11-27-2012

Using the Taylor Center to Teach ODEs

Alexander Gofen

Follow this and additional works at: http://scholarship.claremont.edu/codee

Using the Taylor Center to Teach ODEs

Alexander Gofen

Abstract: This article introduces a powerful ODE solver called the Taylor Center for PCs (http://www.ski.org/gofen/TaylorMethod.htm) as a tool for teaching and performing numeric experiments with ODEs. The Taylor Center is an All-in-One GUI-style application for integrating ODEs by applying the modern Taylor Method (Automatic Differentiation). The Taylor Center also offers dynamic graphics (including 3D stereo vision). After a brief review of the features of the Taylor Center, we consider instructive examples of ODEs in various applications and also several particular examples illustrating intricacies of numeric integration. The article therefore continues the thesis of Borrelli and Coleman [2] that awareness and caution are needed while interpreting the results of numeric integration. We offer practical ideas and advice on how to use the Taylor Center for teaching ODEs, and to increase the motivation and interest of students.

1 Preface

In this article we deal with *holomorphic*[1] ordinary differential equations (ODEs) and their solutions in accordance with the Theorem of Cauchy-Kovalevskaya on existence and uniqueness of the solution for initial value problems (IVPs).

The goals of the article are

- to introduce an ODE solver, called the Taylor Center for PCs [5] as a tool for teaching and numeric experiments with ODEs (Section 2);
- to illustrate its usefulness by addressing problems from the simple (Section 3) to the more advanced (Sections 4–6);
- to show that caution is always prudent in dealing with and interpretation of computer-generated results (Section 7). In doing this we are continuing the discussion started by Borrelli and Coleman [2].

---

[1]A complex function is called holomorphic in a neighborhood of a point if its Taylor expansion at this point exists and has nonzero convergence radius.

2   Introduction

It seems there have been no sophisticated Taylor Solvers designed for PCs since 1994 (ATOMFT [3, 4]). We expect an application with advanced interactive visualization to provide a user interface with graphics and controls that can be adapted to specific models and operational tasks and realistic visualization of the modeled processes employing all appropriate faculties of the human perception, achievable with advanced hardware and multimedia.

With that in mind we introduce the ODE solver called *The Taylor Center*[2], an all-in-one system for integration of ODEs by applying the modern Taylor Method (Automatic Differentiation), and offering powerful dynamic graphics (including 3D stereo vision with conventional monitors and anaglyph Red/Blue glasses).

The modern Taylor Method is a descendant of its classical counterpart. It is an efficient method for numerical integration of Initial Value Problems (IVPs) for ODEs (presuming that no singularities occur on the integration path). What distinguishes the Taylor Method from all other numerical methods for ODEs is the ability to compute the approximation to the solution with virtually unlimited order of approximation (the number of terms in the Taylor expansion). With no singularities on the integration path, the step *does not* approach zero regardless of how high accuracy is specified (presuming the order of approximation may increase to infinity and the length of mantissa is unlimited). It is the distance to the singularities which bounds the finite integration step. Of course, round-off errors due to machine computation can occur.

An unlimited order of approximation is possible because the method performs *Automatic Differentiation* (i.e., exact computing of the derivatives up to any desired order n by optimized formulas for n-order differentiation) which provides the Taylor series of any desired length for the solution components. Hence, the Taylor Center applies to any explicit first-order system whose right hand sides are *generalized elementary* functions of the state variables, explained below.

Automatic Differentiation is applicable to a subclass of holomorphic ODEs whose right hand sides are the so called generalized elementary vector-functions. The generalized elementary functions widen the class of the traditional (Liouville) elementary functions. The generalization was suggested by R. Moore [8] in the 1960s and further developed by Gofen [6]. Moore defined *generalized elementary functions as those that may be represented as solutions of explicit (nonlinear) ODEs whose right hand sides are rational in all variables*. For brevity, we will call them simply "elementary" functions.

It appears that practically all ODEs used in applications are comprised of (generalized) elementary functions. At the moment, only the Gamma function and Gamma integral are proved to be non-elementary [6]. Other candidates suspected (but not yet proved) to be non-elementary are general solutions of elementary ODEs as functions of their parameters or of their initial values. However, the solutions of elementary ODEs as functions of the variable of integration are always elementary (the theorem of closeness of the class of elementary functions [6]). The composition of elementary functions and their inverse are also elementary [6]. Therefore, almost all problems arising in applications and courses of

---

[2]http://www.ski.org/gofen/TaylorMethod.htm

mathematics and physics taught in universities, can be handled by the Taylor solver.

In the most straightforward way this software may be used to illustrate dynamics in every initial value problem taught in universities, and we will point out many such examples further on. However, the Taylor Center also gives an opportunity to explain and demonstrate particularities of numerical integration. In so doing, it raises the awareness of the students when they interpret the computed numerical solutions. All the examples discussed in the article or illustrated here with a *static* picture may be *animated* via the free demonstration version of the Taylor Center[3].

For every illustration in this article, the path via the Main menu's `Demo` item, or the name of the respective script file for loading via File/Open script is given. (After the installation, the sub-folder `Samples` contains all pre-loaded `scr` files and `ode` files). In this article, all illustrations on a black background are stereoscopic images that should be viewed through red/blue glasses[4]. See Appendix A for the complete list of features of the Taylor Center.

3   A powerful tool for dynamic drawing

The Taylor Center is beneficial whenever the drawing of orbits of ODEs is required. It produces plane or stereoscopic three-dimensional orbits that can be seen using red/blue glasses.

Example 3.1. Here is the "Wine glass" solution (see Figure 1) for the ODEs of autocatalytic reaction (from the book of Borrelli and Coleman [1]) graphed as a planar curve. The IVP is

$$x' = e^{-.002t} - .08x - xy^2, \quad x(0) = 0$$
$$y' = 0.08x - y + xy^2, \qquad y(0) = 0.$$

These ODEs are entered into the four panes of the Taylor Center in this manner:

```
t=0            t'=1
x=0            x'=exp(-0.002*t)-0.08*x-x*y^2
y=0            y'=0.08*x-y+x*y^2
```

This is the general layout of the four panes of the Taylor Center. See Figure 2.

