12.010 Computational Methods of Scientific Programming
Lecture 13

Today’s lecture
• More Mathematica functionality
• Mathematica Version of poly_area.f

Review of Lecture 12

• Started looking at Mathematica and went over the basic features:
  – Front End / Kernel
  – Notebooks
  – Arbitrary Numerical Precision
  – Programming Control Features
Resources

• Mathematica Book
• Third Party Books
  – Programming in Mathematica
  – Beginner’s Guide to Mathematica
    • Electronic Book as well!!*
• Web Sites*
  – http://www.wolfram.com
  – http://www.mathsource.com
• Internal*
  – ? and ??
  – Help

Conversion of Units*

• Add the units package: <<Miscellaneous`Units
• Convert[quantity with units, new units]
• ConvertTemperature[number, old scale, new scale]
• SI[ ], MKS[ ], CGS[ ]
Variables and Scope

• Variable Types (optional, but can be helpful)
• Contexts (Global` and others)
• Local variables in Module
• Mathematica iterator in Do, Table, and others*
  – {max}, {symbol, max}, {symbol, min, max}
  – {symbol, min, max, increment}

Assignment and Delayed Assignment

• Assignment and delayed assignment
  – = is immediate
  – := is delayed
• Functions that remember their values*
  – f[x_] := f[x] = function of x
  – Mathematica looks for specific definitions first, so it checks first for previously calculated values are checked
  – ? yields the usage, if defined: f::usage = "string";
  – ?? yields the definition and all the values
String Operations*

- ToString[ ] and ToExpression[ ]
- String Manipulation
  - StringTake
  - StringDrop
  - StringJoin
- Sorting strings
- Characters from strings

Importing and Exporting*

- Directories and $Path
- Importing Data
- Images
- Saving Variables (Save[ ])
- Saving Kernel States (DumpSave[ ])
Mathematica Programming

- Evolution of code
  - Set of statements
  - Group into functions
  - Package of functions
- Use of Module
  - no compiling
- Mathematica Compile[]
  - logical, integer, real, or complex arguments
  - single value returned (not a list)
  - speeds up numerical calculations

Translating poly_area.f

- Expressions go from Mathematica to FORTRAN with FortranForm[]*
- Basic Translation from FORTRAN to Mathematica
  - Possible, but not Mathematica savvy
- Mathematica style
  - Descriptive variable names (begin with lower case and use embedded capitals)
  - Short functions that use the power of Mathematica
  - Use lists instead of arrays
  - "map" over lists rather than loop
- Example of poly_area.f -> polyArea*