Power Series and ODEs
Obsolete? Or cutting edge?
JMM 2025 in Seattle

Roger Thelwell & and the PSM group*

* J.S. Sochacki, G.E. Parker, D.C. Carothers, S.K. Lucas, J.D. Rudmin, A. Tongen, D.A. and P.G. Warne, S. Colafranceschi (EMU), R.D. Neidinger (Davidson College)

James Madison University

January 5, 2025

Outline

1. Preface
2. IVODEs and exact solutions
   - Ex 1: $y' = y(1-y)$
   - Ex 2: $y' = Ky^{\alpha}$
   - Ex 3: $y" = -\sin(y)$ and ODEPSM
   - Ex 4: Three Body
   - Ex 5: $y' = y^2 - y^3$
3. Conclusions

Abstract

Isaac Newton's pioneering work with infinite series laid the framework for solving and analyzing differential equations. Newton utilized series to approximate functions to solve complex problems, demonstrating their utility in representing and manipulating mathematical expressions that describe physical phenomena. But series methods are all too often considered historic and unwieldy, while the numerical methods to which they gave rise are viewed as the future.

Though simple automatic transformations, highly nonlinear ODE systems can be easily analyzed and efficiently solved via power series. ...We will show that series implementations are often a magnitude of order faster than Matlab's ode45 in run time, more accurate than ode89 and ode15s, avoid interpolation error, and allow powerful symbolic analysis. Best of all, the technique and theory is accessible to all with rudimentary coding skill and knowledge of Calculus II.

(Young) Newton

By 1665, Newton had found a way to expand

$$(a+b)^{m/n}$$

as a infinite series – the generalized binomial expansion (pg 168-171 Correspondence of Isaac Newton, Turnbull (ed))

Orginally presented as a mechanism to compute the area under a specific curve ("lines wch could be squared"), Newton recognized the immense power of "reducing" expressions to infinite series, which then allowed him access to a wide range of algebraic techinques.

(Young) Newton

By 1665, Newton had found a way to expand

$$(a + b)^{m/n}$$

as a infinite series – the generalized binomial expansion (pg 168-171 Correspondence of Isaac Newton, Turnbull (ed))

Orginally presented as a mechanism to compute the area under a specific curve ("lines wch could be squared"), Newton recognized the immense power of "reducing" expressions to infinite series, which then allowed him access to a wide range of algebraic techinques.

The reduction of difficult functions to "simple" infinite series, when combined with the "method of fluxions," fueled the birth of calculus.

"By their help, analysis reaches, I might almost say, to all problems." (Newton, *De Analys*)

Example 1: $y' = y(1-y)$

Consider the inital value ODE:

$$\frac{d}{dx}y = y(1-y), \quad y(0) = y_0$$

the logistic equation.
I usually ask my students what they think the solution of

$$\frac{d}{dx}y = y(1-y), \quad y(0) = y_0$$

might look like.

Example 1: $y' = y(1-y)$

Consider the inital value ODE:

$$\frac{d}{dx}y = y(1-y), \quad y(0) = y_0$$

the logistic equation.
I usually ask my students what they think the solution of

$$\frac{d}{dx}y = y(1-y), \quad y(0) = y_0$$

might look like.

Phase portraits?
Integration??

Exact solution

They often try to solve $\frac{d}{dx}y = y(1-y)$ by integrating directly.

$$\begin{aligned}
\frac{d}{dx}y &= y(1-y) \\
\frac{1}{y(1-y)}\frac{d}{dx}y &= 1 \\
\int \frac{1}{y(1-y)}\,dy &= \int 1\,dy
\end{aligned}$$

Exact solution

Lets solve $\frac{d}{dx}y = y(1-y)$ by integrating directly.

$$\frac{d}{dx}y = y(1-y)$$

$$\frac{1}{y(1-y)}\frac{d}{dx}y = 1$$

$$\int \frac{1}{y(1-y)}\,dy = \int 1\,dy$$

$$\int \frac{A}{y} + \frac{B}{1-y}\,dy = \int 1\,dy$$

Exact solution

Lets solve $\frac{d}{dx}y = y(1-y)$ by integrating directly.

$$\begin{aligned} \frac{d}{dx}y &= y(1-y) \\ \frac{1}{y(1-y)}\frac{d}{dx}y &= 1 \\ \int \frac{1}{y(1-y)}\,dy &= \int 1\,dy \\ \int \frac{1}{y} + \frac{1}{1-y}\,dy &= \int 1\,dy \\ \ln(y) - \ln(1-y) &= x + C, \end{aligned}$$

Exact solution

Lets solve $\frac{d}{dx}y = y(1-y)$ by integrating directly.

$$\frac{d}{dx}y = y(1-y)$$

$$\frac{1}{y(1-y)}\frac{d}{dx}y = 1$$

$$\int \frac{1}{y(1-y)}\,dy = \int 1\,dy$$

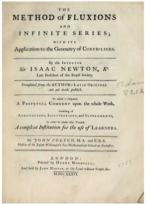
$$\int \frac{1}{y} + \frac{1}{1-y}\,dy = \int 1\,dy$$

$$\ln(y) - \ln(1-y) = x + C,$$

$$\text{so } \frac{y}{1-y} = \exp(x+C) = K\exp(x)$$

$$y(x) = \frac{1}{1+K\exp(-x)} \text{ with } K = \frac{1-y_0}{y_0}$$

aside: Newton?