Constants	Auxiliary variables

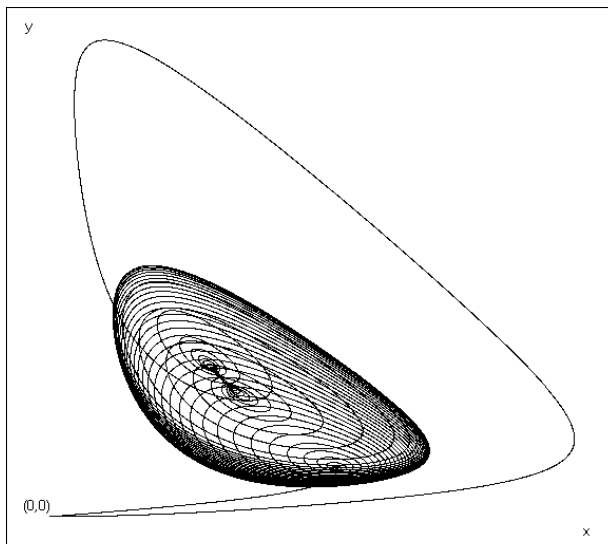| Initial values	ODEs

---

Figure 1: The "Wine glass solution" (script file `Borrelli.scr`)

Note that a trivial equation for the independent variable (like $t' = 1$) may generally be omitted. However it is necessary in this case in order that the "Dynamic Play" feature be possible. Also, it is necessary if the user plans switching from one independent variable to another. The names of the variables may be multi-character identifiers, and the arithmetic expressions must obey the syntax of Pascal (the multiplication sign $*$ cannot be omitted).
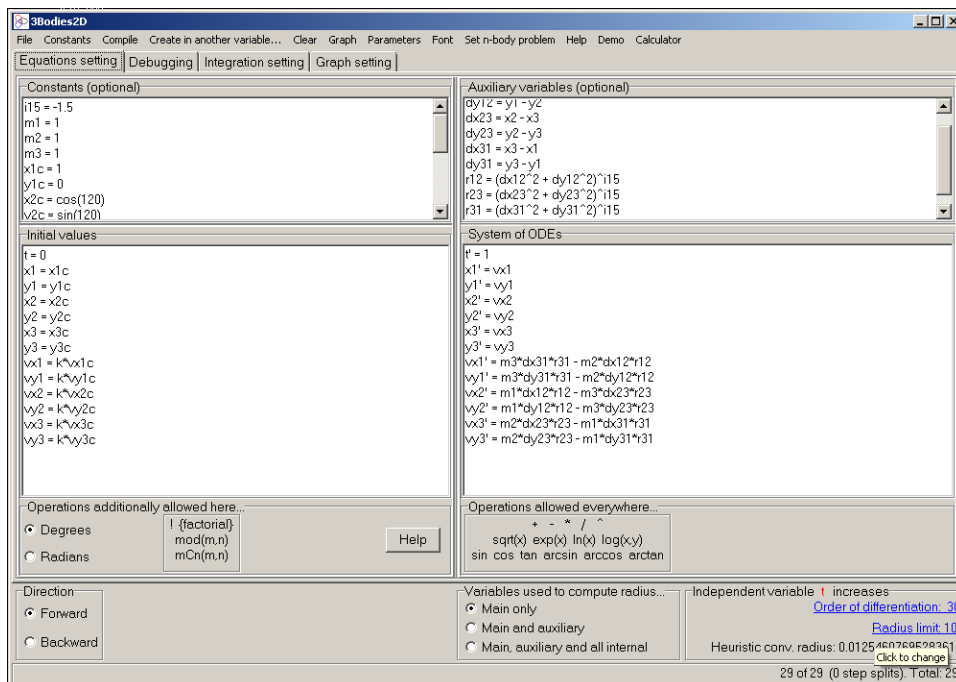

Figure 2: Front panel of the Taylor Center.

The Taylor Center is indispensable for visualizing dynamics. There is a huge difference between a static picture of a trajectory (say in a book) vs. the real time animation of the
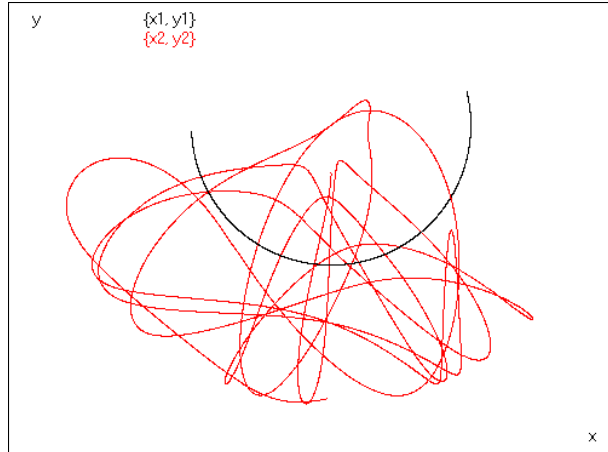
4

Figure 3: Double pendulum (file `DoublePendulum.scr`).

motion along the trajectories in the Taylor Center. In it not only can the viewer watch the near real time evolution of the motion, but also observe its acceleration and decelerations, examine the ODEs, and vary the parameters.

Example 3.2. Figure 3 shows an orbit of the motion of the double pendulum described with the ODEs

$$\theta_1'' = -\frac{L_2\{m_2\cos(\theta_1-\theta_2)[L_1\theta_1'^2\sin(\theta_1-\theta_2) - g\sin\theta_2] + gm\sin\theta_1 + m_2 L_2\theta_2'^2\sin(\theta_1-\theta_2)\}}{L_1 L_2(m_1 + m_2\sin^2(\theta_1-\theta_2))}$$

$$\theta_2'' = \frac{L_1\{gm[\sin\theta_1\cos(\theta_1-\theta_2) - \sin(\theta_1-\theta_2)] + \sin(\theta_1-\theta_2)(m_2 L_2\cos(\theta_1-\theta_2)\theta_2'^2 + mL_1\theta_1'^2)\}}{L_1 L_2(m_1 + m_2\sin^2(\theta_1-\theta_2))}.$$

Here is how the ODEs for the double pendulum are entered in the Taylor Center.

```
g = 9.8                    y1=-L1*cos(Te1)
m1 = 1                     x1=L1*sin(Te1)
m2 = 1                     y2=-L2*cos(Te2)+y1
m = m1+m2                  x2=L2*sin(Te2)+x1
L1 = 1                     Te12=Te1-Te2
L2 = 1                     VTe12=VTe1-VTe2
Te10 = 1.4                 D=-L1*L2*(m1+m2*sin(Te12)^2)
Te20 = -1.4                D1=L2*(m2*cos(Te12)*(L1*VTe1^2*sin(Te12)-g*sin(Te2))+
VTe10 = 0                       +g*m*sin(Te1)+m2*L2*VTe2^2*sin(Te12))
VTe20 = 0.2                D2=-L1*(g*m*(sin(Te1)*cos(Te12)-sin(Te2))+
                                +sin(Te12)*(m2*L2*cos(Te12)*VTe2^2+m*L1*VTe1^2))
```

```
t=0                        t'=1
Te1=Te10                   Te1'=VTe1
Te2=Te20                   Te2'=VTe2
VTe1=VTe10                 VTe1'=D1/D
VTe2=VTe20                 VTe2'=D2/D
```

