reported in the monitor window is higher than what was asked for, because of the imposed node limit of the Student Version. The maximum error probably occurs at a corner of the bar.

Metal Rod across a Parallel Field

Let us now consider a problem where the potential on the outer boundary is not constant. We start with a homogeneous, parallel field $\mathbf{E}_0$ and then place a metal rod across it. Sufficiently far from the rod the field may still be considered to be homogeneous and parallel. In the following descriptor the horizontal boundaries are thus specified to have zero outward derivative, $\partial U/\partial n$, which means that the field is not allowed to cross those lines.

The vertical boundaries are required to pass a field component Ex equal to E0x. Since the normal of the exit (right) boundary is in the direction of positive x, the outward normal derivative becomes dx(U)=-Ex=-E0x. On the input boundary the normal is in the direction of *negative* x, and hence $\partial U/\partial n$ becomes -dx(U)=Ex=E0x.

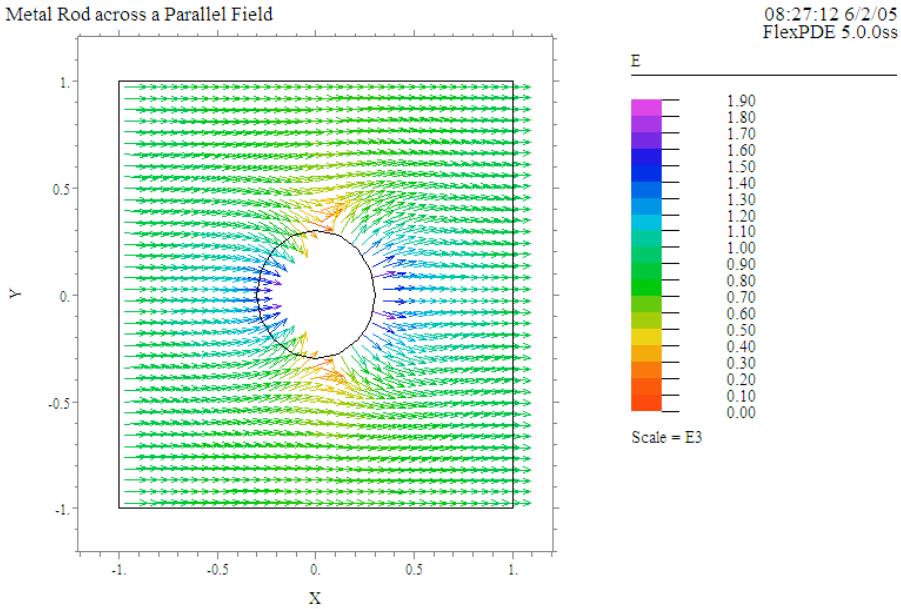
The inner boundary is a metallic surface, having constant potential. Since we are free to choose this value, we specify U=0. In the present example, the domain thus has *value* as well as *natural* boundary conditions, and we need not supply an additional *point value*.

```
TITLE                                                          { exa073.pde }
  'Metal Rod across a Parallel Field'
SELECT
  errlim=1e-4    spectral_colors    ngrid=1
VARIABLES
  U
DEFINITIONS
  L=1    r0=0.3    E0x=1e3
  Ex=-dx(U)    Ey=-dy(U)    E=-grad(U)    Em=magnitude(E)
EQUATIONS
  div( grad( U))=0
BOUNDARIES
region 'domain'
  start 'outer' (-L,-L)  natural ( U)=0  line to (L,-L)
  natural( U)=-E0x line to (L,L)                    { Outward dU/dx }
  natural( U)=0 line to (-L,L)
  natural( U)=E0x   line to close                   { Outward dU/dx }
  start 'rod' (r0,0)                                { Cutout for rod }
  value(U)=0  arc( center=0,0) angle=360
PLOTS
  contour( U)    surface( U)
```

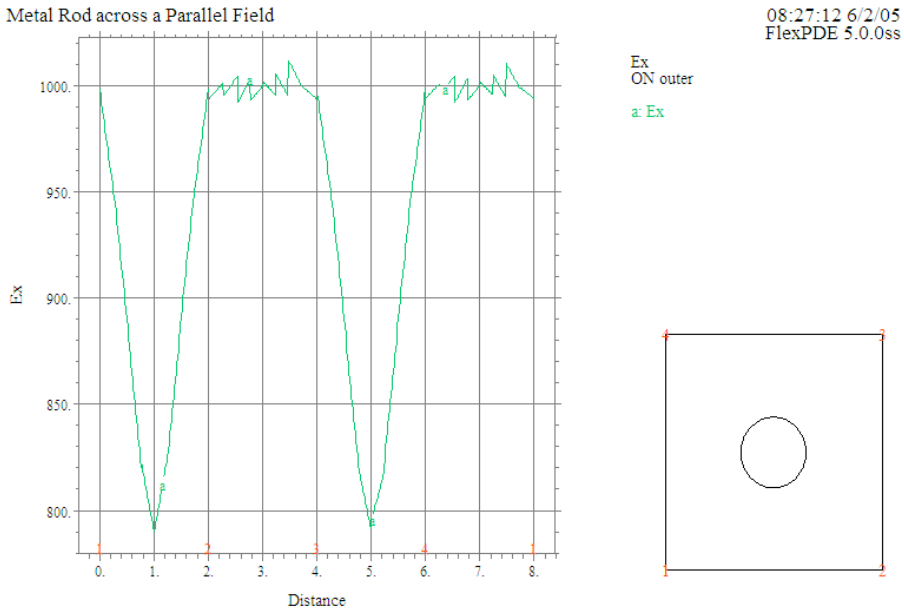
```
  elevation( U) on 'outer'     elevation( U) on 'rod'
  contour( Em)     vector( E) norm
  elevation( Ex) on 'outer'     elevation( Em) on 'rod'
END
```

In this problem, we are concerned with a field that becomes homogeneous at large distance from the central rod. Hence, the potential U is not expected to vary much on the outer boundary lines. The initial number of triangular cells on a straight boundary is 10 for the Student Version, as we already saw on p.10. The presence of the rod in the current example modifies the node density locally as needed. This means that we may make better use of the nodes by choosing a smaller initial number of cells. Hence, we specify ngrid=1 for the above descriptor.

The following vector plot shows that the parallel field enters from the left and leaves on the right side. The color code on the screen indicates the magnitude of **E** as well as the direction. We notice that $|\mathbf{E}|$ takes very small values near the bottom and top surfaces of the rod, which is explained by the constant value of U on the surface.



In the elevation plot of Ex (below), we notice that the segments pertaining to the vertical boundaries confirm that Ex has the specified magnitude and sign. The potential U, on the other hand, is not constant on these lines.



The other figures should be interpreted and compared to each other for a full understanding of all the aspects of the problem. Generally, it is rewarding to ask yourself if all the plots are in accord with the others and with your intuitive expectations.

Exercises

❑ Change *exa071* by displacing the center of the rod diagonally by L/3 in the *x* and *y* directions.
❑ Add a plot of div(grad(U)) in *exa071* to verify that the PDE is also satisfied.
❑ Modify *exa072* by putting the square bar in a square enclosure with sides equal to 2*r0.
❑ Replace the central rod in *exa073* by a "half-moon", keeping the right half of the circle and returning by the point (r0/2,0). Also explore how the solution changes when you reverse the external field.

8 Electrical Conduction in (*x*,*y*) Space

In the preceding chapter, the entire volume of a metal object was at constant potential. If a voltage is applied between different parts of a given metallic object, however, this is no longer true and an electric current field will flow within the volume. In an ordinary metal, the electrical current density **J** is simply related to the field strength **E** by[3p203]

$$\mathbf{J} = \sigma \mathbf{E} = -\sigma \nabla U \qquad \bullet$$

where σ is the *electric conductivity*.

Conservation of charge requires that

$$\nabla \cdot \mathbf{J} + \frac{\partial \rho}{\partial t} = 0 \qquad \bullet$$

where ρ is the charge density, and the equation simply means that the current diverging from a volume element is equal to the rate of decrease of its charge. Since we shall be dealing with static phenomena, the time derivative is zero. Combining these relations we then obtain

$$\nabla \cdot \mathbf{J} = -\nabla \cdot (\sigma \nabla U) = -\frac{\partial}{\partial x}\left(\sigma \frac{\partial U}{\partial x}\right) - \frac{\partial}{\partial y}\left(\sigma \frac{\partial U}{\partial y}\right) = 0 \qquad \bullet$$

This is a PDE of the Laplace type, but the conductivity σ may vary in space and hence cannot be taken out of the parentheses. The program is able to solve it, however, as easily as in previous examples.

In a descriptor, we may simply type the above PDE as `div(J)=0`. FlexPDE contains a symbolic manager that looks for the definition of `J` in terms of `E`, and then `E` in terms of `U`. The final result is that we effectively solve a 2nd order PDE with `U` as the unknown variable. After solving for `U`, we obtain `E` and `J` from the definitions.

Conduction in a Rectangular Plate

Let us apply our scheme to the simplest possible example. We apply a voltage over a rectangular plate, so that the potential U of the lower edge is zero and that of the upper edge is 1.0 V.

Although we only consider a solution in the (x, y) plane, we allow the object to have extension in the z direction. We always assume, however, that the dependent variable (U) does not vary with respect to z. In other words, the solution is assumed to be valid in any parallel plane within the plate. The problem descriptor is shown below.

```
TITLE                                                    { exa081.pde }
  'Conduction in a Rectangular Plate'
SELECT                                                   { Student Version }
  errlim=1e-5     spectral_colors
VARIABLES
  U
DEFINITIONS
  Lx=0.5     Ly=1.0     cond=5.99e7                     { Conductivity of Cu }
  Ex=-dx(U)     Ey=-dy(U)     E=-grad(U)     Em=magnitude(E)
  Jx=cond*Ex     Jy=cond*Ey     J=cond*E     Jm=magnitude(J)
  U_ex=y/Ly*1.0                                          { Exact solution for 1.0 V }
EQUATIONS
  div( J)=0                                              { 2nd order PDE in U }
BOUNDARIES
region 'domain'
  start(-Lx,0)  value(U)=0  line to (Lx,0)
  natural(U)=0 line to (Lx,Ly)                           { Insulated, Ex=Jx=0 }
  value(U)=1.0 line to (-Lx,Ly)
  natural(U)=0 line to close                             { Insulated }
PLOTS
  contour( U)     surface( U)     vector( E)
  contour( Jx)     contour( Jy)     contour( Jm)     contour( U-U_ex)
END
```

Having run the above file you will notice that the solution satisfies the boundary conditions for U on the upper and lower surfaces and that the error is smaller than requested by errlim. You also find that the current density Jx is smaller than Jy by a factor of nearly 10^{12}. We could not expect it to become exactly zero, however, since a numerical solution is always approximate.

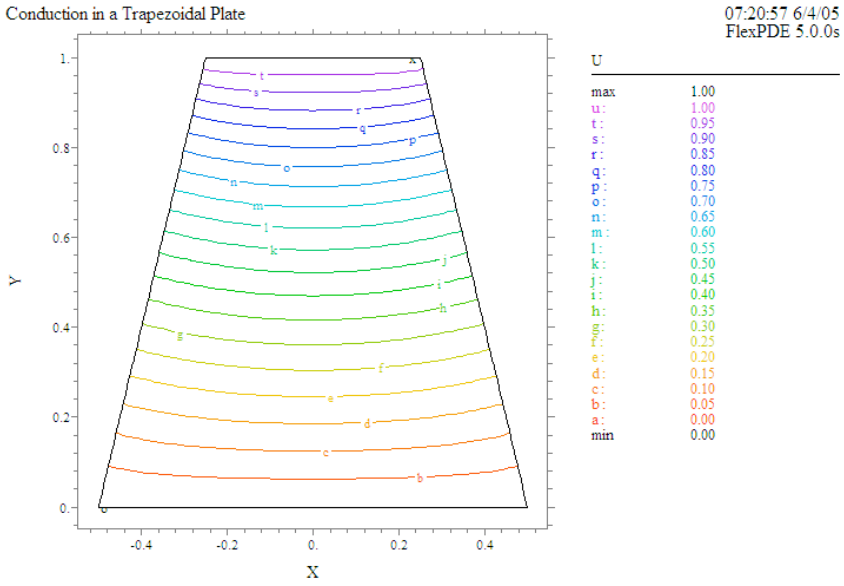
Conduction in a Trapezoidal Plate

We now attempt to solve a problem that does not have an elementary solution. Let us change the rectangle into a trapezoid by the following modified version of *exa081*.

```
TITLE                                                          { exa082.pde }
  'Conduction in a Trapezoidal Plate'
SELECT
  errlim=3e-5    ngrid=1    spectral_colors
VARIABLES
  U
DEFINITIONS
  L1=0.5    L2=0.25    Ly=1    cond= 5.99e7                  { Copper }
  Ex=-dx(U)    Ey=-dy(U)    E=-grad(U)    Em=magnitude(E)
  Jx=cond*Ex    Jy=cond*Ey    J=cond*E    Jm=magnitude(J)
EQUATIONS
  div( J)=0
BOUNDARIES
region 'domain'
  start(-L1,0)  value(U)=0  line to (L1,0)
  natural(U)=0 line to (L2,Ly)                                 { Insulated }
  value(U)=1.0 line to (-L2,Ly)
  natural(U)=0 line to close                                   { Insulated }
PLOTS
  contour( U)    surface( U)    vector( E)  norm
  contour( Jx)    contour( Jy)    contour( Jm)
END
```

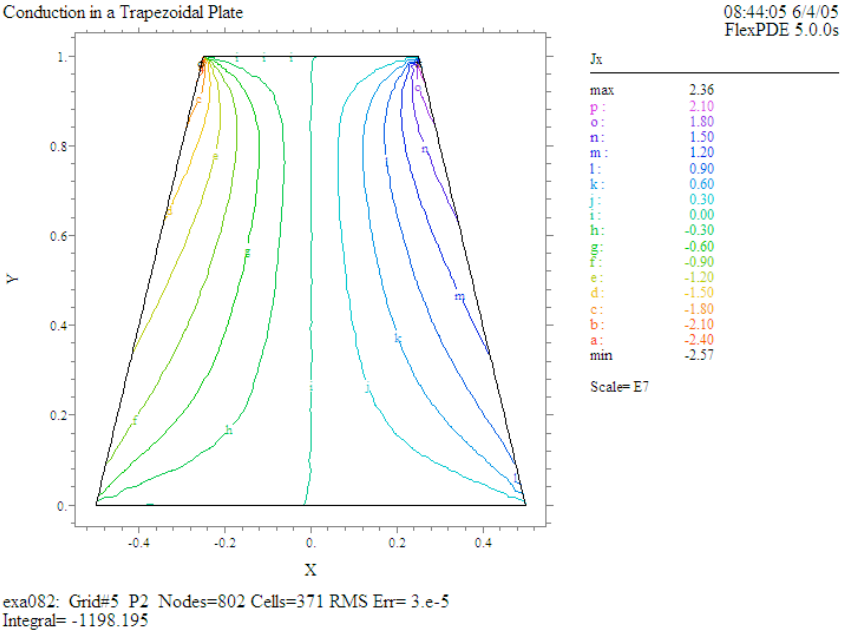
From the following plot of U we see that the boundary conditions are satisfied on the upper and lower edges of the plate. We also note that the contours of U meet the sloping boundaries at right angles, which demonstrates that the normal derivatives vanish.

In the vector plot there are no arrows pointing outwards through the insulated surfaces.



We are thus convinced that the boundary conditions have been respected, but what about the PDE itself ? The program internally checks for numerical errors at the end of each iteration, and the final report in the status window is MAX Err=1.0e-4. This suggests that the solution for U should be good to about 1 part in 10^4 (of the maximum), but this guarantee may not be strictly fulfilled at all points of the domain.

The plot below shows that the horizontal current density Jx is no longer zero, as it was in the rectangular geometry.



From the plot of Jm you will notice that the largest magnitudes occur at the upper corners. We may understand this by considering the contour plot of U above. Since the contours close to the upper surface must be horizontal and all the contours must be perpendicular to the sloping sides where they meet, the curves come closer to each other at the top corners, which enhances the gradient.

Apart from verifying the boundary conditions, we should look for symmetries in the problem and convince ourselves that the solution obtained conforms to it.

Checking the Solution

If we still feel skeptical, we may plot the left side of the PDE as follows. First we define eqn to be div(J) and then make a contour plot of the deviation from zero, usually called the *residual*. To achieve this we need only modify a few lines in *exa082*, as indicated below.

In addition we might check whether the current (per unit thickness of the plate) leaving the bottom surface equals that entering through the top face. We do that by two elevation plots, which report the integrated values (current).

```
TITLE                                                    { exa082a.pde }
  'Conduction in a Trapezoidal Plate, PDE Test'
…
  eqn=div( J)                                          { Left side of PDE }
EQUATIONS
...
  contour( eqn) as 'Residual'
  elevation( Jy) from (-L1,0) to (L1,0)                { ⇨ Current }
  elevation( Jy) from (-L2,Ly) to (L2,Ly)
END
```

You will find that the plot of the residual exhibits the irregular zero contours that we have learnt to associate with a vanishing function. The errors are evidently largest in the upper corners.

The elevation curves indicate that the integrated current densities (the currents) for the top and bottom surfaces agree within about 0.5%.

Radial Conduction in a Foil

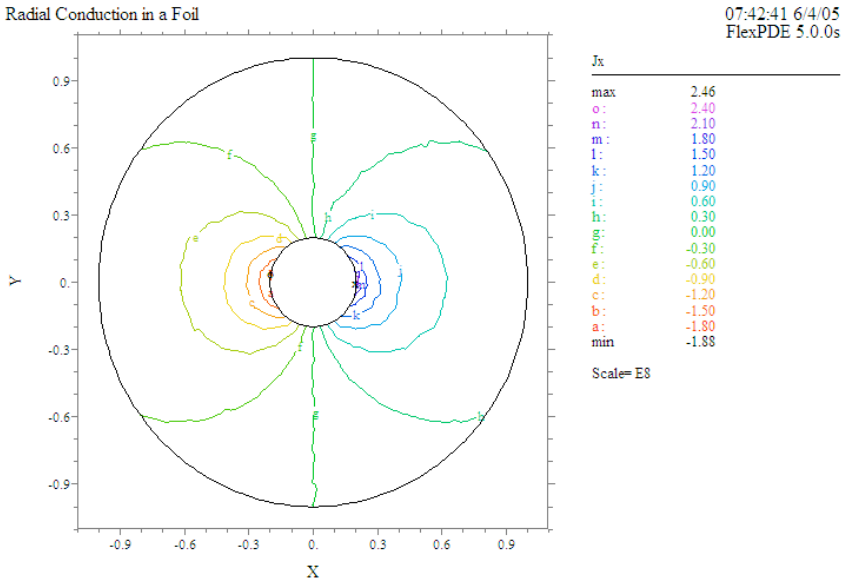
We shall now study the conduction through a thin foil, soldered perpendicularly to a tube and a coaxial rod, both of copper. In the preceding chapter we assumed that the extension of the field in the z direction was virtually infinite, and we studied the field far from the ends. In the case of a conducting plate or foil, we may abandon this restriction, because the boundaries will force the current to be parallel to the (x, y) plane.

A reasonable mathematical model would be a circular foil having constant potential levels on two concentric circles, one for the inner edge of the tube and one for the edge of the central rod. Using *exa082* as a template we modify it as follows.

```
TITLE                                                    { exa083.pde }
  'Radial Conduction in a Foil'
...
DEFINITIONS              { Replace first line of definitions }
  r1=0.2    r2=1    cond=5.99e7    U1=1.0                { Voltage }
...
region 'domain' start(r2,0)
  value( U)=0  arc( center=0,0) angle=360 close          { Outer }
  start(r1,0)  value( U)=U1  arc( center=0,0) angle=360  { Exclude }
PLOTS
  contour( U)    surface( U)    vector( E)  norm
  contour( Jx)    contour( Jy)    contour( Jm)
END
```

The vector plot shows the expected radial field, but the plot of Jx below may be more of a surprise.

The high symmetry of the present problem makes it possible to treat only part of the solution domain, e.g. only one quadrant. Since the field is radial, the radial cuts must have natural(U)=0 for symmetry reasons. The density of nodes will be four times larger, which should improve the quality of the solution.



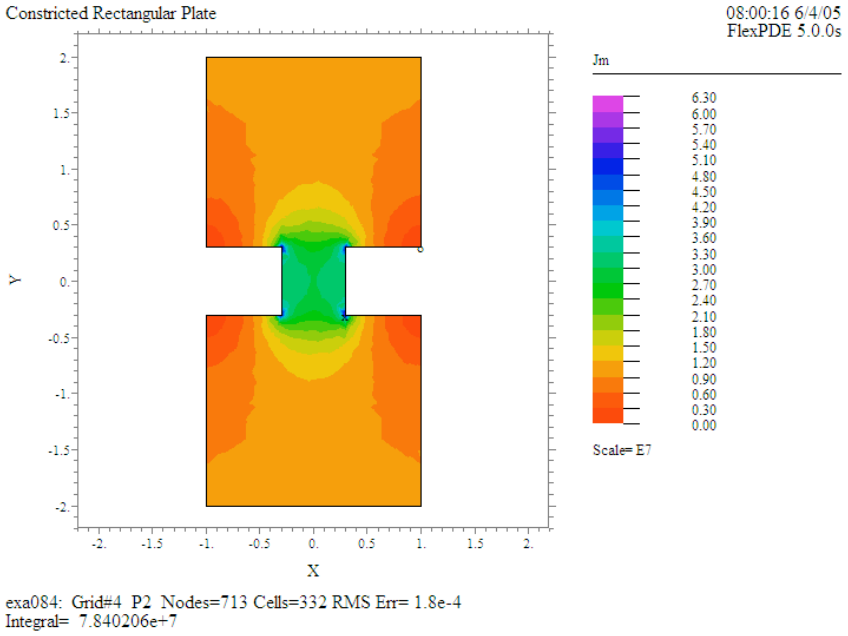
Constricted Rectangular Plate

In the next problem we study a rectangular plate with a constriction. Using *exa082* as a template we again modify to obtain the following descriptor file.

```
TITLE                                                    { exa084.pde }
  'Constricted Rectangular Plate'
SELECT
  errlim=3e-4    ngrid=1    spectral_colors
VARIABLES
  U
DEFINITIONS
  L=1.0    d=0.3    cond=5.99e7                          { Copper }
  Ex= -dx(U)    Ey=-dy(U)    E=-grad(U)    Em=magnitude(E)
  Jx=cond*Ex    Jy=cond*Ey    J=cond*E    Jm=magnitude(J)
EQUATIONS
  div( J)=0
BOUNDARIES
region 'plate'
  start 'boundary' (-L,-2*L)  value( U)=0 line to (L,-2*L)
  natural( U)=0                                          { Insulated }
  line to (L,-d) to (d,-d)  to (d,d)  to (L,d)  to (L,2*L)
  value( U)=1.0 line to (-L,2*L)
  natural( U)=0                                          { Insulated }
```

```
  line to (-L,d) to (-d,d) to (-d,-d) to (-L,-d) close
PLOTS
  contour( U)     surface( U)     vector( E) norm     surface( Em)
  contour( Jx) painted     contour( Jy) painted     contour( Jm) painted
  elevation( U) on 'boundary'     elevation( Jm) on 'boundary'
END
```

The plots resulting from the above descriptor demonstrate how the field deforms on going into a narrower or a wider cross section. The following figure shows the effect on the magnitude of the current density.



Judging from the colors, Jm is rather constant in the central part. The inner corners are points of very high field.

The plots of U and E demonstrate that the fields in the wider regions are strongly deformed, compared to the behavior close to the ends. These indications about the depth of perturbation are of great practical importance.

Plate Made of Two Different Metals

So far we have only been concerned with fields in homogeneous objects. A powerful feature of finite element analysis, however, is that

the materials properties are allowed to vary over a domain, gradually as well as stepwise.

The file *exa082* may serve as a starting point for the following descriptor. Here, we arrange for the current to flow from the upper-left corner to the lower-right one by imposing a potential difference.

The first region ('domain') always adjoins the surrounding world, and thus we must specify the conditions that apply to its boundary.

We assign a value for the conductivity parameter in the *definitions* segment, valid over all of 'domain'. Further down we define a sub-region, named 'iron', where the conductivity is assigned a different value. By overwriting the first assignment we may thus handle objects consisting of more than one material.

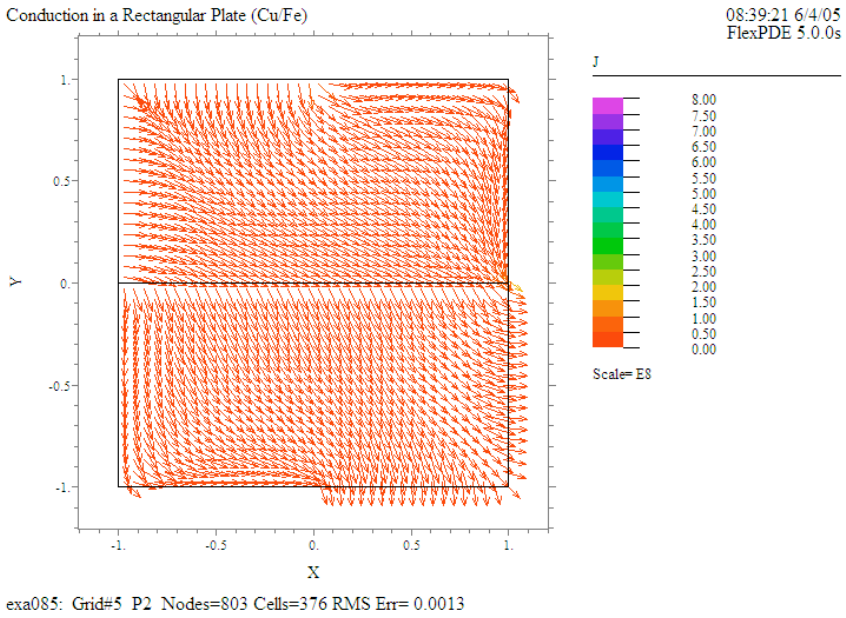
```
TITLE                                                    {exa085.pde }
  'Conduction in a Rectangular Plate (Cu/Fe)'
SELECT
  errlim=3e-4    ngrid=1    spectral_colors
VARIABLES
  U
DEFINITIONS
  Lx=1    Ly=1    cond=5.99e7                          { Conductivity }
  Ex=-dx(U)    Ey=-dy(U)    E=-grad(U)    Em=magnitude(E)
  Jx=cond*Ex    Jy=cond*Ey    J=cond*E    Jm=magnitude(J)
  J_angle=sign( Jy)*arccos( Jx/Jm)/pi*180
EQUATIONS
  div( J)=0
BOUNDARIES
region 'domain'                                          { Copper }
  start 'boundary' (0,-Ly)   value( U)=0 line to (Lx,-Ly)  to (Lx,0)
  natural(U)=0   line to (Lx,Ly)   to  (0,Ly)            { Insulated }
  value(U)=1.0  line to (-Lx,Ly)   to (-Lx,0)
  natural (U)=0  line to (-Lx,-Ly)  close                { Insulated }
region 'iron'   cond=1.03e7
  start(-Lx,-Ly)  line to (Lx,-Ly) to (Lx,0) to (-Lx,0) close
PLOTS
  contour( U)
  elevation( U) on 'boundary'
  elevation( normal( J)) on 'boundary'
  vector( J) norm    contour( J_angle) painted
  contour( Ex) painted zoom(-Lx/2,-Ly/2,  Lx, Ly)
  contour( Ey) painted zoom(-Lx/2,-Ly/2,  Lx, Ly)
  contour( Jx) painted zoom(-Lx/2,-Ly/2,  Lx, Ly)
```

```
  contour( Jy) painted zoom(-Lx/2,-Ly/2,  Lx, Ly)
END
```

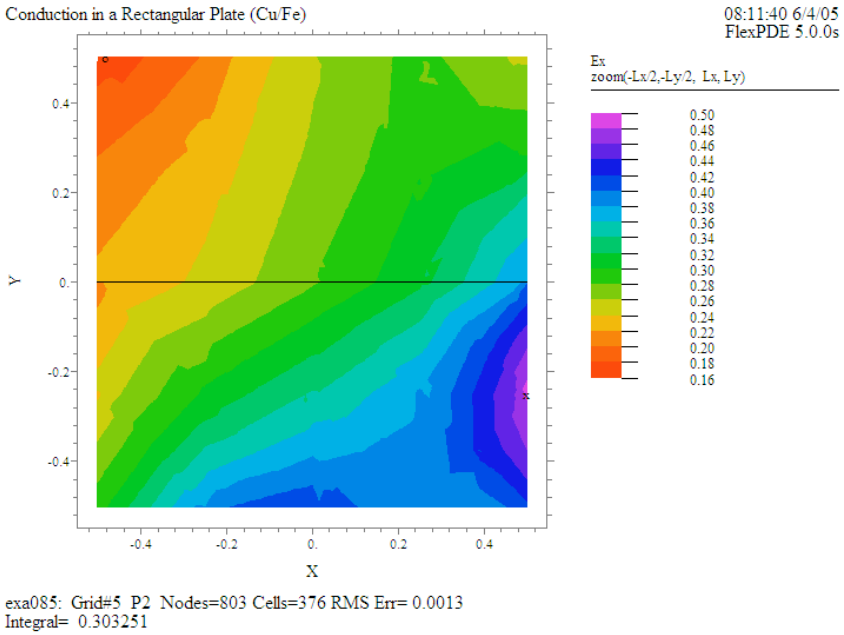
In the last four graphs we employ zoom(xlow,ylow, delx,dely), which restricts the plot to a rectangular portion of the total domain. The two first arguments are the coordinates of the lower-left corner and the last arguments the width and height of the plot window.

Running this descriptor we first find a contour plot of U. The potential is continuous across the interface between copper and iron, but the contours change direction.

Since the current flow vectors are perpendicular to the contours of U, the following plot of **J** is as expected. We also notice that the current direction is refracted on going through the interface.



The following painted contour plot of Ex shows that this function varies continuously across the line where the conductivity changes. This is in accord with the theoretical boundary conditions[3p116].



We also note that Jy varies continuously across the interface, whereas Ey and Jx exhibit discontinuities.

Plate with an Elliptic Insert

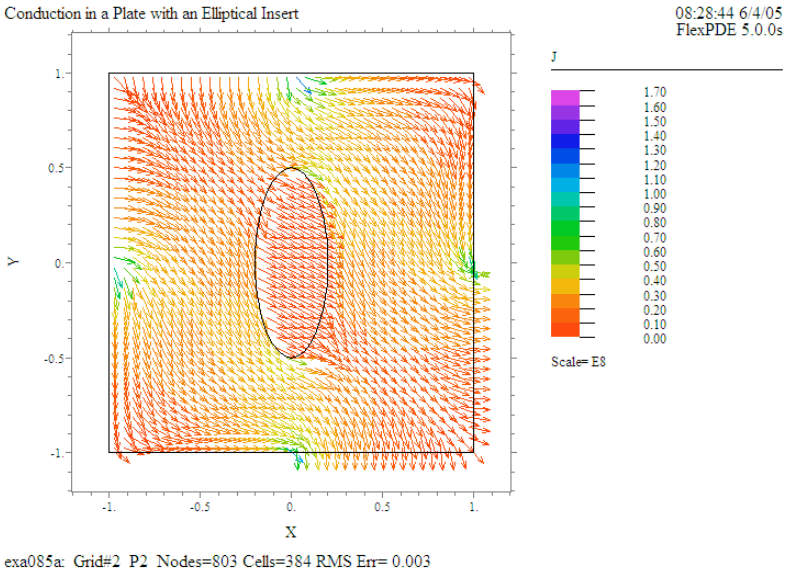
As an additional example of the behavior of the fields in the presence of two materials we introduce a region of elliptic cross-section consisting of iron.

The modification of the descriptor *exa85* really only concerns the *second* region. We shall see, however, that it is desirable to include a special plot over the insert by the modifier on 'iron'.

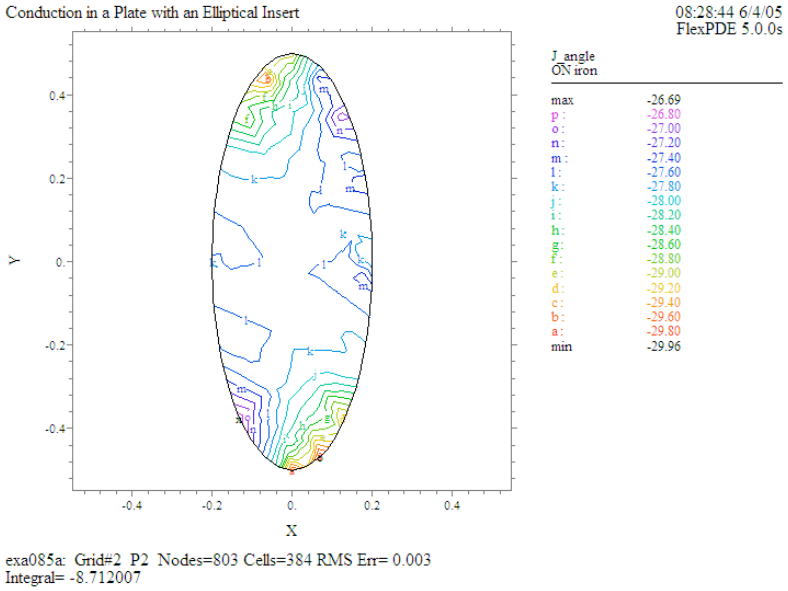
```
TITLE                                                    { exa85a.pde }
  'Conduction in a Plate with an Elliptical Insert'
...
region 'iron' cond=1.03e7
  start(Lx/5,0)
  arc( center=0,0) to (0, Ly/2) to (-Lx/5,0) to (0,-Ly/2)  close
PLOTS
  vector( J) norm    contour( J_angle)    contour( J_angle) on 'iron'
END
```

To create an elliptic curve, we proceeded much as we did when tracing a circle by quarters (p.17). The difference is that the axes are of different lengths.

The next figure is a vector plot of **J** over the entire domain. From that it appears that the flow becomes essentially parallel within the elliptic region.



The plot of the angle over the elliptic insert (below) exhibits a set of contours corresponding to about -28 degrees over most of the region.



Exercises

❑ In *exa082*, add elevation plots of U and Jm on the boundary. Explore what the results mean.
❑ Modify *exa082* by adding a square to the shorter side. Compare the solution with the original one.
❑ Use the symmetry of the circular foil (*exa083*) to solve for one quadrant only. Why is the number of nodes not reduced to one fourth? What is the advantage?
❑ Change *exa083* by moving the cutout by one hole-diameter. Is there still some symmetry to be exploited?
❑ Integrate $\mathbf{E} \cdot d\mathbf{l}$ along a vertical line through the constricted plate (*exa084*). Compare with the applied voltage.

9 Dielectrics in (*x*,*y*) Space

For dielectric materials, which become polarized when subjected to an external field, we should use relations[3p110] involving the electric displacement $\mathbf{D} = \varepsilon\mathbf{E} = \varepsilon_r\varepsilon_o\mathbf{E}$, where ε_r is the relative permittivity. The Maxwell equation[3p153] provides the corresponding PDE

$$\nabla \cdot \mathbf{D} = \rho_v \qquad \bullet$$

where ρ_v is the volume charge density *not* arising from polarization (free charge). In this chapter, the charge density term will always be zero, and hence we may type the FlexPDE equation as div(D)=0.

In terms of the potential U we may also write the Maxwell equation

$$\nabla \cdot \mathbf{D} = \nabla \cdot (\varepsilon\mathbf{E}) = \nabla \cdot (-\varepsilon\nabla U) = \rho_v \qquad \bullet$$

which we recognize as the Poisson type of PDE.

Coaxial Cable

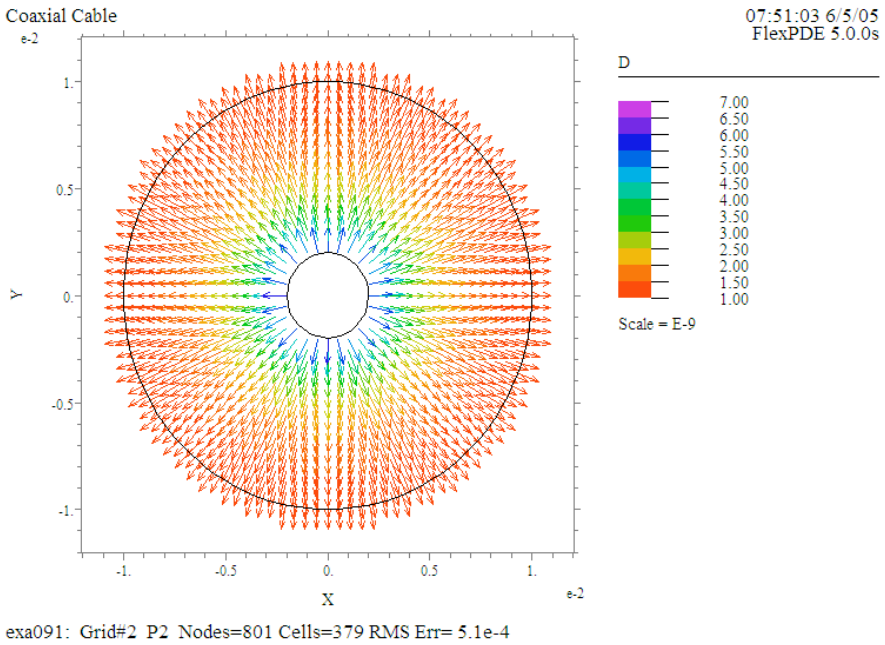
A simple configuration involving an electric field is one where objects are enclosed in a metal shield, which may then be taken as the outer boundary. We assume that the objects are long and parallel, which permits us to calculate the field in a cross-section at mid-distance between ends. The coaxial cable has this simple geometry. It consists of a metallic shield and a central wire, separated by a polymer tube.

From now on we shorten the descriptors by putting the symbols on the same line as the keywords TITLE, SELECT, and VARIABLES.

The descriptor file involves the dielectric displacement **D**. Unfortunately, the *x*-component of **D** may not be denoted Dx, since the program does not distinguish between upper and lower case and would read this as dx, which is the reserved word for a derivative. This is the reason for the unexpected notation Dex.

```
TITLE   'Coaxial Cable'                                          { exa091.pde }
SELECT     errlim=3e-4 ngrid=1     spectral_colors    { Student Version }
VARIABLES   U
DEFINITIONS                                                     { SI units }
   r1=2e-3     r2=1e-2     eps0=8.854e-12     U1=1.0
   eps=2.3*eps0                                                 { Permittivity }
   Ex=-dx(U)     Ey=-dy(U)     E=-grad(U)     Em=magnitude(E)
   Dex=eps*Ex     Dey=eps*Ey     D=eps*E     Dm=magnitude(D)
EQUATIONS
   div( D)=0                                                    { No volume charge }
BOUNDARIES
region 'domain'
   start 'outer' (r2,0)  value(U)=0  arc(center=0,0) angle=360
   start 'inner' (r1,0)  value(U)=U1 arc(center=0,0) angle=360  {Cut-out }
PLOTS
   contour( U)     contour( Dex)     contour( Dey)
   vector( D) norm     contour( Dm)     surface(Dm)
   elevation( Dm) on 'inner'     elevation( Dm) on 'outer'
END
```

The run produces the plot below, which shows that the field is of radial direction everywhere.



We may write similar descriptors to treat problems where the dielectric permittivity varies with the radius, either specified by a

continuous function $\varepsilon(R)$ or by several regions, each with a different but constant value of ε.

Capacitance

According to elementary electrostatics, the charge Q on either electrode of a capacitor is proportional to the voltage applied, or $Q = CU$, where C is the capacitance.

We may easily obtain C for the case of the above coaxial configuration. It is clear from the plots that the field is radial and that its magnitude depends only on the radius. The field **D** leaves or enters the metal in the direction of the normal, and Gauss' law[3p87] implies $D_n = |\mathbf{D}| = \sigma_s$, the latter quantity being the surface charge density.

In the present case, the expression for the capacitance is $C = Q/U_1 = 2\pi r_1 \cdot D_n$, based on the inner circle, and there is an equivalent formula for the outer circle. As we see from the elevation plots, the scatter of Dm (=D_n) over the boundary circles is very large. These plots, however, directly give us accurate charge values by automatic integration, i.e. 7.96e-11 and 7.92e-11 respectively.

The elementary expression for a cylindrical capacitor is

$$C = 2\pi\varepsilon / \ln(r_2 / r_1)$$

per unit axial length. This formula yields C=7.95e-11, and the FEA results agree quite well with that value.

Parallel Plate Capacitor

We shall now consider problems where the domain boundary is not at constant potential. This situation is similar to that of the fields around opposite point charges that we plotted (p.36), in the sense that the field strength decreases to a small value at large distance without actually becoming zero.

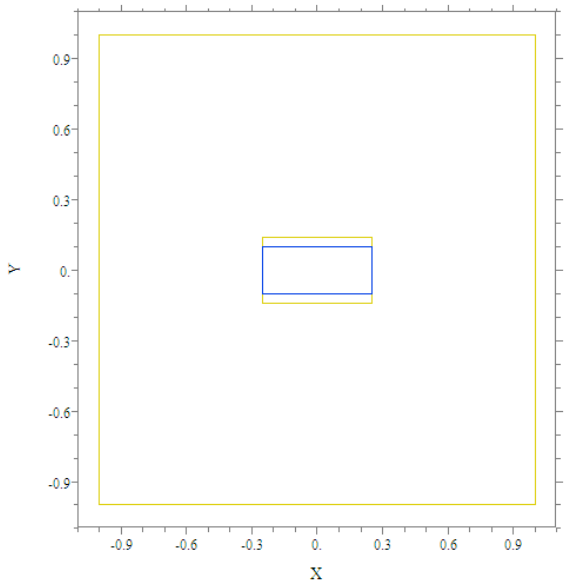
In the next example, we shall assume that the *total charge* on the object is zero. In view of Gauss' law this means that no field leaves the solution domain, on the average. We shall see that confining the

entire field in the domain is a realistic model, provided its size is chosen large enough.

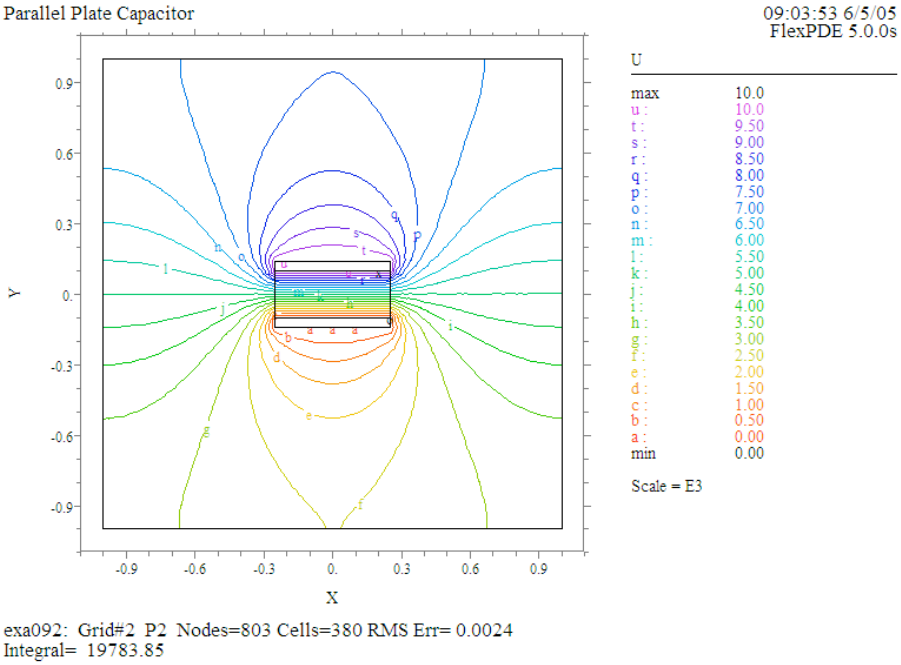
The following descriptor is for a cross-section of a parallel-plate capacitor, assumed to extend both ways along the *z*-axis. We reserve space for a domain of size 2L×2L and exclude two narrow rectangles corresponding to the metal plates. We also assign potential values to these inner boundaries. Finally we define a sub-region for the dielectric material between the plates. The internal plate-to-plate distance is 2*d0.

```
TITLE   'Parallel Plate Capacitor'                              { exa092.pde }
SELECT    errlim=3e-4    ngrid=1    spectral_colors
VARIABLES   U
DEFINITIONS
  L=1    xx=0.25    d0=0.1    dd0=0.4*d0     U0=1.0e4    { Voltage }
  eps0=8.854e-12    eps=eps0                  { Permittivity }
  Ex=-dx(U)    Ey=-dy(U)    E=-grad(U)    Em=magnitude(E)
  Dex=eps*Ex    Dey=eps*Ey    D=eps*E    Dm=magnitude(D)
EQUATIONS
  div( D)=0
BOUNDARIES
region 'domain'
  start 'outer' (-L,-L)   natural(U)=0              { Outer boundary }
  line to (L,-L) to (L,L)  to (-L,L) close
  start 'upper' (-xx,d0) value(U)=U0                { Upper plate }
  line to (xx,d0) to (xx,d0+dd0) to (-xx,d0+dd0) close
  start 'lower' (-xx,-d0-dd0) value(U)=0.0          { Lower plate, ground }
  line to (xx,-d0-dd0) to (xx,-d0) to (-xx,-d0) close
region 'glass'   eps=7.0*eps0                       { Dielectric }
  start(-xx,-d0)  line to (xx,-d0)  to (xx,d0) to (-xx,d0) close
PLOTS
  contour( U)    contour( Dm) log    contour( Dm) painted on 'glass'
  vector( D) norm elevation( Dey) from (-L,0) to (L,0)        { ⇨ Charge }
END
```

Before running this descriptor we click on *Controls, Domain Review* on the top ribbon. As a result we quickly obtain the following sketch of the geometry.

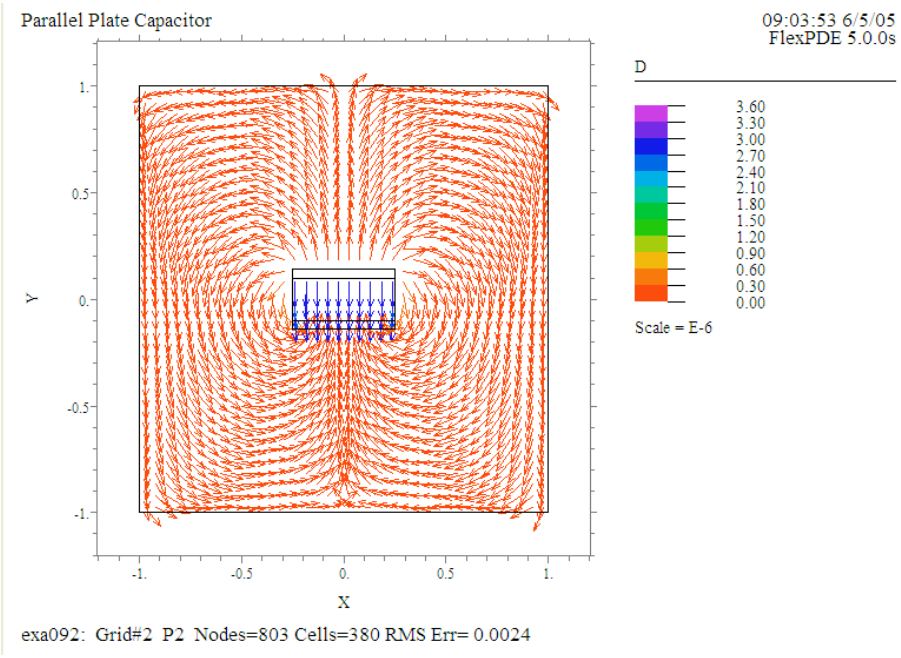


On running the descriptor we quickly obtain the following contour plot of U. The contour lines appear to be highly parallel in the dielectric between the plates, and our boundary conditions force the contours to meet the outer boundary at right angles.



The following vector plot of the field D demonstrates that the boundary conditions constrain the field lines to remain within the domain defined, which means that all lines from positive charges return to the corresponding negative charges on the other plate. A few arrows may appear to penetrate the outer boundary, but in fact they

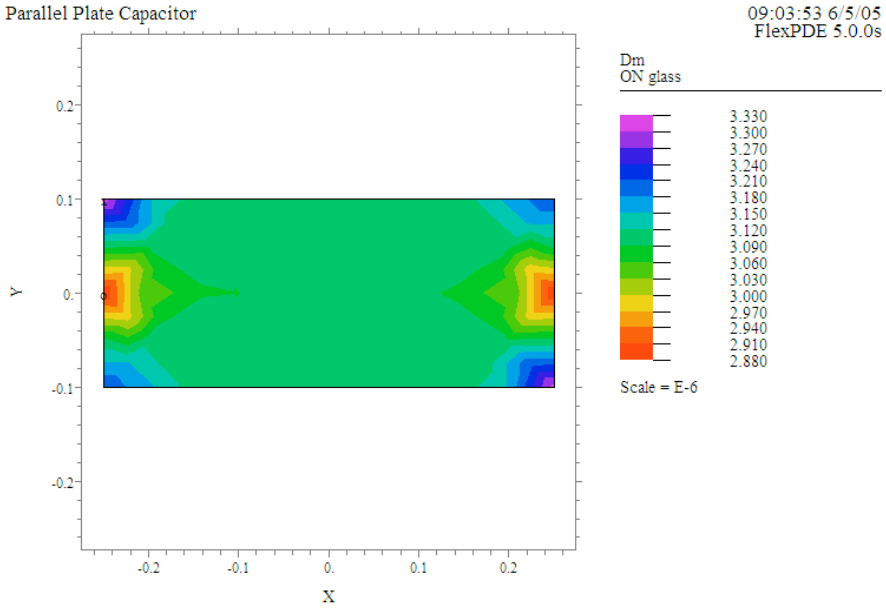
indicate the direction of the field at the *base* of the arrow, not at the tip. We conclude that the charge is positive everywhere on the upper plate, and negative all over the lower one.



The above figure also illustrates how the far-field lines become flattened by the arbitrary rectangular limits we have imposed to simplify the problem. We could expect this to influence the central part of the field to some extent, to be explored later.

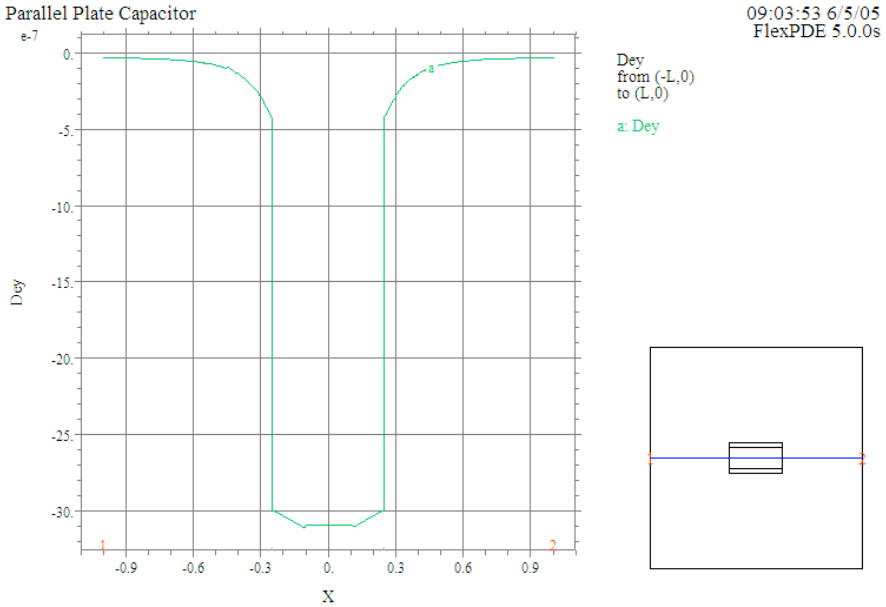
As seen in the logarithmic contour plot of Dm there are no lines in the interior of the capacitor, which suggests that the field there is nearly constant. That plot also shows that the field strength at the outer boundary is only about 1% of that in between the plates.

We further illustrate the constancy of the field between the plates by the following *painted* contour plot over the region of the dielectric material only.



exa092: Grid#2 P2 Nodes=803 Cells=380 RMS Err= 0.0024
Integral= 3.096698e-7

The field **D** leaves or enters the metal plates in the direction of the normal, and according to Gauss' law we have $D_n = |\mathbf{D}| = \sigma_s$. Let us apply this relation to a box including the lower plate and consisting of the lower half of the boundary and the horizontal plane of symmetry. Only the latter gives non-zero contribution to the integral of D_n. The integral given on the elevation plot below thus yields the charge on the lower plate.



exa092: Grid#2 P2 Nodes=803 Cells=380 RMS Err= 0.0024
Integral= -1.676233e-6

Evidently, the charge on each of the plates has the magnitude 1.676e-6, per unit extension in the *z* direction. If we divide the total charge on one plate by the voltage applied (1.0e4), we have the capacitance *C*=1.676e-10. The elementary estimate for the capacitance yields 7.0*eps0*(2*xx)/(2*d0)=1.55e-10, which is smaller because it does not take the fringing field into account.

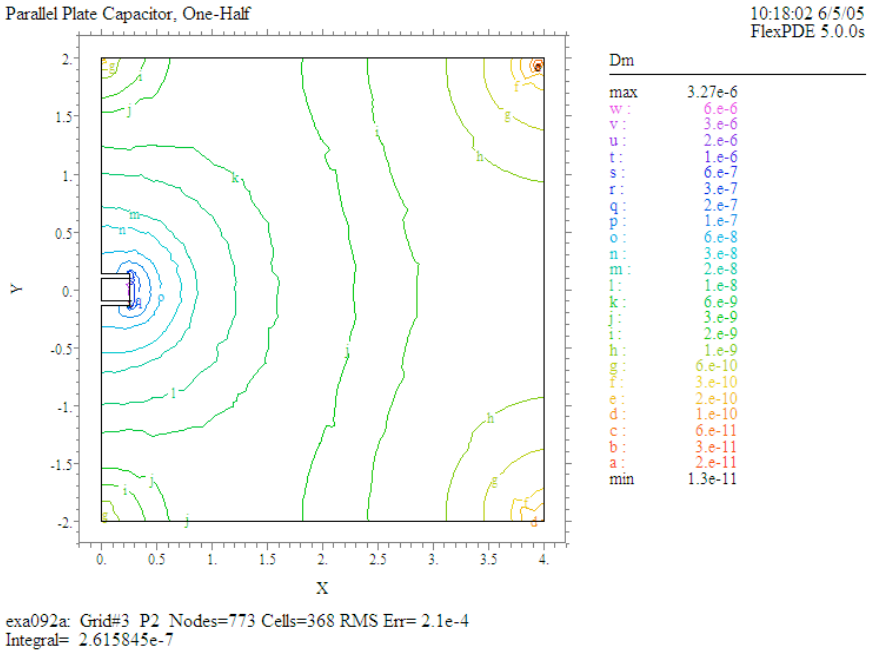
The constraints at the outer boundary may be expected to influence the central field. Using the Professional Version we may easily find out how much the capacitance (given by the integral) changes if we increase the size of the domain by putting L=2, then L=4, and L=8. We find the values 1.685e-6, 1.687e-6 and 1.687e-6, respectively. This indicates that the last value is close to what we would find for infinite space.

Exploiting Symmetry

The Student Version did not permit us to double the value of L in the preceding example, but we may improve conditions by exploiting the left-right symmetry. The following file, based on *exa092*, defines *half* the domain with L=2. Since the metal plates now extend to the boundary, we draw the plate boundaries as parts of a single outline. The line of symmetry must of course obey the condition natural(U)=0.

```
TITLE   'Parallel Plate Capacitor, Half-Domain'              { exa092a.pde }
…
  L=2    xx=0.25    d0=0.1    dd0=0.4*d0     U0=1.0e4    { Voltage }
…
region 'domain'
  start 'outer' (0,-d0-dd0)   natural(U)=0                  { Outer boundary }
  line to (0,-L) to (2*L,-L)  to (2*L,L) to (0,L) to (0,d0+dd0)
  value(U)=U0  line to (xx,d0+dd0) to (xx,d0) to (0,d0)      { Upper plate }
  natural(U)=0  line to (0,-d0)
  value(U)=0  line to (xx,-d0) to (xx,-d0-dd0) close    { Lower plate }
region 'glass'   eps=7.0*eps0                                { Dielectric }
  start(0,-d0)  line to (xx,-d0)  to (xx,d0) to (0,d0) close
PLOTS
  contour( U)     contour( Dm) log     contour( Dm) painted on 'glass'
  vector( D) norm elevation( Dey) from (0,0) to (2*L,0)      { ⇨ Charge }
END
```

The result for the logarithmic plot of Dm is shown below.



The elevation plot reproduces the charge and capacitance values (divided by 2) that we quoted from the Professional Version.

Glass Rod across a Parallel Field

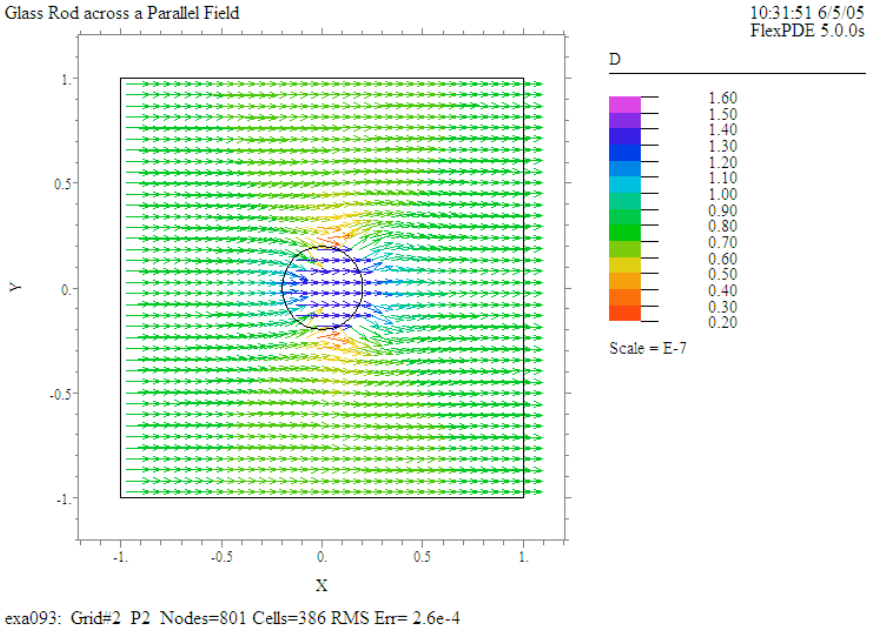
We now turn our attention to a glass rod of circular cross-section, exposed to an external field. We assume the field to have been homogeneous before the rod was placed perpendicularly to it. The file *exa092* is a suitable template for the following descriptor.

```
TITLE   'Glass Rod across a Parallel Field'                    { exa093.pde }
SELECT    errlim=3e-5    ngrid=1    spectral_colors
VARIABLES   U
DEFINITIONS
  L=1    r0=0.2    eps0=8.854e-12    epsr2=7.0      { Glass }
  eps                                               { Declared only }
  Ex0=1e4                                           { Far Field }
  Ex=-dx(U)    Ey=-dy(U)    E=-grad(U)    Em=magnitude(E)
  Dex=eps*Ex    Dey=eps*Ey    D=eps*E    Dm=magnitude(D)
EQUATIONS
  div( D)=0                                          { div( eps*(-grad( U))=0 }
BOUNDARIES
```

```
region 'domain'    eps=eps0                                    { Vacuum outside }
  start 'outer' (-L,-L) natural(U)=0  line to (0,-L) point value(U)=0
  line to (L,-L) natural(U)=eps*Ex0   line to (L,L)             { See p.51 }
  natural(U)=0   line to (-L,L)    natural(U)=-eps*Ex0   line to close
region 'glass'   eps=epsr2*eps0
  start 'circle' (r0,0) arc(center=0,0)   angle=360
PLOTS
  contour( U)     elevation( U) on 'outer'
  elevation( normal( D)) on 'outer' report( eps0*Ex0)
  vector( D) norm     vector( D) on 'glass'
  elevation( tangential( E)) on 'circle' on 'domain'
  elevation( tangential( E)) on 'circle' on 'glass'
  elevation( normal( D)) on 'circle' on 'domain'
  elevation( normal( D)) on 'circle' on 'glass'
END
```

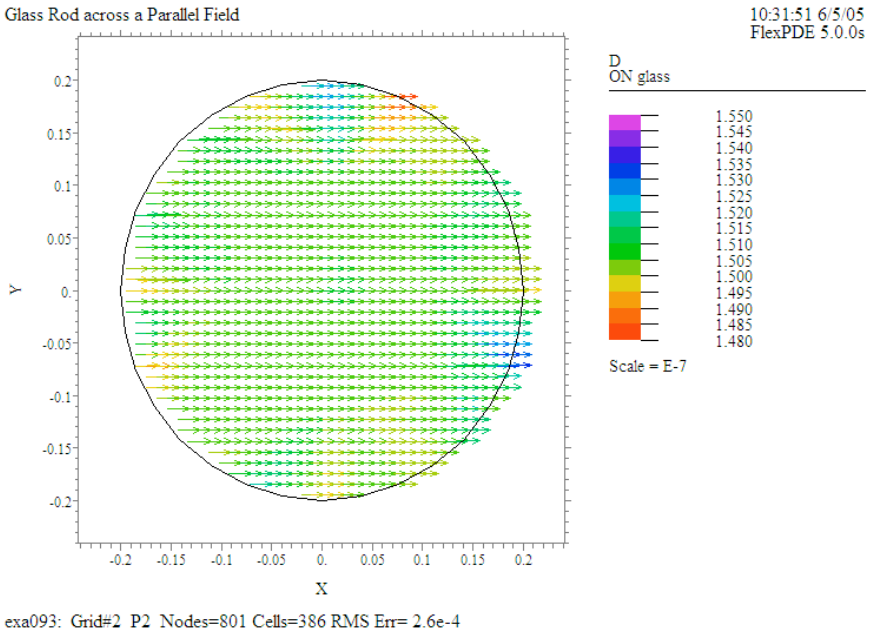
In a previous chapter (p.51) we used the statement natural(U)=0 to specify that the normal component of grad(U) be zero. In the above problem we want to specify non-zero values for the field on the left and right ends. In the PDE, U is effectively multiplied by eps and hence we also have to multiply (p.51) by eps in the natural boundary condition.

On the left boundary the sign of D_n must be negative, since the outward normal to the domain is opposite to the direction of the field. The vector plot below illustrates this.



We may verify that the derivative boundary conditions have taken effect by the elevation plot of normal(D). It is clear that D_n alternates between zero and the specified value for Dx0. In order to make this comparison more convenient, we use the *report* facility. This displays the requested numeric value Dx0 at the bottom of the plot.

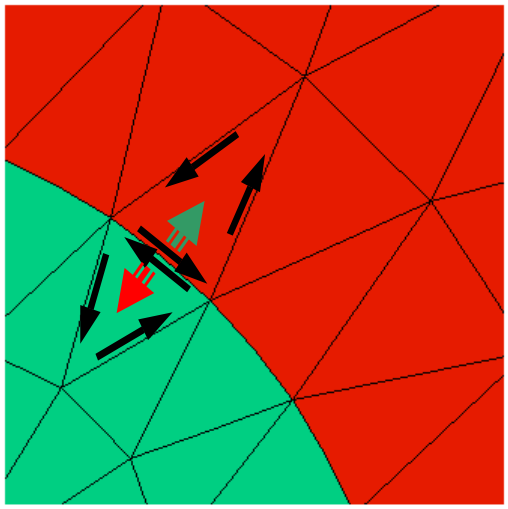
It is remarkable that the field inside the glass rod becomes parallel. We see this more clearly in the following vector plot 'on glass', which also demonstrates the constancy of Dm.



