HP Solve

Calculating solutions powered by HP

In the Spotlight

» **The all-new HP 39gII Calculator has arrived!**

HP Calculators is proud to announce the new HP 39gII, the latest addition to our graphing calculator family. Learn more about all of the newest features available.

Your articles

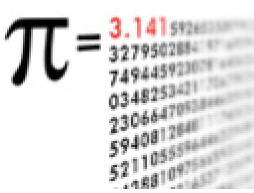
» **HP Contest - Calling all land speed chasers!**

HP will be holding a contest this spring challenging engineering students to submit ideas to help the NAE team reach record-breaking speeds. It's a chance to win a day in the field test running the jet-powered land rocket! For more information, email us at calcenthusiasts@hp.com.

» **HHC 2011 Report**

Richard J. Nelson, Jake Schwartz and Gene Wright

Read this extensive report on the 38th conference dedicated to HP Calculators, the Hewlett-Packard Handheld Conference (HHC). This year's conference was held in San Diego, CA on September 24-25, 2011.



» **The Four Meanings of “Accurate to 3 Places”**

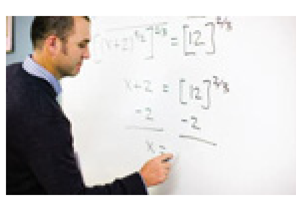
Joseph K. Horn

In this article, Joseph discusses a very important topic of interest to every calculator user—decimal to fraction conversion accuracy.

» **What is Double Injection Molding of HP Calculator Keys?**

Richard J. Nelson

Old time HP calculator users will often mention this unique HP process. Read this short article with photos to understand what this means.



Issue 26
January 2012

Welcome to the twenty-sixth edition of the HP Solve newsletter. Learn calculation concepts, get advice to help you succeed in the office or the classroom, and be the first to find out about new HP calculating solutions and special offers.

» **Download the PDF version** of newsletter articles.

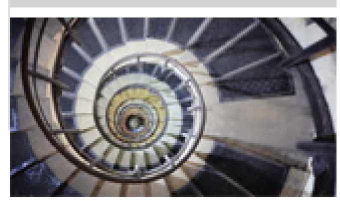
» **NEW! Join our Facebook Fan Page** for additional tutorials, videos, information and more!

» **Contact the editor**

From the Editor

Learn more about current articles and feedback from the latest Solve newsletter including One Minute Marvels and a new column, Calculator Accuracy.

Learn more ›



» **Timing for HP 35s Calculator Instructions**
Richard Schwartz
Every calculator is unique in its method of providing its user interface. In this article, Richard provides a discussion on the methods of making the Instruction timing measurements.

» **Octal Fraction Conversions**
Palmer O. Hanson
This article reviews hand calculation techniques for conversion, illustrates conversions using the Arma tables and discusses methods for conversion using modern hand-held calculators.

» **Fundamentals of Applied Math Series #9**
Richard J. Nelson
The Golden Ratio. This constant is probably one of the most widely intriguing of all mathematical constants. Here is an overview of this artistic number with lots of interesting links for further exploration and study.

Announcing the HP 39gII

Article – Next →

Return to Top

Announcing the HP 39gII

HP Calculators is proud to announce the new HP 39gII, the latest addition to our graphing calculator family. Based on the HP 39gs software architecture with its classic HP app structure, this graphing calculator is designed for the student of mathematics and science. If you are familiar with the HP 39gs, then you already know that the HP 39gII has HP apps that let you save your work and come back to it later; however, check out these cool, new features:

- Context-sensitive Help gets you started quickly and helps you when you need it
- Higher resolution grayscale display makes for crisp graphics and increased readability
- “Adaptive” plotting method gives you very accurate graphs
- Updated programming language, with support for user-defined functions and variables
- Multiple language support now available!
- Units now available!
- App functions now available!

This article gives you an overview of the above features. We will go into detail in later issues.

Online Help

The HP 39gII has an extensive Help system built in. This Help system is context-sensitive; that is, the help displayed is determined by the app, view, menu, or item currently selected. You can enter the Help system at any time by pressing the Shift of the Views key. Press the KEYS menu key to get help on the keyboard keys. Figure 1 shows the help displayed when the Apps key is pressed after entering the Help system from the Home view. Figure 2 shows the help displayed when the Apps Library is open and the Function app is selected. Figure 3 shows the help displayed when the Function app is open and the Symbolic view is active.

Figure 1.

In addition to the help displayed for all apps, views, and menus, each command and function has help as well. Figure 4 shows the Math menu open with the Summation function (Σ) selected. Note the syntax help displayed at the bottom. Pressing Help now will display the help text shown in Figure 5. This part of the Help system explains the syntax of a command or function in detail, often with an example.

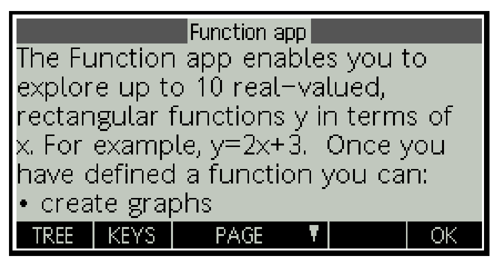


Figure 2.

Crisp and readable display

Figure 6 shows the graph of the Lissajou figure determined by the parametric equations x(t)=9cos(5t) and y(t)=5sin(8t), drawn using the Parametric app. As you saw earlier, Figures 1-5 illustrate the improved readability of the HP 39gII as well.

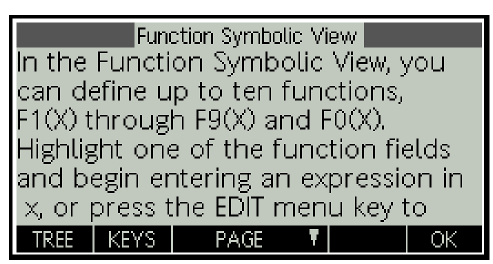


Figure 3.

Adaptive Plotting

Figure 7 shows the graph of the function $y=\sin(e^x)$, as plotted by the Function app. This graph was plotted using the HP 39gII "Adaptive" method rather than the traditional "Fixed-step segment" method used by most graphing calculators. For comparison, Figure 8 shows the same graph using the traditional method. The adaptive method in Figure 7 shows more clearly that the graph continues to oscillate vertically between -1 and 1, as well as giving some feel for the increasing frequency of those oscillations as x increases. The traditional method in Figure 8 indicates neither of these behaviors as clearly. While all graphical displays are limited by pixel resolution and other factors, the Adaptive method often offers more consistent clues to the nature of complicated graphs than the traditional method.

Updated programming language

The HP 39gII has an updated programming language, with support for strings, user-defined functions, and user-defined variables. User-defined objects can be exported; once exported they show up in the Commands and Variables menus just like the system commands and variables.

Figure 9 shows a simple program that defines a new function called ROLLDIE(N), which returns a random number between 1 and N. Figure 10 shows the new function appearing in the User section of the Commands menu of program functions. Figure 11 shows the ROLLDIE function returning random numbers between 1 and 6 as the results of ROLLDIE(6).

User-defined variables can be exported in the same manner. User-defined functions and variables allow you to extend the capabilities of your HP 39gII and customize it to your needs. This is a natural extension of the HP app structure!

Support for multiple languages

Figure 12 shows the Function Plot setup view. Each field in this view has its own help description that appears at the bottom of the display when the field is selected. Figure 13 shows the same view when the default language is changed to Chinese in Home Modes. Note that the field name and the help description are both translated.

Units

Units can be attached to numbers and used in calculations. There is an extensive menu of units, including most common units in length, area, volume, time, speed, acceleration, force, energy, power, pressure, temperature, etc.

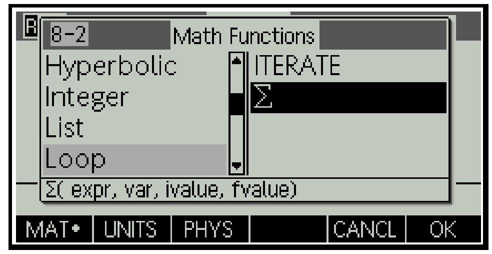


Figure 4.

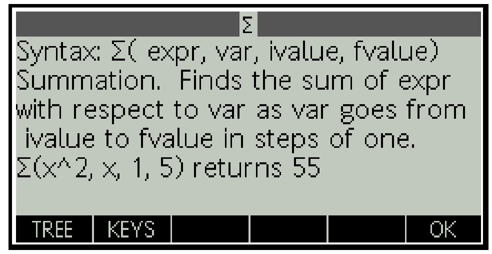


Figure 5.

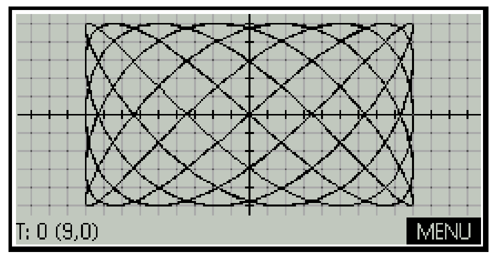


Figure 6.

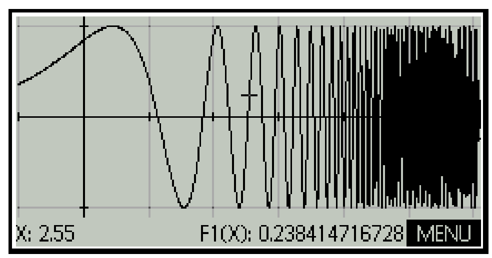


Figure 7.

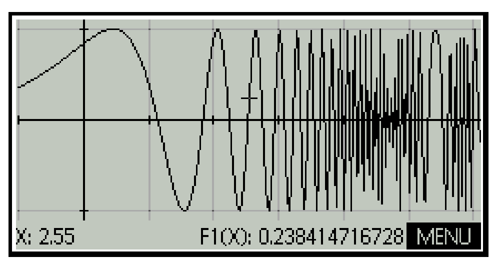


Figure 8.

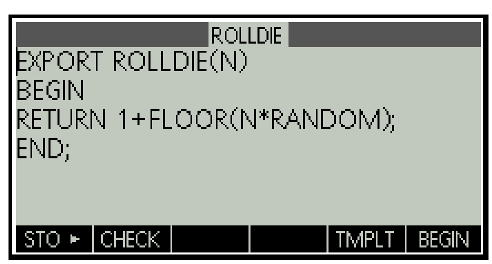


Figure 9.

Figure 14 shows the results of adding 3 meters and 27 feet. It also shows the result when the addition is commuted. The units in the result match the first units encountered in the expression.

In Figure 15, the result in feet is divided by 2 seconds to show 18.4… ft/sec. In Figure 16, this result is converted to km/hr.

App Functions

Many of the HP apps included in the HP 39gII perform specific tasks, such as solving TVM or triangle problems. The functions that perform these tasks are now visible to the user as app functions.

For example, the Triangle Solver app can solve problems involving the lengths of sides and measures of angles of triangles. If given two adjacent side lengths and the included angle measure of a triangle, this app can solve for the third side length and the measures of the other two angles. The app function SAS can be used to solve the same problem from anywhere in the calculator. Figure 17 shows SAS(8, 90, 15), defining a right triangle with Legs of length 8 and 15. The result shows the hypotenuse has a Length of 17 and the other angles have measures of 61.93° and 28.07°.

The HP 39gII is a significant addition to our graphing calculator family. It is the ideal tool for students of mathematics actively engaged in exploring mathematics and solving problems. We hope you enjoyed reading this brief introduction. Get an HP 39gII today!

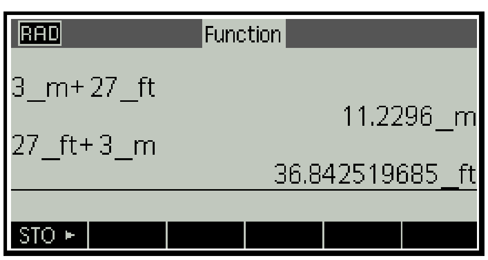


Figure 14

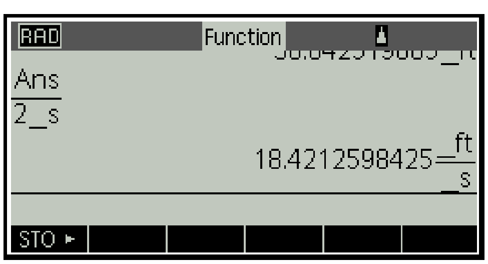


Figure 15

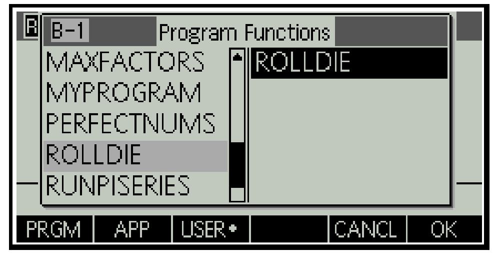


Figure 10.

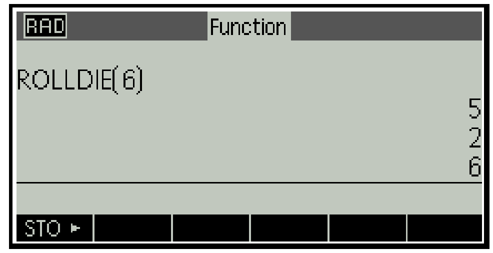


Figure 11.

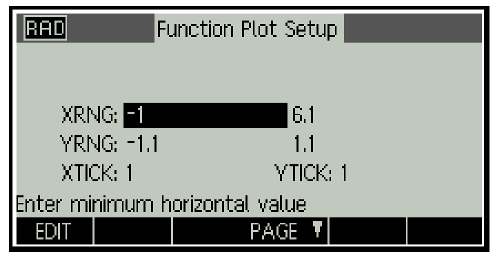


Figure 12.