Example 3.3. Here are ODEs for the planar three-body problem (Figure 4):

```
i15=-1.5
m1=1
m2=1
m3=1
x1c=1
y1c=0
x2c=cos(120)
y2c=sin(120)
x3c=cos(240)
y3c=sin(240)
vx1c=0
vy1c=1
vx2c=cos(210)
vy2c=sin(210)
vx3c=cos(330)
vy3c=sin(330)
k=0.2
```

```
dx12=x1-x2
dy12=y1-y2
dx23=x2-x3
dy23=y2-y3
dx31=x3-x1
dy31=y3-y1
r12=(dx12^2+dy12^2)^i15
r23=(dx23^2+dy23^2)^i15
r31=(dx31^2+dy31^2)^i15
```

```
t=0
x1=x1c
y1=y1c
x2=x2c
y2=y2c
x3=x3c
y3=y3c
vx1=k*vx1c
vy1=k*vy1c
vx2=k*vx2c
vy2=k*vy2c
vx3=k*vx3c
vy3=k*vy3c
```

```
t'=1
x1'=vx1
y1'=vy1
x2'=vx2
y2'=vy2
x3'=vx3
y3'=vy3
vx1'=m3*dx31*r31-m2*dx12*r12
vy1'=m3*dy31*r31-m2*dy12*r12
vx2'=m1*dx12*r12-m3*dx23*r23
vy2'=m1*dy12*r12-m3*dy23*r23
vx3'=m2*dx23*r23-m1*dx31*r31
vy3'=m2*dy23*r23-m1*dy31*r31
```

If this problem were entered as a non-planar system with "z" as a third coordinate and a greater number of variables, the resulting system of ODEs would be much more cumbersome. However, the software generates the equations of n-body problems automatically.

The Taylor Center also visualizes variability of the radius of convergence along the trajectory (the ticks in the progress bar in the lower part of the Figure 4). The direct observation of the dramatic acceleration when the bodies approach the center of the masses is a perfect illustration when teaching Kepler's laws.

The Taylor Center is preloaded with a variety of commonly taught classical problems such as pendulums (planar in files `Pendulum2D.ode`, `DoublePendulum.scr`, and non-planar in files `PendulumFlower.scr`, `PendulumApple.scr`) and n-body problems under different conditions, including the illustration of the Lagrange points (under the `Demo` menu and in the folder `Samples`). Naturally, teachers can add examples of their own interest.
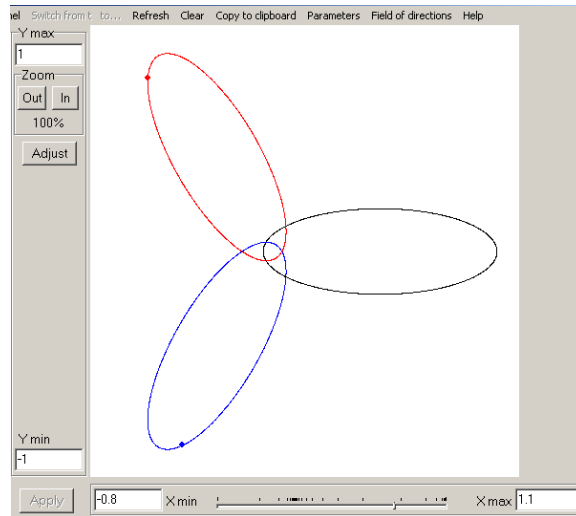
Figure 4: The three-body Lagrange case. (You can find this example by clicking on Demo→3 Bodies→Symmetrical or using the file 3Bodies2D.ode.)

Among mathematical tools, the Taylor Center perhaps is unique in employing stereo vision for displaying non-planar curves. It is always a challenge to draw non-planar trajectories in the conventional axonometric projection. That is why the Taylor Center uses the Red/Blue anaglyph stereo as a cheap, yet efficient system of stereo vision. The curves literally pop up from the screen into the 3D space in front of the viewer, who can turn them and explore them with a 3D cursor controlled with the mouse wheel (the respective 3D coordinates being continuously displayed).

Dynamic playing of trajectories is desirable for almost every ODE problem in mathematics and physics, especially in celestial mechanics. There are numerous pre-loaded script files (in the sub-folder Samples) defining common mechanical problems, such as pendulums, to sophisticated examples in celestial mechanics illustrating the 5 Lagrange (libration) points (Demo→5 Lagrange Points or file LagrangePoints.scr).

In the Taylor Center, we can see the immediate effect of parameters controlling the behavior of the solution. In particular, it is instructive to illustrate instability, for example in the Lagrange solutions of the n-body problem.

Definition 3.4. A solution of the n-body problem is called the Lagrange case if the n masses are equal, and, at the initial moment:

1. The bodies are positioned at the apexes of a regular polygon; and
2. Their initial velocity vectors are equal in absolute value; lie in the plane of the polygon; and
3. Their initial velocity vectors are inclined at the same non-zero angle to the respective radii.

The initial polygonal formation (defined by the properties 1–3) is preserved during the motion, in which all the bodies move along trajectories of the same elliptical type, with the center being the center of the polygon. See, for example, Figure 4.
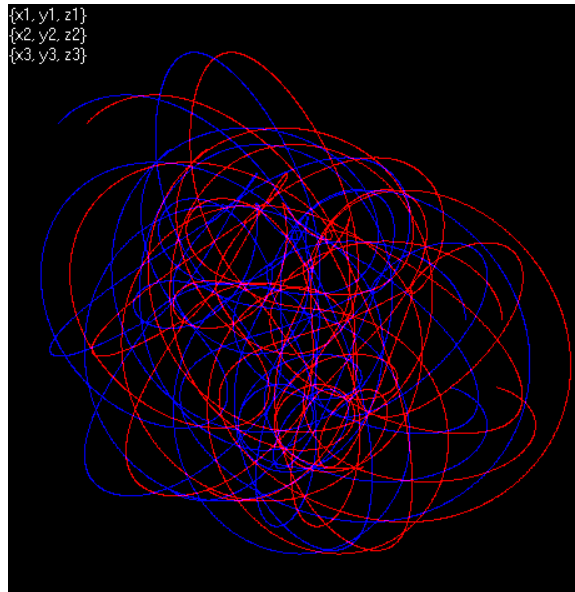
{x1, y1, z1}
{x2, y2, z2}
{x3, y3, z3}

Figure 5: The three-body problem disturbed into non-planar motion (Demo→ 3 Bod-ies→Disturbed→3D or file 3Body3D.scr)

Example 3.5. Finally, let's see what happens when the initial setting in Example 3.3 was disturbed in the direction perpendicular to the plane pushing the trajectories into 3D space. See Figure 5.

You can show students the Euler formation (in line) of 3 bodies (files 3EqBodEuler.scr or 3NonEqBodEuler.scr) and then demonstrate instability of their motion. Or you can generate the Lagrange setting (circular or elliptic) for the n-body problem (n may be up to 99, but for a PC of an average power try numbers, n, not exceeding 20). Here you may explain to the students that the circular Lagrange motion also exemplifies the simplest case of a *Choreography solution*.

