Chapter 7

Advanced Features

This manual is available on-line at the URL

`http://www.geom.umn.edu/locate/pisces/`

The on-line version may be more current, and may contain features not contained in the printed version. For example an the on-line version contains an interactive interface to Pisces via the World-Wide Web.

7.1 Advanced Users Guide

How to create a customized version of Pisces.

How to install

- New Models
- New Derived Models (not yet written)
- New Algorithms (not yet written)

7.2 Creating a customized version of Pisces

This document refers to the "Pisces root directory", sometimes referred to simply as "`$PISCES`". At the Geometry Center the full pathname of this directory is `/u/gcg/Pisces`, but this path will vary from system to system. In a nutshell, `$PISCES` is the directory that contains all of the downloaded Pisces source code.

There are two steps to complete:

1. create a Pisces "work directory";
2. tell Pisces what kind of machine you are on.

Create a Pisces "work directory"

We suggest the following steps:

- `% setenv PISCES /u/gcg/Pisces`
  (change the directory to reflect your configuration)
- `% mkdir $HOME/Pisces/work`
- `% cd $HOME/Pisces/work`
- `% cp $PISCES/work/Make* .`
- `% cp $PISCES/work/model* .`
- `% cp $PISCES/work/pisces .`
- `% chmod +w *`
- Edit the `Makefile`. Change the first line of the `Makefile` to reflect the location of `$PISCES` on your system.
- Edit the `pisces` script and follow its instructions.

Environment setup (tell Pisces your machine type)

Before you do any Pisces development work you should edit your `.cshrc` (or the init file of whatever shell you use) so that it sets the `MACHTYPE` environment variable to a value that depends on the type of machine you're logged into. Currently the valid values are "sgi", "sun", and "next". As we expand Pisces to run on more machine types we will add more to this list.

In the simplest case, you will always run Pisces from the same kind of machine. If, for example, the machine on your desk is a SUN, then you can put the following line in your `.cshrc` file
`setenv MACHTYPE ''sun''`.

The more complicated situation is that several people at your site want to use Pisces, possibly from different machines. At the Geometry Center our system-wide initialization file sets the `CPUTYPE` environment variable, so you can use its value to determine the value for `MACHTYPE`. The following C-shell command does this, so you can put this in your `.cshrc` file:

```
switch ($CPUTYPE)
  case NeXT:
    setenv MACHTYPE ''next'' ;
    breaksw ;
  case iris4:
    setenv MACHTYPE ''sgi'' ;
    breaksw ;
  default:
  case sun4os4:
    setenv MACHTYPE ''sun'' ;
    breaksw ;
endsw
```

The system of Pisces makefiles uses the `MACHTYPE` value.

7.3 Installing a New Model

Before attempting to install a new model, be sure that you have created you own Pisces work directory. In the steps below, we assume that you have a UNIX shell whose current working directory is your Pisces work directory. For example,
`% cd $HOME/pisces/work`

We will illustrate the procedure of installing a new model by using a concrete example. Suppose that you are interested in studying the level sets of the

planar variety defined by the zero set of the function
x^3 - xy + alpha + beta y

To install this function into Pisces:

Create a template file

Copy a preexisting model file or download and save Pisces' model template file.
For example: save the template file as `template_def.c`

Edit the template file

This step involves typing in the equation for the function whose level sets are of interest, and completing a series of "administrative tasks" such as naming your variables and parameters, and giving default interval ranges for each variable and parameter.
For example:

- Using a text editor, name your function "Template".
  `#define MODEL_NAME Template`

- Define your function in C-syntax. For each function evaluation, the current values of your variables will be stored in the C variables `x[0], x[1],...`. The current values of your parameters will be stored in the C variables `p[0], p[1],...`, and the value of your function evaluated at a point should be stored in `f[0], f[1],...`. For example, the cubic polynomial presented above would be defined as
  `f[0]=x[0]*x[0]*x[0] -x[0]*x[1] +p[0] +p[1]*x[1];`
- Define a Jacobian. This step is optional, but is recommended when using Pisces for research or when studying functions with singularities or large derivatives. If you want Pisces to use an analytic Jacobian, follow the instructions provided in the template file.
- Define the model's domain, parameter space, and range space. Each line has the form
  `{''x'', -0.95, -1.0, 1.0}`
  where `x` is the name of a variable you are defining, `-0.95` is its initial value, and *[-1,1]* is the default domain for computations and graphics involving the x variable.

- Tell Pisces whether the function is a polynomial, and, if so, what is its degree.

Add your model to the Pisces model menu

Edit the file `model_config.c` to add a line of the form
`DEFINE_MODEL( Template, ''My First Model'', ''2D'');`
The first entry is the name of the model that you indicated in the `MODEL_NAME` macro of the template file. The second entry is a string that will appear on the Pisces menu of installed models. The third string determines a model "subclass." *Subclasses are not yet implemented.*

Add the name of your template file to the Makefile

Edit the file `Makefile` and add the line
`SRCS= model_config.c template_def.c`
after the initial set of comment lines.

Compile and link your code with the standard Pisces libraries

This only requires that you type `make pisces`, and then subsequently invoke the executable `./pisces` whenever you want to use the model that you installed.

7.4 Installing a New Derived Model

This documentation is not yet written.

7.5 Installing a New Algorithm

This documentation is not yet written.