



## HP Forum Archive 13

[ [Return to Index](#) | [Top of Index](#) ]

### HP-11C Solutions Handbook Errata?

Message #1 Posted by [Patrick](#) on 31 Aug 2003, 6:20 p.m.

In my April 1981 version of the HP-11C Solutions Handbook (00011-90009) there is a formula for a Runge-Kutta solution for a second order differential equation that I think is in error. On the bottom of page 22, the formulas for  $k_2$  and  $k_3$  both include a factor of  $h/8$  in the second argument. I have worked out these formulas myself and I always get  $h/4$ .

Does anyone have a later edition of this book? If so, could you tell me whether yours has an 8 or a 4 in those formulas?

As an aside, I have written programs based on the value 4 which provide correct results to 10 digits of accuracy with a step size of 0.01 while taking over one hundred steps, so I'm pretty sure the 4 is correct and the book is "buggy"!

### Re: HP-11C Solutions Handbook Errata?

Message #2 Posted by [Valentin Albillo](#) on 1 Sept 2003, 7:12 a.m.,  
in response to message #1 by Patrick

Hi, Patrick:

I don't have the Solutions Book at hand now, but if you've tried both values (the one in the book and the one you think is correct) in several test equations and yours provides the accurate answers while the one in the book does not, the conclusion is obvious, don't you agree ? Anyway, I'll check it within 10 hours, back home.

Now, here's a 'challenge' specifically for you (others may apply as well and are welcome to try, of course), your preferred algorithm (Runge-Kutta or whatever), and your preferred HP calculator (physical or emulated):

Given the 1st-order differential equation:

$$y' = x^2 + y^2$$

subject to the initial condition  $y(0) = 0$ , try and find  $y(2)$ , *accurate to 10 or 12 decimal places*, give or take a few ulps.

All correct answers will be rewarded with a pre-release copy of my "Long Live the HP-16C!" incoming article, in PDF format. All those people who do not have my previous articles will receive them as well, if desired. :-)

Of course the answer must be obtained using an HP calculator, so the program and/or procedure used to obtain it must be produced on demand.

Beeest regards from V.

### Re: HP-11C Solutions Handbook Errata?

Message #3 Posted by [dbrunell](#) on 1 Sept 2003, 5:37 p.m.,  
in response to message #2 by Valentin Albillo

I find a pole at  $x=2$ .

### Sorry but no cigar :-)... Patrick ?

Message #4 Posted by [Valentin Albillo](#) on 2 Sept 2003, 4:05 a.m.,  
in response to message #3 by dbrunell

There's *\*no\** pole at  $x = 2$ , sorry. If your software says there is, chances are it was written by MS, but in any case you should definitely get a better one. :-) (joking, of course)

Patrick, are you listening ? Does the expert mathematician cum HP-15C programmer accept this simple 'challenge' :-) ?

What's the 10-digit-accurate value of  $y(2)$  ? Further, if this function has some pole, could any of you provide a 10-digit-accurate value for the corresponding  $x$  ? It's *\*not\** 2.000000000, mind you.

Best regards from V.

*Edited: 2 Sept 2003, 7:55 a.m.*

### Yes, I'm listening

Message #5 Posted by [Patrick](#) on 2 Sept 2003, 4:29 p.m.,  
in response to message #4 by Valentin Albillo

But because it came from you, Valentin, I just knew there had to be a catch! I'm sure, knowing nothing but your reputation, that a 10 digit value for  $y(2)$  derived through numerical means will somehow turn out to be meaningless! I'm just trying to think of why...

Besides, differential equations was.... uh.... not my field. Yea, that's the ticket...