Definition 3.6. A solution of the n-body problem is called a *Choreography solution* if all the n bodies move along the same periodic trajectory.

For a long time the circular Lagrange motion was the only known case of Choreography. It was not until 2000 when a non-circular case of choreography was discovered [9]: the amazing 8−shape motion of 3 bodies (Demo→3 Bodies→Choreography or file Simo.scr). Many others have been found since then.

Now ask the students a provocative question if they think the Lagrange case of n-body motion may be *non-planar*: Say if 4 equal mass bodies are placed into the apexes of the regular tetrahedron (or if the n bodies are positioned at the apexes of the other known regular polyhedra called Platonic bodies). Ask them to suggest the directions for the initial velocities and to try them in the Taylor Center: for a tetrahedron, a cube...

(The correct answer is that a non-planar Lagrange motion is impossible except the trivial radial collision case, because in 3D space it is impossible to turn a solid non-planar pencil of rays in such a way, that all the angular increments are equal—see Appendix B.)
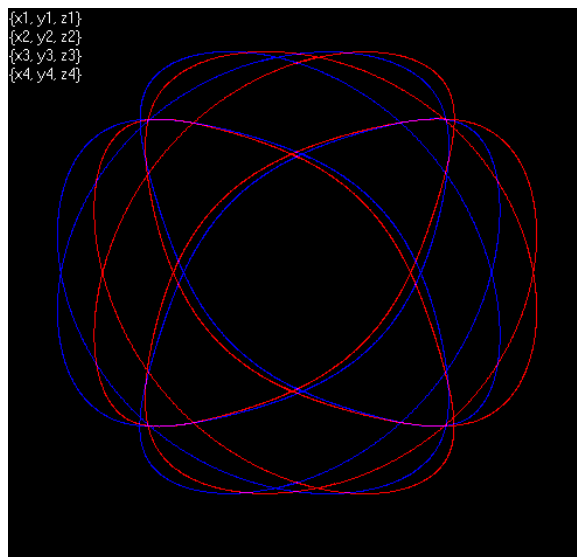
8

Figure 6: The 4-body problem, the trajectories inscribed into a cube (`Demo→ 4 Bod-ies→Cube` or file `4BodiesCubic.scr`)

Finally you may ask the students what their expectations are about the possibility for n equal mass bodies each to move along some cyclic *near planar* orbit so that all the orbits are reciprocally perpendicular planes in 3D space? Then show them the remarkable 4-body orbits inscribed into a cube [7]. See Figure 6.

In the following sections, we will consider ideas where the Taylor Center provides an illustration or a helpful hint beyond mere graphing the solutions.

4 The Taylor Center as a tool for generating the Taylor expansion of the solution

In this section we are going to deal with concepts such as:

- numerical series, Taylor series and their radii of convergence;
- points of singularity of a function at the boundary of its Taylor series' radius of convergence;
- numerical integration of ODEs, the integration step size, its accuracy, and sources of integration errors.

Typically in numerical methods for the integration of ODEs, the input is a vector of initial values along with the name of the subroutine that computes the right-hand side of the differential equations. The output is a table of the values of the solution at a grid with a step small enough to satisfy the accuracy requirements.

Any Taylor solver differs from conventional integrators in that it treats the right-hand side of the differential equation as a symbolic expression, rather than evaluating it numerically at a set of time points.

Correspondingly, the output and interaction of a Taylor solver with other applications have their own peculiarities. The Taylor Center result is not just tabulated values of the solution, but rather the solution's expansion into a Taylor series (an analytical element), or a sequence of such elements.

The possibility of expanding the solution into a Taylor series presumes that the right-hand sides of the ODEs are holomorphic. So the behavior of the solution as a real-valued function is determined by its properties as a complex holomorphic function (while the derivatives in all subsequent *real-valued* ODEs in fact stand for *complex differentiation*)[5]. The Taylor expansion at every point is characterized by its finite (or infinite) radius of convergence equal to the distance of a base point to the nearest point of singularity of the solution (usually unknown). Therefore the Taylor Center operates with the so-called *heuristic convergence radius r_e* (discussed below) always displayed during the process of integration, while the expansions themselves may be viewed at the Debugging page.

Observing the expansion of the solution may be instructive. For example, it would be useful to show students how the expansions differ for the elliptic vs. the circular case of the Lagrange motion. In the elliptic motion example, stop at the slowest and fastest locations of the trajectories to observe how the expansions of the solution dramatically differ, and so does the r_e. The applicable integration step h must always make the fraction $k = h/r_e < 1$, (by default $k = 1/2$).

The Taylor Center can deal only with finite partial sums

$$S_n = \sum_{k=0}^{n} a_k(t - t_0)^k$$

of the Taylor series, whose exact convergence radius (given by the Cauchy-Hadamard formula)

$$R = \frac{1}{\sup |a_k|^{1/k}}$$

usually is unknown. To obtain the *heuristic convergence radius, r_e*, the program uses the Cauchy-Hadamard formula based on the available n terms of the Taylor expansion (by default $n = 30$). The comparison Table 1 contains the heuristic vs. exact values of convergence radii for various types of singularities, demonstrating that the heuristic values reasonably fit the exact radii. (Indeed, the examples of the solutions were chosen in such a way that their points of singularity are available, thus allowing an easy computation of the exact convergence radius).

Now you can ask the students what to expect for solutions which happen to be entire functions, i.e. those which have an infinite convergence radius. Will the program work out the heuristic radius equal to the machine infinity? Does it mean that an arbitrarily large step of integration may be practically applied?

---

[5]In the definition of a derivative of a real valued function at a point, the argument approaches the point along the real axis. In the definition of a derivative of a complex function at a point, the argument approaches the point along any path in the complex plane. Existence of a complex derivative is a much stronger condition than existence of the derivative on the real axis only. However the functions we deal with are holomorphic, which is equivalent to being differentiable at a point and a small neighborhood around it. For these functions, both the real and complex derivatives exist.

Function	Heuristic r_e		Exact R
	$n = 30$	$n = 100$	
$\dfrac{1}{1-t}$	0.951	0.984	1
$\dfrac{1}{(1-t)^2}$	0.803	0.885	1
$\dfrac{1}{(1-t)^3}$	0.692	0.848	1
$\dfrac{1}{1-t^2}$	0.971	0.992	1
$\dfrac{1}{1-t^3}$	0.966	0.991	1
$e^{\frac{1}{1-t}}$	0.696	0.798	1
$\sqrt{1-t}$	1.244	1.110	1
$\ln(1-t)$	1.156	1.058	1
$\tan t$	1.52	1.558	$\dfrac{\pi}{2} \approx 1.57$
$\dfrac{t}{e^t-1}$	5.93	6.133	$2\pi \approx 6.28$

Table 1: Comparison of the heuristic and exact radii in cases of a finite convergence radius.

