A PICARD ITERATION BASED INTEGRATOR *

H. Schaumburg, NIU, DeKalb, IL 60119, USA
B. Erdelyi, NIU, DeKalb, IL 60119, USA

Abstract

Picard iteration is mainly used as a theoretical tool to establish the existence and uniqueness of a solution to an initial value problem. We have developed a method based on Picard iteration that computes the exact Taylor polynomial of the solution to arbitrary order. The method has been implemented in COSY INFINITY to numerically solve Coulomb interactions.

INTRODUCTION AND BACKGROUND

Picard iteration generates a sequence of functions $\phi_n(t)$ related to the solution of the initial value problem

$$\begin{cases} \mathbf{y}' = \mathbf{f}(t, \mathbf{y}) \\ \mathbf{y}(t_0) = \mathbf{y}_0 \end{cases}.$$

When $\mathbf{f}$ satisfies a local Lipschitz condition with respect to $\mathbf{y}$ on U, a connected open subset of $\mathbb{R}^{m+1}$ and $(t_0, \mathbf{y}_0) \in U$, the Picard iterates given by

$$\phi_0(t) = \mathbf{y}_0$$
$$\phi_n(t) = \mathbf{y}_0 + \int_{t_0}^{t} \mathbf{f}\left(s, \phi_{n-1}(s)\right) ds.$$

converge to a unique solution of the IVP up to the boundary of U [1]. In general, ϕ_n may converge slowly to the exact solution.

The Picard iteration based integrator described in this paper has three main advantages. The the integrator has arbitrary order, is time adaptive, and has dense output. Dense output refers to the integrator being able to take time steps of variable length without having to recompute previous steps. These advantages are intertwined to balance local truncation error with computational efficiency. When smaller time steps are required, the order can be reduced to maintain efficiency. When a larger time step is appropriate, the order can be increased to maintain lower local truncation error.

The advantages of the integrator make it well suited for modelling the motion of charged particles. The forces in Coulomb interactions are proportional to the inverse of the square of the distances between particles. There are situations such as when two particles with same signed charges are on a near collision course. If too large of a time step is taken, the large repulsive force between the particles as they move closer to one another may not be considered and the integrator will give physically unrealistic results. An integrator with dense output can avoid these errors.

* This work was supported in part by the U.S. Department of Energy, Office of Nuclear Physics, under Contract No. DE-SC0008588, and Office of High Energy Physics, under Contract No. DE-FG02-08ER41532, with Northern Illinois University.

THEORY BEHIND THE INTEGRATOR

First, we will introduce notation to write Taylor polynomials. When f has a Taylor series centered at a with nonzero radius of convergence, define the operator $\mathcal{T}^n_{t,a}$ acting on function f to be the degree n Taylor polynomial centered at a for f. That is

$$\mathcal{T}^n_{t,a}[f] = \sum_{k=0}^{n} f^{(k)}(a) \frac{(t-a)^k}{k!}.$$

Main Theorem Statement

The main theorem below supplies two Picard iteration based integrators with different compositions. From the initial Taylor polynomial $\mathcal{T}^0_{t,0}[\mathbf{z}(t)] = \mathbf{z}_0$, we can find the next Taylor polynomials using either recursive relationship below. After each iteration, the local truncation error drops by an order of magnitude. Algorithm 1 in the implementation and results section shows how to implement the first composition.

Theorem 1. *Let* $\mathbf{z}\,'(t) = \mathbf{f}(t)$ *and* $\mathbf{z}_0 = \mathbf{z}(0)$*. Suppose* $\mathbf{z}$ *has a Taylor series centered at* 0 *with nonzero radius of convergence* R *and* $t < R$*, then*

$$\mathcal{T}^n_{t,0}[\mathbf{z}(t)] = \mathbf{z}_0 + \int_0^t \mathcal{T}^{n-1}_{s,0}\left[\mathbf{f}\left(\mathcal{T}^{n-1}_{s,0}[\mathbf{z}(s)]\right)\right] ds$$
$$= \mathbf{z}_0 + \int_0^t \mathcal{T}^{n-1}_{s,0}\left[\mathcal{T}^{n-1}_{s,\mathbf{z}_0}[\mathbf{f}] \circ \mathcal{T}^{n-1}_{s,0}[\mathbf{z}(s)]\right] ds.$$

In order to prove theorem 1, we need two lemmas. In order to show the first lemma, we will need the Faà Di Bruno formula [2], which is a generalization of the chain rule.

