

General Picard Integrator Documentation

1 Description

This gives some documentation corresponding to the code `gen.purpose.picard.int.fox` given in §5. An algorithm is given and then an example is explained and analyzed.

We will consider the initial value problem find $x(t) \in \mathbb{R}^n$ when $x'(t) = F(x)$ and $x(0) = x^0$.

2 Algorithm

Algorithm 1 General Picard Iteration Based Method

input Dimension of ODE, Order of Picard Integration, Number of Time Steps to Do, Time Step Size, Initial Conditions x^0 ,
output $x(t)$ for $t = J \times \text{Time Step Size}$, $J = 1, 2, \dots$ Number of Time Steps to Do
 READDIMORDNTR; ▷ Reads all input except initial conditions
 DIMENSION $\leftarrow$ Dimension of ODE
 MAXORDER $\leftarrow$ Order of Picard Integration
 NTR $\leftarrow$ Number of Time Steps to Do
 DELT(1) $\leftarrow$ Time Step Size
 END READDIMORDNTR;
 $X(I) \leftarrow x_I^0$ Read from input for $I = 1, 2, \dots, \text{DIMENSION}$
for $J = 1 \rightarrow \text{number of time steps}$ **do**
 $PHI(I) \leftarrow X(I)$ for $I = 1, 2, \dots, \text{DIMENSION}$
 for $L = 1 \rightarrow \text{MAXORDER}$ **do**
 Set DA order to L
 DERIVATIVES;
 $PHI2(I) \leftarrow F_I(PHI(1), PHI(2), \dots, PHI(\text{DIMENSION}))$ for $I = 1, 2, \dots, \text{DIMENSION}$
 END DERIVATIVES;
 $PHI2(I) \leftarrow X(I) + \int_0^1 PHI2(I) dt$ for $I = 1, 2, \dots, \text{DIMENSION}$
 $PHI(I) \leftarrow PHI2(I)$ for $I = 1, 2, \dots, \text{DIMENSION}$
 end for
 $X(I) \leftarrow PHI2(I)$ Evaluated at DELT(1) for $I = 1, 2, \dots, \text{DIMENSION}$
 Write Current Time $J \times \text{DELT}(1)$ and X to data file
end for

3 Example

For an example we will use the function $y = f(t) = e^{3t}(\sin(2t) + \cos(2t))$. We can then compute

$$\begin{aligned} f'(t) &= 3e^{3t}(\sin(2t) + \cos(2t)) + 2e^{3t}(\cos(2t) - \sin(2t)) \\ &= e^{3t}(\sin(2t) + 5\cos(2t)) \\ f''(t) &= 3e^{3t}(\sin(2t) + 5\cos(2t)) + 2e^{3t}(\cos(2t) - 5\sin(2t)) \\ &= e^{3t}(-7\sin(2t) + 17\cos(2t)) \end{aligned}$$

We then can make the ODE below from the known solution.

$x_1 = e^{3t}$	$x'_1 = 3e^t = 3x_1$	$x_1(0) = 1$
$x_2 = \cos(2t)$	$x'_2 = -2\sin(2t) = -2x_3$	$x_2(0) = 1$
$x_3 = \sin(2t)$	$x'_3 = 2\cos(2t) = 2x_2$	$x_3(0) = 0$
$x_4 = y$	$x'_4 = f'(t) = x_5$	$x_4(0) = f(0) = 1$
$x_5 = y'$	$x'_5 = f''(t) = x_1(-7x_3 + 17x_2)$	$x_5(0) = f'(0) = 5$

Since this ODE has derivative components with continuous spacial partial derivatives in all neighborhood of the initial conditions, the ODE has a unique solution. The code was used to solve this problem. First, an expansion about $t = 0$ was done as shown in §4. Then, several iterations at different orders.