### Re: Sorry but no cigar :-)... Patrick ?

Message #6 Posted by [dbrunell](#) on 2 Sept 2003, 6:20 p.m.,  
in response to message #4 by Valentin Albillo

I tried to find a solution on my 97 using the R-K program in the Math Pac. For successively smaller values of  $h$ , I got successively larger results for  $y(2)$ . Therefore, I concluded there was a pole at  $x=2$ . Then I ran it under Mathematica, and here is what I got:

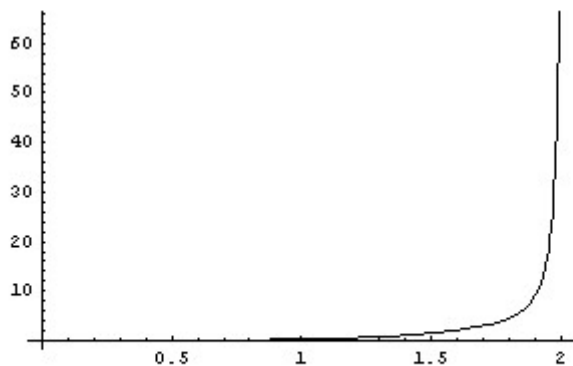
```
In[1]:= NDSolve[{y'[x] == x^2 + y[x]^2, y[0] == 0}, y, {x, 0, 3}]
```

```
NDSolve::ndsz :
```

```
At x == 2.0031287147148746`, step size is  
effectively zero; singularity suspected.
```

```
Out[1]= {{y -> InterpolatingFunction[{{0., 2.00313}}, <>]}}
```

```
In[2]:= Plot[Evaluate[y[x] /. %], {x, 0, 2}]
```



So what am I doing wrong?

### Re: Sorry but no cigar :-)... Patrick ?

Message #7 Posted by [tony](#) on 3 Sept 2003, 1:44 a.m.,

*in response to message #6 by dbrunell*

Looks like the pole is just above 2

**Re: Sorry but no cigar :-)... Patrick ?**

*Message #8 Posted by [tony](#) on 3 Sept 2003, 2:37 a.m.,  
in response to message #6 by dbrunell*

google on

euler maple  $x^2 y^2$

gives a first hit on a pdf at oregonstate.edu which shows maple actually \*solving\* this in closed form using Bessel functions!!!

Their graph only goes to  $x=1$  where it looks like  $y$  is about .35

This is a real challenge!! I wish I knew how to evaluate Bessel functions - the  $Y(2)$  that Maple gives is:

$$2*(BY(-.75,2)-BJ(-.75,2))/(BY(.25,2)-BJ(.25,2))$$

**Re: Sorry but no cigar :-)... Patrick ?**

*Message #9 Posted by [dbrunell](#) on 3 Sept 2003, 3:37 a.m.,  
in response to message #8 by tony*

Here is the function you gave, calculated to 50 decimal places:

```

In[1]:= w = N[BesselY[-3/4, 2], 50]
Out[1]= 0.35912910099873954892594610089701304120963563365074

In[2]:= x = N[BesselJ[-3/4, 2], 50]
Out[2]= -0.44672065795573945332271410953549883280379567646684

In[3]:= y = N[BesselY[1/4, 2], 50]
Out[3]= 0.39273839961538505531541686016466586048231600314247

In[4]:= z = N[BesselJ[1/4, 2], 50]
Out[4]= 0.39781106433817834872522068418500368474780487054308

In[5]:= N[2 * (w - x) / (y - z), 50]
Out[5]= -317.722460675750308399072046810176441472988684038

```

Are you sure you got the signs right? I have Jn on my HP-48 but no Yn, and I'm too lazy to key it in. ;)

*Edited: 3 Sept 2003, 3:37 a.m.*

## A little BIG rant

Message #10 Posted by *Valentin Albillo* on 3 Sept 2003, 4:58 a.m.,  
in response to message #6 by dbrunell

This is going to be a little BIG rant, and I most sincerely hope I don't offend anyone, that's certainly not my intention, at all. Also, though I'm particularizing somewhat on this challenge, most of my remarks actually apply to other such similar challenges I've issued in the past.

The question is, while I really thank those of you such as Patrick and dbrunell and other people who were interested in my 'challenge' and kindly took the trouble to investigate it, I actually feel *\*very\** disillusioned at the approach taken, and I'm now fully convinced that I cannot attain my intended didactic goals no matter how hard I try. I'll explain:

My original message stated this:

*"Now, here's a 'challenge' specifically for you (others may apply as well and are welcome to try, of course), your preferred algorithm (Runge-Kutta or whatever), and your preferred HP calculator (physical or emulated)"*

the intention obviously being to get some HP-programming wizardry in action, a novel approach on how to tackle very difficult problems with our marvelous HP calcs, and thus learning something on the way and appreciating our beloved machines that much more.