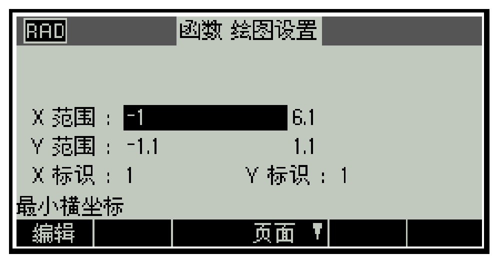


Figure 13.

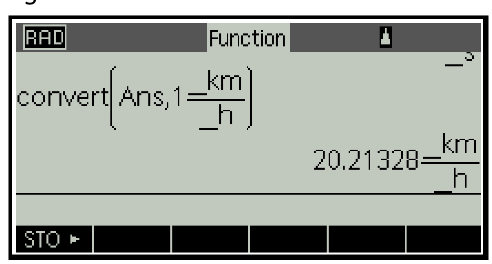


Figure 16

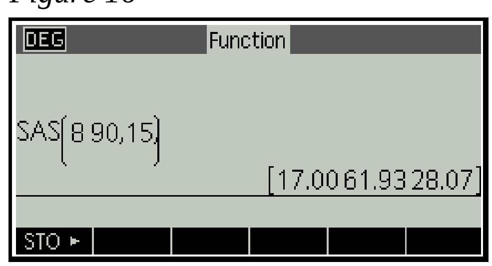


Figure 17

HP Contest - Calling all land speed chasers!

← Previous **Article** – Next →

Return to Top

Advertisement

the need for Speed!

Go Land Speed Chasers!

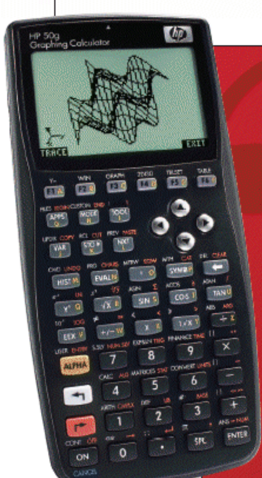
HP will hold a contest this spring challenging engineering students to submit ideas to help the NAE team reach record-breaking speeds. It's a chance to win a day in the field test running the jet-powered land rocket! For more information, email us at **calcenthusiasts@hp.com**.

Ed Shadle and his partner, Keith Zanghi, converted a Lockheed Martin F-104 Starfighter jet to a jet-powered land rocket by clipping its wings, adding a 5 point suspension system, electronics and instrumentation. With the help of innovative engineering, the North American Eagle (NAE) will break the sound barrier without ever leaving the ground! Their mission—reclaim the world land speed record for North America and break the 1997 mark of 763 mph held by the British Thrust-SSC team.

A core team of engineers, physicists, aerodynamicists and mathematicians live in the field perfecting the land speed chaser's performance from the safety of the sidelines. To beat the land speed record they must hit a speed of 771 mph on a 2-way average. Attaining that feat requires careful planning, multiple test runs and an elite set of engineering tools. The team uses the Hewlett-Packard 50g graphing calculator to analyze data gathered during test runs. This information becomes critical as the car transitions from subsonic to supersonic speeds exceeding 750 mph; creating the aerodynamic phenomenon called a 'Sonic BOOM'- broadcasting a successful journey into that supersonic realm!

The team analyzes variables like track condition hardness, rolling resistance, altitude, temperature, relative humidity, and prevailing wind conditions and then simulates airflow around the NAE through Computational Fluid Dynamics (CFD). Then as the NAE tears up the flat lands, a data acquisition system gathers performance data from over 75 channels. The team depends heavily on the HP-50g graphing calculator to quickly analyze this data, inputting pertinent variables to validate the CFD model and quickly reveal results of real-time NAE changes. "This calculator has cut days out of our optimization cycle- huge when you consider each test session exceeds $160,000." reports co-owner, K. Zanghi.

Reclaiming the world record is only one goal within their mission; the team plans to hit 800 mph if analysis enabled by the HP-50g and the laws of physics allow. So what's Ed's driving force? He says, "Perhaps it's my stubbornness, or the thrill of making the physically impossible possible. Whatever it is, my motto in life has always been, "Do what you say you are going to do!"

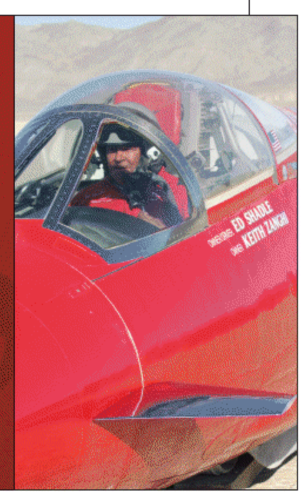


"Just like the North American Eagle that generates 45,500 hp, the HP-50g packs a lot of horsepower. It's no coincidence '50g' is prefaced with 'HP'! It is the standard in the aerospace industry, and it is the standard on Team North American Eagle. I am proud that the engineering and aerodynamics team for the North American Eagle uses HP-50g calculators exclusively."

– Keith Zanghi, Director of Operations, North American Eagle

"HP calculators have always been on the cutting edge of technology. They are designed to provide the depth and variety of capabilities that makes them a great long-term investment. They maintain consistency in their ease of use, feel and functionality. The HP-50G is the latest in this legacy."

– Mike Thoe, Project Manager of Aerodynamics, North American Eagle



Visit **www.landspeed.com** and **www.hp.com/calculators** for more information

HP is a proud sponsor of the North American Eagle project.

HHC 2011 Report

HP Solve #26 page 9

← Previous **Article** – Next →

Return to Top

HHC 2011 Report

Richard J. Nelson, Jake Schwartz, & Gene Wright

Introdction

The 38th conference[(1)] dedicated to HP Calculators, known as the Hewlett-Packard Handheld Conference, HHC, was held at the HP facility in San Diego CA September 24 & 25, 2011. Seventy four serious HP enthusiasts registered from six countries – Argentina, Canada, France, Germany, UK, and the US. Sixty nine of the registrants were present for the mid-day Saturday group photo.

photo by Joseph K. Horn

Fig. 1 – Group photo of the 2011 HHC attendees.

HHC 2011 was truly exceptional

This year was especially exciting because of the six "new" machines that were discussed. These are represented by the photo(s) shown on the cover of the proceedings. See Fig. 2. If you are an up-to-date HP user you should recognize all the machines except that shown symbolically in the second row center.

The first machine in the top row is the HP-41CL[(2)]. This project has been reported in previous issues of ***HP* Solve**. The first Beta test batch of machines were well received and an order list for the second batch is being made. We haven't heard of anyone being disappointed and users are reporting great strides in being able to have the machine contain and back up incredible amounts of software – more than any other calculator.

The top row center machine is the WP 34S. Eric Rechlin brought a hugh number of overlays to sell and donate as door prizes. I don't think he went home with any so at least 70% of the attendees have one to either put on their machine at home or put on a machine (repurposed HP 20b or HP 30b) they obtained at the Conference. Anyone who needed their calculator reprogrammed was able to have it done during the Conference. Getting set up to reprogram the newer HP calculators (20b, 30b, 15c+, 15LE, or 12C+) requires a computer that has a serial interface which is no longer standard on computers these days.

The last machine in the top row is the recently announced HP 12C 30th Anniversary Edition. This machine was well documented in the last issue of ***HP Solve***. The first two of the top row machines are HP user community created machines.

Photos by Richard J. Nelson

Fig. 2 – HHC 2011 Proceedings cover representations of the six "new" machines discussed at HHC 2011. One half of the machines are HP User community created.

The first machine in the bottom row is the recently announced HP 15C Limited Edition. A few people were able to go home with one as a door prize or purchased from a couple of people who had some to sell. This machine was also well documented in the last issue of ***HP Solve***.

The middle "machine" (symbolically) is a new machine that will be announced by HP very soon. See article elsewhere in this issue.

The last machine is the latest incarnation of home made calculators in the build-your-own-calculator series by Eric Smith. He presented a recent new high resolution display for his series of machines reported at previous HHC conferences.

The last one of the bottom row of three machines is a user community created machine. HHC 2011 is the only conference when the user community participated in the presentation of as many "new" machines as HP. Of course all of the three user community machines are based on HP machines in one way or another.

HHC 2011 was so packed with technical presentations that we had to extend the hours and maintain a strict adherence to each speakers allotted time. We tried a new method of doing this using a stop light, see Fig. 3, visible to everyone. When it turned yellow the speaker had a minute left and then the red light ended the presentation! This idea is actually an HP inspired mechanism used at their internal conferences as described by Eric Vogal at a Conference many years ago.

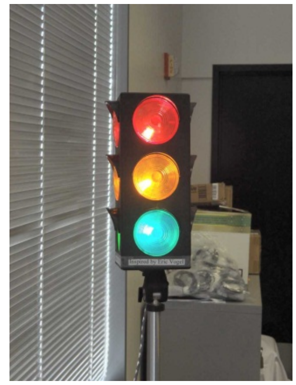


Photo by Jake Schwartz

Fig. 3 - Speaker's stoplight

Conference presentations

HHC 2011 presentations ranged from an extensive and intense confidential presentation by HP to a WP 34S keyboard overlay application demonstration video projected to the large screen. The topics varied from new HP 50g libraries to HP calculator accuracy analysis. See partial list in Fig. 4. This is from the Contents of the Conference proceedings. Many speakers only brought their presentation on a thumb drive in the form of a power point presentation.

#	Title	Author
1	HP 49G+ / 50G O.S. Extension pack.	Andreas Möeller
2	SR-60 Compared to HP Desktops.	David Ramsey
3	Scaled Reptiles of the Nordic Countries.	Eric Smith
4	A User's Perspective of the HP 41CL by Systemyde International.	Geoff Quickfall
5	An HP16C/WP 34S Dictionary.	Jake Schwartz
6	The Four Meanings of "Accurate to 3 Places."	Joseph K Horn
7	How WP 34S Came into Existence.	Marcus von Cube
8	41CL Beta Test Results.	Monte Dalrymple
9	Limited Edition HP 15c Execution Times.	Namir Shammas
10	HP-15C Benchmark: the Devil is in the Details.	Patrice Torchet
11	The PPC Rom 30 Years Later.	Richard J. Nelson
12	Generating normal Deviates.	Richard Schwartz
13	HP-12C Through the Decades.	Wlodek Mier-Jedrzejowicz

Fig. 4 – Partial list (≈60%) of the presentations given at HHC 2011. Monte Dalrymple was not able to give his presentation in person.

We had a special treat when Dennis Harms, one of the HP developers of the voyager series (especially the HP-12C) described the HP software development environment of the late 70's and early 80's. It was very clear that the tools and conditions of "then" and "now" were so different that the current team could not work under the "old" conditions and vice versa.

Best speaker

The attendees vote for the best speaker. Often the difference between first place and second place is one vote. It was very clear that Joseph K Horn deserved winning the Best speaker Award for HHC 2011. His topic was universal and very important for all HP calculator users. See **(3)** for a video link.

His presentation was well prepared and illustrated some very clever techniques for describing how accuracy may be viewed. His HHC 2011 paper may be found elsewhere in this issue.

Photo by Gary Friedman

Fig. 5 – Joseph K. Horn shows certificate.

HP Panel

One of the more important aspects of an HHC is being able to have your questions answered by the HP

Photo by Richard J. Nelson

Fig. 6– HP Q&A Panel. Left to Right: Tim Wessman, R&D; Cyrille de Brebisson, R&D; Laura Harich, Marketing; Julia Wells, Education; and Enrique Ortiz, Latin America Sales.

people who are directly involved with calculators. One hour of time was allocated and everyone was able to question, suggest, and challenge the panel. Fig. 6 shows the HP panel answering questions.

Door prizes

Door prizes[4] were another exceptional part of the 2011 Conference. HP had recently shuffled their offices and many interesting items were collected during the "clean up." These machines and other items were donated to the door prize table. The number of calculators and their variety were greater than anything we have seen at any HHC. You may see some of the door prizes in the background of Fig. 6.

Photo by Jake Schwartz

Fig. 7 – HP donated calculators.

Photo by Jake Schwartz

Fig. 8 – Opposite end of prize tables with lots of technical goodies.

The door prizes are donated by HP, the Committee, and the attendees. They are divided into two groups by the HHC Committee. The most valuable or rare items are put into a premium group – usually 5 to 9 items – by the HHC Committee[5]. See Fig. 9 below. The remainder of the prizes are in the main group as shown above. The best speaker gets first pick of this group. Contest winners then get their pick. The remainder of the prizes are selected by drawing the registration tickets at random. When every prize is given away – very close to three per attendee this year – the tickets are put back into the ticket box and everyone gets a chance for one of the premium group prizes.

Based on the video from both cameras here is what was happened. Because there were so many prizes and time was pressing we were not able to write down more detail.

1. The "regular" door prizes lasted almost exactly three full passes through the tickets....there were only a few left in the third batch when the prizes ran out.

2. We have the order and names of the premium prize winners, but we were not able to see exactly what everyone selected. Table 1 lists the winners, in order:

Photo by Jake Schwartz

Fig. 9 – Most of the premium prizes. This photo was taken early before all the prizes were drawn – 12 total.