For several elevation plots we have used a new, twofold specification. A plot on 'outer' is unique, because no data are available outside the boundary. For a plot on an interface between glass and vacuum, however, we must specify if we mean to use data on one side or the other of the circle. This we achieve by adding the name of the region.

Here, we should pay attention to the way a *region* is named. Under *boundaries* we first define 'domain' by drawing the outer boundary. Then we define 'glass' as part of that domain. By this means we have indirectly defined 'domain' as being the *vacuum* region. This change of definition may be confusing, but it has the advantage that we need not define three regions: one for the total domain, one for the vacuum and one for the glass.

When plotting normal or tangential components we need to understand how to interpret *signs*. The following figure shows a few cells at an interface between two materials. The black arrows indicate the counter-clockwise sense in which the cell boundaries are traced. Evidently, the common side is traced in opposite directions, and this is the reason why we obtain opposite signs for the tangential components. The normal vector always points *out* of a cell, and this means that normal components also have opposite signs.



If we reverse the signs in the plots 'on domain', and then make hardcopies, we may compare the corresponding plots accurately by superimposing the figures with a back-light. These plots then confirm the theoretical findings that the tangential component of **E** and the normal component of **D** are continuous on going through the interface.

Surface Charge of Polarization

An external electric field causes polarization of atoms and molecules, which may produce excess local charge, at least on the interfaces between different materials. The dipole moment per unit volume, **P**, may be written

$$\mathbf{P} = \mathbf{D} - \varepsilon_0 \mathbf{E} = \mathbf{D} - \varepsilon_0 \mathbf{D}/\varepsilon = (1 - \varepsilon_0/\varepsilon)\,\mathbf{D} \qquad \bullet$$

The volume charge density ρ_p due to polarization is given by[3p107]

$$\rho_p = -\nabla \cdot \mathbf{P} = -\nabla \cdot (1 - \varepsilon_0 / \varepsilon)\, \mathbf{D} \qquad \bullet$$

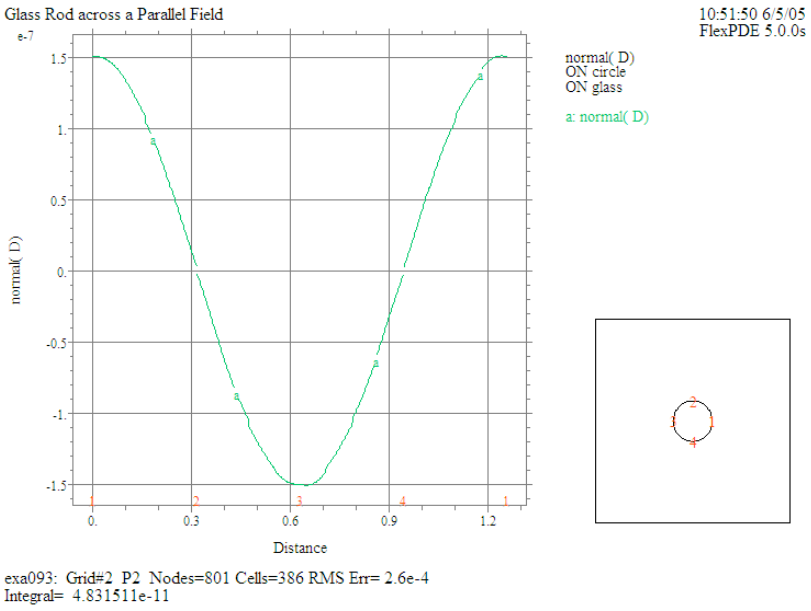
Thus, in any region of constant ε this volume charge will vanish, because $\nabla \cdot \mathbf{D}$ is zero in the absence of free charge (ρ_v). If the electric permittivity ε should vary continuously in space, however, ρ_p may become non-zero.

In this text, we shall not be concerned with continuously varying electric permittivity, but we have already treated *sudden* changes from one material to another. In such situations, polarization charge appears at the interface between materials of different ε. The resulting surface charge density is given by the normal component of **P**, i.e.

$$\sigma_{ps} = \mathbf{P} \cdot \mathbf{n} \equiv P_n = (1 - \varepsilon_0 / \varepsilon)\, D_n \qquad \bullet$$

We may define **P** and calculate the polarization charge density on the rod in *exa093* by an elevation plot, as follows.

```
TITLE   'Glass Rod across a Parallel Field, Polarization'  { exa093a.pde }
...
   P= (1- eps0/eps)*D
EQUATIONS
...
   elevation( normal(P)) on 'circle' on 'glass'
      as 'polarization surface charge'
END
```



From the curve above we see that the polarization charge at the right edge of the rod is positive, in accord with elementary notions. The polarization charges reside in the glass region, not in the vacuum, and according to the rules of the software we thus specify on 'circle' on 'glass'.

It is instructive to include a corresponding plot yielding the polarization charge on the vacuum side of the rod.

Exercises

❑ How will the fields change if you replace the glass in *exa092* by a vacuum? By a dielectric with very high permittivity, say $\varepsilon = 1000\varepsilon_0$?

❑ Make an elevation plot that will permit you to calculate the charge on an interior face of the capacitor in *exa092*. Compare it to the remainder of the charge on that plate.

❑ Modify *exa093* to show that a rod with very large permittivity simulates a metallic cylinder. Use epsr2=1e6, say. Add elevation plots of tangential(D) on the faces on 'circle' on 'domain' and on 'circle' on 'glass'. Compare to the corresponding plots of normal(D) to find information about the field direction. Also compare with the last vector plot of **D**.

❑ Replace the object in *exa093* by a bar of square cross-section with the sides equal to 2*r0.

❑ Change the permittivity in *exa091* to $\varepsilon = 2\varepsilon_0 R/r_1$, illustrating continuously varying materials properties.

10 Steady Fields in (ρ,z) Space

Although recent versions of the software do permit us to handle problems in three spatial dimensions, it is more difficult in 3D than in 2D to obtain and assimilate graphical presentations of the solutions. If the geometry, the materials and the boundary conditions are axially symmetric, however, all cross-sections through the axis of symmetry are equivalent. This permits us to solve a 3D problem in cylindrical coordinates (ρ,z) and present the results in a plane. We may then make an image of the geometry and the plots by mentally rotating the figures around the axis.

Electrical Conduction in a Cone

Our first example of an axially symmetric object is a truncated, circular cone of silicon with metallic electrodes of high conductivity connected to the flat ends, so that we may consider the ends to have constant potential. A potential difference is applied, and we wish to calculate the current distribution and the resistance between the ends. The following descriptor specifies that problem.

In the *coordinates* segment we express our intention to operate in (ρ,z) space (p.44). This permits us to type the PDE in the appropriate variables, and in addition we obtain suitable labels for the figures.

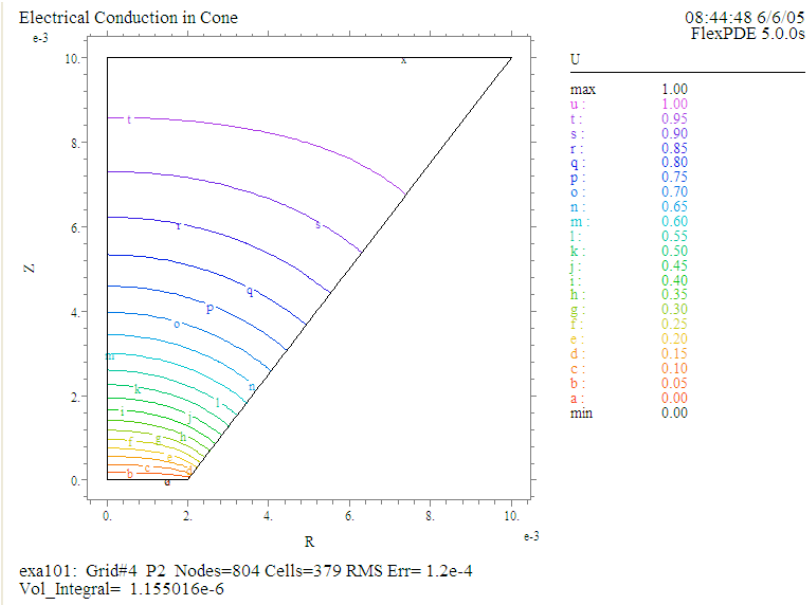
In (ρ,z) coordinates, the gradient expression becomes similar[3p705] to that in (x,y), but the divergence takes a different form, i.e.

$$\nabla\cdot\mathbf{f}=\frac{1}{\rho}\frac{\partial(\rho f_\rho)}{\partial\rho}+\frac{\partial f_z}{\partial z} \qquad \bullet$$

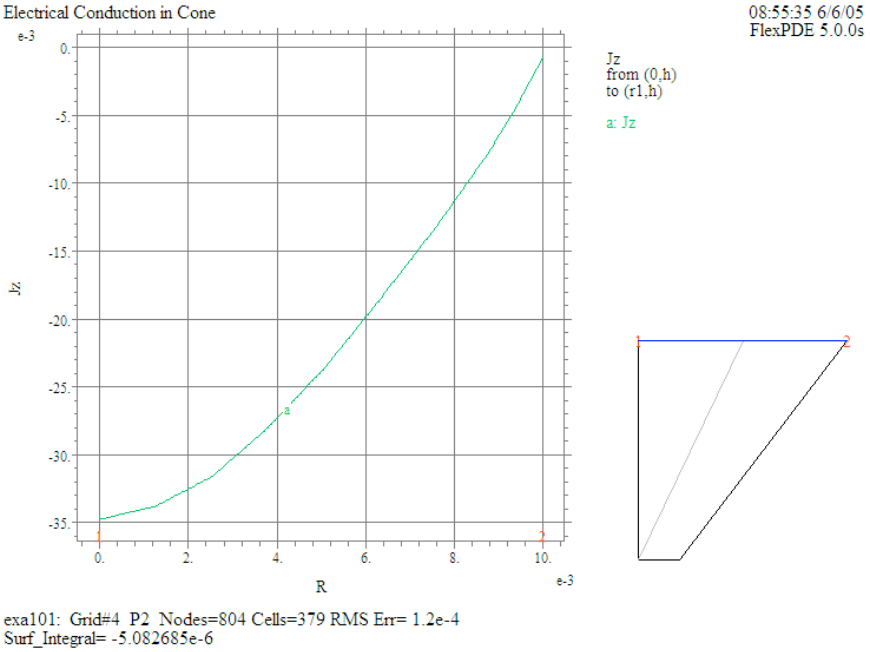
In the case of electrical conduction we have $\mathbf{f}=\mathbf{J}$, as is apparent from the *equations* segment.

```
TITLE   'Electrical Conduction in Cone'                    { exa101.pde }
SELECT    errlim=3e-5    ngrid=1    spectral_colors
COORDINATES   ycylinder( 'r', 'z')                        { Student Version }
VARIABLES   U
DEFINITIONS
  r0=2e-3    r1=10e-3    h=10e-3    cond=1.0e-3    { Silicon }
  Er=-dr(U)    Ez=-dz(U)    E=vector( Er, Ez)    Em=magnitude(E)
  Jr=cond*Er    Jz=cond*Ez    J=vector( Jr, Jz)    Jm=magnitude(J)
EQUATIONS
  (1/r)*dr( r*Jr)+ dz( Jz)=0                              { div(J)=0 }
BOUNDARIES
region 'silicon'  start 'outer' (0,0)
  value(U)=0.0 line to (r0,0)
  natural(U)=0.0   line to (r1,h)
  value(U)=1.0 line to (0,h) natural(U)=0   line to close
feature
  start 'line' (r1/2, h) line to (0,0)
PLOTS
  contour( U)    surface( U)    contour( Em) log    vector( J) norm
  elevation( Jz) surf_integrate from (0,0) to (r0,0)   { ⇨ Current }
  elevation( Jz) surf_integrate from (0,h) to (r1,h)   { ⇨ Current }
  contour( Jm*Em) painted vol_integrate                { ⇨ Dissipation }
  elevation( tangential( E)) line_integrate on 'line'  { ⇨ Voltage }
END
```

The following contour plot of U suggests that the axially symmetric surfaces of constant potential are nearly spherical, except close to the plane ends.



In order to calculate the current crossing a horizontal surface one needs to integrate Jz over a circular area consisting of annular elements of size $2\pi\rho\, d\rho$. The elevation plot shown below yields the current as *surf_integral*. In order to estimate the accuracy of the solution, we compare the current values obtained by integrating over the end surfaces.



The contour plot of the power dissipated ($\mathbf{E}\cdot\mathbf{J}$) evidently integrates to be about equal to the product of the voltage drop (1.0) and the current, as we would expect. The modifier *vol_integrate* makes FlexPDE include the volume element $2\pi\rho\, d\rho\, dz$ in the calculation.

The last plot presents the tangential component of the field along the sloping 'line'. The integral, which involves the length element *dl*, confirms the voltage value we imposed. The modifier *line_integrate* assures that we obtain a line integral, rather than a surface integral over a cone.

Plate Capacitor

In the previous chapter we studied a plate capacitor in (x, y), consisting of long metal slats extending far in the z directions. We may now apply similar calculations to a capacitor having two *circular* plates, using the following descriptor.

```
TITLE    'Plate Capacitor in (r,z) Space'                    { exa102.pde }
SELECT    errlim=3e-4    ngrid=1    spectral_colors
COORDINATES    ycylinder( 'r', 'z')
VARIABLES   U
DEFINITIONS                          { Thickness of glass plate= 2*d0 }
  Lz=1.0    Lr=2*Lz    r0=0.2    d0=0.1    dd=0.4*d0    U0= 1e4
  eps0=8.854e-12    eps
  Er=-dr(U)    Ez=-dz(U)    E=vector( Er, Ez)    Em=magnitude(E)
  Der=eps*Er    Dez=eps*Ez    D=eps*E    Dm=magnitude(D)
EQUATIONS
  (1/r)*dr( r*Der)+ dz( Dez)=0                                { div(D)=0 }
BOUNDARIES
region 'domain'  eps=eps0
  start 'outer' (0,-d0-dd)  natural(U)=0
  line to (0,-Lz) to (Lr,-Lz) to (Lr,Lz)  to (0,Lz)  to (0,d0+dd)
  value(U)=U0   line to (r0,d0+dd) to (r0,d0) to (0,d0)
  natural(U)=0   line to (0,-d0)
  value(U)=0  line to (r0,-d0) to (r0,-d0-dd) close
region 'glass'  eps=7.0*eps0
  start(0,-d0)  line to (r0,-d0) to (r0,d0) to (0,d0) close
feature
  start 'upper'  (0,d0+dd)  line to (r0,d0+dd) to (r0,d0) to  (0,d0)
PLOTS
  contour( U)    contour( Dm) log    vector( D) norm
  surface( Dm) log    contour( Dm) on 'glass' painted
  elevation( -normal(D)) on 'upper'                           { ⇨ Charge }
    report( pi*r0^2* [7.0*eps0*U0/(2*d0)]) as 'Q_elementary'
  elevation( -Dez) from (0,0) to (Lr,0)                       { ⇨ Charge }
END
```

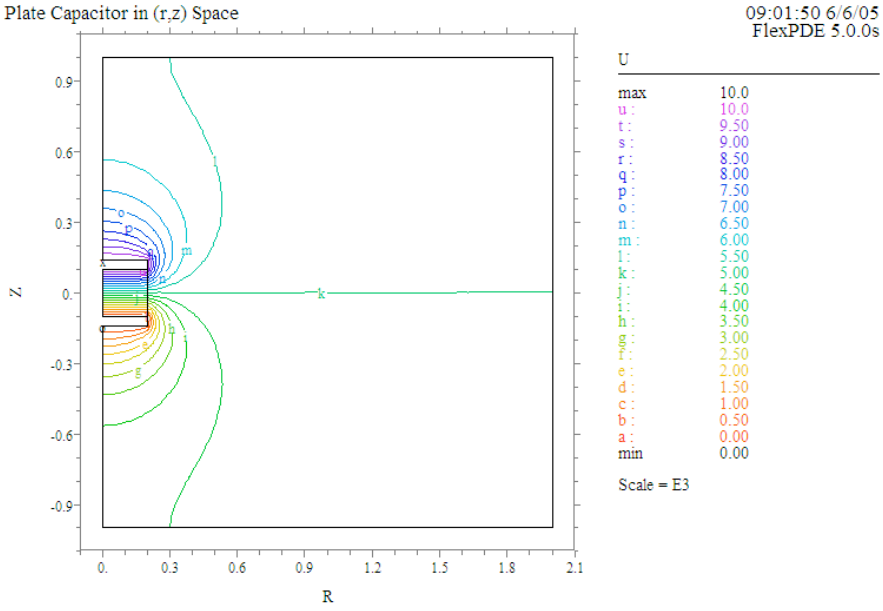
After typing this descriptor, you may click on *Controls, Domain Review* above the editor window to obtain a sketch of the geometry. The capacitor plates and the glass slab are circular, and we may imagine generating the 3D geometry by rotating the figure about the axis of symmetry.

In the descriptor we draw a single outer boundary for the vacuum and for the plates, changing from derivative to value boundary conditions where necessary.

We start on the axis of symmetry at the bottom surface of the lower plate and trace the boundary in the counter-clockwise sense until we reach the top surface of the upper plate.

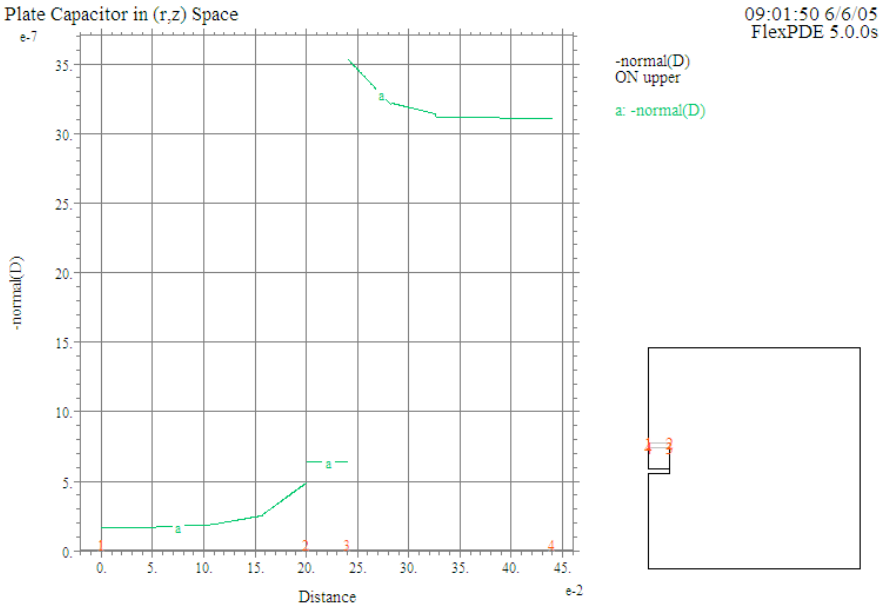
On the outer horizontal lines, and the vertical line on the right side, we need a vanishing normal derivative to confine the field completely in the domain. On this part of the boundary, $\partial U/\partial n$ is thus specified to be zero. On the axis there can be no radial field (by symmetry), which means that the same derivative boundary condition applies.

In this problem we are considering the field in a cross-section passing through the *z*-axis, which is vertical in the plots. The surfaces of constant potential cut the plane of the figure along the curves in the plot below. In other words, you may think of the potential surfaces below as being generated by rotating these contours around the vertical axis.



The logarithmic plot of Dm indicates that the field strength at the upper, lower and right boundaries is about 0.1 % of that between the plates, and hence the constraints on the field at the boundaries should not seriously perturb the charge stored.

The following elevation plot on the *feature* named 'upper' displays the flux of D_n leaving the upper capacitor plate. The program always gives us the normal component pointing out of the *domain*. Hence we need a negative sign in the plot argument in order to obtain the charge on the plate.



For the integration of D_n, we must use annular area elements $(2\pi\rho d\rho)$ for the surface of the plate. This mode is actually the default when cylindrical coordinates have been declared. From the charge so calculated and the applied voltage (1e4) you directly obtain the capacitance.

In order to compare the charge values obtained by integration to that of the elementary capacitor formula, we have added a line for estimating (pi*r0^2)*Dez inside the glass. The *report* command makes the numerical value appear at the bottom of the plot. We note that this simplistic treatment underestimates the capacitance by about 20%.

The last elevation plot of Dez along the horizontal plane of symmetry also yields the charge, since all of the field lines must pass by this plane. This integral is expected to be more accurate, but it does not distinguish between the contributions from various parts of the metal plate.

Glass Ball in a Parallel Field

This application concerns a dielectric ball in an (initially) parallel field of strength E0. As an alternative strategy to that on p.84, we create the applied field by imposing potentials -U0 and U0 on the horizontal boundaries as shown below.

With this descriptor we also introduce short notations for grad and div in (ρ,z). As already mentioned, FlexPDE contains a symbolic, auxiliary program that automatically expands the differential operators in the coordinates declared.

```
TITLE   'Glass Ball in a Parallel Field'                    { exa103.pde }
SELECT    errlim=3e-6    ngrid=1    spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES   U
DEFINITIONS
  Lz=1.0    Lr=2*Lz    r0=0.05    rad=sqrt(r^2+z^2)
  eps0=8.854e-12    epsr1=1.0    epsr2=7.0
  epsr    eps=epsr*eps0    U0=1.0    U_ex    { Declared only }
  E0=-2*U0/(2*Lz)    De0=eps0*E0                  { Far field }
  E=-grad(U)    Em=magnitude(E)    D=eps*E    Dm=magnitude(D)
Dm_ex0=abs[ eps0*epsr2*3/(2+epsr2)*E0]          { Dm inside ball }
EQUATIONS
  div( D)=0                        { Expanded automatically for (r,z) }
BOUNDARIES
region 'domain' epsr=epsr1                                  { Vacuum }
  U_ex=-[1+(1-epsr2)/(2+epsr2)*( r0/rad)^3]*E0*z    { Analytic 1 }
  start 'outer' (0,-Lz)   value(U)= -U0 line to (Lr,-Lz)
  natural(U)=0   line to (Lr,Lz)
  value(U)=U0   line to (0,Lz)
  natural(U)=0   line to close
region 'glass'  epsr=epsr2                                  { Glass }
  U_ex=-3/(2+epsr2)*E0*z          { Analytic solution, yields Dm_ex0 }
  start (0,r0) line to (0,-r0)  arc to (r0,0) close
feature  start  'circle'  (0,-r0) arc to (r0,0) to (0,r0)
PLOTS
  contour( U)    contour( U-U_ex) painted    elevation( U) on 'outer'
  elevation( normal( D)) on 'outer' report( De0)
  vector( D) norm    vector( D) on 'glass' report(Dm_ex0)
  elevation( normal(D), tangential(D)) on 'circle' on 'glass'
  elevation( -normal(D), -tangential(D)) on 'circle' on 'domain'
END
```

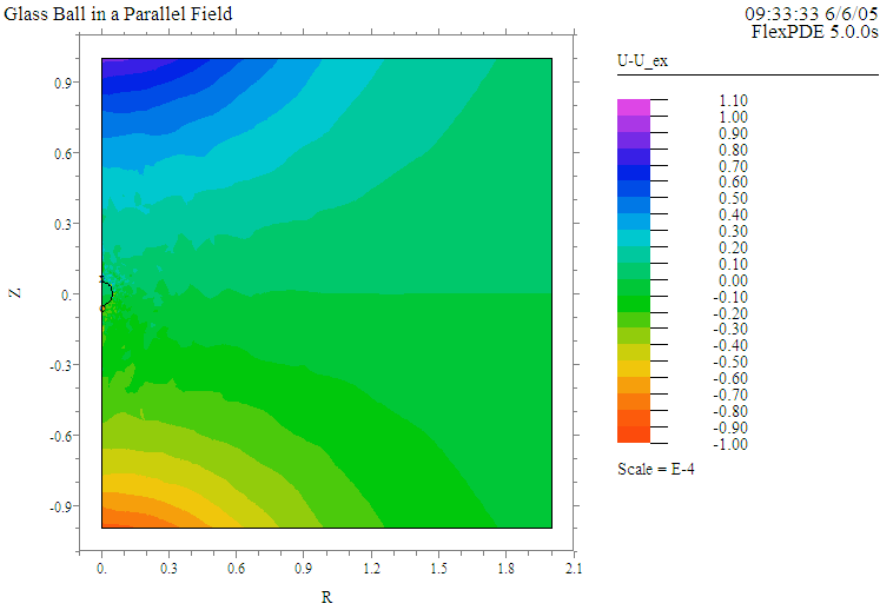
There is an exact solution to a similar problem[3p191], i.e. that of a dielectric sphere in a field which becomes parallel and of constant strength as $z \to \pm\infty$. The boundary conditions on the sphere, where two analytic solutions meet, lead to the following expressions for the potentials (with $R \equiv \sqrt{\rho^2 + z^2}$).

$$U_1 = -E_0 z\left(1 + \frac{\varepsilon_{r1} - \varepsilon_{r2}}{2\varepsilon_{r1} + \varepsilon_{r2}}(r_0 / R)^3\right) \qquad \text{(outside ball, } \varepsilon_r = \varepsilon_{r1}\text{)}$$

$$U_2 = -\frac{3\varepsilon_{r1}}{2\varepsilon_{r1} + \varepsilon_{r2}} E_0 z \qquad \text{(inside ball, } \varepsilon_r = \varepsilon_{r2}\text{)}$$

We type the two solutions for U_ex into their respective *region* segments. At large distances from the sphere, the outside field rapidly approaches a linear expression in *z*, corresponding to a homogeneous field.

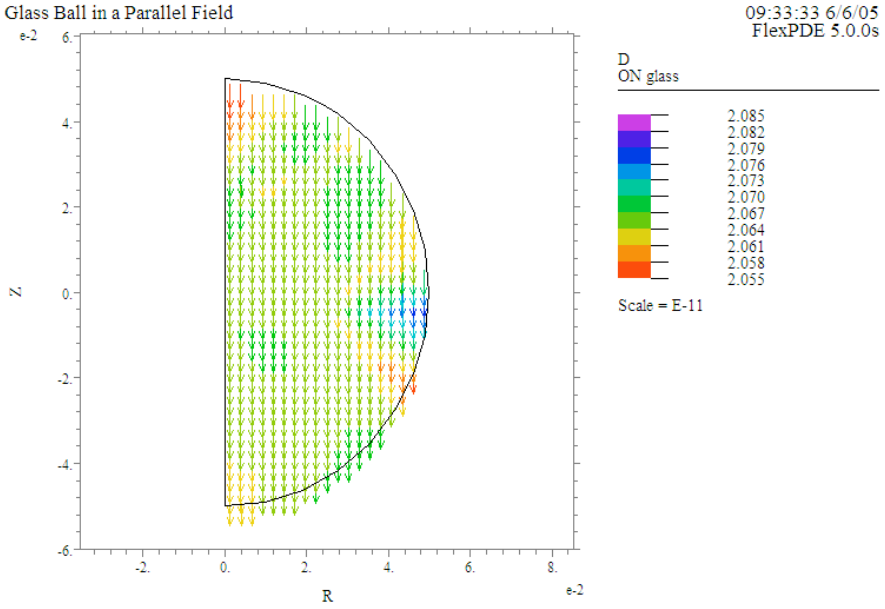
The following figure is a painted contour plot of the deviation of the numeric potential U from the analytic one.



The maximum relative deviation evidently is about 0.01 % of the total span. The difference between the numeric and the analytic solutions cannot be reduced significantly by a smaller errlim, since it is caused by dissimilar boundary conditions. If we were to use the

Professional Version with larger values Lz, however, we would obtain better agreement.

The following zoomed vector plot of **D** shows that the field is parallel inside the sphere, in agreement with the analytic solution.



Here we can also compare with the reported analytic value. Evidently, the agreement is satisfactory on the average.

Charged, Conducting Ellipsoid

In previous examples, the total charge within the PDE domain was zero. We can also handle the case of an enclosed charge Q, however, by imposing a normal component D_n on the outer boundary. To do this, we simply assume that any charged object will generate a field similar to that of the same amount of charge concentrated at the center. The two fields may be expected to become similar at large distances from the center.

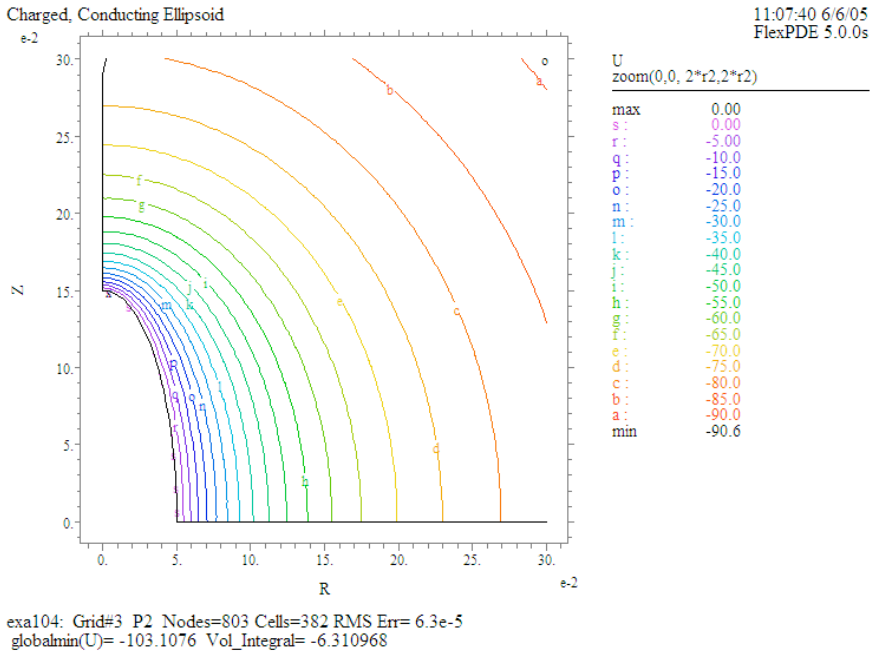
```
TITLE   'Charged, Conducting Ellipsoid'                    { exa104.pde }
SELECT    errlim=1e-4    ngrid=1    spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES   U
DEFINITIONS
```

```
  r1=0.05     r2=0.15     r3=1.0
  eps=8.854e-12     Q=1e-9                                  { Specified charge }
  E= -grad(U)     Em=magnitude(E)     D=eps*E     Dm=magnitude(D)
  Up=Q/(4*pi*eps*sqrt(r^2+z^2))            { Potential of a point charge }
  Ep=-grad(Up)     Dp=eps*Ep     Dpm=magnitude(Dp)
EQUATIONS
  div( D)=0
BOUNDARIES
region 'domain'
  start (r1,0)  natural(U)=0  line to (r3,0)
  natural(U)=Dpm    arc(center=0,0) to (0,r3)
  natural(U)=0  line to (0,r2)  value(U)=0
  arc(center=0,0) close                                    { Quarter ellipse }
feature
  start 'ellipsoid' (0,r2) arc(center=0,0) to (r1,0)
  start 'outer' (r3,0) arc(center=0,0) to (0,r3)
PLOTS
  contour(U) zoom(0,0, 2*r2,2*r2) report( globalmin(U))
  elevation(U) from (r1,0) to (r3,0)
  surface( Dm) zoom(0,0, 2*r2,2*r2)   vector( D) norm
  elevation( Dpm, normal(D)) on 'outer'                    { Test of Dn }
  elevation( 2*normal(D)) on 'outer'                       { ⇨ Charge }
  elevation( -2*normal(D)) on 'ellipsoid'                  { ⇨ Charge }
END
```

Here, we have included the expression Up for the potential of a point charge Q at the origin. Since we have chosen a spherical outer boundary, the resulting field **D** will be perpendicular to the boundary and have a magnitude equal to Dpm. When applying the natural boundary condition, however, we must remember that U, hidden in the PDE, is effectively multiplied by eps. In view of p.51 we must thus specify the normal component of **D**, rather than that of **E**.

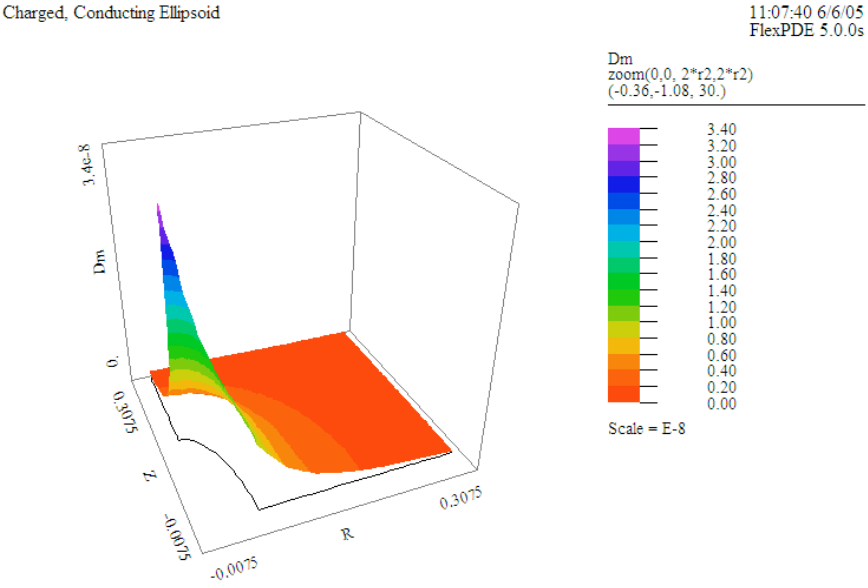
The descriptor is valid for one-half of the total space, which is the reason why we must multiply the integrand by 2 when calculating the charge.

The figure below shows the resulting contours of U, all with negative values. The potential on the conductor itself is of course zero, as specified. Since this plot is zoomed we cannot read off the minimum value U, but we have reported globalmin(U) at the bottom of the figure.



exa104: Grid#3 P2 Nodes=803 Cells=382 RMS Err= 6.3e-5
globalmin(U)= -103.1076 Vol_Integral= -6.310968

The following zoomed surface plot of Dm presents the field strength. We note that the field is much stronger where the radius of curvature is smaller, which illustrates the fact that sparks in the air start from tips.



exa104: Grid#3 P2 Nodes=803 Cells=382 RMS Err= 6.3e-5
Vol_Integral= 1.296579e-10

In addition, this plot depicts the distribution of charge density on the ellipsoid, since Dm on the surface of the conducting object is equal to the normal component Dn.

By the first elevation plot we compare the field due to the point charge with that due to the charged ellipsoid.

By the final elevation plots we compute the charge on the ellipsoid using the Gauss law on the spherical boundary and then on the ellipsoid itself.

The elevation plot of U suggests that the potential could vary roughly as $1/\rho$, and we would be interested in the asymptotic limit for large radii. If we repeat the run using an outer radius of r3=10.0 we find that the first plot reports $U_{\min} = -111.0$. The capacitance of the ellipsoid with respect to infinity may thus be estimated as $Q/U = 1e-9/111 = 9.0e-12$.

After testing ellipsoids of different elongation we discover that the capacitance is rather well approximated by that of a ball with a radius equal to the geometric mean of the half-axes.

Spherical Capacitor

Taking r1=r2 in *exa104* we obtain a spherical object, for which there is a well-known elementary expression for the capacitance. To make this comparison we need to add two *report* statements.

```
TITLE   'Charged, Conducting Sphere'                    { exa104a.pde }
…
DEFINITIONS
  r1=0.1     r2=r1     r3=1
...
  Qe=surf_integral( -2*normal(D), 'ellipsoid')
EQUATIONS
…
  report( Qe/(abs(globalmin(U))) )  as 'Capacitance' { For last plot }
  report( 4*pi*eps*r3*r1/(r3-r1)) as 'C_analytical'
END
```

In the first of the report commands we divide the charge, calculated by *surf_integral*, by the potential drop obtained from *globalmin*.

Charged Metal Balls in a Dielectric

The cylindrical coordinate system permits us to treat problems with several axially symmetric objects on a common axis. The following descriptor is for two metal spheres, embedded in a dielectric and kept at potentials of zero and U2 respectively.

In the *boundaries* segment we use the point charge approximation to specify the component normal(Dp) caused by an enclosed charge *Q*. Having done that, we are assured that the *sum* of the charges will be *Q*, but we cannot specify the charge on each one of the metal balls. We shall discover, however, that the *voltage* (U2) applied between the balls determines the sharing of the total charge.

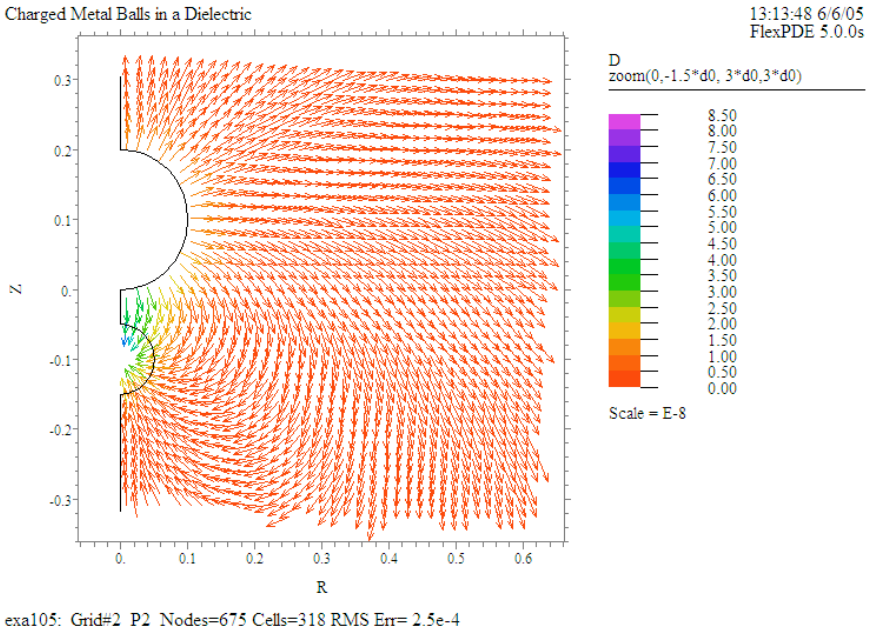
```
TITLE   'Charged Metal Balls in a Dielectric'                    { exa105.pde }
SELECT    errlim=3e-4    ngrid=1    spectral_colors
COORDINATES    ycylinder( 'r', 'z')
VARIABLES   U
DEFINITIONS
  r1=0.05    d0=0.2    r2=0.1    r3=1.0
  eps0=8.854e-12    eps=7.0*eps0        { Glass }
  Q=1e-9    U2=40                       { Voltage controlling charge }
  Er=-dr(U)    Ez=-dz(U)    E=-grad(U)    Em=magnitude(E)
  Der=eps*Er    Dez=eps*Ez    D=eps*E    Dm=magnitude(D)
  Up=Q/(4*pi*eps*sqrt(r^2+z^2))          { Potential of a point charge }
  Ep=-grad(Up)    Dp=eps*Ep              { For outer boundary }
  Q1=surf_integral( -normal(D), 'ball1')              { Charge on 1 }
  Q2=surf_integral( -normal(D), 'ball2')              { Charge on 2 }
EQUATIONS
  div(D)=0
BOUNDARIES
region 'dielectric' start (0,-d0/2-r1)  natural(U)=0  line to (0,-r3)
  natural(U)=normal(Dp) arc( center=0,0) angle=180 to (0,r3)
  natural(U)=0 line to (0,d0/2+r2)  value(U)=U2
  arc(center=0,d0/2) angle=-180                    { Ball2 }
  natural(U)=0  line to (0,-d0/2+r1) value(U)=0
  arc(center=0,-d0/2) angle=-180 close             { Ball1 }
feature
  start 'outer' (0,-r3) arc( center=0,0) angle=180
  start 'ball1'  (0,-d0/2+r1) arc( center=0,-d0/2) angle=-180
  start 'ball2' (0,d0/2+r2) arc( center=0,d0/2) angle=-180
PLOTS
  contour( U)    vector( D) norm zoom(0,-1.5*d0, 3*d0,3*d0)
```

```
  contour( Dm) painted zoom(0,-1.5*d0, 3*d0,3*d0)
  elevation( -normal(D)) on 'ball1'       elevation( -normal(D)) on 'ball2'
  elevation( normal(D)) on 'outer'                              { ⇨ Total charge }
summary
  report(U2) report(Q1) report(Q2) report(Q1+Q2)
END
```

In the *definitions* section of the descriptor we include a *surf_integral* yielding the numerical value of the charge on each ball. The negative sign arises since *normal* gives us the vector component pointing outward from the *domain*, while we need one pointing outward from the *ball*. The integral over the outer boundary, obtained from the elevation plot, should reproduce the sum of those values.

In two of the graphs we employ *zoom*, where the two first arguments are the coordinates of the low-left corner and the last arguments the width and height of the plot window.

The following vector plot shows that there is negative charge on the lower ball and positive charge on the upper one. It also demonstrates that the field is strongest between the two objects and close to the smaller one. At the voltage chosen, the charges are of opposite signs, which means that they will move closer to each other than the distance between the ball centers.



The "plot" named summary just contains a list of values and labels. Here, it reports the charge values and the sum of the charges, to be compared with the value imposed by the boundary condition.

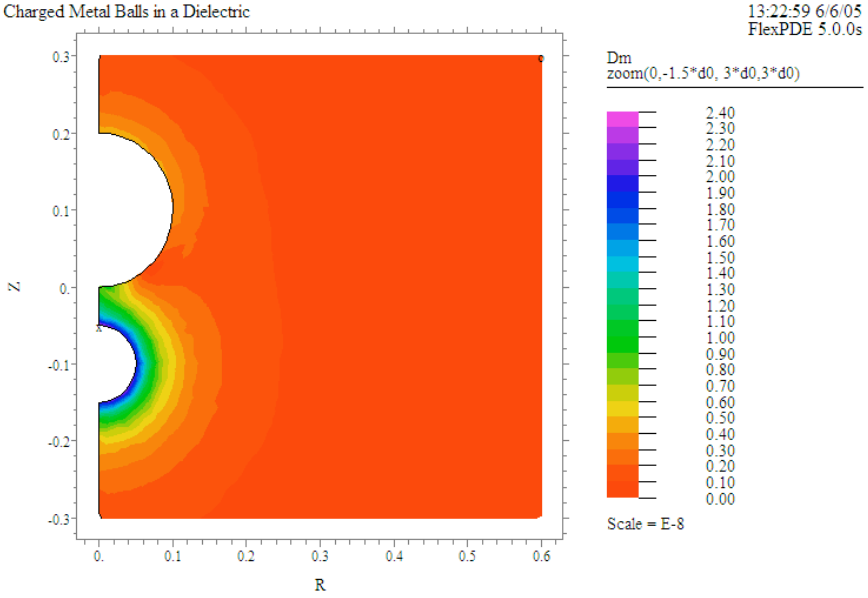
Charged Metal Balls in a Dielectric

13:13:48 6/6/05
FlexPDE 5.0.0s

SUMMARY

U2= 40.00000
Q1= -1.341100e-9
Q2= 2.364533e-9
Q1+Q2= 1.023433e-9

If we change the applied voltage to a *negative* value, **U2=-10**, we obtain the following contour plot of Dm. From the contours around the upper ball, which are shifted upwards, we deduce that the charge distributions now repel each other.



exa105x: Grid#2 P2 Nodes=648 Cells=305 RMS Err= 3.e-4
Vol_Integral= 4.074650e-10

The vector plot shows that the lower ball is positively charged and that the upper ball has both positive and negative charges. These features are confirmed by the elevation plots. The *summary* reports that the total charge on each object is positive and that the charge on the smaller ball is larger.

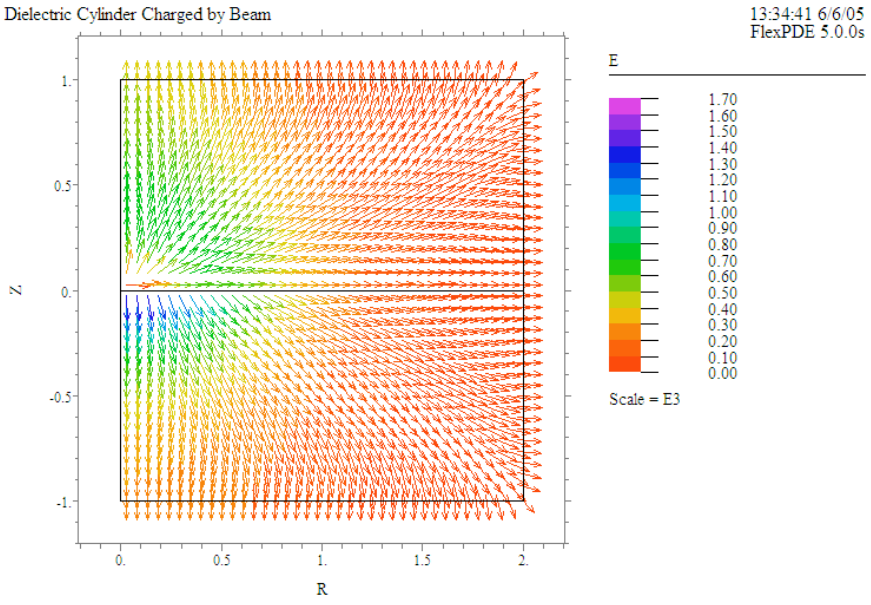
Charged Dielectric

It is difficult to deposit free charges in the interior of a dielectric. One way of doing this is to irradiate the dielectric by a beam of high-energy electrons, penetrating well into the material. Let us assume that the resulting charge density follows a Gauss distribution in a direction transverse to the beam and that it decreases exponentially in the direction of the beam. The specimen is axially symmetric and is completely enclosed in metal as appears from the boundary conditions on the outside. The beam enters from below along the *z*-axis through a thin metal foil.

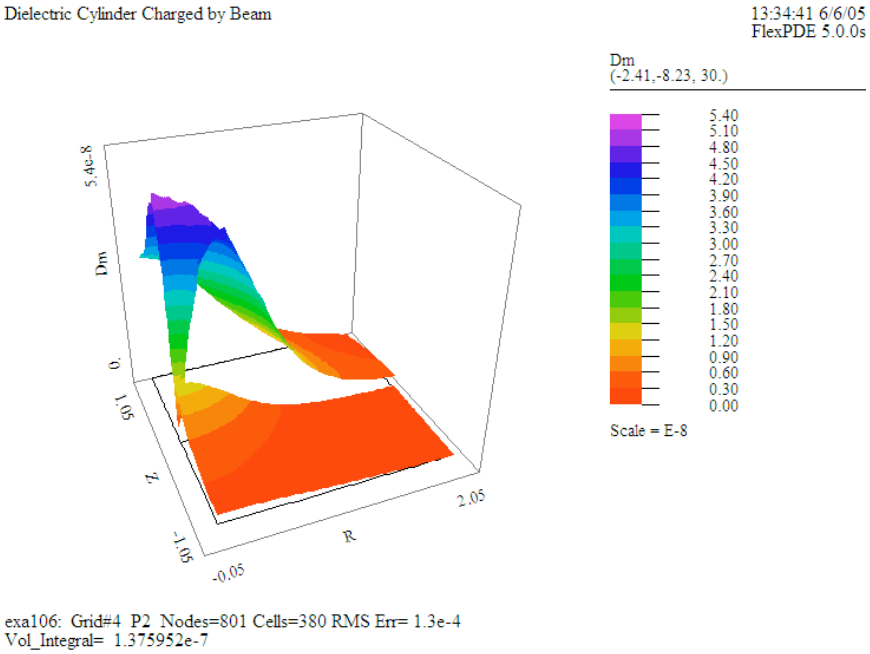
```
TITLE   'Dielectric Cylinder Charged by Beam'                { exa106.pde }
SELECT    errlim=1e-4    ngrid=1    spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES   U
DEFINITIONS
  Lr=2    Lz=1    r0=0.2
  eps0=8.854e-12    epsr    eps=epsr*eps0
  Er=-dr(U)    Ez=-dz(U)    E=-grad(U)    Em=magnitude(E)
  D=eps*E    Dm=magnitude(D)
  rho_v   rho_vg=1e-6*exp(-5*r^2-5*z)          { Charge density in glass }
EQUATIONS
  div( D)=rho_v
BOUNDARIES
region 'domain'  epsr=1.0   rho_v=0              { Air }
  start (0,Lz) natural(U)=0 line to (0,-Lz)
  value(U)=0 line to (Lr,-Lz) to (Lr,Lz) close
region 'glass'  epsr=7.0  rho_v=rho_vg
  start(0,0) line to (Lr,0)  to (Lr,Lz) to (0,Lz) close
PLOTS
  contour( rho_v)    elevation( rho_v) from (0,-Lz) to (0,Lz)
  contour(U)    vector(E) norm    elevation( Ez) from (0,-Lz) to (0,Lz)
  vector( D) norm    surface( Dm)
END
```

Here, we assign different charge densities ρ_v to the two regions of space.

The first plot below shows the vector plot of **E**, exhibiting the highest magnitudes in the air, close to the center of impact of the beam at the origin.



The following surface plot illustrates the variation of Dm over the domain.



Exercises

❑ Introduce cond=1e-3*exp(-300*(r+z)) into *exa101*. Make a contour plot of the conductivity as well.

- ❑ Solve *exa102* assuming a vacuum over the entire domain.
- ❑ Replace the sphere in *exa103* by an ellipsoid with the z-directed axis equal to `1.5*r0`. Also change to `Lr=1`. Will the internal field still be parallel?
- ❑ Exploit the symmetry of the problem of the dielectric sphere (*exa103*) to solve for only the upper half of the domain ($z > 0$). Notice that `U` must vanish on the symmetry plane. Show that the dielectric tends to behave like a metallic ball as `epsr` becomes very large.
- ❑ Use *exa105* as a template for the problem of two metal spheres of equal charge magnitude and with a potential difference of 30.0. Find the capacitance between these spheres.
- ❑ Modify *exa106* to determine the total charge in the dielectric.

11 Electrostatic Force and Energy

It is well known that a Coulomb force exists between point charges, and hence it is only natural that extended objects also attract or repel each other. We expect the surface force per unit area to be given by $\sigma_s \mathbf{E} = D_n \mathbf{E}$, but considering that E is zero inside the metal and non-zero outside, one must use the correct vector field.

The analytic expression[6p223] for the force per unit area of a metallic surface is

$$\mathbf{f} = \tfrac{1}{2} D_n \mathbf{E} \qquad \bullet$$

In fact, if one replaces the charged sheet by a layer of finite thickness one can prove that the halfway field yields the appropriate value for the force.

Forces on a Parallel-Plate Capacitor

In the case of the parallel-plate capacitor (p.79), the vector plot of **D** suggests that the forces on the left and right edges of a plate are *horizontal*, equal and opposite. In other words, they might deform the plate horizontally, but the total force is zero.

The total *vertical* force may be obtained by a line integral of $f_y = \tfrac{1}{2} D_n E_y$ over the boundary of the upper plate, say. To achieve this, it is sufficient to add the following lines to *exa092* (p.79).

```
TITLE   'Forces on a Parallel-Plate Capacitor'                { exa111.pde }
...             { Student Version}                            { SI units }
  elevation( 0.5*normal( D)*Ey) on 'upper'                { Integral ⇨ force }
  report( -1*2*xx* 0.5*[7*eps0*U0/(2*d0)]* U0/(2*d0)) as 'F_elementary'
END
```

The elevation plot gives us the electrostatic force between the plates by the integral under the curve. The answer is -0.0391, which

definitely is a measurable force. We compare that with an elementary estimate of (area)*0.5*Dey*Ey and find the smaller value -0.0387.

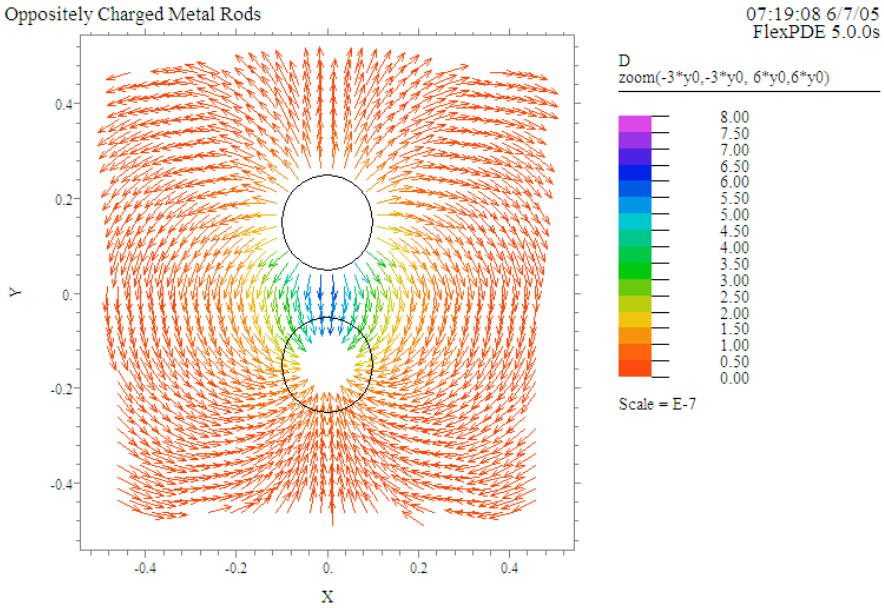
Oppositely Charged Metal Rods

Let us now calculate the force in another simple situation, that between two metal rods of circular cross-section, embedded in glass, at different values of U. Here, we again calculate charges and forces by elevation plots.

```
TITLE   'Oppositely Charged Metal Rods'                    { exa112.pde }
SELECT    errlim=3e-4    ngrid=1    spectral_colors
VARIABLES   U
DEFINITIONS
  L=1    r0=0.1    y0=0.15    U2=1e3
  eps0=8.854e-12    epsr=7.0    eps=epsr*eps0
  Ex=-dx(U)    Ey=-dy(U)    E=-grad(U)    Em=magnitude(E)
  Dex=eps*Ex    Dey=eps*Ey    D=eps*E    Dm=magnitude(D)
  Q=line_integral( normal( D), 'rod1')    { Line integral over rod1 }
  Fy=Q^2/(2*pi*2*y0)/ eps                 { Elementary force estimate }
EQUATIONS
  div( D)=0
BOUNDARIES
region 'glass' start(-L,-L) natural(U)=0
  line to (0,-L) to (L,-L)  to (L,L) to (-L,L)  close
  start 'rod1' (r0,-y0) value(U)=0                  { Exclude }
  arc( center=0,-y0) angle=360
  start 'rod2' (r0,y0) value(U)=U2
  arc( center=0,y0) angle=360
PLOTS
  grid(x,y) zoom(-3*y0,-3*y0, 6*y0,6*y0)
  contour( U)    contour( Dm) painted zoom(-3*y0,-3*y0, 6*y0,6*y0)
  vector( D) norm zoom(-3*y0,-3*y0, 6*y0,6*y0)
  elevation( normal(D)) on 'rod1'                   { ⇨ Charge on rod 1 }
  elevation( normal(D)) on 'rod2'                   { ⇨ Charge on rod 2 }
  elevation( 0.5*normal(D)*Ey) on 'rod1' { ⇨ Force on rod 1 }
  elevation( 0.5*normal(D)*Ey) on 'rod2' { ⇨ Force on rod 2}
     report(Fy) as 'Fy_elementary'                  { Estimate of force }
END
```

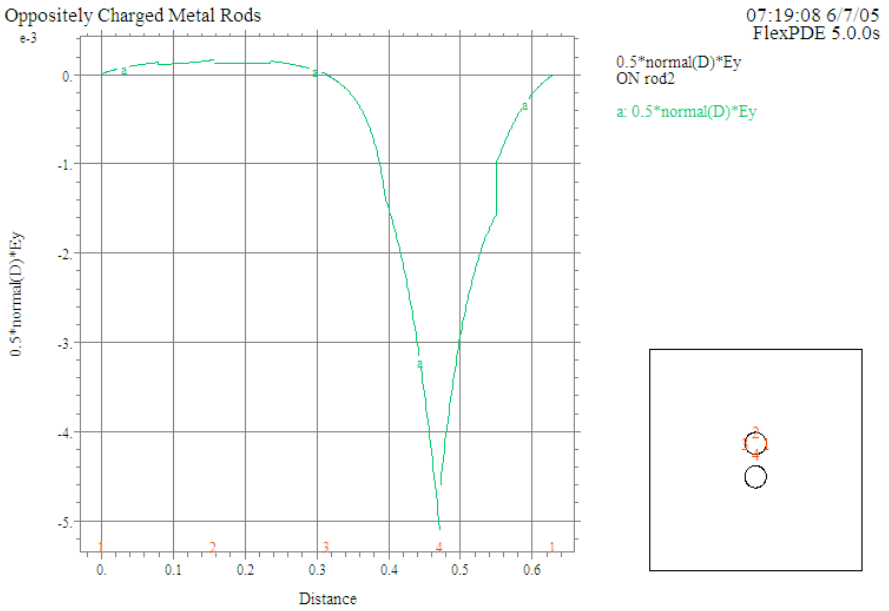
Here, we calculate the charge Q by a *line_integral* over the rim of a rod. This is similar to the integration automatically obtained with an elevation plot, but *line_integral* results in a value stored in a variable that may be used later in an expression.

The painted contour plot of Dm illustrates that the field has its highest magnitude close to the rods and close to the vertical centerline between the rods. The vector plot below confirms this.



The last elevation plot is strongly peaked in the direction pointing toward the opposite rod. This peak is at point 4, indicated on the orientation diagram beside. We may also be guided by the definition of 'rod2' under *boundaries*. The integral of this curve yields the force, and we find that the magnitude of the reaction force on 'rod1' is the same to about 0.2%.

The *elementary* estimate of the force is based on Gauss' law, applied to an infinitely long line charge centered on the first rod, the other rod also being represented by a line charge. The resulting absolute value is only about 70% of that given by the integral, but this is not surprising, considering that most of the charge on a rod resides on the half-surface facing the other rod. The vector plot and the elevation plots of D_n illustrate this fact.



exa112: Grid#2 P2 Nodes=802 Cells=383 RMS Err= 6.9e-4
Fy_elementary= 3.290073e-4 Integral= -4.601588e-4

Using the Professional Version, we may reduce the radii of the rods to one half and remove the node limit. We then find that the mutual force is smaller and that the elementary estimate agrees with the integrated value to about 12 %.

Charged Metal Balls

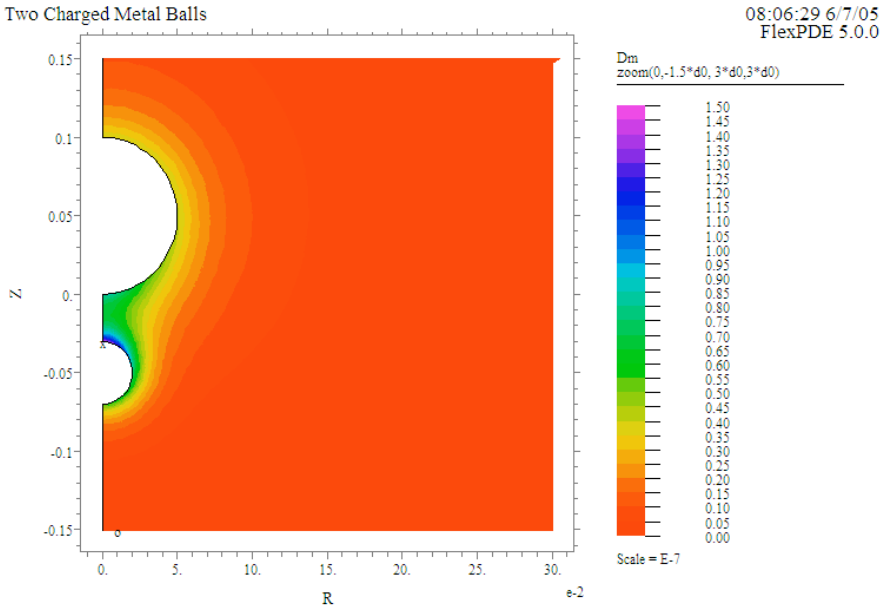
The *cylindrical* coordinate system permits us to treat problems with several axially symmetric objects on a common axis. The following descriptor is for two metal spheres, embedded in a dielectric and kept at potentials of zero and U2 respectively. In this problem, we shall need the accuracy of the Professional Version.

In the *boundaries* segment we use the point charge approximation to specify the component normal(Dp) caused by an enclosed charge *Q*, as we did in *exa104* (p.98). Having done that, we are assured that the *sum* of the charges is *Q*, but we cannot specify the charge on each one of the metal balls. We shall see, however, that the *voltage* (U2) applied between the balls determines the sharing of the total charge.

```
TITLE   'Two Charged Metal Balls'                          { exa113.pde }
SELECT                                          { Professional Version }
  errlim=3e-5     ngrid=1     spectral_colors
COORDINATES     ycylinder( 'r', 'z')
```

```
VARIABLES   U
DEFINITIONS
  r1=0.02    d0=0.1    r2=0.05    r3=1.0
  eps0=8.854e-12    eps=7*eps0          { Glass }
  Q=1e-9    U2=40                       { Voltage controlling charge }
  Er=-dr(U)    Ez=-dz(U)    E=-grad(U)    Em=magnitude(E)
  Der=eps*Er    Dez=eps*Ez    D=eps*E    Dm=magnitude(D)
  Up=Q/(4*pi*eps*sqrt(r^2+z^2))         { Potential of a point charge }
  Ep=-grad(Up)    Dp=eps*Ep             { For outer boundary }
  Q1=surf_integral( -normal(D), 'ball1')                    { Charge 1 }
  Q2=surf_integral( -normal(D), 'ball2')                    { Charge 2 }
  F1=surf_integral( -0.5*normal(D)*Ez, 'ball1')             {  Force 1 }
  F2=surf_integral( -0.5*normal(D)*Ez, 'ball2')             {  Force 2 }
EQUATIONS
  div(D)=0                                  { div( eps*( -grad( U))=0 }
BOUNDARIES
region 'dielectric' start (0,-d0/2-r1)  natural(U)=0  line to (0,-r3)
  natural(U)=normal(Dp) arc( center=0,0) angle=180 to (0,r3)
  natural(U)=0 line to (0,d0/2+r2)  value(U)=U2
  arc(center=0,d0/2) angle=-180                    { Ball2 }
  natural(U)=0  line to (0,-d0/2+r1) value(U)=0
  arc(center=0,-d0/2) angle=-180                   { Ball1 }
feature
  start 'outer' (0,-r3) arc( center=0,0) angle=180
  start 'ball1'  (0,-d0/2+r1) arc( center=0,-d0/2) angle=-180
  start 'ball2' (0,d0/2+r2) arc( center=0,d0/2) angle=-180
PLOTS
  contour( U)    vector( D) norm zoom(0,-1.5*d0, 3*d0,3*d0)
  contour( Dm) painted zoom(0,-1.5*d0, 3*d0,3*d0)
  elevation( normal(D)) on 'outer'                 { ⇨ Total charge }
summary                        { Numerical and textual plot page }
    report(Q1) report(Q2) report(Q1+Q2) report(F1)   report(F2)
    report(Q1*Q2/ [4*pi*eps*d0^2]) as 'Elementary estimate of F2'
END
```

The following contour of Dm indicates the charge distribution on the balls and demonstrates that the charge on each sphere is highest on the side facing the other one. At the voltage chosen the charges are of opposite sign, which means that the distributions will be drawn closer to each other than the distance between the centers.



exa113: Grid#3 p2 Nodes=3126 Cells=1525 RMS Err= 1.8e-5
Vol_Integral= 2.423164e-10

In the *definitions* section of the descriptor we include a surf_integral yielding the numerical value of the charge on each ball. The Gauss integral over the outer boundary, obtained from the elevation plot, should reproduce the sum of those values.