What did I get instead ? People absolutely ignoring the challenging side of the problem, and instanly reaching for their top-gun, pc-based Mathematicas and Matlabs, and whatever, after they tried the most cursory approach on their HP calcs, if at all.

Now, I also own a copy of Mathematica, and my copy can detect that this is an instance of the Ricatti differential equation and provide an explicit symbolic expression for the solution in terms of BesselJ and BesselY functions, which can be evaluated to thousands of decimal places at  $x = 2$ , or at  $y \rightarrow \text{infinite}$ .

So what ? I'm *\*not\** interested in the least in the actual, numeric solution, what I wanted to see was some clever thinking applied to the problem, in order to solve it in an HP-97, an HP-15C, or and HP-25C, say !! Yes, it *\*can\** be solved on an little, puny, 49-step HP-25C !!

So, I'm terribly sorry for this rant, but it's simply that I'm somewhat fed up. Everytime I issue some challenge to be solved using some HP calc, no matter how clearly I state that people should solve it using their HP calcs, it does me no good. People *\*always\** use their PCs do to the job and achieve some results I couldn't be less interested in. As they say, "It's the journey that matters, *\*not\** the destination".

So much for my rant. I really hope noone feels offended, indeed no offence was intended and I heartily apologize if any was taken. Anyway, as this isn't gonna change, as it seems that noone can refrain from putting their HP calcs apart, as the useless toys they surely are, and reaching for their oh-so-wonderful Mathematica software running on a oh-so-marvelous 2.4 Ghz Pentium IV PC, I'll refrain myself from issuing any such numerical 'challenges' in the future, lest I receive yet another 300+-digit Mathematica-computed result I couldn't care less for.

Best regards from V.

*Edited: 3 Sept 2003, 6:31 a.m.*

### **Re: A little BIG rant**

*Message #11 Posted by **Veli-Pekka Nousiainen** on 3 Sept 2003, 6:42 a.m.,  
in response to message #10 by Valentin Albillo*

Can I use a HP 200 LX (with DERIVE student version)??

### **Re: A little BIG rant**

Message #12 Posted by **christof (NoVA US)** on 3 Sept 2003, 4:06 p.m.,  
in response to message #10 by Valentin Albillo

I've also been very interested in HP specific math elegance- partly because I do use them daily, nad partly because I keep a little notebook where I write keystroke programming 'bits' to use later. And they always do seem to get used.

I don't have an answer for this one, though :)

### Re: A little BIG rant

Message #13 Posted by **Wayne Brown** on 3 Sept 2003, 4:29 p.m.,  
in response to message #10 by Valentin Albillo

Is there any reason it can't be done *both* ways -- with an HP calculator and with a computer (or even by hand)? I don't see that some people giving a Mathematica or Matlab solution *prevents* anyone else from offering a calculator solution. Personally, I'd like to see both.

### Re: A little BIG rant

Message #14 Posted by **dbrunell** on 4 Sept 2003, 3:39 a.m.,  
in response to message #10 by Valentin Albillo

OK Valentin, you win.

The solution to the equation can be represented by:

$$y(x) = x \cdot \frac{J_{3/4} \left( \frac{1}{2} x^2 \right) - c J_{-3/4} \left( \frac{1}{2} x^2 \right)}{c J_{1/4} \left( \frac{1}{2} x^2 \right) + J_{-1/4} \left( \frac{1}{2} x^2 \right)}$$

Where c is subject to initial conditions:

$$c = - \frac{y(0) \Gamma \left( \frac{1}{4} \right)}{2 \Gamma \left( \frac{3}{4} \right)}$$

Thanks to your initial condition of y(0)=0, we are simply left with y(2)=2\*J(.75,2)/J(-.25,2).

