SOFTWARE FOR

INTRODUCTION TO SYMMETRY ANALYSIS

Version - 2.0

Release Date - September 1, 2001

Created by - Brian J. Cantwell

CONTENTS OF THE COMPACT DISC

The CD enclosed with the text *Introduction to Symmetry Analysis* contains the following items.

1) A folder called **SymmetryAnalysis** containing a *Mathematica*® notebook called

IntroToSymmetry.nb

and its corresponding autosave package file titled

IntroToSymmetry.m.

In Windows the autosave package file appears as a text file. The notebook file is 136k in size and the autosave file is 104k. Any modifications of the package are made by editing the notebook file. The changes are automatically applied to the package when the notebook is saved. The package was written in *Mathematica*® version 4.0 and requires that version or higher to run.

2) A folder titled **Sample_runs** containing several folders labeled by chapter numbers in the book. The folder **Sample_runs** occupies about 3.3 Mbytes of hard disc space and contains a total of sixty eight sample notebooks. Many of the samples correspond to problems discussed in the text and these notebooks are labeled by the corresponding chapter and section number, example number or exercise number. In addition there are a number of miscellaneous examples illustrating the use of the package in a wide variety of circumstances. A number of the examples demonstrate the use of built-in *Mathematica*® functions to complete the solution for the infinitesimal groups in cases where the package function for solving the determining equations fails. The notebooks are heavily commented.

3) A copy of *MathReader*® with its installer. This application enables a user who does not have access to *Mathematica*® to read the sample notebooks so as to follow the procedure used to find symmetries and study the results that are obtained by the package. Only PC and Mac installers are enclosed. There are several installers for the various versions of Unix and the user can download the appropriate installer from the Wolfram Research website at http://www.wolfram.com/products/mathreader. If you have *Mathematica*® version 4.0 on your machine do *not* install *MathReader*® otherwise it will take priority and your *Mathematica*® files will come up in *MathReader*® first.

4) A copy of the GNU General Public License version 2, June 1991 that applies exclusively to the package and sample runs. This license does not apply to any of the files generated by the *MathReader*® installer or to the installer itself.

GETTING STARTED

On the Macintosh, the folder **SymmetryAnalysis** is copied from the CD and placed in the *Mathematica*® StandardPackages folder on the user's hard disc. The hierarchy of folders is

Mathematica 4.0 Files\AddOns\StandardPackages\SymmetryAnalysis

Installation on a Windows system is similar. First copy the folder **SymmetryAnalysis** from the CD. Then change directory to the *Mathematica*® StandardPackages folder and paste the copied folder into the StandardPackages folder. The hierarchy of directories is

C:\ProgramFiles\WolframResearch\Mathematica\4.0\AddOns\StandardPackages\SymmetryAnalysis

The package is loaded using the **Needs** function. The command is,

```
Needs["SymmetryAnalysis`IntroToSymmetry`"]
```

On a Unix system one is often in the position where the *Mathematica*® folder cannot be changed without Root access. In this case the file **IntroToSymmetry.m** should be located in the user's home directory. The package is loaded using the **<<** symbol. In this case the command is,

```
<<IntroToSymmetry`
```

Note the left leaning apostrophe at the end.

Note that Windows and Unix may retain the read-only file attributes of the CD when the files are copied. Once the files are on your system the file access designations should be changed to read-write.

For students working on a campus system, a distributed site licensed version of *Mathematica*® may be installed on your university network. In this case you will need to contact your instructor or system administrator for information regarding the installation of the package contained on this CD.

Once the software has been loaded on your hard disk go to the folder labeled **Sample_runs** and open one of the sample notebooks. Some of the samples can take a while to run. As a first try I would suggest one of the notebooks from the folder labeled Chapter_8_ODEs or Chapter_9_PDEs. Go to the Kernel item in the menu, pick Evaluation and then Evaluate Notebook. The notebook should run regenerating the results that it had contained already. I strongly suggest re-running all sixty eight sample notebooks at some point since they contain a variety of cases where the package may have difficulty solving for the group infinitesimals and certain strategies are used to complete the solution. At the very least be sure to run all the cases in the folder Chapter_8_ODEs and read the associated comments. Note the timing and memory information contained in each sample notebook.