In fact, the program does work out nearly "infinite" heuristic radius for polynomial solutions, or for certain non-polynomial holomorphic functions. For example, for the system[6]

$$x' = -y^7, \qquad x|_{t=0} = 1$$
$$y' = x^5, \qquad y|_{t=0} = 0$$

(file y7x5_in_t.ode) the Taylor Center computes $r_e = 2.18 \times 10^{1192}$. (To see it, you will have to temporarily change the default radius limit from the value 10 to say 10^{2000}).

Now you can demonstrate to the student an effect of a violent bell-shape growth of the Taylor coefficients (or of the Taylor terms during an attempt to apply a big enough step). In order to do it, enter the trivial ODE $t' = 1$ and the function $x = e^{-100t}$ as an auxiliary variable. Compile it and immediately look into the Debugging page to see the bell-shape growth of the Taylor coefficients particular to functions having an infinite convergence radius.

The function $f(t) = e^{-100t}$ on $[0, +\infty)$ is a typical example of such a function. Its Taylor coefficients are $a_0 = 1$, $a_1 = -100$, $a_2 = 5000$, $a_3 = -16667$, $a_4 = 4.16667 \times 10^6$.

---

[6]Courtesy of Harley Flanders

Function	Heuristic r_e		Exact R
	$n = 30$	$n = 100$	
e^{-t}	5.506	17.4	∞
e^{-10t}	0.505	1.746	∞
e^{-100t}	0.059	0.171	∞
e^{-1000t}	0.0058	0.0174	∞
$e^{-10000t}$	0.00055	0.00170	∞
$\sin t$	8.46	32.15	∞
$\sin 10t$	0.922	3.223	∞
$\sin 100t$	0.0917	0.321	∞
$\sin 1000t$	0.0093	0.0314	∞
$\sin 10000t$	0.00092	0.00321	∞
$\dfrac{\sin t}{t}$	11.753	34.29	∞

Table 2: Comparison of the heuristic radii in cases of an infinite convergence radius.

They violently grow in absolute value, reaching a maximum of $a_{114} = 3.9315 \times 10^{41}$. From that point on, the coefficients decrease in absolute value and decay to zero. For example, $a_{268} = 1.09019$, $a_{269} = -0.405277$, $a_{270} = 0.150103, \ldots, a_{398} = 2.49241 \times 10^{-68}$, $a_{399} = -6.24663 \times 10^{-69}$, $a_{400} = 1.561665 \times 10^{-69}$.

The same bell-shaped pattern may be observed not only for the coefficients a_k proper, but also for the terms $a_k h^k$ in the Taylor expansions $\sum a_k h^k$ of entire functions.

Remark 4.1. The Taylor expansion for every (non-polynomial) entire function $x(t_0 + h) = \sum\limits_{k=0}^{\infty} a_k h^k$ converges for an arbitrary h, with $a_k h^k \to 0$ as a consequence. Yet for any given number k and any large value, M, there exists a step size h, such that the term $|a_k h^k| > M$. (This is obvious for any term for which $a_k \neq 0$, and there must be infinitely many of them in non-polynomial expansions).

Indeed, the integration step h is a finite part of the convergence radius R (whether it is finite or infinite). Moreover, R is usually unknown, so only the available Taylor coefficients allow the determination of the applicable integration step h in order to meet the given error tolerance criteria. The Taylor coefficients allow us to determine the step via the algorithm elaborating the heuristic convergence radius r_e. Table 2 displays values of r_e obtained for various entire functions.

Observe that not only are the heuristic radii obtained by the program *finite* (rather than being "machine infinity"), they tend to get smaller and smaller for those entire functions whose beginning terms in the expansion behave violently and seem to have a bell shaped distribution. Although computed by the Cauchy formula, the heuristic r_e has nothing to do with the actual infinite radius of convergence. Indeed, if the order of the method in the program were specified to be much bigger then the numbers at which the Taylor coefficients disappear, the computed heuristic radius r_e would approach machine infinity.

Otherwise, what the program outputs as r_e is in fact just a numeric characteristic of the available Taylor coefficients, helpful in determining the integration step necessary to achieve the specified accuracy.

Although theoretically the series for entire functions converge for any *arbitrarily large* step,
mous. Also because of the fixed length of the mantissa, the explosive growth of the terms of the series causes a loss of accuracy, and results in the so called *catastrophic subtractive error*. Fortunately, the algorithm in the Taylor Center works out a computationally viable heuristic radius for entire functions, and the number of recalculations necessary to reach the required accuracy is small.

5 Is the highest accuracy of the Taylor method always achievable?

In this section we raise the question of accuracy of numerical methods in general, and of the numerical integration of ODEs in particular. It would be helpful here to remind the students about the concepts of rounding errors in computation, and about the absolute and relative error tolerances. This section focuses on one particular type of error emerging while computing differences of close numbers, called *catastrophic cancellation* or *subtraction* error.

One distinguishing feature of the Taylor Method is its potential to achieve the highest accuracy at a given fixed length floating point representation: the "all-correct-digits" accuracy for the entire integration pass. This potential derives from (but is not guaranteed by) the following two facts.

1. During *one finite step* integration the "all-correct-digits" accuracy is achievable because of an arbitrary high order of approximation in the Taylor method. For the PC the longest native floating point number is represented by a 64 bit mantissa and a 16 bit exponent, resulting in the 80 bit (10 byte) type called *extended* in Delphi. However, the 64th and all subsequent (virtual) digits are assumed to be zero, resulting in *rounding error*. Therefore even after the very first integration step, the "all-correct-digits" result ends up slightly off the original trajectory. The second and all subsequent steps will inevitably cause the computed trajectories move away from the exact trajectories defined by the previous steps. Whether this deviation from the original trajectories increases or not depends on the stability properties of the given ODE.

2. During *multi-step* Taylor integration the number of steps is minimized because the step size is large. The smaller the number of integration steps, the fewer incidents of deviation from the original trajectory due to rounding error. Ideally, to avoid any deviation, we would need to reach the final point in one step.

To specify the "all-correct-digits" accuracy in the Taylor Center, the user has to set the values of the relative error tolerance (for the variables of interest) to something less than 2^{-64} or 10^{-22}.

Besides the inevitable rounding error and possibly intrinsic instability of the ODEs, there is one more effect which may prevent achievement of the "all-correct-digits" accuracy: it is *the catastrophic subtraction* (also called *cancellation*) error. This effect comes up not only for the Taylor method. It may also occur in any computations with floating point binary numbers having the form $m2^n$, where $2^{-1} \le |m| < 1$.