Here is my 48S program for Bessel function of the first kind:

```
%%HP: T(3)A(D)F(.);
\<< \-> N X
  \<< 0 0 1.E99
    FOR K DUP '(-1)
^K*(X/2)^(N+2*K)/(K
!* (N+K)! ' EVAL +
  IF DUP ROT \=/
  THEN 1
  ELSE 2.E99
  END
STEP
\>>
\>>
```

I get an answer of 317.722460354.

I will leave it to you to write a Bessel function for the HP-25. :)

### Re: A little BIG rant

*Message #15 Posted by [tony](#) on 5 Sept 2003, 1:18 a.m.,  
in response to message #14 by dbrunell*

Thanks for the great HP48 program!!

FWIW this produces 317.722 460 675 750 307

I run it using powerbasic.

It requires rather great storage<G> but only really uses the four basic arithmetic operators. It just expands  $y(x)$  as a polynomial (I put  $dy/dx = x^2 + y^2$  and match up coefficients. The same as a Taylor expansion about (0,0)).

```
defext a-z
dim a(6500)
t=6500
x=2
k=3
a(1)=(X^K)/K
```



```

S=a(1)
for i=2 to t
k=k+4
a(i)=0
for j=1 to i-1
a(i)=a(i)+a(j)*a(i-j)
next j
a(i)=a(i)*x/k
s=s+a(i)
print i,a(i);
print using "###.#####";s
next i
end

```

If only I could spot a pattern in the coefficients it could be done with minimal storage and would be a candidate for a calculator.

It runs fine in quickbasic with defdbl in the first line.

### Small HP calc versus MATHEMATICA ! [LONG]

Message #16 Posted by [Valentin Albillo](#) on 5 Sept 2003, 6:48 a.m.,  
in response to message #14 by [dbrunell](#)

For those of you (if any) expecting a solution using just some tiny HP calculator, read on, I include a comprehensive one at the end of this post, but first some pertinent comments:

dbrunell posted:

*"The solution to the equation can be represented by: [an explicit symbolic solution found using Mathematica]"*

First of all, let me clearly state that I thank you very much for your interest in this informal 'challenge' I issued, and rest assured that none of my remarks are meant to be derogatory of your efforts to solve it. It's just that I am the kind of person that likes to abide by the rules of the game and expect others to do likewise, so I'm somewhat upset when someone disregards them, specially when it's totally unnecessary.

As you may guess, using Mathematica, MathLab, Derive, or whatever to solve one of my challenges is *utterly useless* as far as showing and learning new tricks with out beloved HP calculators is concerned, and as far as I'm concerned as well. As I said before, I also own a copy of Mathematica, and can easily achieve these results with no effort and to no purpose whatsoever.

Should you had found the explicit symbolic solution using some CAS or symbolic package in your HP48/49, that would be a thing. But finding it using Mathematica is breaking the rules big time. Now, once Mathematica has given you that weird-looking expression in terms of Bessel functions , you then go on:

*"Here is my 48S program for Bessel function of the first kind: "*

**THIS** is more like it !! That's what I want, some real code running in a real (or emulated) HP calculator, solving some difficult problem. What a pity that you chose to resort to that Mathematica of yours, instead of fully solving it with your powerful 48S from the scratch ! But you then end it all saying:

*"I will leave it to you to write a Bessel function for the HP-25. :) "*

What for ? WHO needs Bessel functions to solve this problem ? Mathematica needs them ? And WHO needs Mathematica for this problem ? Not anyone owning most any HP RPN or RPL calculators, for certain ...

In order to prove my point, here's a complete, reasoned solution to this challenge, using nothing more and nothing else than a vintage HP calculator. The solution is given for the HP-15C, for the one and only reason that it's the HP model I use routinely. A similar solution can be given for most other models, up to *and including* the HP-25 itself, but I don't have an HP-25 at hand right now. And yes, you *can* solve this challenge quite easily with a humble yet powerful HP-25, in a few minutes. Let's begin:

**Given the equation  $y' = x^2 + y^2$ , with  $y(0) = 0$ , find an accurate value for  $y(2)$ .**

### Step 1:

We need to use some algorithm and corresponding program to find numeric solutions for 1st order differential equations. Any will do, the HP-25C Application Programs does include one (Euler's method), the HP-41C Advantage ROM includes a better one (4th order Runge-Kutta method), and there are many others available.