The easiest way to originate a new case is to select one of the sample notebooks, duplicate it and edit it appropriately. If your notebook fails to run properly carefully check the way you entered the equation. Check it where it appears as an input equation and where it appears as a replacement rule. The vast majority of problems are associated with incorrectly formatting the equation or its corresponding replacement rule. The second most likely source of problems is an incorrect choice of the value of one of the parameters required by the package such as the equation order.

MAIN FUNCTIONS CONTAINED IN THE PACKAGE

There are a number of functions provided with the package but the two core functions are called **FindDeterminingEquations** and **SolveDeterminingEquations**.

The function **FindDeterminingEquations** takes a set of ODEs or PDEs provided by the user and executes a purely algorithmic procedure to generate a list of linear PDEs called the determining equations of the group. This is where the vast bulk of hand calculation was needed in the past to implement the Lie algorithm. This function has proven to be extremely fast and reliable and requires only a modest amount of computer memory even for a fairly complicated system of PDEs. There are a number of variables and tables that the user can access to follow the process at any desired level of detail.

Note that the package will not handle equations with derivatives higher than 14th order due to a limitation of *Mathematica*® which only allows up to 13 optional arguements in a defined function. In addition, the package is designed to find Lie-Bäcklund symmetries for systems containing no more than 18 equations. This has to do with the number of equations for which the package will internally generate higher order differential consequences to supplement the replacement rules put in by the user. There is no limit on the number of equations that can be analyzed for point symmetries. These are not significant practical limitations on the use of the package at least for present day computer systems. The fact is, almost any desktop computer will be quickly overwhelmed if the user tries to analyze Lie-Bäcklund symmetries of systems with a dozen or more equations or point symmetries of equations that depend on derivatives above 9th or 10th order.

The function **SolveDeterminingEquations** solves the system of determining equations using a multivariate polynomial approach. This function is useful for finding groups of algebraic form and for finding infinite dimensional groups. Infinitesimals that depend on special or transcendental functions can usually be found interactively by using *Mathematica*® built-in functions to simplify and solve the system of determining equations. Several sample notebooks are included where this sort of interactive process is used to find such groups.

A much more detailed description of the package can be found in Appendix 4 of the text.

LOOKING AT THE RESULTS

The main result of the package is the set of determining equations and these can be viewed by looking at the list of strings **zdeterminingequations**. These are expressed in terms of a standard set of so-called z-variables **(z1,z2,z3,...)**. The determining equations can be viewed in terms of conventional user-defined variables using the command

ToExpression[zdeterminingequations]/.ztableofrules.

The results for the symmetries of the equation(s) in question are available in two forms. The tables **xsefunctions** and **etafunctions** contain lists of strings for the infinitesimals of the group in the form of polynomial expressions with the group parameters as coefficients. The table **infinitesimalgroups** contains a list of the individual groups with the group parameters stripped away. Presenting the list **infinitesimalgroups** as a column using the *Mathematica*® built-in function **ColumnForm** constitutes the most readable form of the

results. The package function **MakeCommutatorTable** can be used to construct the commutator table of the group from the list **infinitesimalgroups** after z-variables have been converted to user-defined variables.

TIMING, MEMORY AND SAVING INTERMEDIATE DATA

The *Mathematica*® built-in functions **Timing** and **MaxMemoryUsed** are used to inform the user of the time and memory requirements of each sample notebook. The computer used is my Mac G4 with 500 Mhz dual processors and 1000 Mbytes of RAM.

When the function **FindDeterminingEquations[...]** is called I usually imbed it in the timing function. Thus the usual command is **Timing[FindDeterminingEquations[...]]** so that at the end of execution, the time in seconds required to execute the function is output to the notebook. For most simple input equations the time required to find the determining equations is usually only a few seconds. For a complex equation that may be one of a system of equations such as one component of the 3-D Navier Stokes equations the time may be several minutes.

Searching for Lie-Bäcklund symmetries of a system of equations is another matter. It is not difficult to define a problem where **FindDeterminingEquations** can take a long time and use lots of memory. The infinitesimals of Lie-Bäcklund symmetries depend on derivatives of the dependent variables and if the number of dependent variables is large then the number of variables on which the infinitesimals depend can be quite large. For example, the four infinitesimals of a first order Lie-Bäcklund symmetry of the incompressible, unsteady, 3-D Navier-Stokes equations depend on 4 independent variables, 4 dependent variables and 16 first derivatives; twenty four in all! Generating the determining equations can take 4-5 hours per equation.

