The UBC Java graphics tutorial

This tutorial has a very limited aim: *to teach you enough Java to construct interesting mathematical presentations which will be accessible on the Internet—i.e. readable through one of the standard Internet browsers*. Java programs that run inside browsers are called **applets**.

I assume that you are somewhat familiar with mathematical graphics of some kind. Other than that, it is not clear to me what prerequisites are necessary in order to follow this tutorial. It will not teach you much about Java as a general programming language; its goal is to introduce some examples which you can modify yourself in presumably elementary ways. At some point, undoubtedly, extending these examples will require more knowledge about both Java and geometry. Some references which you may then find useful are:

- Sun's Java tutorial—track it from `http://www.javasoft.com`. Our local copy is at

  `file:/home/faculty/feldman/java/tutorial/index.html`.
- Sun's Java API (Application Programmer's Interface) documentation, the basic manual for Java. Sun's documentation for the version of Java we are using is at

  `http://www.javasoft.com/products/jdk/1.0.2/apidocs.html`.

  Our local copy is at

  `file:/math/local/java/jdk1.0.2/doc/api/index.html`.
- A good basic reference text on Java, such as *Java in a Nutshell* by David Flanagan. It has among other things an extensive listing of classes and methods. It is published by O'Reilly, whose Internet catalogue is at

  `http://www.ora.com`
- An informal introduction to Java. Which one is largely a matter of personal taste. I happen to like Peter van den Linden's *Just Java*.

If you want to develop Java programs on your own computer, you will need to acquire Java developers' software for yourself. There are fancy ones you can pay real money for, but the basic Java Developer's Kit available at no cost from Sun is quite adequate. It includes the Java API mentioned above. You can download it from

`http://www.javasoft.com/products/jdk/1.1/index.html`.

One thing to keep in mind is that there are several versions of Java currently in use, even among those produced by Sun. This tutorial will use only the earliest among the ones currently in use, Java 1.0.2. This is because at this moment most browsers are not capable of using features from more recent ones. Everything we present here will run under recent versions, but in compiling your programs you may get frequent warnings about obsolescence or 'deprecation' (which you should simply ignore). The obsolete code will still run in future versions of Java, but might not compile. You should keep this in mind if you plan on rebuilding your applets several months from now. The Nutshell book mentioned above attempts to distinguish between features from versions 1.0 and 1.1.

This tutorial will explain a package of routines, developed here in the Mathematics Department at UBC, which you can use in constructing mathematical graphics with Java. It is called **psplot**. All applets using this package will have the same basic structure—they will display a simple drawing surface called a **PlotCanvas**, something like the page on which you draw in the 2D graphics computer language PostScript. Any figures or text have to be placed graphically on this canvas. The commands you use to draw on the canvas are modeled after those in PostScript.

There are other ways to put graphics in an Internet browser, but for the purposes of mathematical exposition Java applets are probably the best, even for 'still pictures'. Quality is good, and since Java is a serious programming language the only limits to what you can do are essentially a matter of performance. However, there are two additional features which make Java unique: applets can be **animated** and they can be **interactive**.

The basic Java language utilities include some graphics capability, but especially as far as mathematical figures are concerned they is severely limited. Coordinates are in integers, and the coordinate system is fixed. The package **psplot** therefore has two important features which the basic Java library doesn't: (1) it allows real numbers as coordinates; (2) you can can change coordinate systems in it. For mathematical purposes, these are vital requirements.

The next major release of Java, version 1.2, will include an 'official' 2D graphics library (the Java-2D library). Specifications are available on the Sun Web site mentioned earlier. It is was developed in collaboration with Adobe, so it shouldn't be too surprising that it has many features in common with PostScript. It will handle many of the functions that our much simpler **psplot** collection does, and because it will have access to underlying system routines, it will presumably be faster and of higher quality than our home grown stuff. Even so the routines of **psplot** will perhaps have some use—in the short run, because the official library will be large and complicated, and it may be quite a while before browsers can use it; in the long run, because it does a few things that you would otherwise have to add to the 2D library anyway.

This entire package will be available through the Internet. One location is from the course page for the fall, 1997 term of Mathematics 308 at UBC:

`http://www.math.ubc.ca/people/faculty/cass/courses/m308-97b.html`

To get the package, download one of the files `psplot.zip` or `psplot.tar.gz`. When you unpack it, you will see a lot of files in three directories, `psplot`, `real`, and `psapplets`. The basic routines are in the first two. The third directory contains examples of Java applet programs which use the packages. In the rest of this tutorial we shall look at three of these examples, which can be run, together with a third, from the file `psapplets/ps.html`.