

PPODESUITE Manual

Pascal A. Pieters

March 2014

Contents

1	Installation	1
1.1	★nix	1
1.2	Windows	1
1.3	General	1
2	Usage	3
2.1	ODE Function	3
2.1.1	Introduction	3
2.1.2	Template	4

Chapter 1

Installation

The *PPODESUITE* package depends on:

MATLAB Tested on MATLAB version R2013b (8.2) 64-bit.

gfortran (GCC) Tested on version 4.8.1 64-bit.

In order to fulfill these requirements under Windows , the ... package is needed.

1.1 Linux and other Unix variants

1. Meet the software requirements by installing MATLAB and the GCC package. MATLAB download and installation instructions can be found on the MathWorks website. The gfortran/GCC package can be obtained from the `★nix` distribution repository through the distribution package manager. Using Ubuntu for example:

```
$ sudo apt-get install gfortran
```

1.2 Windows

1. Do windows stuff
 - (a) Nested item 1
 - (b) Nested item 2

1.3 General

2. Download and extract the *PPODESUITE* package.
3. Open matlab and navigate to the extracted *PPODESUITE* folder. Add the *PPODE* paths to the matlab path variable.

```
>> PPODE.addPaths
```

4. Now the libraries of the different solvers can be build. In order to do so, execute the following command;

```
>> PPODE_init
```

The options 'Debug' can be used to build the libraries with debugging symbols.

```
>> PPODE_init('Debug', 1)
```

Chapter 2

Usage

2.1 The ODE Function

2.1.1 Introduction

The ODE function of the problem should be written in Fortran95. Here are some main Fortranpeculiarities to consider when writing Fortrancode.

Line Formatting The maximum line width is 72 characters. The first character is used to indicate whether the line is a comment line. The second to fifth character are used to indicate labels. The 6th character is used to indicate the continuation of the previous line.

Listing 2.1: Syntax Example

```
1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123456789012
! Comments should be introduced by either a 'c' or a '!'.
  if (answer .gt. 42) go to 4242
4242 ydot(s) = y(1) * (kp * y(s - 1) - gp * y(s)) + gm * y(s + 1) +
+ km * y(s)
```

For Loops For loops are written using the do statement. They should be written in the form do <label> <var>=<start>, <stop>[, <step>]. The label should refer to a continue statement at the end of the loop.

Listing 2.2: Do-Loop

```
a = 0
do 42 i=1, 20
  a = a + 1
42 continue
! a has the value 20 here.
```

Case Sensitivity The Fortranlanguage is not case sensitive.

2.1.2 Template

The Fortransubroutine that defines the ODE system should have the following arguments:

neq *input* Number of equations.

t *input* The current time point.

y *input* The current value of all states. The length of this vector is equal to **neq**.

ydot *output* This is a vector of length **neq** to which all derivatives of the states should be written.

The parameters are passed using a **common** block. The variable **np** represents the number of parameters. The vector **p** contains the values of all parameters.

Listing 2.3: ODE Template

```
!-----
!  
! PPODE ODE function - Model Name  
!   Short model description.  
!  
! DEVELOPED BY:  
!  
!   Pascal Pieters <p.a.pieters@student.tue.nl>  
!  
!-----  
!  
! ARGUMENTS:  
!  
!   neq :in   Number of states/equations.  
!   t   :in   Current time point.  
!   y   :in   Vector of the current values of the states.  
!   ydot:out  Vector of the numerical derivatives of the states.  
!  
! PARAMETERS:  
!  
!   p(1) :in   s   : Parameter description.  
!   p(2) :in   kp  : ...  
!  
!-----  
  
subroutine func (neq, t, y, ydot)  
integer neq, i, s, np  
double precision t, y, ydot, kp  
double precision , pointer :: p(:)  
dimension y(neq), ydot(neq)  
common /funcpar/ np, p  
  
s = int(p(1))  
kp = p(2)  
...  
ydot(i) = ...  
...  
return  
end
```

Examples can be found in the "*PPODESUITE* Source/examples" folder.