♦MoHPC♠ *The Museum of HP Calculators*

HP Forum Archive 20

[Return to Index | Top of Index]

Determinants of the AM1 matrix

Message #1 Posted by ***gene wright*** *on 30 Sept 2011, 3:29 p.m.*

AM1 Matrix

This is one of Valentin's matrices to test accuracy in computing a determinant. We have talked about it before. The exact determinant should be 1. Let's gather a list of results for various NON-exact mode machines, non-TINY ELEMENT flag machines as an article for accuracy.

WP 34S	0.999997119354
HP-28S	0.970960198039
HP-15C	1.080204421
HP-35s	1.00282960115
HP-71B	0.97095056196
HP-48G	0.999945522778
Casio FX-9860G Slim	1.0000000369657

Here are some starting values.

Re: Determinants of the AM1 matrix

Message #2 Posted by ***Bunuel66*** *on 30 Sept 2011, 3:50 p.m.,*
in response to message #1 by gene wright

TI82stats: 0.9996468043

Re: Determinants of the AM1 matrix

Message #3 Posted by ***Crawl*** *on 30 Sept 2011, 5:02 p.m.,*
in response to message #1 by gene wright

HP50g (with flags set as stated) gives the same answer as the HP48g.

The TI89 in approx mode gives 0.999718361859.

Re: Determinants of the AM1 matrix

Message #4 Posted by ***Jesse Dodd*** *on 30 Sept 2011, 5:40 p.m.,*
in response to message #1 by gene wright

TI nSpire CX CAS:

0.9997183618585

Re: Determinants of the AM1 matrix

Message #5 Posted by ***Peter Murphy (Livermore)*** *on 30 Sept 2011, 6:53 p.m.,*
in response to message #1 by gene wright

My shiny 39g+, acquired at HHC 2011, yields 1 exactly for AM1, suggesting that it's running in exact mode.

Does the 39g+ have a non-exact option, or am I stuck with exactness even if I don't want it? The user's guide appears not to discuss the subject.

Re: Determinants of the AM1 matrix

Message #6 Posted by ***Crawl*** *on 30 Sept 2011, 8:13 p.m.,*
in response to message #5 by Peter Murphy (Livermore)

For the 50g, to not get 1 exactly, not only do you need to turn off exact mode but you also have to set the "retain small element" flag.

Re: Determinants of the AM1 matrix

Message #7 Posted by ***Peter Murphy (Livermore)*** *on 30 Sept 2011, 9:17 p.m.,*
in response to message #6 by Crawl

The 39g+ appears to be an unflagged calculator.

Changing the (7,7) element of AM1 to 69.0 didn't trick the calculator: the determinant is still 1.

Re: Determinants of the AM1 matrix

Message #8 Posted by ***Crawl*** *on 30 Sept 2011, 10:14 p.m.,* *in response to message #7 by Peter Murphy (Livermore)*

I skimmed through the 39g+ manual; it doesn't seem to have an exact mode.

My best guess is that it may be in fraction mode, so it's rounding 0.999.. to the fraction 1. You should be able to check in the Mode menu, under number format.

Re: Determinants of the AM1 matrix

Message #9 Posted by ***Bunuel66*** *on 1 Oct 2011, 6:04 a.m.,*
in response to message #8 by Crawl

HP39GS: 1.0000000000E0 quite instantaneously.

If I get some time, I'll experiment a bit with the sensitivity of the determinant computation to coefficients of the matrix.

Regards

Re: Determinants of the AM1 matrix

Message #10 Posted by ***Les Bell*** *on 30 Sept 2011, 8:09 p.m.,*
in response to message #1 by gene wright

HP-41CX with Advantage ROM: 1.082014365, in just over 14 seconds.

Best,

--- Les
[http://www.lesbell.com.au]

Re: Determinants of the AM1 matrix

Message #11 Posted by ***Ángel Martin*** *on 1 Oct 2011, 4:32 a.m.,*
in response to message #10 by Les Bell

Same result (but of course) but instantaneous on the CL.

Edit: it gave me a good opportunity to put my PMTM method to test, a real breeze inputing all those matrix elements "in a row"

Edited: 1 Oct 2011, 5:15 a.m.

Re: Determinants of the AM1 matrix

Message #12 Posted by ***Palmer O. Hanson, Jr.*** *on 30 Sept 2011, 10:46 p.m.,*
in response to message #1 by gene wright

Gene:

I suggest that you should go back and look at all the work that we did with this matrix back in 2005. A good place to start would be with the thread "More Matrix Results on HP and TI Machines" at http://www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/archv015.cgi?read=72366 . One of the inputs that you will find there is your submission "TI CC-40 beats the HP-71B" which gives the CC-40 result as 1.00282671. The thread also includes the result for the TI-95 and other machines.

Palmer

Re: Determinants of the AM1 matrix

Message #13 Posted by ***gene wright*** *on 30 Sept 2011, 11:19 p.m.,*
in response to message #12 by Palmer O. Hanson, Jr.

Thanks, Palmer!

I had already grabbed those results and put them into a spreadsheet to keep up with all of this. The best results on this matrix so far are:

```
Casio FX-9860G Slim            1.000000036966
WP 34S                         0.999997119354
```

Quite amazed at the FX-9860G.

Re: Determinants of the AM1 matrix

Message #14 Posted by ***Thomas Chrapkiewicz*** *on 1 Oct 2011, 12:25 a.m.,*
in response to message #13 by gene wright

Hmmm....I'm thinking that these results along with the LBL A + loop, Savage benchmark, 8 queens results, etc could be added to this spreadsheet...

Perhaps you're doing this already?