As I'm going to use my everyday-use HP-15C, this is the program I wrote from scratch, ad-hoc for this task. It's unoptimized, hastily written, but it does the job. It implements Runge-Kutta 4th order method, in 45 steps. The 4th-order RK algorithm goes like this:

Given  $y' = f(x,y)$ , with  $y_0 = y(x_0)$ , we use a suitably small 'step' size,  $h$ , to compute additional values,  $(x_1, y_1)$ ,  $(x_2, y_2)$ , ..., like this:

$$\begin{aligned}x_1 &= x_0 + h \\ y_1 &= y_0 + (k_1 + 2k_2 + 2k_3 + k_4)/6\end{aligned}$$

where  $k_1 = h \cdot f(x_0, y_0)$   
 $k_2 = h \cdot f(x_0 + h/2, y_0 + k_1/2)$   
 $k_3 = h \cdot f(x_0 + h/2, y_0 + k_2/2)$   
 $k_4 = h \cdot f(x_0 + h, y_0 + k_3)$

once (x1, y1) have been computed, you then go on likewise to compute (x2, y2), etc.

and the resulting HP-15C implementation is this:

```

01 LBL A      13 STO 3      25 RCL 1      37 GSB 2
02 STO 0      14 GSB 0      26 PSE       38 STO+ 3
03 Rdn       15 GSB 0      27 RCL 2      39 RTN
04 RTN       16 RCL+ 2     28 PSE       40 LBL 2
05 LBL B      17 RCL 0      29 GTO 1      41 RCL+ 1
06 STO 1      18 GSB 2      30 LBL 0      42 GSB C
07 X<>Y      19 RCL 3      31 2          43 RCLx 0
08 STO 2      20 6          32 /         44 STO+ 3
09 LBL 1      21 /          33 RCL+ 2    45 RTN
10 X<>Y      22 STO+ 2     34 RCL 0     46 LBL C
11 GSB C      23 RCL 0      35 2          47 RTN
12 RCLx 0     24 STO+ 1     36 /

```

where you define your function  $f(x,y)$  under LBL C, assuming y is in the Y register and x is in the X register. Set User mode and specify the step value h, like this:

h, A

Then enter the initial condition (x0,y0) and proceed to compute (x1,y1), (x2, y2), etc, like this:

y0, ENTER, x0, B -> (x1) -> (y1) -> (x2) -> y(2)

press R/S to stop the program when done. By the way, this program *doesn't use any unique HP-15C features*, apart from recall arithmetic (RCLx 0) which you can easily simulate (RCL 0, x), so *it will run* essentially unchanged on an HP-11C, HP-34C, and most any models having subroutine capability and labels. A version for the HP-25 and similar GSB-less machines is available as well.

### Step 2:

Now, we need to define our equation,  $y' = x^2 + y^2$ . To this purpose, insert these steps after 46 LBL C

```

47 X^2
48 X<>Y
49 X^2
50 +

```

### Step 3:

Let's try to find  $y(2)$ . In User mode, FIX 7, using  $h = 0.1$  (.1, A), and  $x_0 = 0$ ,  $y_0 = 0$  (0, ENTER, 0, B), we let the program run until it arrives at  $x = 2$ , duly noting the final result and several intermediate ones for  $x = 1, 1.7, 1.8$ , and  $1.9$ . To check the accuracy of the result, we repeat the procedure a couple of times, halving the step size to  $h = 0.05$ , then to  $h = 0.025$ . This is what we get:

x	y (h=0.1)	y (h=0.05)	y (h=0.025)
0	0	0	0
1	0.3502337	0.3502320	0.3502319
1.7	2.9727649	2.9727984	2.9727975
1.8	4.6866510	4.6880373	4.6881254
1.9	9.5161319	9.5622688	9.5666582
2	71.5789968	124.2866329	194.9352142

#### Step 4:

Now, instead of blindly accepting the last value for  $y(2)$ , we must evaluate the results. From the table above, it's obvious that while  $y(1) = 0.3502319$  is accurate to 7 decimals,  $y(1.8)$  is accurate to no more than 4 decimals,  $y(1.9)$  only to 2 decimals at best, and  $y(2)$  isn't accurate to any given digits. Further, the  $y$ -values seem to be more than doubling for a small linear increase ( $h$ ) of the  $x$ -values, and this exponential growth should make us suspect a pole (singularity) at or near  $x = 2$ .