Table 1 – List of Premium Prize Winners

1. Eddie Shore selected the HP71B
2. Egan Ford (took some sort of cable-connected device not seen in Fig. 9 - what was it?)
3. Andreas Moiller (*Germany*)
4. Mark RIngrose (*UK*)
5. Geoff Quickfall took the HP80 (*Canada*)
6. Felix Gross (*Germany*)
7. David Hayden
8. Jeff Bronfeld
9. Roger Hill
10. Howard Owen
11. Neil Hamilton
12. David Ramsey (for spouse Mary)
(3 of the last 4 prizes were the HP48GII, the HP48G+ and the HP 49G+)

Programming contest

Every HHC has to have a programming contest. We conducted an RPL RPN Programming Contest for the HP 50g (conducted by Bill Butler) and then a contest for legacy RPN machines (Gene Wright). See appendix A for the Contest details. Cyrille de Brebisson and Egan Ford won each contest respectively.

The winner of the legacy RPN contest used the WP 34s. Code:

```
001 Rv                     013 RCL 01
002 Rv                     014 X^2
003 STO 01                 015 -
004 DSE 01                 016 SQRT
005 GTO 02                 017 CEIL
006 GTO 03                 018 STO+00
007 LBL 02                 019 DSE 01
008 STO 00                 020 GTO 01
009 X^2                    021 RCL 00
010 STO 02                 022 LBL 03
011 LBL 01                 023 4
012 RCL 02                 024 x
```

The execution time for a radius of 5000 was about 28 seconds.

After the conference, solutions were posted on the HP Museum forum that were faster and for older machines. For reference the HP 67 found the answer for a radius of 5000 in about 1.4 hours.

The fastest program posted to the museum was for the WP 34S. It solved the 5000 radius problem in just under 2 seconds, as it was found that integer mode on the WP 34S worked much faster.

```
001 BASE 10                014 RCL- Y
002 RCL Z                  015 RCLx Y
003 FILL                   016 SQRT
004 STO+ Z                 017 FS? C
005 RCLx X                 018 INC X
006 2                      019 SL 1
007 /                      020 STO+ Z
008 SQRT                   021 DROP
009 INC X                  022 DSE X
010 STO Z                  023 BACK 10
011 STOx Z                 024 4
012 -                      025 RCLx Z
013 RCL T                  026 DECM
```

The HP Museum thread showing many examples of programs for various machines can be found here:

http://www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/archv020.cgi?read=197720

Observations and conclusions

The number of new machines discussed at HHC's has been decreasing in the last decade. The exception for "new" machines was made this year with a total of six machines to discuss with the actual software engineers that spirit their development. Only one developer was not present, see Fig. 4. Perhaps not all team members were able to present, but at least one team member for each machine was at HHC 2011.

The technical challenges of setting up the Conference were substantial this year because HP had several conferences of their own taking place during "our" weekend. At 1 PM on Friday it looked like we wouldn't have enough tables, chairs, or square footage. We had 88 attendees pre-registered on the website and tables for 32. Space was a bit tight as the photos show, but t these problems were solved, and the Conference was a great experience for everyone. The official HHC Hotel, The Holiday Inn, even pitched in delivering tables on Friday afternoon. These were returned to the hotel by an attendee's truck. Great work everyone!

It is the enthusiasm and problem solving attitude of all the attendees that makes our conferences unique. Will it will be possible to top 2011? Who knows, we are a long way from September 2012 so all bets are off. The future of HP calculators is indeed bright.

Notes for HHC 2011 Report

(1). To review all HHC's of this century see: hhuc.us.

To review a list of all past HHC's see: http://hhuc.us/2011/conflist.htm

(2). To get more information on the HP-41CL See ***HP Solve*** issues #24, page 35, #23 page 38, and #23 page 11.

http://h20331.www2.hp.com/Hpsub/cache/580500-0-0-225-121.html?jumpid=reg_R1002_USEN

(3) To watch Joseph's HHC 2011 talk see: http://www.youtube.com/watch?v=CYR-1jBTUa4

(4). See this link for a partial list of (non-HP) door prizes. The final number was at least seven times those on the list. http://hhuc.us/2011/Door-Prizes-2011.pdf

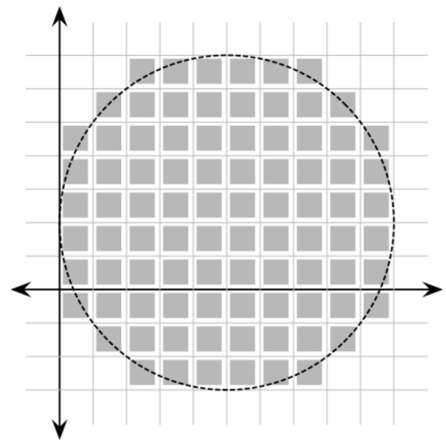
(5). The HHC 2011 Committee is comprised of the following:
Gene Wright – Registration.
Richard J. Nelson – Hotel, Speakers Schedule, Proceedings.
Joseph K. Horn – Website.
Jake Schwartz – Videographer, Historian.
Eric Rechlin – Reality checker, general helper.

Appendix A – HHC 2011 Programming Contests – Page 1 of 2

HHC 2011 Legacy RPN Programming Contest Rules

Problem Description: Did you know that if you draw a circle that fills the screen on your 1080p high definition display, almost a million pixels are lit? That's a lot of pixels! But do you know exactly how many pixels are lit? Let's find out!

Assume the display is set on a Cartesian grid where every pixel is a perfect unit square. For example, one pixel occupies the area of a square with corners (0,0) and (1,1). A circle can be drawn by specifying its center in grid coordinates and its radius. A pixel on the display is lit if any part of is covered by the circle; pixels whose edge or corner are just touched by the circle, however, are not lit



You must compute the exact number of pixels "lit" when a circle with a given position and radius is drawn.

Input: Each test case consists of three integers, x, y, and r ($1 \leq x, y, r \leq 5000$), specifying respectively the center (x, y) and radius of the circle drawn. The radius will be loaded into stack register Z, the y coordinate of the center of the circle into stack register Y, and the x coordinate of the circle into stack register X. Assume successive program runs are to be started by simply entering new values and pressing R/S. Assume that all circles fit on the display panel even if in reality they would not.

Output: Return the number of pixels that are lit when the specified circle is drawn.

Sample Cases: (A) Input of 1 ENTER 1 ENTER 1 R/S should return 4. This represents a circle with a center of (1,1) and a radius of 1. The display would have 4 pixels "on" to represent this circle. (B) Input of 5 ENTER 2 ENTER 5 R/S should return 88. This represents a circle with a center of (5,2) and a radius of 5. The display would have 88 pixels "on" to represent this circle. This is the circle shown in the figure above. 88 pixels are "on" in this picture.

Machines Eligible: This contest is open to any and all RPN machines: 15c, 15c+, 15c LE, 34S, 41CL, 42S, 67, 65, etc. RPL users are welcome to try the problem, but this is for RPN machines only.

Rules: (aka the fine print)

1) The decision of the judge is FINAL. No appeals are allowed to anyone or anything.
2) The purpose of this contest is to have fun and learn.
3) At least two contestants must submit an entry.
4) No custom built ROM or machine code can be built and used for this problem. Any already existing functionality in the machine is ok.
5) You must submit a machine with your program already keyed in to the judge AS WELL as a legible listing of your program with your name on the listing AND the machine. Machines with no names that are given to the judge are assumed to be gifts to the judge. Thank you!
6) Submission must be made by the end of the contest (Time is TBA).
7) Assume the program will start running with step 001 and/or a R/S.
8) By submitting a program, you agree to allow it to be shared with the community.
9) This is a contest between individuals, not teams. One submittal <> one person.
10) You may not access the internet for any help in any fashion. Do not cheat in any way. Do not check the HP Museum Forum either.
11) You must be present to win.
12) If a point is unclear, ask immediately. No excuses for ignorance. Clarifications will be shared with the entire group during the conference.
13) Assume default machine settings. Your program must stop with the default settings in place.
14) Winner will be the program with the fastest times over the test cases giving correct results. If in the judge's sole discretion, two entries are "about the same speed," the winner will be the shortest routine. In case of a tie, the most elegant solution (according to the judge) wins.
15) The purpose of this contest is to learn and have fun. Happy Programming.

HHC 2011 RPL RPN Programming Contest Rules

The Problem

Write a program for the HP50g in RPN Mode which takes a non-empty string of any length consisting of some or all of the 26 letters A, B, C … Z and returns, as a type 28 integer, the exact number of distinct arrangements of these letters in the string. (Permuting multiple occurrences of the same letter does not change the arrangement.)

Examples

"DEEDED"	→	20
"ANTITRINITARIAN"	→	126126000
"ABCDEFGHIJKLMNOPQRST"	→	2432902008176640000
"AAAAABBBBBCCCCCDDDDDEEEEE"	→	623360743125120

Note that these results are 6!/3!3!, 15!/3!3!3!4!2!, 20!/1!^20 and 25!/5!^5 respectively.

The Rules

1. The program must be a (self-contained) single object in user code which does not call itself by name.
2. Default flag settings (except for flag –95) are assumed and must be restored if changed.
3. The stack, apart from input and output, must be left as found.
4. The program must not contain KILL or otherwise interfere with the programmatic testing and evaluation of submissions.
5. Your program must be transferred to the judge's machine under some identifying three-letter name before the announced deadline.
6. The winning program will be the one for which size*speed (bytes*sec) is least, where the speed of execution will be determined for one or more longer input strings, probably of several hundred letters, chosen by the judge.
7. The purpose of the contest is to have fun and the decision of the judge is final.

HHC 2011

The Four Meanings of "Accurate to 3 Places"

HP Solve #26 page 20

The Four Meanings of "Accurate to 3 Places"

Joseph K. Horn

INTRODUCTION

If someone were to tell you to recite π accurately to three places, you would probably say "3.141" since those are the first three decimal places of π. But if you were asked, "What is π accurate to 3 places?" you would put your HP calculator in FIX 3 mode, press π, and reply, "3.142". Two different answers, but both are correct.

Even further, suppose somebody were to ask, "What fraction does the HP 50g return when the input is π, the display is in FIX 3 mode, and the →**Q** function is executed?" The answer is $\frac{333}{106}$. But exactly what effect does FIX 3 have on →**Q**? The answer is more complicated than the simple examples in the previous paragraph. FIX 3 tells →**Q** that the answer must not differ from the input by more than 10^{-3}. In other words, →**Q** looks for $input - output \le 0.001$. Notice the three digits after the decimal point. That's what FIX 3 tells →**Q** to look for. This is yet a third meaning of "accurate to 3 places".

A fourth meaning must be addressed: What answer does my teacher's calculator give? No matter how good my calculator is objectively, it is worse than useless if it gives answers that differ from the teacher's calculator, since that is the norm used for grading, especially if the teacher is unfamiliar with other calculator models.

Thus we have four radically different meanings of "accurate to three places", namely

1. Truncated to three decimal places.
2. Rounded to three decimal places.
3. Differing from the input by < 0.0001
4. Whatever the teacher's calculator says.

In this paper, we will examine an example for which all four meanings have different values, namely, $\sqrt{84}$.

CONTINUED FRACTIONS

To accomplish our task, we must break down $\sqrt{84}$ into its equivalent "continued fraction."

Consider the following expression of nested reciprocals:

$$1+\cfrac{1}{2+\cfrac{1}{3+\cfrac{1}{4}}}.$$

Critters that look like this are called "continued fractions." Converting continued fractions to simple fractions is easy, working from the bottom up, as you can see below. Be sure to follow each step.

$$1+\cfrac{1}{2+\cfrac{1}{3+\cfrac{1}{4}}}=1+\cfrac{1}{2+\cfrac{1}{\frac{13}{4}}}=1+\cfrac{1}{2+\frac{4}{13}}=1+\cfrac{1}{\frac{30}{13}}=1+\frac{13}{30}=\frac{43}{30}$$

To avoid this cumbersome notation, continued fractions are usually written as a list containing the leading integer and then the denominators. For example, the above continued fraction is written as [1, 2, 3, 4]. The numbers in the list are called the "partial quotients" of the continued fraction.

Everybody knows that the square roots of non-square integers are non-repeating, non-terminating decimal numbers, right? Amazingly, square roots are all repeating continued fractions! The partial quotients can be seen to fit a repeating pattern. For example, $\sqrt{84}$ = [9, 6, 18, 6, 18, 6, 18, 6, 18, …] repeating forever.

Some other surprising continued fractions are:

The golden ratio $\left(\frac{\sqrt{5}+1}{2}\right)$ = [1, 1, 1, 1, 1, 1, …]. This is the simplest possible continued fraction.

e = [2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, 10, 1, 1, 12, …]

tan(1 radian) = [1, 1, 1, 3, 1, 5, 1, 7, 1, 9, 1, 11, 1, 13, …]

Unfortunately, not all irrational numbers have continued fractions with reapeating partial quotients. For example, π = [3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, 2, 1, 1, 2, 2, 2, 2, 1, 84, 2, 1, 1, 15, 3, …]. It never terminates, but it never repeats.

ITERATIVE FRACTION GENERATION

There is a truly marvelous method for generating approximate fractions for any number. As an example, let's find the best fractions that approximate π.

First you make a list of the continued fraction of π, stopping at the first large number (let's stop at 292). Then you create a table with three rows. Fill in the top row with the list of partial quotients (see the red numbers above) and fill in leading 0's and 1's like this:

		3	7	15	1	292
0	1					
1	0					

The bottom two rows represent the fractions that approximate π, beginning with 0/1 (zero) and 1/0 (infinity). As we proceed, the fractions will get closer and closer to π with amazing rapidity.