THE
METHOD of FLUXIONS
AND
INFINITE SERIES;
WITH ITS
Application to the Geometry of CURVE-LINES.

By the INVENTOR
Sir ISAAC NEWTON, Kt.
Late President of the Royal Society.

Translated from the AUTHOR's LATIN ORIGINAL
not yet made publick.

To which is subjoined,
A PERPETUAL COMMENT upon the whole Work,
Consisting of
ANNOTATIONS, ILLUSTRATIONS, and SUPPLEMENTS,
In order to make this Treatise
A compleat Institution for the use of LEARNERS.

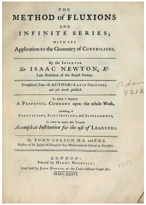
By JOHN COLSON, M.A. and F.R.S.
Master of Sir Joseph Williamson's free Mathematical-School at Rochester.

LONDON:
Printed by HENRY WOODFALL;
And Sold by JOHN NOURSE, at the Lamb without Temple-Bar.
M.DCC.XXXVI.

"... all kinds of complicated terms ... may be reduced to the class of simple quantities, i.e., to an infinite series of fractions whose numerators and denominators are simple terms, which will thus be freed from those difficulties that in their original form seem'd almost insuperable." (pg n21, Isaac Newton, Derek Whiteside, ed:)

Newton had realized that algebra on (convegent) power series is easy... add, subtract, differentiate, and integrate term-by-term!

aside: Newton?


THE
METHOD of FLUXIONS
AND
INFINITE SERIES;
WITH ITS
Application to the Geometry of CURVE-LINES.

By the INVENTOR
Sir ISAAC NEWTON, *K^t.*
Late President of the Royal Society.

Translated from the AUTHOR's LATIN ORIGINAL
not yet made publick.

To which is subjoin'd,
A PERPETUAL COMMENT upon the whole Work,
Consisting of
ANNOTATIONS, ILLUSTRATIONS, and SUPPLEMENTS,
In order to make this Treatise
A compleat Institution for the use of LEARNERS.

By *JOHN COLSON*, M.A. and F.R.S.
Master of Sir *Joseph Williamson's* free Mathematical-School at *Rochester*.

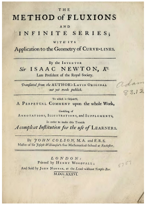
LONDON:
Printed by HENRY WOODFALL;
And Sold by JOHN NOURSE, at the *Lamb* without *Temple-Bar*.
M.DCC.XXXVI.

"... all kinds of complicated terms ... may be reduced to the class of simple quantities, i.e., to an infinite series of fractions whose numerators and denominators are simple terms, which will thus be freed from those difficulties that in their original form seem'd almost insuperable." (pg n21, Isaac Newton, Derek Whiteside, ed:)

Newton had realized that algebra on (convegent) power series is easy... add, subtract, differentiate, and integrate term-by-term! Differentiate?

If $A = \sum_{n=0} a_n x^n$, what is A'?

aside: Newton?


THE
METHOD of FLUXIONS
AND
INFINITE SERIES;
WITH ITS
Application to the Geometry of CURVE-LINES.

By the INVENTOR
Sir ISAAC NEWTON, Kt.
Late Prefident of the Royal Society.

Tranflated from the AUTHOR's LATIN ORIGINAL
not yet made publick.

To which is fubjoin'd,
A PERPETUAL COMMENT upon the whole Work,
Confifting of
ANNOTATIONS, ILLUSTRATIONS, and SUPPLEMENTS,
In order to make this Treatife
A compleat Inftitution for the ufe of LEARNERS.

By JOHN COLSON, M.A. and F.R.S.
Mafter of Sir Jofeph Williamfon's free Mathematical-School at Rochefter.

LONDON:
Printed by HENRY WOODFALL;
And Sold by JOHN NOURSE, at the Lamb without Temple-Bar.
M.DCC.XXXVI.

"... all kinds of complicated terms ... may be reduced to the class of simple quantities, i.e., to an infinite series of fractions whose numerators and denominators are simple terms, which will thus be freed from those difficulties that in their original form seem'd almost insuperable." (pg n21, Isaac Newton, Derek Whiteside, ed:)

Newton had realized that algebra on (convegent) power series is easy... add, subtract, differentiate, and integrate term-by-term! Differentiate?

If $A = \sum_{n=0} a_n x^n$, what is A'? Just write it out...

$$A = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots$$

$$A' = \qquad a_1 + 2a_2 x + 3a_3 x^2 + \dots$$

$$A' = \sum_{j=1}^{\infty} j \cdot a_j x^{j-1} = \sum_{n=0}^{\infty} (n+1) a_{n+1} x^n$$

aside: manipulating power series

Multiplication? If $A = \sum_{n=0} a_n x^n$ and $B = \sum_{n=0} b_n x^n$, what is $A \cdot B$?

aside: manipulating power series

Multiplication? If $A = \sum_{n=0} a_n x^n$ and $B = \sum_{n=0} b_n x^n$, what is $A \cdot B$?