Theorem 2 (Faà Di Bruno)**.** *If g and f are functions with a sufficient number of derivatives, then*

$$\frac{\mathrm{d}^m}{\mathrm{d}t^m} f(g(t)) = \sum \frac{m!}{\prod_{i=1}^m b_i!} f^{(k)}(g(t)) \prod_{i=1}^{m} \left(\frac{g^{(i)}(t)}{i!}\right)^{b_i}$$

where the sum is over all different solutions in nonnegative integers $b_1, \dots, b_m$ *of* $b_1 + 2b_2 + \cdots + mb_m = m$ *and* $k \equiv b_1 + b_2 + \cdots + b_m$*.*

Let $S_{m,k}$ denote the nonnegative integer solutions of $b_1 + 2b_2 + \cdots + mb_m = m$ and $k \equiv b_1 + b_2 + \cdots + b_m$, and $T_{n,m,k}$ denote the nonnegative integer solutions of $b_1 + 2b_2 + \cdots + nb_n = m$ and $k \equiv b_1 + b_2 + \cdots + b_m$.

Lemma 1. *Suppose g has a Taylor series centered at a and f has a Taylor series centered at $g(a)$ with nonzero radii of convergence. Then,*

$$\mathcal{T}^n_{t,a}[f \circ g] = \mathcal{T}^n_{t,a}\left[\mathcal{T}^n_{t,g(a)}[f] \circ \mathcal{T}^n_{t,a}[g]\right].$$

Proof $\mathcal{T}^n_{t,a}\left[\mathcal{T}^n_{t,g(a)}[f] \circ \mathcal{T}^n_{t,a}[g]\right]$ will be directly evaluated. The two things needed to start are

$$\mathcal{T}^n_{t,a}[g] = \sum_{k=0}^{n} g^{(k)}(a)\frac{(t-a)^k}{k!}$$

and

$$\mathcal{T}^n_{t,g(a)}[f] = \sum_{k=0}^{n} f^{(k)}(g(a))\frac{(t-g(a))^k}{k!}.$$

Plugging the first one into the second one, the following can be done:

$$\mathcal{T}^n_{t,g(a)}[f] \circ \mathcal{T}^n_{t,a}[g] = \sum_{k=0}^{n} f^{(k)}(g(a))\frac{(\mathcal{T}^n_{t,a}[g]-g(a))^k}{k!}$$

$$= \sum_{k=0}^{n}\frac{f^{(k)}(g(a))}{k!}\left(\sum_{i=0}^{n} g^{(i)}(a)\frac{(t-a)^i}{i!} - g(a)\right)^k$$

$$= \sum_{k=0}^{n}\frac{f^{(k)}(g(a))}{k!}\left(\sum_{i=1}^{n}\frac{g^{(i)}(a)}{i!}(t-a)^i\right)^k$$

$$= \sum_{k=0}^{n}\frac{f^{(k)}(g(a))}{k!}\sum_{T_{n,m,k}}\frac{k!}{\prod_{i=1}^{n} b_i!}\prod_{i=1}^{n}\left(\frac{g^{(i)}(a)}{i!}(t-a)^i\right)^{b_i}$$

$$= \sum_{k=0}^{n}\sum_{T_{n,m,k}}\frac{f^{(k)}(g(a))}{k!}\frac{k!}{\prod_{i=1}^{n} b_i!}\prod_{i=1}^{n}\left(\frac{g^{(i)}(a)}{i!}(t-a)^i\right)^{b_i}.$$

The fourth step is done using a combinatorial argument. Think of each $\frac{g^{(i)}(a)}{i!}(t-a)^i$ as a letter α_i. What is being counted by $\frac{k!}{b_1!\, b_2!\, b_3! \ldots b_n!}$ is the number of words with k letters that have b_1 letter α_1's, b_2 letter α_2's, … , and b_n letter α_n's.