#### Step 5:

If a singularity is present nearby, reducing the step size  $h$  further to  $h = 0.001$ ,  $0.00001$ , or smaller, will be no use. The running times will be astronomic and the accumulation of rounding errors to reach  $x = 2$  after 200,000 steps or more will make the final result inaccurate anyway. Do we reach for Mathematica? No way, it's just that a different approach is called for.

As it's well known, HP vintage calcs were "*designed by engineers for engineers*". Though many people now tend to take the easy ride and use pure brute force (i.e., Mathematica's  $N[ \dots, 100]$ ) instead of neuron-power, engineers of old were expected to have some familiarity with Calculus, at least elementary Calculus, plus a fine intuition for difficulties. On suspecting a pole, a singularity, any engineer worth his/her salt would think along these lines:

*"Aw, gee, we've got some nasty singularity nearby, which is spoiling our algorithm's accuracy. Let's remove it out of existence! How? Easy ... let's make some change of variable, such as to turn a singularity-ridden function, like this unknown  $y(x)$ , into some smooth, singularity-free one ... how about the  $\arctan(x)$  function? It will return a finite value,  $\text{Pi}/2$  at most, for any argument  $x$ , even if it goes to infinity. Let's try!"*

#### Step 6:

Lo and behold, let's make a change of variable:

$$[1] \quad y = \tan(z)$$

so then we have:

$$[2] \quad z = \arctan(y)$$

its derivative will be:

$$[3] \quad z' = y' / (1 + y^2)$$

isolating  $y'$  gives:

$$[4] \quad y' = z' \cdot (1 + y^2) = z' \cdot (1 + \tan^2(z))$$

but the original equation says:

$$[5] \quad y' = x^2 + y^2 = x^2 + \tan^2(z)$$

so equating both expressions [4] and [5], we get:

$$[6] \quad z' \cdot (1 + \tan^2(z)) = x^2 + \tan^2(z)$$

which, after isolating  $z'$  gives:

$$[7] \quad z' = (x^2 + \tan^2(z)) / (1 + \tan^2(z))$$

which is the new differential equation (in  $[x, z, z']$ , instead of the original  $[x, y, y']$ ) we must now solve, subject to the initial condition:

$$z(0) = \arctan(y(0)) = \arctan(0) = 0$$

and once we find  $z(2)$  using our HP-15C program, we'll simply have:

$$y(2) = \tan(z(2))$$

### Step 7:

The new differential equation is programmed under LBL C (after removing the original equation) like this:

```

47 X^2      52  1
48 X<>Y    53  LASTX
49 TAN     54  +
50 X^2     55  /
51 +       56  RTN

```

and now, in RAD mode, using  $h = 0.1$ , then  $h = 0.05$ , we get

x	y (h=0.1)	y (h=0.05)
0	0	0
1	0.3368811	0.3368813
1.7	1.2463033	1.2463031
1.8	1.3606417	1.3606412
1.9	1.4666491	1.4666485
2.0	1.5676491	1.5676489

which clearly shows that the singularity has been effectively removed, so we've got:

$$z(2) = 1.5676489$$

and thus:

$$y(2) = \tan(z(2)) = 317.7225457$$

which has *nearly 7 digits correct*, greater accuracy being possible by simply using a smaller step size, h. Further, the exact x value corresponding to the pole near x=2 can be found accurately with our little HP calculator using a single iteration of Newton's method, like this:

$$x_{\text{pole}} = 2 - (z(2) - \pi/2) / z'(2)$$

where we already have z(2) and can compute z'(2) by simply using the differential equation [7]. So, calling z(2) = 1.5676489 = k, we then compute:

$$\begin{aligned} x_{\text{pole}} &= 2 - (k - \pi/2) / z'(2) \\ &= 2 - (k - \pi/2) / ((4 + \tan^2(k)) / (1 + \tan^2(k))) \\ &= 2.0031473 \end{aligned}$$

which is *accurate to all 8 digits shown*.