Fill in the boxes with this pattern: Use the first and second row to get $3 \times 1 + 0 = 3$ and fill it in here:

		3	7	15	1	292
0	1	**3**				
1	0					

Now do the same thing with the first and third row: $3 \times 0 + 1 = 1$ and fill it in there:

		3	7	15	1	292
0	1	3				
1	0	**1**				

Thus our first fraction approximating π is 3/1. Not very impressive, but it gets better quickly. Following the same pattern as before, do these two calculations and fill them in: $7 \times 3 + 1 = 22$ and $7 \times 1 + 0 = 7$:

		3	7	15	1	292
0	1	3	**22**			
1	0	1	**7**			

Thus π is approximately 22/7, the approximation they taught us in school. Do the next column: $15 \times 22 + 3 = 333$ and $15 \times 7 + 1 = 106$:

		3	7	15	1	292
0	1	3	22	**333**		
1	0	1	7	**106**		

This is surprising! 333/106 is a better approximation than 22/7, but nobody ever mentions it! Continue the process: $1 \times 333 + 22 = 355$ and $1 \times 107 + 7 = 113$:

		3	7	15	1	292
0	1	3	22	333	**355**	
1	0	1	7	106	**113**	

Ah yes, we've all heard of 355/113, which is even better than 3332/106. Now do the final column: $292 \times 355 + 333 = 103993$ and $292 \times 113 + 106 = 33102$:

		3	7	15	1	292
0	1	3	22	333	355	**103993**
1	0	1	7	106	113	**33102**

Thus our final approximation is 103993/33102.

OUR CHALLENGE

$$\sqrt{84} = 9.165\ldots$$

Row	Fraction	Continued Fraction	Approx.	Error
1	9 / 1	[9]	9	.1651514
2	37 / 4	[9, 4]	9.25	.0848486
3	46 / 5	[9, 5]	9.2	.0348486
4	55 / 6	[9, 6]	9.1666667	.0015153
5	504 / 55	[9, 6, 9]	9.1636364	.0015150
6	559 / 61	[9, 6, 10]	9.1639344	.0012170
7	614 / 67	[9, 6, 11]	9.1641791	.0009723
8	669 / 73	[9, 6, 12]	9.1643836	.0007678
9	724 / 79	[9, 6, 13]	9.1645570	.0005944
10	779 / 85	[9, 6, 14]	9.1647059	.0004455
11	834 / 91	[9, 6, 15]	9.1648352	.0003162
12	889 / 97	[9, 6, 16]	9.1649485	.0002029
13	944 / 103	[9, 6, 17]	9.1650485	.0001028
14	999 / 109	[9, 6, 18]	9.1651376	.0000138

Table 1

Table 1 shows the first 14 fractions that approximate $\sqrt{84}$. Which one best approximates $\sqrt{84}$ accurate to three places?

(1) If "accurate to three places" means "differing from $\sqrt{84}$ by less than 0.001," then we look down the last column for the first "error" starting with *three zeros*. We find it in row 7. Therefore, the correct answer is 614/67.

(2) If "accurate to three places" means "displaying the same in FIX 3 mode," then we look down the "Approx." column for the first value that *rounds* to 9.165. We find it in row 9. Therefore, the correct answer is 724/79.

(3) If "accurate to three places" means "having exactly the same three digits, truncated," then we look down the "Approx." column for the first value that begins with *exactly the digits* 9.165. We find it in row 13. Therefore the correct answer is 944/103.

(4) If "accurate to three places" means "Whatever the teacher gets," then we look down the "Continued Fraction" column for the first whole subset of partial quotients below the answer obtained in (1) above. (!!!) This is the answer obtained by every calculator on the planet (except for the HP-33s and HP-35s, and the HP 49/50 running the PDQ algorithm, none of which are used by teachers). We find this entry in row 14, which has the complete subset of partial quotients [9, 6, 18]. Therefore the correct answer is 999/109.
Note: If the teacher actually has an HP-33s or HP-35s, or an HP 49/50 with PDQ in it, then they will understand the above already and will give full marks to any student who obtains any of these "correct" answers.

IMPLEMENTATION

Calculator designers who decide to include a "fraction button" are therefore faced with a difficult choice (unless they are ignorant of it). Which of these four meanings of "accurate to four places" should be implemented in their calculator? HP's RPL models feature a function called **→Q** ("to Quotient"). It turns a decimal number into a fraction whose accuracy is controlled by the display's FIX setting, as we saw in the second paragraph of this article. In other words, it uses definition (1) above. So does the PPC ROM's "DF" ("Decimal to Fraction") program.

The September 2011 issue of ***HP Solve*** features an HP-15C program that converts decimals to fractions using definition (2) above. The FIX setting is used to round the input and the output until they are the same. No other calculator or program does this, to my knowledge.

Ordinary people use definition (3) above. That's why nobody implements it.

The HP 32SII uses definition (4) above. So do all non-HP calculators that have a fraction button with user-controllable accuracy.

My suggestion to calculator designers is to follow definition (2) and use the FIX setting to control the accuracy. When the rounded input equals the rounded output, stop. The amount of required math is roughly the same as the other definitions above, and is easily implemented.

Joseph K. Horn
joehorn@holyjoe.net

About the Author

Joseph K. Horn is a high school math teacher in Orange County California. He has authored many articles related to HP calculators in addition to his book titled, HP-71 Basic Made Easy. He is sometimes known in the HP user community as "The math bug hunter."

Joseph serves on the HP Handheld conference, HHC, committee and is web-master for the HHC website.

What is Double Injection Molding of HP Calculator Keys?

HP Solve #26 page 26

What is Double Injection Molding of HP Calculator Keys?

Richard J. Nelson

When HP started the scientific (*and financial*) calculator business in 1972 the typical engineering approach that was applied to its instrument products was also used for designing its calculators. Because these were unique products there was literally no competition and no other products to which the HP-35A (*or HP-80A*) could be compared. The calculators were intended to be used hand held and the industry instrument quality standards[(1)] of the time were used. It was well known[(2)] that many people have highly acid perspiration and skin oils and the key top notations would need to be as wear resistant as possible. The key wear issue was addressed mechanically with a key design using double injection molding which used two different colored plastic for the key. One color was used for the overall key color and a second color was used for the key notations.

The idea behind double injection molding was that as the keys were used and became worn the notations could not be "worn off" because they extended deep into the plastic key.

Fig. 1 shows a 1979 HP-41C ENTER key that has had three holes cut with an end mill cutter to provide a flat hole bottom. The holes are identified A, B, & C.

Fig. 1 – Three holes "drilled" into HP-41C ENTER Key,

A – A very shallow cut.

B – A deeper (0.028", >1/4th of an inch) cut showing that the white lettering plastic extends into the key and is not just on the surface. See Fig. 2.

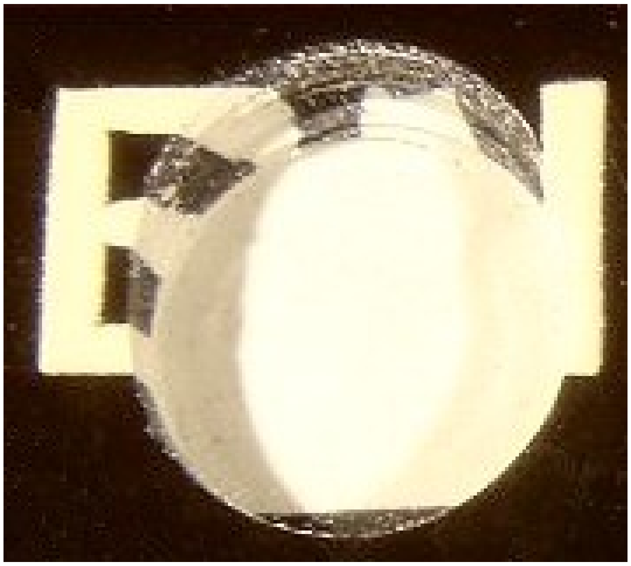


Fig. 2 – Closer view of the "B" cut.

Fig. 3 – Closer view of the "C" cut.

C – A second deep (0.028" ", >1/4th of an inch) cut showing that the blue lettering is only "painted" on the sloping front surface of the key. This surface is not normally touched by the fingers. See Fig. 3.

When other calculator manufactures entered the market HP had to better compete and the very costly double injection molding process for calculator keys had to give way to newer epoxy paints for the key notations. A host of newer technology manufacturing processes[(3)] are used thirty three years later and calculator costs have come down so that students can better afford the latest technology of greatly increased functionality.

HP Calculators are high quality and every calculator collector, no matter what category or manufacturer they collect, will have a few HPs in their collection. The oldest HP calculators are 41 years old and many users still use them. Many "old timers" lament the loss of double injection molding, but the question that must be asked is: How many customers are willing to pay for calculators the would cost many times what is available in the market place for this very expensive plastics process? HP has wisely kept up with the key making technology with their current manufacturing processes.

It may be of interest for new customers to know a bit of history with the current interest/popularity of legacy models such as the HP-41C, HP-12C, and HP-15C, and to know what is meant by double injection molding. The photos above should provide the answer to, "What is double injection key molding?"

<u>Legacy user trivia questions.</u>

(1) Has any other handheld calculator manufacturer made a calculator with double injection key molding?

(2) What was the last HP calculator model that used double injection key molding?

(3) Has there ever been a triple injected (three color) molded handheld calculator key?

Notes: What is Double Injection Molding of HP Calculator Keys?

(1) This was before the category of personal instrumentation products came into existence. Instruments were sold to be used "forever" and they were built for heavy duty, often field, use. HP refused to comply with costly US Government regulations to make special ruggedized instruments for the military because they were rugged enough as normally designed and manufactured.

(2) Museum curators had long before learned this because of the acid wear of "touchable" exhibits by the general public. Joseph K. Horn, an active HP calculator user, is one of those people who apparently have strong acid skin oils and his most frequently used newer calculators reflect this unusual form of wear.

(3) In the early days the keys were individually molded. Later they were molded together in one shot to reduce assembly time, errors, and cost. Not all keys were required to be double injected molded, and normal manufacturing technology advancements brought the inevitable changes to making calculator keys.

Timing for HP 35s Calculator Instructions

HP Solve #26 page 29

← Previous **Article** – Next →

Return to Top

Timing for HP 35s Calculator Instructions

Richard Schwartz

Motivation

My original goal was to develop a normal deviate generator for use in some simple simulation and optimization problems. But as I worked, I became aware of my ignorance as to what was fast and what was slow on the HP-35s. Hence this effort to uncover the execution times of various instructions. When I started this unfinished task, I grossly underestimated the effort required. Hopefully, others will be able to apply these methods to discover how to make machines like the HP-35s and the HP-15C work harder for us.

Methods

Time Measurement: Alas, the HP-35s, although an electrical power engineer's dream machine, has no clock function. Time must be measured with an external stopwatch or by listening to ticks from WWV. The method I use is to include the instruction being tested in a loop and count the number of loops in a fixed time (usually about five minutes). I do this by starting the stopwatch, and then starting the calculator when the stopwatch reaches ten seconds. When the stopwatch reaches the desired time, I stop the calculator and retrieve the loop counter. How accurate is this? I cleared the machine with a simultaneous press of **C**, **R/S**, and **i**, entered the code in listing 1, and ran the empty loop fourteen times for twenty seconds. The counts in the B register averaged 600.5, with a standard deviation of 2.8. That gives a three sigma (99.7%) error limit of about 280 msec. You may wish to try different stopwatch techniques to see what works best for you.

```
LBL B   CLX   STO B
R/S                          (press R/S when watch gets to 10 seconds)
DEG                          (DEG here is a minimally invasive entry point)
ISG B  SF 4                  (B counts the loops; SF4 is never executed.  If you see
GTO B005                       flag 4 set, check your code or get a new machine!)
```

Listing 1: empty loop

Recording Your Work: I find that a simple spreadsheet is useful for recording the probing programs and their runtimes. Use the spreadsheet to reduce the chance of blunder in the calculations of instruction execution times as described below.

Probe Programs: To measure the time of an instruction, you could put it into a program of the form LBL A, instruction, R/S. Go to A, and press R/S as your stopwatch starts. There are some problems with this method. First, the time for the LBL A and the R/S instructions are included in the measured time. Second, the elapsed time is going to be too short to be measured with a stopwatch.

A better way, suggested by Richard Nelson, is to insert hundreds of copies of the instruction being tested. This dilutes the time taken by the LBL A and the R/S, but it is a lot of work to enter hundreds of repetitions. Furthermore, some tests require setup instructions for the instruction being probed. As a compromise, I use fifty repetitions in a loop and count the number of loops that were executed. The instruction time is the loop execution time of the program, less the loop time without the instruction being probed, divided by fifty. (A simple spreadsheet does this.) Instructions may have two or more execution times depending on stack lift, stack drop, condition being tested, or argument of a function.

Stack Drop Dependence: Modify listing 1 to get listing 2. This will evaluate the time that addition via the + operation takes. Note that this involves dropping the stack and copying the top of the stack.

LBL B CLX STO B	
c STO A	(get the speed of light from the CONST menu)
ENTER ENTER ENTER	(stuff the stack)
R/S	(press R/S when watch gets to 10 seconds)
DEG	(DEG here is a minimally invasive entry point)
+ (x50)	(instruction being evaluated . x50 means fifty occurrences.)
ISG B SF 4	(B counts the loops; SF4 is never executed. If you see
GTO B005	flag 4 set, check your code or get a new machine!)

Listing 2

The result for listing 1 was 8930 loops in 300 seconds, or 33.6 msec per empty loop. The result for listing 2 was 508 loops in 360 seconds, or 708.7 msec per loop. The fifty **+** operations added 708.7-33.6 = 675.1 msec per loop, or 13.50 msec per + operation. What if each of the **+** operations in listing 2 is replaced with RCL+A? That resulted in 380 seconds for 751 loops. The instruction time was 9.45 msec, a lot faster when there is no stack management. This was first noted in 1974. See reference [1].

Stack Lift Dependence: An instruction that may lift the stack or not, depending on the state of the stack lift is RCL. If it is preceded by a CLX, it does not lift the stack. If it is preceeded by another RCL, it will lift the stack.