In $T_{n,m,k}$, $m = b_1 + 2b_2 + \cdots + nb_n$. It must be that $b_i = 0$ when $i > m$ for $i = 1, 2, \cdots, n$. In $\mathcal{T}^n_{t,a}\left[\mathcal{T}^n_{t,g(a)}[f] \circ \mathcal{T}^n_{t,a}[g]\right]$ each term has degree n or less, so $m \le n$. In this case, $T_{n,m,k} = S_{m,k}$, and we have

$$\mathcal{T}^n_{t,a}\left[\mathcal{T}^n_{t,g(a)}[f] \circ \mathcal{T}^n_{t,a}[g]\right]$$

$$= \sum_{k=0}^{n}\sum_{S_{m,k}}\frac{(t-a)^m}{m!}\frac{m!}{\prod_{i=1}^{m} b_i!}f^{(k)}(g(a))\prod_{i=1}^{m}\left(\frac{g^{(i)}(a)}{i!}\right)^{b_i}$$

$$= \sum_{m=0}^{n}\frac{(t-a)^m}{m!}\sum_{S_{m,k}}\frac{m!}{\prod_{i=1}^{m} b_i!}f^{(k)}(g(a))\prod_{i=1}^{m}\left(\frac{g^{(i)}(a)}{i!}\right)^{b_i}$$

$$= \sum_{m=0}^{n}\frac{(t-a)^m}{m!}\frac{\mathrm{d}^m}{\mathrm{d}t^m}f(g(t))$$

$$= \mathcal{T}^n_{t,a}[f \circ g].$$

□

Lemma 2. *Suppose g has a Taylor series centered at a and f has a Taylor series centered at $g(a)$ with nonzero radii of convergence. Then,*

$$\mathcal{T}^n_{t,a}\left[f(\mathcal{T}^n_{t,a}[g])\right] = \mathcal{T}^n_{t,a}[f \circ g].$$

Proof

$$\mathcal{T}^n_{t,a}\left[f(\mathcal{T}^n_{t,a}[g])\right] = \mathcal{T}^n_{t,a}\left[f \circ (\mathcal{T}^n_{t,a}[g])\right]$$

$$= \mathcal{T}^n_{t,a}\left[\mathcal{T}^n_{t,(\mathcal{T}^n_{t,a}[g])(a)}[f] \circ \mathcal{T}^n_{t,a}\left[\mathcal{T}^n_{t,a}[g]\right]\right]$$

$$= \mathcal{T}^n_{t,a}\left[\mathcal{T}^n_{t,g(a)}[f] \circ \mathcal{T}^n_{t,a}[g]\right]$$

$$= \mathcal{T}^n_{t,a}\left[f \circ g\right]$$

The second line is from applying lemma 1. The third line can be seen noting that $\left(\mathcal{T}^n_{t,a}[g]\right)(a) = g^{(0)}(a) = g(a)$ and $\mathcal{T}^n_{t,a} \circ \mathcal{T}^n_{t,a} = \mathcal{T}^n_{t,a}$. □

We are now ready to prove theorem 1.