The Cauchy Product:

$$\begin{aligned}(a_0+a_1x + a_2x^2 + a_3x^3 + \ldots)\cdot(b_0 + b_1x + b_2x^2 + b_3x^3 + \ldots) \\ = (a_0 + b_0) + (a_0b_1 + a_1b_0)x + (a_0b_2 + a_1b_1 + a_2b_0)x^2 \ldots \\ + (a_0b_3 + a_1b_2 + a_2b_1 + a_3b_0)x^3 + \ldots \\ = \sum_{n=0}\left[\sum_{i=0}^{n} a_i b_{n-i}\right] x^n,\end{aligned}$$

with the n^{th} component of $A \cdot B$ given by $\sum_{i=0}^{n} a_i b_{n-i}$.

Ex 1 (again): $y' = y(1-y)$

We have the tools, so let's "solve" $\frac{d}{dx}y = y(1-y)$ using power series. Assume

$$y(x) = \sum_{n=0}^{\infty} y_n x^n.$$

Substituting,

$$\sum_{n=0}^{\infty}(n+1)y_{n+1}x^n = \left(\sum_{n=0}^{\infty} y_n x^n\right) \cdot \left(1 - \sum_{n=0}^{\infty} y_n x^n\right)$$

aside: MATLAB

In MATLAB,

$$\sum_{i=0}^{n} a_i b_{n-i}$$

is

```
function cn = cauchy_product(avec,bvec,degree)
 % find nth degree coefficient of product a*b = c
cn = 0;
for i = 1:degree+1
    j = degree-i+1;
    cn = cn + avec(i)*bvec(j);
end
```

aside: Matlab

In Matlab,

$$\sum_{i=0}^{n} a_i b_{n-i}$$

is

```
function cn = cauchy_product(avec,bvec,degree)
 % find nth degree coefficient of product a*b = c
cn = 0;
for i = 1:degree+1
    j = degree-i+1;
    cn = cn + avec(i)*bvec(j);
end
```

or even simpler: `avec(1:n+1)*bvec(n+1:-1:1).'`

Putting it together...

We now have:

$$\begin{aligned}\sum_{n=0}^{\infty}(n+1)y_{n+1}x^n &= \left(\sum_{n=0}^{\infty} y_n x^n\right)\cdot\left(1-\sum_{n=0}^{\infty} y_n x^n\right)\\ &= \sum_{n=0}^{\infty} y_n x^n - \sum_{n=0}^{\infty}\sum_{i=0}^{n} y_i y_{n-i} x^n)\\ &= \sum_{n=0}^{\infty}(y_n - \underbrace{\sum_{i=0}^{n} y_i y_{n-i}}_{c_n})x^n\end{aligned}$$

where c_n is the result of the Cauchy product $y \cdot y$.

Putting it together...

We now have:

$$\begin{aligned}\sum_{n=0}^{\infty}(n+1)y_{n+1}x^n &= \left(\sum_{n=0}^{\infty} y_n x^n\right)\cdot\left(1-\sum_{n=0}^{\infty} y_n x^n\right)\\ &= \sum_{n=0}^{\infty} y_n x^n - \sum_{n=0}^{\infty}\sum_{i=0}^{n} y_i y_{n-i} x^n)\\ &= \sum_{n=0}^{\infty}(y_n - \underbrace{\sum_{i=0}^{n} y_i y_{n-i}}_{c_n})x^n\end{aligned}$$

where c_n is the result of the Cauchy product $y \cdot y$. Equating powers of x^n allows us to recover a simple recurrence relation:

$$(n+1)y_{n+1} = y_n - \sum_{i=0}^{n} y_i y_{n-i}$$

and it easy to find y_{n+1} in terms of lower order coefficients...

MATLAB?

And MATLAB makes it easy. If $y(0) = y_0$, we have

$$y_{n+1} = \frac{1}{n+1}\left(y_n - \sum_{i=0}^{n} y_i y_{n-i}\right)$$

```
function y = solve_logistic(y0,degree)
y(1) = y0;
for n =1:degree
   c(n) = cauchy_product(y,y,n);
   y(n+1) = 1/(n+1) * ( y(n) - c(n) );
 end
```

Example 1: $y' = Ky^\alpha$

Consider the IVODE

$$y' = Ky^\alpha, \quad y(x_0 = 0) = y_0$$

- Why?