The mantissa m (in the native machine representation proper to the processor) is usually of a fixed length of 64 bits. (For emulated operations it may be 128 binary digits or something bigger, yet it is still fixed). The binary exponent n, however, is allowed to be between -4951 and $+4932$. To accurately represent a sum of two numbers having such extreme values of the exponents, the mantissa ought to have nearly 10000 binary digits.

Because of the limited length of the mantissa, in the Intel generic floating point machine arithmetic

$$1 + 10^{-22} = 1$$

and

$$0.1234567890123456789 - 0.1234567890123456788 = 0$$

Therefore, the summation of a Taylor expansion having terms that grow very quickly may cause a loss of accuracy, similar to the loss of accuracy that occurs when finding the difference of two very close numbers.

To clarify the concept of catastrophic subtraction error and its fundamental distinction from the rounding error, consider the following two seemingly similar expressions for some differentiable function $f(x)$ when $h \to 0$:

Catastrophic subtraction error	Rounding error
$\lim \dfrac{f(x_0 + h) - f(x_0)}{h}$	$\lim \dfrac{f(x_0 + h) + f(x_0)}{2}$

The first one inevitably becomes machine zero for small enough h. It therefore never approaches the derivative $f'(x_0)$ if $f'(x_0) \ne 0$, and may differ from it dramatically. The second one always approaches $f(x_0)$ and may differ from it only in the 64th binary digit.

In ODE solvers the catastrophic subtraction error emerges when the integration pass approaches a point of singularity of the ODE, or another special point. Such examples are provided in Chapter 5 "Tricks and Traps of Automatic Differentiation" in the Taylor Center User Manual[7].

6 Regular solutions of singular ODEs

This section provides grounds for a discussion about points of singularity in ODEs and their solutions, illustrated in the pre-loaded examples (see the Taylor Center User Manual).

First, advise the students about the distinction between points of singularity of the *solution* vs. points of singularity of the *ODEs* in their phase space. For example, the meromorphic functions $x = 1/(1-t)$ and $y = \tan t$ have singularities at $t = 1$ and $t = \pi/2$,

---

[7] http://www.ski.org/gofen/TaylorUserManual.doc

respectively. However, each of them satisfies a polynomial ODE ($x' = -x^2$ and $y' = y^2 + 1$), for which every finite point of the phase space is regular.

And vice versa, the right hand sides of ODEs may happen to be singular at particular points of the phase space. For example the ODE $x' = ax/t$ is singular when $t = 0$. Yet its solution $x = t^a$ is regular for any natural (non-negative integer) parameter a. This solution also happens to satisfy a regular ODE $x' = at^{a-1}$.

However, there exist regular solutions (in fact entire functions) which at certain points can satisfy no explicit polynomial ODE, nor any rational ODE with non-zero denominator [6]. For example, such functions are $x = (e^t - 1)/t$, $x(0) = 1$, or $y = (\sin t)/t$, $y(0) = 1$, or $z = \cos \sqrt{t}$. All the above mentioned functions are entire (holomorphic at all finite points). The problem at the point $t = 0$ for these functions is that the explicit Taylor method (i.e. the explicit formulas for the evaluation of the derivatives implied by the Taylor method) is not applicable at this special point. The elementariness of these functions is possibly violated at this point [6]. Therefore some other—implicit ODEs and implicit formulas—must be used at such points, generating the respective Taylor expansions. For many classical functions these expansions at the special point are known.

A feature of the recent version of the Taylor Center is that it can even integrate some ODEs near their points of singularity, provided that the Taylor expansions at the special points are known from other sources. Some examples of this feature are pre-loaded and their descriptions provided in the Taylor Center User Manual.

The following section may serve as a good illustration of the statement in Borrelli and Coleman [2]:

> Using a numerical solver to produce an approximate solution of an initial value problem is not a mindless operation; It is not merely inserting an IVP into a package solver and out pops a decent approximate solution.

7 Properties of weird examples of real-valued solutions

We expect that even a simple looking ODE, e.g with a polynomial or rational function on the right-hand side, may have a solution that is not "simple" at all. The following ODE, however, exemplifies the opposite situation.

Example 7.1. Consider the initial value problem:

$$x' = -\sqrt{x}, \qquad x|_{t=0} = 1.$$

In the neighborhood of $t = 0$, its solution is a polynomial $x = (t - 2)^2/4$. Indeed, the solution exists and may be continued analytically to all points of the complex plane, but the right-hand side of the ODE, $-\sqrt{x}$, cannot. Here the right-hand side is deliberately chosen to be the negative branch of the 2-branched function $\sqrt{x}$. Therefore, this ODE may be satisfied only by the *decreasing* portion of the parabola $x = (t - 2)^2/4$. This example illustrates the strange effects that may be encountered during a naïve attempt to integrate an ODE of this type. (See Figure 7).
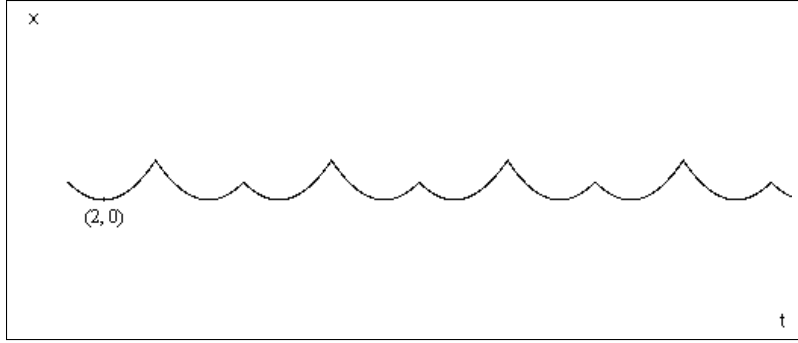
Figure 7: A naïve (and incorrect) attempt to integrate $x' = -\sqrt{x}$ (file `strange.ode`). This piecewise curve (instead of a parabola) is an artifact of the program.

The other remarkable example[8] is the function $x = \cos\sqrt{t}$, also pre-loaded and integrable in the Taylor Center: both for negative and positive t. Indeed, this version of the Taylor Center cannot deal with complex variables directly. Observe that even though $\sqrt{t}$ becomes imaginary for $t < 0$, the cosine of purely imaginary values is real (and bigger than 1 because $\cos it = \cosh t$).