LBL B CLSTK STO B	
R/S DEG	(DEG is used as a minimally invasive entry point)
CLX (x50) ISG B	(here, x50 means that the instruction appears 50 times)
DEG GTO B005	(DEG is used as a no-op here)

Listing 3

This ran for 420 seconds and executed the loop 3027 times, for 138.8 msec per loop. (Note that this is exactly the program needed to time the CLX instruction, which appears in the table below.) Next, insert a RCL B after each CLX in the above program and time it again to get 1396 loops in 420 seconds, for 300.9 msec per loop. The time added by the RCL instruction was 3.24 msec per occurrence. To see what happens when stack lift is enabled, remove the fifty CLX instructions and run the program again. I ran it 400 seconds with 1443 loops. This works out to 4.87 msec per RCL instruction, showing the additional 1.63 msec required for stack lift.

Entry of Numbers: For the entry of large numbers, a different timing method was needed. Fifty entries of -5.555555555E-55 could not be interrupted by the R/S key, so the instruction time could not be accurately determined. At first, I thought the R/S key was malfunctioning, but it always operated reliably to START the program. To time data entry I set a limit of 600 loops with an ISG counter (see listing 4), started the stopwatch as described above, started the calculator at 10 seconds, and stopped the stopwatch when the calculator RUNNING display ended. After several attempts, I knew when to expect the end of the program and got a reasonably accurate time. See Table 1 for the result.

```
LBL B  .600   STO B              (set up B for 600 loops)
R/S DEG                          (DEG is used as a minimally invasive entry point)
-5.555555555E-55 ( x50)          (number entry time being probed)
ISG B
GTO B005  R/S                    (DEG is used as a no-op here)
```

Listing 4

Conditional Branch Instructions

Branching instructions have two different execution times, depending on the truth value of the condition being tested. This was recognized in 1974; see reference [2]. Using the above program, we insert 50 X=0? DEG pairs. The DEG instructions are not executed because the value in X is always k.

```
LBL B   CLX   STO B
k                          (Boltzmann's constant from CONST menu)
ENTER  ENTER  ENTER        (stuff the stack with k)
R/S   DEG                  (DEG is used here as an entry point with minimal side effect)
X=0?  DEG (2x50)           (2x50: the two instruction sequence occurs 50 times)
ISG B  DEG   GTO B010      (DEG is used as a no-op here; it is never executed)
```

Listing 5

In listing 5, we examine the instruction time when the condition is false. Each X=0? DEG pair requires five keystrokes; it takes a long time to enter, and there is some chance of error. It is wise to review all of the code before running it. This program performed 1614 loops in 300 seconds, for a loop time of 185.9 msec, for a time of 156.1 msec for fifty instructions, and 3.12 msec per instruction.

To see what happens when the X=0? test is true and the test does not skip the next instruction, remove the k instruction from the preamble and the fifty DEG instructions from the loop. (The calculator will automatically adjust the final GTO instruction.) This results in 2111 loops in 420 seconds, for an instruction time of 3.38 msec, significantly different from the false condition.

Battery Calibration

The running speed of the HP-35s is variable, depending on battery voltage and possibly on other unknown variables such as temperature. Instruction timing results must be adjusted for these unknowns so that users of different machines with different batteries can add to these results and so that you can compare results on your own machine tomorrow, when the speed will have changed. To adjust for the speed of the machine, I make the simplifying assumption that all calculator operations are affected in the same proportions by differences in battery and temperature. I used the program in listing 1 and assigned a standard time of 33 msec per loop, so the calibration factor is 33/loop time. Multiply all timing measurements by this factor. You should include a battery calibration run during every work session. The calibration factor seems to be less than 1 when the batteries are new and greater than 1 when they are old.

A Few Surprises

The results of the above examples are summarized in the table. I am still working on the complete list. In the instructions I have timed so far, there is a lot of commonality with the HP-67/97 Better

Programming booklet. [3] One surprise is 3091 vector additions in a minute, only 4441 real number additions in a minute. Vector additions go twice as fast, so some parallel processing might be happening, or possibly some overhead is being shared or saved. Some exploration needs to be done to find ways of vectorizing problems. Most math functions do not work on vectors.

Approximate Timing Summary

Instruction	Time	Loops	Time/Loop	Time/Instruction	Calibrated
None	300	8930	33.6		
+	360	508	708.7	13.50	13.26
RCL+	380	751	506.0	9.45	9.28
CLX	420	3027	138.8	2.10	2.07
RCL without lift	420	1396	300.9	3.24	3.18
RCL with lift	400	1443	277.2	4.87	4.79
X=0? False	300	1614	185.9	3.05	2.99
X=0? True	420	2111	199.0	3.31	3.25

Another surprise was that addition and subtraction are no faster than multiplication. Historically, computers multiplied by a shift, bit test, and add procedure that took 32 machine cycles for a 32 bit word. This disparity has had a major impact on twentieth century numerical methods that make every effort to avoid multiplication and division. Perhaps new methods will evolve to take advantage of the repeal of the multiplication penalty.

And another surprise was the slow execution of entering a numerical constant. This was also an issue in 1978. See the top of page 7 of reference [3]. Some things have not changed in forty years of advancing technology at HP! If you want a program to run fast, you should pre-store all numerical constants and recall them from memory. Use store and recall arithmetic; it is faster than stack arithmetic because the machine does not have to manage the stack. Recalling constants from the built in physical constants list is incredibly fast; it would be nice if this list could include commonly used constants like 0, 1, 2, ½, 2 π, 1/Sqrt(2 π), Sqrt(e), and more that might be gleaned from a survey of user programs. HP has demonstrated the ability to resurrect old machines long after the team that developed them has departed; let us hope that the future is bright this time.

References

[1] 65 Notes, v1n1p2, June 1974.

[2] 65 Notes, v1n2p2, July 1974.

[3] Kolb, Kennedy, & Nelson, Better Programming on the HP-67/97, PPC 1978.

About the Author

Richard Schwartz is a "retired" electronics Engineer from Southern California who spends a lot of his time analyzing the stock market and analyzing supports for telescope mirrors. He has many hobbies including Amateur Radio.

He enjoys advanced math subjects and is a frequent speaker at HHC Conferences. His HHC 2011 presentation was titled Generating Normal Deviates covering the subject in just 11 pages with a tremendous Power Point presentation that kept even the non-math people in the audience interested.

Octal Fraction Conversions

HP Solve #26 page 34

Octal Fraction Conversions

Palmer O. Hanson

Introducton

In 1960 I was a field service engineer for Honeywell supporting an inertial system which was installed on the SD-5 surveillance drone being developed by Fairchild for the Army. The system used the M-252 airborne computer manufactured by Hughes. Tasks associated with use of the computer were the conversion of octal fractions to decimal fractions and the reverse. We did the conversions using time consuming and error prone methods by hand calculation or on a Friden. Then in 1961 while I was supporting our systems at the Fairchild Electronics Systems Division plant in Wyandanch, Long Island I was introduced to a set of tables which made the conversions easier and less error prone. The tables were originally compiled by a Mr. Jack Roy Morris of American Bosch Arma and bear a 1959 copyright. The document includes nine pages of tables for conversion from octal to decimal and eighteen pages of tables for conversion from decimal to octal. Figure 1 is a sample page from the document. This paper reviews hand calculation techniques for conversion, illustrates conversions using the Arma tables and discusses methods for conversion using modern hand-held calculators.

Decimal To Octal Fractional Conversions

Suppose that you had the decimal fraction 0.169148123 and you needed to convert it to an octal fraction for use in the M-252 computer. There is nothing special about this number other than that the table entries which will be used later in the analysis all appear on the single page of the table which is attached. The accepted way to do the conversion back in the time, with a Friden or by hand, was to

1. Multiply the decimal fraction by 8.

2. Subtract any integer part of the result but save it for use in the octal equivalent.

3. Repeat the process as many times as needed.

For the decimal fraction 0.169148123 the sequence yields

0.169148123 x 8 = 1.353184984
0.353184984 x 8 = 2.825479872
0.825479872 x 8 = 6.603838976
0.603838976 x 8 = 4.830711808
0.830711808 x 8 = 6.645694464
0.645694464 x 8 = 5.165555712
0.165555712 x 8 = 1.324445696
0.324445696 x 8 = 2.595565568
0.595565568 x 9 = 4.764524544
0.764524544 x 8 = 6.116196352
0.116196352 x 8 = 0.929570816
0.929570816 x 8 = 7.436566528
0.436566528 x ...

and the octal equivalent assembled from the integer parts of the results of the multiplications is 0.126465124607 ....

When the tables of the reference were available the user wrote down the three octal equivalents for the three 3-digit segments of the decimal number from Table 1 as follows

Table 1: Excerpts from the Decimal to Octal Tables in the Reference:

Decimal To Octal Fraction Conversion Table
Accuracy 11 And 2*3 Places

Decimal Fraction	Octal Equivalents		
N	N	N x 10^-3	N x 10^-6
.121	.075747331054	.000037560260	.000000020172
.122	.076355442640	.000037766440	.000000020276
.123	.076763554426	.000040174616	.000000020404
.147	.113207126010	.000046422002	.000000023564
.148	.113615237574	.000046630162	.000000023672
.149	.114223351360	.000047036342	.000000023776
.168	.126010142232	.000054024450	.000000026430
.169	.126416254020	.000054232626	.000000026534
.170	.127024365604	.000054441006	.000000026642

Octal equivalent of 0.169 = 0.126416254020
Octal equivalent of 0.000148 = 0.000046630162
Octal equivalent of 0.000000123 = 0.000000020404

Octal equivalent of 0.169148123 = 0.126465124606

where the difference in the last digit is due to the truncation in the table. In those days we found that some individuals had difficulty adding octal numbers. For such individuals we recommended that they convert the octal fractions to digital fractions, adding the digital fractions and converting the digital sums back into octal; e.g.,

0.126416254020 = 0. 001 010 110 100 001 110 010 101 100 000 010 000
0.000046630162 = 0. 000 000 000 000 100 110 110 011 000 001 110 010
0.000000020404 = 0. 000 000 000 000 000 000 000 010 000 100 000 100

Digital Sum	0. 001 010 110 100 110 101 001 010 100 110 000 110
Octal Equivalent	0. 1 2 6 4 6 5 1 2 4 6 0 6

In the olden days hand-held calculators were not available and access to mainframe computers was not easy to attain. With a modern hand-held calculator it is easy to write a program which can perform the iterative sequence which was used by hand or on a Friden; e.g., with an HP-35s one can use

```
A001 LBL A          A014 RCL E
A002 INPUT D        A015 INTG
A003 STO E          A016 STO- E
A004 INPUT N        A017 RCL A
A005 STO A          A018 INTG
A006 1000           A019 10^X
A007 STO/ A         A020 /
A008 1              A021 STO+ O
A009 STO+ A         A022 ISG A
A010 0              A023 GTO A012
A011 STO O          A024 RCL O
A012 8              A025 STOP
A013 STOx E
```

The decimal fraction is entered in response to the prompt "D?" The number of digits in the octal equivalent is entered in response to the prompt "N?" The program stops with the octal equivalent in the display, but note that the calculator is NOT in octal mode. For the problem at hand enter 0.169148123 on response to "D" and 9 in response to "N" and see 0.126465124 as the result.

I have not been able to do a keyboard conversion from a decimal fraction to an octal fraction directly with the conversion capabilities on any of the machines that I have in my possession; e.g., the the HP-16C, HP-28S, HP-32S, HP-33S, HP-35S HP-48S, TI-Programmer, TI-85, TI-86, Casio fx-7000G, Casioi fx115, Casio Fx 115D and Casio fx-115ES because those machines only do integer conversions. A fairly efficient conversion can be made with keyboard sequences on machines which have a binary arithmetic capability with sufficient word length.

For machines such as the HP-16C, HP-28S and HP-48S which carry 64 digits and the problem under consideration here one possible procedure is to enter the nine digit decimal fraction as a nine digit integer, convert it to an octal integer, multiply the octal integer by an octal integer 1000000000 (nine zeroes), enter the decimal integer 1000000000 (nine zeroes), convert it to an octal integer and divide. For the HP-28S a possible sequence is

1. Press 2nd BINARY to place the desired menu at the bottom of the screen.

2. Press OCT in the menu. See a box in the OCT label indicating octal as the current base.

3. Press 169148123 R>B (in the menu). See # 1205177333o at level 1

4. Press # 1000000000 (nine zeroes) X . See # 12051773330000000... at level 1

5. Press 1000000000 (nine zeroes) R>B . See # 7346545000o at level 1 and # 12051773330000000... at level 2.

6. Press / and see # 126465124o which are the first nine digits of the octal equivalent of decimal 0.169148123.

To see twelve digits as in the result from the tables use twelve zeroes instead of nine zeroes in the fourth step. A nearly identical sequence can be used with the HP-48S but I find it slightly less convenient since the # is a second function on that machine. A similar procedure can be used with the HP-16C but I find the necessity to scroll back and forth due to the limited display length to be a nuisance. Similar sequences can also be used with the TI-85 and TI-86. I was surprised to find that the TI-89 does not support octal calculations but does support hexdecimal calculations. That brought to mind a minor irritation with the use of hexadecimal code on the Fairchild drone program. Hughes used a through f for 10 through 15 in the M-252 documantation. The Army used U through Z for 10 through 15 for the interface with other systems.