Example 1: $y' = Ky^{\alpha}$

Consider the IVODE

$$y' = Ky^{\alpha}, \quad y(x_0 = 0) = y_0$$

- Why?
- Because we have an analytic solution!

$$y(x) = \left(\left(Kx - K\alpha x + {y_0}^{1-\alpha} \right)^{(\alpha-1)^{-1}} \right)^{-1}$$

Example 2: $y' = Ky^{\alpha}$

First represent $y(x) = \sum_{n=0}^{\infty} y_n (x - x_0)^n$,.
Since

$$y'(x) = \sum_{n=0}^{\infty} (n+1)\, y_{n+1} (x - x_0)^n = \sum_{n=0}^{\infty} (n+1)\, y_{n+1}\, x^n,$$

We would have a simple recursion to recover coefficients y_n if we can compute the a_n coefficients for

$$y^{\alpha} = \sum_{n=0}^{\infty} a_n (x - x_0)^n = \sum_{n=0}^{\infty} a_n\, x^n,$$

For example:

$$(n+1) \cdot y_{n+1} = K a_n \quad \text{means} \quad y_{n+1} = \frac{1}{n+1} K a_n$$

Example 2: $y' = Ky^\alpha$

First represent $y(x) = \sum_{n=0}^{\infty} y_n (x-x_0)^n$,.
Since

$$y'(x) = \sum_{n=0}^{\infty} (n+1)\, y_{n+1} (x-x_0)^n = \sum_{n=0}^{\infty} (n+1)\, y_{n+1}\, x^n,$$

We would have a simple recursion to recover coefficients y_n if we can compute the a_n coefficients for

$$y^\alpha = \sum_{n=0}^{\infty} a_n (x-x_0)^n = \sum_{n=0}^{\infty} a_n\, x^n,$$

For example:

$$(n+1)\cdot y_{n+1} = Ka_n \quad \text{means} \quad y_{n+1} = \frac{1}{n+1} Ka_n$$

We could be stubborn and push this through, but the coefficients of $(a_n)'$ are a little bit easier to compute.

Example 2: $y' = Ky^{\alpha}$

Option A

We have

$$y' = Ky^{\alpha}, \quad y(0) = y_0,$$

and consider the following change of variables:
$x_1 = y$, $x_2 = Ky^{\alpha}$, and $x_3 = y^{-1}$.

Then,

$$\begin{aligned} x_1' &= y' = x_2 & x_1(0) &= y_0, \\ x_2' &= K\alpha y^{\alpha-1}y' = \alpha K\frac{y^{\alpha}}{y}y' = \alpha x_2^2\, x_3 & x_2(0) &= Ky_0^{\alpha}, \\ x_3' &= -y^{-2}y' = -(y^{-1})^2y' = -x_3^2\, x_2 & x_3(0) &= y_0^{-1}. \end{aligned} \tag{1}$$

Example 2: $y' = Ky^{\alpha}$
Option A

We have

$$y' = Ky^{\alpha}, \quad y(0) = y_0,$$

and consider the following change of variables:
$x_1 = y$, $x_2 = Ky^{\alpha}$, and $x_3 = y^{-1}$.

Then,

$$\begin{aligned} x_1' &= y' = x_2 & x_1(0) &= y_0, \\ x_2' &= K\alpha y^{\alpha-1}y' = \alpha K\frac{y^{\alpha}}{y}y' = \alpha x_2^2\ x_3 & x_2(0) &= Ky_0^{\alpha}, \\ x_3' &= -y^{-2}y' = -(y^{-1})^2 y' = -x_3^2\ x_2 & x_3(0) &= y_0^{-1}. \end{aligned} \tag{1}$$

Is there an even simpler system?

Example 2: $y' = Ky^\alpha$

Option B

Yes! Let $w = y^{\alpha-1}$.

Then,

$$\begin{aligned} y' &= Ky^\alpha & y(0) &= y_0 \\ &= Kyy^{\alpha-1} \\ &= Kyw, \\ w' &= (\alpha-1)y^{\alpha-2}y' & w(0) &= y_0^{\alpha-1} \\ &= (\alpha-1)y^{\alpha-2}Ky^\alpha \\ &= (\alpha-1)Ky^{2\alpha-2} \\ &= (\alpha-1)K(y^{\alpha-1})^2 \\ &= (\alpha-1)Kw^2, \end{aligned}$$

Example 2: $y' = Ky^{\alpha}$

A comparison

Using a degree 4 power series...

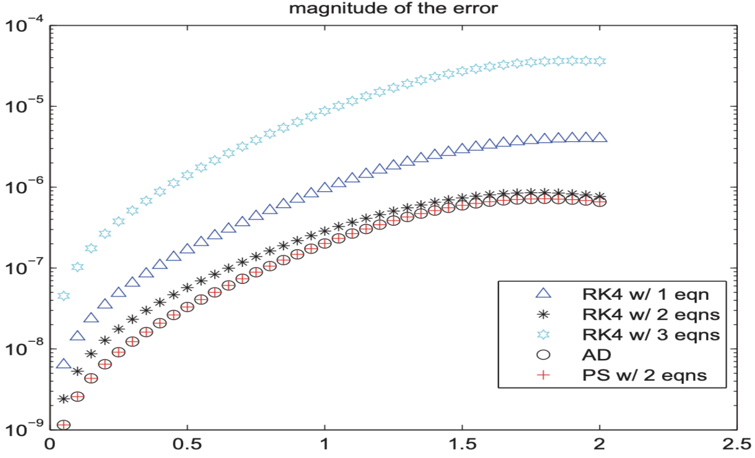


Figure: Error when using a fixed step Runge-Kutta on [0,2] with $h = .05$ and $y_0 = 1, K = 1, \alpha = e/2 + i/\pi$.