Proof From lemma 2,

$$\mathcal{T}^{n-1}_{s,0}\left[\mathbf{f}\left(\mathcal{T}^{n-1}_{s,0}[\mathbf{z}(s)]\right)\right] = \mathcal{T}^{n-1}_{s,0}\left[\mathbf{f}\left(\mathbf{z}(s)\right)\right].$$

This implies

$$\mathbf{z}_0 + \int_0^t \mathcal{T}^{n-1}_{s,0}\left[\mathbf{f}\left(\mathcal{T}^{n-1}_{s,0}[\mathbf{z}(s)]\right)\right] ds$$

$$= \mathbf{z}_0 + \int_0^t \mathcal{T}^{n-1}_{s,0}\left[\mathbf{f}\left(\mathbf{v}(s)\right)\right] ds$$

$$= \mathbf{z}_0 + \int_0^t \sum_{k=0}^{n-1}\frac{\mathrm{d}^k}{\mathrm{d}s^k}\left[\mathbf{f}\left(\mathbf{z}(s)\right)\right](0)\frac{(s)^k}{k!}ds$$

$$= \mathbf{z}_0 + \sum_{k=0}^{n-1}\int_0^t \frac{\mathrm{d}^k}{\mathrm{d}s^k}\left[\mathbf{f}\left(\mathbf{z}(s)\right)\right](0)\frac{(s)^k}{k!}ds$$

$$= \mathbf{z}_0 + \sum_{k=0}^{n-1}\frac{\mathrm{d}^k}{\mathrm{d}s^k}\left[\mathbf{f}\left(\mathbf{z}(s)\right)\right](0)\int_0^t \frac{(s)^k}{k!}ds$$

$$= \mathbf{z}_0 + \sum_{k=0}^{n-1}\frac{\mathrm{d}^k}{\mathrm{d}t^k}\left[\mathbf{f}\left(\mathbf{z}(t)\right)\right](0)\frac{t^{k+1}}{(k+1)!}$$

$$= \mathbf{z}_0 + \sum_{k=0}^{n-1}\mathbf{z}^{\,(k+1)}(0)\frac{t^{k+1}}{(k+1)!}$$

$$= \sum_{k=0}^{n}\mathbf{z}^{\,(k)}(0)\frac{t^k}{k!}ds = \mathcal{T}^n_{t,0}[\mathbf{z}(t)].$$

From lemma 1,

$$\mathcal{T}^{n-1}_{s,0}\left[\mathbf{f}\left(\mathbf{z}(s)\right)\right] = \mathcal{T}^{n-1}_{s,0}\left[\mathcal{T}^{n-1}_{s,\mathbf{z}(0)}[\mathbf{f}] \circ \mathcal{T}^{n-1}_{s,0}[\mathbf{z}(s)]\right].$$

It follows then that

$$\mathcal{T}^n_{t,0}[\mathbf{z}(t)] = \mathbf{z}_0 + \int_0^t \mathcal{T}^{n-1}_{s,0}\left[\mathbf{f}\left(\mathcal{T}^{n-1}_{s,0}[\mathbf{z}(s)]\right)\right] ds$$

$$= \mathbf{z}_0 + \int_0^t \mathcal{T}^{n-1}_{s,0}\left[\mathcal{T}^{n-1}_{s,\mathbf{z}_0}[\mathbf{f}] \circ \mathcal{T}^{n-1}_{s,0}[\mathbf{z}(s)]\right] ds.$$

□

ISBN 978-3-95450-138-0

IMPLEMENTATION AND RESULTS

Implementation

The Picard iteration based integrator was implemented using COSY INFINITY. COSY has several unique data types including differential algebra (DA) vectors which were used to store the Taylor polynomials. Many operations on DA vectors are also efficiently coded into COSY such as polynomial multiplication and composition. Built in operations allow for computing the Taylor expansions of various functions to specified order [3].

Algorithm 1 below describes how to compute $\mathbf{z}(t)$ at the points $t = \frac{T}{N}, \frac{2T}{N}, \cdots, T$. Here, T is the total time interval under investigation with N time steps of length Δt. This algorithm uses the first composition in Theorem 1.