For machines such as the HP-33s and HP-35s which carry only 32 digits I had to settle for solving for fewer digits. (I am not saying that more digits cannot be obtained, but only that I haven't figured out how to do it.) For the HP-35s in RPN mode a possible sequence is:

1. Press BlueShift BASE 3 to set octal mode.

2. Enter 16914 (five digits) and press ENTER . See 41022o in the display.

3. Enter 100000 (five zeroes) in the display which would be a decimal number.

4. Press BlueShift BASE 7. See 100000o in the display.

5. Press X and see 4102200000o in the display.

6. Enter 100000 (five zeroes) in the display which again is a decimal number.

7. Press / and see 12646o in the display which shows the first five digits of the octal equivalent of decimal 0.16914. .

Octal To Decimal Fractional Conversions

Similar procedures can be used to convert from octal fractions to decimal fractions. Consider the octal fraction 0.077341706 The accepted way to do the conversion by hand, back in the time the tables were in use, was to

1. Multiply the octal fraction by 12o.
2. Subtract any integer part of the result but save it for use in the decimal equivalent.
3. Repeat the process as many times as needed.

For the octal fraction 0.077341706 and operating in octal the sequence yields

0.077341706 x 12 = 1.172322674
0.172322674 x 12 = 2.310074530
0.310074530 x 12 = 3.721136560
0.721136560 x 12 = 11.053663140
0.053663140 x 12 = 0.666377700
0.666377700 x 12 = 10.440776600
0.440776600 x 12 = 5.511763400
0.511763400 x 12 = 6.343603000

0.343603000 x 12 = 4.345436000
0.345436000 x 12 = 4.367454000
0.367454000 x 12 = 4.653670000

and the decimal equivalent assembled from the discarded integer parts to 11 decimal places is 0.12390856444. Multiplying two octal numbers by hand is easier said than done. For the particular problem under consideration here of multiplying a nine digit octal fraction by 12o some individuals found it easier to obtain the desired result with octal addition by writing the octal fraction twice followed by the octal fraction with the octal point moved one place to the right and adding. For the sequence above the solution would be

0.077341706
0.077341706
0.77341706

1.172322674

0.172322674
0.172322674
1.72322674

2.319974539

and so on.

When the tables were available the user wrote down the three decimal equivalents for the three 3-digit segments of the octal number from Table 2 as follows

Decimal equivalent of 0.077 = 0.1230468750
Decimal equivalent of 0.000341 = 0.0008583069
Decimal equivalent of 0.000000706 = 0.0000033826
Decimal equivalent of 0.077341706 = 0.1239085645

Table 2: Excerpts from the Octal to Decimal Tables in the Reference

Octal To Decimal Fraction Conversion Table
Accuracy Ten Places Rounded

Octal Fraction	Decimal Equivalents		
N	N	N x 8^-3	N x 8^-6
.076	.1210937500	.0002365112	.0000004619
.077	.1230468750	.0002403259	.0000004694
.100	.1250000000	.0002441406	.0000004768
.340	.4375000000	.0008544922	.0000016689
.341	.4394531250	.0008583069	.0000016764
.342	.4414062500	,0008621216	.0000016838

.705	.8847656250	.0017280579	.0000033751
.706	.8867187500	.0017318726	.0000033826
.707	.8886718750	.0017356873	.0000033900

An HP-35s program which can perform the iterative sequence which was used by hand is

```
B001 LBL B          B018 ENTER
B002 DEC            B019 777777777o     (nine sevens)
B003 INPUT O        B020 AND
B004 STO E          B021 STO E
B005 INPUT N        B022 Roll Down
B006 STO A          B023 1000000000o    (nine zeroes)
B007 1000           B024 /
B008 STO/ A         B025 DEC
B009 1              B026 RCL A
B010 STO+ A         B027 INTG
B011 0              B028 10^x
B012 STO D          B029 /
B013 OCT            B030 STO+ D
B014 12o            B031 ISG A
B015 STOx E         B032 GTO B013
B016 RCL E          B033 RCL D
B017 ENTER          B034 STOP
```

where there is some added complexity relative to the decimal to octal program due to the inability of the HP-35s (and every other calculator in my inventory) to work with fractions in octal mode. In response to the prompt "O" the octal fraction is entered as if it had been multiplied by 1000000000o (nine zeroes). For the problem at hand enter 77341706o where the o is entered by pressing Blue Shift 7.. The number of digits of the desired equivalent decimal fraction is entered in response to the prompt "N". For the problem at hand enter 9. The program stops with the decimal equivalent 0.123908564 in the display.

As with decimal-to-octal conversions more direct octal-to-decimal conversion can be made with machines which have a binary arithmetic capability; however, word length is not the issue. The issue is with decimal-to-octal conversions. One possible procedure is to enter the digits of a nine digit octal fraction as a nine octal digit integer, convert it to an decimal (but NOT a binary integer) enter 1000000000o (nine zeroes), convert it to a decimal, and divide. For the problem under consideration here, i.e., conversion of .077341706o to decimal, a possible keyboard sequence for the HP-28s is:

1. Press 2nd BINARY to place the desired menu at the bottom of the screen.

2. Press OCT in the menu. See a box in the OCT label indicating octal as the current base.

3. Press # 77341706 (0.077341706o multiplied by 1000000000o) B>R . See 16630726 at level 1 .

4. Press # 1000000000 (nine zeroes) B>R . See 16630726 at level 2 and 134217728 at level 1.

5. Press / . See ..123908564448 which is the decimal equivalent of 0.07734176o to 12 decimal places.

That uses 25 key srokes. A nearly identical sequence may be used with the HP-48s. For the HP-35s a possible keyboard sequence is:

1. Press Blue Shift BASE 1 to set decimal mode.

2. Press 77341706 Blue Shift BASE 7 . See 77341706o in the lower level of the display. .

3. Press ENTER . See 16,630,726 in the lower level of the display.

4. Press 1000000000 Blue Shift BASE 7 . See 1000000000o in the lower level of the display and 16,630,706 in the upper level of the display.

5. Press / . See 0.12390856 in the displayif the machine isin FIX 9 display mode. That is the decimal equivalent of 0.077341706o to 8 decimal places That uses 29 keystrokes...

A more efficient solution (i.e., fewer keystrokes} can be obtained using the little LeWorld Scientific Calculator that used to be available at drugstore chain outlets for five dollars:

2ndF >OCT 77341706 / 1000000000 2ndF >DEC = and see 0.123908564 in display

That uses 24 keystrokes. A solution on the TI-85 is

77341706 2nd BASE TYPE o / 1000000000 o ENTER and see .123980564448 in the display after 25 keystrokes.

I'll close with a little challenge for readers of this publication. Find an HP machine and a keystroke sequence which will return the answer in fractional form in less than 24 keystrokes. . .

Reference: Numeric Conversion Tables from Octal to Decimal and Decimal to Octal, by Jack Roy Morris, copyright 1959 by American Bosch Arma Co.

About the Author

Palmer Hanson worked on autopilot and bombing systems on F-100 and F-101 aircraft and on inertial navigation systems for aircraft such as the A-11 and YF-12 Blackbirds, the X-15, the B-52, and the AV-8. He authored six technical papers on inertial navigation. He met his first computer in the Navy in 1952 -- the electromechanical MK 1A which was part of the Mk 37 Gun Fire Control System. He met his first digital computer at the University of Minnesota in 1960 - the RemRand 1103. He met his first programmable calculator at Honeywell -- the TI-59. He authored five articles and numerous letters to the editor on the use of portable computers and programmable calculators in publications such as TRS-80 News, Byte, and PPX Exchange. From 1983 through 1991 he was the editor and publisher of TI PPC Notes, a newsletter for users of hand-held programmable calculators. That work was recognized by inclusion in the Who's Who in the South and Southwest. Since retirement he spends about half the year in Largo, Florida and half in Brevard, North Carolina.

From the Editor

HP Solve #26 page 42

← Previous **Article** – Next →

Return to Top

From The Editor – Issue 26

Winter has seriously started and its effects are even visible here in the Sonoran desert. We don't have snow, but as one having grown up in the mid-west I was reminded of this while going to my mail box and noticing my West side neighbor's tree(s). →

It can be cold in the winter, but the comfortable day time temperatures are the reason people live here. Temperatures below freezing are uncommon here at 1,241 feet.

Of course the best temperatures of any place on the planet are those of Southern California and I am reminded of this each year when I return to visit for – this year – the 2011 HHC, and the annual meeting of the CHHU/PPC calculator group in LA.

Let's hope for improved economic conditions so that ***HP Solve*** can be published more frequently. Of course that is also dependent on you, the reader. If we do not hear from you we will not really know that you are interested in receiving the newsletter.

See **HP News** below for some good news in this regard. Let me hear from you at:

hpsolve@hp.com

Photo by Richard J. Nelson

Fig. 1 – Even the Sonoran desert colors change.

HP News

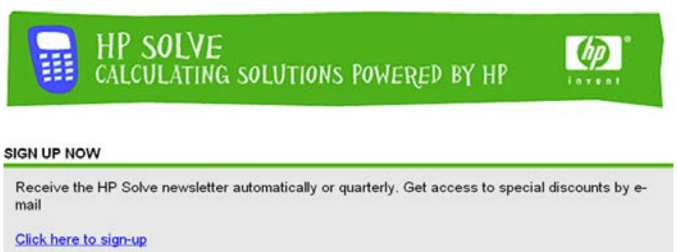


*Fig. 2 - **HP Solve** sign up is now especially easy.*

It seems that there is an issue with the part of the HP website that is called HP Passport. If you signed up for the newsletter in the past, and you were actually signed up for HP Passport, you probably didn't receive the newsletter. I know because this situation applies to me.

If you link to the ***HP Solve*** newsletter page as shown in Fig. 2 you will see a new, as of late September, choice to sign up for ***HP Solve***. Let me know if you have any problems. Of course you may download any issue at any time.

Reader Feedback

I used to have a column dedicated to an especially interesting and educational calculation problem. The Internet, however, has the answers to most problems so I gave this up after issue #19. The second and last problem (resistors on the edges of a cube) caught the attention of reader Francisco Chavez and his well-prepared solution follows.

A More Complete Solution to the Cube of Resistors Problem

Francisco Chavez

August 2011. The first time I ran into this problem was in an introductory physics course in college: twelve resistors are connected in a cubical arrangement, such that each resistor lies along each of the edges of a cube. The problem consists then in determining the equivalent resistance, R_{eq}, of the arrangement when a voltage differential is applied between opposite vertices of the cube.
When all 12 resistances are equal, say to R, an elegant solution is possible based on symmetry principles. The total equivalent resistance in this particular case can be shown to be $R_{eq}=(5/6)$ R, or about 86% of the resistance of the individual resistors.

In here, I present a more general solution, which provides R_{eq} in the case of non-equal individual resistances.

Let us identify each of the vertices of the cube by A, B, C,… H and let R_{ij} be the resistance of the resistor located between nodes i and j. Let us further assume that voltage is applied between opposite vertices A and H. Notice that this numbering is completely arbitrary. For the following discussion, l have followed the numbering shown in the following diagram, where the cube has been flattened, for simplicity of drawing.

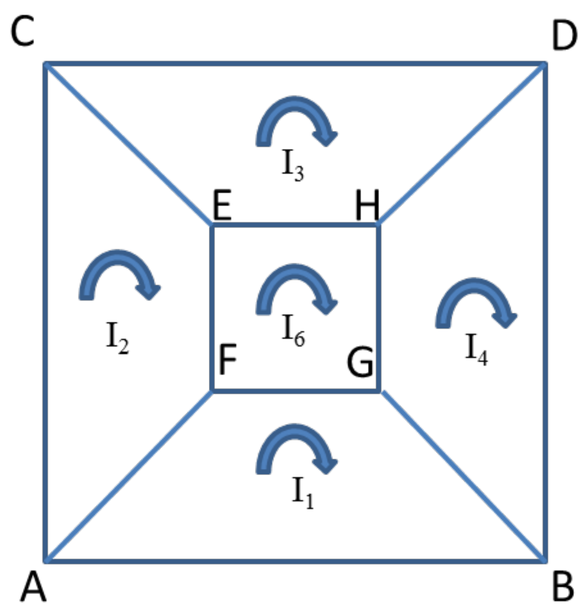


Fig. 1. A two dimensional representation of the cube of resistors. Voltage is applied between vertices A and H

We apply the well-known mesh analysis method with the loops defined as in Fig. 1. The basic idea of the method is that if we go around each loop and calculate the change of voltage through each resistor we find, when we return to the node we started from, the total change of voltage should be zero. Each time we pass through a resistor, Ohm's law is applied: $V = R_{ij}$ I. As an illustration here is the equation for loop 1:

$$R_{FG}(I_5+I_1-I_6) + R_{GB}(I_1-I_4) + R_{AB}I_1 + R_{AF}(I_5+I_1-I_2)=0$$

Notice that mesh currents are added or subtracted depending on their directions as defined by the arrows in Fig. 1, when they run through a particular resistor. Another part that may be tricky is the definition of Loop 5. This runs from nodes A-F-G-H-A (back to A), but notice that in this case the total differential of voltage is not zero, but precisely the applied voltage V.