Example 3: Other nonlinearities?

What about other types of nonlinearities, or nonlinear systems? Once again, a series ansatz often leds to an easy to implement recurrence relations.
As a simple example, consider the (non-dimensionalized) nonlinear pendulum

$$y'' = -\sin(y) \qquad y(t_0) = y_0, y'(t_0) = y_1 \tag{2}$$

If we let

$$x_0 = y, \quad x_1 = y', \quad U_1 = \sin(y), \quad \text{and} \quad U_2 = \cos(y) \tag{3}$$

we get a polynomial system.

Example 3a: $y'' = -\sin(y)$

Taking $x_0 = y, \quad x_1 = y', \quad U_1 = \sin(y)$ and $U_2 = \cos(y)$, then

$$\begin{aligned}
x_0' &= y' = -x_1 & x_0(t_0) &= y_0 \\
x_1' &= \sin(y) = U_1 & x_1(t_0) &= y_1 \\
U_1' &= \cos(y) \cdot y' = U_2 \cdot x_1 & U_1(t_0) &= \sin(y_0) \\
U_2' &= -\sin(y) \cdot y' = -U_1 \cdot x_1 & U_2(t_0) &= \cos(y_0)
\end{aligned} \tag{4}$$

Solve this system iteratively! But setting up these systems is getting tedious...

Example 3a: $y'' = -\sin(y)$

Taking $x_0 = y, \quad x_1 = y', \quad U_1 = \sin(y)$ and $U_2 = \cos(y)$, then

$$\begin{aligned} x_0' &= y' = -x_1 & x_0(t_0) &= y_0 \\ x_1' &= \sin(y) = U_1 & x_1(t_0) &= y_1 \\ U_1' &= \cos(y) \cdot y' = U_2 \cdot x_1 & U_1(t_0) &= \sin(y_0) \\ U_2' &= -\sin(y) \cdot y' = -U_1 \cdot x_1 & U_2(t_0) &= \cos(y_0) \end{aligned} \tag{4}$$

Solve this system iteratively! But setting up these systems is getting tedious...

Can we automate the process?

Example 3a: auto-generated code?

There are several packages that produce numerical ODEs solutions throught auto-generated code. Nearly all grew from the AD community.

- ATOMFT (Fortran: Chang & Corliss) 1982
- TAYLOR (C: Jorba & Zou) 2001
- The Taylor Center (Delphi: Gofen) 2006
- DAETS (C++: Pryce & Nedialkov) 2007
- TIDES (Mathematica engine: Abad, Barrio, Blesa, Rodriquez) 2010
- QBee (python: Bychkov & Pogudin) 2021
- ODEPSM(Matlab: Neidinger) 2023

ODEPSM

Rich Neidinger started to develop ODEPSM in 2016 while on sabattical at JMU. His current version is availible on GitHub, which includes his pre-print as (recently!) submitted to ACM-TOMS. (`https://github.com/rineidinger/psm4odes`)

The syntax for `ODEPSM` mimic solvers of MATLAB's ODEsuite. The `ODEPSM` auto-generates a function to generate a series based approximate IVODe solution. Adaptive or fixed step numerical, variable precision, and symbolic evaluation of arbitrary degree taylor series coefficients may be returned.

Thanks to Rich and his ODEPSM tool, we can finally get this talk moving!

ODEPSM Github

Example 3a: ODEPSM

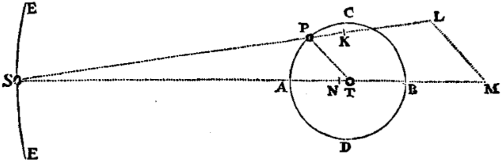
$>$ f = @(t,y) [y(2); -sin(y(1)] $>$ odepsmJZ(f,[0,1],[1;1],1e-1);

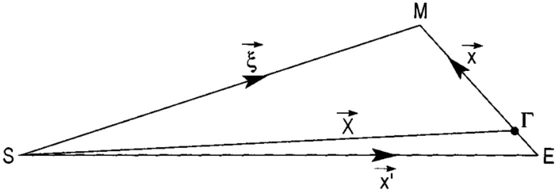
```
function coefs = fseries(t0, y0, deg)
%  FSERIES finds series coefs soln of y' = f(t,y) about t0 to deg
Y(:,1) = y0

U(1,1) = sin(Y(1,1));
U(2,1) = cos(Y(1,1,));
% Update Y linear coefficeint by y' = f(t,y)
Y(1,2) = Y(2,1);
Y(2,2) = -U(1,1);

% Now recurrence rules for each operation in evaluation of f
for j = 1:deg
   tempprime = ( Y(1,2:j) .* (1:(j-1)) ).';
   U(1,j) = ( U(2,(j-1):-1:1) * tempprime)/(j-1);
   U(1,j) = -( U(1,(j-1):-1:1) * tempprime)/(j-1);
   % update Y next coeffient using y' = f(t,y)
   Y(1,j+1) = Y(2,j)/j;
   Y(2,j+1) = -U(1,j)/j;
end
coefs = Y;
```

ODEPSMJZ generates this subfunction, calls a stepper routine on it, calculates an effective local time step, evaluates (using horner's algorithm), and re-initializes for the next step in the global time interval.