The *summary* reports the forces obtained by integration and compares those to an elementary estimate by Coulomb's law. The following figure shows the relevant lines.

Two Charged Metal Balls
08:06:29 6/7/05
FlexPDE 5.0.0

SUMMARY

Q1= -4.369071e-10
Q2= 1.433359e-9
Q1+Q2= 9.964524e-10
F1= 9.750415e-8
F2= -9.836818e-8
Elementary estimate of F2= -8.040754e-8

Since the charge distributions attract each other, it is not surprising to find that the elementary estimate of the force (between point charges placed at the centers) is *smaller* than the integral value.

If we change the applied voltage to a *negative* value, U2=-10, we obtain the following summary. The individual charges are now of the *same* sign, which means that the charge distributions will repel each other.

Two Charged Metal Balls 08:24:16 6/7/05
FlexPDE 5.0.0

SUMMARY

Q1= 3.642023e-10
Q2= 6.297148e-10
Q1+Q2= 9.939171e-10
F1= -2.255116e-8
F2= 2.253883e-8
Elementary estimate of F2= 2.944687e-8

The contour plot of Dm confirms that the charges now are on the far side of the balls. Consequently, the point charge estimate becomes *higher* than that obtained from the FEA solution.

Forces on a Dielectric

As we have already noted, an applied electric field induces no polarization *volume* charges in a dielectric object of constant permittivity. A polarization *surface* charge $\sigma_p = P_n$ will occur, however. The field **E** will of course exert a force on that surface charge. It is possible to show[6p106] that the force per unit area f_s becomes, in terms of the tangential (t) and normal (n) components,

$$f_s = \tfrac{1}{2}(\varepsilon_2 - \varepsilon_1)\left(E_{t1}^2 + E_{n1}^2 \varepsilon_1 / \varepsilon_2\right)\mathbf{n} \qquad \bullet$$

The index 2 refers to the object and 1 to the surrounding material. Surprisingly, the force is directed along the normal **n**, just as in the case of a metal object.

For the descriptor it is convenient to have an expression in terms of the field *inside* the object (2). Since E_t and D_n are continuous across the interface, we have $E_{t1} = E_{t2}$ and $\varepsilon_1 E_{n1} = \varepsilon_2 E_{n2}$. Thus the force per unit area may be written

$$f_s = \tfrac{1}{2}(\varepsilon_2 - \varepsilon_1)\left(E_{t2}^2 + E_{n2}^2 \varepsilon_2 / \varepsilon_1\right)\mathbf{n} \qquad \bullet$$

Metallic and Dielectric Rod across a Parallel Field

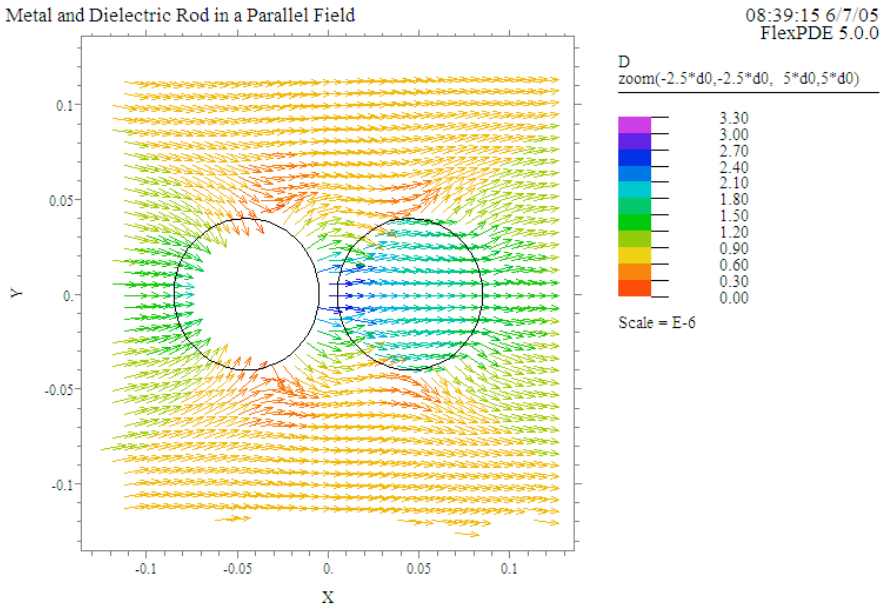
We shall now calculate the force on a dielectric rod, using a metal rod as the second element for comparison. These objects are in a vacuum and exposed to a transverse, initially parallel field. In view of Newton's third law the forces must be equal and opposite.

Considering that we shall have to integrate the *x*-component of a normal force over the surface, we must multiply fs by the direction cosine of the angle between the normal and the *x*-axis. For this purpose we define a vector field unitx of constant magnitude, pointing along the *x*-axis. In the descriptor, normal(unitx) is equivalent to the direction cosine.

```
TITLE   'Metal and Dielectric Rod in a Parallel Field'        { exa114.pde }
SELECT                                                       { Professional Version }
  errlim=1e-6    ngrid=1    spectral_colors
VARIABLES   U
DEFINITIONS
  r0=0.04    d0=0.045    L=1.0
  eps0=8.854e-12    epsr    eps=epsr*eps0
  Ex0=1e5    Dx0=eps0*Ex0                                    { Far Field }
  Ex=-dx(U)    Ey=-dy(U)    E=-grad(U)    Em=magnitude(E)
  D=eps*E    Dm=magnitude(D)
  F1=line_integral( -0.5*normal(D)*Ex, 'rod1')  { Force on metal rod }
  fs=0.5*(eps-eps0)*[ tangential(E)^2+normal(E)^2*epsr]  { Force/area }
  unit_x=vector(1,0)                                         { Direction of x }
  f=fs*normal( unit_x)                { Force/area, x-component }
  F2=surf_integral( f, 'rod2')        { Integrated force, x-component }
EQUATIONS
  div( D)=0                           { div( eps*( -grad( U)))=0 }
BOUNDARIES
region 'domain'   epsr=1                                     { Vacuum }
  start(-L,-L)  natural(U)=0  line to (0,-L)  to (L,-L)
  natural(U)=Dx0  line to (L,L)
  natural(U)=0  line to (-L,L)
  natural(U)=-Dx0  line to close
  start 'rod1' (-d0-r0,0)  value(U)=0                        { Exclude for metal rod }
  arc( center=-d0,0) angle=360
region 'glass'  epsr=7.0   start 'rod2' (d0+r0,0)
  arc( center=d0,0) to (d0,r0) to (d0-r0,0) to (d0,-r0) close
PLOTS
  grid( x, y) zoom(-5*d0,- 5*d0,  10*d0,10*d0)
  contour( Dm) zoom(-2.5*d0,-2.5*d0,  5*d0,5*d0)
  vector( D) norm zoom(-2.5*d0,-2.5*d0,  5*d0,5*d0)
    report( F1) as 'F1 on metal' report( F2) report((F1+F2)/F1)
END
```

In this descriptor, we have included a plot of the grid to show the density of nodes on the outlines where we make the integration.

The normalized vector plot below illustrates the geometry.



The bottom line reports on the forces. We notice that the magnitudes of the forces are equal to about 2%, in agreement with the law of action and reaction. A smaller errlim would further reduce the difference.

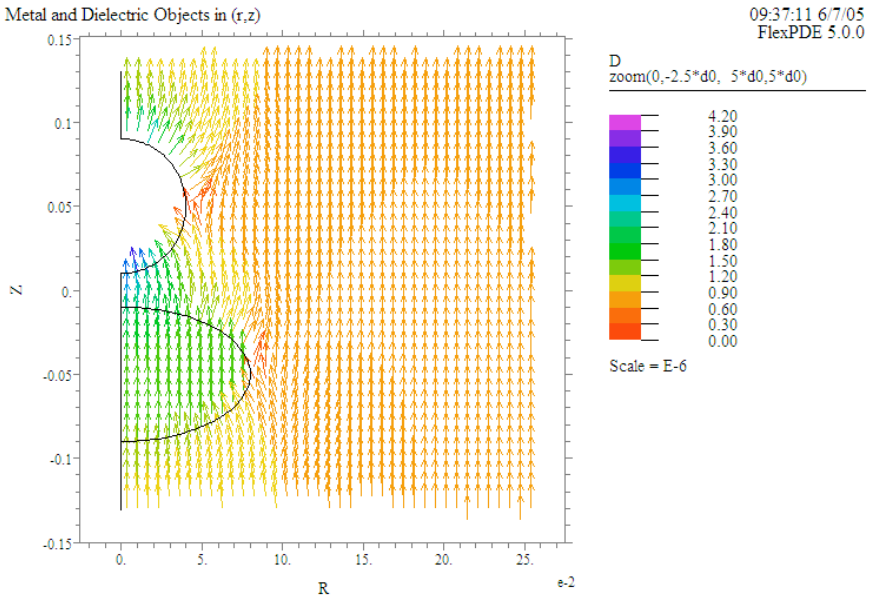
Metal Ball and Dielectric Ellipsoid

Let us now modify the preceding descriptor to deal with axial symmetry, using *exa105* as a guide. We keep the metal ball and add an oblate dielectric ellipsoid on the same axis. It is an easy matter to make the cross section of the dielectric object elliptical. We only need to double the radius at one point of the arc as indicated below.

The surface integrals include the factor $2\pi\rho$ for the annular area element. Otherwise, the calculation is very similar to that for the rods. In this case we need to define a field of unit vectors (unit_z) pointing in the *z* direction in order to generate the cosine factor.

```
TITLE   'Metal and Dielectric Objects in (r,z)'                { exa115.pde }
SELECT                                                  { Professional Version }
  errlim=1e-7     ngrid=1     spectral_colors
COORDINATES   ycylinder('r','z')
VARIABLES   U
DEFINITIONS
  r0=0.04     d0=0.05     Lr=2.0     Lz=1.0
  eps0=8.854e-12     epsr     eps=epsr*eps0
  Ez0=1e5     Dz0=eps0*Ez0                        { Far Field }
  Er= -dr(U)     Ez= -dz(U)     E= -grad(U)     Em=magnitude(E)
  D= eps*E     Dm=magnitude(D)
  F1=surf_integral( -0.5*normal(D)*Ez, 'ball')   { Force on metal }
  fs=0.5*(eps-eps0)*[normal(E)^2*epsr+tangential(E)^2]   { Dielectric }
  unit_z=vector(0,1)                                { Direction of z }
  f=fs*normal( unit_z)                        { Force/area, z-component }
  F2=surf_integral( f, 'ellipsoid')                 { Force on dielectric }
EQUATIONS
  div( D)=0
BOUNDARIES
region 'domain'   epsr=1                            { Vacuum outside }
  start(0,-Lz)  natural(U)= -Dz0  line to (Lr,-Lz)
  natural(U)=0 line to (Lr,Lz)
  natural(U)=Dz0  line to (0,Lz)
  natural(U)=0  line to (0,d0+r0) value(U)=0
  arc( center=0,d0) angle= -180 to (0,d0-r0)    { Metal ball }
  natural(U)=0 line to close
region 'glass'  epsr=7.0
  start 'ellipsoid' (0,-d0-r0)
  arc( center=0,-d0) to (2*r0,-d0) to (0,-d0+r0) line to close
feature
  start 'ball' (0,d0+r0) arc( center=0,d0) angle=-180 to (0,d0-r0)
PLOTS
  vector( D) norm     contour( Dm) painted zoom(0,-2.5*d0,  5*d0,5*d0)
  vector( D) norm zoom(0,-2.5*d0,  5*d0,5*d0)
    report( F1) as 'F1 on metal'   report( F2)   report((F1+F2)/F1)
END
```

The following plot displays the field direction in the vacuum and in the dielectric ellipsoid. The maximum field intensity occurs where the two objects come close. The force F2 on the metal rod, reported on the bottom line, agrees within about 0.7% with the opposite force on the metal ball. Notice that we have used quite different expressions for these two calculations.



Evidently, we are able to calculate forces for a large variety of geometrical configurations and materials.

Electrostatic Energy

The expression for the electrostatic field energy[3p138] is given by

$$W = \frac{1}{2}\iiint \mathbf{D}\cdot\mathbf{E}\, dv \ , \qquad \bullet$$

where the integral is to be taken over the volume of the field. It is worth noticing that the expression for the energy density is similar to the one we just integrated to obtain the force.

The software readily provides routines for evaluating W. Let us calculate the energy per unit length of the capacitor in (x,y) space (p.79) and then compare it to the energy required to charge the capacitor, which is known to be $W = CU^2/2 = QU/2$ from simple theory. For this task we add the following lines to *exa111*. The Student Version is now sufficient.

```
TITLE    'Field Energy for a Plate Capacitor'                { exa111a.pde }
…                                                            { Student Version }
summary
  report('Comparison of energy values:')
  report( area_integral( 0.5* Dm*Em)) as 'W'                 { Per unit length }
  report( line_integral( -normal(D), 'upper')* U0/2) as 'QU/2'
END
```

On running this extended descriptor we find the volume integral to yield the energy 8.40e-3. Using a value for the capacitance derived from the charge on the upper plate, we find $QU/2 = 8.24\text{e-}3$.

Field Energy of Metal Balls in Glass

We shall now look into the detailed energy balance in the case of two charged metal spheres. Here, they carry charges of equal magnitude but opposite signs, as required by the *natural* boundary conditions. The potential difference U2 is maintained by a voltage source, such as a battery. The descriptor *exa113* could be used as a template.

```
TITLE   'Field Energy of Metal Balls in Glass'               { exa116.pde }
SELECT                                                  { Professional Version }
  errlim=1e-6    ngrid=1    spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES   U
DEFINITIONS
  r1=0.03    d0=0.100    r2=0.03    r3=1.0    U2=1.0       { Voltage }
  eps0=8.854e-12    eps=7*eps0                              { Glass }
  Er=-dr(U)    Ez=-dz(U)    E=-grad(U)    Em=magnitude(E)
  D=eps*E    Dm=magnitude(D)
  Q1=surf_integral(-normal(D), 'ball1')                    { Charge 1 }
  Q2=surf_integral(-normal(D), 'ball2')                    { Charge 2 }
  Q=0.5*[ abs(Q1)+ abs(Q2)]                                { Average }
  F1=surf_integral(-0.5*normal(D)*Ez, 'ball1')             { Force 1 }
  F2=surf_integral(-0.5*normal(D)*Ez, 'ball2')             { Force 2 }
  W=vol_integral(0.5*Dm*Em)                                { Energy }
EQUATIONS
  div( D)=0
BOUNDARIES
region 'domain'
  start (0,-d0/2-r1)  natural(U)=0 line to (0,-r3)
```

```
  natural(U)=0 arc(center=0,0) angle=180  line to (0,d0/2+r2)
  value(U)=U2  arc(center=0,d0/2) angle=-180
  natural(U)=0  line to (0,-d0/2+r1)
  value(U)=0  arc(center=0,-d0/2)  angle=-180 close
feature
  start 'outer' (0,-r3) arc( center=0,0) angle=180
  start 'ball1' (0,-d0/2+r1) arc( center=0,-d0/2) angle=-180
  start 'ball2' (0,d0/2+r2) arc( center=0,d0/2) angle=-180
PLOTS
  vector(D) norm zoom(0,-d0, 2*d0,2*d0)
summary
  report(U2)  report(d0)  report(Q1)  report(Q2)
  report(Q)  report(W)  report(Q*U2/2)
  report(F1)  report(F2)  report((abs(F1)+ abs(F2))/2) as 'F_mean'
END
```

Studying the resulting *summary* page below we find reasonable agreement between opposite charges and also between the force and its reaction force. We also notice that the elementary estimate $QU_2/2$ for a plate capacitor agrees well with the integrated field energy W.

Field Energy of Metal Balls in Glass 12:37:36 6/7/05
FlexPDE 5.0.0

SUMMARY

U2= 1.000000
d0= 0.100000
Q1= -1.689503e-11
Q2= 1.689319e-11
Q= 1.689411e-11
W= 8.455644e-12
Q*U2/2= 8.447055e-12
F1= 4.193710e-11
F2= -4.190440e-11
F_mean= 4.192075e-11

It is interesting to compare the above solution to that obtained with an increased distance between the balls. Re-running the descriptor with **d0=0.102** we find the following summary.

Field Energy of Metal Balls in Glass 12:49:54 6/7/05
FlexPDE 5.0.0

SUMMARY

U2= 1.000000
d0= 0.102000
Q1= -1.673302e-11
Q2= 1.673152e-11
Q= 1.673227e-11
W= 8.374483e-12
Q*U2/2= 8.366135e-12
F1= 3.919790e-11
F2= -3.917037e-11
F_mean= 3.918414e-11

It is easy to succumb to the false notion that the positive work done (F*0.002) on the charged balls would *increase* the field energy. In

fact, we see that the energy *decreases*. So does the charge, however, which means that some charge is transferred via the battery from the positive to the negative ball, falling through the potential difference U2. This means that the work done is added to the liberated field energy to produce heat (or chemical energy) in the battery.

In fact, it can be shown analytically[3p141] that the mechanical work done and the change of field energy W each amount to one-half of the energy delivered from (or to) the battery. The latter energy is equal to the difference in Q multiplied by the voltage drop, or 1.584e-13, which would imply 7.92e-14 for each of the other energy terms.

The mechanical work may be estimated as the average force multiplied by the displacement 0.002, which becomes 8.11e-14. The change in electrical field energy is found to be 8.11e-14. These two terms are hence reasonably equal, but they are somewhat higher than half of the battery energy.

Energy Minimum

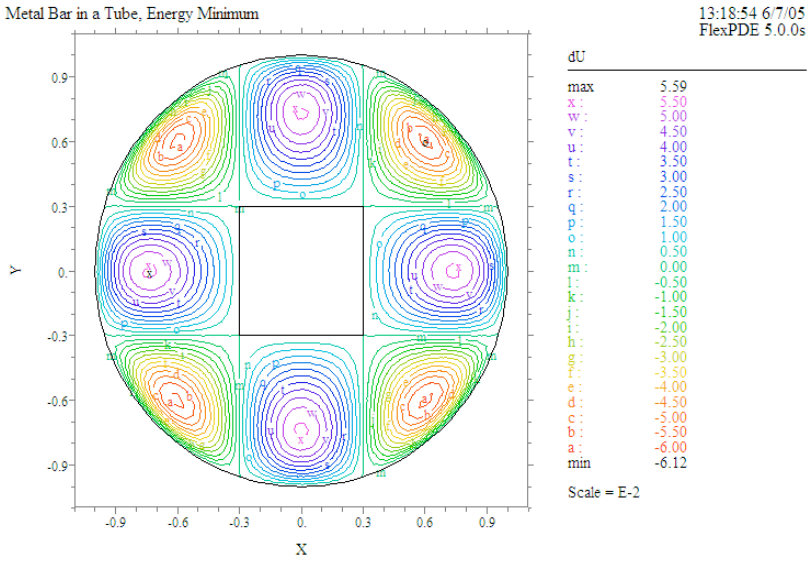
The static field due to a system of charged conductors in a dielectric medium may be complicated, but according to a classical theorem by Thomson the solution is simply the function corresponding to the minimum field energy. From a practical point of view this seems self-evident, since any excess energy could be imparted to the electrons, which would move in the conductors until this energy becomes dissipated. The solution should thus correspond to a smaller field energy value than any other function obeying the same boundary conditions.

We shall now illustrate the minimal energy theorem by an extension of *exa072* (p.58). In the following descriptor we construct a function dU that vanishes on the outer and inner boundaries, and we use that to perturb the solution. As indicated the Student Version may be used in this case.

```
TITLE   'Metal Bar in a Tube, Energy Minimum'              { exa117.pde }
SELECT                                                    { Student Version }
  errlim=3e-5     ngrid=1     spectral_colors
VARIABLES   U
```

```
DEFINITIONS
  r0=1.0     b=0.3     Ub=1.0     eps0=8.854e-12
  E=-grad(U)     Em=magnitude(E)
  dU=3*( x^2+y^2- r0^2)*(x-b)*(x+b)*(y-b)*(y+b)       { Perturbation }
  W0=area_integral( 0.5* eps0*Em^2)          { Energy for solution }
  U1=U-dU     E1=-grad( U1)     E1m=magnitude( E1)
  W1=area_integral( 0.5* eps0*E1m^2)  { Over the entire domain }
  U2=U+dU   E2=-grad( U2)     E2m=magnitude( E2)
  W2=area_integral( 0.5* eps0*E2m^2)
EQUATIONS
  div( grad(U))=0
BOUNDARIES
region 'domain'
  start 'tube' (r0,0) value(U)=0  arc( center=0,0) angle=360
  start 'bar' (-b,-b) value( U)=Ub  line to (b,-b) to (b,b) to (-b,b)  close
PLOTS
  contour( U)     contour( dU) report( W0) report( W1) report( W2)
END
```

The function dU is arbitrary, except for the condition that it vanish on the domain boundaries, as shown below. The functions U-dU and U+dU then both satisfy the boundary conditions, and W1 and W2 are the corresponding field energies.



The area_integral gives us the field energy (per unit length in z) over the (x, y) domain. The contour plot above shows dU and the

numerical results for the field energy. We find that the values W1 and W2 are clearly higher than that pertaining to the true solution (W0).

Exercises

❑ Compare two equivalent expressions for the electrostatic energy in the case of the circular-plate capacitor (*exa102*).
❑ Find the field energy associated with the charged, conducting ellipsoid (*exa104*). Expand the domain to estimate how much energy is missing.
❑ Explore how the total charge is shared between the metal balls in *exa113* as the applied voltage U2 is varied.
❑ Calculate the force between a charged metal ball and a thick slab of dielectric material (half-space dielectric).
❑ Modify the cross-section of the dielectric rod in *exa114* to make it an ellipse with the major axis vertical and twice as large as the minor axis.
❑ Solve the problem of a charged metal ball and a dielectric toroid on a common axis. Explore the resulting forces.
❑ Modify *exa117* by taking the absolute value of the expression for dU. Change the perturbing function again by taking the square root of the absolute value.

12 Magnetostatics in (*x*,*y*) Space

From Maxwell's equations[3p323] we know that $\nabla \times \mathbf{H} = \mathbf{J} + \partial \mathbf{D}/\partial t$, which in the static case simply becomes

$$\nabla \times \mathbf{H} = \mathbf{J} \qquad \bullet$$

or expressed in Cartesian components

$$\nabla \times \mathbf{H} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial z} \\ H_x & H_y & H_z \end{vmatrix} = \begin{Bmatrix} \dfrac{\partial H_z}{\partial y} - \dfrac{\partial H_y}{\partial z} \\ \dfrac{\partial H_x}{\partial z} - \dfrac{\partial H_z}{\partial x} \\ \dfrac{\partial H_y}{\partial x} - \dfrac{\partial H_x}{\partial y} \end{Bmatrix} = \mathbf{J} = \begin{Bmatrix} J_x \\ J_y \\ J_z \end{Bmatrix} \qquad \bullet$$

If one of the first components of **J** is non-zero, then H_z may also become non-zero. Since we shall consider magnetic fields with *x*- and *y*-components only, we assume vanishing J_x and J_y. The above vector equation thus reduces to

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = J_z \qquad \bullet$$

Electromagnetic theory[3p232] shows that the magnetic flux density **B**, may be expressed in terms of a *vector potential* $\mathbf{A}(x,y,z)$, i.e.

$$\mathbf{B} = \nabla \times \mathbf{A} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial z} \\ A_x & A_y & A_z \end{vmatrix} = \begin{Bmatrix} \dfrac{\partial A_z}{\partial y} - \dfrac{\partial A_y}{\partial z} \\ \dfrac{\partial A_x}{\partial z} - \dfrac{\partial A_z}{\partial x} \\ \dfrac{\partial A_y}{\partial x} - \dfrac{\partial A_x}{\partial y} \end{Bmatrix} \qquad \bullet$$

We wish to study vector fields **H** and **B** that are parallel to the (x, y) plane. As seen from the third component of the above vector equation, non-zero functions A_x and A_y might produce a perpendicular component, H_z, and hence we keep only the last component of the vector potential, A_z. With $\mathbf{B} = \mu\mathbf{H}$, the above equation then simplifies to

$$\begin{Bmatrix} B_x \\ B_y \\ B_z \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial A_z}{\partial y} \\ -\dfrac{\partial A_z}{\partial x} \\ 0 \end{Bmatrix} \quad \text{or} \quad \begin{Bmatrix} H_x \\ H_y \\ H_z \end{Bmatrix} = \begin{Bmatrix} \dfrac{1}{\mu}\dfrac{\partial A_z}{\partial y} \\ -\dfrac{1}{\mu}\dfrac{\partial A_z}{\partial x} \\ 0 \end{Bmatrix}$$

Putting these components of **H** into the equation involving J_z, we obtain the Poisson-like (p.53) equation

$$\frac{\partial}{\partial x}\left(\frac{1}{\mu}\frac{\partial A_z}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{\mu}\frac{\partial A_z}{\partial y}\right) = -J_z \qquad \bullet$$

Although we could well exploit this second-order PDE, we prefer the simpler first-order equation containing the components of **H**. FlexPDE automatically expands the partial derivatives in terms of A_z as we run the descriptor.

After solving for A_z we may easily compute the magnetic flux density from $\mathbf{B} = \nabla \times \mathbf{A}$.

Magnetic Field around a Wire

Our first application will be to the magnetic field in a plane perpendicular to a long wire of circular cross-section, carrying a uniform current density $J_{z0} = 1.0$. We assume the permeability to be that of free space (μ_o), which is valid for the vacuum and also for a metal such as copper (to sufficient accuracy).

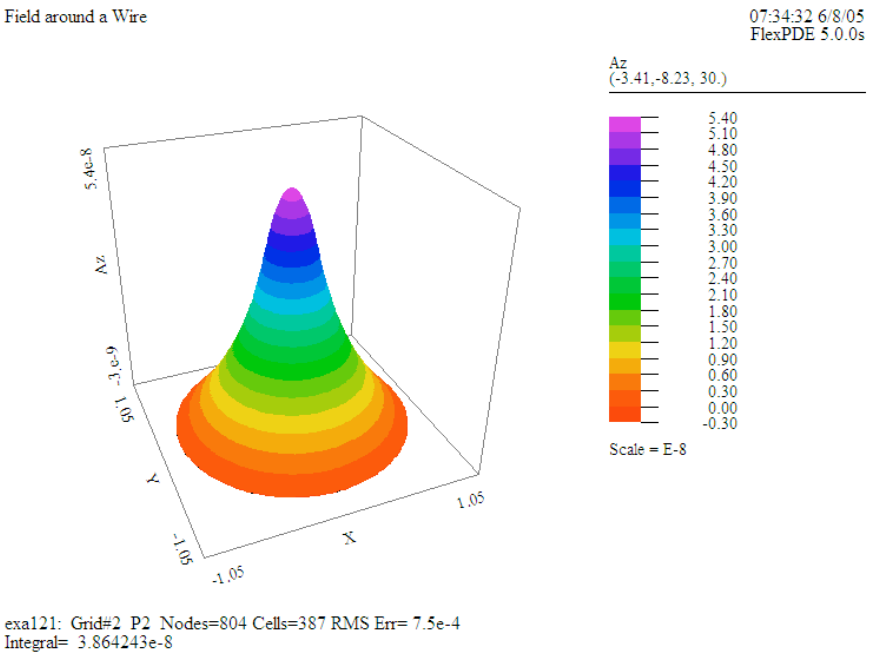
In the descriptor below, the components of **B** are denoted Bex and Bey, for the simple reason that By would be equivalent to by, which is a reserved word in the software.

```
TITLE   'Field around a Wire'                                  { exa121.pde }
SELECT                                          { Student Version }
  errlim=3e-4    ngrid=1    spectral_colors
VARIABLES   Az                          { Magnetic vector potential }
DEFINITIONS                                     { SI units }
  r0=0.2    r1=1.0    rad=sqrt( x^2+y^2) { Radius }
  mu0=4*pi*1e-7   mu=mu0    Jz0=1.0  { Jz in wire }
  Bex=dy(Az)    Bey=-dx(Az)
  B=vector( Bex, Bey)    Bm=magnitude( B)
  Hx=Bex/mu    Hy=Bey/mu    H=B/mu    Hm=Bm/mu
  Jz    Bm_ex                                     { Declared only }
EQUATIONS
  dx( Hy)- dy( Hx)=Jz
BOUNDARIES
region 'domain'  Jz=0
  Bm_ex=mu*Jz0*r0^2/(2*rad)      { Exact solution, see text below }
  start 'outer' (-r1,0) value(Az)=0  arc(center=0,0) angle=360
region 'wire' Jz=Jz0
  Bm_ex=mu*Jz0*rad/2
  start (-r0,0) arc( center=0,0) angle=360
PLOTS
  elevation( normal(B)/Bm) on 'outer'
  surface( Az)    contour( Bm)    contour( Bm_ex)
  vector( B) norm zoom(-2*r0,-2*r0,  4*r0,4*r0)
  contour( Bm-Bm_ex) report( globalmax( Bm))
  elevation( Bm, Bm_ex) from (-r1,0) to (r1,0)
END
```

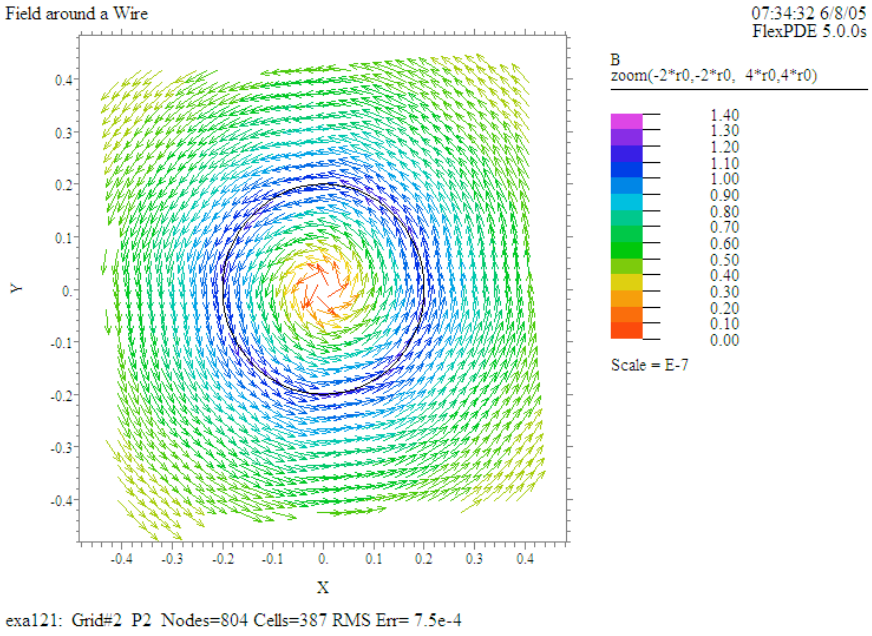
The condition imposed at the outer boundary, value(Az)=0, should be explained. At any point of the boundary we may consider $\mathbf{B} = \nabla \times \mathbf{A}$ in a rectangular coordinate system having its axes tangential (**t**) and normal (**n**) to the curve. Guided by the expression for the components of **B** on p.124 we find that the *normal* component could be written $B_n = -\partial(A_z)/\partial t$, where *t* is the coordinate along the tangent. This means that the constant value we have chosen for A_z implies that the normal component B_n vanishes at the boundary.

The first elevation plot, showing the ratio of normal(B)/Bm, confirms that B_n on the boundary is very small indeed compared to Bm.

The figure below shows a surface plot of the vector potential Az.



The next figure shows the direction of **B,** and the colors also give a rough idea about the field strength. The field is perpendicular to the radius everywhere and takes its largest value at the edge of the wire.



We notice that the field circulates counter-clockwise. This is because the *z*-axis points out of the figure on the screen, which means that the current also flows toward us.

The simple problem of a current through a straight wire has an exact solution which we obtain from the integral form of the Maxwell equation[3p228]

$$\oint \mathbf{B} \cdot d\mathbf{l} = \mu_0 I \qquad \bullet$$

where I is the current inside the path of integration. Making use of the axial symmetry we integrate **B** along concentric circles.

The expression for the current enclosed takes the constant value $I = J_{z0}\pi r_0^2$ for field points *outside* r_0, and the above integral relation thus becomes $2\pi r B_m = \mu_0 J_{z0}\pi r_0^2$.

For points *inside* the wire we have $I = J_{z0}\pi r^2$, and hence we may obtain B_m from the equation $2\pi r B_m = \mu_0 J_{z0}\pi r^2$. The exact expression for B_m will thus be different for the two regions of space.

The contour plot of Bm-Bm_ex indicates that the difference reaches to about 10% of the maximum Bm (globalmax). Over most of the domain, however, the error is much smaller. We must not forget that errlim refers to the potential Az, which we need to differentiate to obtain **B**. It is not astonishing that this operation adds some numeric scatter.

Alternatively, we may compare the FEA solution to the analytic one in other ways. The elevation plot of Bm and Bm_ex confims the agreement by curves that are very close. We also notice that the integrals of Bm and Bm_ex agree to about 0.07%.

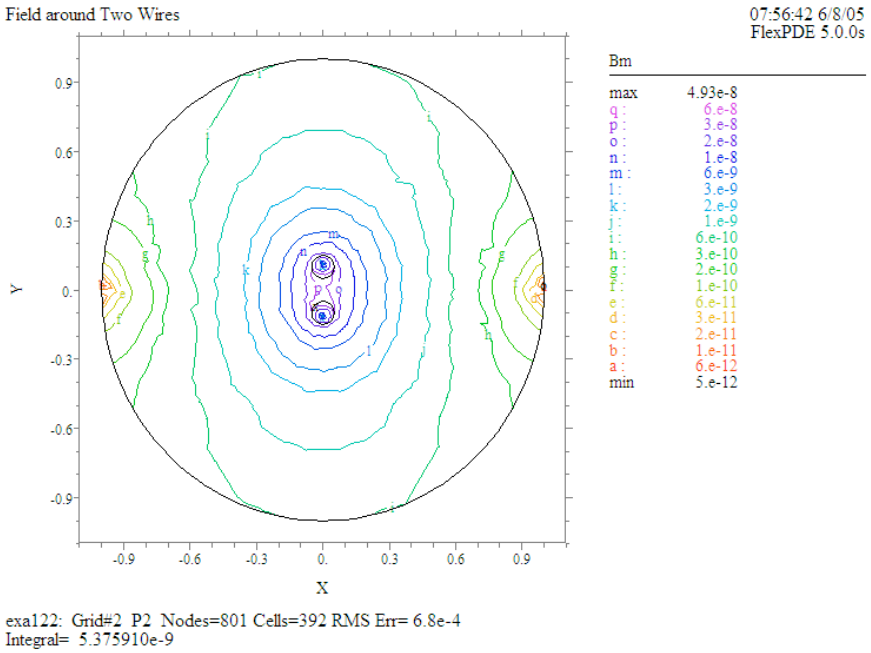
Field around Two Wires

We next consider the case of two long wires, conducting current in opposite directions, as in the following descriptor. In this case there are two circular sub-regions where the current density J_z is non-zero. Here we use specific names for these regions, *lower* and *upper*, which will be convenient for the integration in the next section.

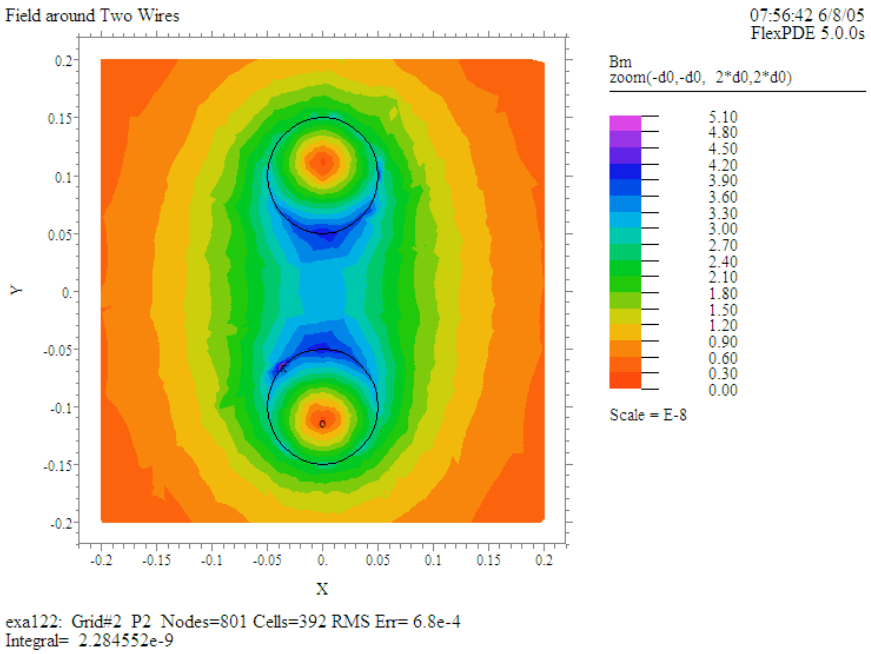
```
TITLE   'Field around Two Wires'                          { exa122.pde }
SELECT   errlim=3e-4    ngrid=1    spectral_colors
VARIABLES   Az
DEFINITIONS
```

```
  r0=0.05     d0=0.2     r1=1.0
  mu0=4*pi*1e-7     mu=mu0     Jz                          { Current density }
  Bex=dy(Az)     Bey=-dx(Az)
  B=vector( Bex, Bey)     Bm=magnitude( B)
  Hx=Bex/mu     Hy=Bey/mu     H=B/mu     Hm=Bm/mu
EQUATIONS
  dx( Hy)- dy( Hx)=Jz
BOUNDARIES
region 'domain'  Jz=0
  start(-r1,0)   value(Az) =0   arc(center=0,0) angle=360
region 'lower' Jz=  -1.0
  start (r0,-d0/2)  arc(center=0,-d0/2)  angle=360
region 'upper' Jz= +1.0
  start (r0,d0/2) arc(center=0,d0/2)  angle=360
PLOTS
  contour( Bm) log     vector( B) norm
  contour( Bm) painted zoom(-d0,-d0,  2*d0,2*d0)
  elevation( Bm) from (0,-r1) to (0,r1)
  vector( B) norm zoom(-d0,-d0,  2*d0,2*d0)
END
```

As is evident from the following logarithmic contour plot of Bm, the minimum magnitude of **B** occurs to the extreme left and right of the domain.



The zoomed, painted plot below exhibits two additional minima, on the far side of the center of each wire. There are also two maxima on the rims facing the other wire. The elevation plot illustrates these fact even more clearly.



The first vector plot is very similar to what we obtained for the dipole of charged wires in *exa053a* (p.41).

Even if this problem appears to be simple, the analytic solution is quite complicated to evaluate. It may be written as a double integral over the solution for a thin, single wire.

Force between Two Wires

Wherever we know the current density and have been able to calculate the field, we may easily obtain values for the forces between various objects. The force on a volume element carrying a current density **J** is given by the Lorentz expression

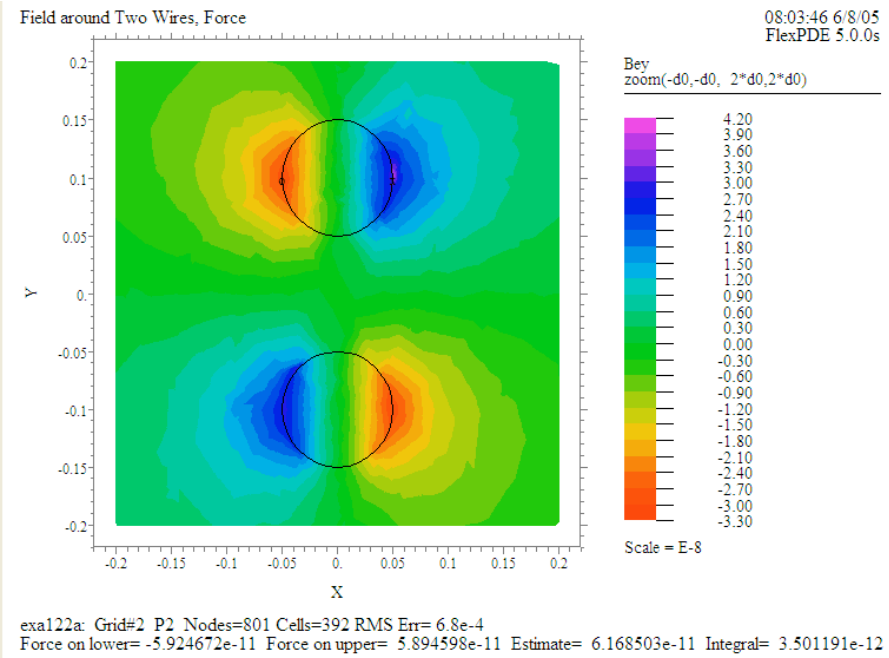
$$\mathbf{F} = \mathbf{J} \times \mathbf{B} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ J_x & J_y & J_z \\ B_x & B_y & B_z \end{vmatrix}$$ ●

or in terms of components: $F_x = 0 - J_z B_y$ and $F_y = J_z B_x - 0$. The two wires in the preceding problem are exposed to forces in the y direction.

In order to calculate these forces, let us add the following lines at the end of the descriptor. Here we compare the force obtained by integration over the region of the upper wire with an estimate using the force between two long, infinitely thin wires carrying the same current. The magnetic flux density at a distance d from a wire is[3p230] $|\mathbf{B}| = \mu I/(2\pi d)$, and since that field is perpendicular to the other wire we just need to multiply by the current I to obtain the Lorentz force.

```
TITLE   'Field around Two Wires, Force'                    { exa122a.pde }
…
  contour( Bey) painted zoom(-d0,-d0,  2*d0,2*d0)
     report( area_integral( Jz*Bex, 'lower')) as 'Force on lower'
     report( area_integral( Jz*Bex, 'upper')) as 'Force on upper'
     report( mu*(1.0*pi*r0^2)/(2*pi*d0)*(1.0*pi*r0^2)) as 'Estimate'
END
```

On running this extended descriptor we gather from the last plot (below) that the term $F_x = -J_z B_y$ produces no net force because of the odd symmetry of B_y.



For F_y we find that the elementary estimate agrees with the integrals to about 4 %. The force on the lower wire is negative and on the upper one positive, which means *repulsion* between the wires.

Two Wires inside a Magnetic Shield

Again using *exa122* as a template, we now install a tube of linear magnetic material around the two wires, as suggested below.

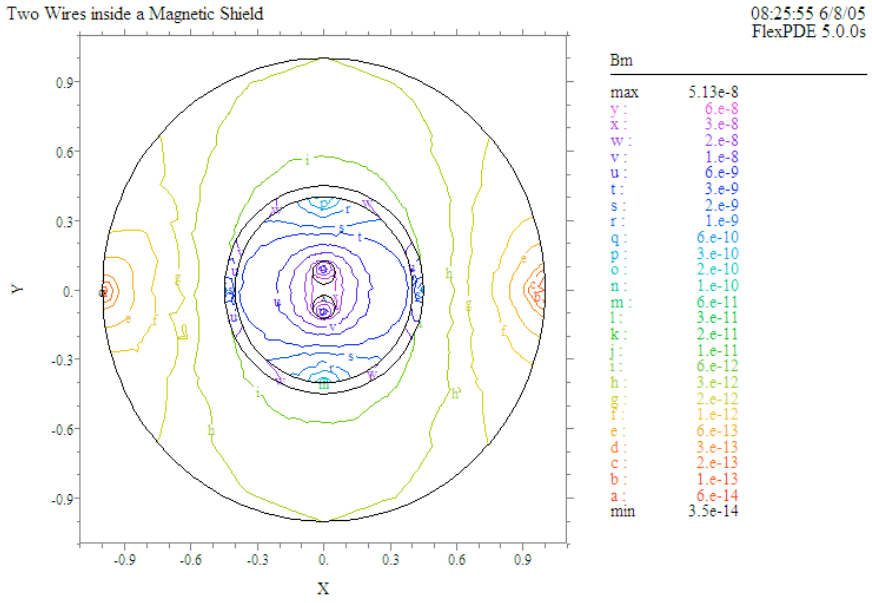
```
TITLE   'Two Wires inside a Magnetic Shield'              { exa122b.pde }
SELECT     errlim=1e-4    ngrid=1     spectral_colors
…                                                  { Professional Version }
  r0=0.05     d0=3*r0     r1=1.0     ra=0.4     rb=0.45          { Shield }
...
region 'shield' Jz=0  mu=2500*mu0                  { Magnetic material }
  start (0,-rb)  arc( center=0,0) angle=+360
  start (0,-ra)  arc( center=0,0) angle=-360
PLOTS
…
  elevation( Bm) log from (0,-r1) to (0,r1)
END
```

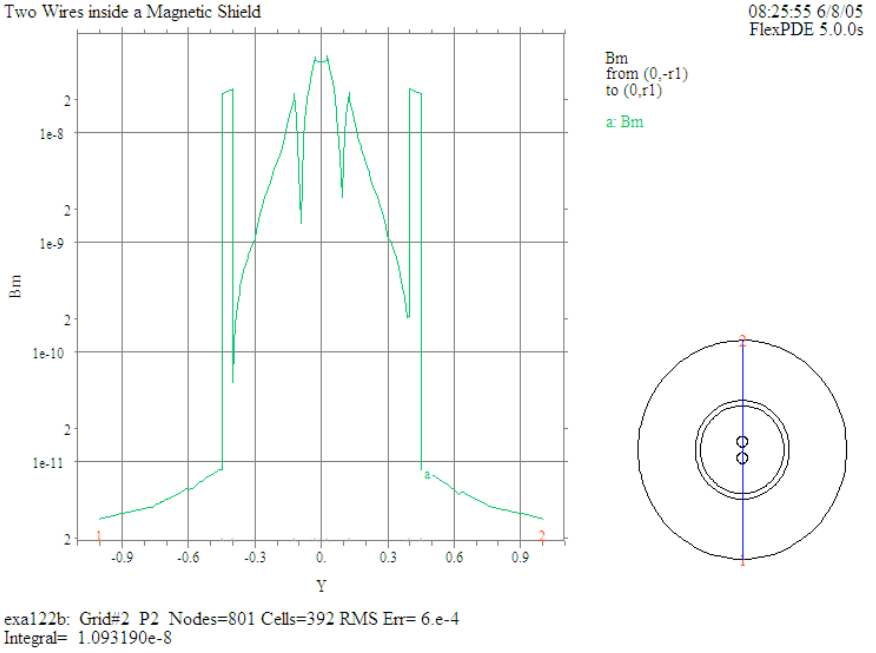
For the region named 'shield' we define a new value for the permeability μ, which overrides the previous definition. The two circles together enclose this annular region. We draw the two limiting circles (with radii ra and rb) in such directions as to keep the magnetic region on our left.

The following two plots indicates the magnitude of the field both inside and outside the magnetic shield. In the contour plot, which shows the new geometry, we find minima and maxima much as before. The presence of the shield apparently divides the domain in fields of quite different strengths.

In order to take a closer look at Bm in the various regions we created the elevation plot. Clearly the field is very weak outside the shield.

We notice that magnetic fields in different materials may be treated after the model of electric fields, since the formalisms are similar.



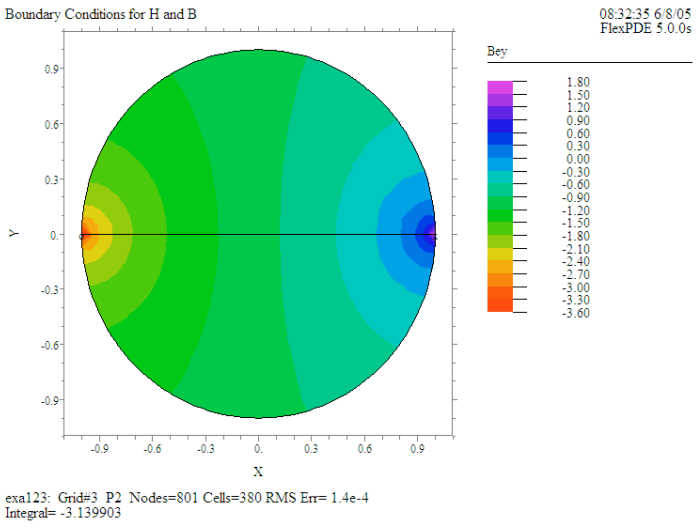


Boundary Conditions for Magnetic Vectors

FlexPDE makes is convenient to illustrate how the field vectors **H** and **B** change at a magnetic interface. Let us assume an arbitrary vector potential function Az on the outer boundary as follows.

```
TITLE 'Boundary Conditions for H and B'                    { exa123.pde }
SELECT    errlim=1e-5    ngrid=1    spectral_colors
VARIABLES   Az                                          { Student Version }
DEFINITIONS
  r1=1.0    mu0=4*pi*1e-7    mu   Jz=0                  { Current density }
  Bex=dy(Az)   Bey=-dx(Az)   B=vector( Bex, Bey)   Bm=magnitude(B)
  Hx=Bex/mu    Hy=Bey/mu    H=B/mu    Hm=Bm/mu
EQUATIONS
  dx( Hy)- dy( Hx)=Jz
BOUNDARIES
region 'domain'  mu=mu0
   start (-r1, 0) value(Az)=x+y  arc(center=0,0) angle=360
region 'magnet'  mu=1000*mu0
  start (-r1,0) arc(center=0,0) angle=180 line to close
feature
  start 'interface' (-0.9*r1,0) line to (0.9*r1,0)
PLOTS
  contour( Hx) painted    contour( Hy) painted
  contour( Bex) painted    contour( Bey) painted
  elevation( Bex) on 'interface' on 'magnet'
  elevation( Bex) on 'interface' on 'domain'
  elevation( Bey) on 'interface' on 'magnet'
  elevation( Bey) on 'interface' on 'domain'
END
```

Evidently, the arbitrary potential function imposed on the boundary does correspond to a solution, as judged by the small error. The contour plot below illustrates that Bey is continuous across the interface, and the same seems to be true for Hx.



The elevation plots taken on either side of the interface show continuity and discontinuity more clearly. The plots of Bey yield accurately the same integral value, which already is an indication of continuity. If this does not convince you, hardcopy the two plots and superimpose them.

Force on a Magnetic Object

Dielectric materials usually obey a linear relation $\mathbf{D} = \varepsilon\mathbf{E}$, which means that ε is independent of **E**. In magnetic materials, on the other hand, a linear relation $\mathbf{B} = \mu\mathbf{H}$ is mostly encountered in paramagnetic and diamagnetic materials. In practice, μ for such materials is very close to μ_0, which means that the forces usually are negligible. Ferromagnetic materials have substantially higher permeability, but on the other hand they are generally non-linear. In the following example, we shall use *linear* theory, which means that we should not expect high accuracy in the ferromagnetic case.

By analogy with the electrostatic case, the local surface force on a linear magnetic object may be written[6p270]

$$\mathbf{f}_s = \frac{1}{2}(\mu_2 - \mu_1)\left(H_{t1}^2 + \frac{\mu_1}{\mu_2} H_{n1}^2 \right)\mathbf{n} \qquad \bullet$$

where the index 2 refers to the object and the index 1 to the surrounding space. The field quantities involved are the tangential and normal components of **H** just *outside* the object, as indicated by the index 1. As with the electric field, the force acts in the direction of the normal.

For calculations by means of FlexPDE it is useful to have an alternative expression in terms of the field *inside* the object. Since H_t and B_n are continuous across the interface[3p262], we have $H_{t1} = H_{t2}$ and $\mu_1 H_{n1} = \mu_2 H_{n2}$. The above formula then becomes

$$\mathbf{f}_s = \frac{1}{2}(\mu_2 - \mu_1)\left(H_{t2}^2 + \frac{\mu_2}{\mu_1} H_{n2}^2 \right)\mathbf{n} \qquad \bullet$$

It just remains to integrate over the surface of the object.

Force on a Magnetic Cylinder

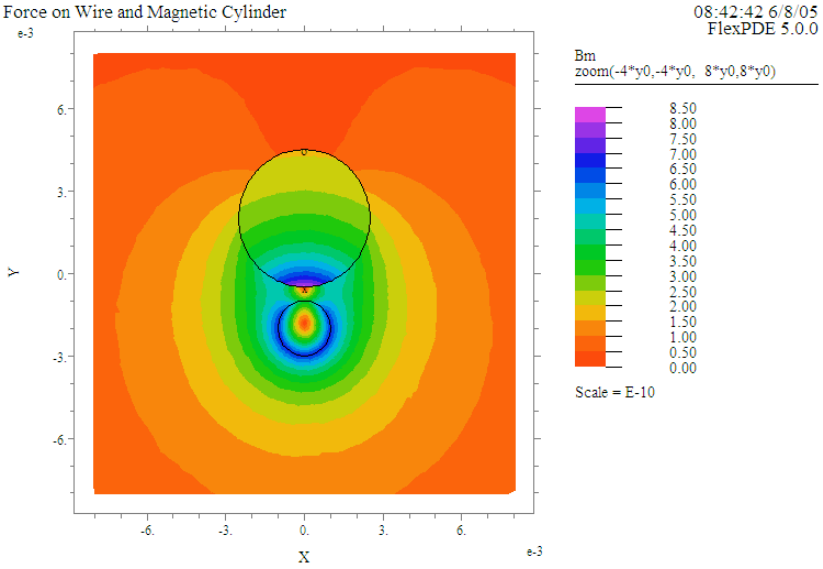
The force on a cylinder of linear magnetic material provides a simple application of the above expression. Let us consider the case of a current-carrying wire and a magnetic cylinder, parallel to the wire. Assuming these objects to be very long, we may treat the problem of the field in (x,y) coordinates.

```
TITLE   'Force on Wire and Magnetic Cylinder'              { exa124.pde }
SELECT                                          { Professional Version }
  errlim=1e-4    ngrid=1    spectral_colors
VARIABLES   Az
DEFINITIONS
  mu0=4*pi*1e-7    mu    Jz0=1.0    Jz  { Jz in wire }
  r0=1e-3    rc=2.5e-3    y0=2e-3    r1=1.0
  Bex=dy(Az)    Bey=-dx(Az)
  B=vector( Bex, Bey)    Bm=magnitude( B)
  Hx=Bex/mu    Hy=Bey/mu    H=B/mu    Hm=Bm/mu
  F_wire=area_integral( Jz*Bex, 'wire') { Lorentz force }
  fs=0.5*(mu-mu0)*[ tangential(H)^2+ mu/mu0*normal(H)^2]
  unit_y=vector(0,1)                            { Unit vector along y }
  f=fs*normal( unit_y)                          { y-component }
  F_magnet=line_integral( f, 'circle', 'magnet')
EQUATIONS
  dx( Hy)- dy( Hx)=Jz
BOUNDARIES
region 'domain'  Jz=0  mu=mu0  start 'outer' (-r1,0) value(Az)=0
  arc(center=0,0) angle=360
region 'wire'  Jz=Jz0  mu=mu0  start (r0,-y0)    { Densify mesh }
  mesh_spacing=0.2*r0 arc( center=0,-y0) angle=360
region 'magnet' Jz=0  mu=1e3*mu0  start 'circle' (rc,y0)
  mesh_spacing=0.2*r0  arc( center=0,y0) angle=360
PLOTS
  contour( Bm) painted zoom(-4*y0,-4*y0,  8*y0,8*y0)
    report(F_wire)   report(F_magnet)
  contour( Bm) painted zoom(-4*y0,-4*y0,  8*y0,8*y0)
    fixed range(0, 3e-10)
END
```

In order to calculate the surface force on the magnetic cylinder, we first type the expression for fs in terms of the tangential and normal components of **H**. The force per unit area points along the normal,

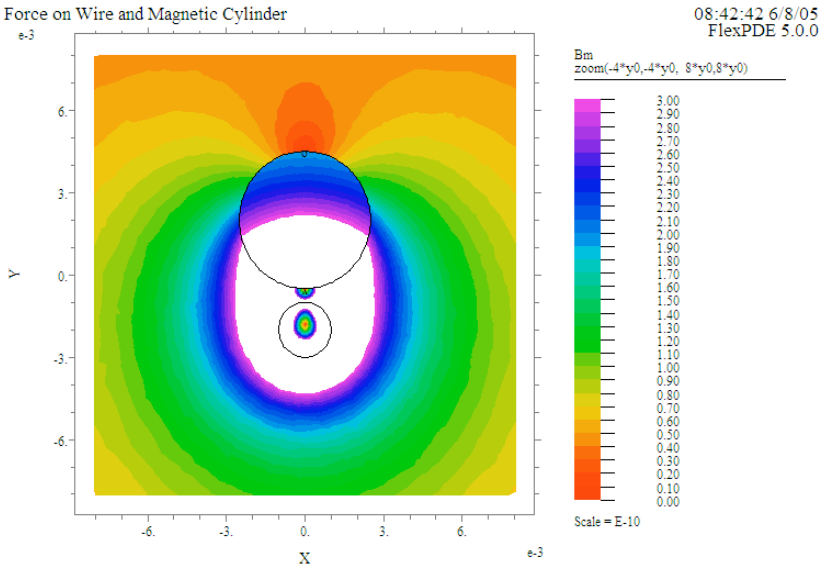
and we are interested in the *y* component. This we obtain by multiplying by normal(unit_y), which is the direction cosine. Then we integrate that over the circle, specifying the magnet side of the outline.

As seen below, the force on the wire and its reaction force on the magnetic cylinder evidently have closely the same magnitude, in spite of our using completely different expressions for the force.



exa124: Grid#2 p2 Nodes=6036 Cells=3007 RMS Err= 3.8e-5
F_wire= 3.157852e-16 F_magnet= -3.156327e-16 Integral= 3.908139e-14

The second figure (below) illustrates that the field generated by the wire is "shadowed" by the magnetic cylinder.



exa124: Grid#2 p2 Nodes=6036 Cells=3007 RMS Err= 3.8e-5
Integral= 3.908139e-14

Here, we used fixed range to exclude values outside the interval from zero to 3e-10 from the plot (white area). Thereby the minimum just above the magnetic cylinder becomes clearly visible.

Model of a Permanent Magnet

We may represent a permanent magnet in 2D by a rectangular cross-section in the (x, y) plane. In real space, this would correspond to a bar extending both ways in the *z*-directions, the bar being magnetized transversely in the direction of the *y*-axis. We specify a constant normal component, B_y, on the two *y*-directed faces of this bar and $B_x = 0$ for the other faces. The PDE domain is in a cross-section remote from both ends.

By the use of $\mathbf{B} = \nabla \times \mathbf{A}$ the boundary conditions for this magnet may be based on the relation (p.124●4)

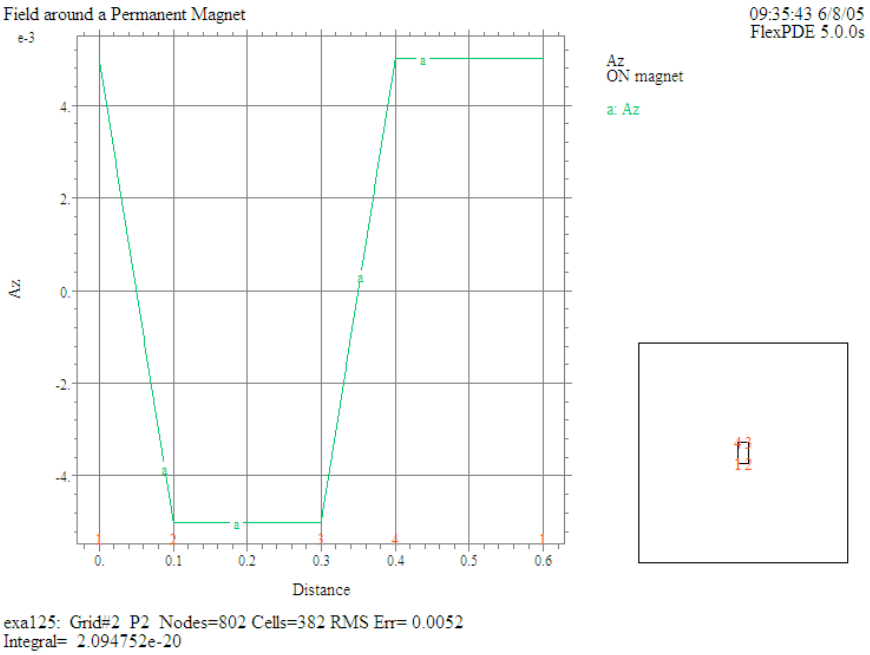
$$B_y = -\frac{\partial A_z}{\partial x}$$

on the sides parallel to the *x*-axis, and $B_x = 0$ on the other two sides. To simplify, we merge these two conditions into one by putting $A_z = -B_0 x$ over the rectangular boundary of the magnet.

```
TITLE   'Field around a Permanent Magnet'                    { exa125.pde }
SELECT                                                  { Student Version }
  errlim=3e-4    ngrid=1    spectral_colors
VARIABLES   Az
DEFINITIONS
  L=1.0   L0=0.05
  mu0=4*pi*1e-7   mu=mu0   Jz=0   B0=0.1    { Magnetic induction }
  Bex=dy(Az)   Bey=-dx(Az)
  B=vector( Bex, Bey)   Bm=magnitude( B)
  Hx=Bex/mu   Hy=Bey/mu   H=B/mu   Hm=Bm/mu
EQUATIONS
  dx( Hy)- dy( Hx)=Jz
BOUNDARIES
region 'domain'
  start 'outer' (-L,-L) value(Az)=0 line to (L,-L) to (L,L) to (-L,L) close
  start 'magnet' (-L0,-2*L0) value(Az)=-B0*x   { Exclude }
  line to (L0,-2*L0) to (L0,2*L0) to (-L0,2*L0) close
```

```
PLOTS
  elevation( Az) on 'magnet'                  { Verify Az on inner boundary }
  elevation( -normal( B)) on 'magnet'
  contour( Bm) log   vector( B) norm
  contour( Bm) painted zoom(-5*L0,-5*L0,  10*L0,10*L0)
  vector( B) norm zoom(-5*L0,-5*L0,  10*L0,10*L0)
END
```

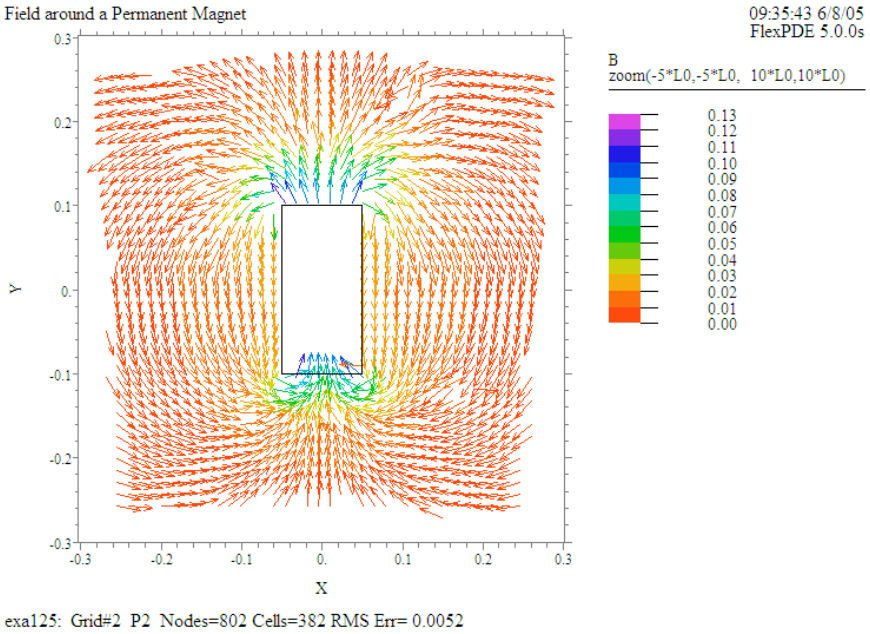
The first plot verifies the conditions we have imposed on the boundary corresponding to the magnet. The curve from 1 to 2 thus shows the linear variation of Az on the lower face of the magnet, and so on.