We end up with a system of 6 linear equations that can be solved for the six unknown currents I_n. I will not write down all the equations but offer instead only the final linear system in matrix form:

[RES] [I] = [V]
Where,

[**RES**] =

$$\begin{pmatrix} R_{FG}+R_{GB}+R_{AB}+R_{AF} & -R_{AF} & 0 & -R_{GB} & R_{FG}+R_{AF} & -R_{FG} \\ -R_{AF} & R_{EF}+R_{AF}+R_{AC}+R_{CE} & -R_{CE} & 0 & -R_{AF} & -R_{EF} \\ 0 & -R_{CE} & R_{EH}+R_{HD}+R_{DC}+R_{CE} & -R_{HD} & 0 & -R_{EH} \\ -R_{GB} & 0 & -R_{HD} & R_{HD}+R_{DB}+R_{BG}+R_{GH} & R_{GH} & -R_{GH} \\ R_{AF}+R_{FG} & -R_{AF} & 0 & R_{GH} & R_{FG}+R_{GH}+R_{AF} & -R_{FG}-R_{GH} \\ -R_{FG} & -R_{EF} & -R_{EH} & -R_{GH} & -R_{FG}-R_{GH} & R_{FG}+R_{EF}+R_{EH}+R_{GH} \end{pmatrix}$$

$$[\,\mathbf{V}\,] = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ V \\ 0 \end{pmatrix}$$

The system can be solved to obtain the matrix of mess currents **[I].**

Please notice that these are mesh currents and not branch or actual currents. However, if needed, branch currents can be obtained in a straightforward manner. For instance, $I_{AF} = I_1 - I_2 + I_5$. Notice that I_2 and I_1 have different signs become they run in opposite directions through branch AF.

Finally, the equivalent resistance of the cube can be obtained again from Ohm's law: $V = R_{eq}\, I_5$

The following is an implementation of the solution of this problem on the hp-28s calculator. At around the time I solved this problem I had just acquired an hp-28s in the second hand market and thought that it would be a good project to get myself acquainted with the programming capabilities of this device.

For this implementation, the individual resistances R_{ij} are given in the vector **[R]**,

1	2	3	4	5	6	7	8	9	10	11	12
Rab	Rbd	Rcd	Rac	Rfg	Rgh	Reh	Ref	Raf	Rbg	Rce	Rdh

One needs to also define the one column matrix of voltages, which is

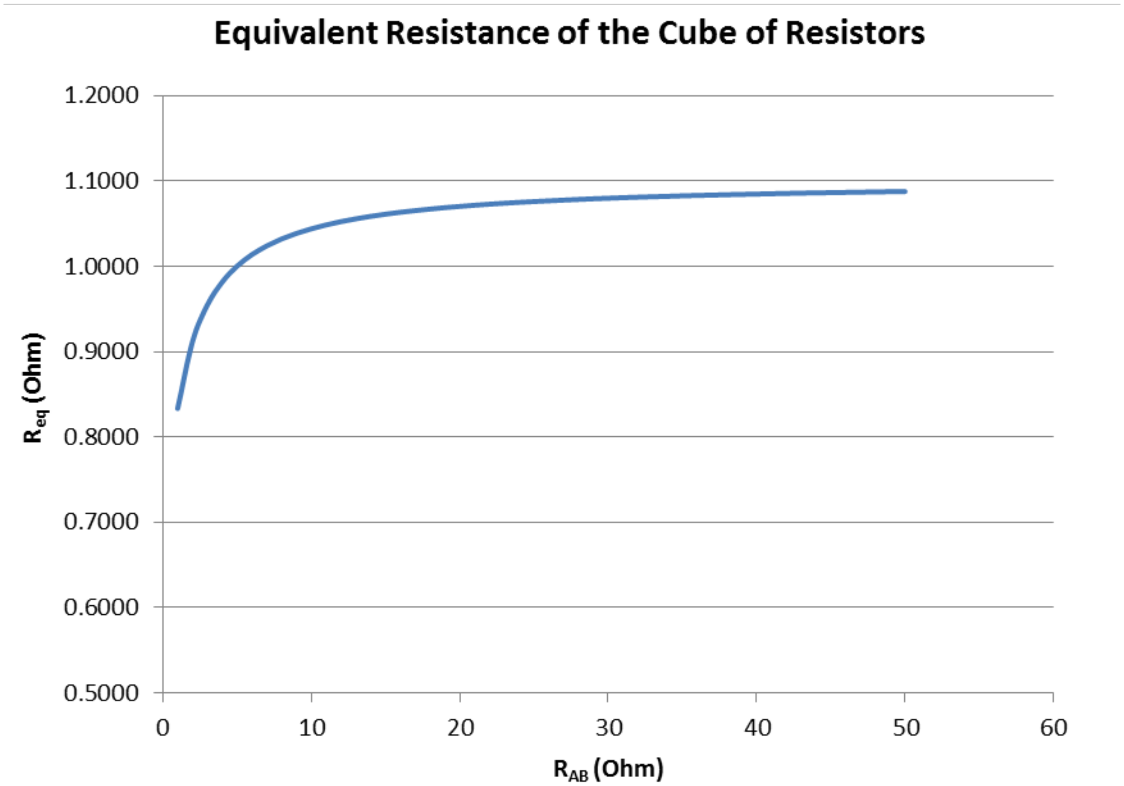
VOL= [[0],[0],[0],[0],[V],[0]]

In my tests, the program runs in about 12 seconds. The vast majority of the code consists in populating the matrix [RES] with elements from the individual resistances Rij, stored in the input vector **[R].** I decided to do it this way, because then it is easier to run the program for different values of R_{ij} just by changing the vector **R** and letting the program make the proper changes to matrix **[RES].**

After populating [**RES**] the program proceeds to solve the 6x6 system.

Two examples:
When all $R_{ij}=1$: $R_{eq}=5/6$.
When all $R_{ij}=1$ except $R_{AB}=R_1= 5$: $R_{eq}=1.000$

One can calculate now the equivalent resistance for any set of individual resistors. Just as an illustration, the following chart shows how the equivalent resistance grows with increasing values of R_{AB}, keeping all other R_{ij} = 1 Ohm.



PROGRAM FOR THE HP-28S TO FIND THE EQUIVALENT RESISTANCE OF A CUBE OF RESISTORS.
THE 12 INDIVIDUAL RESISTANCES ARE STORED IN A VECTOR **R**
THE VOLTAGES OF THE SIX LOOPS ARE STORED IN VECTOR **VOL.**
THE PROGRAMS THEN POPULATES THE MATRIX **CUB** WITH ELEMENTS OF **R** AND SOLVES FOR THE MATRIX OF CURRENTS **I**.

```
<<
VOL                                'Puts Voltage Matrix in Stack
CUB    {1,1}                       'Puts Matrix in the stack to be filled with the values of Rij
```

```
R  {5}   GET   R  {10}  GET    R {1} GET     R {9} GET   +  +  +  PUTI              'Fills Matrix Row 1
R {9}  GET  NEG  PUTI
0 PUTI
R  {10} GET NEG  PUTI
R  {5} GET    R{9} + PUTI
R  {5}  GET NEG PUTI

R {9} GET NEG PUTI
R {8} GET   R {9} GET   R {4}  GET   R {11}  GET  + + +   PUTI                    'Fills Matrix Row 2
R {11} GET NEG PUTI
0 PUTI
R {9} GET NEG PUTI
R {8} GET NEG PUTI

0 PUTI                                                              'Fills Matrix Row 3
R {11}  GET NEG PUTI
R {7}  GET  R{12} GET   R{3} GET   R{6} GET +++  PUTI
R {12} GET NEG PUTI
0 PUTI
R {7}  GET NEG PUTI

R {10} GET NEG PUTI                                                       'Fills Matrix Row 4
0 PUTI
R {12} GET NEG PUTI
R {12} GET   R{2} GET    R{10} GET   R{6} GET  + + +  PUTI
R {7}  GET PUTI
R {7}  GET NEG PUTI

R {9} GET R {5} GET + PUTI                                                'Fills Matrix Row 5
R {9} GET NEG PUTI
0 PUTI
R {6} GET PUTI
R {9} GET R {5} GET R{6} GET + + PUTI
R{5} GET NEG R{6} GET NEG  + PUTI

R{5} GET NEG PUTI                                                         'Fills Matrix Row 6
R{8} GET NEG PUTI
R{7} GET NEG PUTI
R{6} GET NEG PUTI
R{5} GET NEG R{6} GET NEG + PUTI
R{5} GET R{8} GET R{7} GET R{6} GET +++
PUT

VOL  /                                                      'Solves the 6x6 system
'IMes' STO                                                  'Store Matrix of Mesh Currents

IMes {5,1} GET INV                                          'Calculates Req
>>
```

Here is the content of this issue.

S01 – A New HP School Graphing Calculator is announced. Described by GT springer this new machine keeps things simple for students and general users alike.

S02 – The HP 50g as used in racing. The North American Eagle is challenging the land speed record.

S03 – HHC 2011 Report by Jake Schwartz, Gene Wright and me provides the details of an exceptional conference. The Conference Committee started setting up on Friday afternoon and it was a challenge getting all the tables and chairs we needed because of three other meetings (very rare) at the same facility. With the help of a lot of people, including our hotel, we were able to get set up for 74 registered attendees. It was a super great Conference that had the most dense presentations of any Conference (hhuc.us) so far. See how we solved the problem of speakers keeping to their allocated time.

S04 – The Four Meanings of "Accurate to 3 Places" by Joseph K. Horn is an exceptional article. It is so clearly written that you are sad to see it come to a conclusion – always a test of a well written article. It discusses a very important topic of interest to every calculator user – decimal to fraction conversion accuracy. Joseph was voted the Best Speaker of HHC 2011. You may watch his HHC presentation via the video link in the article. Thanks to Eric Rechlin for making the video and posting it.

S05 – What is Double Injection Molding of HP Calculator Keys? Old time HP calculator users will often mention this unique HP process. Read this short article with photos to understand what this means.

S06 – Timing for HP 35s Calculator Instructions by Richard Schwartz. Every calculator is unique in its method of providing its user interface. The execution timing of the various functions and commands depends on a very complex series of situations that often surprise the users who eventually notice that similar operations to not have similar execution times. Richard provides a discussion on the methods of making the Instruction timing measurements. Not all instructions have been timed so there is a need for others to fill in the gaps.

S07 – Octal Fraction Conversions by Palmer O. Hanson. This exotic problem may not interest you, until you actually have to make octal fraction conversions. You will get an idea of how this problem was solved historically. Program examples are provided. If, however, you really need to make octal fraction conversions you may need an HP 50g and a suitable (available) library.

S08 – Regular Columns This is a collection of news items and repeating/regular columns. A new column, Calculator Accuracy, continues with this issue.

- **From the editor.** This column provides feedback and commentary from the editor.
- **One Minute Marvels.**
- **Calculator Accuracy.** This is an important topic for HP calculator users to explore and use to be better calculator users. No calculator is 100% accurate – yet.

What do you think about a series of articles on the lore (lure?) of HP calculators? Let me hear from you.

S09 – #9 in the Math Review Series: Mathematical Constants – The Golden Ratio This constant is probably one of the most widely intriguing of all mathematical constants. Here is an overview of this artistic number with lots of interesting links for further exploration and study.

That is it for this issue. I hope you enjoy it. If not, tell me!

Also tell me what you liked, and what you would like to read about.

X < > Y,

Richard

Email me at: hpsolve@hp.com

HP 48 One Minute Marvel – No. 13 – Day of Week

One Minute Marvels, OMMs, are short, efficient, unusual, and fun HP 48 programs that may be entered into your machine in a minute or less. These programs were developed on the HP 48, but they will usually run on the HP 49 and HP 50 as well. Note the HP48 byte count is for the program only.

One of the really powerful functions of the HP-48/49/50 series of RPL machine is the built in 8,419 year (October 15, 1582 to December 31, 9999) calendar. Using the powerful date commands all kinds of powerful and useful date routines may be written.

Knowing the day of the week is useful for relating it to other days. A weekend day, Sunday or Saturday, for example, might mean that you were not working. A particular day might be meaningful, Friday, for example, if it occurs on the 13th of the month, might be considered unlucky. For details see Issue 25.

The HP 48/49/50 series machines have a function that accepts the date and returns information related to that date. Here is a ten second Marvel that uses TSTR to return a three character string for the day rather than a conventional number for the day. See the second routine that returns a number for the day.

Of course both of these routines assume that the user has properly set the current date.

Input a standard date in mm.ddyyyy format and **'dow1'** (system flag –42 clear) returns a three-letter day. the zero is required (in addition to the date) as an input for TSTR. Check your AUR.

'dow1' << 0 TSTR 1 3 SUB >>

5 commands, 22.5 Bytes, # A8D0h. Timing: 8.211999 ⇒ "SAT" in 26.2_ms.

Joseph K. Horn suggests using DDAYS and a known date to calculate the day of week. The known date is a Sunday (year 3,000) and is selected to have the day and month the same so the system flag –42 setting doesn't matter. He had to "hunt" for a date that met these requirements. Given a date in mm.ddyyyy format, **'dow2'** returns a number between 0 (Sunday) and 6 (Saturday). Example: HHC 2011date 9.242011, **'dow2'** returns 6 (Saturday).

'dow2' << 2.023 SWAP DDAYS 7 MOD >>

5 commands, 30.5 bytes, #B181h. Timing: 8.211999 ⇒ 0 in 7.17_ms.

Two different techniques are used to return the day of week given a date. Both use five commands, but one (**'dow2'**) is 3.7 times faster.

Reader challenge. Use a different technique to return the DOW. Suppose you have a programmable calculator that does not have a built in calendar. How would you write a program to return DOW? The simplest approach might be to input the current date and DOW. This is not a trivial challenge because the months are not the same number of days and you have to account for leap years. Hint: Convert to/from Julian Day.

Calculator Accuracy – Part 2 – Guard Digits

Introduction

In Part 1 the suggestion that the reputation of the manufacturer is what the average calculator user thinks about when thinking about calculator accuracy. It was also suggested that some functions of a 10 digit machine may only have an accurate 7 digits (9 digits on a 12 digit machine) for more complex functions.