4 Results

The computed and actual Taylor series of the example closely agree as is shown below. The left column is the computed solution and the right is the expansion of the exact solution about 0 (computed by COSY).

```

gen.purpose.picard.int.example
--- BEGINNING COMPILATION
--- BEGINNING EXECUTION
DIMENSION: 5.000000000000000
MAXORDER: 10.000000000000000
NUMBER OF ITERATIONS: 1.000000000000000
STEP SIZE: 0.500000000000000E-001
COMPUTED SOLUTION

```

I	COEFFICIENT	ORDER	EXPONENTS
1	1.000000000000000	0	0
2	3.000000000000000	1	1
3	4.500000000000000	2	2
4	4.500000000000000	3	3
5	3.375000000000000	4	4
6	2.025000000000000	5	5
7	1.012500000000000	6	6
8	0.4339285714285714	7	7
9	0.1627232142857143	8	8
10	0.5424107142857142E-01	9	9
11	0.1627232142857143E-01	10	10

```

-----
I COEFFICIENT ORDER EXPONENTS
1 1.000000000000000 0 0
2 -2.000000000000000 2 2
3 0.666666666666666 4 4
4 -.888888888888888E-01 6 6
5 0.634920634920634E-02 8 8
6 -.2821869488536156E-03 10 10
-----
I COEFFICIENT ORDER EXPONENTS
1 2.000000000000000 1 1
2 -1.333333333333333 3 3
3 0.266666666666667 5 5
4 -.253968253968254E-01 7 7
5 0.141093474426807E-02 9 9
-----
I COEFFICIENT ORDER EXPONENTS
1 1.000000000000000 0 0
2 5.000000000000000 1 1
3 8.500000000000000 2 2
4 6.166666666666667 3 3
5 0.416666666666666E-01 4 4
6 -3.958333333333334 5 5
7 -3.976388888888890 6 6
8 -2.183134920634921 7 7
9 -.7142609126984126 8 8
10 -.819968033509700E-01 9 9
11 0.539729387125220E-01 10 10
-----
I COEFFICIENT ORDER EXPONENTS
1 5.000000000000000 0 0
2 17.00000000000000 1 1
3 18.50000000000000 2 2
4 0.166666666666667 3 3
5 -19.79166666666667 4 4
6 -23.85833333333334 5 5
7 -15.28194444444445 6 6
8 -5.714087301587301 7 7
9 -.7379712301587300 8 8
10 0.5397293871252200 9 9
11 0.4304334766313928 10 10
-----