The second elevation plot shows the resulting variation of the normal component of B.

The full-scale vector plot of B looks a little like the dipole field from charged wires we studied on p.41. The following zoomed vector plot of B illustrates the field at closer range.

In this model of a permanent magnet we assumed B_y to be constant on the horizontal end surfaces. This is not the most realistic model one can imagine, and we shall illustrate another representation in the next chapter.



Exercises

❑ Change the circular outer boundary in *exa121* to a square of size 2*r1×2*r1. Make a full-scale vector plot of **B** to study the field near the boundary.

❑ Change the circular cross-section of the wire in *exa121* to a square of size 2*r0×2*r0.

❑ Recast *exa122* to study the field around two parallel wires conducting current in the *same* direction. To what extent are the results different?

❑ Integrate $\mathbf{B} \cdot d\mathbf{l}$ over a half-circle enclosing one of the wires in *exa122*. Compare by means of the pertinent relation on p.128.

❑ After the model of *exa124*, calculate the force between a current-carrying wire and a magnetic plate (virtually infinitely thick).

13 Magnetostatics in (ρ, z) Space

Already in the preceding chapter (*exa121*) we had an example with axial symmetry. In fact, there are two different types of such symmetry, both of which will be treated in the present chapter.

In cylindrical coordinates (ρ, φ, z) the relation between **H** and **J** becomes[3p705]

$$\nabla \times \mathbf{H} = \frac{1}{\rho}\begin{vmatrix} \mathbf{e}_\rho & \rho\mathbf{e}_\varphi & \mathbf{e}_z \\ \dfrac{\partial}{\partial\rho} & \dfrac{\partial}{\partial\varphi} & \dfrac{\partial}{\partial z} \\ H_\rho & \rho H_\varphi & H_z \end{vmatrix} = \mathbf{J} = \begin{Bmatrix} J_\rho \\ J_\varphi \\ J_z \end{Bmatrix} \qquad \bullet$$

This equation we shall now apply in two different ways.

Magnetic Field around a Wire

As the first application, we shall solve *exa121* as a one-dimensional problem, putting the z-axis along the wire. Only the third component (J_z) of the current density is then non-zero. The above vector equation thus reduces to

$$\frac{1}{\rho}\frac{\partial(\rho H_\varphi)}{\partial\rho} - \frac{1}{\rho}\frac{\partial H_\rho}{\partial\varphi} = J_z$$

where the second term vanishes under axial symmetry. The remainder is simply

$$\frac{1}{\rho}\frac{\partial(\rho H_\varphi)}{\partial\rho} = J_z \qquad \bullet$$

For the flux density **B** we still have the expression $\mathbf{B} = \nabla \times \mathbf{A}$ (p.124), which we expand in cylindrical coordinates as follows.

$$\mathbf{B} = \begin{Bmatrix} B_\rho \\ B_\varphi \\ B_z \end{Bmatrix} = \nabla \times \mathbf{A} = \frac{1}{\rho}\begin{vmatrix} \mathbf{e}_\rho & \rho\mathbf{e}_\varphi & \mathbf{e}_z \\ \dfrac{\partial}{\partial\rho} & \dfrac{\partial}{\partial\varphi} & \dfrac{\partial}{\partial z} \\ A_\rho & \rho A_\varphi & A_z \end{vmatrix} = \begin{Bmatrix} \dfrac{1}{\rho}\dfrac{\partial A_z}{\partial\varphi} - \dfrac{1}{\rho}\dfrac{\partial(\rho A_\varphi)}{\partial z} \\ \dfrac{\partial A_\rho}{\partial z} - \dfrac{\partial A_z}{\partial\rho} \\ \dfrac{1}{\rho}\dfrac{\partial(\rho A_\varphi)}{\partial\rho} - \dfrac{1}{\rho}\dfrac{\partial A_\rho}{\partial\varphi} \end{Bmatrix} \quad \bullet$$

Since **H** is perpendicular to the (ρ, z) plane, this is also true for $\mathbf{B} = \mu\mathbf{H}$. Hence only the middle component will be non-zero, leaving us with the expression

$$B_\varphi = \frac{\partial A_\rho}{\partial z} - \frac{\partial A_z}{\partial\rho} = -\frac{\partial A_z}{\partial\rho} \quad \bullet$$

The first term vanishes because there is no variation with z.

Substituting H_φ into the above equation for J_z we obtain the following 2nd-order PDE.

$$\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\frac{\rho}{\mu}\frac{\partial A_z}{\partial\rho}\right) = -J_z \quad \bullet$$

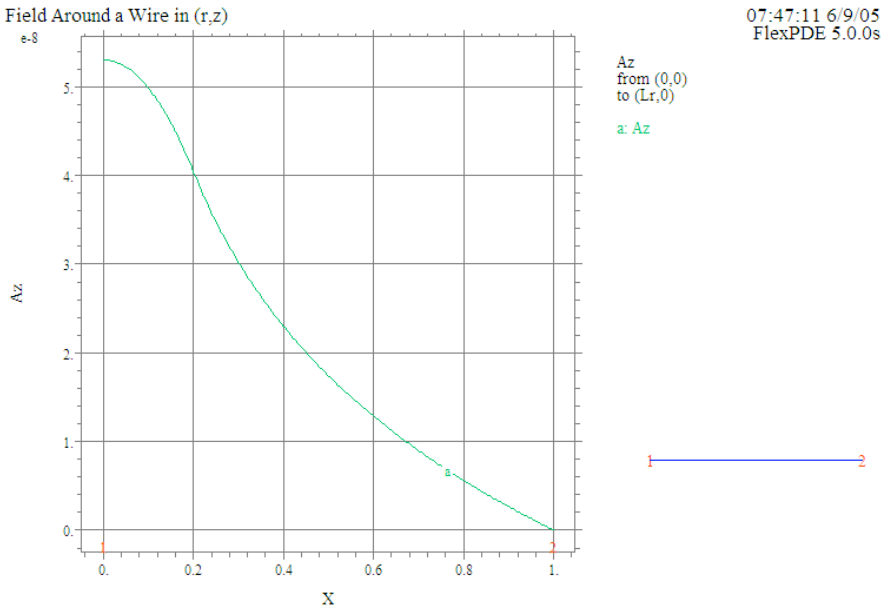
Although this equation only involves one independent variable, and hence is an *ordinary* differential equation (ODE), we may still solve it by means of FlexPDE. We declare the ODE by cartesian1, which assumes the notation x for the radial coordinate ρ. We arbitrarily specify a constant value of Az (0) on the right boundary.

```
TITLE   'Field Around a Wire in (r,z)'                    { exa131.pde }
SELECT                                                    { Student Version }
  errlim=1e-12    ngrid=1    spectral_colors
COORDINATES   cartesian1
VARIABLES   Az                        { Magnetic vector potential }
DEFINITIONS                                               { SI units }
  mu0=4*pi*1e-7    mu=mu0    Jz0=1.0                      { Current density }
  r0=0.2    Lr=1.0    Lz=0.01
  B_phi=-dx( Az)    H_phi=1/mu*B_phi
  Jz    B_ex                                              { Declared only }
EQUATIONS
  1/r*dx( r/mu*dx( Az))=-Jz                               { x=radius }
BOUNDARIES
region 'domain'  Jz=0    B_ex=mu*Jz0*r0^2/(2*r)           { Exact solution }
```

```
  start 'outer' (0)  line to (Lr) point value(Az)=0
region 'wire' Jz=Jz0  B_ex=mu*Jz0*r/2
  start (0) line to (r0)
PLOTS
  elevation( Az) from (0,0) to (Lr,0)
  elevation( B_phi, B_ex) from (0,0) to (Lr,0)
  elevation( B_phi- B_ex) from (0,0) to (Lr,0) report(globalmax( B_phi))
END
```

In this example, we can employ the same exact expression for B_phi as in *exa121*, remembering that rad corresponds to r in cylindrical coordinates.

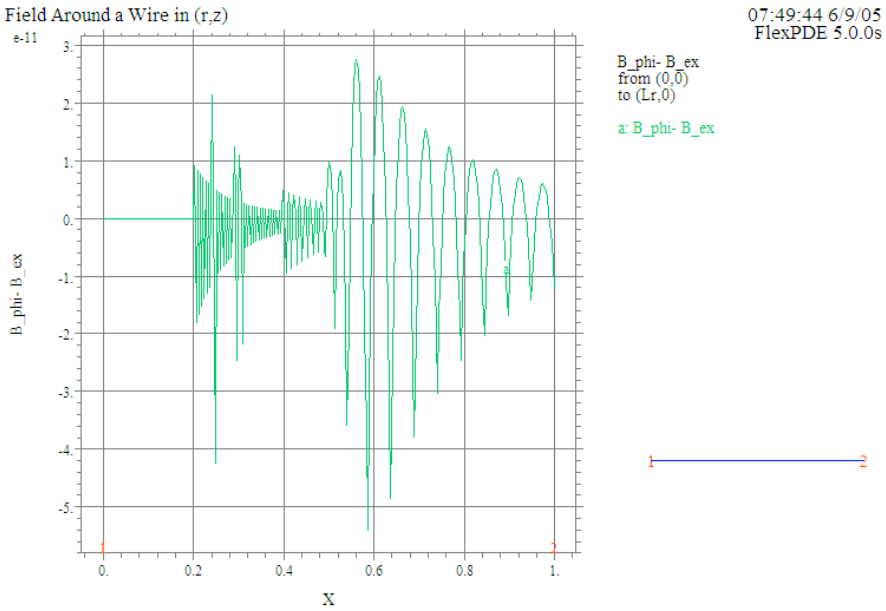
The plot of Az below is evidently similar to the right half of the one we obtained in (x, y).



exa131: Grid#5 P2 Nodes=101 Cells=50 RMS Err= 4.e-13
Integral= 2.178168e-8

The second plot (not shown here) compares the numerical value of B_phi with the analytical one. The integrals evidently agree within 0.004%.

The deviation from the exact expression is shown directly in the next figure. We notice that the maximum error is about 5e-11, or in relative terms about 0.1% of the largest value of B_phi.



The maximum deviation thus seems to be orders of magnitude larger than requested in errlim. The latter refers to the dependent variable (Az), however, and it is clear from the deviation plot that the differentiation adds violent oscillation in the region where the solution has high curvature.

It would be more fair to compare Az directly to the exact expression. In fact, we may integrate the expressions for B_ex by means of the relation $H_\varphi = -(1/\mu)\partial A_z/\partial\rho$ to obtain Az in each region. Doing this, we need to determine the arbitrary constant such that Az becomes zero at the far boundary. Then we must add the appropriate constant expression to the solution inside the wire in order to make the two functions join continuously. Comparing the resulting expressions for Az to the solution is left as an exercise at the end of the chapter.

Field along the (ρ,z) Plane

We shall now deal with the situation where the current density **J** is perpendicular to the (ρ,z) plane. The middle equation in p.141●1 then reduces to

$$\frac{\partial H_\rho}{\partial z} - \frac{\partial H_z}{\partial \rho} = J_\varphi \qquad \bullet$$

Under axial symmetry, the derivatives with respect to φ vanish, and the relation $\mathbf{B} = \nabla \times \mathbf{A}$ (p.142●1) expands into

$$B_\rho = -\frac{1}{\rho}\frac{\partial(\rho A_\varphi)}{\partial z}, \qquad B_z = \frac{1}{\rho}\frac{\partial(\rho A_\varphi)}{\partial \rho} \qquad \bullet$$

Substituting the components of $\mathbf{H} = \mathbf{B}/\mu$ into the above equation for J_φ we obtain a 2nd-order PDE of the Poisson type.

$$\frac{\partial}{\partial \rho}\left(\frac{1}{\mu\rho}\frac{\partial(\rho A_\varphi)}{\partial \rho}\right) + \frac{\partial}{\partial z}\left(\frac{1}{\mu}\frac{\partial A_\varphi}{\partial z}\right) = -J_\varphi$$

Since the preceding 1st-order PDE is simpler and directly emerges from the Maxwell relation, we shall use that in this chapter. FlexPDE automatically replaces the components H_ρ and H_z by derivatives of A_φ when we run the descriptor.

Simple Magnet Coil

Is there a practical device where an electrical current can run in the direction of increasing or decreasing φ? A good approximation to a single, current-carrying turn is a number of layers of wire, wound on a circular cylinder. In the (ρ, z) plane we would only see the cross section of this coil.

Just as in the previous chapter the normal field component B_n across the distant boundary will vanish, and this we achieve by imposing value(A_phi)=0. The same condition applies on the axis, in view of the axial symmetry of the radial field component B_r.

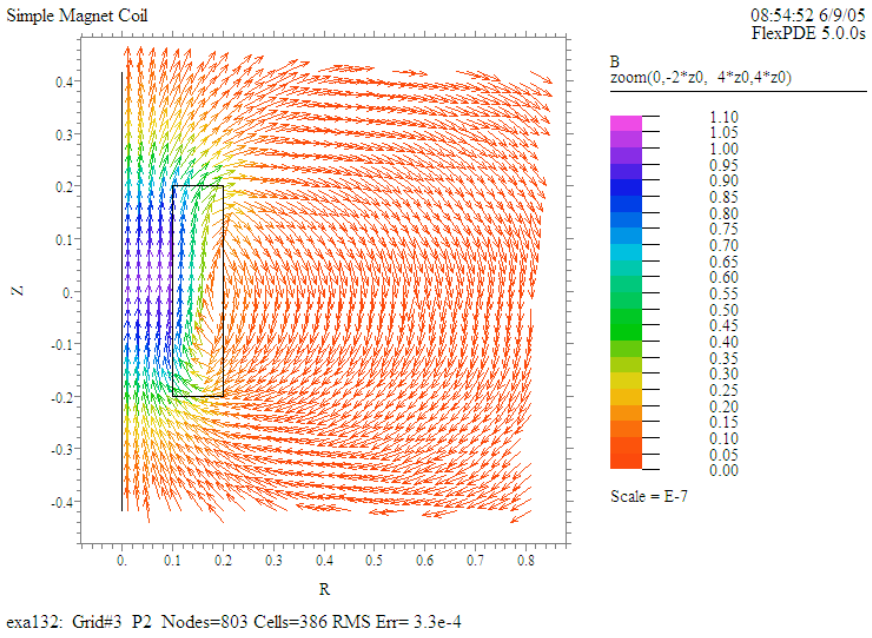
```
TITLE   'Simple Magnet Coil'                                { exa132.pde }
SELECT    errlim=3e-4    ngrid=1    spectral_colors
COORDINATES   ycylinder( 'r', 'z')                    { Student Version }
VARIABLES   A_phi
DEFINITIONS
  r1=0.1    r2=0.2    z0=0.2    L=1.0
```

```
  mu0=4*pi*1e-7     mu=mu0     J_phi
  Br=-dz(A_phi)     Bz=1/r*dr(r*A_phi)
  B=vector( Br, Bz)     Bm=magnitude( B)
  Hr=Br/mu     Hz=Bz/mu     H=B/mu     Hm=Bm/mu
EQUATIONS
  dz( Hr)- dr( Hz)=J_phi
BOUNDARIES
region 'domain' J_phi=0
  start(0,-L) value(A_phi)=0 line to (2*L,-L) to (2*L,L)  to (0,L) close
region 'coil'  J_phi=1.0                                          { Current density }
  start(r1,-z0)  line to (r2,-z0) to (r2,z0) to (r1,z0) close
PLOTS
  contour( Bm) log     vector( B) norm
  contour( Bm) painted zoom(0,-2*z0,  4*z0,4*z0)
  vector( B) norm zoom(0,-2*z0,  4*z0,4*z0)
END
```

The full-scale vector plot of **B** shows the consequences of our boundary conditions. A few arrows seem to cross the boundary, but we must remember that the direction refers to the point at the base of the arrow, not at the tip.

The painted contour plot of Bm illustrates that the highest value occurs at half-height and just inside the coil. There is also a minimum in the outer layers.

The zoomed vector plot below shows the details of the near field.



Even if the magnitude of **B** is not constant inside the coil, the direction of the field seems to be essentially parallel to the axis.

Here, there is a point to be noticed about signs. When we use an (x, y, z) system, the z-axis points out of the figure on the screen. Applying the same rule to the present (ρ, φ, z) system, we find that the φ-axis points *into* the figure. This accounts for the sense of circulation of **B**.

Helmholtz Coil

We next explore the field inside an ingenious device known as the Helmholtz coil. It consists of two identical, simple coils on the same axis, with a mean spacing equal to the mean radius. Such a system of coils is known to create a highly constant field in the region of the center of symmetry.

Much of the descriptor *exa132* may be kept, as the following file demonstrates.

```
TITLE 'Helmholtz Coil'        { Professional Version }      { exa133.pde }
SELECT    errlim=1e-3  ngrid=1    spectral_colors
COORDINATES  ycylinder( 'r', 'z')
VARIABLES  A_phi
DEFINITIONS
  r1=0.15    r2=0.25    z0=0.05    d0=0.1    L=1.0
  mu0=4*pi*1e-7    mu=mu0    J_phi
  Br=-dz(A_phi)    Bz=1/r*dr(r*A_phi)
  B=vector( Br, Bz)    Bm=magnitude( B)
  Hr=Br/mu    Hz=Bz/mu    H=B/mu    Hm=Bm/mu
  mesh_spacing=0.1*d0*(1+(r/d0)^2+(z/d0)^2)       { Maximum size }
EQUATIONS
  dz( Hr)- dr( Hz)=J_phi
BOUNDARIES
region 'domain'  J_phi=0
  start(0,-L) value(A_phi)=0 line to (2*L,-L) to (2*L,L)  to (0,L) close
region 'coil1' J_phi=1.0
  start(r1,-d0-z0) line to (r2,-d0-z0) to (r2,-d0+z0) to (r1,-d0+z0) close
region 'coil2' J_phi=1.0
  start(r1,d0-z0) line to (r2,d0-z0) to (r2,d0+z0) to (r1,d0+z0) close
PLOTS
  contour( Bm) log    contour( Bm) painted    vector( B) norm
```

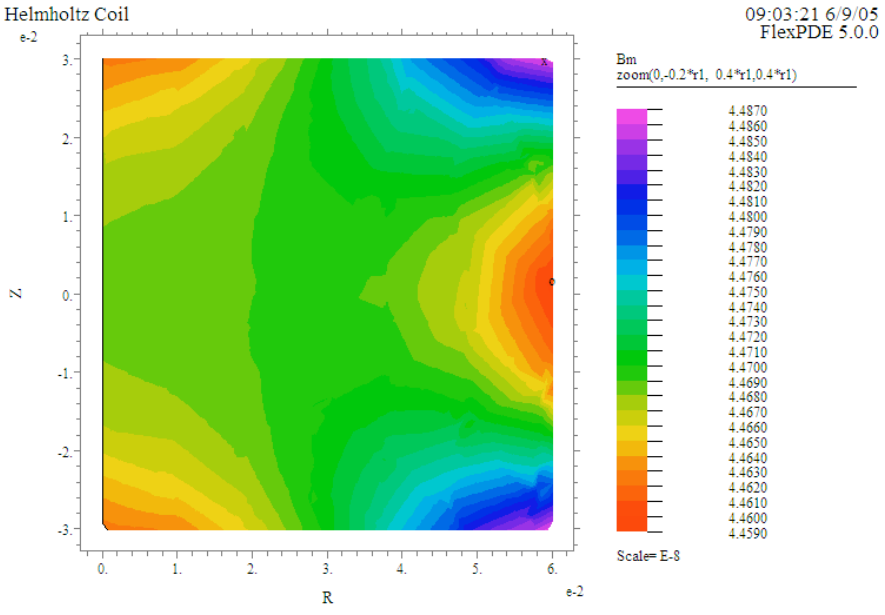
```
  contour( Bm) painted zoom(0,-0.2*r1,  0.4*r1,0.4*r1)
  vector( B) zoom(0,-0.2*r1,  0.4*r1,0.4*r1)
END
```

In the present example, it is of interest to densify the grid inside the coils and near the axis, to illustrate the constancy of Bm. We would also like to favor a symmetric distribution of cells. Although FlexPDE automatically splits cells where needed, the program cannot know what regions we are primarily interested in.

The command *mesh_spacing* lets us specify a *maximum* size of the cells as a function of the coordinates, and the program subdivides these cells where it is necessary to meet the error limit requirement. We specify the size by a function that takes a small value near the center of the double coil and rapidly increases outside the coil.

The first plots of Bm roughly confirm the constancy of the field close to the central maximum. It also shows that the values of Bm fall off rapidly toward the outer boundary.

The following zoomed, painted contour plot presents the variation of Bm in the interior of the coil system. Larger values occur as we approach a coil. There is a central region, however, where Bm is highly constant even at rather large radii.



exa132: Grid#2 p2 Nodes=1341 Cells=648 RMS Err= 5.4e-4
Vol_Integral= 3.044659e-11

The zoomed vector plot shows that the field over the same region is parallel, as far as can be discerned.

Helmholtz found the optimum ratio of distance-to-radius assuming vanishingly small coil cross-sections. In our realistic case of finite cross-sections, the ratio yielding the most uniform field may well be slightly different, and one of the exercises asks you to find a better ratio.

Forces on Coils

We may calculate the force on a coil by additional lines in the descriptor. The Lorentz formula $\mathbf{F} = \mathbf{J} \times \mathbf{B}$ gives us the components $F_\rho = J_\varphi B_z$ and $F_z = -J_\varphi B_\rho$ in the present coordinates and with only J_φ being non-zero. For reasons of symmetry, the net radial force becomes zero, but the radial component does produce hoop stress that deforms the coil.

The *z* component, on the other hand, yields net forces of opposite signs on the two coils. In order to calculate the force on the upper coil, we modify *exa133* as follows. The *summary* groups the *report* commands, which will present the numerical results as a separate list.

```
TITLE   'Helmholtz Coil, Force'                              { exa133a.pde }
...                                        { Professional Version }
  area=(r2-r1)*2*z0   J0=1.0          { Cross-sectional area and current }
EQUATIONS
...
PLOTS
summary
  report( vol_integral( J_phi*Bz,  'coil2'))  as 'Radial force, integrated'
  report( vol_integral( -J_phi*Br, 'coil2')) as 'Axial force on coil 2'
  report( -2*pi*(r1+r2)/2* mu*(J0*area)/(2*pi*2*d0)* (J0*area))
    as 'Estimate'                  { Length*current*mu/2/pi*current/distance }
END
```

We obtain the elementary estimate of the attractive force by replacing each curved coil wire by a straight wire, which of course is a very rough approximation. The magnetic flux density at a distance *d* from a straight wire is[3p230] $|\mathbf{B}| = \mu I/(2\pi d)$, and since that field is perpendicular to the other wire we just need to multiply by the current *I* to obtain the Lorentz force.

```
Helmholtz Coil, Force                                    12:45:52 6/9/05
                                                         FlexPDE 5.0.0

  SUMMARY

  Radial force, integrated=  2.190333e-10
  Axial force on coil 2= -7.316650e-11
  Estimate= -1.256637e-10
```

The radial force turns out to be positive (repulsive), tending to expand the coil. The axial force on coil2 has a negative value, which means that the two coils attract each other.

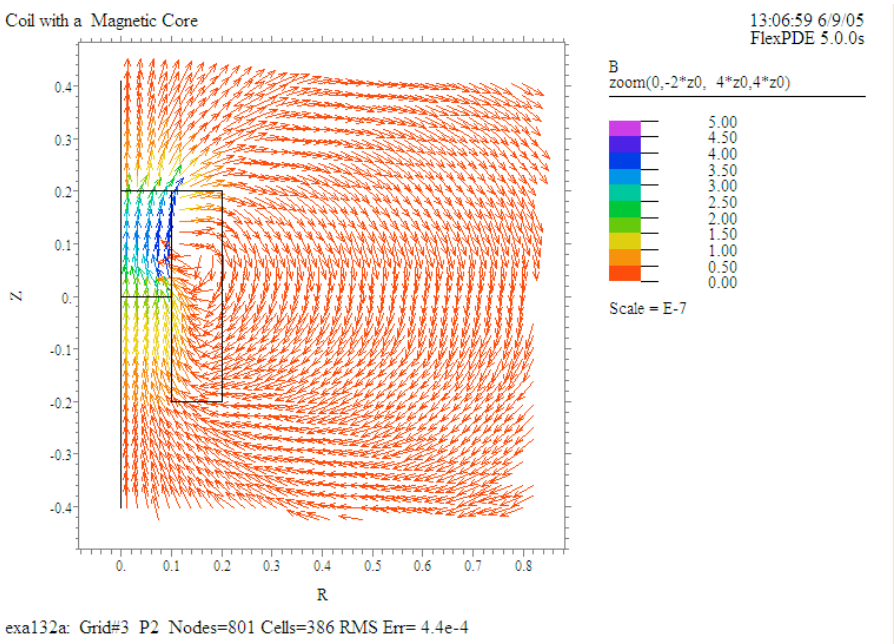
Coil with a Magnetic Core

So far we have assumed that coils are surrounded by air. It is easy to explore the effects of inserting a core of a magnetic material where **B** is proportional to **H**. Let us use the descriptor for the simple magnet coil (*exa132*) as a template and add a region containing the magnetic core. For comparison we let the core extend over only half of the coil.

```
TITLE   'Coil with a  Magnetic Core'                    { exa132a.pde }
…              { Student Version }
  { Keep 'domain' and 'coil' }
region 'core'  J_phi=0  mu=1000*mu0     { Overwrite default value of mu }
   start(0,0) line to (r1,0) to (r1,z0) to (0,z0) close
PLOTS
  contour( Hm) painted
  contour( Hm) painted zoom(0,-2*z0,  4*z0,4*z0)
  elevation( Hm) from (0,-0.2) to (0,0.2)
  elevation( Bm) from (0,-0.2) to (0,0.2)
…
```

Inspecting the plots of Hm and Bm we find that Hm becomes very small inside the core, while Bm takes a maximum in the magnetic material.

The following vector plot of B illustrates new features in the field pattern. In the winding, the directions shown by the arrows are now highly asymmetric. The colors indicate that Bm is much larger in the core than in the other half of the space inside coil.



Force between a Coil and a Magnetic Ellipsoid

We have already used the expression for the force on the surface of a magnetic cylinder (p.135), i.e.

$$\mathbf{f}_s = \frac{1}{2}(\mu_2 - \mu_1)\left(H_{t2}^2 + \frac{\mu_2}{\mu_1} H_{n2}^2 \right)\mathbf{n} \qquad \bullet$$

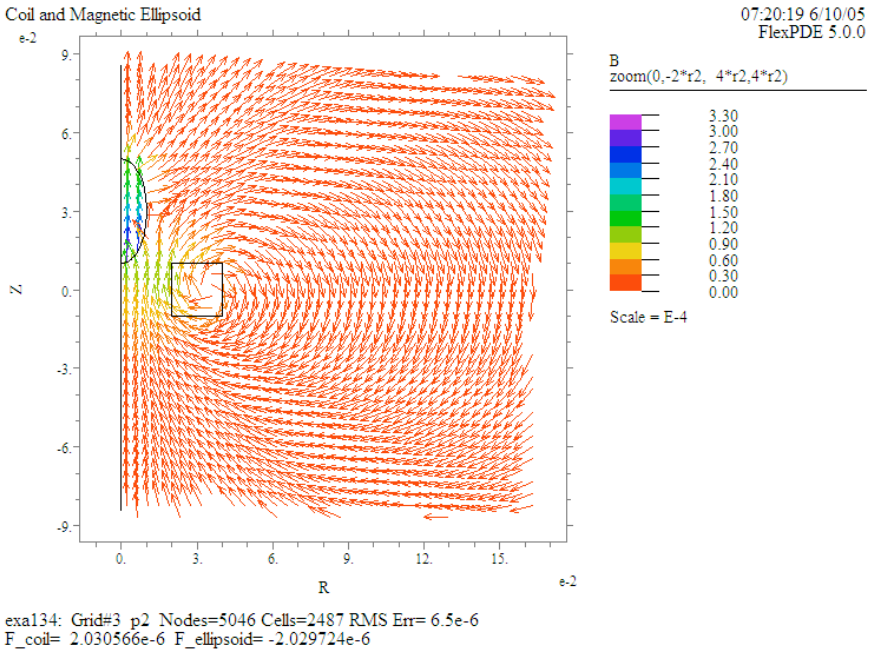
where the field components refer to the *interior* of the material.

Now we shall apply this result in an axially symmetric situation. In the following example, a current through a coil supplies the magnetic field, acting on an ellipsoidal linear-magnetic object.

```
TITLE   'Coil and Magnetic Ellipsoid'                    { exa134.pde }
SELECT                                                  { Professional Version }
  errlim=1e-5    ngrid=1    spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES     A_phi
DEFINITIONS
  r0=0.02    r00=0.01    z0=0.01    r1=0.02    z1=0.01    r2=0.04   L=1.0
  mu0=4*pi*1e-7     mu     J_phi
  Br=-dz(A_phi)     Bz=1/r*dr(r*A_phi)
  B=vector( Br, Bz)     Bm=magnitude( B)
  Hr=Br/mu     Hz=Bz/mu     H=B/mu     Hm=Bm/mu
  F_coil=vol_integral( -J_phi*Br,'coil')
```

```
  fs=0.5*(mu-mu0)*[ tangential(H)^2+ mu/mu0*normal(H)^2]
  unit_z=vector(0,1)                          { Unit vector field along z }
  f=fs*normal( unit_z)                        { z-component }
  F_ellipsoid=surf_integral(  f,'ellipse')
EQUATIONS
  dz( Hr)- dr( Hz)=J_phi
BOUNDARIES
region 'domain'   mu=mu0   J_phi=0
  start(0,-L) value(A_phi)=0 line to (2*L,-L) to (2*L,L)  to (0,L) close
region 'coil'  mu=mu0   J_phi=1e4          { Current density }
  start(r1,-z1) line to (r2,-z1) to (r2,z1) to (r1,z1) close
region 'ellipsoid'   mu=1000*mu0   J_phi=0   start 'ellipse' (0,z0)
  arc( center=0,z0+r0) to (r00,z0+r0) to (0,z0+2*r0) line to close
PLOTS
  contour( Bm) painted zoom(0,-2*r2,  4*r2,4*r2)
  vector( B) norm zoom(0,-2*r2,  4*r2,4*r2)
    report(F_coil)   report(F_ellipsoid)
END
```

The next figure shows the field **B** in the vicinity of the coil and the ellipsoid. The maximum magnitude occurs at the lower end of the prolate ellipsoid, as could be expected.



We conclude that the magnitude of the forces on the coil and on the ellipsoid are equal to better than 0.04 %. It is worth noting that the

two expressions used for comparison are quite different, but yield virtually the same result.

Magnetostatic Field Energy

The expression for the magnetic field energy[3p280] is given by

$$W = \frac{1}{2}\iiint \mathbf{B}\cdot\mathbf{H}\,dv \ , \qquad \bullet$$

which is closely analogous to what we used in electrostatics. We shall illustrate this concept by a current-carrying coil. Energy is required to establish a current through this coil, even if the wire is super-conducting. We shall compare the energy of the field with the electric energy required to set it up.

It is easy to calculate the magnetic field energy W, and we already performed a similar operation in the case of the plate capacitor (p.118). The electric work necessary to produce this field is related to the integral form of one of Maxwell's equations[3p323], i.e.

$$V = \oint \mathbf{E}\cdot d\,\mathbf{l} = -\iint \frac{\partial \mathbf{B}}{\partial t}\cdot d\,\mathbf{s} = -\frac{\partial \phi}{\partial t}$$

This equation means that the changing flux ϕ through the coil creates a voltage V around the wire loop, and this we need to *overcome* in order to produce the field.

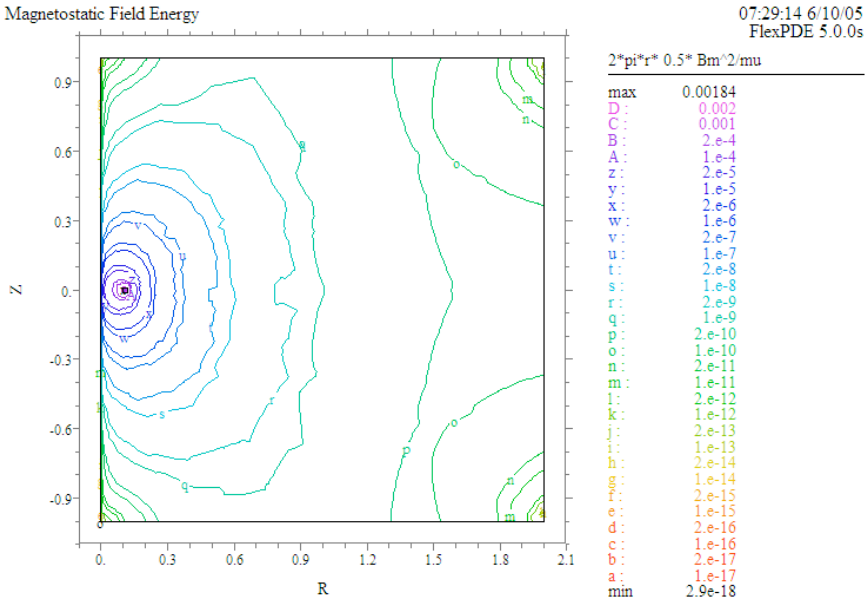
In setting up the field we have to increase the current $I(t)$ up to the final value I_0, and during this process we may assume that the flux remains proportional to the current, i.e. $\phi = c\,I(t)$. The total electric work may now be written

$$W_0 = \int V\,I\,dt = -\int \frac{d\,\phi}{d\,t} I\,dt = c\int I\,\frac{d\,I}{d\,t}dt = c\int I\ dI = \frac{cI_0^2}{2} = \frac{\phi_0 I_0}{2}$$

which we use for comparison with W. In the descriptor we obtain the final flux ϕ_0 by an integral of B_z over a flat surface through the symmetry plane of the coil.

```
TITLE   'Magnetostatic Field Energy'                    { exa135.pde }
SELECT                                                  { Student Version }
  errlim=1e-3     ngrid=1     spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES   A_phi
DEFINITIONS
  r1=0.1     r2=0.12     z0=0.01     L=1.0
  mu0=4*pi*1e-7     Jd=1e4     mu=mu0     J_phi
  Br=-dz(A_phi)     Bz=1/r*dr(r*A_phi)
  B=vector( Br, Bz)     Bm=magnitude( B)
  Hr=Br/mu   Hz=Bz/mu     H=B/mu     Hm=Bm/mu
  W=vol_integral( 0.5*Bm*Hm)                    { Field energy }
  flux1=surf_integral( Bz, 'radius1')           { Flux through coil }
  flux2=surf_integral( Bz, 'radius2')           { Flux through coil }
  J0=Jd*(r2-r1)*(2*z0)                          { Current }
  W1=flux1*J0/2   W2=flux2*J0/2                 { Electric work }
EQUATIONS
  dz( Hr)- dr( Hz)=J_phi
BOUNDARIES
region 'domain' J_phi=0
  start  'outer'  (0,-L)     value(A_phi)=0
  line to (2*L,-L)  to (2*L, L)  to (0, L) close
region 'coil' J_phi=Jd                          { Current density }
  start(r1,-z0) line to (r2,-z0) to (r2,z0) to (r1,z0) close
feature
  start 'radius1' (0,0) line to (r1, 0)
  start 'radius2' (0,0) line to (r2, 0)
PLOTS
  contour( 2*pi*r* 0.5* Bm^2/mu) log  report( W) report( W1) report( W2)
END
```

In the plot below we have multiplied the energy density by 2*pi*r in order to compare the contribution of various regions to the total energy. Judging from the maximum and minimum values on the logarithmic contour plot the region for integration seems to be sufficiently large to let us approximate the total energy value by an integration over the domain.



We encounter a trivial difficulty when we wish to estimate the electric work W_0 required to set up the magnetic field. It is easy enough to integrate to obtain the flux enclosed by an infinitely thin wire, but FlexPDE can only handle coils of finite cross-sections. What we can do is to integrate from 0 to r1 to obtain the value W_1 and then from 0 to r2 to obtain W_2. We expect the results W_1 and W_2 to bracket the magnetic field energy W.

From the results reported we gather that the limiting estimates for the magnetization energy indeed bracket the energy obtained by integrating the magnetic energy density over space. The last value on the bottom line is for an integral that automatically includes the 2*pi*r factor; hence this factor enters twice, which makes the result for that integral irrelevant.

Simple Model of a Permanent Magnet

We already modeled a permanent magnet on p.138, and we may extend the same idea to cylindrical coordinates. In order to obtain constant B_z at the circular end of the magnet we use the expression (p.142●1)

$$B_z = \frac{1}{\rho}\frac{\partial(\rho A_\varphi)}{\partial\rho} = B_{z0}$$

Expanding the derivative we are faced with a 1st-order differential equation, which only involves ρ as the independent variable. It is easy to verify that

$$A_\varphi = \tfrac{1}{2} B_{z0}\rho$$

is a solution, and we may use that expression for the end face.

It is easy to type a descriptor analogous to *exa125*, valid for (ρ, z). We shall not make this an example, however, since there is a more realistic model, which we shall apply in the following section.

Permanent Magnetization

Although the simple model is useful in many cases, a real permanent magnet does not have uniform **B**, but uniform magnetization **M**. This means that the internal magnetic flux density may change under the influence of external magnetic fields.

In order to treat the case of a magnet bar having constant **M** throughout its volume, we need to modify our theoretical approach slightly. The equation $\nabla \times \mathbf{H} = \mathbf{J}$ remains valid and yields the PDE as before. From $\mathbf{B} = \nabla \times \mathbf{A}$ we still obtain the flux density components B_r and B_z. The field **H**, however, must be defined by means of the relation

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M}) \qquad \bullet$$

where we have specified the magnetization **M** over the region of the magnet.

```
TITLE   'Permanent Magnetization'                    { exa136.pde }
SELECT                                          { Professional Version }
  errlim=1e-4     ngrid=1     spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES   A_phi
DEFINITIONS                                     { Uniform magnetization }
  L=1.0     r0=0.1    Lz=0.1     J_phi=0     Mz0     Mz00=1e5
  mu0=4*pi*1e-7     mu=mu0
  Br=-dz(A_phi)     Bz=1/r*dr(r*A_phi)
```

```
  B=vector( Br, Bz)     Bm=magnitude( B)
  Hr=Br/mu0-0     Hz=Bz/mu0-Mz0          { Mr0=0 }
  H=vector( Hr, Hz)   Hm=magnitude( H)   Mz=Bz/mu0-Hz
  z1=z+Lz     Bz_ex=                                  { Continued on next line }
  mu0*Mz00/2*[ z1/sqrt(z1^2+r0^2)- (z1-2*Lz)/sqrt((z1-2*Lz)^2+r0^2)]
EQUATIONS
  dz( Hr)- dr( Hz)=J_phi
BOUNDARIES
region 'domain' Mz0=0  start 'outer' (0,-L) value(A_phi)=0
  line to (2*L,-L) to (2*L,L) to (0,L) line to close
region 'magnet' Mz0=Mz00
  start (0,-Lz) line to (r0,-Lz) to (r0,Lz) to (0,Lz)
  mesh_spacing=0.1*Lz line to close
PLOTS
  contour( Hz) painted     contour( Mz) painted
  contour( Bz) painted
  elevation( Bz, Bz_ex) from (0,-2*Lz) to (0,2*Lz)
  vector( H) norm zoom(0,-3*Lz,  6*Lz,6*Lz)
  vector( B) norm zoom(0,-3*Lz,  6*Lz,6*Lz)
  contour( Hz+Mz) painted zoom(0,-3*Lz,  6*Lz,6*Lz)
  elevation( Hz+Mz) from (0,0) to (2*r0,0)
END
```

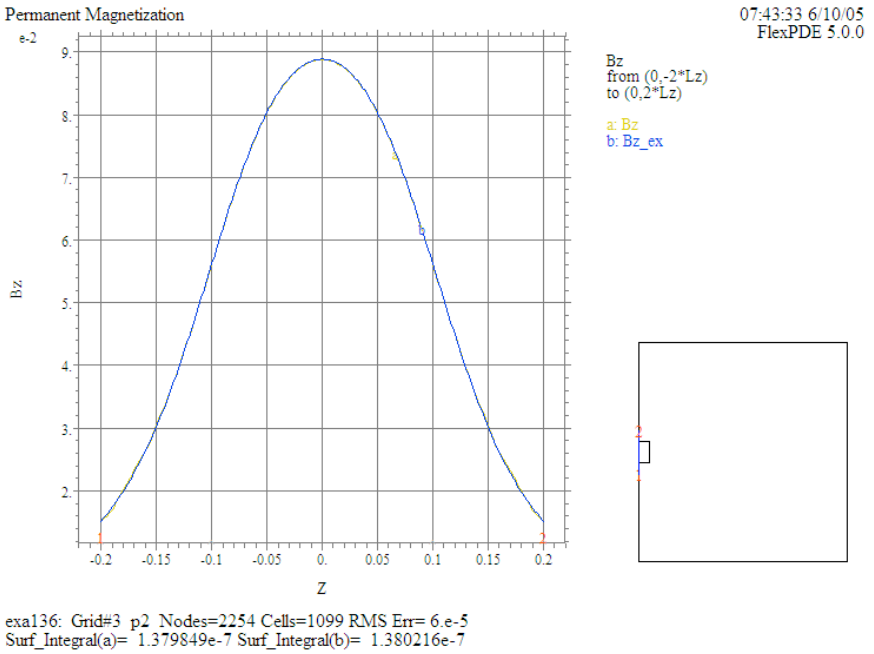
There is a simple analytic expression for Bz on the axis[3p247], specifically

$$B_z = \frac{\mu_0 M_0}{2}\left[\frac{z}{\sqrt{z^2+b^2}} - \frac{z-L}{\sqrt{(z-L)^2+b^2}}\right]$$

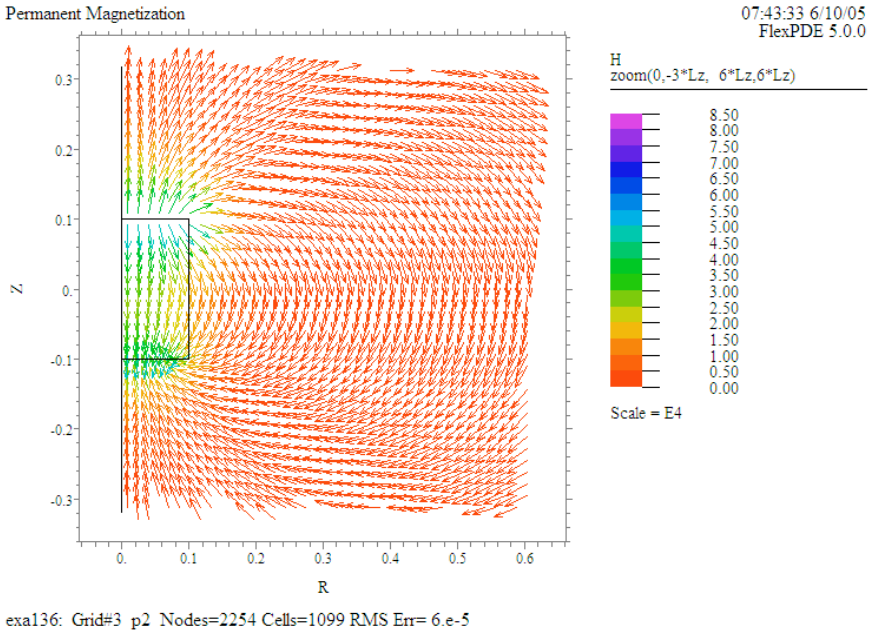
Here we must convert to our notation by the substitutions $z \to$z+Lz, $L \to$2*Lz, and $b \to$r0. In order to obtain a finer mesh close to the axis we use node_spacing on the left boundary of the magnet.

The plot below compares the solution to the analytic expression for Bz. The agreement is very good along the z-axis, as judged by the integrals, and we may thus have confidence in the FEA solution.

It is evident from the plots that Bz is far from constant inside the magnet.



The next figure shows the vector plot of **H**. Here we notice that the field inside the magnet acts in a direction opposite to that of the magnetization. Hence the internal **H** is known as the *demagnetizing field*.



The last two plots of Hz+Mz will be used for comparison in the next section, where we model the permanent magnet as a thin magnet coil.

Permanent Magnet as a Current Sheet

An accepted emulation of a permanent magnet bar is a thin coil of the same radius and length as the magnet. Atomic dipole moments may be seen as minute current loops, stacked inside the volume of the material. The currents in these loops cancel in the interior, and the global result is a current sheet on the cylindrical surface. According to this model, the current per unit *length* of the coil is equal to M_{z0}. To convert to current per unit *area* we must divide by the thickness d of the winding.

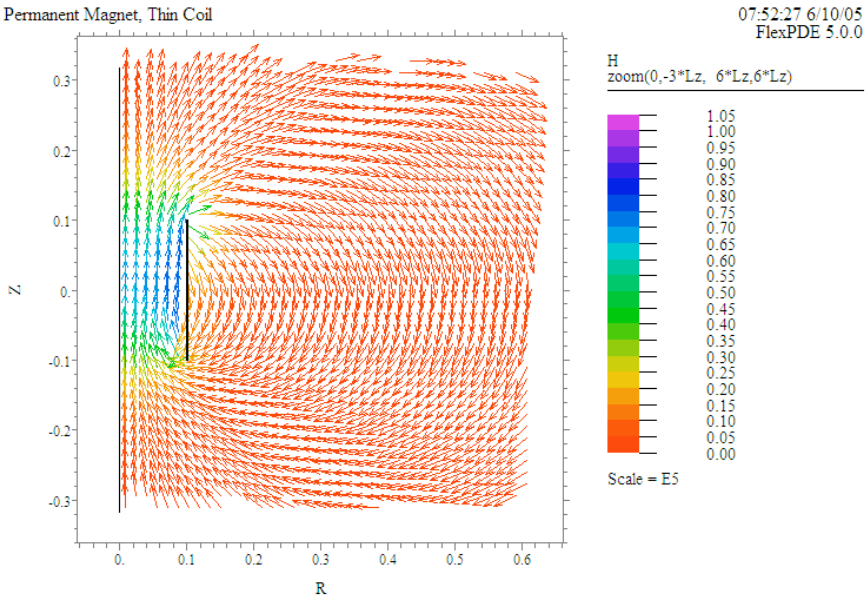
We shall now explore this model of a magnet by the following descriptor, which is based on *exa132*. The thickness d is 1% of the mean radius rm. Since the results are expected to approach those of *exa136*, we include the analytic expression for Bz on the axis.

```
TITLE   'Permanent Magnet, Thin Coil'                    { exa137.pde }
SELECT                                          { Professional Version }
  errlim=1e-4     ngrid=1     spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES   A_phi
DEFINITIONS
  L=1.0     d=1e-3    r1=0.1-d/2     r2=0.1+d/2    rm=(r1+r2)/2     Lz=0.1
  mu0=4*pi*1e-7     mu=mu0     J_phi
  Br=-dz(A_phi)     Bz=1/r*dr(r*A_phi)
  B=vector( Br, Bz)     Bm=magnitude( B)
  Hr=Br/mu     Hz=Bz/mu     H=B/mu     Hm=Bm/mu
  Mz00=1e5     z1=z+Lz   Bz_ex=          { Exact solution on the axis }
  mu0*Mz00/2*[ z1/sqrt(z1^2+rm^2)- (z1-2*Lz)/sqrt((z1-2*Lz)^2+rm^2)]
  Mz=Bz/mu0-Hz
  mesh_spacing=0.1*Lz*(1+(r/Lz/0.001)^2+(z/Lz/3)^2)
EQUATIONS
  dz( Hr)- dr( Hz)=J_phi
BOUNDARIES
region 'domain'  J_phi=0  start(0,-L) value(A_phi)=0
  line to (2*L,-L) to (2*L,L)  to (0,L) close
region 'coil'  J_phi=Mz00/d                                  { Current density }
  start(r1,-Lz)  line to (r2,-Lz) to (r2,Lz) to (r1,Lz) close
PLOTS
  contour( Hz)  painted zoom(0,-3*Lz,  6*Lz,6*Lz)
  contour( Mz) painted zoom(0,-3*Lz,  6*Lz,6*Lz)
  contour( Bz) painted zoom(0,-3*Lz,  6*Lz,6*Lz)
```

```
  elevation( Bz, Bz_ex) from (0,-2*Lz) to (0,2*Lz)
  vector( H) norm zoom(0,-3*Lz,  6*Lz,6*Lz)
  vector( B) norm zoom(0,-3*Lz,  6*Lz,6*Lz)
  elevation( Hz) from (0,0) to (2*r2,0)
END
```

We may compare these plots with those of *exa136*, using the task strip at the top of the Editor to swap from any given plot to the corresponding one from the other file.

Comparing the contour plots we notice that the one for Bz is very similar in the two cases, whereas the plots of Hz are clearly different. A consequence of the present model is that the demagnetizing field Hz inside the magnet vanishes, as shown by the next plot.



The cause of this discrepancy is that the present model does not make use the magnetization **M**. In fact, **H**+**M** in *exa136* corresponds to **H** in the present file. If we compare the contour plot Hz here to that of Hz+Mz in the preceding example, we find that they are similar.

The magnetic flux density Bz on the axis again agrees very well with the analytic expression copied from the preceding file.

The final elevation plot of Hz exhibits a sudden drop in going through the coil, in agreement with the tangential boundary condition for **H**. The same phenomenon was apparent in the elevation plot of Hz+Mz in the preceding example.

If we wish to calculate the force on a permanent magnet, we may exploit this model. The problem is then reduced to computing the force on a coil carrying the appropriate current.

Exercises

❑ Integrate the expression for B_φ (p.142●2) to obtain an analytic solution for Az, using B_ex in *exa131*. First chose the constant of integration such that the potential becomes zero at the right end. Then add a term to the expression for Az inside the wire, so that the solution become continuous at the interface. Compare to the numerical result and calculate the RMS error by integration over the rectangular space.
❑ Modify *exa132* to obtain the field for a coil of circular cross-section. Use a small radius with the Professional Version. Compare with an elementary estimate.
❑ Change the spacing between the Helmholtz coils in *exa133* to slightly smaller and larger values to find out how the constancy of the central field is influenced.
❑ Using *exa103* as a model, type a descriptor to study a magnetic cylinder across an initially parallel magnetic field.
❑ Use *exa125* and *exa136* as templates to compute the field around a 2D magnet with specified magnetization.
❑ In *exa132a*, compute the change in magnetic energy caused by introducing the magnetic core (just change the value of mu).
❑ Calculate the force between a coil and a thick magnetic slab.
❑ Using *exa103* and *exa134* as templates, calculate the mutual force between two magnetic balls in an initially parallel field.
❑ Modify *exa136* to include two equal magnets on the same axis, spaced by Lz/2. Compare to the original fields.

14 Heat Conduction in (x, y) Space

If an object has a steady distribution of temperature, this means that the heat leaving any volume element equals the quantity produced. The corresponding mathematical statement is[4p10]

$$\nabla \cdot \mathbf{f} = h(x, y) \qquad \bullet$$

where $\mathbf{f} = -\lambda \nabla T$ is the *heat flux density* (power transferred per unit area), λ the thermal conductivity, T the thermodynamic (absolute) temperature, and h the heating power per unit volume. Internal heating may for instance be caused by an electric current flowing through the material, by electromagnetic or nuclear radiation, or by a chemical reaction. The PDE is effectively of second order, i.e.

$$\nabla \cdot (-\lambda \nabla T) = h \qquad \bullet$$

The above PDE is similar to that for the potential in a dielectric material (p.76), the heat production term replacing the volume charge. Electric conduction is also a closely analogous process, as summarized by the following table.

Electricity	**Heat**
Potential, U	Temperature, T
Electric field, $\mathbf{E} = -\nabla U$	Temperature gradient, ∇T
Conductivity, σ	Conductivity, λ
Current density, $\mathbf{J} = \sigma \mathbf{E}$	Heat flux density, $\mathbf{f} = -\lambda \nabla T$

Obviously, many of the previous problems may be directly transcribed into descriptors for heat conduction.

Two Hot-Water Tubes

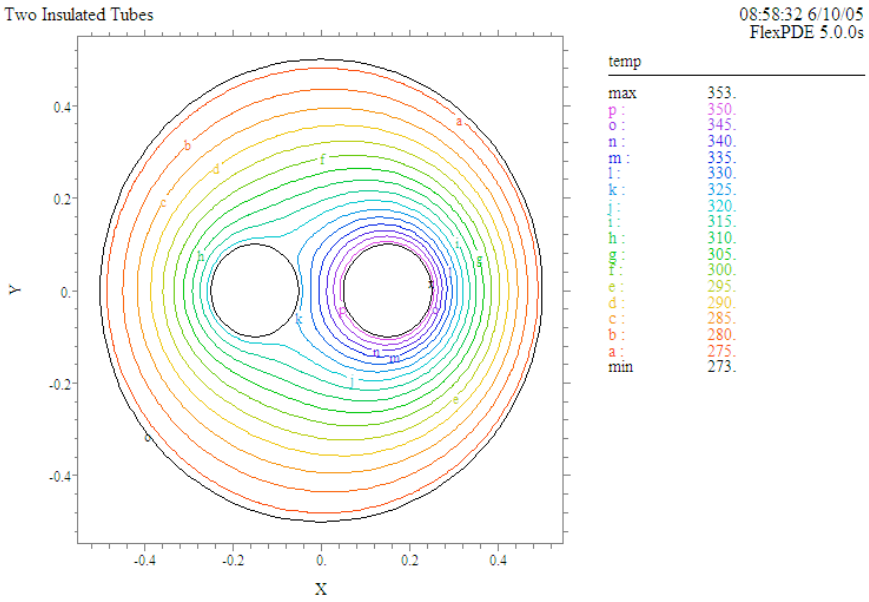
The following example concerns the flow of heat in a cylinder of insulating material containing two tubes carrying water of different temperatures, the outer enclosure being in contact with a soil environment.

Here, we can again make do with the Student Version. We cannot use the notation *T* for temperature, since the program would confuse this with *t*, which is the FlexPDE variable for time. The components of heat flux density are denoted fluxd_x and fluxd_y.

```
TITLE    'Two Insulated Tubes'                                  { exa141.pde }
SELECT                                                  { Student Version }
  errlim=3e-4     ngrid=1     spectral_colors
VARIABLES   temp                                        { SI units: m, K, W }
DEFINITIONS
  r0=0.1     d=0.15     r1=0.5     Lx=0.3     Ly=0.2
  k=0.03     heat=0            { Thermal conductivity and power density }
  fluxd_x=-k*dx(temp)     fluxd_y=-k*dy(temp)
  fluxd=vector( fluxd_x, fluxd_y)     fluxdm=magnitude( fluxd)
  f_angle=sign(fluxd_y)*arccos(fluxd_x/fluxdm)/pi*180
EQUATIONS
  div( fluxd)=heat
BOUNDARIES
region 'domain'
  start 'outer' (0,-r1) value(temp)=273             { Frozen soil }
  arc( center=0,0)  angle=360
  start 'left' (-d-r0,0) value(temp)=323        { Exclude left hot water tube }
  arc( center=-d,0) angle=360
  start 'right' (d-r0,0) value(temp)=353            { Right hot water tube }
  arc( center=d,0) angle=360
PLOTS
  contour( temp)     surface( temp)
  vector( fluxd) norm      contour( f_angle)
  contour( fluxd_x)     contour( fluxd_y)     contour( fluxdm)
  elevation( normal( fluxd)) on 'outer'
  elevation( -normal( fluxd)) on 'left'                 { Outwards from tube }
  elevation( -normal( fluxd)) on 'right'
END
```

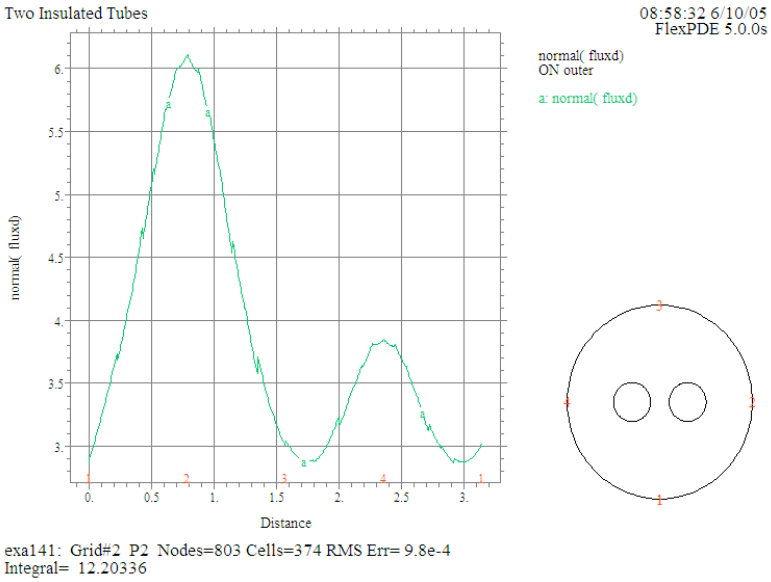
The figure below shows the temperature distribution obtained. According to the vector plot, the tube to the left both receives and

delivers heat, as is also evident from the plot of fluxd_x. The angle plot confirms this and yields more details.



An important derived quantity is the power transfer (per unit length) from the heat-carrying tubes to the surrounding soil. We may calculate this power by elevation plots of the outward normal component of the flux density f_n, which the program automatically integrates over a cylindrical boundary.

We calculate the heat flux leaving the outer tube by means of an elevation plot (below) of f_n on the curve named 'outer'.



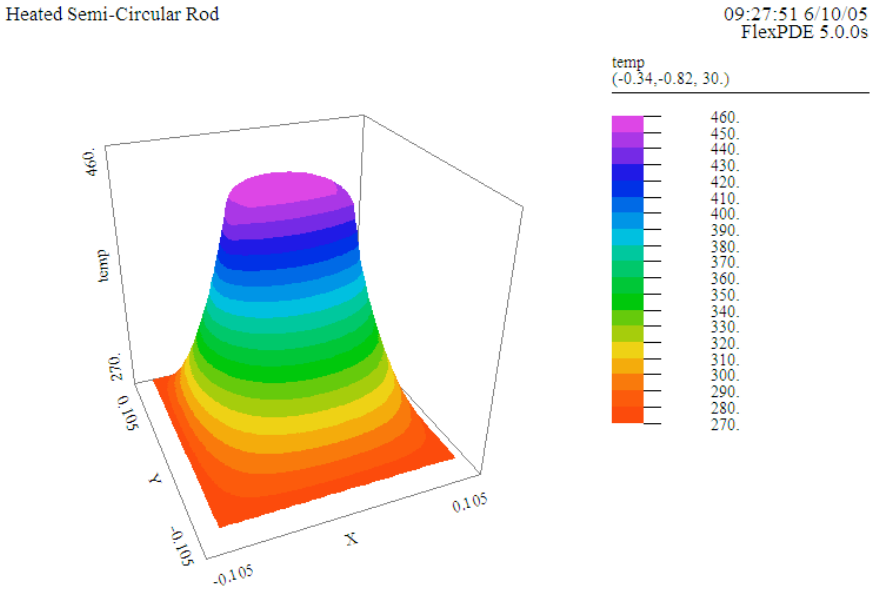
In a similar manner, we may integrate the flux leaving each of the hot tubes (thus *entering* the domain). From these elevation plots we gather that the heat flux (per unit length) from the left tube is 2.78 and from the right one 9.38. The sum of these terms is 12.16, which agrees with the flux from the outer tube within 0.3%. If we use the Professional Version with a smaller errlim the error may be further reduced.

Uniformly Heated, Semi-Circular Rod

In the next problem, an electric current uniformly heats a long, semi-circular rod of stainless steel. A granite block, cooled on the outside by a mixture of ice and water, surrounds the steel rod. Here we need to include the power per unit volume (heat) in the PDE, its value being zero in the granite and 1e6 in the steel.

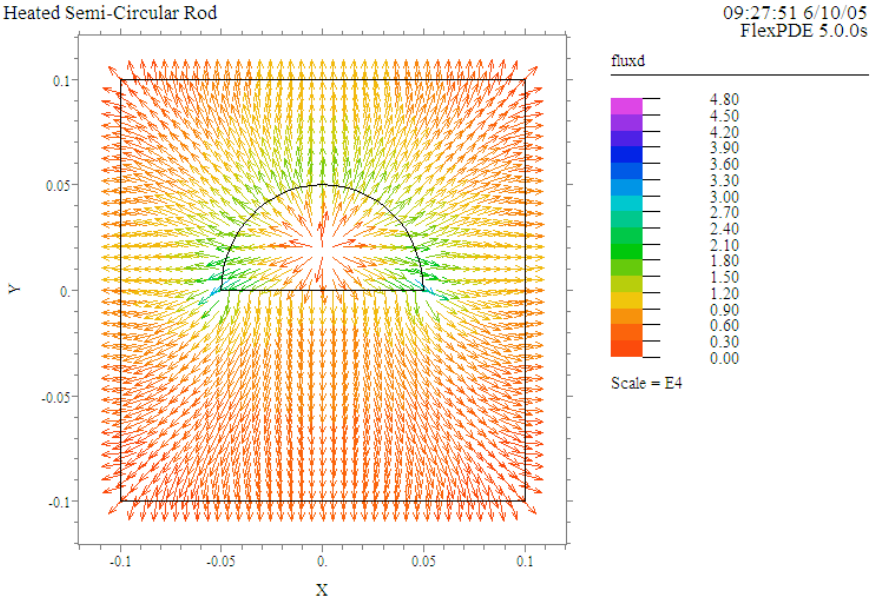
```
TITLE   'Heated Semi-Circular Rod'                          { exa142.pde }
SELECT    errlim=3e-4    ngrid=1    spectral_colors
VARIABLES   temp
DEFINITIONS
  Lx=0.1    Ly=0.1    r0=0.05
  k    heat          { Conductivity and power per unit volume, declared }
  fluxd_x=-k*dx(temp)    fluxd_y=-k*dy(temp)
  fluxd=vector( fluxd_x, fluxd_y)    fluxd_m=magnitude( fluxd)
EQUATIONS
  div( fluxd)=heat
BOUNDARIES
region 'domain'   k=3.5   heat=0                            { Granite }
  start  'outer' (-Lx,-Ly) value(temp)=273
  line to (Lx,-Ly)  to (Lx,Ly)  to (-Lx,Ly)  close
region 'steel'  k=45  heat=1.0e6
  start 'rod' (-r0,0) line to (r0,0) arc to (0,r0)  close
PLOTS
  contour( temp)    surface( temp)    contour( temp) painted on 'steel'
  vector( fluxd) norm      contour( fluxd_m) painted
  elevation( normal(fluxd)) on 'rod'
  elevation( normal(fluxd)) on 'outer'
     report( pi*r0^2/2* 1e6) as 'Heating power'
END
```

The surface plot below suggests that there is a temperature maximum within the region of the steel rod. The painted contour plot demonstrates this more clearly.



exa142: Grid#3 P2 Nodes=803 Cells=380 RMS Err= 0.0014
Integral= 13.65766

The following vector plot shows the direction of heat flow. It brings out the position of the temperature maximum as the point from which the flow diverges.



exa142: Grid#3 P2 Nodes=803 Cells=380 RMS Err= 0.0014

The way the flux arrows diverge from the corners of the rod suggests that the flux density is highest there. The painted contour plot of fluxd_m demonstrates this more directly.