The three body problem

By 1687 in the Principia (I.66), Newton presented a geometric approach to analyze the motion of the moon.





His analysis of lunar motion was essentially a clever perturbation method, one unfortunately hampered by the choice of coordinates and the approximations that were made.

Hill's Lunar equations

Hill's 1877 "On the Part of the Motion of the Lunar Perigee which is a Function of the Mean Motions of the Sun and the Moon"
Key features:

- earth-centric coordinate frame
- sun massive, infinitely far away.
- moon as point mass in rotating coordinate, one axis towards sun.
- looked for a periodic orbit of a perturbed system
- not a systematic perturbation theory, but thoughtful expansion of variables

For much more detail, see survey by:
Martin C. Gutzwiller, Moon-Earth-Sun: The oldest three-body problem, Rev. Mod. Phys. 1998 [4]

Hill's Lunar equations

From [Waldvogel, 1997] in geocentric cartesian (x, y):

$$\ddot{x} - 2\dot{y} = 3x - xr^{-3} \tag{5}$$

$$\ddot{y} + 2\dot{x} = \quad + yr^{-3} \tag{6}$$

with a conserved quantity $h = \frac{1}{2}\left(\dot{x}^2 + \dot{y}^2\right) - \frac{3}{2}x^2 - \frac{1}{r}, \quad r = \sqrt{x^2 + y^2}$

Hill's Lunar equations: How to integrate?

Power series?

Hill's Lunar equations: How to integrate?

Power series?

Power series methods may provide *effectively symplectic* integration of conservative systems through a (numerically) faithful algorithm.

```
function dYdt = fhill3_xy(t,Y)

recipr3 = 1/(Y(1)^2 + Y(3)^2)^(3/2);

dYdt = [ Y(2);
    2*Y(4) + 3*Y(1) - Y(1)*recipr3;
    Y(4);
    -2*Y(2) - Y(3)*recipr3];
end
```

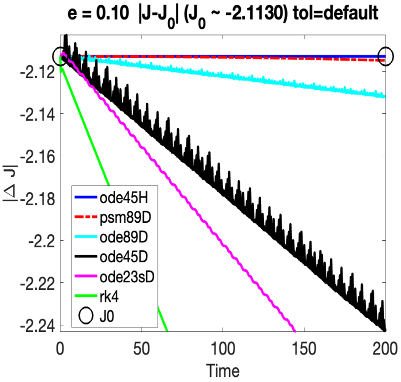
Hill's Lunar equations: PSM (x,y)

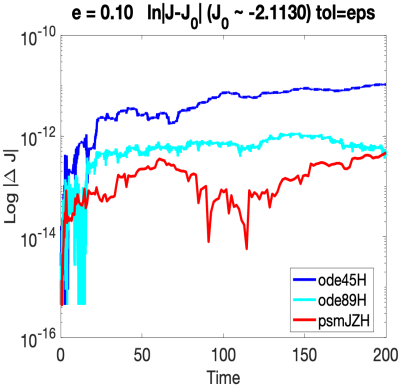
A system of recurrence relations as auto-generated by ODEPSM [?]

```
>> analyze(fhill3_xy,0,[1;0;0;1])
    'Definition           Series recur **=CP                      '
    'u1 = y1 * y1         u1 = y1 * y1                            '
    'u2 = y3 * y3         u2 = y3 * y3                            '
    'u3 = u1 + u2                                                 '
    'u4 = u3^1.5          u3 * u4' = 1.5 u4 * u3' solve for u4(k)'
    'u5 = 1 / u4          1 = u5 * u4 solve for u5(k)             '
    'u6 = 2 * y4                                                  '
    'u7 = 3 * y1                                                  '
    'u8 = u6 + u7                                                 '
    'u9 = y1 * u5         u9 = y1 * u5                            '
    'u10 = u8 - u9                                                '
    'u11 = -2 * y2                                                '
    'u12 = y3 * u5        u12 = y3 * u5                           '
    'u13 = u11 - u12                                              '
    '                     y1' = y2                                '
    '                     y2' = u10                               '
    '                     y3' = y4                                '
    '                     y4' = u13                               '
```

Example 4: Non-stiff

$e = 0.1$

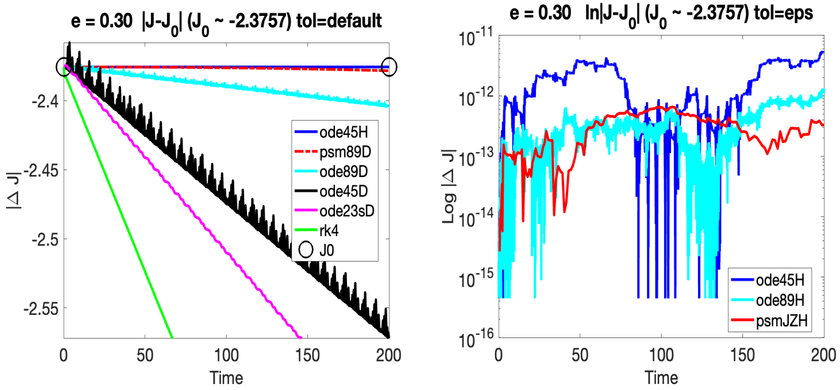




method	ode89D	psm89D	ode45H	ode89H	psmJZH
Run Time:	1.84E-01	1.45E-01	1.66E-01	1.16E-01	3.38E-02
Steps:	985	272	159077	12729	165