```

```

EXACT SOLUTION

```

I	COEFFICIENT	ORDER	EXPONENTS
1	1.000000000000000	0	0
2	3.000000000000000	1	1
3	4.500000000000000	2	2
4	4.500000000000000	3	3
5	3.375000000000000	4	4
6	2.025000000000000	5	5
7	1.012500000000000	6	6
8	0.4339285714285714	7	7
9	0.1627232142857143	8	8
10	0.5424107142857142E-01	9	9
11	0.1627232142857143E-01	10	10

```

-----
I COEFFICIENT ORDER EXPONENTS
1 1.000000000000000 0 0
2 -2.000000000000000 2 2
3 0.666666666666666 4 4
4 -.888888888888888E-01 6 6
5 0.634920634920634E-02 8 8
6 -.2821869488536155E-03 10 10
-----
I COEFFICIENT ORDER EXPONENTS
1 2.000000000000000 1 1
2 -1.333333333333333 3 3
3 0.266666666666667 5 5
4 -.253968253968253E-01 7 7
5 0.141093474426807E-02 9 9
-----
I COEFFICIENT ORDER EXPONENTS
1 1.000000000000000 0 0
2 5.000000000000000 1 1
3 8.500000000000000 2 2
4 6.166666666666667 3 3
5 0.416666666666607E-01 4 4
6 -3.958333333333334 5 5
7 -3.976388888888887 6 6
8 -2.183134920634921 7 7
9 -.7142609126984133 8 8
10 -.819968033509700E-01 9 9
11 0.539729387125222E-01 10 10
-----
I COEFFICIENT ORDER EXPONENTS
1 5.000000000000000 0 0
2 17.00000000000000 1 1
3 18.50000000000000 2 2
4 0.166666666666667 3 3
5 -19.79166666666666 4 4
6 -23.85833333333333 5 5
7 -15.28194444444444 6 6
8 -5.714087301587306 7 7
9 -.7379712301587346 8 8
10 0.5397293871252216 9 9
11 0.4304334766313941 10 10
-----

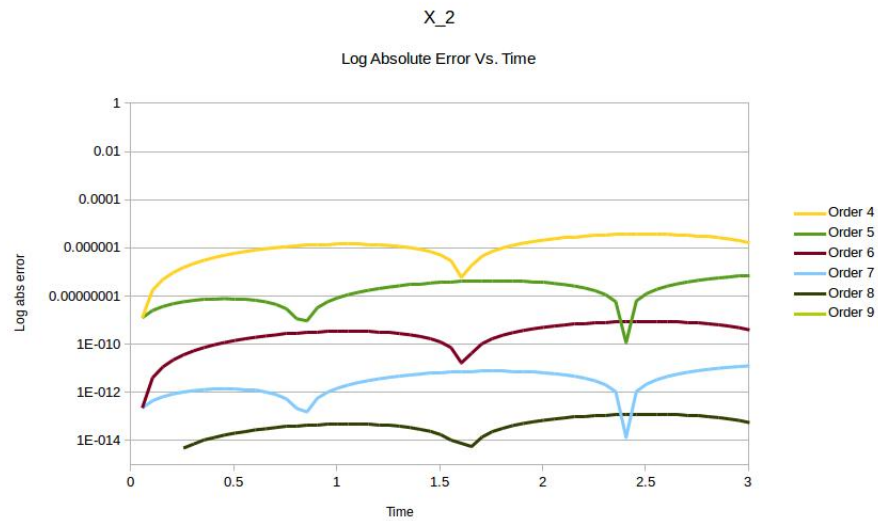
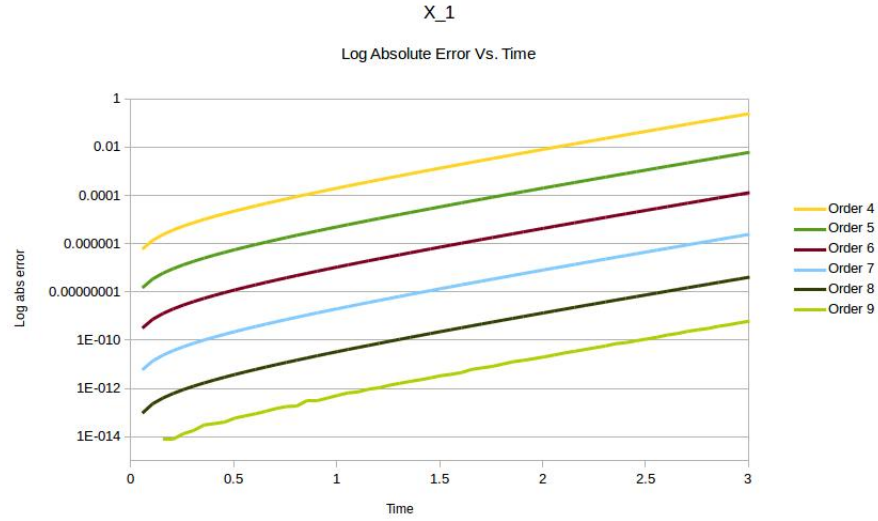
```