Thusly, we've succeeded in finding both the value of y(2) and the singularity, to 7-8 digits accuracy, using just our small HP calculator and a little bit of the elementary knowledge and basic intuition any engineer should have in spades.

If any of you did find some of this useful or even learnt something, I think I will have proved my point and achieved my goal.

Best regards from V.

### Re: Small HP calc versus MATHEMATICA ! [LONG]

Message #17 Posted by *Werner Huysegoms* on 5 Sept 2003, 7:20 a.m.,  
in response to message #16 by Valentin Albillo

Thank you, Valentin. I liked it a lot, even if I had to look up the formula for Runge-Kutta again (it's been too long). (yes I was trying to solve it) Your comments are spot-on: having an all-powerful Mathematica on your PC is no substitute to a bit of careful analysis - for some problems will remain beyond even Mathematica's power - and then what?

Werner Huysegoms

### Re: Small HP calc versus MATHEMATICA ! [LONG]

Message #18 Posted by [Valentin Albillo](#) on 5 Sept 2003, 8:08 a.m.,  
in response to message #17 by Werner Huysegoms

Hi, Werner:

Werner posted: "*Thank you, Valentin. I liked it a lot, even if I had to look up the formula for Runge-Kutta again (it's been too long). (yes I was trying to solve it)*"

Thanks for your kind comments, much appreciated. I'm glad to know that you were giving it a try and liked my approach to solving it. I usually hesitate a lot to post such challenges and corresponding long answers, for it's been my (sad) experience that they get by and large ignored, or even worse, are interpreted in a bad light, and frankly, it's a lot of work ! :-)

*"Your comments are spot-on: having an all-powerful Mathematica on your PC is no substitute to a bit of careful analysis - for some problems will remain beyond even Mathematica's power - and then what?"*

Certainly, but sadly few people seem to realize that. For instance, re my 'challenge', Mr. dbrunell used Mathematica to find an explicit solution to  $y' = x^2 + y^2$  and Mathematica succeeded, producing one in terms of Bessel functions. So far so good.

But what if the problem were *slightly* changed to  $y' = x^2 + y^{2.00281}$  or  $y' = x^2 + y^e$ , say ? Though quite similar, they aren't Ricatti-equation instances anymore, and Mathematica thus **fails** to produce an explicit, closed symbolic representation. No Bessel functions you can evaluate on your 48S (what for, if you've got Mathematica ...), no nice closed-form representation for the constant integration factor. The slightest change in the equation completely changes the nature of the solution.

On the other hand, the approach I detailed, using a humble, small HP calculator, is absolutely *oblivious* to the fact that  $y^2$  has been changed to  $y^{2.00281}$  or  $y^e$ , and will work exactly the same, unaffected by the change, because it does not depend on such specifics but in a sound numerical algorithm and some sound judgment and professional proficiency on the part of the user. Brute force only goes so far, and will spend gigabytes and gigaflops to get results that a 49-byte handheld calculator and its knowledgeable user can get much more economically, while learning something in the process.

Thanks again and best regards from V.

*Edited: 5 Sept 2003, 9:21 a.m.*

**Re: Small HP calc versus MATHEMATICA ! [LONG]**

*Message #19 Posted by [tony](#) on 5 Sept 2003, 7:59 p.m.,  
in response to message #16 by Valentin Albillo*

Valentin, thanks for taking the time to prepare and publish your elegant solution!

If TAN is not available then  $y=z/(1-z)$  is the best substitution I can find. The equation becomes  $z'=(x^2)*(1-z)^2 + z^2$ , and we start from  $x=0, z=0$ . With  $h=.1$  I end up with  $y(2)=318$  and with  $h=.05$  I get  $y(2)=317.73$ . So, it's not as powerful as  $y=TAN(z)$ , but not a bad substitute.