The first elevation plot yields the power leaving the rod via the integral around the periphery. The second elevation plot yields the power leaving the domain, which should have the same magnitude. These integrals should also equal the electrical heating power calculated on the *report* line included on the last plot.

Cooling by Forced Convection

So far we have always specified a constant value or a function of the space coordinates as the *natural* boundary condition. We are also allowed, however, to use natural conditions containing the dependent variable *T*.

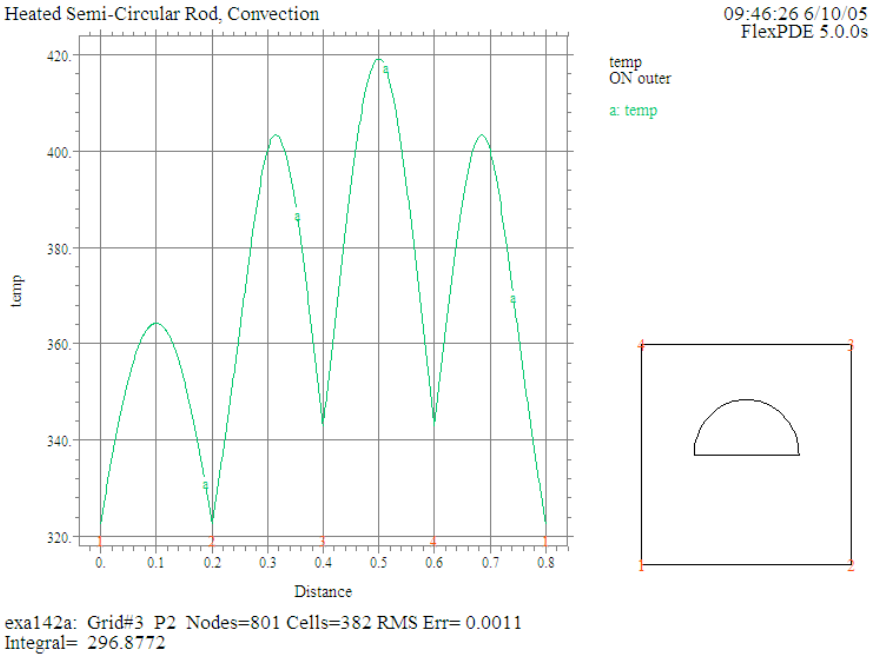
This new usage is required in problems involving *forced convection* at the boundary. If a fluid is driven at constant speed by means of a pump or a fan past a hot object, the latter is cooled at a heat flux density proportional to the difference between the temperature of the object and the temperature of the fluid. This proportionality, which is more of a rule of thumb than a law of nature, is known as Newton's law of cooling. Loss by infrared radiation leads to a similar expression for the natural boundary condition.

To apply Newtonian cooling we modify *exa142* as follows. Instead of imposing a constant temperature of 273 on the outer rectangular surface, we assume it to be cooled in proportion to the temperature difference with respect to the coolant. The fluid has a constant temperature of 273 and we specify a cooling coefficient of 50.

```
TITLE   'Heated Semi-Circular Rod, Convection'          { exa142a.pde }
…
region 'domain'  k=3.5   heat=0                          { Granite }
   start  'outer' (-Lx,-Ly) natural(temp)=50*(temp- 273)
   line to (Lx,-Ly) to (Lx,Ly) to (-Lx,Ly)  close
{ Keep region 'steel' }
…
   elevation( temp) on 'outer'
END
```

Notice that we only specify natural boundary conditions, and still we do not need any point value (p.52). The cooling term contains the difference between temp and 273, the fluid temperature and this supplies a temperature reference.

The following plot demonstrates that the temperature on the boundary now varies strongly, being considerably higher than in the cooling fluid.



The painted contour plot on 'steel' shows that the maximum temperature also is higher than before by about a hundred K. The input and output power values still balance reasonably well.

Conduction in Anisotropic Wood

In the areas of gravitation, electro- and magnetostatics, and heat conduction we applied PDEs of the type

$$\nabla \cdot (\beta \nabla f) = \gamma$$

which is valid if the materials property β is a scalar function of the space variables. We shall now consider a situation where this parameter depends on the direction of the flux density. In that case we

need to write the PDE such as to reference the x and y directions explicitly, i.e.

$$\frac{\partial}{\partial x}\left(-\lambda_x(x,y)\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(-\lambda_y(x,y)\frac{\partial T}{\partial y}\right)=h(x,y) \quad \bullet$$

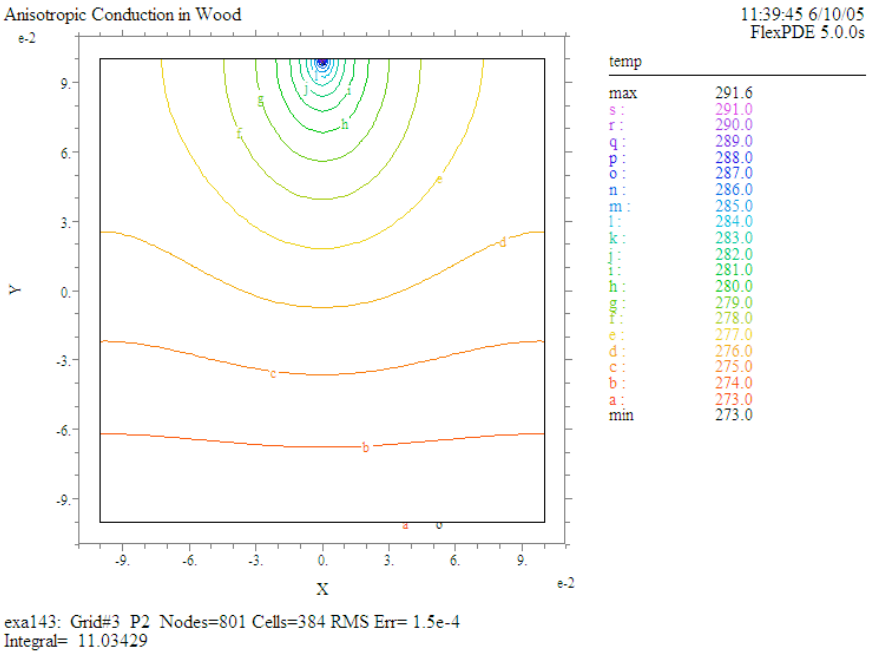
The descriptor below provides a realistic example of conduction in an anisotropic material. The thermal conductivity of pine is considerably higher for flow in the direction of the fibers than across. We consider a bar of such wood, cut so that the fibers are in the y direction (vertical). On the top surface we mount an electric heater in the form of a strip of width $2w$, parallel to the z-axis. The bottom surface of the wooden block is held at ice-water temperature, all other surfaces being well insulated, including the side of the heater that is not in contact with the wood.

Since the power dissipated in the heater can only flow *into* the specimen, we represent the heater by a downward flux density of 1000 over a segment of width $2w$ at the upper boundary.

```
TITLE   'Anisotropic Conduction in Wood'                    { exa143.pde }
SELECT    errlim=1e-4    ngrid=1    spectral_colors
VARIABLES   temp
DEFINITIONS
  L=0.1    w=0.001    kx=0.14    ky=0.35    heat=0         { Pine }
  fluxd_x=-kx*dx(temp)    fluxd_y=-ky*dy(temp)
  fluxd=vector( fluxd_x, fluxd_y)    fluxd_m=magnitude( fluxd)
EQUATIONS
  dx[-kx*dx(temp)]+ dy[-ky*dy(temp)]=heat
BOUNDARIES
region 'domain'  start 'outer' (-L,-L) value(temp)=273   line to (L,-L)
  natural(temp)=0  line to (L,L)  to (w,L)
  natural(temp)=-1000  line to (-w,L)                       { Heater flux }
  natural(temp)=0 line to (-L,L) close
PLOTS
  contour( temp)    surface( temp)
  contour( temp) painted zoom(-0.1*L,0.8*L,  0.2*L,0.2*L)
  vector( fluxd) norm
END
```

The temperature plot below exhibits contours which are vertically elongated in the neighborhood of the heater strip. In spite of the

concentrated heating on part of the upper surface, the temperature curve closest to the bottom of the specimen is almost flat.



Remembering that we have already seen an exact solution for the electric field (p.39), which has a similar PDE, we may imagine a generalized form valid in the present situation. Let us try the function

$$T = A + B\,\ln\left(x^2/a^2 + y^2/b^2\right)$$ ●

On substituting we find that this function satisfies the PDE, provided that $a^2 = \lambda_x$ and $b^2 = \lambda_y$. The solution $T(x,y)$ is thus constant on ellipses with half-axes proportional to $\sqrt{\lambda_x}$ and $\sqrt{\lambda_y}$. This may be confirmed by measurements on the contour plot.

We have chosen a very narrow heater strip in order to hide the effect of its finite width. If we increase *w* by a factor of 10, however, we will detect an elongation of the near contours in the *x* direction.

Exercises

❑ Modify *exa141*, assuming that the left tube is empty.
❑ One of the inner tubes in *exa141* both receives and delivers heat. Determine the two opposite contributions to the flux separately by

plots of the outward normal component of heat flux density. Integrate the flux density over the *entire* boundary of the tube, then integrate the *absolute* value of the flux density, then add/subtract.

❑ Combine the half-cylinder of steel (*exa142*) with a magnesium oxide bar of triangular cross-section, having a thermal conductivity of 55. Let the flat surface be common to the two rods and place the remaining corner of the triangle at (*r*0,-*r*0).

❑ Using *exa142a* as a model, modify *exa141* by introducing forced convection at the outer boundary.

❑ Solve the problem of the heated wood rod (*exa143*) under other conditions. First assume that the fibers are in the *z* direction, i.e. that $k_x = k_y = 0.14$. Next, assume that the fibers are horizontal.

15 Heat Conduction in (ρ,z) Space

In the case of complete axial symmetry, the PDE for heat conduction may still be written

$$\nabla \cdot \mathbf{f} = h$$

but the divergence must be transformed for use with cylindrical coordinates[3p705]. Knowing that

$$\nabla \cdot \mathbf{f} = \frac{1}{\rho}\frac{\partial(\rho f_\rho)}{\partial \rho} + \frac{\partial f_z}{\partial z}$$

we immediately obtain the appropriate form

$$\frac{1}{\rho}\frac{\partial}{\partial \rho}\left(-\rho\lambda\frac{\partial T}{\partial \rho}\right) + \frac{\partial}{\partial z}\left(-\lambda\frac{\partial T}{\partial z}\right) = h$$

where λ is the thermal conductivity. The radial and axial components of heat flux density simply become

$$f_\rho = -\lambda\frac{\partial T}{\partial \rho} \text{ and } f_z = -\lambda\frac{\partial T}{\partial z}\ . \qquad \bullet$$

Radial Conduction in a Hollow Cylinder

As an elementary test of the program under axial symmetry, let us study a case of heat conduction where T is independent of z. The following descriptor applies to a hollow cylinder of magnesium oxide, uniformly heated from the inside by electrical power. The outer cylindrical surface is kept at 300 K by water cooling, and the flat ends are well insulated to ensure radial flow.

The heating power is taken to be 10^4 per unit of axial length, and from this we calculate the *flux density* entering in the radial direction from the central bore. The normal vector on the inner boundary points

out of the solution domain (or toward the origin), which means that the input flux density must be negative.

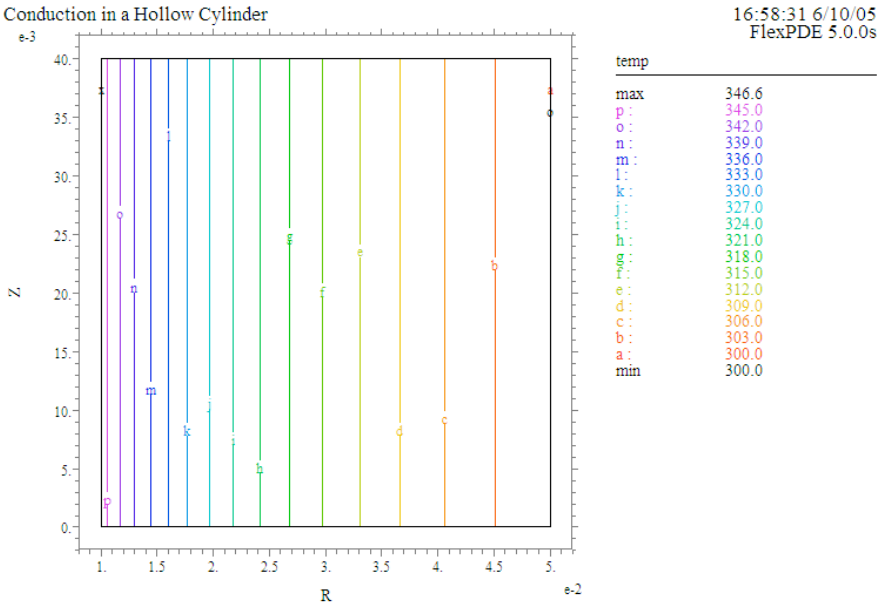
We may still express the PDE using the div operator, which the program automatically transforms into cylindrical coordinates.

This problem has an exact solution, which we may construct using a function (p.39) of the form $A + B\ln(\rho)$. The boundary conditions determine the constants A and B.

```
TITLE   'Conduction in a Hollow Cylinder'                    { exa151.pde }
SELECT                                                  { Student Version }
  errlim=1e-5     ngrid=1     spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES   temp
DEFINITIONS
  r1=10e-3     r2=50e-3     L=40e-3
  temp2=300     heat=0     k=55                    { Magnesium oxide }
  power_m=1e4     fluxd_in=power_m/(2*pi*r1)
  fluxd_r=-k*dr(temp)     fluxd_z=-k*dz(temp)
  fluxd=vector( fluxd_r, fluxd_z)     fluxdm=magnitude( fluxd)
  temp_ex=r1*fluxd_in/k*ln(r2/r)+ temp2
EQUATIONS
  div( fluxd)=heat
BOUNDARIES
region 'domain'
  start(r1,0)  natural(temp)=0   line to (r2,0)
  value(temp)=temp2  line to (r2,L)
  natural(temp)=0   line to (r1,L)
  natural(temp)=-fluxd_in   line to close
PLOTS
  contour( temp)     contour( temp- temp_ex) as ' Error in temp '
  contour( fluxdm)     vector( fluxd) norm
  contour( fluxd_r)     contour( fluxd_z)
END
```

There is an important point to notice when using *natural* boundary conditions: if we effectively multiply the dependent variable (temp) by a factor (k) in the PDE, we must also multiply the derivative in the boundary condition by the same factor (p.51). If only *value* boundary conditions occur, however, multiplying all terms of the equation by the same factor does not change the solution.

The plot below shows the temperature contours, which crowd close to the central bore.



The plot of the deviation with respect to the exact solution yields irregular contours as expected. The maximum deviation is about 0.01% of the total temperature drop. The root-mean-square deviation reported by the program, however, is about 1e-5.

The vector plot and the contour plots of fluxd_r and fluxd_z demonstrate that the heat flow is highly radial.

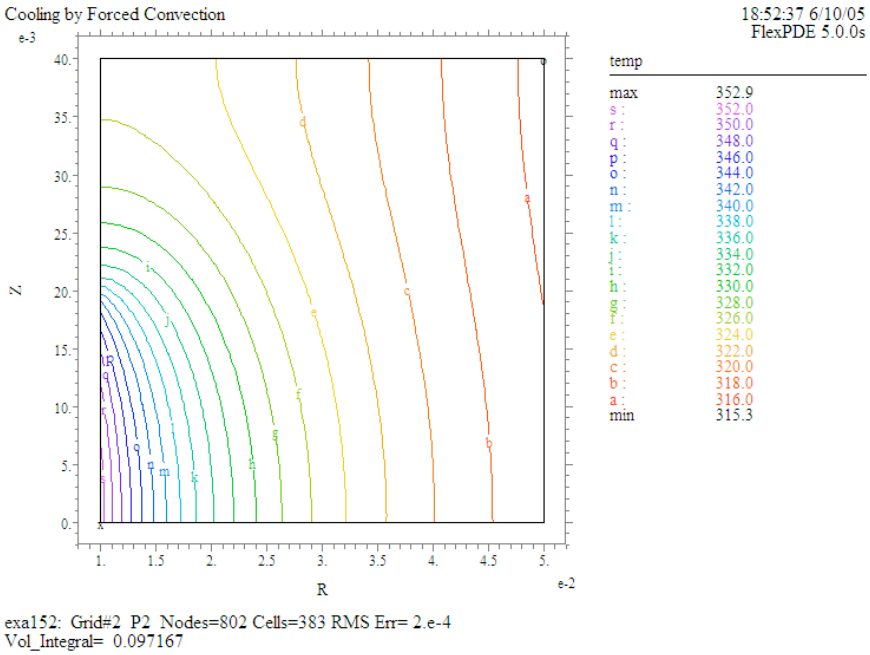
Cooling by Forced Convection

To apply Newtonian cooling (as on p.167) we modify *exa151* as follows. We heat from the inside over the lower half of the bore, and assume the outer cylindrical surface to be cooled in proportion to the temperature difference with respect to the coolant. The temperature of the fluid is 300 and we specify a cooling coefficient of 1e3.

```
TITLE   'Cooling by Forced Convection'                    { exa152.pde }
…                                                  { Delete temp_ex }
region 'domain' start(r1,0)  natural( temp)=0  line to (r2,0)
   natural(temp)=1e3*(temp- temp2)  line to (r2,L)
   natural(temp)=0  line to (r1,L)  to  (r1,L/2) mesh_spacing=0.2*r1
   natural(temp)=-fluxd_in   line to close
PLOTS
   contour( temp)     vector( fluxd) norm     contour( fluxdm) painted
```

```
  contour( fluxd_r) painted     contour( fluxd_z) painted
END
```

You should obtain a temperature plot similar to the one shown below.



The vector plot clearly illustrates the pattern of heat flow. It is of some interest to compare that to the plots of the radial and axial flux densities.

Continuously Varying Thermal Conductivity

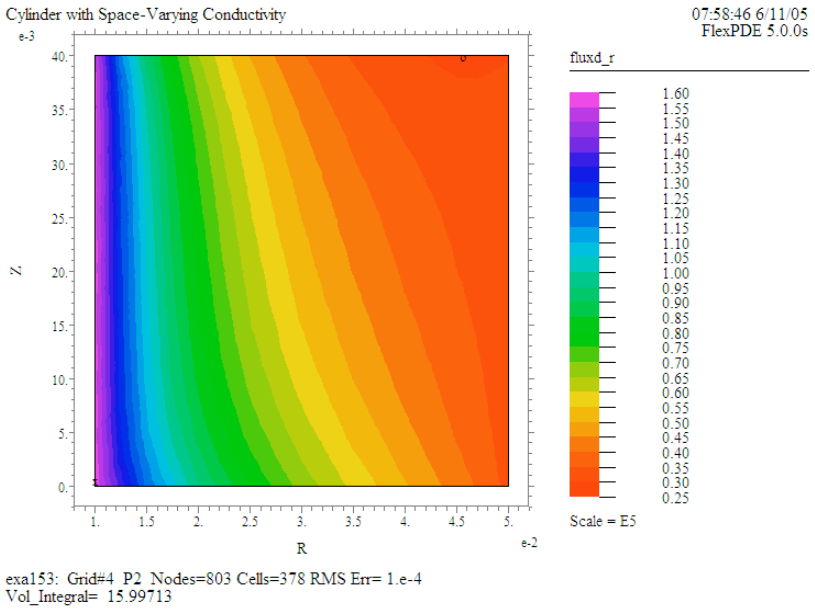
In the above examples the thermal conductivity was constant throughout the object. We may introduce a space-varying conductivity, however, depending on the axial as well as the radial coordinate.

Shaking a mixed powder, for instance, would make the heavy grains segregate from the lighter ones, thus creating a thermal conductivity gradient. Centrifuging about the vertical axis would produce a corresponding gradient in the radial direction. After consolidating the powder under high pressure we would be left with a solid with varying conductivity, but still with axial symmetry.

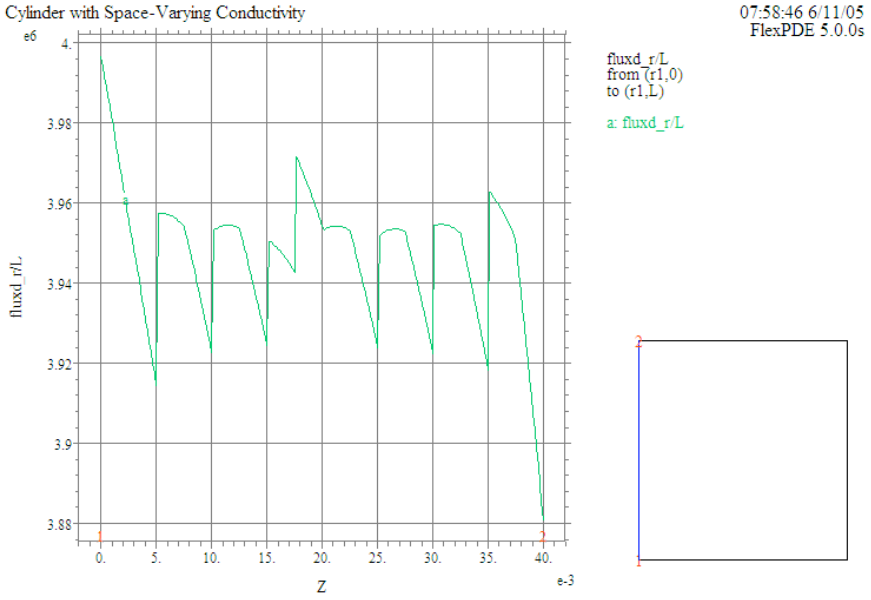
The next descriptor is based on *exa151*. It refers to an axially symmetric object, having a thermal conductivity (k) that varies both radially and axially. The boundary conditions apply to heating along the entire inner boundary and forced cooling on the outside.

```
TITLE   'Cylinder with Space-Varying Conductivity'          { exa153.pde }
…
   temp2=300     heat=0     k=55*exp[20*(r-z)] { Shaken, not stirred }
…                                                          { Delete temp_ex }
region 'domain'
   start(r1,0)  natural( temp)=0  line to (r2,0)
   natural(temp)=1e3*(temp- temp2)  line to (r2,L)
   natural(temp)=0  line to (r1,L)
   natural(temp)=-fluxd_in   line to finish
PLOTS
   contour( k) painted
   contour( temp)     vector( fluxd) norm     contour( fluxdm) painted
   contour( fluxd_r) painted     contour( fluxd_z) painted
   elevation( fluxd_r/L) from (r1,0) to (r1,L)       { Compare power_m }
   elevation( fluxd_r/L) from (r2,0) to (r2,L)
END
```

We notice that the program solves this problem without difficulty, although an analytic solution is unlikely to exist. The figure below is a plot of the radial component of the flux density.



At first, it may be surprising to find that fluxd_r decreases toward larger radii, considering that energy must be conserved. The two elevation plots (the first one shown below) illustrate, however, that the heat transported past the inner radius is nearly equal to that leaving at the outer radius. Dividing by L we obtain the energy delivered per unit length of the heater, which evidently agrees with power_m.



exa153: Grid#4 P2 Nodes=803 Cells=378 RMS Err= 1.e-4
Surf_Integral= 9920.231

Using an extended descriptor, you could easily test if the boundary conditions are satisfied, by plotting the function fluxd_r/(temp-300) on the outer cylindrical boundary.

Steel Tube with Cooling Flanges

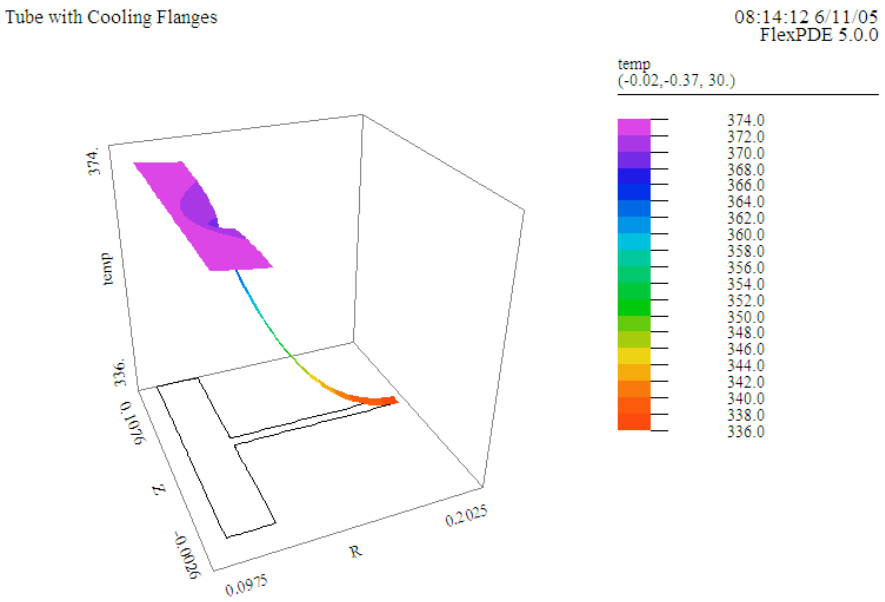
We shall now proceed to a device of practical importance: a series of circular cooling flanges on a steel tube. The flanges are located at equal distances along the tube, and all of them are at about the same temperature, so that there is no axial heat flow at mid-distance between two adjacent flanges. Thus we may limit our study to *one* flange and a half-length of tube on either side of it (i.e. one periodic unit). The contour plots illustrate the geometry.

The following descriptor may be based on *exa151*. Let us assume cooling by airflow at a speed corresponding to a cooling coefficient of 20.

```
TITLE   'Tube with Cooling Flanges'                          { exa154.pde }
SELECT    errlim=3e-5    ngrid=1    spectral_colors
COORDINATES   ycylinder( 'r','z')
VARIABLES   temp
DEFINITIONS
  r1=100e-3    r2=120e-3    r3=200e-3    L=5e-2    dL=0.5e-2
  temp2=300    heat=0    k=45                                  { Steel }
  fluxd_r=-k*dr(temp)    fluxd_z=-k*dz(temp)
  fluxd=vector( fluxd_r, fluxd_z)    fluxdm=magnitude( fluxd)
EQUATIONS
  div( fluxd)=heat
BOUNDARIES
region 'domain'
  start (r1,0)  natural(temp)=0   line to (r2,0)               { Insulated }
  natural( temp)=20*(temp-temp2)
  line to (r2,L)  to (r3,L)  to (r3,L+dL)  to (r2,L+dL)  to (r2,2*L+dL)
  natural( temp)= 0 line to (r1,2*L+dL)
  value( temp)=373 line to close                               { Inside tube }
feature                                                        { Cooling surface }
  start 'outer' (r2,0)
  line to (r2,L) to (r3,L) to (r3,L+dL) to (r2,L+dL) to (r2,2*L+dL)
PLOTS
  contour( temp) painted    surface( temp)
  vector( fluxd) norm    contour( fluxdm) painted
  contour( fluxd_r) painted    contour( fluxd_z) painted
  elevation( normal( fluxd)) on 'outer'
  elevation( fluxd_r) from (r1,2*L+dL) to (r1,0)
END
```

The surface plot below is a striking presentation of the temperature distribution.

Elevation plots permit us to compute the power leaving this section of the tube and also suggest how the design may be improved by changing the flange thickness.



exa154: Grid#5 p2 Nodes=1407 Cells=662 RMS Err= 2.e-5
Vol_Integral= 0.679246

Both elevation plots report the power extracted from the liquid that flows through the tube. Since we have declared (r,z) coordinates, the area element factor 2*pi*r is automatically included. The two integral values are in tolerably good agreement but could be improved with the Professional Version of FlexPDE.

Exercises

❑ Modify *exa151* by specifying 400 K on the inner surface and 300 K on the outer one. Calculate the power transferred per unit length using elevation plots.

❑ In *exa152*, plot the power per unit length leaving the tube and compare to the convection term. Integrate to find the total power and compare that to the power delivered on the axis.

❑ Verify numerically that the solution to *exa153* satisfies the PDE as well as the boundary conditions.

❑ In *exa154* the cooling efficiency depends on the radius of the flange. How would you plan a calculation of the maximum ratio of cooling power to flange weight? Assume the thickness of the tube to be unchanged, chosen thick enough to stand the inside pressure.

16 Non-Linear Heat Transfer

The PDEs we have considered so far have been linear in the *dependent* variable, even if some coefficients have been non-constant, such as in the PDE $\nabla(-\lambda(x,y)\nabla T) - h(x,y) = 0$. If a linear, homogeneous equation $\nabla(-\lambda(x,y)\nabla T) = 0$ has two solutions, T_1 and T_2, then $T = T_1 + T_2$ will also be a solution.

If the conductivity λ should depend on *T,* however, the PDE is no longer linear in *T*, which means that we cannot generate new solutions by summing, and hence very few analytic solutions have been found. Fortunately, FlexPDE usually permits us to solve non-linear equations as well.

Temperature-Dependent Conductivity

The descriptor below is for a plate of MgO, thick enough to let us neglect the infrared radiation power from the surfaces in comparison with that conducted through the material. There is a hole in the center of the plate. The edges are at very different temperatures, 300 and 2300 K, and since the thermal conductivity λ is known to be proportional to $1/T$ for this material, the PDE will evidently be non-linear.

In the new descriptor segment *initial values* we give a constant value as a guide in searching for a solution. Depending on the case, this hint may or may not be necessary. If the equation is nearly linear, FlexPDE may well solve it without help, but the run will perhaps take more time. Here, we take the initial temperature to be halfway between the extremes, but even the extreme values would do.

```
TITLE   'MgO Plate with a Hole, k(T)'                    { exa161.pde }
SELECT                                          { Student Version }
  errlim=1e-4     ngrid=1     spectral_colors
VARIABLES   temp
```

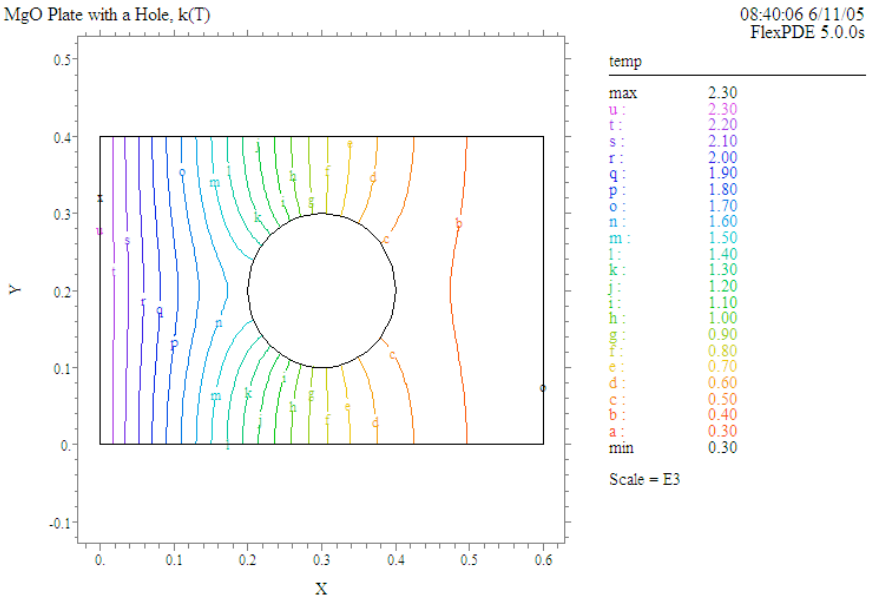
```
DEFINITIONS
  Lx=0.6    Ly=0.4    r0=0.1
  heat=0    k=15000/temp                    { Thermal conductivity k(T) }
  fluxd_x=-k*dx(temp)    fluxd_y=-k*dy(temp)
  fluxd=vector( fluxd_x, fluxd_y)    fluxdm=magnitude( fluxd)
INITIAL VALUES
  temp=1300
EQUATIONS
  div( fluxd)=heat
BOUNDARIES
region 'domain'
  start (0,0)  natural(temp)=0  line to (Lx,0)          { Insulated }
  value(temp)=300  line to (Lx,Ly)
  natural(temp)=0  line to (0,Ly)
  value(temp)=2300   line to close
  start (Lx/2-r0,Ly/2)  natural(temp)=0                 { Exclude hole }
  arc (center=Lx/2,Ly/2)  angle=360
MONITORS
  elevation(temp) from (0,0) to (Lx,0)
PLOTS
  contour( temp)    contour( k) painted    vector( fluxd) norm
  contour( fluxd_x)    contour( fluxd_y)    contour( fluxdm)
  elevation( temp) from (0,0) to (Lx,0)
END
```

In non-linear problems it is of some importance to study the convergence of the consecutive trial functions toward the final solution. For this purpose we include a *monitors* segment, which permits us to inspect progress at a glance by various plots. Monitor plots are in fact evoked by the usual commands, with the difference that they are displayed on the screen each time a new trial function has been obtained. There may be several of these survey plots.

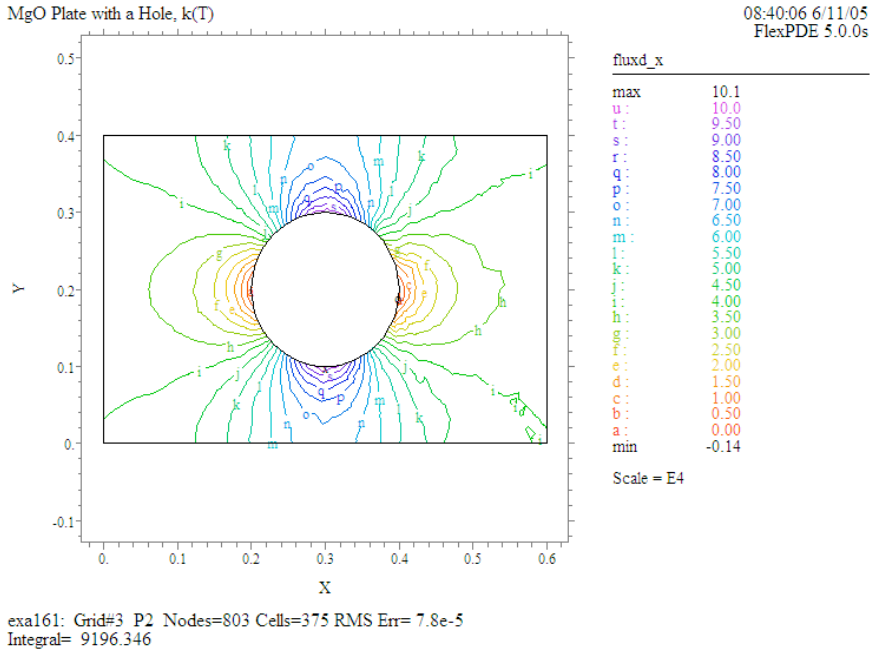
In this case we introduce an elevation plot as a monitor. During the solution process we see three such plots, but they are very similar.

The figure below shows how the temperature contours crowd in the hot region to the left, where the thermal resistivity ($1/k$) is high, and also where the hole creates constrictions.

The temperature plot suggests asymmetry of the horizontal gradient $\partial T/\partial x$, which may also be demonstrated by a separate contour plot of dx(temp).



The following figure is a contour plot of the *x*-component of the flux density **f**.



The symmetry of the above plot may be a surprise, in view of the asymmetry of the temperature. We may understand this behavior if we remember that both the PDE and the boundary conditions are independent of λ. An asymmetric function fluxd_x would also violate the conservation of energy.

Steel Foil Emitting Infrared Radiation

In the previous case, conduction through the material dominated any radiative loss from the surfaces. We shall now consider a thin steel sheet, where the loss by infrared radiation becomes considerable in comparison with the heat conducted through the foil. To simplify, we assume the thermal conductivity of steel to be independent of temperature, which is approximately true. The program would solve the problem even if λ varied with T, but we had better study one effect at a time.

The following file is rather similar to the preceding one, but we have eliminated the circular exclusion. The PDE now has a source term (here $h < 0$) representing the radiative loss per unit volume. Strictly speaking, the loss occurs at the front and rear surfaces, but in the 2D version of the program we do not have access to the normal derivatives in the z direction. Hence we assume the sheet to be thin enough so that the temperature variation across the foil may be neglected. In other words, we assume the volume to be radiating, instead of the surfaces.

The heat power radiated per unit area is given by the formula $f = \varepsilon\sigma T^4$, where ε is the emissivity (for which we take the value 0.3) and σ the Stefan-Boltzmann constant. The foil loses heat at this rate, but it also gains by radiation from the surroundings, which we assume to be at 300 K. We multiply f by 2 in order to take both faces into account. Since we consider the net surface power to be produced in a volume of thickness d, we must divide by d to express it as power per unit *volume*.

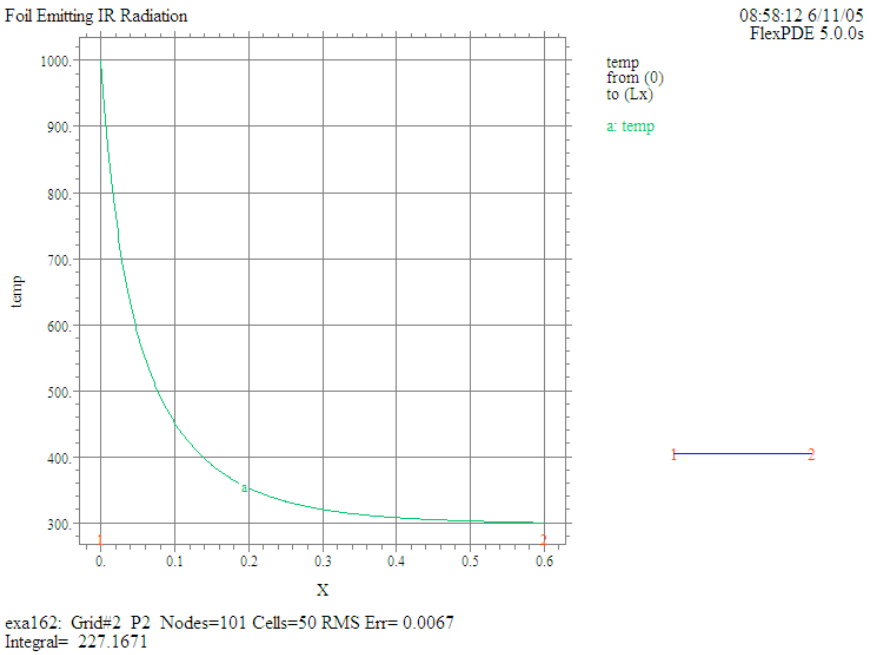
Although this PDE is strongly non-linear, the initial value is not of critical importance. We could even omit this segment, which makes the program assume an initial value of zero.

In this problem, there is no variation in the y direction. Hence we may use an *ordinary* differential equation in x only (ODE), declared by cartesian1. This simplifies the script considerably.

```
TITLE   'Foil Emitting IR Radiation'                    { exa162.pde }
SELECT
  errlim=1e-4     spectral_colors
COORDINATES
```

```
  cartesian1
VARIABLES   temp
DEFINITIONS
  Lx=0.6     Ly=0.4     d=1e-3     k=45                    { Steel }
  heat=-2* 0.3*  5.67e-8*(temp^4-300^4)/d                  { Per unit volume }
         { 2 faces, emissivity, Stefan-Boltzmann formula  }
  fluxd_x=-k*dx(temp)
  equ1=dx( fluxd_x)     equ2=heat
INITIAL VALUES
  temp=600                                                 { May be omitted }
EQUATIONS
  dx( fluxd_x)=heat
BOUNDARIES
region 'domain'
  start (0) point value(temp)=1000  line to (Lx) point value(temp)=300
MONITORS
  elevation( temp) from (0,0) to (Lx,0)
PLOTS
  elevation( temp) from (0) to (Lx)
  elevation( equ1, equ2) from (0) to (Lx)
END
```

The surface plot below shows the final temperature distribution along the metal strip, obtained after a few iterations.



Since the PDE now has two terms, which must be equal (equ1 and equ2 under *definitions*), we may compare these two expressions in a single elevation plot. The expression equ2 is equal to the power lost by radiation. Evidently, the two terms of the PDE are in reasonable agreement. The curve for equ1 is based on 2nd derivatives and is in fact piecewise constant.

Exercises

- ❑ Remove the circular cutout in *exa161*. Run the resulting, simpler problem in cartesian1 and verify that the *x*-component of flux density is constant for all values of *x*, as expected.
- ❑ Run *exa162* with d=1e-4 and explain why the results are radically different.
- ❑ Add suitable plots to *exa162* to compare the total power radiated to the power delivered by conduction at the high-temperature end.
- ❑ Modify *exa162* to include a circular cutout as in *exa161*. Solve it also for an upper temperature of 600.

17 Simultaneous Electric and Thermal Conduction

An electric current always heats the material through which it passes (unless the material is a superconductor). If the heating is non-uniform, which is often the case, temperature differences will also occur, and we will have conduction of heat as well as conduction of charge. The electrostatic potential distribution is still governed by the equation

$$\nabla \cdot \mathbf{J} = \nabla \cdot (\sigma \mathbf{E}) = \nabla \cdot (-\sigma \nabla U) = 0$$

if the material is isotropic and there is no source of charge inside the material.

The resulting current density causes dissipation of heat at the rate $\mathbf{J} \cdot \mathbf{E} = |\mathbf{J}||\mathbf{E}| \equiv JE$ per unit volume. This electric power will appear as a source term (h) in the PDE for heat conduction (p.162), i.e.

$$\nabla \cdot \mathbf{f} = JE \qquad \bullet$$

where $\mathbf{f} = -\lambda \nabla T$.

In order to calculate both the electric potential and the temperature we thus have to face the following *system* of simultaneous PDEs

$$\begin{cases} \nabla \cdot \mathbf{J} = 0 \\ \nabla \cdot \mathbf{f} = JE \end{cases} \qquad \bullet$$

It is obvious that this problem is non-linear, because the expressions for both J and E contain U. If the electrical conductivity σ had been independent of the temperature T, we could have solved the first equation with respect to U, then used that solution in the second equation. That approximation is rarely valid, however.

For a pure metal, such as copper, the electric conductivity is roughly proportional to $1/T$ over a large range of temperatures. Hence, both equations are non-linear in a dependent variable, and the equations have to be solved simultaneously.

The program solves a system almost as easily as if it were a single equation. It is wise, however, to supply *initial values* for the dependent variables, if at all possible. If you have a rough idea about the solution, give an expression for it. At least you could indicate a typical value for the solution. If no initial value is supplied, the program assumes that value to be zero, which may or may not be a good starting point.

There are as many dependent variables (solutions) as there are equations. For each of these dependent variables you must supply boundary values or normal derivatives. If not, the program will assume vanishing normal derivatives, which may be quite different from what you intended.

Copper Block heated by an Electric Current

In our first example, a block of copper has a potential difference (voltage) between opposite faces. We ignore the practical problems of keeping constant potential values over each of the two faces. Similarly, we assume a constant temperature of 300K on these faces. For this first application we choose the simplest possible conditions, including electric and thermal insulation on the other boundaries, which makes the problem one-dimensional.

In the descriptor below we have specified a linear function for U as the *initial value* for the potential. For the temperature temp we simply choose the value imposed at the boundaries.

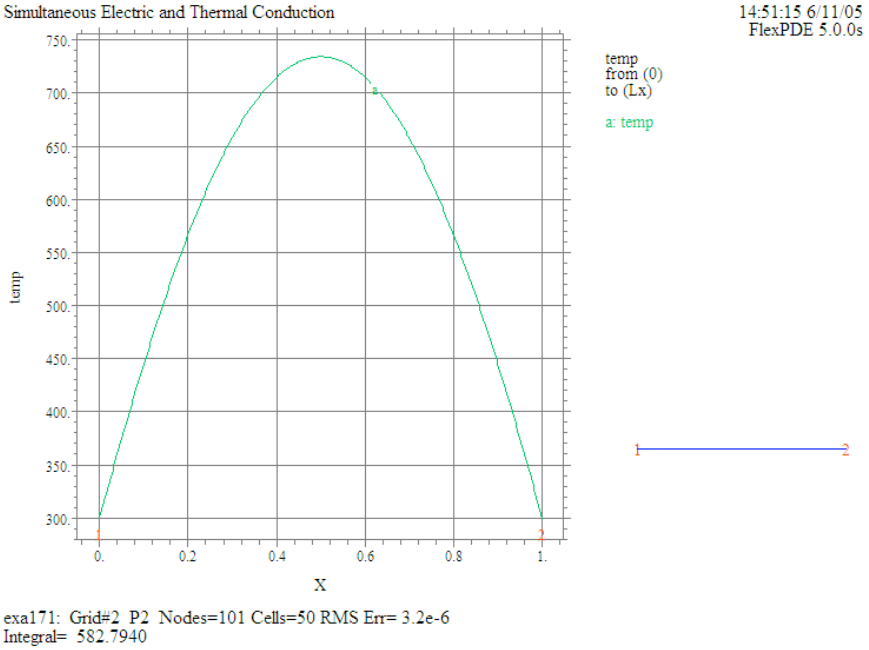
Multiple PDEs must be *labeled* with their *dominant* dependent variable. The first one does not even depend on temp, so obviously we should tag it with the variable U.

The second PDE depends on both U and T, but the derivation shows that it expresses heat balance. For that reason we should choose the remaining variable temp as the tag.

```
TITLE   'Simultaneous Electric and Thermal Conduction'   { exa171.pde }
SELECT                                                   { Student Version }
  errlim=1e-5     spectral_colors
COORDINATES
  cartesian1                                             { 1D calculation }
VARIABLES   U    temp                                    { Two dependent variables }
```

```
DEFINITIONS
  Lx=1.0     temp0=300     U1=-0.2
  cond0=1/1.67e-8     cond=cond0*temp0/temp     k=400     { Copper }
  Ex=-dx(U)     Jx=cond*Ex
  fluxd_x=-k*dx( temp)
INITIAL VALUES
  U=U1*x/Lx   temp=300
EQUATIONS
  U:          dx( Jx)= 0                  { U is the only dependent variable }
  temp:       dx( fluxd_x)=Jx*Ex          { Variable temp is dominant }
BOUNDARIES
region 'domain'
  start (0) point value(U)=0  point value(temp)=300
  line to (Lx)  point value(U)=U1  point value(temp)=300
PLOTS
  elevation( temp) from (0) to (Lx)     elevation( fluxd_x) from (0) to (Lx)
  elevation( U) from (0) to (Lx)     elevation( Ex) from (0) to (Lx)
  elevation( Jx) from (0) to (Lx)
END
```

The elevation plot below shows the temperature distribution within the object.



It turns out that the final solution for the temperature becomes symmetrical with respect to the middle of the specimen, which is

what we could expect. If the program had not worked properly, this surface could have been of any shape.

The elevation plot of Ex shows a maximum at mid-distance, which is reasonable considering that the electrical resistivity is largest there.

We also note that the main component of current density, Jx, is constant (with small oscillations) along the object, still according to expectations.

We have thus obtained a plausible solution to a non-linear system of PDEs, involving electrical as well as thermal variables. If we wish to verify the solutions in more detail, we may compare the left and right members of the second equation.

Electrically Heated, Radiating Copper Foil

Let us now replace the copper block by a thin foil of the same material, emitting infrared radiation in the positive and negative *z* directions. In the same manner as in the chapter before (p.183) we lump the radiative loss from the front and rear sides into a (negative) volume heating term.

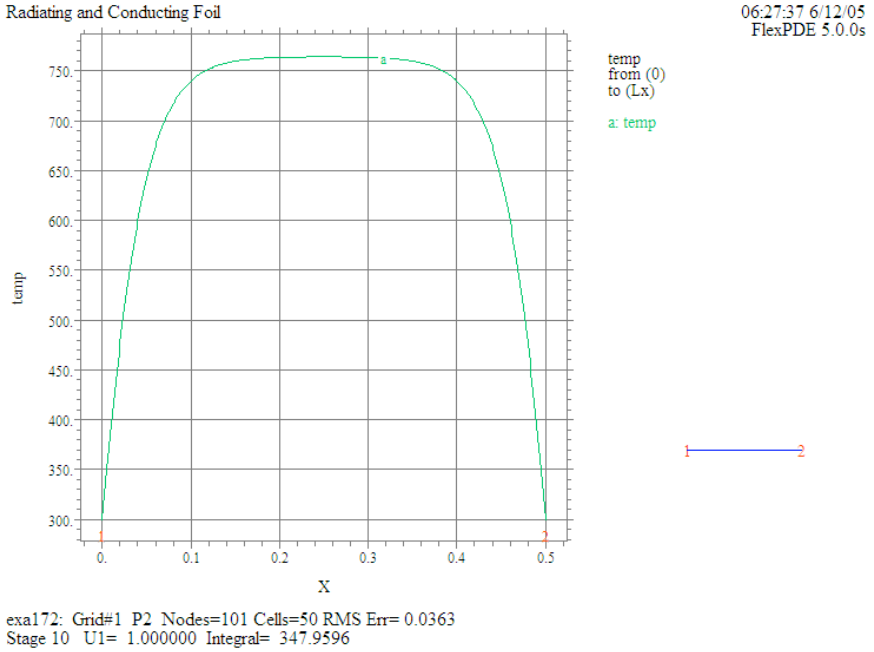
This is a case of strong non-linearity, and the program may work unstably at high radiative loss. For this reason we use a device which permits us to introduce the non-linearity gradually. We first announce that we intend to employ 5 stages. After having made this declaration we may use stage as an integer variable, which automatically steps from 1 to 5. The program solves the equations for the first value of U1, then steps stage to the next higher integer value. The program exploits a solution as the first approximation for the next stage of the calculations.

In the expression for the electrical conductivity cond we have taken the *absolute value* of temp. While searching for the solution the program might occasionally divert to a negative value, which of course would not be physically acceptable, and in fact fatal.

```
TITLE   'Radiating and Conducting Foil'                    { exa172.pde }
SELECT    errlim=1e-4    spectral_colors    stages=10
COORDINATES
  cartesian1                                          { 1D calculation }
VARIABLES   U   temp
```

```
DEFINITIONS
  Lx=0.5     d=1e-4     temp0=300
  U1=stage*0.1     k=400                                    { Copper }
  cond0=1/1.67e-8     cond=cond0*temp0/abs(temp)
  Ex=-dx(U)     E=-grad(U)
  Jx=cond*Ex
  fluxd_x=-k*dx( temp)
  heat=Jx*Ex- 2* 0.3*  5.67e-8* (temp^4-300^4)/d
                    { 2 faces, emissivity, Stefan-Boltzmann formula  }
INITIAL VALUES
  U=0   temp=400
EQUATIONS
  U:          dx( Jx)=0
  temp:       dx( fluxd_x)=heat
BOUNDARIES
region 'domain'
  start (0)  point value( U)=0  point value(temp)=300  line to (Lx)
  point value( U)=U1  point value(temp)=300
PLOTS
  elevation( temp) from (0) to (Lx)  report( U1)
  elevation( fluxd_x) from (0) to (Lx)
END
```

The figure below shows the final elevation plot for temp, the one for the highest voltage.



The temperature distribution is virtually flat over the middle part, where the electrical dissipation balances the loss by radiation. Hence, there is practically no conduction of heat out of this central region of the foil (fluxd_x).

Apart from helping the program to cope with non-linearity, the use of stages evidently gives us a whole series of results for different applied voltages.

The program shows results for all consecutive stages without stopping, which of course saves time. You can view each figure at your ease, however, by clicking on *File* at the upper left corner and then on *View*. This opens a list of the graphics files accumulated, and after choosing the current file you may click your way through the figures.

After many runs with different descriptors, the plot files may take too much space and need to be purged. You can conveniently select and delete the graphics files, and perhaps also the *log* files.

Semicircular Foil, Heated by an Electric Current

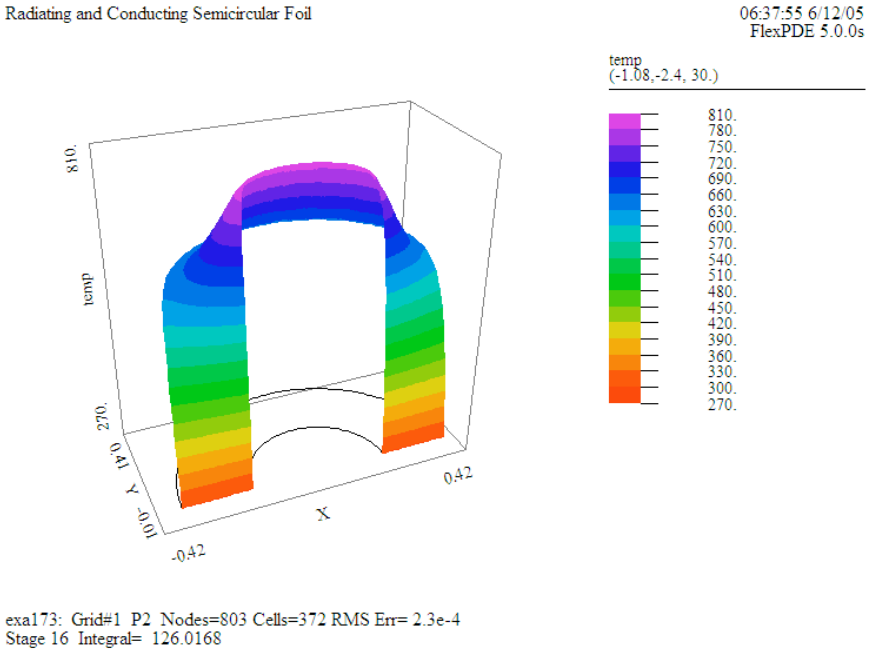
Finally we investigate the temperature distribution in a thin, semi-circular strip heated by a current, taking both radiation and conduction into account as before. The object and the boundary conditions now have less symmetry, however, which means that the solutions will vary significantly in both directions. Hence, we need a contour plot to monitor how the calculations are proceeding.

The following descriptor is analogous to *exa172*.

```
TITLE   'Radiating and Conducting Semicircular Foil'        { exa173.pde }
SELECT
  errlim=1e-4     spectral_colors     stages=16
VARIABLES   U   temp
DEFINITIONS
  r1=0.2     r2=0.4     d=1e-4                                { New geometry }
  U1=stage*0.1     k=400     temp0=300
  cond0=1/1.67e-8     cond=cond0*temp0/abs(temp)
  Ex=-dx(U)     Ey=-dy(U)     E=-grad(U)     Em=magnitude(E)
  Jx=cond*Ex     Jy=cond*Ey     J=cond*E     Jm=cond*Em
  fluxd_x=-k*dx(temp)     fluxd_y=-k*dy(temp)
  fluxd=vector( fluxd_x, fluxd_y)     fluxdm=magnitude( fluxd)
```

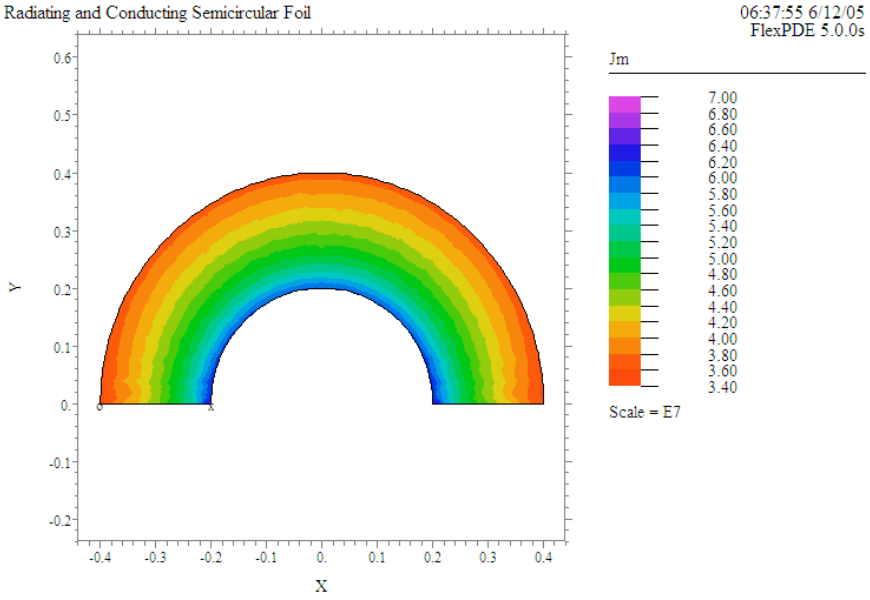
```
  heat=Jm*Em- 2* 0.3*  5.67e-8* (temp^4-300^4)/d
                    { 2 faces, emissivity, Stefan-Boltzmann formula  }
INITIAL VALUES
  U=0   temp=400
EQUATIONS
  U:          div( J)=0
  temp:       div( fluxd)=heat
BOUNDARIES
region 'domain'
  start (-r2,0)  value(U)=U1/2  value(temp)=300  line to (-r1,0)
  natural(U)=0  natural(temp)=0  arc to (0,r1) to (r1,0)
  value(U)= -U1/2  value(temp)=300  line to (r2,0)
  natural(U)=0  natural(temp)=0  arc to (0,r2) close
MONITORS
  contour( cond)
PLOTS
  contour( temp) painted     surface( temp)
  contour( U) painted     contour( Em) painted
  contour( Jm) painted     vector( J) norm
END
```

The following figure is a surface plot of the temperature for the final stage of this example. Evidently, the temperature is considerably higher on the inner edge of the strip than on the outer one. Here, the processes of charge and heat conduction obviously are strongly dependent on each other.



The contour plot of U may give the impression that the potential is constant on radial lines, but this is only approximately true, as indicated by the plot of Em.

The following plot of Jm, on the other hand, appears to be axially symmetric around (0,0). This reflects the fact that the first of the two PDEs is independent of the electrical conductivity $\sigma(T)$. An axially symmetric field **J** automatically satisfies the condition that charge is neither destroyed nor created.



We may of course test the solution set by various plots of the PDE and the boundary conditions, as suggested in earlier chapters.

Exercises

❑ Use an elevation plot to compare the first and second terms in the second equation of *exa171*.

❑ In *exa172* there are in fact three terms in the PDE for heat. Plot curves for each of the two terms in the right member. Also compare the sum of them to the left member of the PDE.

❑ Change *exa172* for the copper foil from rectangular to trapezoidal shape. Let the width at the left be 0.05 as before, and that at the right 0.1.

❑ Calculate the current flowing through the semicircular strip in *exa173* and also the resistance between the ends.
❑ Find the temperature distribution in *exa173* for the fictitious case of vanishing radiation and U1= 0.1 (only the first stage).

18 Transients in One Dimension

In the preceding chapters we treated cases of steady conduction, where the each solution depended only on the space variables. We shall now consider more general problems, explicitly involving the time variable, which are also called *transient* problems.

The time-dependent equation for heat conduction[4p8] is

$$\nabla \cdot \mathbf{f} + \rho c_p \frac{\partial T}{\partial t} = h \qquad \bullet$$

where ρ is the mass density and c_p the specific heat capacity. This PDE just expresses the energy balance for a volume element.

FlexPDE allows us to use the time t as an independent variable, and dt as the partial derivative operator $\partial/\partial t$. This means that the time variable is treated very much like the space variables, and transient problems are hardly more difficult to handle than static ones. We first confine ourselves to two independent variables, t and x.

Iron Bar with a Temperature Step

The descriptor below specifies a rectangular iron bar, initially at 300 K. The keyword *initial values* now has a completely different sense than before. It no longer signifies a guess value, intended to help the program on its way, but it states that at $t = 0$ the object has exactly the temperature given in this segment. The initial value need not be a constant, as in the present case, but could be any function of space.

The other temperature given is that pertaining to the left *boundary* of the object. Here we have entered a constant value, 400, which means that the left face will keep this temperature for $t > 0$. In practice, we could for instance bring the end of the bar into contact with a large copper block, kept at 400 by a thermostat.

There is an analytic solution to a similar problem[4p63], which we may use for comparison to conquer our natural skepticism. That solution applies to a bar with *infinite* extension along the positive *x*-axis. For sufficiently short times, when the temperature at the right end of our bar has not increased noticeably, it should be an excellent approximation. You could verify, by direct substitution, that the following function satisfies the PDE.

$$T = (T_i - T_0)\,\mathrm{erf}\left(\frac{x}{2\sqrt{\lambda t / \rho c_p}}\right) + T_0$$

where $\mathrm{erf}(y) = \frac{2}{\sqrt{\pi}}\int_0^y \exp(-\eta^2)d\eta$ is available in FlexPDE.

The new keyword *time* in the descriptor applies when the problem contains the time explicitly.

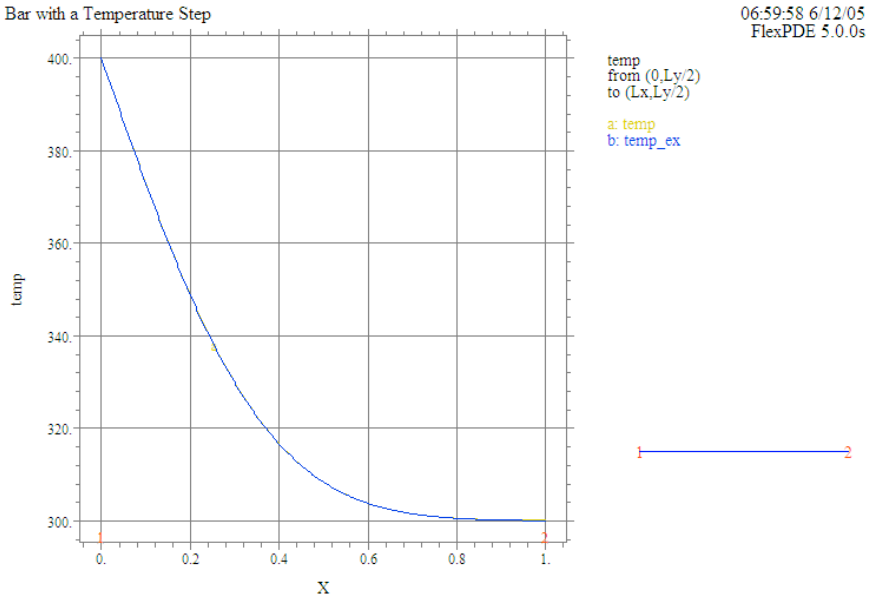
The plot statement involving for...by...to obviously specifies at what times we wish to see snapshots of the solution.

```
TITLE   'Bar with a Temperature Step'                    { exa181.pde }
SELECT                                                   { Student Version }
  errlim=3e-4     spectral_colors
COORDINATES
  cartesian1                                             { 1D }
VARIABLES   temp
DEFINITIONS
  Lx=1.0    Ly=0.1    heat=0    k=82    rcp=7.87e3*449          { Iron }
  tempi=300    temp0=400          { Initial and boundary temperatures }
  fluxd_x=-k*dx(temp)
  temp_ex=(tempi-temp0)*erf[ x/(2*sqrt(k*t/rcp))]+temp0
INITIAL VALUES   temp=tempi
EQUATIONS
  dx( fluxd_x)+ rcp*dt( temp)=heat
BOUNDARIES
region 'domain'  start (0) point value(temp)=temp0 line to (Lx)
TIME   from 0 to 3000
PLOTS
  for t=100 by 100 to 3000
  elevation( temp, temp_ex) from (0,Ly/2) to (Lx,Ly/2)
END
```

Just as with *stages*, the program shows all the plots without stopping. If we later evoke *File*, *View*, however, we may inspect the entire series of plots at our own pace by selecting *View, Next* etc.