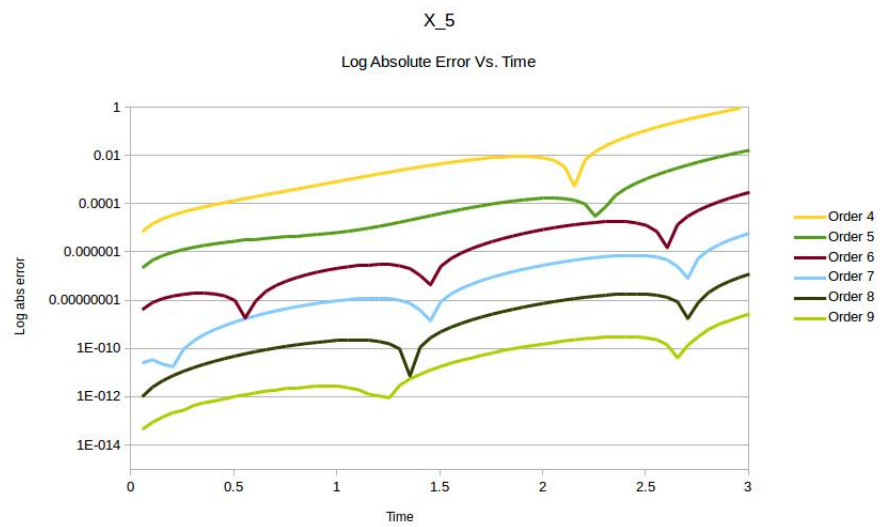
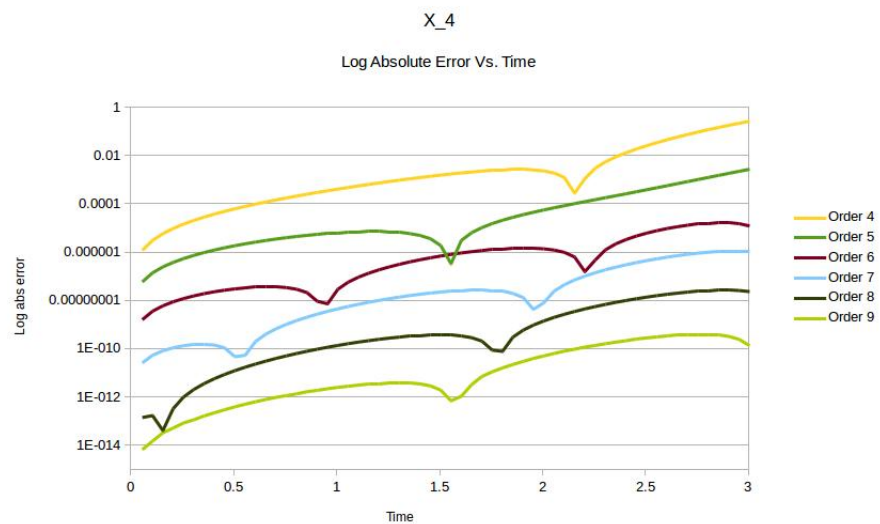
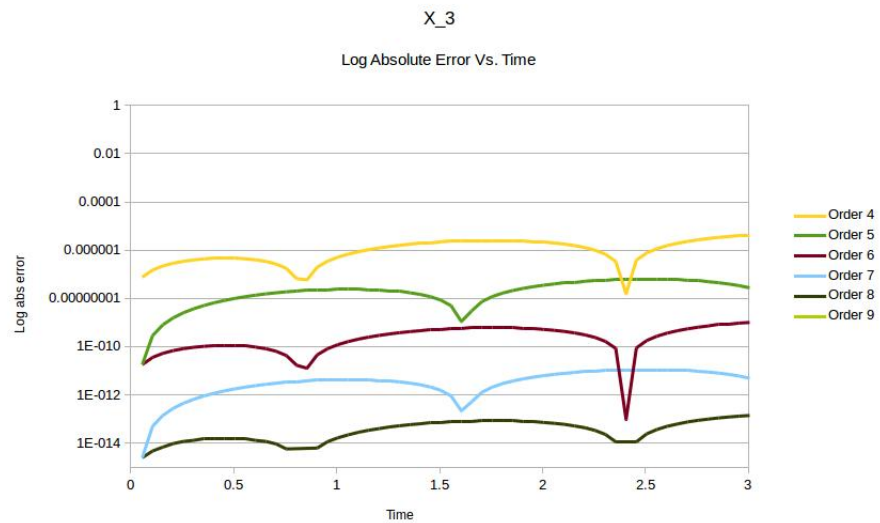
```

=== COSY RUN FINISHED ===

```

With a time step size of 0.05 and doing 60 iterations to solve the example problem. The following plots were made based on the exact solution. There was no difference between the computed solution from the algorithm and the computed exact solution for x_2 and x_3 at order 9.





5 Code

```

1 {
2 General Purpose Picard Iteration Based Integrator
3 Goto Procedure Derivatives to Change Derivative Function
4  $F(\text{PHI}(1), \text{PHI}(2), \dots, \text{PHI}(\text{DIMENSION}))$ .
5
6 Input File: gen.purpose.picard.int.input.ssv
7
8 Output File: gen.purpose.picard.int.output.ssv
9
10 Input File Format
11 DIMENSION – The dimension of the ODE.
12 MAXORDER – The number of Picard Iterations to Perform
13 NITR – The number of iterations
14 DELT – The step size
15 Initial Values for x-1 thru x-n on separate lines
16
17 Output File Format:
18
19 Where t is the elapsed time:
20
21  $t \ x_1(t) \ x_2(t) \ \dots \ x_{\text{DIMENSION}}(t)$ 
22 }
23
24 BEGIN;
25
26 { Global Variables }
27
28 VARIABLE DIMENSION 1; { Dimension of ODE }
29 VARIABLE DUMBY 1; { Dummy Variable }
30 VARIABLE MAXORDER 1; { Maximum Order For Picard Integrator }
31 VARIABLE NITR 1; { Maximum Number of Iterations }
32 VARIABLE DELT 1 1; { TIME STEP SIZE }
33 VARIABLE STRI 1000; { STRING }
34
35 FUNCTION SF R FORM ; RECST R FORM SF ;          ENDFUNCTION ; { REAL TO STRING }
36
37 PROCEDURE READDIMORDNITR;
38 OPENF 42 '.\gen.purpose.picard.int.input.ssv' 'OLD';
39 READ 42 DIMENSION;
40 STRI:= 'DIMENSION:  ';
41 STRI:=STRI & ST(DIMENSION);
42 WRITE 6 STRI;
43 READ 42 MAXORDER;
44 STRI:= 'MAXORDER:  ';
45 STRI:=STRI & ST(MAXORDER);
46 WRITE 6 STRI;
47 READ 42 NITR;
48 STRI:= 'NUMBER OF ITERATIONS:  ';
49 STRI:=STRI & ST(NITR);
50 WRITE 6 STRI;
51 READ 42 DUMBY;