For most calculations the accuracy is of little concern and perhaps the best example of where accuracy is more important is calculations made on a financial calculator. You want and require the answer to be correct to the penny. A very good example of the importance of accuracy was provided in the last issue of ***HP Solve***, #25, in the article titled The HP-12C, 30 Years and Counting. See the topic on page 13 "A penny for your thoughts." The correct answer is $ 331,667.01[1] for HP calculators. For the most common non-HP financial calculators the results were:

#	Result	Error	% Error
1.	$ 293,539.16	Short by $ 38,127.85	-11.5
2.	$ 334,858.18	Over by $ 3,191.17	+0.96
3.	$ 331,559.38	Short by $ 107.63	-3.25

If the bank is going to pay you based on the calculation you want them to use the calculator in example #2 and certainly not the result of example #1. The bottom line: You want a correct/accurate answer.

The definition of accuracy is not as simple as it may seem. Joseph K. Horn provides a very clear example of how difficult the accuracy issue is in his article in this issue The Four Meanings of "Accurate to 3 Places" . Be sure to watch the HHC video in the link in the article. When you realize this you can understand why manufacturers are reluctant to provide a precise accuracy statement for their machines.

Another point made in part 1 with the Calculator Forensics Results Tables was the difference between BCD calculators and binary calculators. The HP calculators made during the first three decades of HP calculators were all BCD. Integrated circuit technology slowly started to impact HP calculators and machines like the HP 9g, 30s, 10s and WP 34S are binary machines. As long as the number of bits is high enough a binary machine is certainly as accurate – as Tables 1 & 2 in part 1 illustrate – as required.

To determine if your calculator is BCD or binary See ***HP Solve***, #20 page 37.

Calculator Arithmetic

Calculator arithmetic is done using registers of a finite length. In terms of the display[2], usually10 digits or 12 digits, the number of digits that are calculated is greater to insure accuracy. These not-displayed digits are, in general terms, called guard digits, and for HP calculators a ten digit calculator will usually calculate with 13 digits internally, and a 12 digit calculator will calculate with 15 digits internally to provide three guard digits..

The (philosophical) question to ask is what do you do with the guard digits after a calculation? Joseph addresses the extra accuracy digits in his article on this issue. Basically there are two possibilities. 1) simply display the first 10 or 12 digits, or 2) display the first 10 or 12 digits, but showing the last displayed digit rounded. HP follows 2).

The next question to ask is, "Do you keep the guard digits for the next calculation?" Again there are two possibilities, 1) keep them for use in chained calculations, or 2) discard them. HP follows 2) and this policy is different from nearly every other calculator. The HP result is "what you see is what you get," WYSIWYG. The whole point is to make your calculations based on what you see or put into the display and not using numbers you don't even know are there. Inexperienced calculator users often think that keeping the guard digits results in greater accuracy. If the accuracy were improved (and it was important) there would be a key similar to the "SHOW" key on some HP machines to display the additional digits.

Do not confuse guard digits with the digits you see based on the display mode.

In the vast majority of HP's calculators (except the few binary machines; ≈ 2 % of the total machines made in 41 years) the user cannot access the guard digits. This is just one accuracy feature that makes HP calculators unlike all the others.

Displayed digits, rounding, truncating, and the internal algorithms all play a role in determining how accurate or meaningful the number is in your HP calculator display. We will explore specific examples in part 3 of this series.

Observations and conclusions

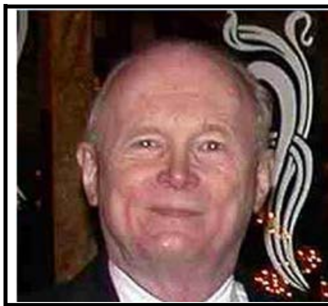
Calculator accuracy is a very complex topic and not as obvious as you might expect. Understanding the ideas involved with making the calculations and how the answer is displayed is important to HP calculator users because HP's machines are, over all, considered the most accurate. Knowing and specifying this, however, is a challenge because an experienced bug hunter using a computer can search for, and find, answers that are "not correct." Defining accuracy and correctness however, is nearly impossible as Joseph K. Horn clearly explains in his HHC 2011 paper also found in this issue.

Notes: Calculator Accuracy – Part 2 – Guard Digits

(1) William Kahan of the University of California Berkeley consulted with HP on the HP-12C and other calculators (HP-15C & HP-34C) to insure algorithmic accuracy. He has a long and productive career in promoting computational accuracy in computers and calculators. See page 15 of Mathematics Written in Sand, Version of 22 November, 1983 at: http://www.cs.berkeley.edu/~wkahan/MathSand.pdf.

(2) An exception is the HP OfficeCalc series of machines which display 14 digits.

About the Editor



Richard J. Nelson is a long time HP Calculator enthusiast. He was editor and publisher of *HP-65 Notes*, *The PPC Journal*, *The PPC Calculator Journal*, and the *CHHU Chronicle*. He has also had articles published in *HP65 Key Note* and *HP Key Notes*. As an Electronics Engineer turned technical writer Richard has published hundreds of articles discussing all aspects of HP Calculators. His work may be found on the Internet and the HCC websites at: hhuc.us He proposed and published the PPC ROM and actively contributed to the UK HPCC book, *RCL 20*. You may also reach Richard at: rjnelsoncr@cox.net

Fundamentals of Applied Math Series #9

HP Solve #26 page 52

← Previous Article

Return to Top

Mathematical Constants – The Golden Ratio - φ, and Φ

Richard J. Nelson

Introduction – What is a mathematical constant?

In past issues of ***HP Solve*** a mathematical constant was described as a special number, usually a real number, that is especially interesting to mathematicians. Constants arise in many different areas of mathematics and two especially well known constants are Euler's number *e*, and Pi, *π*. *e* was discussed in Math Review #6, ***HP Solve*** issue 23, and *π* was discussed in Math Review #8, ***HP Solve*** issue 25. The Math Review series started in ***HP Solve*** issue 18 with a review column in every issue. The constant φ and its reciprocal Φ is discussed in this colum.

φ is also known as the golden ratio, the golden section, and the divine proportion. The exact value of Φ is $\frac{1+\sqrt{5}}{2}$. Expressed for calculator solution to save an RPN keystroke $\varphi = \frac{\sqrt{5}+1}{2}$. The golden ratio to 45 decimal places is:

φ = 1.61803 39887 49894 84820 45868 34365 63811 77203 09179 . . .

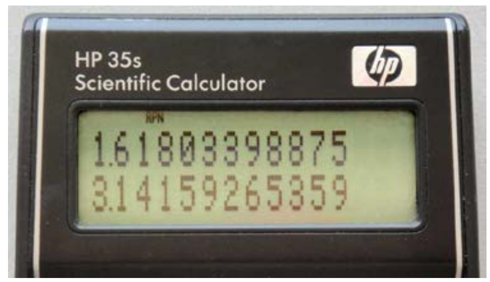


Fig. 1 – HP-35s Display showing π & Φ.

If you are photographing a calculator φ would be an interesting number to put into the display. Depending on the number of digits the display uses each digit, 0 - 9, would be represented as shown below.

8 digits: 1.61803 40 are missing 2, 5, 7, and 9. **[4]**
10 digits: 1.61803 3989 are missing 2, 4, 5, and 7. **[4]**
12 digits: 1.61803 39887 5 are missing 2, 4, and 9. **[3]**
14 digits: 1.61803 39887 499 are missing 2. **[1]**

What makes φ interesting and perhaps the-most-interesting-display number?

1. Its reciprocal, Φ**,** has the same decimal digits. 1/φ = 0.61803 39887 49894 . . .
2. Its square is the same as adding 1. φ^2 = 2.61803 39887 49894 . . .

Φ has a rich history

φ has been known since at least 300 BC when the Greek mathematician Euclid described it (its construction) in *Elements* and φ may have been a factor in the design of the Great Pyramid in circa 2540 BC. The golden ratio is especially appealing to the human eye in terms of buildings and the human body.

φ has artistic value in that it is used to hang paintings and size rectangles because these proportions are aesthetically pleasing and have been used since the renaissance period.

φ and its reciprocal Φ are irrational numbers. Its value has been calculated: to 10 million digits in December 1996, and to 1.5 Billion digits in May 2000. φ expressed in any base does not have any ultimate repeating pattern in their digits.

Da Vinci, during the renaissance, claimed that there were a number of applications of the golden ratio in the human body. He found that a perfectly structured human body would have the golden ratio between:

- first finger joint and second - second joint to both.
- hand to lower arm - both hand and lower arm.
- many proportions creating the perfect face[(1)].
- so on, all over the body.

Defining φ

The greek letter phi, φ, was suggested to represent the golden Ratio by Mark Barr (20th century). This was inspired because the Greek letter phi (**φ**) is the initial letter of the Greek sculptor Phidias's name.

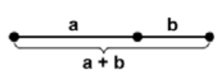
φ and the Fibonacci numbers[(2)] are related and it may be shown how the Fibonacci number (ratio of successive Fibonacci numbers) arise from φ.

Let's start with the first two decimal numbers 0, 1. If we make a series by having the next term being the sum of the two previous terms the next term is 1 ⇨ 0, 1, 1. The next term is 1 + 1 or 2 ⇨ 0, 1, 1, 2. The next term is 3 and the Fibonacci series is: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, …

If we divide each term by the previous term we will have the results as shown below.

1/1 = 1
2/1 = 2
3/2 = 1.5
5/3 = 1.6666...
8/5 = 1.6
13/8 = 1.625
21/13 = 1.61538...
34/21 = 1.61904...
55/34 = 1.61764…
89/55 = 1.61818…
144/89 = 1.61797…
233/ 144 = 1.61805…
377/233 = 1.61802…
610/377 = 1.61803…

The ratio seems to be settling down to a particular value which in fact is the Golden Ratio, (φ=1.61803...).

Geometrically the golden ratio may be expressed as: 

, if $\frac{a+b}{a}=\frac{a}{b}=\varphi$.

If length ab is unity, a= 61.8%, b =38.2%. Hanging a painting 61.8% of the ceiling height from the floor is considered an appealing location.

The golden Ratio is everywhere

Why do shapes that exhibit the Golden Ratio seem more appealing to the human eye? No one really knows for sure. But we do have evidence that the Golden Ratio seems to be Nature's perfect number**.** Take, for example, the head of a daisy. Someone discovered that the individual florets of the daisy (and of a sunflower as well) grow in two spirals extending out from the center. See fig. 2. The first spiral has

21 arms, while the other has 34. Do these numbers sound familiar? They should - they are Fibonacci numbers! Their ratio is the Golden Ratio. The spirals of a pinecone are similar where spirals from the center have 5 and 8 arms, respectively (or 8 and 13, depending on the size). These are also two Fibonacci numbers:

A pineapple has three arms of 5, 8, and 13 which additional evidence that this is not a coincidence.

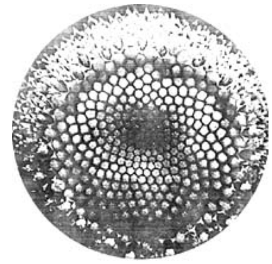


Fig. 2 – Head of a daisy.

Fig. 3 – Pine cone and pineapple examples of the Golden ratio in nature.

Nature is obviously efficient. Why do plants grow in this way? Some scientists speculate that plants that grow in a spiral formation - in Fibonacci number formation - because this arrangement makes for the perfect spacing for growth. Fibonacci numbers provide the perfect arrangement for optimum growth and survival of the plant.

The Golden Ratio is used in such diverse applications[3] as Architecture, Book Design, Finance, Industrial Design, Music, Nature, Optimization, Painting, Perceptual Studies, Web Design[4], and of course many branches of Mathematics.

The Pearl Musical company of Japan positions the air vents on its four Masters Premium drum models based on the golden ratio. The company claims that this arrangement improves bass response and has applied for a patent on this innovation.

Rectangles that are Golden Ratio proportioned[6] are supposed to be more astatically appealing. If two sides of a rectangle have the Golden Ratio, then cutting a square off the rectangle leaves a smaller rectangle having the same proportions.

Observations and Conclusions

The golden Ratio is one of the most interesting mathematical constants[5] from an aesthetics perspective. I personally "discovered" the Golden Ratio many years ago (before the Internet) while playing with my calculator trying to find a number that the digits wouldn't change when I took its reciprocal.

The Golden Ratio is used in such diverse applications[(3)] as Architecture, Book Design, Finance, Industrial Design, Music, Nature, Optimization, Painting, Perceptual Studies, Web Design[(4)], and of course many branches of Mathematics.

Notes for Mathematical Constants – The Golden Ratio - φ, and Φ

(1). The human face Golden Ratio is nicely described at: http://www.goldennumber.net/face.htm

(2). For additional calculator related information (HP 39gs) on Fibonacci Numbers see Tutorial: One Stubborn Ratio: Using the HP 39gs by GT Springer.
http://h20331.www2.hp.com/Hpsub/cache/429025-0-0-225-121.html

(3). Additional examples of the aesthetics of the Golden Ratio may be found at:

http://en.wikipedia.org/wiki/Golden_ratio

(4) Web Page layout: ***http://webdesign.about.com/od/webdesignbasics/a/aa071607.htm***

(5) Important constants to 100D: ***http://home.adelphi.edu/~stemkoski/mathematrix/constant.html***

(6) Golden Ratio rectangle calculator: ***http://www.blocklayer.com/goldenratio.aspx***

(7) Additional useful Golden Ratio links.

a. **http://mathworld.wolfram.com/GoldenRatio.html**

b. I*mages*: (one long link)
http://www.google.com/search?q=golden+ratio&hl=en&rlz=1C2SKPM_enUS412&prmd=imvns&tbm=isch&tbo=u&source=univ&sa=X&ei=LsLoTtneGumKsgLr_7XyCw&sqi=2&ved=0CFIQsAQ&biw=740&bih=534

c. The math of beauty: http://www.intmath.com/numbers/math-of-beauty.php