Example 4: Non-stiff

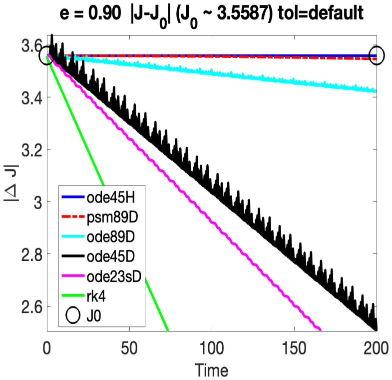
$$e = 0.3$$

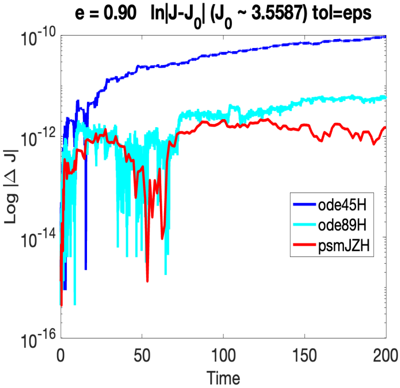


method	ode89D	psm89D	ode45H	ode89H	psmJZH
Run Time:	3.86E-03	1.52E-02	1.20E-01	2.46E-02	2.50E-02
Steps:	985	272	159829	12753	164

Example 4: Non-stiff

$e = 0.9$





method	ode89D	psm89D	ode45H	ode89H	psmJZH
Run Time:	3.64E-03	1.34E-02	1.23E-01	3.05E-02	2.67E-02
Steps:	993	274	161577	12825	161

Example 5: Stiff ODEs?

What is a “stiff” ODE? Wikipedia says that it is an ODE where:

> *.... certain numerical methods for solving the equation are numerically unstable, unless the step size is taken to be extremely small. It has proven difficult to formulate a precise definition of stiffness, but the main idea is that the equation includes some terms that can lead to rapid variation in the solution.*

Can PSM help explain this?

Example 5: Stiff ODEs?

What is a "stiff" ODE? Wikipedia says that it is an ODE where:

> *.... certain numerical methods for solving the equation are numerically unstable, unless the step size is taken to be extremely small. It has proven difficult to formulate a precise definition of stiffness, but the main idea is that the equation includes some terms that can lead to rapid variation in the solution.*

Can PSM help explain this? We don't know.. 🥴

Example 5a: Flame

The description below is directly from

https://www.mathworks.com/company/newsletters/articles/stiff-differential-equations.html

... When you light a match, the ball of flame grows rapidly until it reaches a critical size. Then it remains at that size because the amount of oxygen being consumed by the combustion in the interior of the ball balances the amount available through the surface. The simple model is

$$\frac{dy}{dt} = y^2 - y^3, \quad y(0) = \delta \qquad 0 < t < \frac{2}{\delta}$$

The scalar variable $y(t)$ represents the radius of the ball. The y^2 and y^3 terms come from the surface area and the volume. The critical parameter is the initial radius, δ , which is "small." We seek the solution over a length of time that is inversely proportional to δ.

Example 5: Flame

when `delta = 1e-7`, ODE45 and ODE23 struggle:

```
>> tic; [t45,y45] = ode45(FF,[0 2/delta],delta);toc
Elapsed time is 6.557995 seconds.
>> tic; [t23,y23] = ode23(FF,[0 2/delta],delta);toc
Elapsed time is 3.811527 seconds.
```

Example 5: Flame

when `delta = 1e-7`, ODE45 and ODE23 struggle:

```
>> tic; [t45,y45] = ode45(FF,[0 2/delta],delta);toc
Elapsed time is 6.557995 seconds.
>> tic; [t23,y23] = ode23(FF,[0 2/delta],delta);toc
Elapsed time is 3.811527 seconds.
```

Terrible!

Example 5: Flame

when `delta = 1e-7`, ODE45 and ODE23 struggle:

```
>> tic; [t45,y45] = ode45(FF,[0 2/delta],delta);toc
Elapsed time is 6.557995 seconds.
>> tic; [t23,y23] = ode23(FF,[0 2/delta],delta);toc
Elapsed time is 3.811527 seconds.
```

Terrible! But PSM is even worse!

```
>> tic; [tpsm,ypsm,deg] = odepsmJZ(FF,[0 2/delta],delta,1e-6);
Elapsed time is 37.443010 seconds.
```

PSM attempts to construct a degree 8 approximation over 2147686 time steps!
Hmmmm.....

Example 5a: Flame

What if we ask for a more precise solution?

```
>> tic; [tpsm,ypsm] = odepsmJZ(@fflame,[0,endt],y0,1e-14);toc
Elapsed time is 0.011421 seconds.
```

This is a degree 18 approximation over 93 time steps.