Alternatively, we may evoke *View*, *Movie* to display the plot pages at constant speed.

In spite of the difficulties in handling the temperature discontinuity at the left boundary, the program generally yields excellent agreement with the analytic solution. The following snapshot at t=1800, for instance, shows coincident curves and the integral values confirm this agreement.



exa181: Cycle=84 Time= 1800.0 dt= 88.728 P2 Nodes=51 Cells=25 RMS Err= 2.2e-6
Integral(a)= 323.0844 Integral(b)= 323.0577

At the largest time, however, the elevation curves are seen to diverge near the right end. The fault is not with the numerical solution; the difference is a consequence of the different boundary conditions at the right face. The analytic solution applies to the case of infinite extension of the bar, whereas our descriptor specified a finite length. We only need to move the right face further along the *x*-axis to restore the agreement.

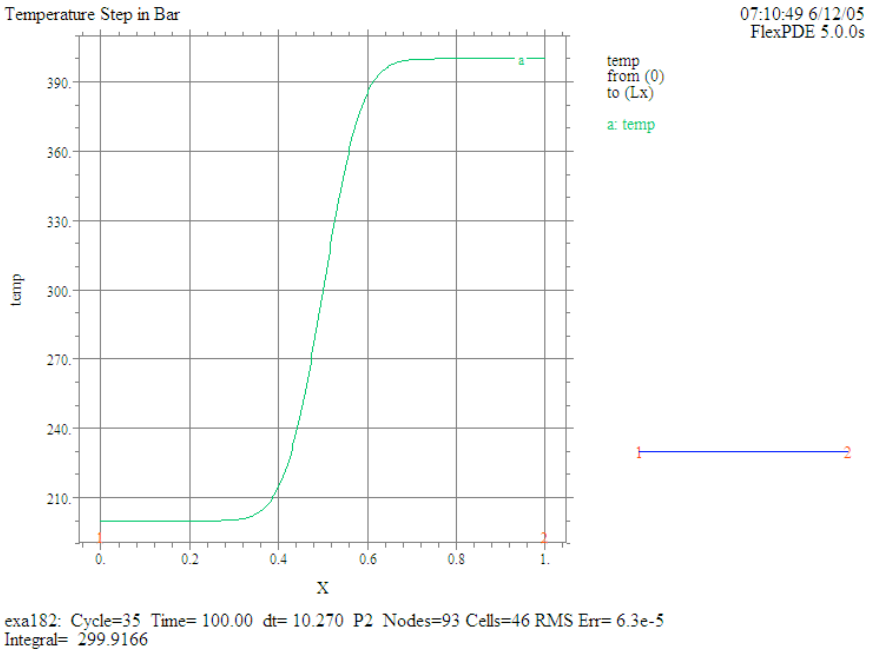
Bar with a Temperature Step Halfway

We shall now modify the preceding descriptor to solve an essentially different problem. Let us insulate all boundaries of the iron bar and introduce a temperature step in the middle. In practice, we could have two equal bars, kept at different temperatures, and join them at $t = 0$. We need to introduce this temperature step in the *initial values* segment, by means of the discontinuous function ustep as shown below. The function ustep(a) is zero at negative values of *a* and jumps to 1 at a=0.

```
TITLE   'Temperature Step in Bar'                          { exa182.pde }
SELECT
  errlim=3e-4     spectral_colors
COORDINATES
  cartesian1
VARIABLES    temp
DEFINITIONS
  Lx=1.0     heat=0     k=82     rcp=7.87e3*449              { Iron }
  fluxd_x=-k*dx(temp)
INITIAL VALUES
  temp=200+200*ustep( x-Lx/2)          { Unit step function }
EQUATIONS
  dx( fluxd_x)+ rcp*dt( temp)=heat
BOUNDARIES
region 'domain'
  start(0) line to (Lx)
TIME   from 0 to 100000
PLOTS
  for t=10, 30, 100, 300, 1000, 3000, 10000, 30000, 100000
  elevation( temp) from (0) to (Lx)
END
```

Here we have a case where there are only *natural* boundary conditions. Any ambiguity is avoided, however, by the *initial values*.

In the plot segment we use a variant of the previous snapshot formula, which is expedient when you prefer a non-uniform distribution of times. The elevation plot below shows the solution at t=100, where the initial step function has become rounded off.



The series of elevation plots demonstrate that the integral of the temperature is independent of time, a result that we readily understand in terms of energy conservation.

Iron Bar Soldered to a Copper Bar

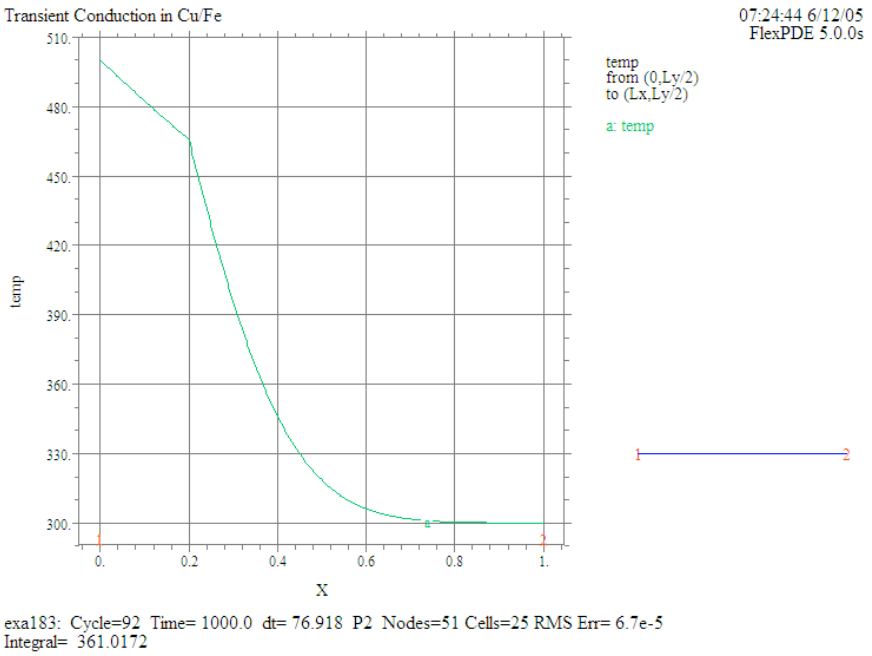
Even in transient problems the materials properties are allowed to vary in space. In the following example we let the left half of the specimen be of copper, the second half of iron. Several lines are as in *exa181*, but we impose temp=500 on the left face.

This descriptor contains the new keywords *histories* and *history*, which permit us to draw curves of the time dependence at various points in space.

```
TITLE   'Transient Conduction in Cu/Fe'                    { exa183.pde }
SELECT
  errlim=3e-4     spectral_colors
COORDINATES
  cartesian1
VARIABLES    temp
DEFINITIONS
  Lx=1.0     Ly=0.1     heat=0     k     rcp
  fluxd_x=-k*dx(temp)
  tempi=300     temp0=500
```

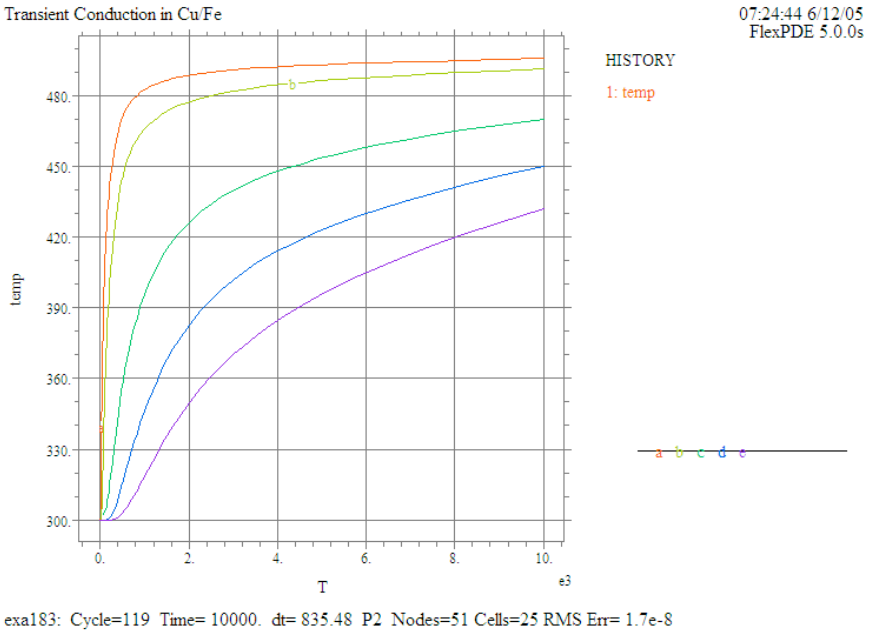
```
INITIAL VALUES
  temp=tempi
EQUATIONS
  dx( fluxd_x)+ rcp*dt( temp)=heat
BOUNDARIES
region 'domain' k=82  rcp=7.87e3*449                    { Fe }
  start (0) point value(temp)=temp0  line to (Lx)
region 'copper'  k=400  rcp=8.96e3* 385
  start (0)  line to (0.2*Lx) to (0.2*Lx)
TIME   from 0 to 10000
PLOTS
  for t=10, 30, 100, 300, 1000, 3000, 10000
  elevation( temp) from (0,Ly/2) to (Lx,Ly/2)
  elevation( fluxd_x) from (0,Ly/2) to (Lx,Ly/2)
HISTORIES
  history( temp) at (0.1*Lx) at (0.2*Lx) at (0.3*Lx)
    at (0.4*Lx) at (0.5*Lx)
END
```

The elevation plots illustrate what happens at the Cu/Fe junction. The plot below shows that the discontinuity in thermal conductivity causes a sudden change in slope of the temperature curve. The x-component of the flux density varies continuously with x, as it should.



The next plot shows the temperature *history* of the composite bar at five different points of the domain. The insert to the right of the

plot indicates the positions of the points where these recordings were taken.



Ramp Function at a Boundary

Boundary values may be functions of time. Let us now pass from a constant value to the simplest extension, i.e. a ramp function. The descriptor mostly follows *exa183*.

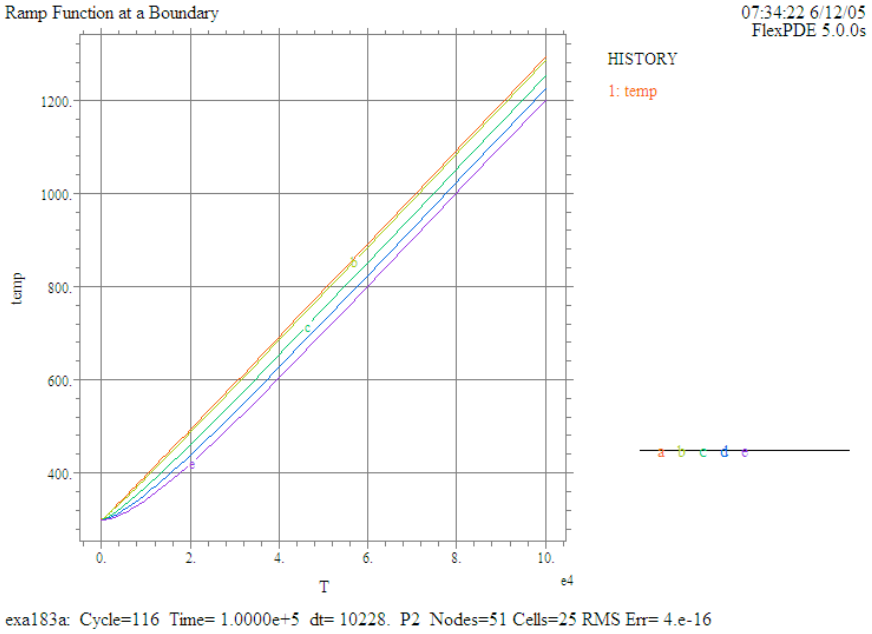
```
TITLE   'Ramp Function at a Boundary'                    { exa183a.pde }
SELECT
  errlim=3e-4     spectral_colors
COORDINATES
  cartesian1
VARIABLES   temp( threshold=1e-3)
DEFINITIONS
  Lx=1.0     Ly=0.1     heat=0     k     rcp
  fluxd_x=-k*dx(temp)
  tempi=300     temp0=300+0.01*t
INITIAL VALUES
  temp=tempi
EQUATIONS
  dx( fluxd_x)+ rcp*dt( temp)=heat
BOUNDARIES
```

```
region 'domain' k=82  rcp=7.87e3*449                    { Fe }
  start (0) point value(temp)=temp0  line to (Lx)
region 'copper'  k=400  rcp=8.96e3* 385
  start (0)  line to (0.2*Lx) to (0.2*Lx)
TIME   from 0 to 100000
PLOTS
  for t=10, 30, 100, 300, 1000, 3000, 10000, 30000, 100000
elevation( temp) from (0,Ly/2) to (Lx,Ly/2)
  elevation( fluxd_x) from (0,Ly/2) to (Lx,Ly/2)
HISTORIES
  history( temp) at (0.1*Lx) at (0.2*Lx) at (0.3*Lx)
    at (0.4*Lx) at (0.5*Lx)
END
```

We have now imposed a time-dependent temperature on the left face, starting at temp=300, which happens to be equal to the initial temperature.

There is now a *threshold* declaration for the temperature variable, giving the program a hint about tolerable *absolute* errors. Even if we omit this statement, the solution of this particular example will be successful. In other cases, however, the program might have to work long and hard in the initial phase. It would notice that the temperature increase is very slight in the beginning and try to produce a solution with an error less than 3e-4 of that small increment.

The *history* plot below demonstrates that the temperature curves eventually become almost parallel.



Exercises

- ❑ Modify *exa181* by inputting a suitable flux density through the left face.
- ❑ Modify *exa181* again by dissipating power uniformly in the bar at 1e6 W/m^3, keeping the other parameters unchanged. Explain the resulting curves.
- ❑ Starting with *exa182*, introduce a different material (Cu) in the right half of the bar. Will the integral of temp still be independent of time? Could you find a function other than the temperature that would yield an invariant integral?
- ❑ Modify *exa183* by keeping the right face at 300 K.
- ❑ Try another function of time at the left face in *exa183a*. For instance, choose the function temp0=300+100*ustep(1000-t) and add a plot of temp0.

19 Transients in (x,y) and (ρ,z) Space

In the preceding chapter all examples were effectively in one space dimension, but there is nothing more to learn about descriptors in order to treat problems in two space dimensions and the time. We only need to supply the appropriate boundary conditions.

Internally Heated Steel Bar with a Loss

In the following example a steel bar with rectangular cross section is heated by electromagnetic induction, such that the heat dissipation is uniform and constant. Wads of glass fiber tend to insulate the bar from the rectangular enclosure, which is kept at 300 K by external cooling.

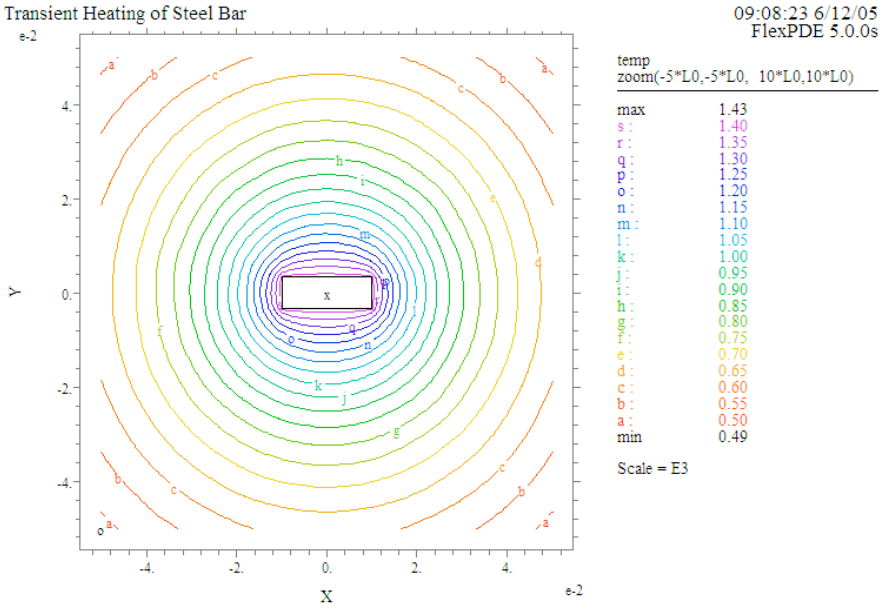
In this example the program has no inkling beforehand of how much the temperature is going to rise. For this reason we supply a rough estimated value under *variables*. This shortens the calculation time considerably.

In order to reduce the number of plots, while still reproducing the main features of the heating process, we use an approximately logarithmic scale of times for the plots.

```
TITLE   'Transient Heating of Steel Bar'                    { exa191.pde }
SELECT                                              { Student Version }
  errlim=1e-3    ngrid=1    spectral_colors
VARIABLES   temp( threshold=0.1)                    { Absolute tolerance }
DEFINITIONS
  L=100e-3    L0=10e-3    heat    k    rcp    tempi=300
  fluxd_x=-k*dx(temp)    fluxd_y=-k*dy(temp)
  fluxd=vector( fluxd_x, fluxd_y)    fluxdm=magnitude( fluxd)
  power_d=1e7                                       { Power density }
  power=(2*L0)* (2*L0/3)*power_d                    { Power from bar }
INITIAL VALUES
  temp=tempi
EQUATIONS
```

```
  div( fluxd)+ rcp*dt( temp)=heat
BOUNDARIES
region 'domain'  heat=0  k=0.5  rcp=2e6                { Glass fiber }
  start (-L,-L)  value( temp)=tempi
  line to (L,-L) to (L,L) to (-L,L)  close
region 'steel'  heat=power_d  k=45  rcp=3.5e6          { Stainless steel }
  start (-L0,-L0/3) line to (L0,-L0/3) to (L0,L0/3) to (-L0,L0/3) close
TIME   from 0 to 1e6
PLOTS
  for t=100, 300, 1e3, 3e3, 1e4, 3e4, 1e5, 3e5, 1e6
  contour( temp)     contour( (temp-tempi)*rcp) report( power*t)
  contour( temp) zoom(-5*L0,-5*L0,  10*L0,10*L0)
END
```

As shown by the first plot the temperature contours become rounded-off squares on approaching the outer boundary. The zoomed contour plot below shows the final temperature distribution around the bar. We notice that the contours become almost circular at a distance of one bar length from the origin.



The plot of (temp-tempi)*rcp automatically displays the increase in enthalpy (heat energy) over the entire domain. We may compare these values to the heat delivered from the bar (power*t) by *File*, *View*. Up to a time of 1000 s the pairs of values agree to high precision, but at larger times the automatic integral gradually

becomes smaller than the input heat energy. The cause of this deviation is the loss of heat through the boundary, which is at constant temperature.

If we replace the *value* boundary condition by complete insulation (natural(temp)=0), the agreement becomes perfect.

Capsule and Sample in a Scanning Calorimeter

When using a differential scanning calorimeter (DSC) one encloses a sample in a metal capsule, heated by contact with a base plate of linearly varying temperature. The instrument measures the power necessary to make the sample follow this temperature ramp. The question arises to what extent the sample retains a uniform temperature during this process.

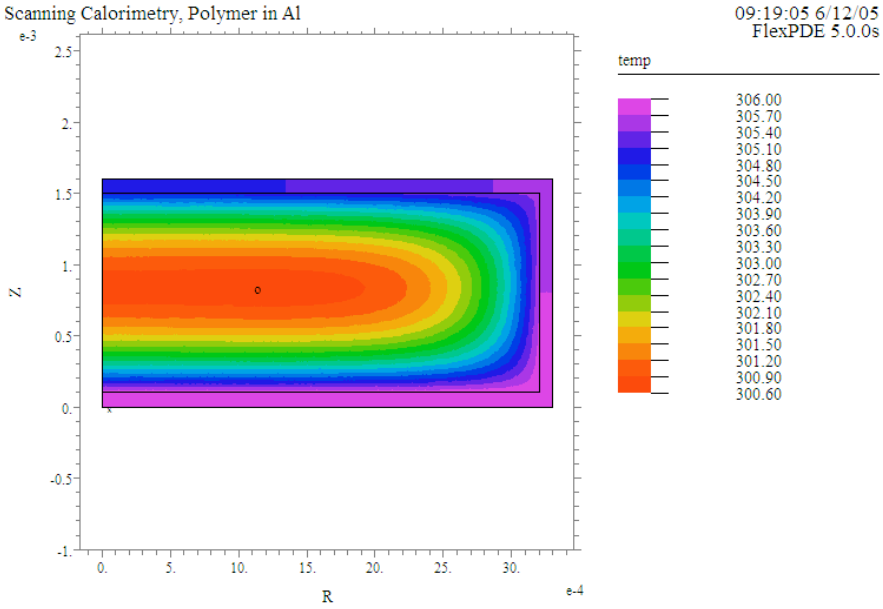
The following descriptor defines the geometry and the thermal properties of an aluminum capsule, filled with a polymeric sample. The capsule is axially symmetric, and we thus use the PDE pertinent to cylindrical coordinates. We assume that the temperature at the bottom of the capsule varies linearly at a rate of 2.0 K/s.

In the plot section we use the *viewpoint* command. The first two arguments of *viewpoint* are the ρ and z coordinates of the eye of the viewer, and the third argument is its angular elevation above the (ρ,z) plane.

```
TITLE   'Scanning Calorimetry, Polymer in Al'                { exa192.pde }
SELECT
  errlim=1e-3     ngrid=1     spectral_colors
COORDINATES   ycylinder( 'r', 'z')
VARIABLES   temp( threshold=0.01)
DEFINITIONS
  D=0.1e-3     r0=3.3e-3     Lz=1.6e-3     k     rcp
  heat=0     rate=2.0     tempi=300
  fluxd_r=-k*dr(temp)     fluxd_z=-k*dz(temp)
  fluxd=vector( fluxd_r, fluxd_z)     fluxdm=magnitude( fluxd)
INITIAL VALUES
  temp=tempi
EQUATIONS
  div( fluxd)+ rcp*dt( temp)=heat
BOUNDARIES
```

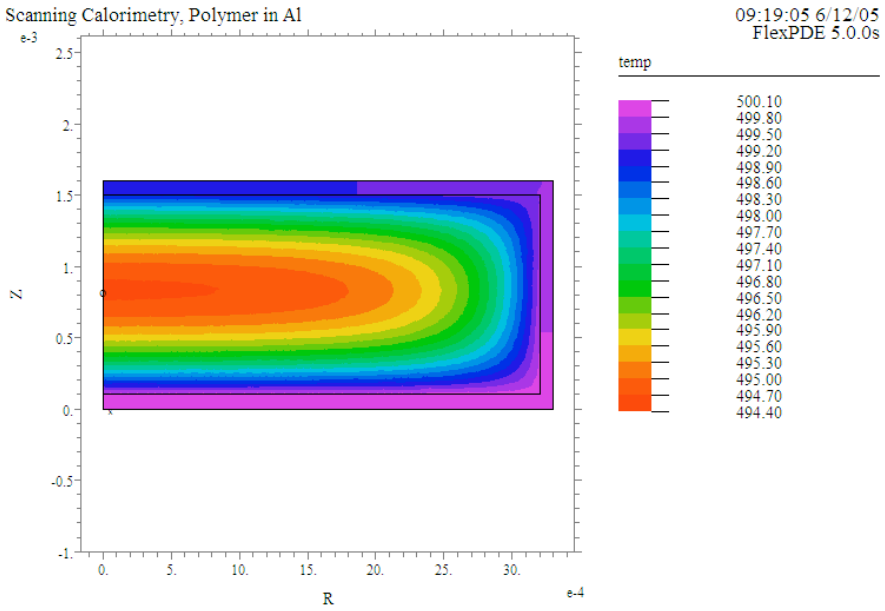
```
region 'domain'   k=0.3   rcp=3e6                                  { Polymer }
start (0,0)  value(temp)=tempi+ rate*t  line to (r0,0)
  natural(temp)=0  line to (r0,Lz) to (0,Lz) close
region  'aluminum'   k=238   rcp=3e6
  start (0,0)  line to (r0,0) to (r0,Lz) to (0,Lz)
  to (0,Lz-D) to (r0-D,Lz-D) to (r0-D,D) to (0,D) close
TIME   from 0 to 100
PLOTS
  for t=1, 3, 10, 30, 100
  elevation(temp) from (0,0) to (0,Lz)      contour(temp) painted
  surface(temp) viewpoint( -r0, Lz/2, 0)
END
```

The figure below shows the temperature distribution after 3 s. The heat spreads from below, and we see that the aluminum capsule conducts very well. In the middle of the specimen, however, the temperature has only just started to rise above the initial value.



exa192: Cycle=85 Time= 3.0000 dt= 0.2567 P2 Nodes=790 Cells=379 RMS Err= 3.4e-4
Vol_Integral= 1.661750e-5

The next figure shows the temperature field at 100 s. It is clear that the top of the capsule has not quite reached the temperature of the bottom plate. The interior of the specimen lags behind even more than the top of the capsule. It is interesting to observe how these conditions change as the measurements proceed.



Moving Heat Source

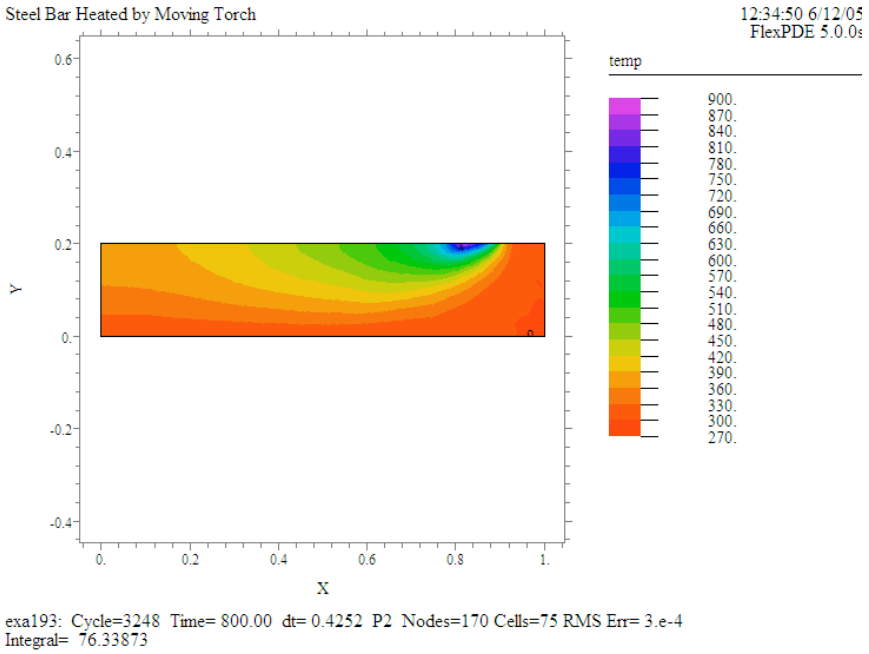
We shall now consider a steel bar of rectangular cross-section, heated by a welding torch moving to the right along its top surface. We assume the torch to heat uniformly over a narrow rectangular area across the bar. The front, rear and end faces of the bar are assumed to be adequately insulated by air, while the bottom is kept at T= 300.

We model the torch by two step functions (ustep) providing a constant flux density (fluxd0) over a short interval of x.

```
TITLE   'Steel Bar Heated by Moving Torch'                    { exa193.pde }
SELECT
  errlim=1e-3    ngrid=1    spectral_colors
VARIABLES   temp( threshold=0.1)
DEFINITIONS
  Lx=1.0    Ly=0.2    d0=0.1    vx= 1e-3              { Torch velocity }
  heat=0    tempi=300    k=82    rcp=7.87e3*449    { Steel }
  fluxd_x=-k*dx(temp)    fluxd_y=-k*dy(temp)
  fluxd=vector( fluxd_x, fluxd_y)    fluxdm=magnitude( fluxd)
  fluxd0=-1e6*[ ustep( vx*t+d0- x)- ustep( vx*t- x) ]
INITIAL VALUES
  temp=tempi
EQUATIONS
```

```
  div( fluxd)+ rcp*dt( temp)=heat
BOUNDARIES
region 'domain'
  start (0,0)  value(temp)=tempi  line to (Lx,0)
  natural(temp)=0 line to (Lx,Ly)
  natural(temp)=fluxd0  line to (0,Ly)
  natural(temp)=0 line to close
TIME   from 0 to 800
PLOTS
  for t=50 by 50 to 800
  elevation( temp) from (0,Ly) to (Lx,Ly)
  contour( temp) painted     vector( fluxd) norm
END
```

The following is the last contour plot of the sequence.



Inspecting the other plots of this kind as they are produced, one finds that the shape of the contours tends to a stationary pattern. In other words, the last plot may be obtained by displacing the preceding one by the proper distance in the positive *x* direction.

Stationary State for a Moving Heat Source

The existence of a stationary state of the temperature field suggests that it might be possible to obtain the time-independent field by treating the process in a coordinate system that is *moving with the torch*. In terms of the old coordinates (x', y) the time-dependent PDE reads (with ∇' as a reminder of x')

$$\nabla' \cdot \mathbf{f} + \rho c_p \frac{\partial T}{\partial t} = h$$

We now change to the new variable $x = x' - v_x t$. The spatial partial derivatives remain the same after this transformation, but the time derivative becomes a spatial one. The term x' does not modify the differential, since this is the spatial variable in the above PDE. Hence, we have $\Delta x = -v_x \Delta t$ and the time derivative transforms according to

$$\frac{\partial T}{\partial t} \rightarrow -v_x \frac{\partial T}{\partial x}$$

Our new PDE for the stationary state thus becomes

$$\nabla \cdot \mathbf{f} - \rho c_p v_x \frac{\partial T}{\partial x} = 0$$ ●

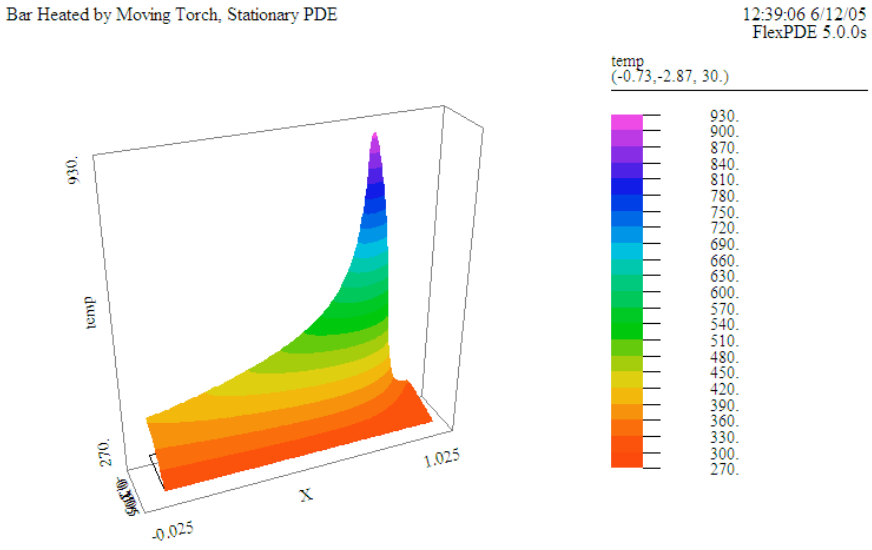
The descriptor file required to test this model is of course quite similar to the preceding one. We make a change of variable so that the torch becomes effectively stationary, and modify the PDE accordingly.

```
TITLE   'Bar Heated by Moving Torch, Stationary PDE'     { exa194.pde }
SELECT
  errlim=1e-3     ngrid=4     spectral_colors
VARIABLES   temp
DEFINITIONS
  Lx=1.0      Ly=0.2     d0=0.1     vx=1e-3                { Velocity }
  k=82     rcp=7.87e3*449                                  { Steel }
  heat=0     tempi=300     t0=800
  fluxd_x=-k*dx(temp)     fluxd_y=-k*dy(temp)
  fluxd=vector( fluxd_x, fluxd_y)     fluxdm=magnitude( fluxd)
  fluxd0=-1e6*[ ustep( vx*t0+d0- x)- ustep( vx*t0- x) ]
EQUATIONS
```

```
  div( fluxd)- rcp*vx*dx(temp)=heat
BOUNDARIES
region 'domain'
  start (0,0)  value(temp)=tempi  line to (Lx,0)
  natural(temp)=0 line to (Lx,Ly)
  natural(temp)=fluxd0  line to (0,Ly)
  natural(temp)=0 line to close
PLOTS
  elevation( temp) from (0,Ly) to (Lx,Ly)
  contour( temp) painted     surface( temp)     vector( fluxd) norm
END
```

The plots resulting from the above file are expected to approach those obtained at the largest times in the transient problem (*exa193*). The contour plot on p.209 for the largest time is evidently similar to the stationary one.

The following figure is a surface plot of the temperature for the stationary case.



exa194: Grid#5 P2 Nodes=507 Cells=236 RMS Err= 4.5e-4
Integral= 78.95313

Exercises

❑ Modify *exa191* to display the flux leaving the bar. Compare that value to the power dissipated by induction.
❑ How does the scanning calorimeter in *exa192* perform at lower rates of heating, say at 0.1 K/s, with respect to temperature uniformity in the specimen?
❑ Run *exa192* again, this time including a thermal resistance between the base plate and the bottom of the capsule. Simulate the thermal resistance by an extra plate of suitable thermal conductivity and negligible heat capacity.
❑ Let the MgO plate in *exa161* (p.180) initially be at 300 K and apply sudden heating to 1000 K for $t>0$ at the left edge. Display the magnitude of the heat flux density as well.

20 Time-Sinusoidal Problems

Let us now consider a particular kind of time-dependence, i.e. where a boundary value or a source term oscillates around an average. For instance, the heating power per unit volume could vary with time according to $h = h_0 \,\mathrm{Re}(\exp(i\omega t))$, where h_0 is a time-constant amplitude and Re(z) means the real part of the complex quantity z.

The equations for a periodic problem need not involve the time explicitly. For the equation of heat conduction[4p8] (with $\mathbf{f} = -\lambda \nabla T$)

$$\nabla \cdot (-\lambda \nabla T) + \rho c_p \frac{\partial T}{\partial t} = h \qquad \bullet$$

we attempt a solution involving the complex amplitude T_0, i.e.

$$T = T_0(\mathbf{r}) \exp(i\omega t) = \left[T_{0r}(\mathbf{r}) + i T_{0i}(\mathbf{r})\right]\left[\cos(\omega t) + i \sin(\omega t)\right] \qquad \bullet$$

Separating this solution into its real and imaginary parts we obtain

$$T_r = T_{0r} \cos(\omega t) - T_{0i} \sin(\omega t), \qquad T_i = T_{0r} \sin(\omega t) + T_{0i} \cos(\omega t)$$

and for the time derivatives

$$\partial T_r / \partial t = -\omega [T_{0r} \sin(\omega t) + T_{0i} \cos(\omega t)] = -\omega T_i$$

$$\partial T_i / \partial t = \omega [T_{0r} \cos(\omega t) - T_{0i} \sin(\omega t)] = \omega T_r$$

where the real and imaginary amplitudes are functions of the spatial variables.

Substituting $T = T_r + i T_i$ and $\partial T / \partial t = -\omega T_i + i \omega T_r$ into the PDE we obtain two equations, one for the real term and one for the imaginary term, as follows

$$\begin{cases} \nabla \cdot (-\lambda \nabla T_r) - \omega \rho c_p T_i = h_0 \\ \nabla \cdot (-\lambda \nabla T_i) + \omega \rho c_p T_r = 0 \end{cases} \qquad \bullet$$

Here we assume that the materials properties (λ, ρ and c_p) are independent of T, although they may vary with the space coordinates, even in a discontinuous fashion. For small temperature oscillations,

this would be a good approximation. The system of PDEs is then linear.

The interpretation of the solution is completely analogous to that commonly used with vibrating structures and alternating current circuits[1p97]. At any given time, the temperature must of course have a real value. The actual temperature increment is given by the *real* part of the complex solution (including the time factor), viz.

$$T = T_{0r}\cos(\omega t) - T_{0i}\sin(\omega t) \qquad \bullet$$

This means that once we have solved for T_r and T_i we know the temperature over the whole domain and at all times.

Oscillating Temperature in a Steel Block

As a first example we choose the simplest possible situation: a steel block, insulated except on the front face, where the temperature oscillates around 300 K at a constant amplitude and phase. There are two dependent variables, the real and imaginary parts of *T*, both of which must be declared. We must also supply boundary conditions for both.

The imposed temperature oscillation, applied on the left side, is chosen to be real. Oscillations at any point of the object hence refer to this input phase angle (0). Since the attenuation of the temperature wave is extremely strong we need logarithmic plots to present the variation.

There is an analytic solution to a related problem that we may use for comparison. This solution is valid for the same boundary conditions, except that the right boundary is at infinite distance. In view of the strong attenuation, this difference may not always be significant. The exact solution[4p65] is

$$T_{ex} = a_0 \exp\left(-s_q x(1+i)\right) \quad \text{with} \quad s_q = \sqrt{\omega \rho c_p / (2\lambda)} \qquad \bullet$$

which yields, for the real part,

$$T_{r,ex} = a_0 \exp(-s_q x)\cos(-s_q x)$$

and similarly for the imaginary part. You may easily verify that these functions satisfy the PDEs and the boundary conditions.

Here, we must remember to *tag* the PDEs with the dominant dependent variables. On p.213 we separated the complex PDE of heat conduction into a real and an imaginary part, and we just tag the equations accordingly.

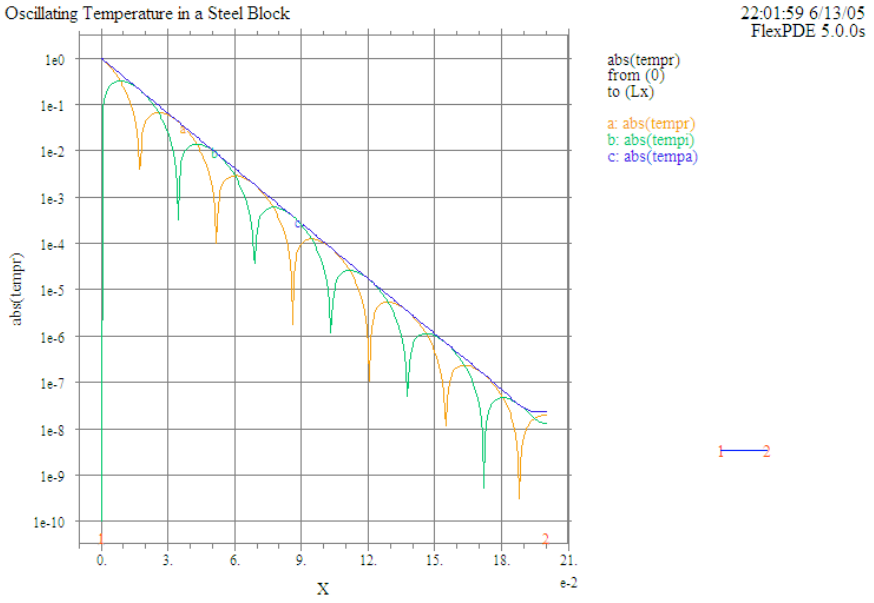
```
TITLE   'Oscillating Temperature in a Steel Block'          { exa201.pde }
SELECT                                                      { Student Version }
  errlim=3e-5     spectral_colors
COORDINATES
  cartesian1                                                { 1D }
VARIABLES   tempr   tempi
DEFINITIONS
  Lx=0.2     heat=0     k=45    rcp=3e6                     { Steel }
  a0=1                                                      { tempr at left boundary }
  omega=0.25                                                { Angular frequency }
  tempa=sqrt(tempr^2+ tempi^2)                              { Amplitude }
  phase=sign(tempi)*arccos(tempr/tempa)/pi*180     { Angle, degrees}
  sq=sqrt( omega*rcp/(2*k))                                 { Exact solutions ...}
       tempr_ex=a0*cos(-sq*x)*exp(-sq*x)
       tempi_ex=a0*sin(-sq*x)*exp(-sq*x)
       tempa_ex=a0*exp(-sq*x)
EQUATIONS      { Real and imaginary PDE }
  tempr:    dx(-k*dx( tempr))- omega*rcp*tempi=heat
  tempi:    dx(-k*dx( tempi))+ omega*rcp*tempr=0
BOUNDARIES
region 'domain'  start (0) point value(tempr)=a0  point value(tempi)=0
  line to (Lx)              { Imposed temperature oscillation at x=0 }
PLOTS
  elevation( tempr) from (0) to (Lx)
  transfer( tempr) file='tempr'                             { Store data in file }
  elevation( tempi) from (0) to (Lx)
  transfer( tempi) file='tempi'
  elevation( phase) from (0) to (Lx)
  elevation( abs(tempr), abs(tempi), abs(tempa)) log(10)
    from (0) to (Lx)
  elevation( abs(tempr), abs(tempr_ex)) log
    from (0) to (Lx)
  elevation( abs(tempa), abs(tempa_ex)) log
    from (0) to (Lx)
END
```

After the first two elevation plot statements we have added lines to transfer the solution to a file. For the first plot, the name of this file becomes 'tempr'. We shall use these transferred files later for creating animated plots.

As the first plot of tempr indicates, the real part oscillates between positive and negative values. Consequently, we should interpret the sharp minima in the plot of abs(tempr) as being effectively zeros.

In such plots, the interval from one zero crossing to the next is one-half wavelength. The comparison with the analytic solution shows reasonable agreement, down to well below of 10^{-7} of the input amplitude.

The figure below is a combined elevation plot of the absolute values of the solution. Here, we have limited the logarithmic scale to 10 decades.



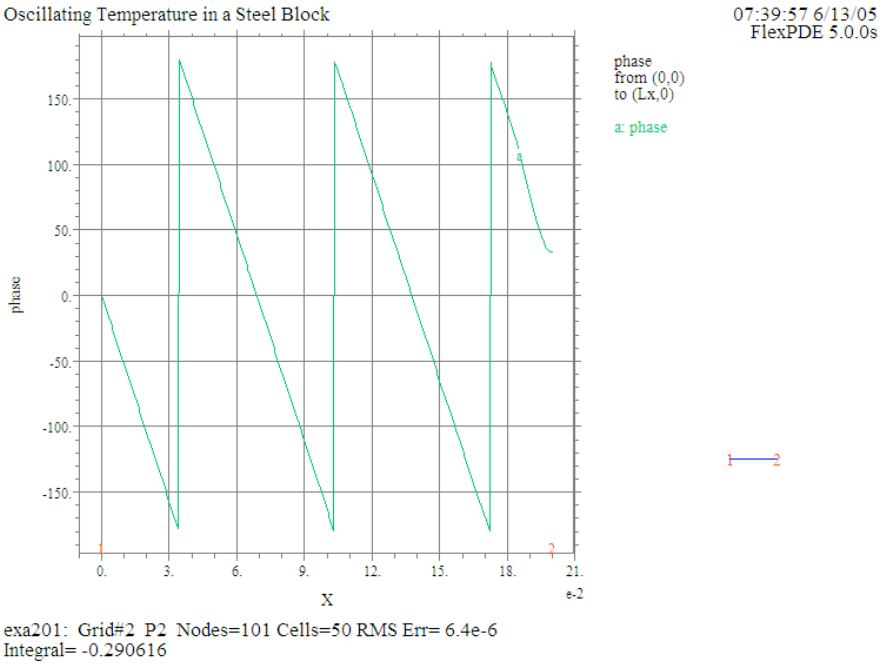
For particular values of the time, corresponding to $\omega t = 2\pi n$ (n being an integer), the cosine in the expression for T (p.214) becomes equal to 1 and the sine vanishes. For these times, the temperature increment is thus simply given by

$$T = T_{0r}\cos(\omega t) - T_{0i}\sin(\omega t) = T_{0r}$$ ●

Thus the real part $T_r(x)$ shown in the above figure is a "snapshot" of the temperature distribution at particular points in time. Similarly, $T_i(x)$ yields the temperature distribution at middle points. The

agreement with respect to wavelengths, shown by the plots of the real part, also seems satisfactory. The real or imaginary parts of a temperature oscillation may be measured by lock-in techniques, using the heater voltage as the reference signal.

The following figure shows the *phase angle* of the solution versus *x*. Evidently the angle continuously decreases as we move to the right, until the value reaches -180^0, where the angle shifts by 360^0 to display the next period. If we so wish, we could ignore the vertical jumps and consider the phase angle to be presented by a continuous curve, forever decreasing. This plot shows, in even greater detail, how far the calculations reproduce the wave: after 2.5 wavelengths the distortion becomes noticeable.



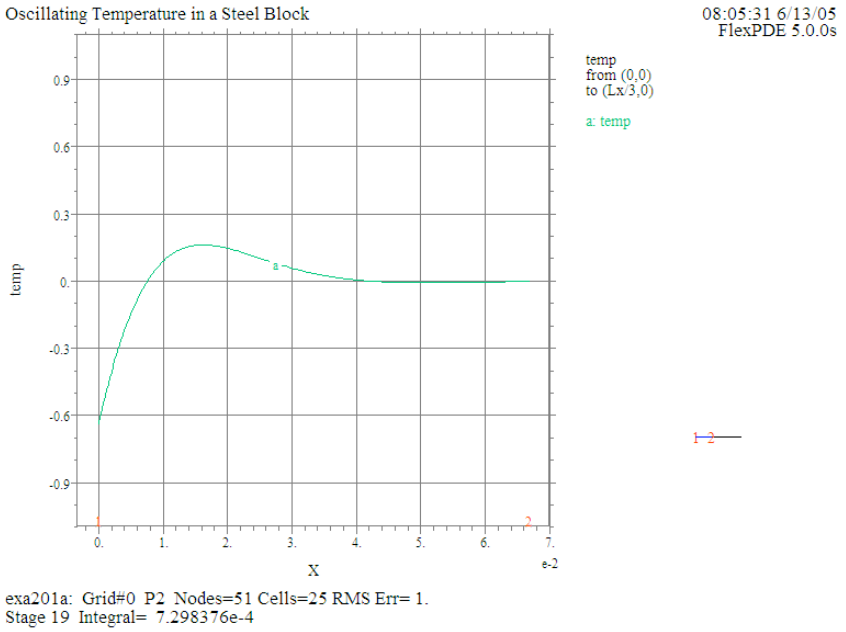
Temperature waves are commonly observed natural phenomena. In summertime the sun heats the surface of the earth much more intensely than during winters, and the result is a temperature wave, which of course is largest at ground level. These oscillations may, however, also be measured deep down in a mineshaft – attenuated and retarded as suggested by our calculations.

Animation of an Oscillating Temperature

The following file combines the real and imaginary parts into temperature, which we may plot for a sequence of times by the stages device. The command transfer transforms the data files into functions that may be plotted.

```
TITLE   'Oscillating Temperature in a Steel Block'          { exa201a.pde }
SELECT   stages=200
COORDINATES
  cartesian1
DEFINITIONS
  Lx=0.2
  transfer('tempr', tempr)     transfer('tempi', tempi)
  wt=(stage-1)/50* 2*pi                          { Omega*t }
  temp=tempr*cos(wt)- tempi*sin(wt)      { Temperature }
BOUNDARIES
region 'domain'  start (0) line to (Lx)
PLOTS
  elevation( temp) from (0) to (Lx/3) fixed range(-1,1)
END
```

After running *exa201* and *exa201a* we obtain an animated plot of the temperature as a function of both space and time. The modifier fixed range avoids disturbing changes of scale during the presentation. The following is a sample of the plot sequence.



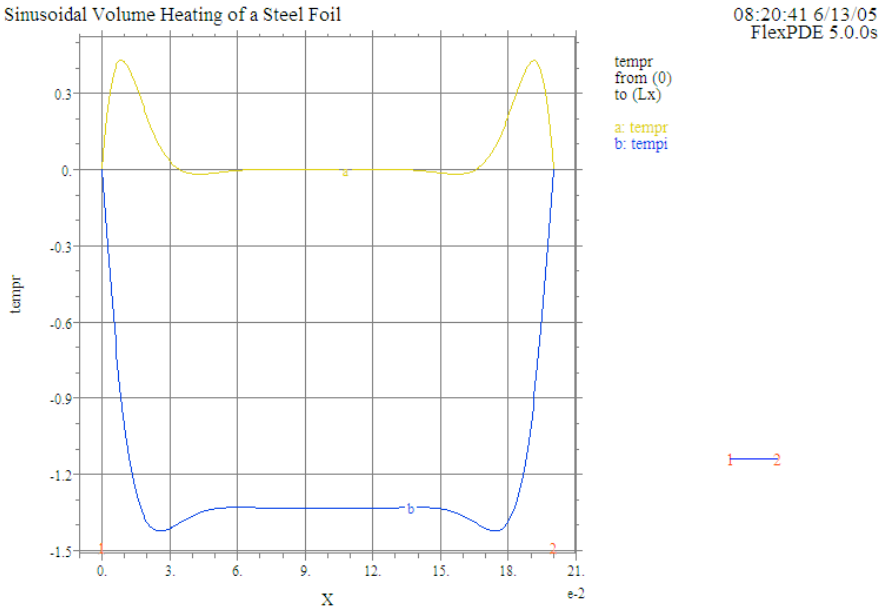
The *File*, *View* option using *View*, *Movie* makes the graphics rendering more attractive.

Sinusoidal Volume Heating of a Steel Foil

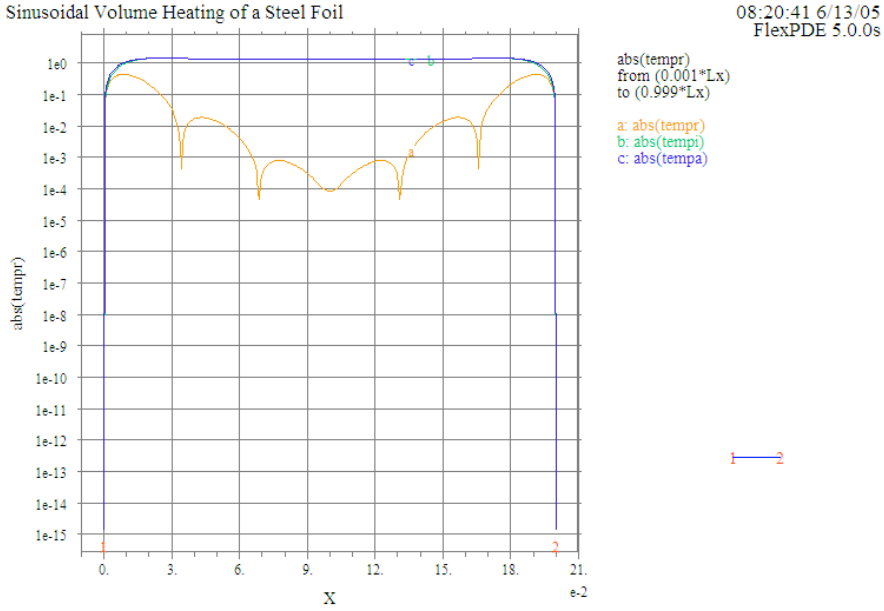
The next case for study is a steel sheet, clamped between copper jaws, so that the left and right edges are forced to have a constant temperature. The jaws are also used to conduct an electric current producing a sinusoidally varying power. Most of *exa201* still applies and may be used as a template.

```
TITLE   'Sinusoidal Volume Heating of a Steel Foil'          { exa202.pde }
SELECT
  errlim=3e-5     spectral_colors
COORDINATES
  cartesian1                                          { 1D }
VARIABLES     tempr     tempi
DEFINITIONS
  Lx=0.2 k=45     rcp=3e6     heat=1e6                { Real value }
  omega=0.25                                          { Angular frequency }
  tempa=sqrt(tempr^2+ tempi^2)                        { Amplitude }
  phase=sign(tempi)*arccos(tempr/tempa)/pi*180
EQUATIONS
  tempr:     dx(-k*dx( tempr))- omega*rcp*tempi=heat
  tempi:     dx(-k*dx( tempi))+ omega*rcp*tempr=0
BOUNDARIES
region 'domain'
  start (0) point value( tempr)=0 point value( tempi)=0
  line to (Lx) point value( tempr)=0 point value( tempi)=0
PLOTS
  elevation( tempr, tempi) from (0) to (Lx)
  elevation( phase) from (0) to (Lx)
  elevation( abs(tempr), abs(tempi), abs(tempa)) log
    from (0.001*Lx) to (0.999*Lx)
END
```

The plot below shows that the real and imaginary parts both are zero at the left and right edges, as required. In the middle of the sheet the imaginary part dominates, i.e. the temperature oscillations are retarded by about 90^0 with respect to the power applied.



The logarithmic plot of the real part, shown below, indicates the wavelength.



The two examples above were effectively one-dimensional, but thermal waves may equally well be studied in 2D.

Exercises

❑ In *exa201* the curve of absolute values diverges gradually from the exact one as we approach the right edge. Test if this is due to the influence of the right boundary, by moving it further to the right.

❑ Modify *exa201* by inputting an oscillating *flux density* on the left surface, instead of an oscillating temperature. What is the major change you observe in the solution? Modify the exact solution, by allowing for a complex amplitude (*a0*), and compare to the FEA solution.

❑ Modify *exa201* into a two-dimensional problem, where the heater acts over only a small part at the center of the left face (Ly=0.1). Let the width of the surface heater be 0.04.

❑ Animate the solution for *exa202* using a surface plot, suitably zoomed.

21 Electron and Hole Conduction

In the chapter on electric fields inside conductors (p. 63) we assumed that charge could move freely inside such an object. We treated an electric current in a semi-classical way without even mentioning the role of electrons. Furthermore, we did not introduce the temperature as a crucial parameter in the conduction process. The new phenomenon encountered in this chapter is the *diffusion of charge carriers*, which is the random transport caused by the thermal motion of atoms.

The conductivity is limited by electrons interacting with atoms, being scattered off their course but moving along the field (**E)** on the average. Only a minority of the electrons (usually one or two per atom) are free to move and to participate in the charge transport. We denote by n_{e0} the number of free electrons per unit volume of a homogeneous metal that is uncharged and not exposed to an external field. In a general situation, the *local* free-electron density n_e may be different from n_{e0}.

Steady-State Electron Current

We shall confine the following discussion to the case of a current field that does not change with time. Electrons *drift* in a direction opposite to **E**, while they *diffuse* in the direction opposite to the electron concentration gradient. We may thus express the *flux density* of electrons as

$$\mathbf{f}_e = -\mu_e n_e \mathbf{E} - D_e \nabla n_e \qquad \bullet$$

where n_e is the local electron concentration and μ_e the electron mobility. The factor D_e is the diffusion coefficient given by

$$D_e = \mu_e k_B T / q \qquad \bullet$$

in terms of the Boltzmann constant k_B, the thermodynamic temperature T, and the elementary charge q (absolute value).

Multiplying the above expression by the electron charge $(-q)$ we immediately obtain the corresponding electric current density

$$\mathbf{J}_e = q\, n_e \mu_e \mathbf{E} + q\, D_e \nabla n_e \quad \bullet$$

The first term in the expression for $\mathbf{J}_e$ corresponds to the drift of electrons along the field, i.e. $q\, n_e \mu_e \equiv \sigma$ (conductivity).

The second term is due to thermal unrest, which causes diffusion of the electrons to an extent depending on the temperature.

In a conductor that is not exposed to an external field the free electrons are compensated by a corresponding number of positive ions, and hence the overall charge density is zero. We now introduce the *local charge density* by $\rho = (-q)(n_e - n_{e0})$, which yields

$$q n_e = q n_{e0} - \rho$$

Substituting this expression into the above equation we find

$$\mathbf{J}_e = q\, n_e \mu_e \mathbf{E} + q D_e \nabla n_e = (q n_{e0} - \rho)\mu_e \mathbf{E} + D_e \nabla (q n_{e0} - \rho) \quad \bullet$$

We could consider $\mathbf{E}$ and ρ to be the dependent variables in this PDE. One of the Maxwell equations provides the second PDE, i.e.

$$\rho = \nabla \cdot \mathbf{D} = \nabla \cdot (\varepsilon \mathbf{E}) \quad \bullet$$

If we now substitute this expression for ρ into the preceding equation we obtain the non-linear PDE

$$\mathbf{J}_e = q n_{e0} \mu_e \mathbf{E} - \nabla \cdot (\varepsilon \mathbf{E})\, \mu_e\, \mathbf{E} + D_e q \nabla n_{e0} - D_e \nabla^2 (\varepsilon \mathbf{E}) \quad \bullet$$

Hence, if $\mathbf{J}_e$ is known inside the conductor and $\mathbf{E}$ on the boundary, we should be able to solve for the components of $\mathbf{E}$.

One-Dimensional Conduction in a Metal

The above PDE for the steady state simplifies considerably if $E_y = 0$ and E_x is a function of x only. Then $(\mathbf{J}_e)_x$ will also depend only on x, and charge conservation in fact requires it to be constant. The above PDE then takes the form

$$J_0 = q\mu_e n_{e0} E_x - \frac{\partial(\varepsilon E_x)}{\partial x}\mu_e E_x + D_e q\frac{\partial n_{e0}}{\partial x} - D_e\frac{\partial^2(\varepsilon E_x)}{\partial x^2}$$

We now assume that the basic electron density n_{e0} is constant in space, which means that we may eliminate the third term to obtain the equation

$$J_0 = q\mu_e n_{e0} E_x - \frac{\partial(\varepsilon E_x)}{\partial x}\mu_e E_x - D_e\frac{\partial^2(\varepsilon E_x)}{\partial x^2} \quad \bullet$$

The second term is obviously non-linear, but we shall see that FlexPDE is able to solve this equation.

Field at a Metal-Vacuum Interface

The simplest application of the above PDE is to a metal film in a vacuum, exposed to a perpendicular electric field.

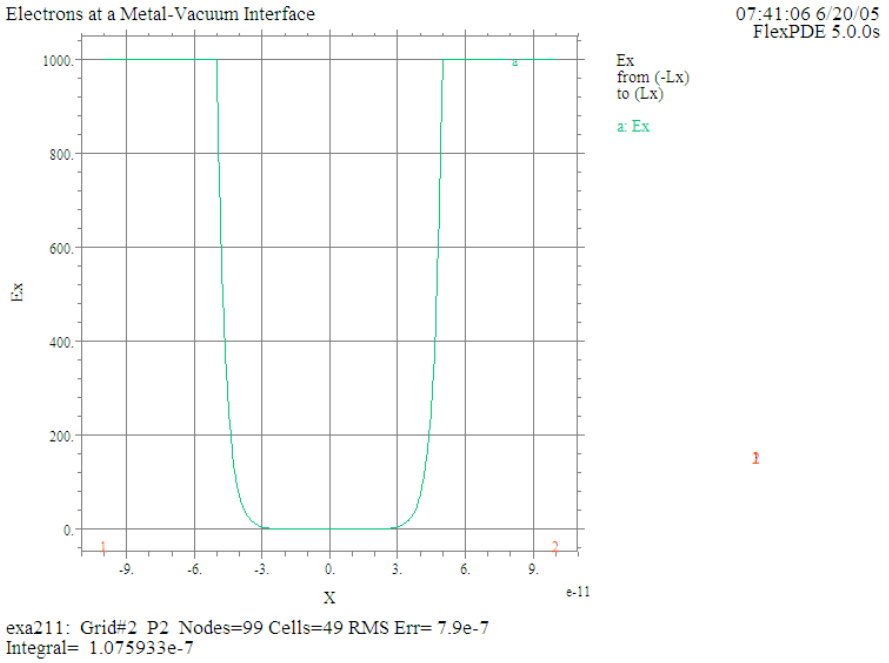
The current density in this situation is obviously zero. In the *boundaries* segment we specify a constant value of 1e3 for E_x at both ends of the domain.

The program cannot handle vanishing conductivity in the vacuum, and hence we must be content to specify values that are very small compared to those in the metal. The permittivity of the metal should be approximately equal to that of the vacuum (eps).

```
TITLE   'Electrons at a Metal-Vacuum Interface'          { exa211.pde }
SELECT     errlim=1e-4                                   { Student Version }
COORDINATES
  cartesian1
VARIABLES   Ex
DEFINITIONS
  Lx=1e-10     Lc=Lx/2
  J0=0     Ex0=1e3                                      { External Ex }
  q=1.60e-19     kb=1.38e-23     eps=8.85e-12     temp=300
  cond     ne0
  mue=cond/q/ne0     De=mue*kb*temp/q     rho=dx( eps*Ex)
  F_ex=0.5*eps*Ex0^2                                   { Force }
EQUATIONS
  q*mue*ne0*Ex- dx( eps*Ex)*mue*Ex- De*dxx( eps*Ex)=J0
BOUNDARIES
```

```
region 'domain'  cond=1.0  ne0=1.0                { Avoid zero in mue }
  start (-Lx)  point value( Ex)=Ex0
  line to (Lx) point value( Ex)=Ex0
region 'copper'  cond= 5e7  ne0=1.0e29   start(-Lc)
  mesh_spacing=0.03*Lx  line to (Lc)           { Denser at interface }
PLOTS
  elevation( ne0) from (-Lx) to (Lx)     elevation(Ex) from (-Lx) to (Lx)
  elevation( rho*Ex) from (0) to (Lc) report(F_ex)
END
```

As seen from the second plot (below), the field penetrates into the metal over a short distance and gradually fades away. The internal field remains comparable to the external Ex over a layer about 1e-11 m. The thickness of this layer is much smaller than the size of an atom, and the result reflects the fact that we are using a continuum model for the metal.



We may test this model by calculating the force per m^2 on the metal surface due to the external field. To achieve this, we must integrate ρE_x with respect to x over the metallic region. The integral value reported on the last plot may be compared with the expression $\mathbf{f} = \tfrac{1}{2} D_n \mathbf{E}$ that we used on p.108. Evidently, the agreement is excellent.

On changing the temperature to 1000 K you will find that the charged layer has become thicker. This is caused by a larger electron diffusion coefficient.

Semiconductors

In semiconductors we have to consider two types of charge carriers, i.e. electrons and holes[8p326] (positive). Electrons drift in a direction opposite to **E**, and holes in the direction of **E**. For both carrier types, diffusion is in the direction opposite to the corresponding density gradient. In the steady state, we may thus express the *flux densities* of electrons and holes as

$$\begin{cases} \mathbf{f}_e = -\mu_e n_e \mathbf{E} - D_e \nabla n_e \\ \mathbf{f}_h = \mu_h n_h \mathbf{E} - D_h \nabla n_h \end{cases} \quad \bullet$$

where n_e and n_h are the corresponding carrier densities, μ_e, μ_h the mobilities, and D_e, D_h the diffusion coefficients given by

$$D_e = \mu_e k_B T/q, \quad D_h = \mu_h k_B T/q \quad \bullet$$

Multiplying the above expressions by the charges $(\mp q)$ we immediately obtain the corresponding electric current densities[7p16].

$$\begin{cases} \mathbf{J}_e = q n_e \mu_e \mathbf{E} + q D_e \nabla n_e \\ \mathbf{J}_h = q n_h \mu_h \mathbf{E} - q D_h \nabla n_h \end{cases} \quad \bullet$$

For the total current **J** we thus obtain

$$\mathbf{J} = \mathbf{J}_e + \mathbf{J}_h = q n_e \mu_e \mathbf{E} + q D_e \nabla n_e + q n_h \mu_h \mathbf{E} - q D_h \nabla n_h \quad \bullet$$

The terms containing **E** express the drift of carriers in the field, which means the electrical conductivity may be written

$$\sigma = q n_e \mu_e + q n_h \mu_h \quad \bullet$$

If **J** is known inside the semiconductor and **E** on the boundary we might hope to solve the above PDE for **E**.