It is worth noting a fundamental fact of automatic differentiation [6], that ODEs whose right hand sides contain non-rational elementary functions (transcendental or algebraic) may be equivalently transformed into larger systems of ODEs whose right hand sides are rational only. The function $\cos\sqrt{t}$ is elementary, except at the point $t = 0$. Hence the problem may be treated as though it has real-valued variables as soon as we find a rational ODE satisfied by this elementary function. There exists a general way to eliminate non-rational functions in ODEs [6]. In this particular case, differentiate x twice:

$$x' = -\frac{\sin\sqrt{t}}{2\sqrt{t}};$$

$$x'' = -\frac{(\cos\sqrt{t})\dfrac{2\sqrt{t}}{2\sqrt{t}} - (\sin\sqrt{t})\dfrac{1}{\sqrt{t}}}{4t} = -\frac{x + 2x'}{4t},$$

obtaining the required rational ODE:

$$x'' = -\frac{x + 2x'}{4t}, \qquad x|_{t=0} = 1, \qquad x'|_{t=0} = -\frac{1}{2}.$$

This ODE still has $t = 0$ as a point of singularity, because this is the *unremovable* or "*regular*" singularity of the function $x(t)$ [6]. The Taylor expansion of this ODE at $t = 0$ is easily available and pre-loaded into the Taylor Center, so that the program can integrate this ODE and graph the solution in both directions. See Figure 8.

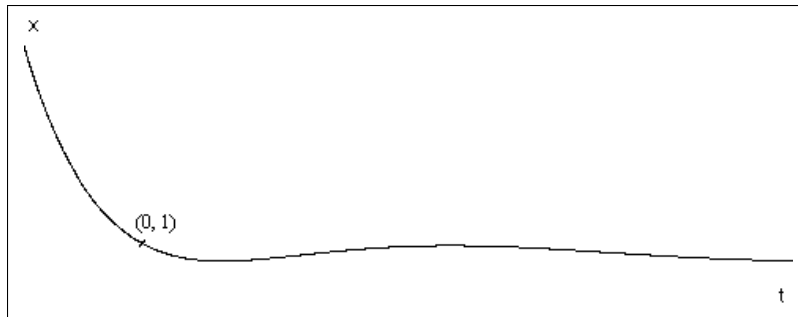[8]Courtesy of George Bergman

Figure 8: Function $x = \cos \sqrt{t}$ (file `SpecPoints\cos(sqrt(t))-2.ode`)

8   Conclusions

The Taylor Center may assist in teaching elementary and advanced ODEs. Moreover, it could also be useful in other mathematical fields such as complex analysis, general and celestial mechanics, numerical methods, and Automatic Differentiation (in the context of the Unifying View [6]).

Perhaps the Taylor Center's most obvious benefits to the teacher are the sophisticated dynamic graphics which teachers can easily use to illustrate examples of ODEs and examine the effects of system parameters. This article describes several such examples appropriate for a basic course in ODEs. Teachers can also enter their own examples.

Possible applications of the Taylor Center are not limited to basic ODEs. It can also be used to illustrate various ideas in advanced courses, as was shown in this article. Additionally, the Taylor Center is indispensable for numerical experiments in research and advanced student projects because of its potential to achieve the all-correct-digits accuracy in numerical integration, to obtain the roots of solutions to ODEs, to explore the radii of convergence and expansions of these solutions, and to graphically display the dynamics in real time.

A   The full list of features of the Taylor Center

With the current version of the Taylor Center you can:

- Specify and study any Initial Value Problem presented as a system of explicit first order *elementary* ODEs (the standard format) with numeric and symbolic constants and parameters;
- Perform numerical integration of IVPs with an arbitrary high accuracy along a path without singularities, while the step of integration remains finite and does not approach zero (under the assumption that the order of approximation or the number of terms may increase to infinity, and the length of the mantissa is unlimited);
- Apply an arbitrarily high order of approximation (by default 30), and obtain the solution in terms of analytical elements - Taylor expansions covering the required domain;
- Study Taylor expansions and the radius of convergence for the solution at all points of interest up to any high order (the terms in the series must not exceed the maximum

17

value of about 10^{4932} implied by the Intel processor generic implementation of the *extended* real type as 10-bytes with a 63-bit mantissa);

- Perform integration either "blindly" (observing only the numerical changes), or with graphical visualization. Perform integration for a given number of steps, or until an independent variable reaches the terminal value, or until a (former) *dependent* variable (now a new independent variable) reaches a terminal value (as explained in the next item);
- Switch integration between several states of ODEs defining the same trajectory, but with respect to different independent variables. For example, it is possible to switch the variable of integration from t to x or y, to reach the terminal value of a former dependent variable (x or y). In particular, if the initial value is nonzero and the terminal value is set to zero, the root (the zero) of the solution may be obtained directly without iterations.
- Integrate piecewise-analytical ODEs;
- Specify different methods to control the accuracy and the step size;
- Specify accuracy for individual components either as an absolute or relative error tolerance, or both;
- Graph curves (trajectories) in color for any pair of variables of the solution (up to 99) on one screen - either as plane projections, or as 3D stereo images (for triplets of variables) to be viewed through anaglyphic (Red/Blue) glasses. The 3D cursor with audio feedback (controlled by a conventional mouse) enables "tactile" exploration of the curves virtually "hanging in thin air";
- Play dynamically the near-real time motion along the computed trajectories either as 2D or 3D stereo animation of moving bullets;
- Graph the field of directions - actually the field of curvy segments, whose length is proportional to the radius of convergence.
- Explore examples such as the problem of Three and Four Bodies supplied with the package. Symbolic constants and expressions allow parameterization of the equations and initial values to try different initial configurations of special interest.
- Automatically generate ODEs for the classical Newtonian n-body problem for n up to 99 and then integrate and explore the motion. For $n = 99$, there are 298 ODEs, 19404 auxiliary equations, compiled into over 132000 variables and over 130000 AD processor's instructions: a "heavy duty" integration!
- Integrate a few special instances of singular ODEs having regular solutions near the points of the "*regular singularities*" [6].

In particular, the demo version of this software comes with numerous instructive ODEs including the Choreography for the Three Body motion, a figure eight orbit discovered in 2000 by Chenciner and Montgomery [9]. Users can "feed" the ODEs of their interest into the Taylor Center, integrate them, draw the curves, and play the motion in the real-time mode all in the same program. Another example describes the four body non-planar trajectories inscribed in a cube [7].

The Taylor Center is a 32-bit software which runs under both 32- and 64-bit Windows (up to Windows 7). The executable module is only 1 Mb. As a 32-bit application, the program can use no more than 4 Gb of available memory for variables and their

expansions—the limit far exceeding any practical needs—see the memory requirements below. (When the new 64-bit Delphi compiler became available, the project is recompiled into a 64-bit application doing away with this 4 Gb limitation).

The memory consumption depends on the number of variables `VarNum` (a function of the number of ODEs and their complexity) and on the specified `Order` of approximation. If the expansions are not stored, the program takes 2×VarNum×Order×10 bytes of memory. If the expansions in P points are stored, it additionally requires P×VarNum×Order×10 bytes.

A benchmark for the 10 body planar problem comprised of 41 ODEs, $45 + 90 = 135$ auxiliary equations, parsed into 811 AD instructions, takes 38 seconds for 10000 steps of integration (or 3.8 ms per step) at 2.4 GHz Pentium.