Example 5a: Flame

What if we ask for a more precise solution?

```
>> tic; [tpsm,ypsm] = odepsmJZ(@fflame,[0,endt],y0,1e-14);toc
Elapsed time is 0.011421 seconds.
```

This is a degree 18 approximation over 93 time steps.

How do MATLAB's stiff solvers handle this?

Example 5a: Flame

What if we ask for a more precise solution?

```
>> tic; [tpsm,ypsm] = odepsmJZ(@fflame,[0,endt],y0,1e-14);toc
Elapsed time is 0.011421 seconds.
```

This is a degree 18 approximation over 93 time steps.

How do MATLAB'S stiff solvers handle this?

```
>> tic; [t23s,y23s] = ode15s(@fflame,[0,endt],y0);toc
Elapsed time is 0.200175 seconds. (using 140 steps)

>> tic; [t23s,y23s] = ode23s(@fflame,[0,endt],y0);toc
Elapsed time is 0.072760 seconds. (in 77 steps)
```

Example 5a: Flame

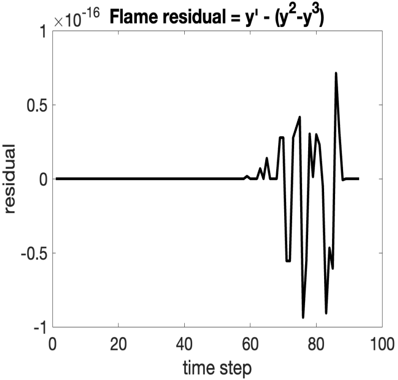
What about accuracy?

Example 5a: Flame

What about accuracy? We can compute the residual error from our continuous PSM solution, which is something that we can NOT do with solutions from discrete solvers.

```
>>k = 1;
>> for y0 = ypsm
         coefs(k,:) = fflameseries(0,y0,30); k = k+1;
      end
>> flame_res = abs(coefs(:,2) - (ypsm.^2 - ypsm.^3));
```

Example 5a: $y' = y^2 - y^3$



Question: Why does raising the tolerance seem to work in this case?

Conclusions

Our Power Series Methods (PSM) methods rely on the use of auxiliary variables to build a system of polynomial IVODEs. Once the system is polynomial, series methods allow remarkably direct analysis. We demonstrated features of PSM:

- Flexible and fast
- Arbitrarily high order
- Adaptive in time and order
- Compact and efficient data footprint
- Simple to apply thanks to auto generated code
- Generate piecewise polynomials - either numerically or symbolically
- Continuous, so no interpolation needed
- Allow detailed analysis

These techniques apply to a broad range of highly nonlinear ODE. And something we didn't talk about - PSM has an error bound!

Thank you

Thanks!

Questions?

thelwerj@jmu.edu

JMU PSM GROUP: J.S. Sochacki[+], G.E. Parker[+], D.C. Carothers[+], S.K. Lucas, J.D. Rudmin, A. Tongen, D.A. and P.G. Warne
and S. Colafranceschi (EMU), R.D. Neidinger[+] (Davidson College)
([+] emeritus)

ODEPSM: `https://github.com/rineidinger/psm4odes`

References: Hill, and history

George William Hill.
On the part of the motion of the lunar perigee which is a function of the mean motions of the sun and moon.
1886.

George William Hill.
Researches in the lunar theory.
American journal of Mathematics, 1(1):5–26, 1878.

Siegfried Bodenmann.
The 18th-century battle over lunar motion.
Physics Today, 63(1):27–32, 2010.

Martin C Gutzwiller.
Moon-earth-sun: The oldest three-body problem.
Reviews of Modern Physics, 70(2):589, 1998.

References: Symplectic methods

J Waldvogel.
Symplectic integrators for hill's lunar problem.
In The Dynamical Behaviour of our Planetary System: Proceedings of the Fourth Alexander von Humboldt Colloquium on Celestial Mechanics, pages 291–305. Springer, 1997.

Thomas Quinn, Randall P Perrine, Derek C Richardson, and Rory Barnes.
A symplectic integrator for hill's equations.
The Astronomical Journal, 139(2):803, 2010.

Robert I McLachlan and G Reinout W Quispel.
Geometric integrators for odes.
Journal of Physics A: Mathematical and General, 39(19):5251, 2006.

Brett Gladman, Martin Duncan, and Jeff Candy.
Symplectic integrators for long-term integrations in celestial mechanics.
Celestial Mechanics and Dynamical Astronomy, 52:221–240, 1991.

References: PSM

E. Fehlberg
Numerical integration of differential equations by power series expansions, illustrated by physical examples.
Technical Report NASA-TN-D-2356, NASA, 1964.

David Carothers et. al.
Some properties of solutions to polynomial systems of differential equations.
Electronic Journal of Differential Equations, 2005:1–18, 2005.

P. G. Warne et. al.
Explicit a-priori error bounds and adaptive error control for approximation of nonlinear initial value differential systems.
Comput. Math. Appl., 52(12):1695–1710, 2006.