Algorithm 1 Picard Iteration Based Integrator

input $\mathbf{z}_0$, $\mathbf{f}(\mathbf{z}, t)$ T, N, Tolerance
$\Delta t \leftarrow \frac{T}{N}$
$\hat{\mathbf{z}_0} \leftarrow \mathbf{z}_0$
for $i = 1 \to N$ **do**
  Compute the "best" smaller time step to use $\frac{\Delta t}{M}$, and an appropriate ORDER for the Taylor series that keeps the local truncation error $\mathcal{O}$(Tolerance).
  $\hat{t} \leftarrow \frac{\Delta t}{M}$
  for $j = 1 \to M$ **do**
    $\phi_0(t) \leftarrow \hat{\mathbf{z}}_0$
    for $k = 1 \to$ ORDER **do**
      $\phi_k(t) \leftarrow \hat{\mathbf{z}}_0 + \int_0^t \mathcal{T}_{s,0}^{k-1} \left[\mathbf{f}\left(\phi_{k-1}(s), s\right)\right] ds$
    end for
    $\hat{\mathbf{z}}_0 \leftarrow \phi_{\text{ORDER}}(\hat{t})$
      ▷ *The above is* $\mathbf{z}\left([i-1+j/M]\Delta t\right)$.
  end for
  Write $\hat{\mathbf{z}}_0$ ▷ *This is* $\mathbf{z}(i\Delta t)$.
end for

In the loop from $j = 1$ to $j = M$ the time scale for the initial value problem is shifted left when j increases so that the initial condition involves $t_0 = 0$.

$$\begin{cases} \mathbf{z}' = \mathbf{f}(t, \mathbf{z}) \\ \mathbf{z}(t_0) = \mathbf{z}_0 \end{cases} \to \begin{cases} \mathbf{z}' = \mathbf{f}(t - t_0, \mathbf{z}) \\ \mathbf{z}(0) = \mathbf{z}_0 \end{cases}$$

This matches the conditions for Theorem 1 and reduces the work of evaluating the definite integral by making the second indefinite integral evaluation 0.

Results

One test of the integrator was to consider a system with a proton following a circular orbit around another super massive particle with opposite charge. The momentum of the proton was set at $\frac{2\ m\ c}{3\times10^5}$ where m is the mass of a proton, and the mass of the massive particle was set at $10^{40}\ m$. Balancing the centrifugal and Coulomb forces, the radius of the orbit is approximately $0.512 \times 10^{-5} m$. For this test, the time step is fixed at around $\frac{0.24169}{c}$s. The theoretical trajectory and integrator computation for the four thousandth orbit are plotted in Fig. 1 as thick black semicircles and red circled points, respectively.

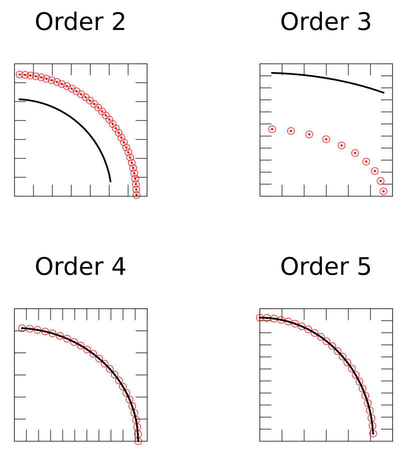

Figure 1: Orbiting Particle Trajectories at Different Orders

Additional details concerning the integrator's performance have been studied [4].

CONCLUSION

By utilizing the Picard iteration based integrator described, a wide class of functions can be integrated numerically. The integrator has the advantages of being variable order, being time adaptive, and having dense output. These advantages have been utilized in a successful implementation to solve Coulomb interactions using COSY INFINITY.

REFERENCES

[1] W. Walter, Ordinary Differential Equations, Springer-Verlag New York, Inc. 1998.

[2] W. P. Johnson, "The Curious History of Faà di Bruno's Formula," The American Mathematical Monthly 109:217-234, March 2002.

[3] K. Makino and M. Berz, "COSY INFINITY Version 9," Nuclear Instruments and Methods in Research Physics Section A, 558:345-350 December 2005.

[4] A. A. Marzouk, "Numerical Integrator for Coulomb Collisions," in these proceedings.