For one space variable (x) equation p.226●4 takes the form

$$q\,n_e\,\mu_e E_x + q\,D_e\frac{\partial n_e}{\partial x} + q\,n_h\,\mu_h E_x - qD_h\frac{\partial n_h}{\partial x} = J_0 \qquad \bullet$$

The PDE may seem to contain three dependent variables: E_x, n_e, and n_h. In the steady state, however, n_e and n_h will be related according to the statistical law of mass action[8p367]

$$n_e\,n_h = n_i^2 \qquad \bullet$$

where n_i is an intrinsic property of the material.

Under the influence of a field, the carriers move to new positions, creating the local charge density ρ.

$$\rho = \rho_e + \rho_h = (n_e - n_{e0})(-q) + (n_h - n_{h0})\,q = (-n_e + n_{e0} + n_h - n_{h0})\,q$$

Using the above statistical law we obtain

$$\rho/q = -n_e + n_{e0} + n_i^2\,/n_e - n_{h0}$$

Next, we express n_e in terms of ρ, which leads to the quadratic equation

$$n_e^2 + n_e(\rho/q - n_{e0} + n_{h0}) = n_i^2$$

which has the solution

$$n_e = -\frac{1}{2}(\rho/q - n_{e0} + n_{h0}) \pm \sqrt{\frac{1}{4}(\rho/q - n_{e0} + n_{h0})^2 + n_i^2} \qquad \bullet$$

For n_h we obtain a similar relation, i.e.

$$\rho/q = -n_i^2\,/n_h + n_{e0} + n_h - n_{h0}$$

and

$$n_h = -\frac{1}{2}(n_{e0} - n_{h0} - \rho/q) \pm \sqrt{\frac{1}{4}(n_{e0} - n_{h0} - \rho/q)^2 + n_i^2} \qquad \bullet$$

We finally express ρ in terms of E_x by means of the Maxwell equation

$$\rho = \nabla \cdot (\varepsilon E_x)$$ ●

Substituting this expression for ρ in n_e and n_h, and these in turn into the above PDE, we reduce the number of dependent variables in the above PDE to a single one, i.e. E_x. The PDE in its complete form would be monstrously complicated, but we can let FlexPDE make all substitutions internally.

It would seem that the governing PDE p.227●1 contains no 2nd-order derivative, but n_e is approximately equal to ρ/q and the last substitution introduces a second differentiation.

Semiconductor in an Electric Field

In a pure semiconductor, both electrons and holes contribute to the conduction of charge through the material. Doping by impurities[8p372] leads to modified equilibrium densities of free charge carriers, n_{h0} and n_{e0}, so that one type of material (p) may conduct mainly by positive holes and another type (n) by negative carriers. These carrier densities are constants of the p and n materials respectively. If no field is present, each of these carriers resides close to an ion of the opposite charge, so that the material remains electrically neutral.

Let us apply the above formalism to a semiconductor film, surrounded by a vacuum and exposed to an external field $\mathbf{E}_0$ perpendicular to its surface. The configuration is similar to that for the metal plate in *exa211*. As we have just seen, the governing PDE is

$$q\, n_e \mu_e E_x + q\, D_e \frac{\partial n_e}{\partial x} + q\, n_h \mu_h E_x - q D_h \frac{\partial n_h}{\partial x} = J_0$$ ●

We first consider the term J_0 appearing in the right member of the PDE. In the 1D model, charge conservation requires that it take the same value everywhere.

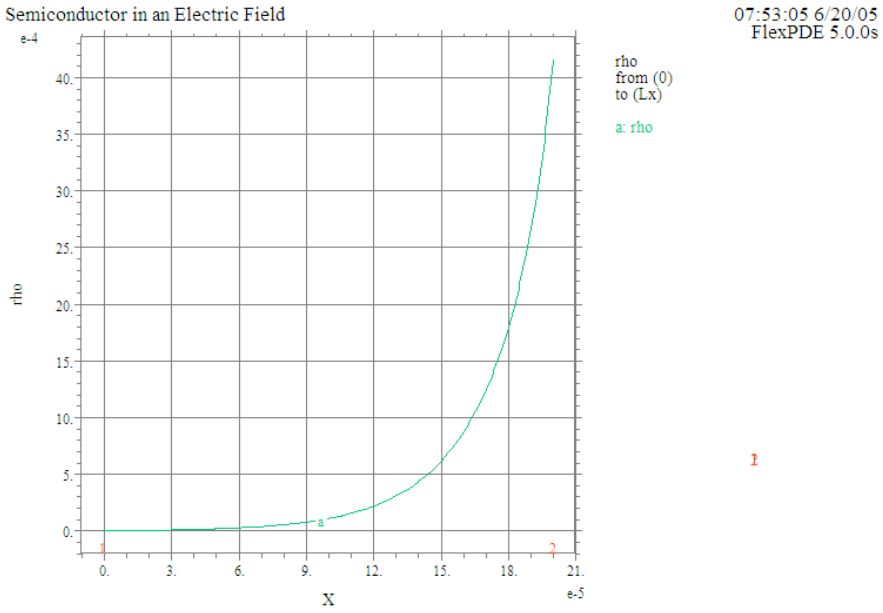
Under *definitions* we specify the equilibrium numbers of free electrons and holes for the semiconductor material. Using staged we specify a sequence of three values for nh0, and hence for ne0. The first value is for an electron-type material, the middle one for an intrinsic semiconductor, and the last one for a hole-type material.

The boundary conditions on the right and left faces of the film impose the value Ex0 just *inside* the surfaces. The outside of the faces is not part of the domain, but we could imagine a field E_v on the vacuum side, given by the equation of continuity $D_x = \varepsilon E_{x0} = \varepsilon_0 E_v$.

```
TITLE   'Semiconductor in an Electric Field'                    { exa212.pde }
SELECT     errlim=1e-12       { Three stages are indicated by staged }
COORDINATES
  cartesian1
VARIABLES   Ex
DEFINITIONS
  Lx=2e-4
  temp=300     J0=0     Ex0=1e3                  { Applied Ex }
  q=1.60e-19     kb=1.38e-23     eps0=8.85e-12     eps=12*eps0
  ni=1e16     mue=0.13     muh=0.05
  nh0=staged(0.1*ni, ni, 10*ni)     ne0=ni^2/nh0
  De=mue*kb*temp/q     Dh=muh*kb*temp/q
  rho=dx( eps*Ex)                                  { Volume charge density }
  ne=-0.5*( rho/q- ne0+ nh0)+ sqrt[ 0.25*( rho/q- ne0+ nh0)^2+ ni^2]
  nh=-0.5*( ne0- nh0- rho/q)+ sqrt[ 0.25*( ne0- nh0- rho/q)^2+ ni^2]
  Je=q*mue*ne*Ex     Jh=q*muh*nh*Ex  { Drift current density }
  Jde=q*De*dx( ne)     Jdh=-q*Dh*dx( nh)       { Diffusion current d. }
EQUATIONS
  q*ne*mue*Ex+ q*De*dx( ne)+ q*nh*muh*Ex- q*Dh*dx(nh)=J0
BOUNDARIES
region 'domain'
  start (-Lx)  point value(Ex)=Ex0 line to (Lx) point value(Ex)=Ex0
PLOTS
  elevation( Ex) from (-Lx) to (Lx) report( nh0) report( ne0)
  elevation( rho) from (-Lx) to (Lx)
  elevation( rho) from (0) to (Lx) report( eps*Ex0)     {  Charge density }
  elevation( nh) from (-Lx) to (Lx) report( 2*Lx*nh0)
  elevation( ne) from (-Lx) to (Lx) report( 2*Lx*ne0)
  elevation( Jh, Jdh) from (-Lx) to (Lx)
  elevation( Je, Jde) from (-Lx) to (Lx)
  elevation( nh*ne/ni^2) from (-Lx) to (Lx) range(0.999,1.001)
END
```

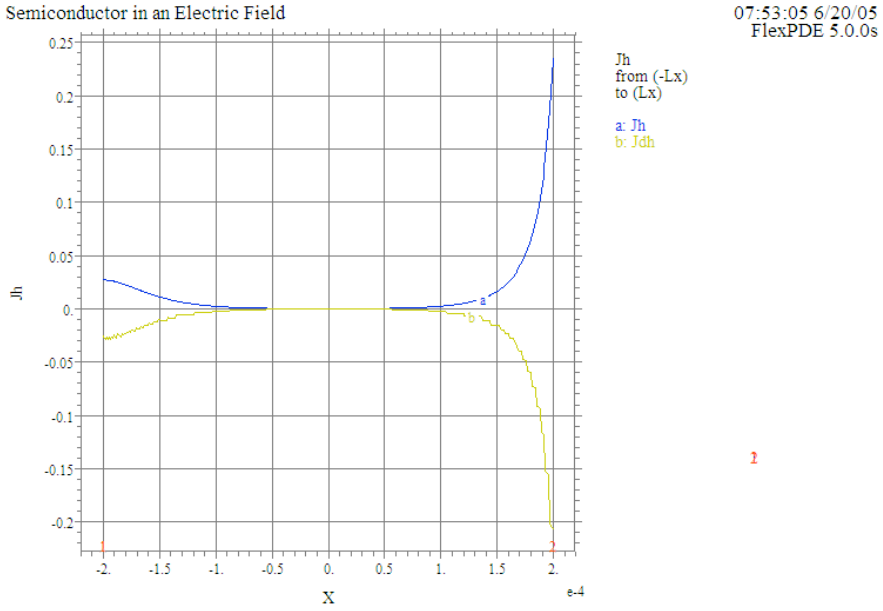
For the *second* stage we obtain the following plot of the charge density over the right half of the domain. The general shape of this curve resembles that for a metal, but the thickness of the charge distribution is much larger. If we consider this to be a surface charge,

the relation $D_x = \sigma$ should be valid. In fact, we notice that the integral of ρ over this layer is closely equal to the value of εE_{x0}, also displayed under the plot.



exa212: Grid#1 P2 Nodes=101 Cells=50 RMS Err= 8.e-14
Stage 2 eps*Ex0= 1.062000e-7 Integral= 1.059879e-7

Evidently, the charge density arises partly from holes (nh) and partly from electrons (ne). The figure below illustrates that the currents due to drift and diffusion cancel to produce vanishing net current density.



exa212: Grid#1 P2 Nodes=101 Cells=50 RMS Err= 8.e-14
Stage 2 Integral(a)= 5.389024e-6 Integral(b)= -5.327839e-6

Inspecting the plots of nh and ne, we notice that these curves become similar if we reverse the sign of x for one of the plots. The drift currents, Jh and Je, due to holes and electrons remain different, however, because of the difference in mobility.

Proceeding to the plot of nh it is worth noting that the integral is reasonably equal to the total number of holes at zero field (2*Lx*nh0). Finally, it is satisfactory that nh*ne turns out to be equal to ni^2, as we assumed when deriving the PDE.

Corresponding observations may be made as regards the first and third stages in these calculations.

Current through a Semiconductor

We shall now modify the preceding descriptor for the case of a film carrying a current. The PDE already contains the current density J0, but we need to apply the appropriate boundary conditions.

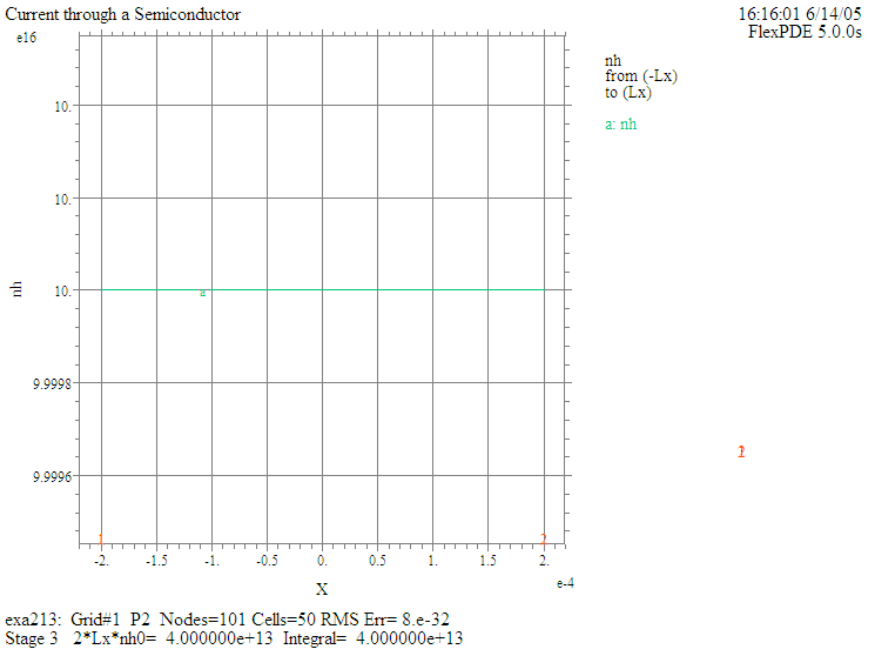
In a region where the carrier densities vary little with x we may eliminate the derivatives to obtain

$$J_0 = \sigma E_x = (q\, n_e\, \mu_e + q\, n_h\, \mu_h) E_x \qquad \bullet$$

Let us assume that the derivatives vanish and verify this only *a posteriori*. In *exa212* we used a numerical value for Ex0, but now we input $E_{x0} = J_0 / (q\, n_e\, \mu_e + q\, n_h\, \mu_h)$. The changes are as follows.

```
TITLE   'Current through a Semiconductor'                { exa213.pde }
…
  temp=300     J0=1.0                        { Defer definition of Ex0 }
…
  Ex0=J0/(q*ne*mue+q*nh*muh)
EQUATIONS
…
```

As is evident from the next plot, the results of the run are trivial. The hole density nh is now constant throughout the film, and there is no charge concentration at the surfaces.



The results of this example are thus rather predictable, but in the following section we shall see that a sandwich of different semiconductor films may exhibit much more complex properties.

Model of a Silicon Junction Diode

Let us now apply the above formalism to the simplest possible *p*-*n* diode, consisting of a film with predominately positive charge carriers (named *p*) and an adjacent film with negative carriers (*n*). We do not include the metal films normally used to connect this device to the external world. As we have just seen, the governing PDE is

$$q\,n_e\,\mu_e E_x + q\,D_e\frac{\partial n_e}{\partial x} + q\,n_h\,\mu_h E_x - qD_h\frac{\partial n_h}{\partial x} = J_0 \qquad \bullet$$

From the section before we know that J_0 must be constant in space in the steady state.

We may express J_0 in terms of $E_x = E_2$ by

$$J_0 = \sigma E_2 = (q\,n_{e2}\,\mu_e + q\,n_{h2}\,\mu_h)E_2 \qquad \bullet$$

which holds near the right face (2) of the *n* film, sufficiently far from the junction to make the diffusion terms negligible.

The value E_2 is to be supplied as a constant boundary value at the right face (2). We also need a boundary value for the left face (1) of the sandwich. From the corresponding relation

$$J_0 = (q\, n_{e1}\, \mu_e + q\, n_{h1}\, \mu_h) E_1 \qquad \bullet$$

we directly obtain the value E_1 for the left boundary.

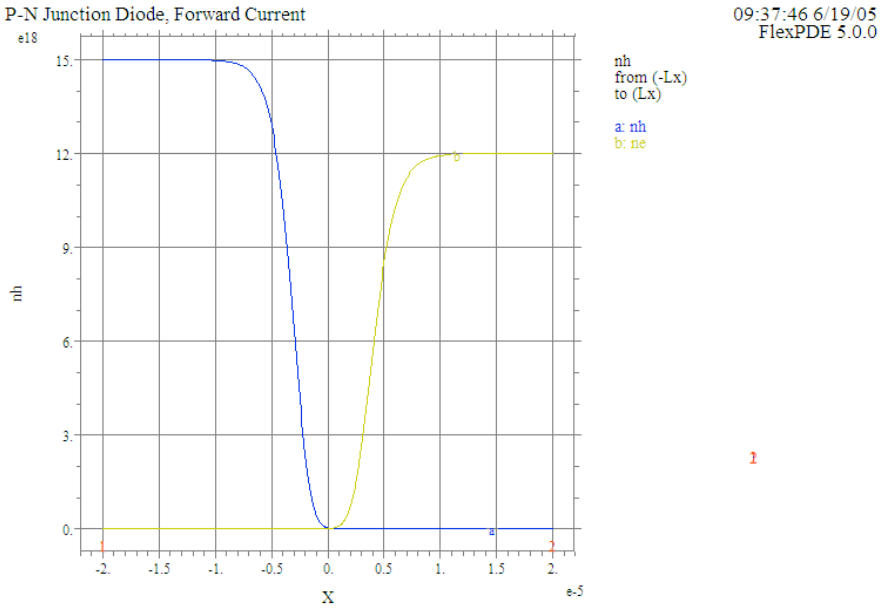
Under *definitions* we specify the equilibrium numbers of free electrons and holes, valid for the semiconductor materials before being joined into a sandwich. We choose clearly different values of carrier density, to emphasize the fact they do not have to be equal

Using stages we obtain solutions for a sequence of values of E2 at the right end of the domain. The relation to J0 permits us to calculate the corresponding field E1 at the left end. This is all we need for the boundary conditions.

```
TITLE   'P-N Junction Diode, Forward Current'              { exa214f.pde }
SELECT    errlim=1e-6    stages=10
COORDINATES                                        { Professional Version }
  cartesian1
VARIABLES   Ex
DEFINITIONS
  Lx=2e-5    temp=300
  q=1.60e-19    kb=1.38e-23    eps0=8.85e-12    eps=12*eps0
  ni=1.5e16    mue=0.13    muh=0.05    ne0    nh0
  E_res                                { Field due to resistive voltage drop }
  nh1=1000*ni    ne1=ni^2/nh1    ne2=800*ni    nh2=ni^2/ne2
  De=mue*kb*temp/q    Dh=muh*kb*temp/q
  rho=dx( eps*Ex)                      { Volume charge density }
  ne=-0.5*( rho/q- ne0+ nh0)+ sqrt[ 0.25*( rho/q- ne0+ nh0)^2+ ni^2]
  nh=-0.5*( ne0- nh0- rho/q)+ sqrt[ 0.25*( ne0- nh0- rho/q)^2+ ni^2]
  Je=q*mue*ne*Ex    Jh=q*muh*nh*Ex  { Drift current density }
  Jde=q*De*dx( ne)    Jdh= -q*Dh*dx(nh)      { Diffusion current d. }
  E2=(stage-1)*1e4    J0=(q*ne2*mue+q*nh2*muh)*E2
  E1=J0/(q*ne1*mue+q*nh1*muh)
  U_res=E1*Lx+ E2*Lx        { Approximation to resistive voltage drop }
  Ex_reduced=Ex- E_res      { Field with resistive part subtracted }
EQUATIONS
  q*mue*ne*Ex+ q* De*dx( ne)+ q*muh*nh*Ex- q*Dh*dx(nh)=J0
BOUNDARIES
region 'domain'  nh0=nh1   ne0=ne1              { Mostly holes }
  E_res=E1                             { Resistive field in the hole region }
```

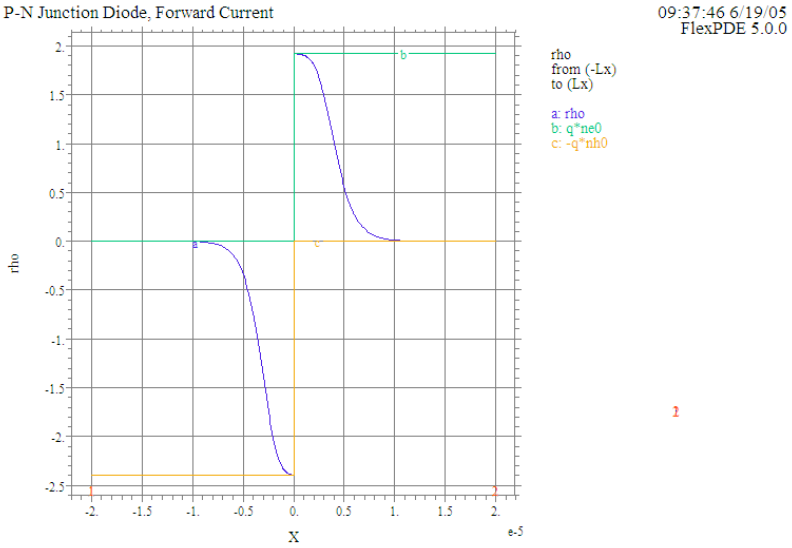
```
  start (-Lx) point value(Ex)=E1
  line to (Lx) point value(Ex)=E2
region 'n_conductor'  nh0=nh2  ne0=ne2   E_res=E2
  start (0) line to (Lx)
PLOTS
  elevation( Ex) from (-Lx) to (Lx)
  elevation(Ex_reduced) from (-Lx) to (Lx)
  elevation( rho, q*ne0, -q*nh0) from (-Lx) to (Lx)
  elevation( nh0, ne0) from (-Lx) to (Lx)
  elevation( nh, ne) from (-Lx) to (Lx)
  elevation( Jh, Je) from (-Lx) to (Lx)
  elevation( Jdh, Jde) from (-Lx) to (Lx)
  elevation( nh*ne) from (-Lx) to (Lx)      { Verification of equilibrium }
    report( J0)  report( U_res)
END
```

The plot below shows the equilibrium distribution of carriers near the junction. Here, no net current is flowing through the diode. In the preceding figure we find the values that would apply if the junction were broken (nh0, ne0).



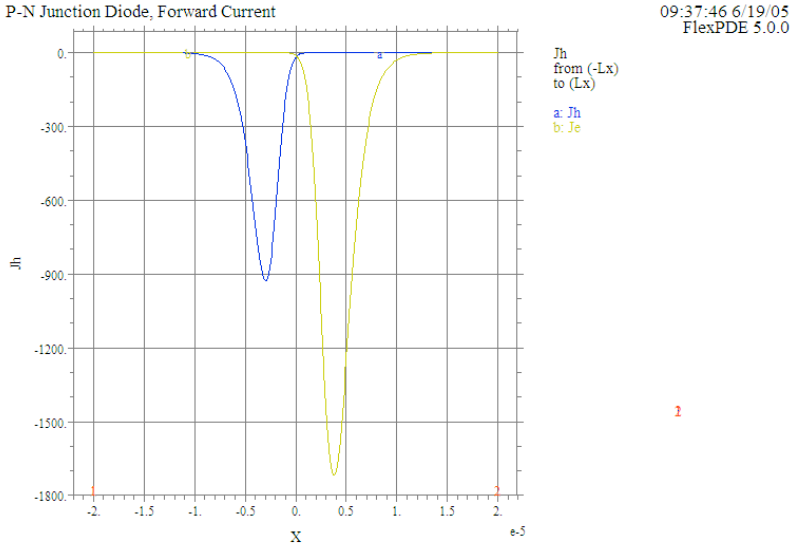
The following plot also pertains to the case of zero current. It shows that an electric dipole barrier is present at the junction. The charged layers arise by diffusion of holes into the right region (where the concentration of positive carriers is smaller), and by diffusion of

electrons in the opposite direction. We estimate the integral over the positive charge to be about 7e-6, to be compared with the integral over both branches of the curve (-1e-7) given below the plot. This confirms the expectation that the total charge vanishes.



exa214f: Grid#2 p2 Nodes=283 Cells=141 RMS Err= 1.e-11
Stage 1 Integral(a)= -1.078154e-7 Integral(b)= 3.835205e-5 Integral(c)= -4.806006e-5

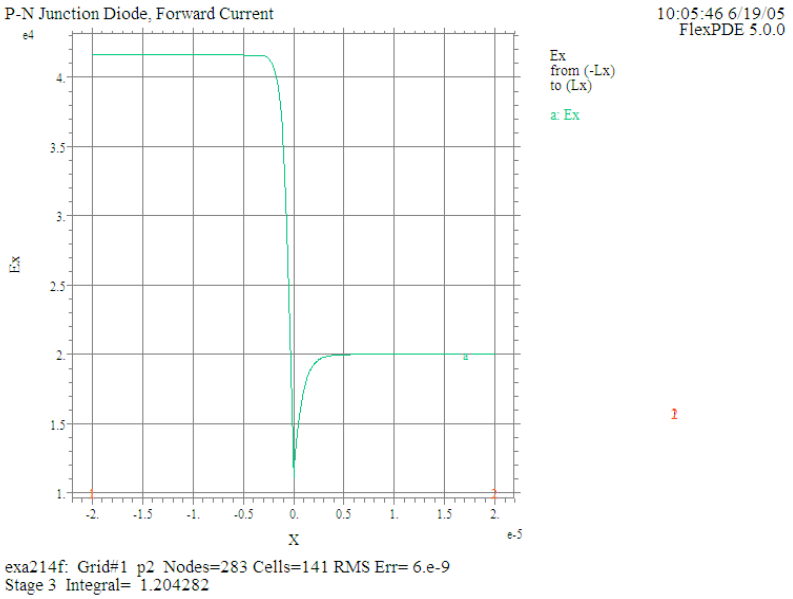
The charge distribution sets up an internal electric field, yielding the currents Jh and Je below, which limit the amount of diffusion.



exa214f: Grid#2 p2 Nodes=283 Cells=141 RMS Err= 1.e-11
Stage 1 Integral(a)= -3.104993e-3 Integral(b)= -6.457632e-3

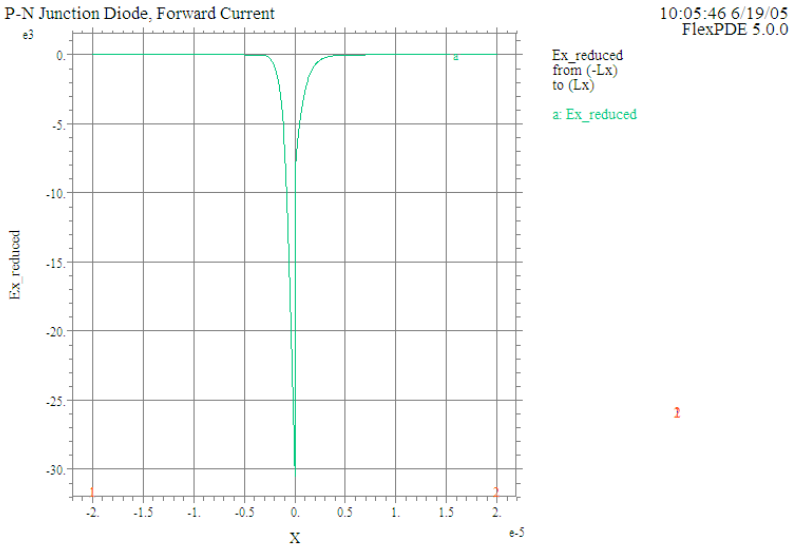
The plot of the diffusion currents Jdh and Jde demonstrates that drift and diffusion currents just cancel for each of the carrier types. The integral values confirm this observation quantitatively.

Later stages illustrate what happens when there is a net forward current through the diode. The next plot for stage 3 shows that Ex still is constant far from the junction, but takes different values at the ends. The external current flowing through the resistance of the *p* and *n* materials respectively produces these fields.



exa214f: Grid#1 p2 Nodes=283 Cells=141 RMS Err= 6.e-9
Stage 3 Integral= 1.204282

If we subtract the resistive fields (E_res) from Ex we are left with the negative cusp displayed below.



exa214f: Grid#1 p2 Nodes=283 Cells=141 RMS Err= 6.e-9
Stage 3 Integral= -0.028257

Evidently, there is an internal field pointing from right to left. The field at the junction creates an increase in potential over a short

region. We may calculate this *contact potential* as the integral of the reduced Ex (given at the bottom of the plot).

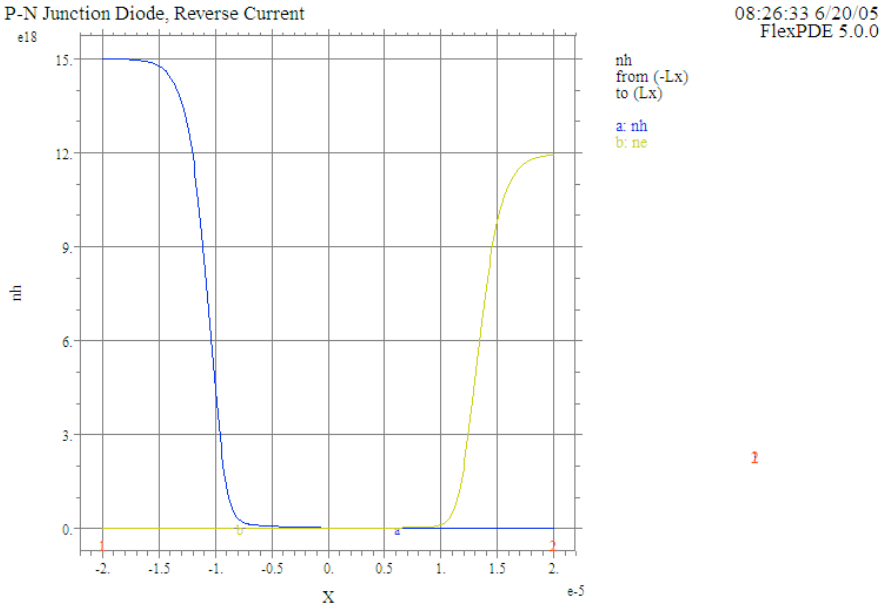
At zero current, the contact potential is about 0.35 V but becomes much smaller in the next stage of the calculations. Examining later stages we notice that the contact potential does nothing to reduce the forward current.

Reverse Current

It is easy to transform the above descriptor to model a *p-n* diode subject to reverse current. We only need to change the expression for E2 as follows.

```
TITLE   'P-N Junction Diode, Reverse Current'              { exa214r.pde }
…
  E2=-(stage-1)*40     J0=q*(ne2*mue+nh2*muh)*E2
…
```

The difference between the results is dramatic, as illustrated below. In the final stage of the calculation we find a wide depletion region, where virtually all the carriers are swept away from the junction at the center.

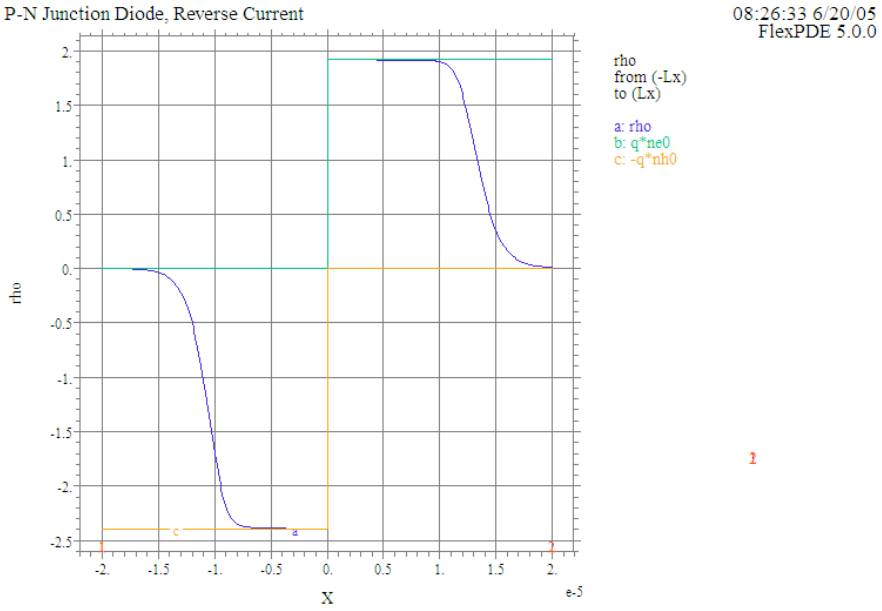


exa214r: Grid#1 p2 Nodes=283 Cells=141 RMS Err= 1.e-11
Stage 10 Integral(a)= 1.374299e+14 Integral(b)= 7.717311e+13

Looking at the following plot we find that the charge density reaches a limiting value as we approach the junction from each side.

On the right side of the junction there are no free electrons, and this means that the remaining charge there stems from the positive donor ions. The resulting charge density is thus $q\,n_{e0}$.

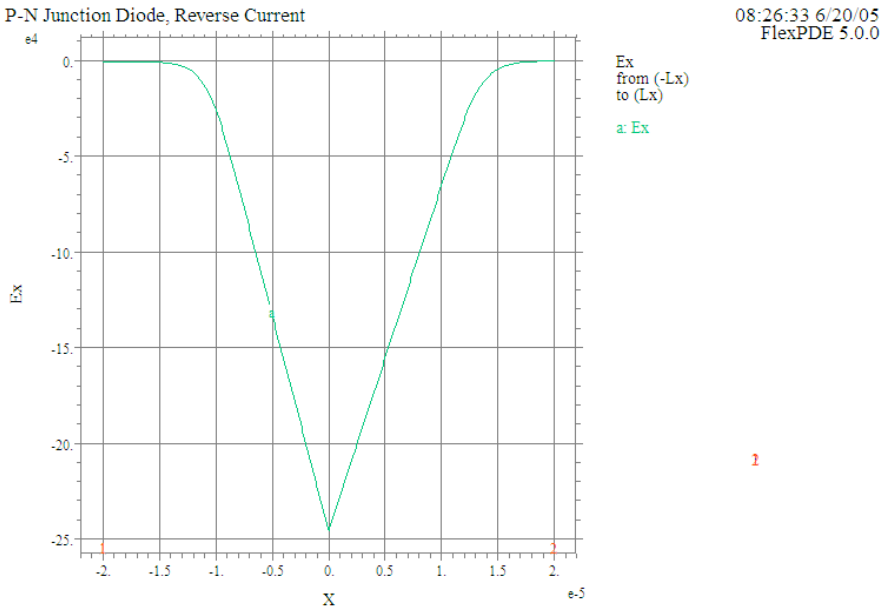
On the left side of the junction, all holes will be filled, which creates the charge density $-q\,n_{h0}$.



exa214r: Grid#1 p2 Nodes=283 Cells=141 RMS Err= 1.e-11
Stage 10 Integral(a)= -6.692545e-8 Integral(b)= 3.835205e-5 Integral(c)= -4.806006e-5

The next plot of Ex shows that the resistive voltage drop (near the ends) due to the reverse current is negligible compared to the internal potential difference, obtained by integrating the V-shaped part of the curve. Since the latter dominates, we conclude from the integral that the total reverse voltage is 3.08 V. In this case, the contact potential counteracts the applied voltage.

With this reverse voltage of 3.08 V, the current density is not zero but about 90 A/m^2, but for a junction diameter of, say, 10 μm this implies a reverse current of only 9 nA.



In order to explore the rectifying action of the diode, we should now compare to the results of *ex224f*. The integrals under the last plot of the descriptors indicate that, for an applied *forward* voltage over the junction of only 0.13 V, the current becomes J0=2.2e4. This is 244 times larger than the *reverse* current at -3.08 V.

Exercises

❑ Explore the effects of a reversed field Ex in *exa211*.
❑ Modify *exa211* to show the (trivial) phenomenon of conduction through a metal.
❑ Expand *exa211* by studying a sandwich made of an aluminum foil and a copper foil. Calculate the field for both directions of Ex.
❑ Replace the aluminum in the above sandwich by a superconductor, simulated by a conductivity of, say, one million times that of Cu.

22 Fields in Three Dimensions

So far, we have solved problems using one- and two-dimensional models, and in cases of rotational symmetry we were able to access three dimensions by resorting to cylindrical coordinates. FlexPDE permits us, however, to employ Cartesian coordinates in (x,y,z) space. The PDEs then become simpler than in (ρ,z), and we are no longer limited to axial symmetry. The run times will generally be longer, however.

Let us return to p.34, where we plotted the field around point charges. We shall now map fields in 3D, where the expression for the electrostatic potential of a point charge at (x_0,y_0,z_0) is

$$U = \frac{q}{4\pi\varepsilon_0 R} = \frac{q}{4\pi\varepsilon_0\sqrt{(x-x_0)^2+(y-y_0)^2+(z-z_0)^2}}$$ ●

Two Point Charges of Different Signs

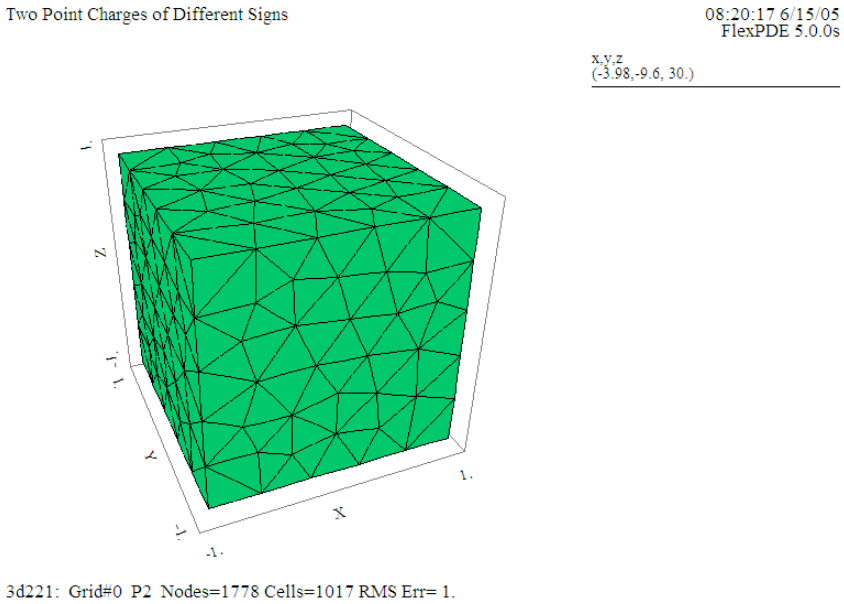
Under *definitions* in the following descriptor, we enter an expression for the potential U, consisting of two terms of the above type.

The new feature here is *extrusion*, which takes us into the third dimension. Under boundaries we define a base plane, which the program automatically divides into triangular cells. The program extrudes each of these cells into a cylinder of triangular cross-section, and each of these cylinders is finally filled with tetrahedra, up to a maximum level defined by the top surface.

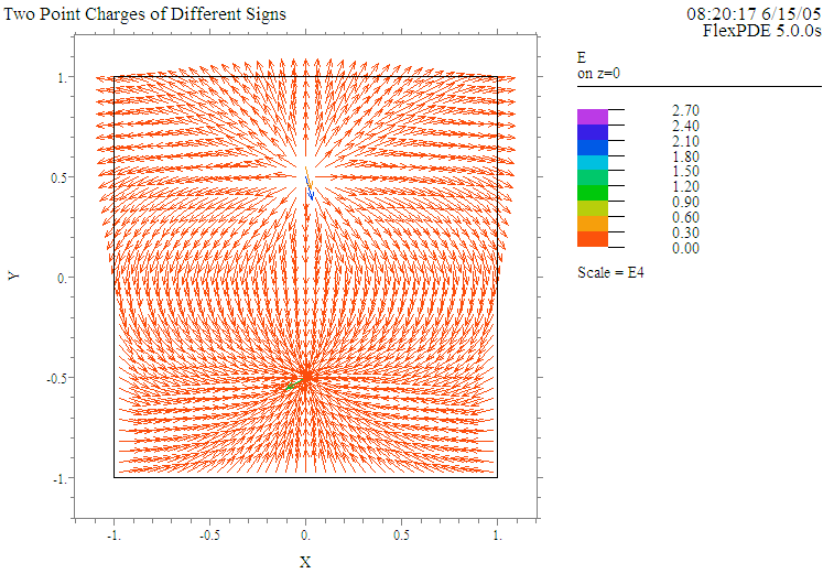
```
TITLE   'Two Point Charges of Different Signs'          { 3d221.pde }
SELECT     spectral_colors                        { Student Version }
COORDINATES     cartesian3
DEFINITIONS
  L=1.0     d0=0.5     q=1e-10
  eps0=8.85e-12     c=1/(4*pi*eps0)
```

```
  U=-q*c/sqrt( x^2+(y+d0)^2+ z^2)+ q*c/sqrt( x^2+(y-d0)^2+ z^2)
  Ex=-dx( U)    Ey=-dy( U)     Ez=-dz( U)       { Field components }
  div_xy=dx( Ex)+ dy( Ey)
  E=-grad(U)    Em=magnitude(E)
EXTRUSION                      { Extrude a cube through the (x,y) plane }
  surface 'bottom' z=-L        { Limiting surfaces }
  surface 'top' z=L
BOUNDARIES
region 'domain'
  start(-L,-L)                 { Trace outer boundary of base plane }
  line to (L,-L) to (L,L) to (-L,L) close
PLOTS
  grid( x,y,z)
  contour( U) painted on z=0     contour( U) painted on z=0.3
  contour( U) painted on z=1.0
  vector( E) norm on z=0     vector( E) norm on z=1.0
  elevation( Ex, Ey, Ez) from (0,-L,1) to (0,L,1)
  contour( Ez) painted on z=1.0     report( val( Ez, 0, 0.84, 1))
  contour( div_xy) on z=1.0     contour( div( E)) on z=1.0
END
```

The figure below shows the three-dimensional cell structure resulting from the extrusion from the bottom to the top face.

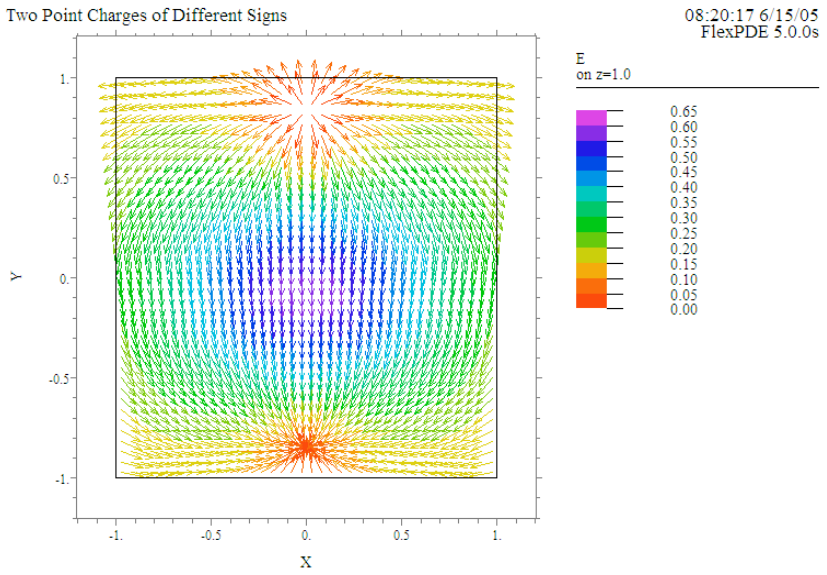


The other plots refer to a particular plane in space, which we must specify by a command such as on z=0. The vector plot below depicts the field directions in that plane, which contains the point charges.

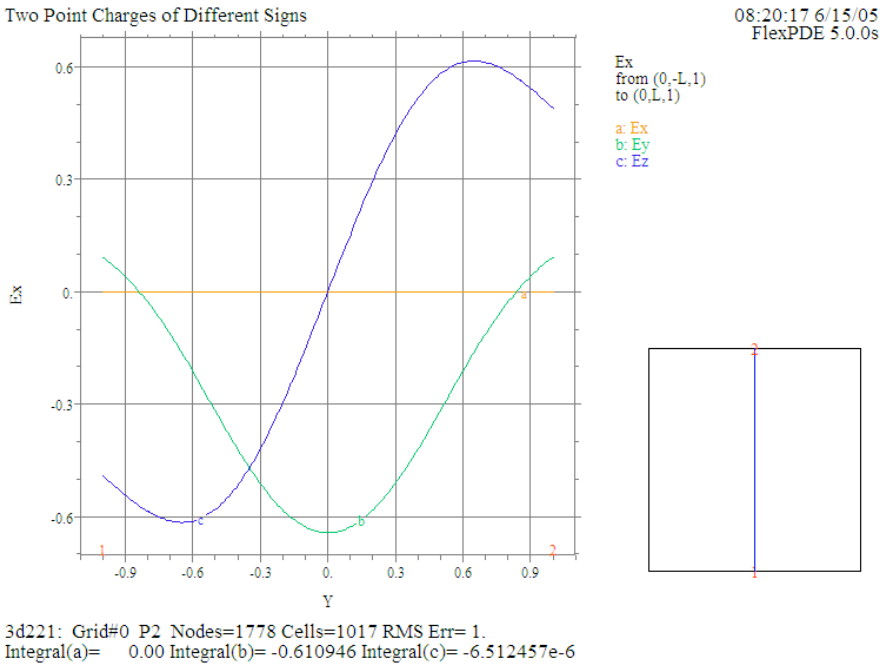


Here, all vectors appear to have the same magnitude, since the points of infinity obstruct the color-coding. The line through the charges at (0,-0.5,0) and (0,0.5,0), is an axis of symmetry. Hence, *any* vector plot on a plane going through $x = z = 0$ will look the same.

The following vector plot shows that the field diverges from a point at about $y = 0.84$, whereas the corresponding charge is located at $y = d_0 = 0.5$. In view of the axial symmetry, the above vector plot may also be taken as valid for the plane $x = 0$. We see that E_y vanishes at about $y = 0.84$ on the boundary, in agreement with the plot on $z = 1.0$.



From the colors and directions of the arrows it appears that the field vanishes at (0,0.84,1.0). This is only true of the in-plane components, however. The elevation plot below clarifies the variation of the field components along the vertical line $(0, y, 1.0)$. Obviously, Ex is zero everywhere, and Ey vanishes at about $y = 0.84$.



The contour plot of Ez also reports the value 0.57 at (0,0.84,1.0). From the last two plots we learn that the divergence in the (x, y) plane is non-zero (p.32), while the 3D divergence vanishes.

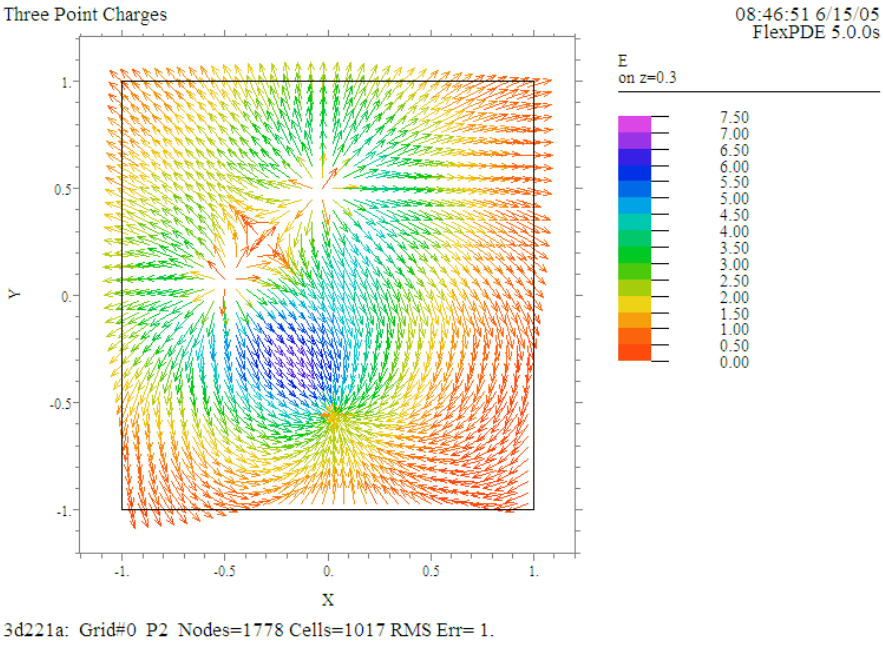
Non-Linear Set of Point Charges

The preceding charge configuration was axially symmetric. In the next example we shall consider three point charges that are not in line. It suffices to add a (positive) charge on the *x*-axis to destroy the axial symmetry.

```
TITLE   'Three Point Charges'                                { 3d221a.pde }
SELECT     spectral_colors
…
  U=-q*c/sqrt( x^2+(y+d0)^2+ z^2) +q*c/sqrt( x^2+(y-d0)^2+ z^2)
      +q*c/sqrt( (x+d0)^2+y^2+ z^2)
…
```

```
PLOTS
  contour( U) painted on z=0     contour( U) painted on z=0.3
  contour( U) painted on x=0     contour( U) painted on y=0
  contour( U) painted on z=1.0     contour( U) painted on y=x
  vector( E) norm on z=0     vector( E) norm on z=0.3
  contour( Em) painted on z=0.3
END
```

The following vector plot illustrates the field in a plane parallel to that of the charges.



Although a system of three point charges appears to be a simple one, the appearance of the 3D plots is far from trivial and it is instructive to analyze them. For instance, in the plot of U at $z = 0.3$ why are the extreme points different in magnitude? Why is the plot of U on the plane $y = x$ so simple?

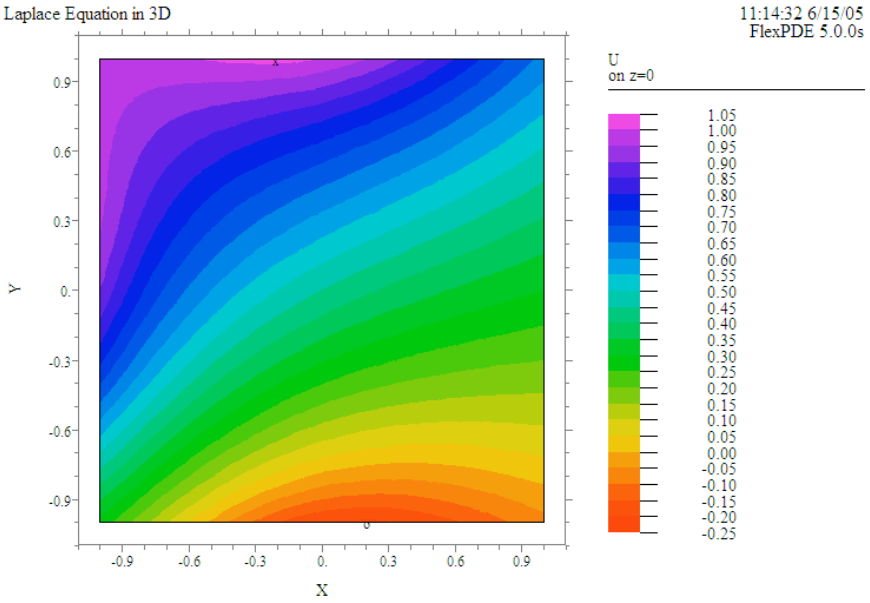
Laplace Equation in 3 D

We shall now revisit the example on p.49, where we treated the Laplace equation in 2D. In that case, it was easy to obtain an analytic solution for comparison. In 3D, this is less convenient but the sum of a set of potentials of the type p.240●1 would still be a solution. In the

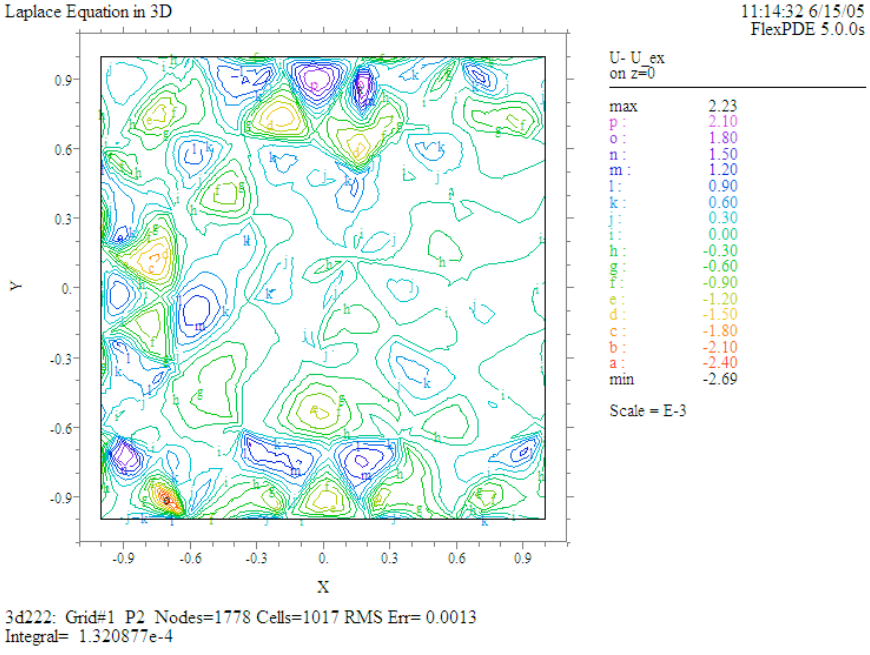
following descriptor we use *3d221a* as a template and place the point charges outside the domain by a suitable value of d0.

```
TITLE   'Laplace Equation in 3D'                                { 3d222.pde }
SELECT     errlim=1e-3     spectral_colors
COORDINATES     cartesian3
VARIABLES     U
DEFINITIONS
  L=1.0     d0=2.0     q=1e-10
  eps0=8.85e-12     c=1/(4*pi*eps0)
  U_ex=-q*c/sqrt( x^2+(y+d0)^2+ z^2)+ q*c/sqrt( x^2+(y-d0)^2+ z^2)
      +q*c/sqrt( (x+d0)^2+y^2+ z^2)                { Exact solution }
  Ex=-dx( U)     Ey=-dy( U)      Ez=-dz( U)        { Field components }
  div_xy=dx( Ex)+ dy( Ey)
  E=-grad(U)     Em=magnitude(E)
EQUATIONS
  div( grad( U))=0
EXTRUSION                                           { Limiting surfaces }
  surface 'bottom' z=-L
  surface 'top' z=L
BOUNDARIES
  surface 'bottom'  value( U)=U_ex
  surface 'top' value( U)=U_ex
region 'domain'
start(-L,-L)  value( U)=U_ex           { Outer boundary on base plane }
  line to (L,-L) to (L,L) to (-L,L) close
PLOTS
  grid(x,z) on y=0
  contour( U) painted on z=0     contour( U_ex) painted on z=0
  contour( U- U_ex) on z=0           { Deviation from exact U }
  contour( U- U_ex) on z=0 zoom(-0.5,-0.5,  1.0,1.0)
  contour( U) painted on z=0.3     contour( U_ex) painted on z=0.3
  contour( U) painted on z=1.0     contour( U_ex) painted on z=1.0
  contour( U- U_ex) on z=1.0
END
```

The following figure shows the solution in the plane of the point charges, of which there are some signs at the edges.



The next plot shows the solution error. Evidently, the deviation is about 0.3% of the maximum solution value.



The first zoomed plot demonstrates that the agreement in the interior of the domain is about 0.1%. The Professional Version with a larger number of nodes of course yields results of much higher quality.

Coin in a Metal Box

This problem is analogous to the elementary electrostatic example on p.57. We assume that a coin is suspended in the middle of a metallic box by thin nylon threads.

In previous examples, the solution domain was a simple cube. Here, we shall have to exclude a volume corresponding to the metallic coin, and we do this in three steps.

Firstly, we divide the cube into three *layers* by *surfaces* parallel to $z = -L$: the middle layer for the coin, the upper and lower ones for air.

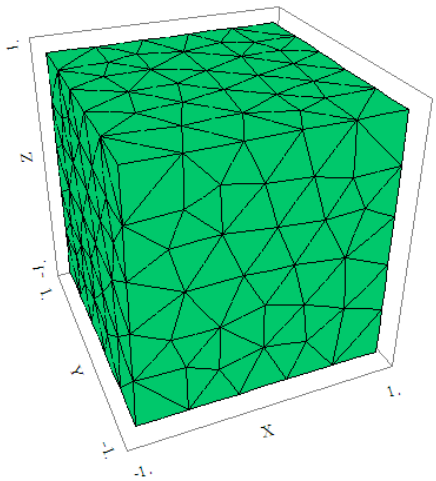
Secondly, we create a special *region* for the coin and specify the value of U on the flat surfaces. The 3D version of FlexPDE lets us exclude this region by the command void.

Finally, we define the circular shape as in 2D, also declaring the boundary value on the border to be U_coin. The cylinder defined by the line start…finish extends all the way from bottom to top of the box, and hence we have to add layer 'metal' to restrict the validity of the boundary value to the appropriate layer.

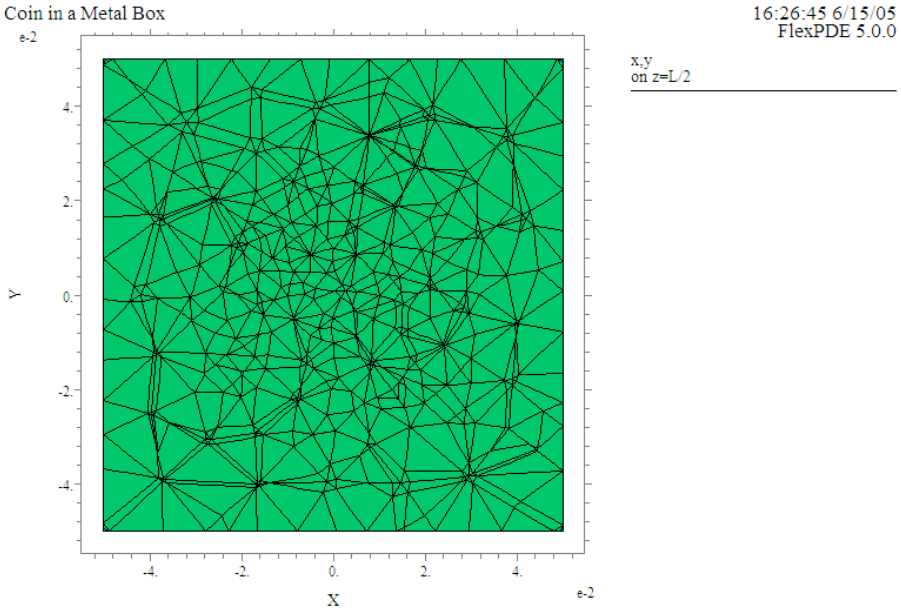
```
TITLE   'Coin in a Metal Box'                              { 3d223.pde }
SELECT    errlim=1e-3    ngrid=1    spectral_colors
COORDINATES    cartesian3                    { Professional Version }
VARIABLES    U
DEFINITIONS
  L=0.05       r0=0.01    d0=2e-3    U0=1.0
  Ex=-dx( U)    Ey=-dy( U)    Ez=-dz( U)
  E=-grad( U)    Em=magnitude( E)
EQUATIONS
  div( grad( U))=0
EXTRUSION                                          { Parallel surfaces }
  surface 'bottom' z=-L
  layer 'air 1'                                    { Layer below coin }
  surface 'lower' z=-d0
  layer 'metal'                                    { Layer containing coin }
  surface 'upper' z=d0
  layer 'air 2'                                    { Layer above coin }
  surface 'top' z=L
BOUNDARIES
  surface 'bottom'  value( U)=0
  surface 'top'  value( U)=0
```

```
region 'domain'                                    { Full solution domain }
  start (-L,-L)  value( U)=0                       { For all four faces }
  line to (L,-L)  to (L,L)  to (-L,L)  close
region 'coin'                                      { Exclude volume }
  surface 'lower' value( U)=U0
  surface 'upper' value( U)=U0
  layer 'metal' void
  start (r0,0) layer 'metal' value( U)=U0          { Coin }
  arc( center=0,0) angle=360
PLOTS
  grid(x,y,z)     grid(x,y) on z=L/2
  contour( U) painted on z=0     contour( U) painted on z=2*d0
  contour( U) painted on z=0.5*L     contour( U) painted on x=0
  vector( E) norm on z=0.5*L
  contour( Em) painted on x=0
  vector( E) norm on x=0
  elevation( U) from (-L,0,d0) to (L,0,d0)
END
```

FlexPDE shows all the details of *layers* and *regions* before the solutions starts. Clicking on *Domain* and then *Continue* also displays successive surfaces and layers from the bottom upwards. Finally we obtain the full 3D domain below.

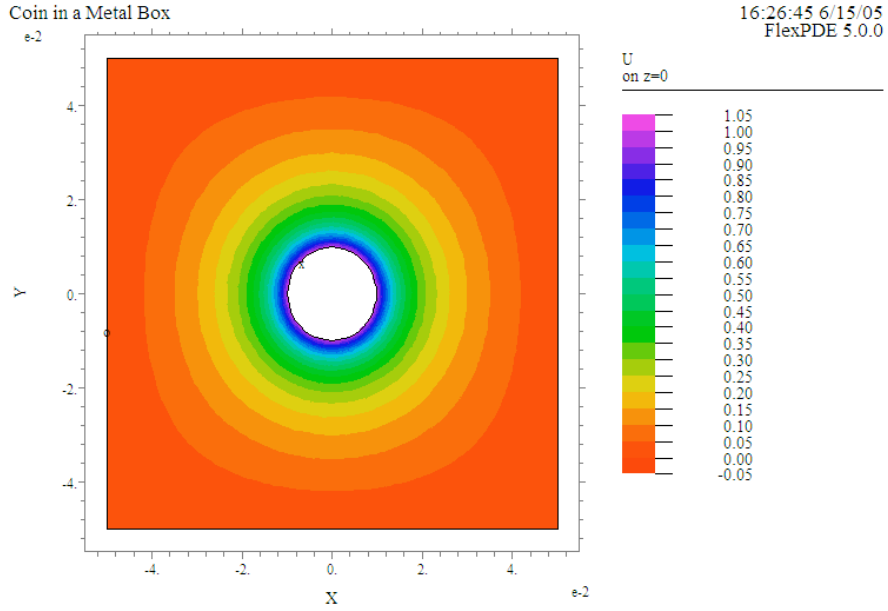


The 2D grid plot below looks quite complicated. It shows the intersection of the plot plane (z=L/2) with the tetrahedral 3D cells. The cell density is higher at the center, due to the extension of region 'coin'.



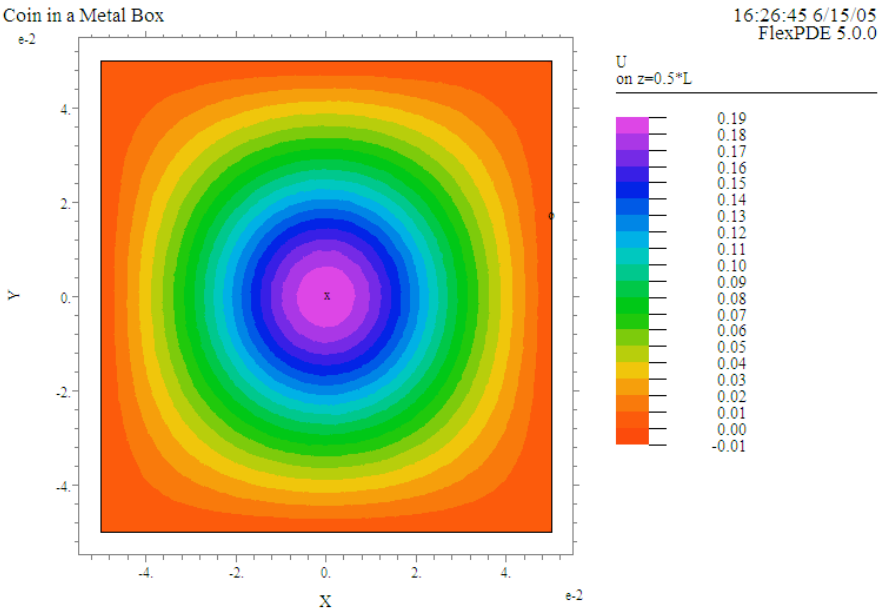
3d223: Grid#3 p2 Nodes=26074 Cells=18357 RMS Err= 4.8e-4

The following plot shows the potential around the coin in the middle plane $(z = 0)$.