It is important therefore to realize that the user need not generate all the results in a single run of the notebook. Results can be written to the hard disc by the *Mathematica*® built-in function **Save**. For example, the command

```
Save["zdeterminingequations1file", zdeterminingequations]
```

takes the table of determining equations called **zdeterminingequations** and saves it on the hard disc in a file called **zdeterminingequations1file**. The file is located in the *Mathematica*® folder. Intermediate results that take a long time to generate can be saved and the notebook can be closed. At a later time when the notebook is reopened the table **zdeterminingequations** can be called back into memory using the command **<<zdeterminingequations1file**. The notebook can then pick up where it left off without having to re-execute the function **FindDeterminingEquations** that generated **zdeterminingequations** previously. This way the results of a long calculation can be generated and saved in several sessions.

Similarly, when the function **SolveDeterminingEquations[...]** is called I generally imbed it in the timing function. The call is **Timing[SolveDeterminingEquations[...]]** so that at the end of execution the time in seconds needed to solve the determining equations is output to the notebook. The package function **SolveDeterminingEquations** substitutes a multivariable polynomial expansion of the infinitesimals into the determining equations and various products of the dependent and independent variables are then gathered together. The factors multiplying each product are set to zero forming a linear

system of equations for the polynomial coefficients. The function **SolveDeterminingEquations** then calls the *Mathematica*® built-in function **Solve** which uses Gaussian elimination to symbolically determine the polynomial coefficients (most of which are usually zero). In the end, the nonzero coefficients are expressed in terms of a small subset of the original polynomial coefficients and that subset becomes the set of arbitrary group parameters of the infinitesimal transformation. If the number of linear equations for the polynomial coefficients is large then the process of symbolic solution via Gaussian elimination can take a long time. To help the user estimate the time required, **SolveDeterminingEquations** outputs to the notebook the number of unknowns being solved for and the (generally larger) number of equations being solved before **Solve** begins executing. If it looks like the time is going to be excessive the user can interrupt the calculation. The time required to solve the equations can be estimated from,

$$\frac{time}{time_{ref}} = \left(\frac{number\ of\ equations}{number\ of\ equations_{ref}}\right)^n$$

where the exponent is generally between 2.4 and 2.7 depending on the details of the structure of the system of equations being solved. By running **SolveDeterminingEquations** for polynomials of **order=1** and **order=2**, the user can get a good estimate of the exponent and thus an accurate estimate of the time required for order 3 and higher. Most of the sample programs run in a few seconds or perhaps several minutes. A few, involving systems of equations and polynomials of order 3 or more, can take several hours.

The memory requirements also grow rapidly as the polynomial order increases. Most sample notebooks run in 8-9 Mbytes most of which is occupied by the *Mathematica*® application. The 2-D Navier Stokes equations with 6 variables expressed as 3rd order polynomial expansions takes 21 Mbytes to solve 2392 equations whereas the 3-D Navier-Stokes equations with 8 variables require 120 Mbytes to solve 8514 equations.

The user should be aware that it is quite easy to pose a problem for **SolveDeterminingEquations** that will bring even the largest, fastest computer to its knees. This is particularly true when one is trying to find Lie-Bäcklund transformations that depend on derivatives of order two or more as selected by the parameter **rorder**. As noted above, the number of variables that the infinitesimals are assumed to depend on grows rapidly as **rorder** is increased. In addition, rather high order polynomials are required to capture high order Lie-Bäcklund groups. For example, searching for the **rorder=3** Lie-Bäcklund infinitesimals of the Blasius equation (a modest third-order ODE with two unknown infinitesimals) expressed in terms of 5th order polynomials requires the solution of 730 equations, uses10.7 Mbytes and runs in about 81 seconds. Searching for third order Lie-Bäcklund symmetries of the Burgers potential equation, a PDE with three unknown infinitesimals expressed as 5th order polynomials, requires the solution of 3850 equations, uses 97 Mbytes and takes 8764 seconds to execute.

The package is fully capable of finding high order Lie-Bäcklund symmetries of a system of equations but this requires very substantial supercomputing resources to execute.

Brian Cantwell

Palo Alto, September 2001