Generally for each system of ODEs there is a value of the accuracy tolerance small enough so that, at this high accuracy, the Taylor method beats any fixed order method due to the unlimited order of approximation in the Taylor method.

B   Impossibility of the non-planar Lagrange motion

Consider the earlier given definition of the Lagrange case for the n-body problem hypothetically applied to one of the five regular polyhedra in 3D. Let us call it a *non-planar Lagrange case*.

Lemma B.1. *If a non-planar Lagrange case with the masses at the apexes of a regular polyhedron is possible, the pencil of rays from the center to the apexes in this regular polyhedron must be rotatable in such a way that the angular increments are equal for all the rays.*

Proof. Absolute values of the initial velocities must be equal and inclined at the same angle to the radii at all the apexes. Therefore, they must cause equal angular increments.   □

The issue of the possibility of a non-planar Lagrange motion therefore relies on a purely geometric question: whether a solid non-planar pencil of rays is rotatable preserving all its angular increments (no regularity of the pencil is now assumed). A *planar* solid pencil of rays (Figure 9) *is* definitely rotatable this way in its plane.

Now consider an arbitrary non-planar pencil of rays rotating in 3D around the point of apex. According to Euler's Theorem, the motion of a rigid body about a fixed point is equivalent to the rotation of the body about an instantaneous axis of rotation. Let this instantaneous axis be OQ (Figure 10).

Lemma B.2. *During rotation around an axis of a pencil of rays, a ray with a bigger angle to the axis has a bigger rotational angular increment.*

Proof. Let the ray OA be at angle α to the axis OQ, the ray OB be at angle β to the axis, and $\alpha > \beta$. Let the pencil of rays turn around the axis at angle γ so that $m\angle AQA' = m\angle BQB' = \gamma$, while $CC' = BB'$. Observe that
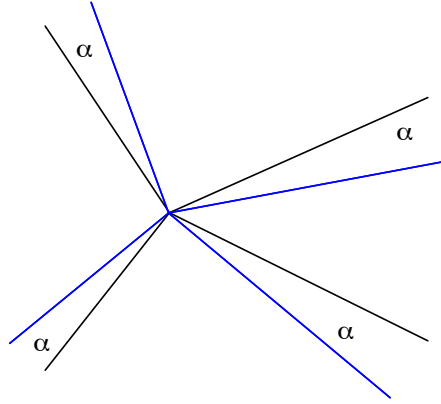
$$\frac{CO}{CQ} > \frac{AO}{AQ}$$

19

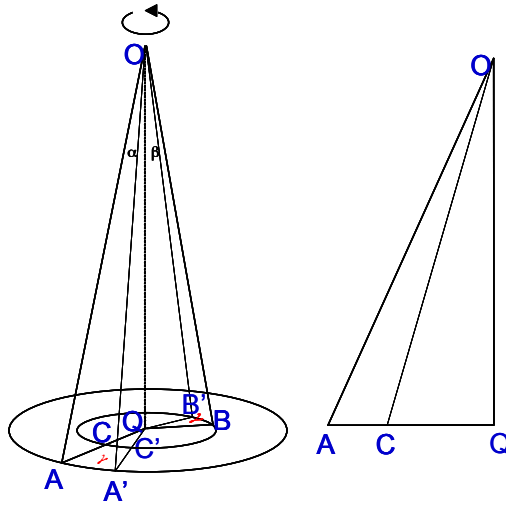Figure 9: A solid pencil of rays in a plane turned at angle α.

Figure 10: Rotation of a solid pencil of rays in 3D.

because $m\angle OCQ > m\angle OAQ$. Therefore

$$\frac{CO}{AO} > \frac{CQ}{AQ} = \frac{CC'}{AA'}$$

(due to similarity of $\triangle CQC'$ and $\triangle AQA'$), and

$$\frac{CC'}{CO} < \frac{AA'}{AO}$$

meaning that $m\angle BOB' < m\angle AOA'$. □

Conclusion 1. A rotation of a non-planar pencil of rays preserving all the angular increments in the 3D space is possible if and only if the rays belong to the surface of a direct circular cone.

Proof. Follows from the Lemma. □

20

Conclusion 1 may be also proved using analytical geometry. Denote the rays as vectors $\mathbf{a}_k$, which rotate around an unknown axis $\mathbf{x}$. In order that axis $\mathbf{x}$ be inclined to each of the vectors $\mathbf{a}_k$ at the same angle, all the cosines expressed through the scalar product must be equal to the same unknown value α:

$$\frac{(\mathbf{a}_k, \mathbf{x})}{|\mathbf{a}_k| \cdot |\mathbf{x}|} = \alpha. \tag{B.1}$$

If $\alpha \neq 0$, this non-linear system expresses the fact that all vectors $\mathbf{a}_k$ belong to a direct circular cone. Otherwise if $\alpha = 0$, (B.1) means that the axis $\mathbf{x}$ must be perpendicular to all vectors $\mathbf{a}_k$, which is possible only if $\mathbf{x}$ belongs to a space of dimension 4 or higher.

In fact, the pencil of rays from the center to the apexes of the polyhedra does not belong to a direct circular conic surface in any of the regular Platonic polyhedra. Therefore, there exists no way to orient the vectors of the initial velocity at the apexes of the regular polyhedra so that the bodies preserve the initial formation during the motion unless all these vectors are collinear with the respective radii. If this is the case, the motion takes place along the radii, and it either ends up with a collision, or the bodies escape to infinity.

References

[1] R. Borrelli and C. Coleman. *Differential Equations: A Modeling Perspective.* Wiley, New York, 2004.

[2] R. Borrelli and C. Coleman. Pitfalls and pluses in using numerical software to solve differential equations. *CODEE Journal*, February 2009. URL http://www.codee.org/ref/CJ09-0157.

[3] Y. F. Chang and G. Corliss. ATOMFT: Solving ODEs and DAE using Taylor series. *Computers Math. Applic.*, 28(10-12):209–233, 1994.

[4] G. Corliss and Y. F. Chang. Solving ordinary differential equations using Taylor series. *ACM Transactions on Mathematical Software*, 8(2):114–144, 1982.

[5] A. Gofen. Interactive environment for the Taylor integration (in 3D stereo). *Proceedings of the 2005 International Conference on Scientific Computing (CSC 05)*, pages 67–73, 2005.

[6] A. Gofen. The ordinary differential equations and automatic differentiation unified. *Complex Variables and Elliptic Equations*, 54(9):825–854, September 2009.

[7] C. Moore and M. Nauenberg. New periodic orbits for the n-body problem. *J. Comput. Nonlinear Dynam.*, 1(4):307, October 2006.

[8] R. E. Moore. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, N.Y., 1996.

[9] C. Simò. Dynamical properties of the figure eight solution of the three-body problem, 2001. URL http://www.maia.ub.es/dsg/2001/0104simo.txt,http://www.maia.ub.es/dsg/3body.html.