Cheers, Tony

**Re: Small HP calc versus MATHEMATICA ! [LONG]**

*Message #20 Posted by [Tom Scott](#) on 5 Sept 2003, 9:37 p.m.,  
in response to message #16 by Valentin Albillo*

Thanks Valentin. Very educational!

Tom Scott Lander, WY

**Re: Small HP calc versus MATHEMATICA ! [LONG]**

*Message #21 Posted by [dbrunell](#) on 6 Sept 2003, 12:45 a.m.,  
in response to message #16 by Valentin Albillo*

Dear Valentin:

Thank you for taking the time to create that lengthy and educational post. Having taken differential equations over 20 years ago and never using the knowledge in my professional endeavors, I admit to being quite rusty. I would never have considered a change of variable.

Incidentally, the form of the solution I gave in my post (using Bessel functions) came not from Mathematica but from the book "Differential Equations and Linear Algebra" by Edwards and Penney.

I have printed out your post, and it will become a permanent part of my reference material.



Best regards,

David Brunell

### Thanks to all of you. Best regards. [no text]

Message #22 Posted by **Valentin Albillo** on 6 Sept 2003, 9:08 p.m.,  
in response to message #21 by dbrunell

Best regards from V.

### Re: HP-11C Solutions Handbook Errata?

Message #23 Posted by **tony** on 4 Sept 2003, 12:14 a.m.,  
in response to message #2 by Valentin Albillo

Valentin, I am trying a polynomial form for  $y(x) = (x^3)/3 + (x^7)/63 + (x^{11})^2 / (11 * 9 * 7 * 3) + \text{terms in } X^{15}, X^{19}$  etc. If only I could get a manageable pattern in the coefficients, I reckon I could implement it on a calculator with minimal storage. As it is I think I need to remember each coefficient to get the next one. Am I getting close? My algebraic manipulation skills are not what they once were<G>. This is the only way I can see of doing it - following the slope from (0,0) in small steps does not help it is a long journey and one is not sure one is still correctly connected to (0,0).

- later - maybe you use TAN in the solution? I'm after a hint<G>. TAN could be handy because of the  $y^2$  as  $d(\text{TAN}(x))/dx = 1 + \text{TAN}(x)^2$

Yep, maybe better to start with TAN rather than the  $(x^3)/3$  - looks like that polynomial will need a lot of terms for Y(2) and we maybe need the power of TAN?

*Edited: 4 Sept 2003, 1:26 a.m.*

### Re: HP-11C Solutions Handbook Errata?

Message #24 Posted by **Gordon Dyer** on 2 Sept 2003, 4:44 p.m.,  
in response to message #1 by Patrick

Hi, I have the original manual and an Addendum card with all of the corrections.

Send me an email and I can return scans of the addendum card, 2 sides, 389KB and 249KB .jpg files.

### Re: HP-11C Solutions Handbook Errata?

Message #25 Posted by **Gordon Dyer** on 3 Sept 2003, 4:46 p.m.,

*in response to message #24 by Gordon Dyer*

Sorry, the Addendum sheet is for the Owner's Handbook, not the Solutions Handbook.

### **Re: HP-11C Solutions Handbook Errata?**

*Message #26 Posted by [bill platt](#) on 3 Sept 2003, 5:03 p.m.,*

*in response to message #1 by Patrick*

I know I have the solutions handbook at home 2, from my orig. HP. And I remember addendum sheets. I'll take a look tonite and C what I find.

regards,

bill

### **Re: HP-11C Solutions Handbook Errata?**

*Message #27 Posted by [bill platt](#) on 3 Sept 2003, 11:33 p.m.,*

*in response to message #26 by bill platt*

I found the same thing that Gordon found: yes, my copy has the same equation Patrick found:

$$k3 = hf\{xi+(h/2), yi+(h/2)y'i+(h/8)k1, y'i+(k2/2)\}$$

So, that is interesting. If anyone finds addendums to the solutions handbook, let us know.

-Bill

---

[ [Return to Index](#) | [Top of Index](#) ]



[Go back to the main exhibit hall](#)