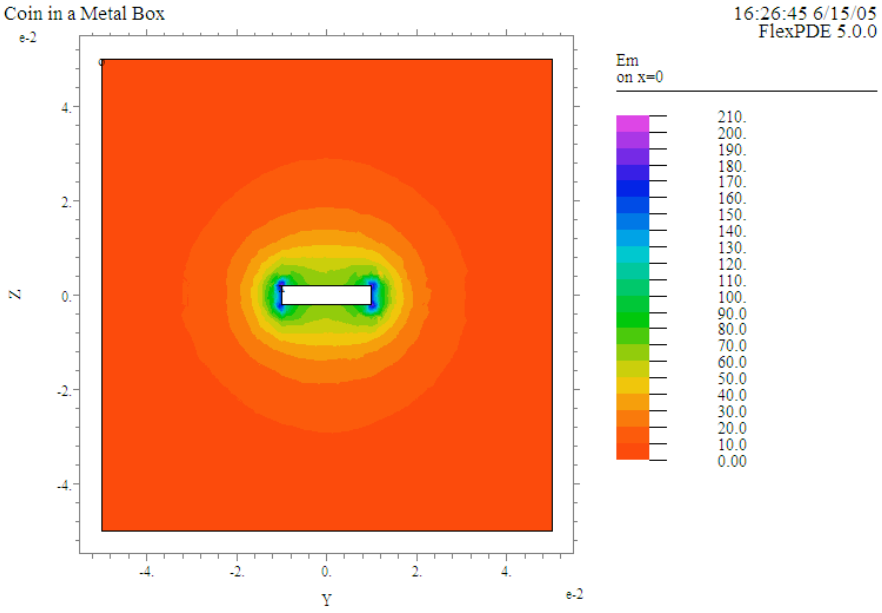
3d223: Grid#3 p2 Nodes=26074 Cells=18357 RMS Err= 4.8e-4
Integral= 1.217857e-3

The next plot is taken farther from the coin, halfway to the wall. We notice that the maximum value of the potential is only 20% of that in the middle plane.



3d223: Grid#3 p2 Nodes=26074 Cells=18357 RMS Err= 4.8e-4
Integral= 5.564608e-4

The next plot shows another projection of the coin and indicates the magnitude of the field in a symmetry plane.



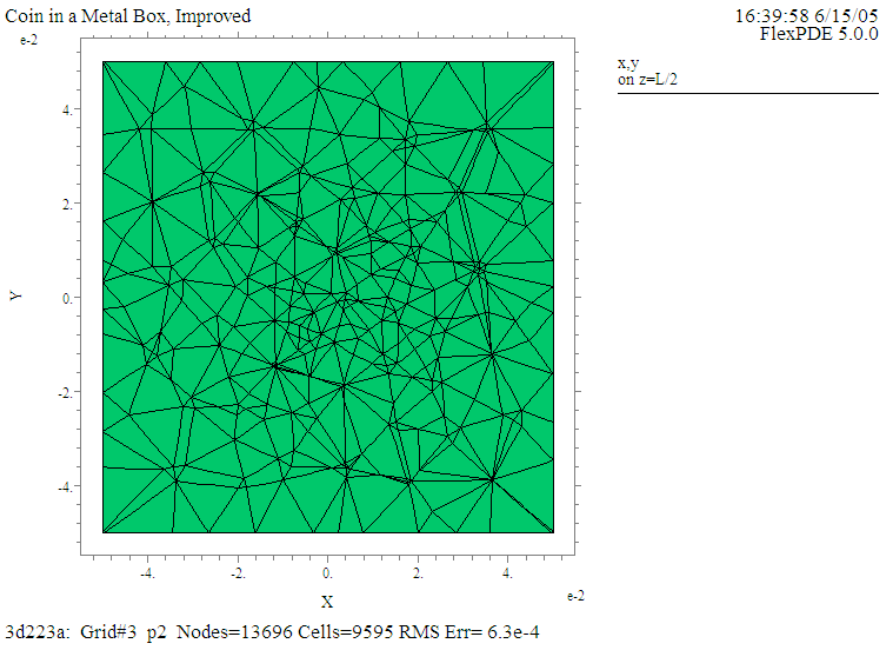
3d223: Grid#3 p2 Nodes=26074 Cells=18357 RMS Err= 4.8e-4
Integral= 0.111908

Improved Gridding around the Coin

In the preceding example, the grid extrusion implies that the node density remains high in region 'coin' all the way from bottom to top. This is a waste of nodes and of run time, because the solution is not expected to vary much far below and above the coin. As shown below, we may simplify the grid by using a **limited** region 'coin'.

```
TITLE   'Coin in a Metal Box, Improved'                    { 3d223a.pde }
…
  line to (L,-L)  to (L,L)  to (-L,L)  close
limited region 'coin'                                      { Exclude volume }
  surface 'lower' value( U)=U0
…
```

The next plot shows that the cell density is no longer enhanced around the central region. As a result, the total node number is reduced by nearly one half.



Electrical Conduction in a Cone

On p.91 we studied the current field in a cone with a voltage applied between the flat ends. Let us now repeat that example in 3D, using the

same shape, etceteras, so that the results may be compared. The first parts of the descriptor look somewhat like that on p.65.

This example is simpler than *3d223*, in the sense that there is only one layer. On the other hand, it may be confusing that the domain is limited from below by a broken surface. That surface consists of a flat, circular part of radius r0 and a conical part extending to meet the upper, flat surface.

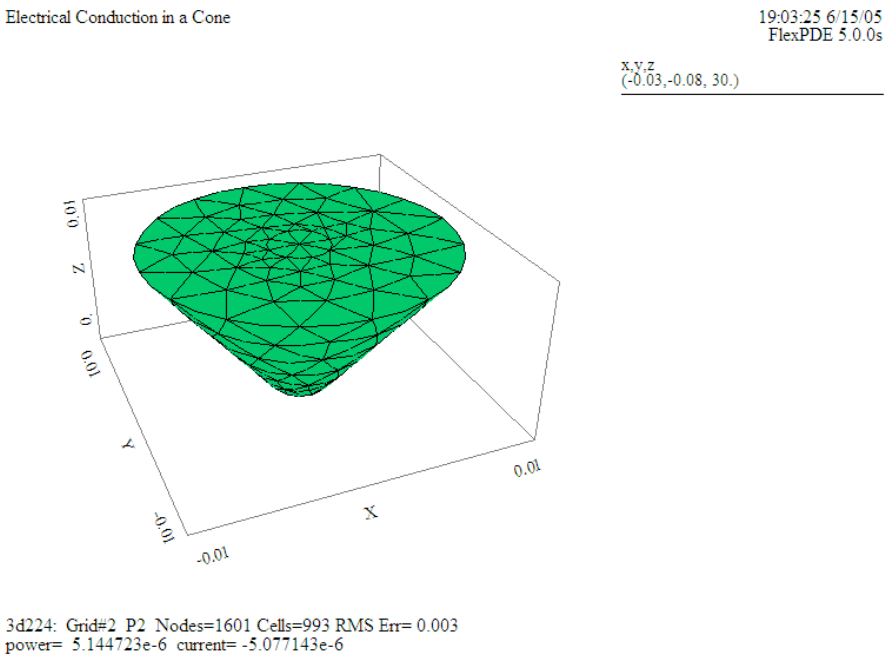
Under *extrusion* we first define the conical part as a function $z(x,y)$. FlexPDE permits us to include the flat part by using the function max(a,b). By this simple means we truncate the cone by stating that $z(x,y) \geq 0$. We could also have used the logical if…then. The concept of *extrusion* in FlexPDE means that this surface, having the lowest z, is extruded into a volume toward higher z. The line start(r1,0)…finish defines the size of the object, as projected on the (x,y) plane.

We wish to impose *value* boundary conditions on the flat parts and a *natural* boundary condition on the conical part. For this reason only, we define a central region in the shape of a cylinder with the radius r0. The remaining region is a ring limited by a circular cylinder and a conical surface. Surface 'lower' thus is conical in one region and flat in the other. By referring to the names of both the surface *and* the region we may also plot over a selected part.

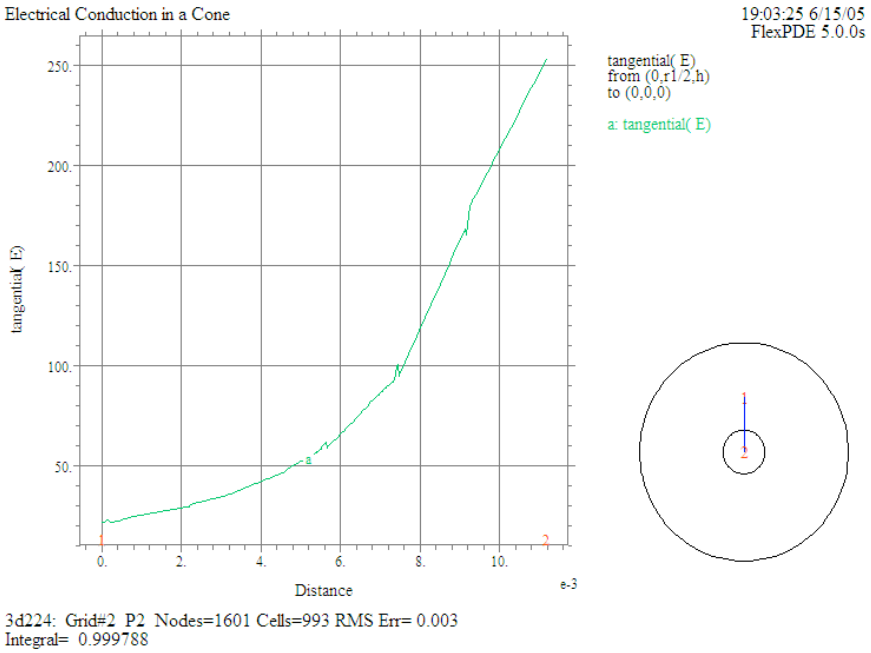
```
TITLE   'Electrical Conduction in a Cone'                    { 3d224.pde }
SELECT    errlim=1e-3    spectral_colors              { Student Version }
COORDINATES    cartesian3
VARIABLES   U
DEFINITIONS
  r0=2e-3    r1=10e-3    h=10e-3    cond=1.0e-3    { Silicon }
  Ex=-dx( U)    Ey=-dy( U)    Ez=-dz( U)
  E=-grad( U)    Em=magnitude( E)
  Jx=cond*Ex    Jy=cond*Ey    Jz=cond*Ez
  J=cond*E    Jm=magnitude( J)
  rad=sqrt( x^2+ y^2)                              { Radius }
  power=vol_integral( Jm*Em)                       { Dissipation }
  current=surf_integral( normal( J),'top')
EQUATIONS
  div( J)=0
EXTRUSION
  surface 'lower' z=max( 0, h*(rad-r0)/(r1-r0)) { Flat base plus cone }
```

```
  surface 'top' z=h                                  { Flat upper surface }
BOUNDARIES
  surface 'top' value( U)=1.0                        { 1 volt applied on top }
region 'conical ring'                                { Total domain }
  surface 'lower' natural( U)=0                      { dU/dn=0 on cone }
  start (r1, 0)  arc( center=0,0) angle=360          { Extension in (x,y) }
region 'cylinder'                                    { Overwrites total }
  surface 'lower' value( U)=0                        { Lower flat face }
  start (r0, 0)  arc( center=0,0) angle=360
PLOTS
  grid( x,y,z) report(power) report(current)
  contour( U) on x=0     vector( J) norm on x=0
  elevation( tangential( E))
     line_integrate from (0,r1/2,h) to (0,0,0)       { Voltage }
END
```

The plot below illustrates the geometry and the outside mesh. It also reports values of the power dissipated and the current. Since the voltage is 1.0, we expect these values to be equal.



For an elevation plot we have to supply an end point of a straight line by three coordinates, but in other respects such a plot is similar to what we have used in 2D. The following figure shows the tangential component of **E** along an arbitrary line from the top face to the bottom. The integral value confirms the voltage applied.



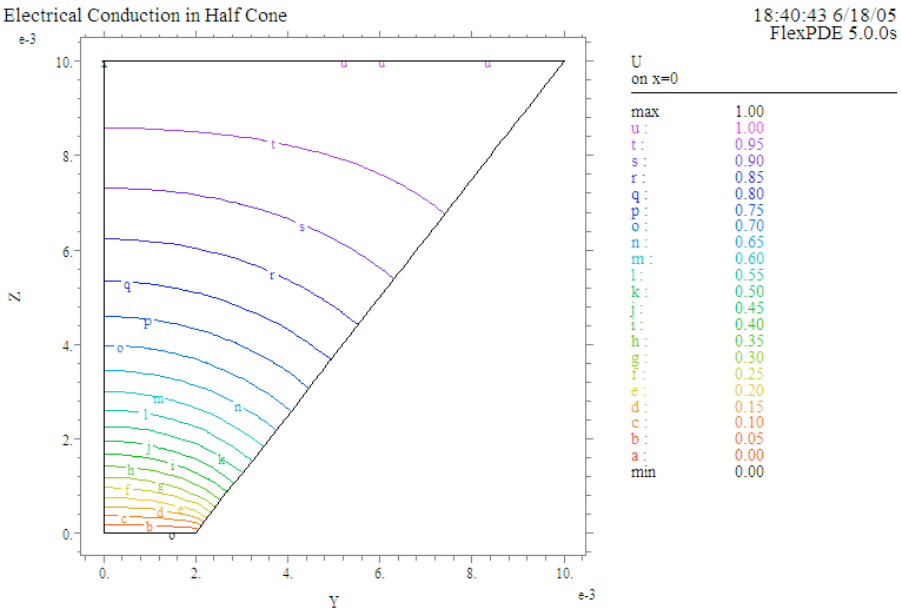
Exploiting Symmetry

The plots on x=0 from the above file are not identical to those from *exa101*, since cylindrical coordinates involve a smaller domain. We may, however, exploit the mirror symmetry of the 3D problem to solve over only half of the volume of the cone. Many problems have mirror symmetry without having axial symmetry, and in those cases we may reduce the run time considerably by this means. Alternatively, we may reduce the error without increasing the run time.

```
TITLE   'Electrical Conduction in Half Cone'                { 3d224a.pde }
…
  start (r1,0)  arc( center=0,0) angle=180 to (-r1,0) line to close
region 'cylinder'                                          { Overwrites total }
  surface 'lower' value( U)=0                              { Lower flat face }
  start (r0,0)   arc( center=0,0) angle=180 to (-r0,0)
…
  contour( 2*pi*y* U) on x=0                  { Integrate over the volume }
END
```

The contour plot below may be directly compared to that on p.91. The *integral values* are completely different, however, because the

volume element factor $2\pi r$ is automatically included when using cylindrical coordinates.



The last plot, which explicitly includes the factor $2\pi r$, yields the integral value 1.155e-6, close enough to the result from *exa101*.

Glass Block in a Parallel Electric Field

This example is analogous to *exa093*, part of which may serve as a template. The 3D elements are somewhat similar to those in *3d223*.

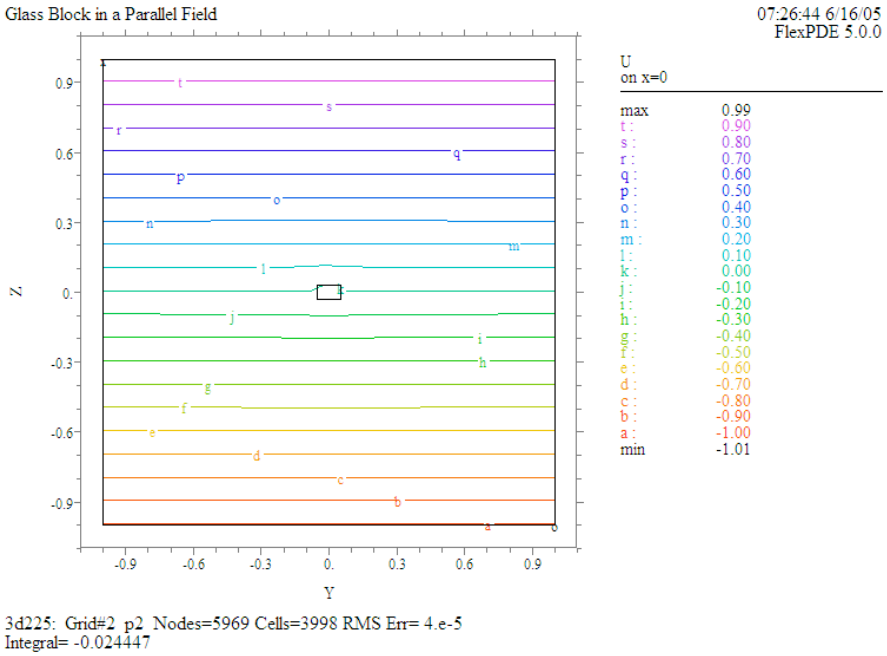
```
TITLE   'Glass Block in a Parallel Field'                    { 3d225.pde }
SELECT    errlim=1e-4    ngrid=1    spectral_colors
COORDINATES   cartesian3                          { Professional Version }
VARIABLES   U
DEFINITIONS
  L=1.0    Lx=0.1    Ly=0.05    Lz=0.03
  eps0=8.854e-12
  epsr    eps=epsr*eps0
  E0=1.0    De0=eps0*E0                           { Far field }
  E=-grad(U)    Em=magnitude(E)    D=eps*E    Dm=magnitude(D)
EQUATIONS
  div( D)=0
EXTRUSION                                         { Parallel surfaces }
```

```
  surface 'bottom' z=-L
  layer 'air 1'                                    { Layer below block }
  surface 'lower' z=-Lz                            { Lower face of block }
  layer 'glass'                                    { Containing glass }
  surface 'upper' z=Lz                             { Upper face of block }
  layer 'air 2'                                    { Layer above block }
  surface 'top' z=L
BOUNDARIES
  surface 'bottom'  natural( U)=De0                { Since D=-eps*grad(U) }
  surface 'top'  natural( U)=-De0
region 'domain'                                    { Full solution domain }
  layer 'air 1'   epsr=1
  layer 'glass'   epsr=1
  layer 'air 2'   epsr=1
  start (-L,-L)   natural( U)=0   line to (L,-L)  to (L,L)  to (-L,L)  close
region 'block'                                     { Redefine space}
  layer 'air 1'   epsr=1
  layer 'glass'   epsr=7.0                         { Glass block }
  layer 'air 2'   epsr=1
  start (-Lx,-Ly) line to (Lx,-Ly) to (Lx,Ly) to (-Lx,Ly) close
PLOTS
  grid( x, y, z)
  contour( U) on x=0     contour( U) on y=0
  vector( E) norm on x=0
  vector( E) norm on x=0 zoom(-0.2,-0.2,  0.4,0.4)
  vector( D) norm on x=0 zoom(-0.2,-0.2,  0.4,0.4) report(De0)
END
```

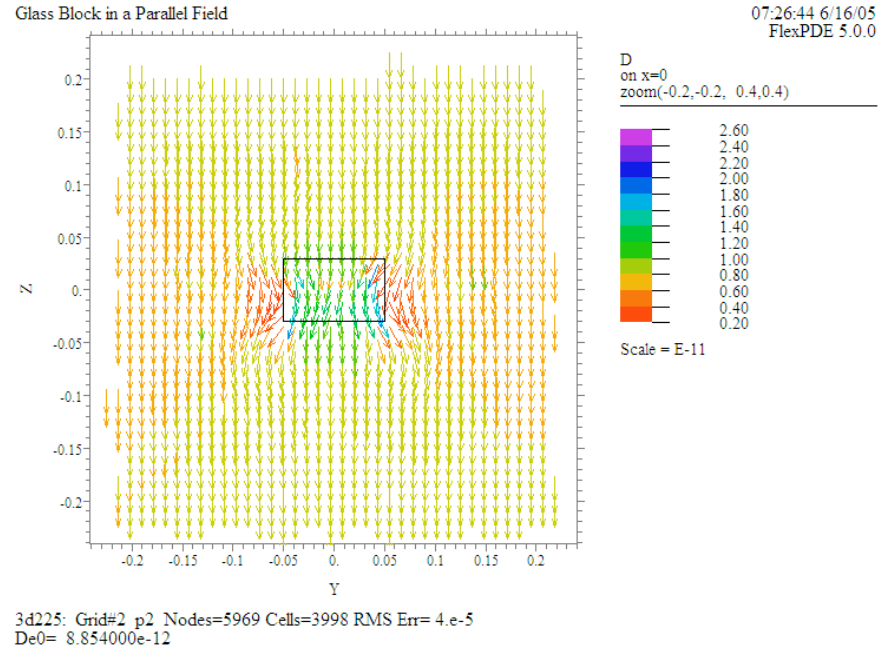
We first specify *natural* boundary conditions on the bottom and top surfaces by values of the applied normal component of **D**.

It is essential to understand how FlexPDE manages *regions* and *layers*. The region 'domain' has three layers, or *compartments*, all originally filled with air. Then we create the region 'block', which is an extrusion of a rectangle from z=-L to z=L. This region thus overlaps the domain first created, and there are hence six compartments, which could contain materials with different properties, in this case given by values of epsr. The volume common to the region 'block' and the layer 'glass' is a smaller compartment that contains only glass.

The following figure shows the geometry in one projection.



The next plot illustrates the distortion of the field near the glass block. The minimum value of Dm is only about 20% of the field imposed at the boundaries.



The following descriptor fraction suggests a simpler way of specifying epsr for the various compartments. In the 2D examples we found it expedient to assign properties to each of the regions, but in 3D there may be a large number of them. The above problem has one

compartment only with epsr>1. Thus it seems natural to predefine epsr=1 everywhere, and only change that default for the glass block.

```
TITLE   'Glass Block in a Parallel Field, Simplified'          { 3d225a.pde }
…
  epsr=1.0     eps=epsr*eps0
…
region 'domain'                                                { Full solution domain }
  start (-L,-L)   natural( U)=0   line to (L,-L)  to (L,L)  to (-L,L)  close
limited region 'block'                                         { Redefine space}
  layer 'glass'   epsr=7.0                                     { Glass block }
  start (-Lx,-Ly) line to (Lx,-Ly) to (Lx,Ly) to (-Lx,Ly) close
PLOTS
…
```

We also use limited to remove unnecessary cells above and below the glass block. This reduces the run time.

Simple Magnet Coil

The next example corresponds to that on p.145, but the PDEs and the expression **B** may be taken from p.124. The coil is parallel to the (x,y) plane, which implies that the current density Jz will vanish. We assume that the coil is wound by thin wire, so that the current density is essentially uniform over the cross-section.

On p.125 found the following PDE for the 2D geometry.

$$\frac{\partial}{\partial x}\left(\frac{1}{\mu}\frac{\partial A_z}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{\mu}\frac{\partial A_z}{\partial y}\right)=-J_z$$

Straightforward generalization to 3D gives us

$$\frac{\partial}{\partial x}\left(\frac{1}{\mu}\frac{\partial A_z}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{\mu}\frac{\partial A_z}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{1}{\mu}\frac{\partial A_z}{\partial z}\right)=-J_z$$

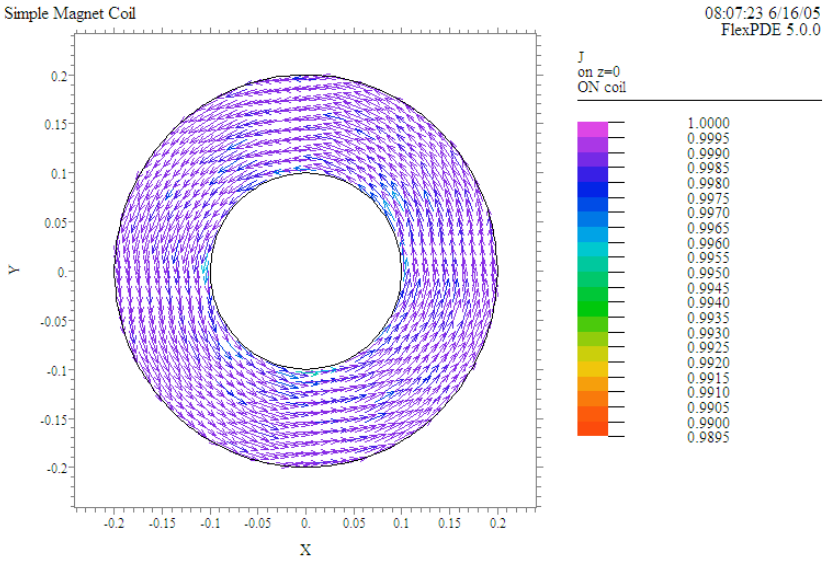
and there are of course corresponding equations for the remaining components of **J**.

The run time for this example will be longer than for the preceding ones.

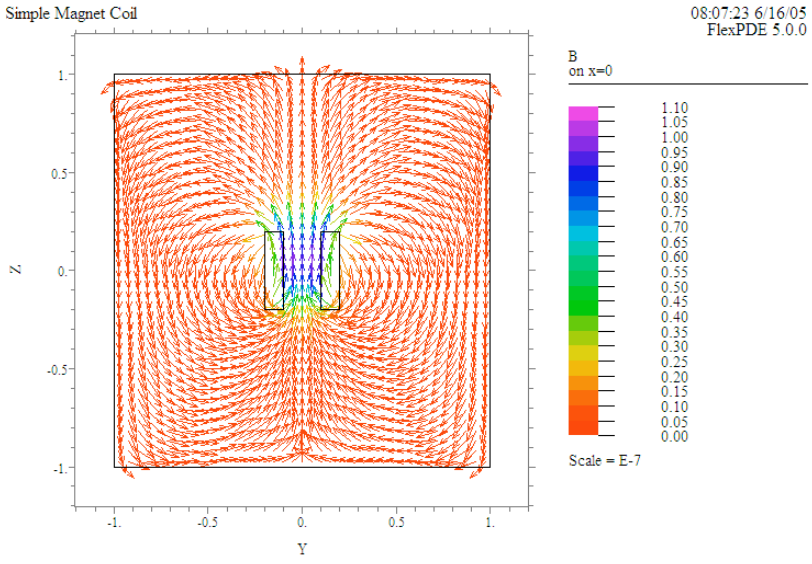
```
TITLE   'Simple Magnet Coil'                                  { 3d226.pde }
SELECT     errlim=5e-4   ngrid=1  spectral_colors
COORDINATES     cartesian3                      { Professional Version }
VARIABLES      Ax     Ay     Az
DEFINITIONS
   L=1.0     r1=0.1     r2=0.2     z0=0.2     rad=sqrt( x^2+ y^2)
   mu0=4*pi*1e-7    mu=mu0     J00=1.0          { Current density in coil }
   Bex=dy(Az)-dz(Ay)     Bey=dz(Ax)-dx(Az)     Bez=dx(Ay)-dy(Ax)
   B=vector( Bex, Bey, Bez)     Bm=magnitude( B)
   Hx=Bex/mu     Hy=Bey/mu     Hz=Bez/mu     Hm=Bm/mu
   J0=0                                          { Default current density }
   Jx=-J0*y/rad    Jy=J0*x/rad     Jz=0
   J=vector( Jx, Jy, Jz)
EQUATIONS
   Ax:        dx( dx(Ax)/mu)+ dy( dy(Ax)/mu)+ dz( dz(Ax)/mu)=-Jx
   Ay:        dx( dx(Ay)/mu)+ dy( dy(Ay)/mu)+ dz( dz(Ay)/mu)=-Jy
   Az:        dx( dx(Az)/mu)+ dy( dy(Az)/mu)+ dz( dz(Az)/mu)=-Jz
EXTRUSION                                        { Parallel surfaces }
   surface 'bottom' z=-L
   layer 'air 1'                                 { Layer below coil }
   surface 'lower' z=-z0
   layer 'coil'                                  { Layer containing coil }
   surface 'upper' z=z0
   layer 'air 2'                                 { Layer above coil }
   surface 'top' z=L
BOUNDARIES
   surface 'bottom'  value(Ax)=0   value(Ay)=0   value(Az)=0
   surface 'top' value(Ax)=0   value(Ay)=0   value(Az)=0
region 'domain'                                  { Full solution domain }
   start (-L,-L) value(Ax)=0   value(Ay)=0   value(Az)=0
   line to (L,-L)  to (L,L)  to (-L,L)  close
limited region 'coil'                            { Winding }
   layer 'coil' J0=1.0
   start (r2,0) arc( center=0,0) angle=360
limited region 'hole'                            { Air space }
   layer 'coil' J0=0
   start (r1,0) arc( center=0,0) angle=360
PLOTS
    vector( J) on z=0 on 'coil'
    vector( B) norm on x=0
    vector( B) norm on x=0 zoom(-0.4,-0.4,  0.8,0.8)
    vector( B) norm on z=0    vector( B) norm on z=L
    contour( Bm) painted on x=0
```

```
   contour( Bm) painted on y=0
END
```

The first vector plot shows the current density. The magnitude is evidently constant, and the current directions follow circles.



The geometry in this problem is the same as in *exa132*, and we may directly compare the results for the vector **B**. Of course, we could also improve conditions in the present example by exploiting the two mirror symmetries. This would yield more accurate result for the same run time.



Steady Heat Conduction

The following example illustrates heat conduction through a copper rod, soldered to an iron plate, as shown in the first figure. Under *extrusion* we first define the three parallel surfaces limiting the metal components.

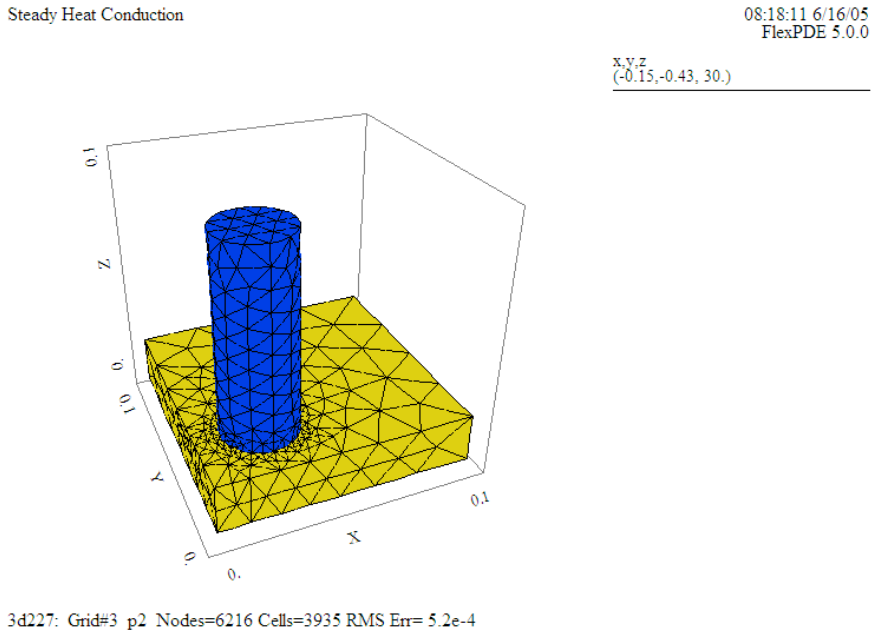
The next task is to assign properties to the two layers delimited by the surfaces. The lower layer consists of iron. In the first step, under 'domain', we declare the upper layer to be void, or empty space. Then we extrude a *cylinder* from the circle in the bottom plane. The conductivity k is non-zero inside this cylinder, but the rest of the upper layer remains insulating.

Under *boundaries* we then specify vanishing flux through the bottom plane and the value 1e6 through the top circle. The bottom of the plate is thus insulated, as is one of the side faces. The other sides are kept at 300 kelvin.

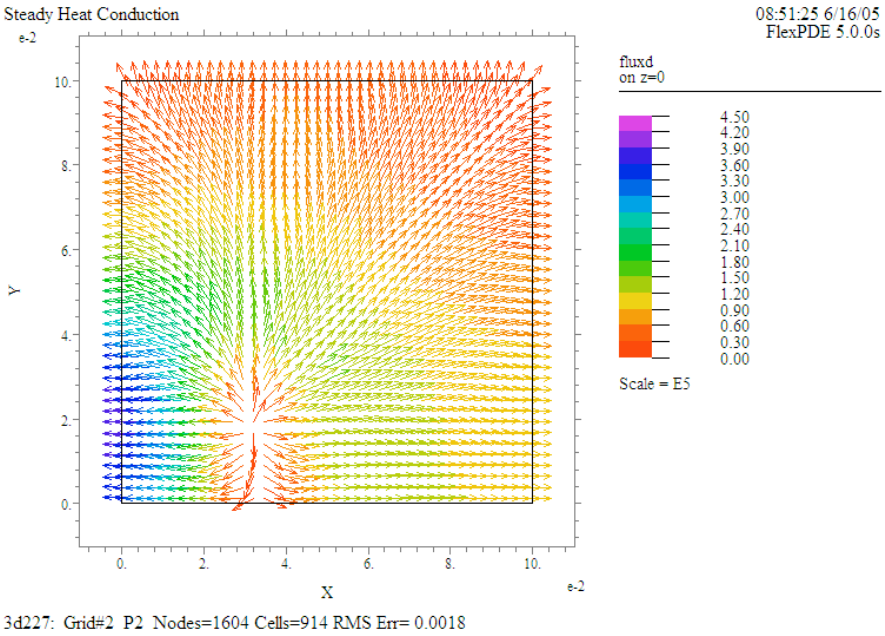
```
TITLE   'Steady Heat Conduction'                              { 3d227.pde }
SELECT    errlim=1e-3    ngrid=1    spectral_colors
COORDINATES   cartesian3                        { Student Version }
VARIABLES   temp
DEFINITIONS
  L=0.1    r0=0.015    z0=0.02                  { Plate thickness=z0 }
  k    heat=0                  { Thermal conductivity and power density }
  in_fluxd=1e6                 { Input heat flux density at top }
  fluxd_x=-k*dx(temp)    fluxd_y=-k*dy(temp)    fluxd_z=-k*dz(temp)
  fluxd=vector( fluxd_x, fluxd_y, fluxd_z)    fluxdm=magnitude( fluxd)
EQUATIONS
  div( fluxd)=heat
EXTRUSION                                       { Parallel surfaces }
  surface 'bottom' z=0
  layer 'iron'
  surface 'middle' z=z0                         { Interface }
  layer 'copper'
  surface 'top' z=L
BOUNDARIES
  surface 'bottom'  natural( temp)=0            { Insulated }
  surface 'top'  natural( temp)=-in_fluxd       { Input flux density }
region 'domain'                                 { Full solution domain }
  layer 'iron'    k=82                          { Thermal conductivity }
  layer 'copper'  void
```

```
  start 'outer' (0,0)   natural( temp)=0
    line to (L,0)   value( temp)=300   line to (L,L)  to (0,L)  close
region 'cylinder'
  layer 'iron'    k=82                                  { Thermal conductivity }
  layer 'copper'   k=400                                { Redefine void }
  start (L/4+r0,L/4)  arc( center=L/4,L/4) angle=360
PLOTS
  grid( x,y,z)
  contour( temp) on x=y     contour( fluxdm) painted on x=y
  vector( fluxd) norm on x=y     vector( fluxd) norm on z=0
  contour( temp) on y=0     contour( temp) on y=L/4
  contour( temp) on x=L/4
  contour( temp) on z=z0 on 'iron'    contour( temp) on z=0
END
```

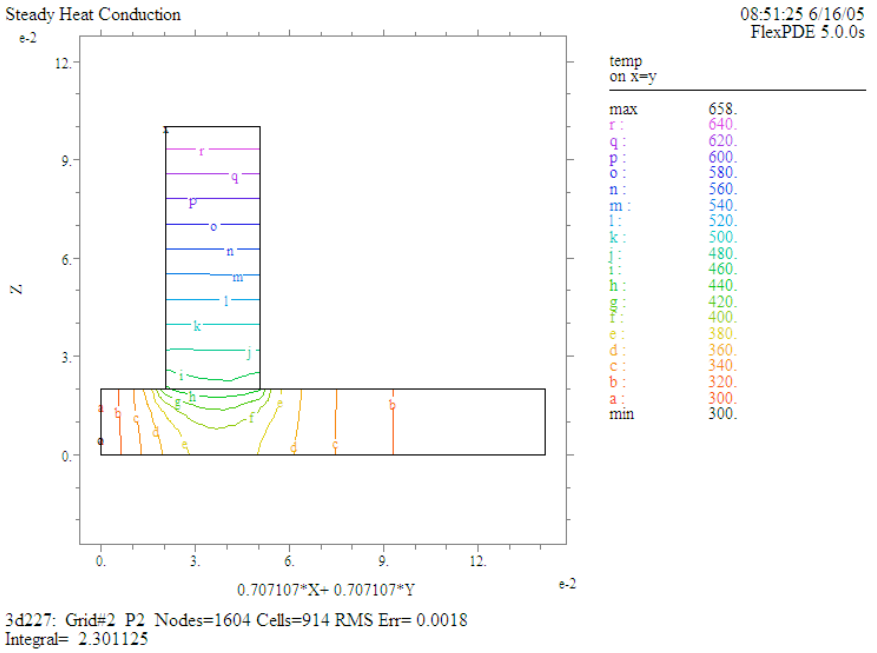
In the following plot, the vertical cylinder is of copper and the base plate of iron.



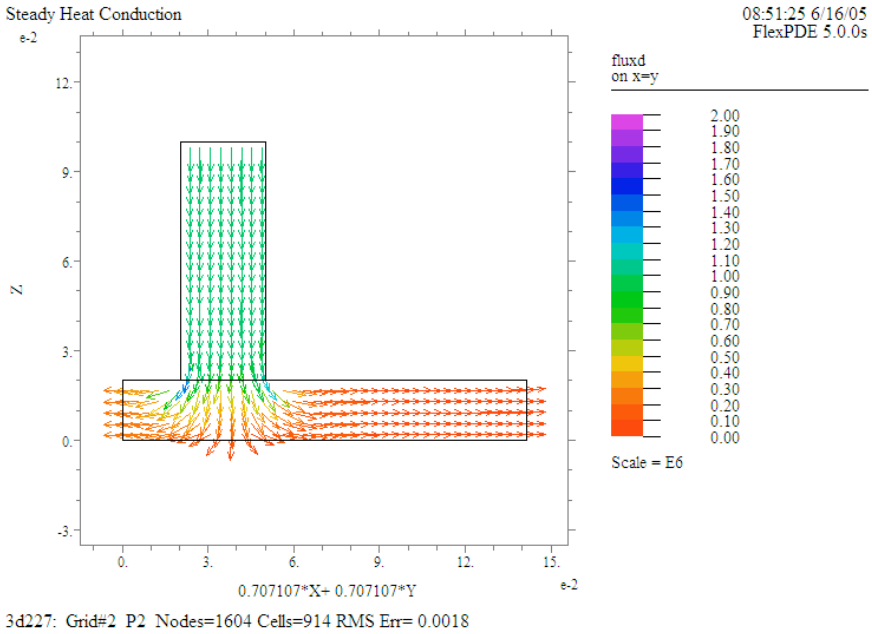
The following vector plot confirms that one side of the plate is insulating. Most of the heat flux exits through the part of the left side that is closest to the copper rod.



The plot of the temperature below shows a linear decrease along the copper rod, except close to the lower end. It also confirms that the iron plate is at 300 kelvin at the corners of the diagonal plane.



The vector plot below finally illustrates the flux density in a diagonal plane, which cuts the copper cylinder in two equal parts. As far as can be judged from the arrow colors, the input flux density is what we specified.



The last two contour plots indicate that three of the sides have uniform temperature (300) as intended.

Temperature Transients

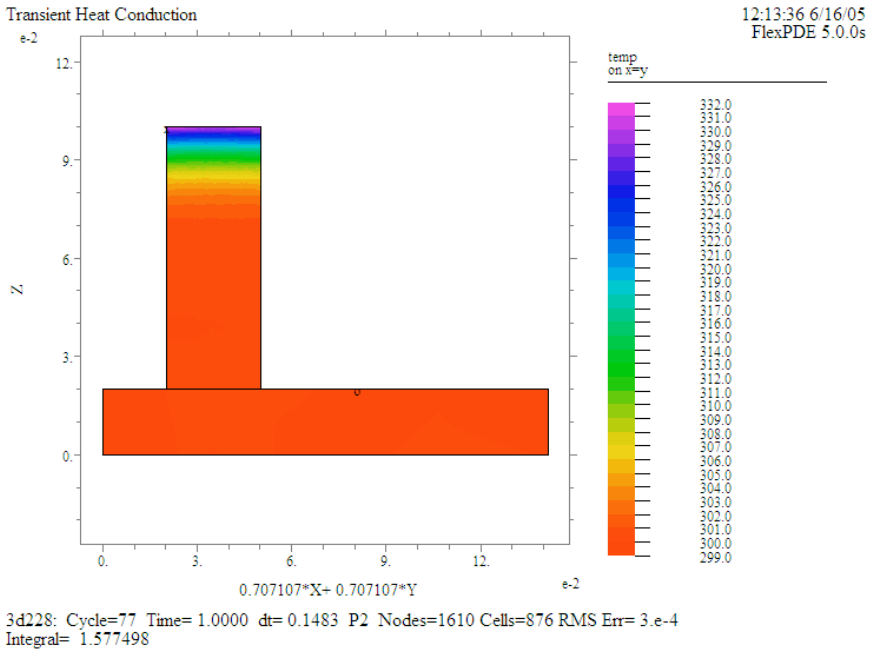
This problem has the same geometry and materials as the preceding one, but now we study the temperature distribution as a function of time, from the moment we apply a constant heat flux at the top.

This problem is analogous to *exa183*. We just need to change *3d227* as follows. This calculation takes rather long time.

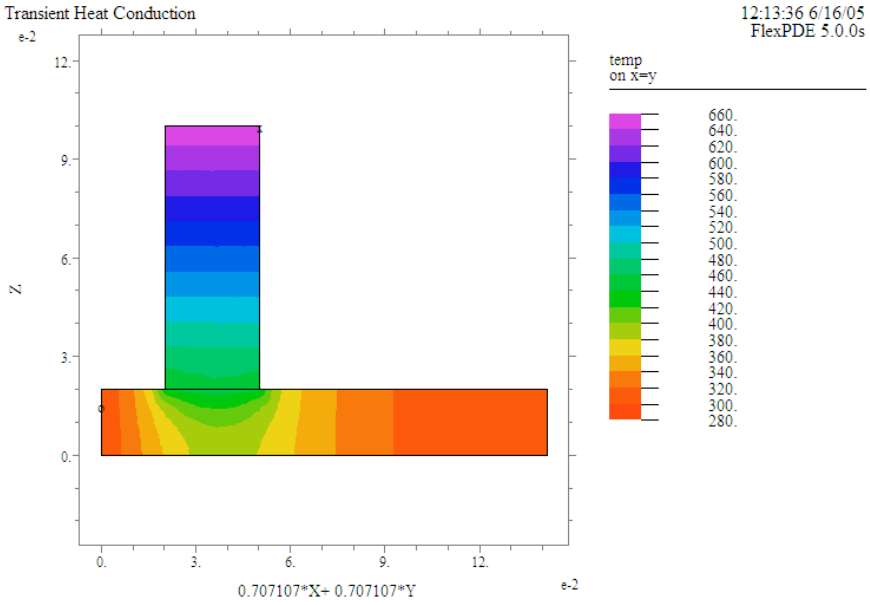
```
TITLE 'Transient Heat Conduction'                    { 3d228.pde }
SELECT errlim=1e-3 ngrid=1 spectral_colors
COORDINATES cartesian3                               { Student Version }
VARIABLES temp( threshold=0.2)
DEFINITIONS
  L=0.1 r0=0.015 z0=0.02
  k rcp heat=0                                       { Thermal parameters }
  in_fluxd=1e6
  fluxd_x=-k*dx(temp) fluxd_y=-k*dy(temp) fluxd_z=-k*dz(temp)
  fluxd=vector( fluxd_x, fluxd_y, fluxd_z) fluxdm=magnitude( fluxd)
  f_angle=sign(fluxd_y)*arccos(fluxd_x/fluxdm)/pi*180
INITIAL VALUES
  temp=300
EQUATIONS
```

```
  div( fluxd)+ rcp*dt( temp)=heat
EXTRUSION                                   { Parallel surfaces }
  surface 'bottom' z=0
  layer 'iron'
  surface 'middle' z=z0                     { Interface }
  layer 'copper'
  surface 'top' z=L
BOUNDARIES
  surface 'bottom'  natural( temp)= 0       { Insulated }
  surface 'top'  natural( temp)=-in_fluxd   { Input flux density }
region 'domain'                             { Full solution domain }
  layer 'iron' k=82  rcp=7.87e3*449         { Fe }
  layer 'copper'  void
  start 'outer' (0,0)   natural( temp)=0
    line to (L,0) value( temp)=300 line to (L,L)  to (0,L)  close
region 'cylinder'
  layer 'iron' k=82  rcp=7.87e3*449         { Fe }
  layer 'copper' k=400  rcp=8.96e3* 385     { Redefine void to Cu }
  start (L/4+r0,L/4)   arc( center=L/4,L/4) angle=360
TIME 0 to 1e6
PLOTS
for t=1, 10, 100, 1e3, 1e4, 1e5, 1e6
  contour( temp) painted on x=y
END
```

On running this file we obtain a sequence of plots like the following one, which shows the temperature distribution at 1.0 s.



The following is the final plot, which corresponds to 1e6 s.



At sufficiently large times the temperature distribution should approach the one previously found in the static case.

Exercises

❑ Replace the coin in *3d223* by a metal cube with a side length of 2*r0.

❑ Modify *3d224* to model a copper cylinder of radius r0 inside an iron block with side lengths equal to 8e-3 and height h=20e-3. Note that the result is predictable.

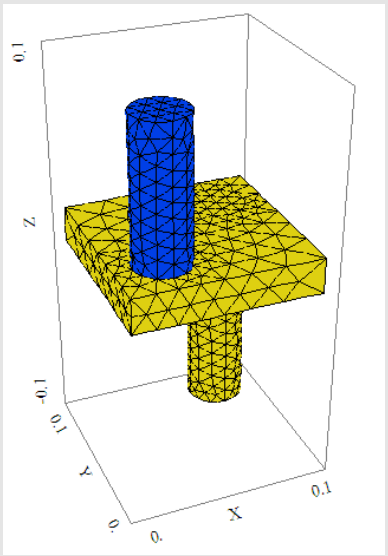
❑ Fill the central cylindrical region in *3d224* with iron and let the conical ring be of copper.

❑ Change *3d225* to make the field *y*-directed. Also modify the 'block' region to obtain a square projection with corners in the *x* and *y* directions. Add suitable plots to show the field concentration at the edges.

❑ Add a second copper rod to *3d227*, on the same side of the plate but centered at (3*L/4,3*L/4). Let all the faces of the iron plate be insulated and let the temperature be 300 kelvin at the end of one

copper rod and 400 at the end of the other one. Hint: create a new region for the second copper rod and specify the temperatures differently on surface 'top'.

❑ Modify the preceding exercise by attaching the second copper rod to the lower side of the plate, as suggested by the figure below.



Appendix:
Principles of Finite Element Analysis

The main idea of finite element analysis (FEA) is to divide the domain of interest into sub-regions (cells) of simple shapes and to solve over each one of those simultaneously. FlexPDE divides the domain into triangles, or rather prisms of triangular cross-sections. Cells in contact with the boundary may have the outer side slightly curved. The cells we see on the screen are thus projections of the prisms on the viewing plane.

The program solves a PDE by determining the values of the dependent variables at discrete points (nodes), i.e. at the corners of the triangles and at the midpoints between corners. The only record of the solution process is thus a list of values at these nodes. In order to obtain function values and derivatives at other points of a triangular cell, the program applies an algorithm for interpolation between nodes.

Interpolation and Differentiation in 2D

Since the interpolation algorithm is of paramount importance in FEA, we begin by studying how it performs. At the same time we test the differentiation routines. *A priori* we do not know precisely how the program calculates derivatives, but we may easily check if the results are accurate.

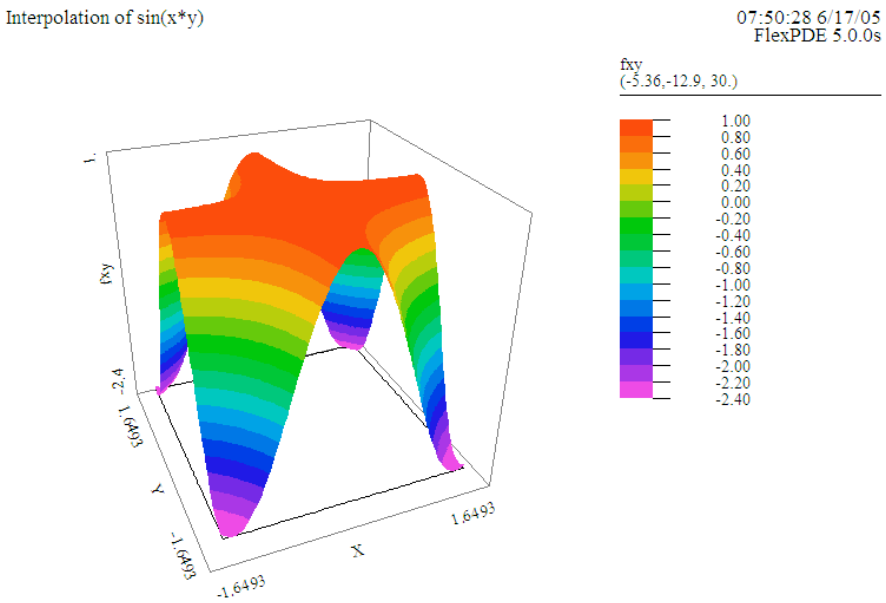
Let us take a simple function of two variables, plot it as well as its derivatives, and compare to the exact expressions for these. This is the objective of the following descriptor, which concerns the function $f(x) = \sin(xy)$.

```
TITLE   'Interpolation of sin(x*y)'                    { apx1.pde }
DEFINITIONS                                          { Student Version }
  Lx=pi/2     Ly=pi/2
```

```
  f=sin(x*y)     fx=dx(f)     fx_ex=cos(x*y)*y              { Exact derivative }
  fxy=dx(dy(f))     fxy_ex=-sin(x*y)*x*y+ cos(x*y)
BOUNDARIES
region 'domain'  start(-Lx,-Ly)
  line to (Lx,-Ly) to (Lx,Ly)  to (-Lx,Ly) close
PLOTS
  contour( f)     surface(f)     surface( fx)
  contour( fx- fx_ex)     surface( fxy)     contour( fxy- fxy_ex)
END
```

The contour plot of the error of fx (not shown) yields exactly zero, which must mean that the program internally uses symbolic differentiation, employing the same analytic procedures as any mathematician would. Our observations thus suggest that the program treats derivatives, and functions of derivatives, specified in the *definitions* segment by exact means. In other words, a single differentiation of an analytic expression does not introduce additional errors.

The next figure is a surface plot of the mixed derivative, which does differ somewhat from the analytic one, as shown by the last plot of the deviation fxy- fxy_ex.



apx1: Grid#0 P2 Nodes=525 Cells=242 RMS Err= 1.
Integral= 2.493731

Interpolating to Obtain a Solution

Once the program has produced a solution, it stores it in a table where the values of the dependent variable (or variables) are given for each node point. To compute the solution at any point in space the program interpolates this table by a quadratic algorithm.

The procedure involved in this interpolation is simple. For each triangular region the program uses a polynomial of the following form

$$P(x,y) = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 .$$

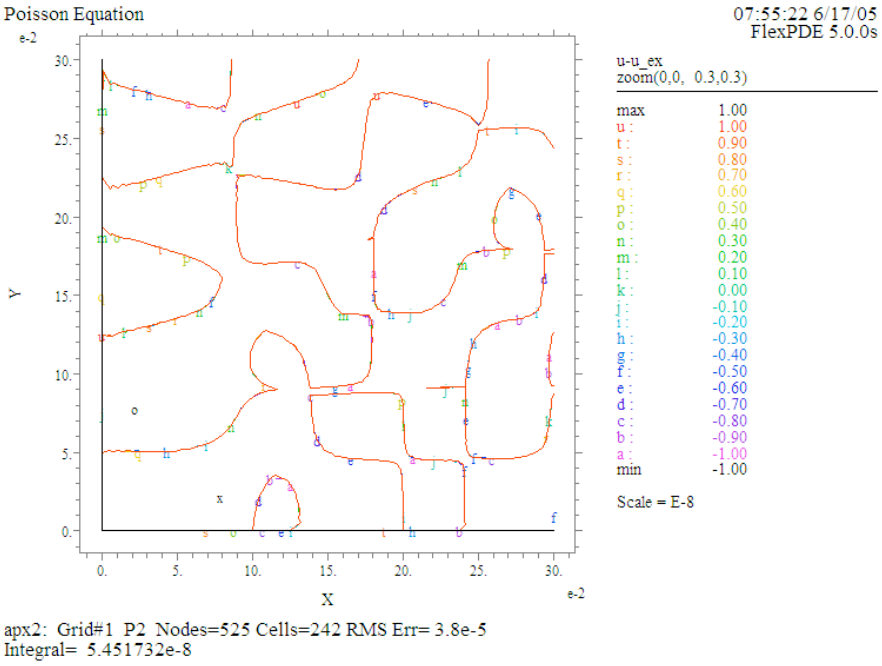
Function values are known at three corners and three midpoints, a total of six points, corresponding exactly to the number of coefficients in $P(x,y)$. In order to determine the coefficients a_i the program just solves a system of six linear equations.

We shall now solve a simple Poisson equation and observe how the program interpolates to obtain a complete solution on the basis of node values only. Starting with an arbitrary function, we apply the Laplacian operator to it. We then use the result $r(x,y)$ of that operation to construct a PDE which has the function we selected as a solution. The equation $\nabla^2 U = r(x,y)$ obviously meets our needs. The value boundary conditions we obtain directly from the initial function, and the descriptor becomes as listed below.

For clarity, we wish to have a small number of nodes, hence the nodelimit statement. Furthermore, we use fixed range to limit the plotted function values to a minute interval around zero.

```
TITLE   'Poisson Equation'                                    { apx2.pde }
SELECT   errlim=1e-15   nodelimit=10
VARIABLES     u
DEFINITIONS
  Lx=1     Ly=1     u_ex=y^2* sin(x)
EQUATIONS
   del2(u)= -(y^2-2)*sin(x)
BOUNDARIES
region 'domain' start 'outer'  (0,0)  value(u)=u_ex
   line to (Lx,0) to (Lx,Ly) to (0,Ly) to finish
PLOTS
   grid( x,y) zoom(0,0,  0.3,0.3)
   contour( u-u_ex)  zoom(0,0,  0.3,0.3)  fixed range(-1e-8, 1e-8)
END
```

The error with respect to the exact solution is of particular interest. Inspecting the following plot we notice that all contour curves are superimposed and correspond to virtually zero. Hence that combined curve should pass through all the nodes, where the agreement is very good. Since the contour curves are generated by interpolation of the error surface, which is rather ragged, we cannot expect to find crossings exactly at the node points. If we overlay the plot below with the grid plot for the same region, however, we find that the zero curve mostly passes through the corners and the midpoints.



apx2: Grid#1 P2 Nodes=525 Cells=242 RMS Err= 3.8e-5
Integral= 5.451732e-8

Solving for Node Values

In FEA, knowing the node values of the dependent variables is equivalent to having a solution to the problem. If the dependent variables are known at all the nodes of the grid, we may interpolate to obtain values at any point of the solution domain and also differentiate the polynomials to obtain derivatives of first and second order.

The crucial task of calculating the values pertaining to the nodes is much more difficult than to interpolate the results. A general, linear PDE of second order reads

$$E_q = \frac{\partial^2 u}{\partial x^2} + f_1 \frac{\partial^2 u}{\partial x \partial y} + f_2 \frac{\partial^2 u}{\partial y^2} + f_3 \frac{\partial u}{\partial x} + f_4 \frac{\partial u}{\partial y} + f_5 u + f_6 = 0$$

where f_k are functions of x and y. If $u(x, y)$ is an exact solution to the PDE, then E_q remains zero over all of the solution domain D. Integrating over this domain we also have

$$\iint_D E_q \, dxdy = 0$$

It is *not* true, however, that a function $u(x, y)$ that satisfies this relation must be a solution: E_q might take positive as well as negative values, which could cancel in the integral. If, on the other hand, we take the square of the PDE, the condition

$$\iint_D {E_q}^2 \, dxdy = 0$$

implies that $u(x, y)$ is a solution. In fact, it is possible to use this relation for solving the equation numerically.

For a given PDE, the above integral of E_q^2 may be regarded as a function of the node values.

$$I(u_1, u_2, \ldots u_n) \equiv \iint_D E_q(u_i, x, y)^2 \, dxdy = 0 \qquad \bullet$$

Of course, in view of the limited accuracy of numerical computation we could never hope to find a set of node values u_i which makes the integral $I(u_1, u_2, \ldots u_n)$ exactly equal to zero, so we must be content with a set that minimizes the integral. This is equivalent to finding the minimum of a function of many variables, and there are standard methods available for solving this type of problem. The effect of taking the square of E_q, however, is to make the integrand more complicated and also non-linear.

An alternative method, which only involves linear analysis, as long as the PDE itself and its boundary values are linear, proceeds as follows. Instead of condensing the problem into a single integral expression as in the scheme above, we now introduce a whole set of equations of the same type (numbered j= 1, 2, ...m):

$$I_j(u_1, u_2, \ldots u_n) \equiv \iint_D W_j(x,y) E_q(u_1, u_2, \ldots u_n,\ x, y)\, dxdy = 0 \qquad \bullet$$

These equations are no longer squared. $W_j(x,y)$ is a weight function, which may emphasize one sub-domain (cell) inside the total solution domain *D*. We are free to choose the various weight functions as we please, but from the point of view of computations the simplest strategy would be to let the weight be either 0 or 1. In the first equation, for instance, we might use unit weight within one of the triangular cells and zero in all the others. The principle is to force the trial function (eventually the solution) to satisfy the PDE over every cell.

One way of applying the above method would be to use unit weight in cell number *j* and zero elsewhere, which would give us *m* equations concerning various parts of the solution domain. The problem is that the number of cells (*m*) always is much smaller than the number of nodes (*n*).

Let us assume that there is only one dependent variable (*u*) and that the boundary conditions are specified by *value* all around. All the values on the boundary are hence known, and we only need to solve for values on the interior nodes. Even this number is generally larger than the number of cells, however, which means that we need some flexible means of increasing the number of equations.

Several methods of choosing weight functions are known[5]. One set of rules that would give the right number of equations is the following. If node *j* is a midpoint, let $W_j = 1$ for the two cells sharing this point and choose $W_j = 0$ elsewhere. If node *j* is a corner, let $W_j = 1$ for the cells sharing this corner and let $W_j = 0$ elsewhere. If a point is on the boundary, we use only one cell. This procedure provides the correct number of equations, and all the sub-domains for integration become different, hence yielding independent equations.

Naturally, we need more sophisticated weighting schemes in the case of several dependent variables. The producers of FlexPDE do not reveal the details of the method actually used.

If the PDE and the boundary conditions are linear, the result of the analysis is a system of linear equations.

If the PDE or the boundary conditions are non-linear we obtain a more general system of algebraic equations. The program solves non-linear systems by iterative methods, which generally take longer time and may be capricious.

Natural Boundary Conditions

So far we have assumed that the *values* of the solution are given on the entire boundary. If the problem involves *natural* conditions over the entire boundary or part of it, this information must be incorporated by special means. A well-known theorem that connects the integral of a PDE with the outward normal component of a field **F** is

$$\iiint \nabla^2 U dV = \iiint \nabla \cdot (\nabla U) dV = \iiint \nabla \cdot \mathbf{F} dV = \oiint \mathbf{F} \cdot d\mathbf{s}$$

When we integrate the PDE over a sub-domain, the volume integral of the Laplacian may be replaced by a surface integral involving the *natural* boundary condition. The other two surfaces of the cell yield integrals that must be matched against those of adjacent cells.

Whenever there are *natural* boundary conditions it is important to remember that multiplying the PDE by a certain factor is bound to change the volume integral, and hence the surface integral over the boundary. Thus, if we multiply through by a factor *f*, the corresponding *natural* boundary condition must also be multiplied by the same factor.

Exercises

❑ Calculate and plot the function $f = \sin(x)\cos(y)$ and its second derivatives. Determine the errors (magnified if necessary) with respect to the exact derivatives.

❑ Investigate the effect of multiplying the PDEs in *exa061* (p.49) and *exa063* by the factor $1 + xy$.

Conclusion

While working your way through this volume, you have probably found that

- you only need a small sub-set of the FlexPDE syntax to generate curves and surfaces
- the simple and powerful plot routines make it easy to display analytic expressions for fields
- solving a standard Laplace or Poisson equation only requires a few more commands
- a solution, including graphical presentation of the results, usually takes only a few seconds
- the gross structure of a descriptor remains the same for a wide variety of problems
- the visual presentation of results is ideal for exploring fields in university education
- at least half of the conventional analytic field theory may be replaced by FEA exploration
- using FlexPDE prepares for future professional FEA work

Academic teachers and students who do *not* agree with some of the above points are invited to discuss these matters by email under the address

gunnar.backstrom@physics.umu.se

References

As an example of the notation in this volume, the superscript^{3p55} *means reference 3, page 55.*

[1] Kreyszig, E. *Advanced Engineering Mathematics, 3rd edn*, John Wiley and Sons, 1972.
[2] Synge, J. L. and Griffith, B. A. *Principles of Mechanics, 3rd edn*, McGraw-Hill, 1959.
[3] Cheng, D. K. *Field and Wave Electromagnetics, 2nd edn*, Addison-Wesley, 1989.
[4] Carslaw, H. S. and Jaeger, J. C. *Conduction of Heat in Solids, 2nd edn*, Clarendon, 1959.
[5] Bickford, W. B. *A First Course in the Finite Element Method*, Richard D. Irwin, 1990.
[6] Küpfmüller, K. *Einführung in die theoretische Elektrotechnik*, Springer, 1968.
[7] Selberherr, S. *Analysis and Simulation of Semiconductor Devices*, Springer, 1984.
[8] Kittel, C. *Introduction to Solid State Physics*, 4th ed., John Wiley & Sons, 1971.

Vocabulary of FlexPDE

The following table is a reminder of the syntax rules, given as descriptor fragments. The commands in blue pertain to 3D. The numbers refer to pages in the book where the usage has been illustrated by examples. More details are available in the *Manual* and under *Help* while using the program.

		pages
TITLE		
'FileName'		7
SELECT		
spectral_colors		10
errlim=1e-5		49
ngrid=1 stages=5 nodelimit=10		60, 189, 270
COORDINATES	{ Default: (x,y) }	
cartesian1	{ One-dimensional (x) }	7
ycylinder('r','z')		44
cartesian3	{ (x,y,z) }	240
VARIABLES		
temp(threshold=1e-3)		201
DEFINITIONS	{ SI units }	
grad_f=vector(dx(f),dy(f)) del2 div		10, 40, 32, 49
E=-grad(U) Em=magnitude(E)		34
globalmin(U) globalmax(Bm)		99, 126
mesh_spacing=0.1*d0*…	{ Applied to domain }	136
Q=line_integral(-normal(D),'rod1')		109
Qe=surf_integral(-2*normal(D),'ellipsoid')		46, 92, 101
W0=area_integral(0.5*eps0*Em^2)		119, 122
W=vol_integral(0.5*Dm*Em)		119
nh0=staged(0.1*ni, ni, 10*ni)		229
unit_x=vector(1,0) ustep(x)		115, 198
transfer('tempr', tempr)	{ Data from file 'tempr' }	218