```

```

52 DELT(1):=DUMBY;
53 STRI:= 'STEP SIZE:  ';
54 STRI:=STRI&ST(DELT(1));
55 WRITE 6 STRI;
56 CLOSEF 42;
57 ENDPROCEDURE; {READDIMORDNITR}
58
59 PROCEDURE PICINT;
60 VARIABLE X 1 DIMENSION;
61 VARIABLE I 1; {Index Variable}
62 VARIABLE J 1; {Index Variable}
63 VARIABLE L 1; {Index Variable}
64 VARIABLE PHI (MAXORDER+1) DIMENSION;
65 VARIABLE PHI2 (MAXORDER+1) DIMENSION;
66 VARIABLE NM 1;
67
68 PROCEDURE DERIVATIVES;
69 PHI2(1):=3*PHI(1);
70 PHI2(2):=-2*PHI(3);
71 PHI2(3):=2*PHI(2);
72 PHI2(4):=PHI(5);
73 PHI2(5):=PHI(1)*(-7*PHI(3)+17*PHI(2));
74 ENDPROCEDURE; {DERIVATIVES}
75
76 DAINI MAXORDER 1 0 NM;
77 {READING INITIAL CONDITIONS}
78 OPENF 42 '.\\gen.purpose.picard.int.input.ssv' 'OLD';
79 READ 42 DUMBY; {Skipping Dimension}
80 READ 42 DUMBY; {Skipping Maximum Order}
81 READ 42 DUMBY; {Skipping NITR}
82 READ 42 DUMBY; {Skipping DELT}
83 LOOP I 1 DIMENSION;
84   READ 42 DUMBY;
85   X(I):=DUMBY;
86 ENDLOOP;
87 CLOSEF 42;
88
89 STRI:=SF(0, '(E22.16)')& ' ';
90 LOOP I 1 DIMENSION;
91   STRI:=STRI & SF(X(I), '(E22.16)')& ' ';
92 ENDLOOP;
93 OPENF 43 '.\\gen.purpose.picard.int.output.ssv' 'REPLACE';
94 WRITE 43 STRI;
95
96 LOOP J 1 NITR;
97   {Starting "zeroth" Picard Iteration}
98   LOOP I 1 DIMENSION;
99     PHI(I):=X(I);
100   ENDLOOP;
101   {Finished "zeroth" Picard Iteration}
102   LOOP L 1 MAXORDER;
103     DANOT L; {LOWERS DA VECTOR ORDER}
104     DERIVATIVES;
105     LOOP I 1 DIMENSION;

```

```

106      PHI2(I):=X(I)+PHI2(I)%(-1);
107      ENDLOOP;
108      LOOP I 1 DIMENSION;
109          PHI(I):=PHI2(I);
110      ENDLOOP;
111  ENDLOOP;
112  { Finished Picard Iteration }
113  { Evaluating for the approximate solution }
114  POLVAL 1 PHI2 DIMENSION DELT 1 X DIMENSION;
115
116  STRI:=SF(DELT(1)*J, '(E22.16)')& ' ';
117  LOOP I 1 DIMENSION;
118      STRI:=STRI & SF(X(I), '(E22.16)')& ' ';
119  ENDLOOP;
120  WRITE 43 STRI;
121 ENDLOOP;
122 CLOSEF 43;
123 ENDPROCEDURE; { PICINT }
124
125 { MAIN EXECUTION: }
126
127 READDIMORDNITR;
128 PICINT;
129 END;

```