TomC

Re: Determinants of the AM1 matrix

Message #15 Posted by ***Palmer O. Hanson, Jr.*** *on 2 Oct 2011, 10:56 p.m.,*
in response to message #13 by gene wright

> Quote:
>
> I had already grabbed those results and put them into a spreadsheet to keep up with all of this.

Why don't you dump those results as an Article in this site?

Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #16 Posted by ***gene wright*** *on 4 Oct 2011, 11:43 a.m.,*
in response to message #15 by Palmer O. Hanson, Jr.

I will soon! Results so far! The ratio column is the error of the model listed divided by the error of the best model. The 39g+ is not listed because I do not believe it can possibly really return a 1 for a result. Sorry...

Calc	Value	Error	Ratio of error to best
Casio FX-9860G Slim	1.00000003696570	0.00000003696570	
WP 34S New Pauli firm	1.00000008439107	0.00000008439107	2.28
WP 34S RRM	0.99999711935400	0.00000288064600	77.93
HP-48G	0.99994552277800	0.00005447722200	1,473.72
TI-89	0.99971836185900	0.00028163814100	7,618.90
TI nSpire CX CAS	0.99971836185850	0.00028163814150	7,618.90
TI-86	0.99964680433800	0.00035319566200	9,554.69
TI-82	0.99964680430000	0.00035319570000	9,554.69
TI-83+	0.99964680400000	0.00035319600000	9,554.70
TI-95	1.00067670842400	0.00067670842400	18,306.39
TI-59	0.99836009871850	0.00163990128150	44,362.78
TI CC-40 Computer	1.00282671000000	0.00282671000000	76,468.46
HP-35s	1.00282960115000	0.00282960115000	76,546.67
HP-28S	0.97096019803900	0.02903980196100	785,587.77
HP-48S	0.97096019803900	0.02903980196100	785,587.77
HP-71B	0.97095056196000	0.02904943804000	785,848.45
HP-15C	1.08020442100000	0.08020442100000	2,169,698.42
HP 41 Advantage ROM	1.08201436500000	0.08201436500000	2,218,661.21
HP-41 Math ROM	(1.81476270400000)	2.81476270400000	76,145,256.27

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #17 Posted by ***Marcus von Cube, Germany*** *on 4 Oct 2011, 12:25 p.m.,*
in response to message #16 by gene wright

```
WP 34S New Pauli firm    1.00000008439107        0.00000008439107        2.28
```

The latest firmware should do better. :-)

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #18 Posted by ***Paul Gaster*** *on 4 Oct 2011, 1:23 p.m.,*
in response to message #16 by gene wright

Maybe I am missing something but how did you get an answer with a 35S? Are you using Stefan's matrix program?

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #19 Posted by ***gene wright*** *on 4 Oct 2011, 2:59 p.m.,*
in response to message #18 by Paul Gaster

Actually, I'm not sure. I took that result from the earlier post about this matrix from the forum.

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #20 Posted by ***Paul Gaster*** *on 4 Oct 2011, 8:32 p.m.,*
in response to message #19 by gene wright

I found it now. Yes, the 35S result is from Stefan's Matrix Multi-Tool Program for the HP-35s.

hope these links work:

link

link

link

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in

now!

Message #21 Posted by ***Walter B*** *on 5 Oct 2011, 1:37 a.m.,*
in response to message #20 by Paul Gaster

Thanks for excavating! I found the oldest thread the most interesting - there are two more matrices to investigate :-)

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #22 Posted by ***Paul Dale*** *on 5 Oct 2011, 1:38 a.m.,*
in response to message #21 by Walter B

The 34S gets 1 and 1 for their determinants :-)

- Pauli

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #23 Posted by ***Walter B*** *on 5 Oct 2011, 1:45 a.m.,*
in response to message #22 by Paul Dale

Amazing, our baby, isn't it? Thanks for coding :-)

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #24 Posted by ***Paul Dale*** *on 4 Oct 2011, 5:14 p.m.,*
in response to message #16 by gene wright

The current 34S result is 1 exactly. I very much doubt we'll go back to the less accurate version.

This will make all the other errors infinite :-)

- Pauli

Edited: 4 Oct 2011, 5:15 p.m.

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #25 Posted by ***Palmer O. Hanson, Jr.*** *on 4 Oct 2011, 10:14 p.m.,*
in response to message #16 by gene wright

Gene:

The next thing that you might want to explore could be the results with AM2 and AM3 which are even more difficult.

Palmer

AM3 determinant on the new 34s firmware

Message #26 Posted by ***gene wright*** *on 4 Oct 2011, 10:25 p.m.,*
in response to message #25 by Palmer O. Hanson, Jr.

works out to 1.000 ... to the limits of returned precision.

The internal precision was (according to Pauli):

1.000000000000000000000000004678227405010

Amazing.

However, it did turn out a lock-up error trying to take the inverse of the AM3 matrix, which is being worked on.

Palmer, we need to get you a 34s. Email me!

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #27 Posted by ***Paul Dale*** *on 5 Oct 2011, 4:13 a.m.,*
in response to message #16 by gene wright

In case anyone is interested, I've added a matrixtest.wp34s file to the library that contains the AM1, AM2 and AM3 matrices. In each case the 49 elements are stored from register 20 and the determinant is taken. The matrix descriptor for all three is 20.07.

Just assemble and install in the emulator or on the real hardware.

- Pauli

We need a new calc.bin file first ;-)

Message #28 Posted by ***gene wright*** *on 5 Oct 2011, 10:11 a.m.,*
in response to message #27 by Paul Dale

to fix the memory issues with the inverse function?

Re: We need a new calc.bin file first ;-)

Message #29 Posted by ***fhub*** *on 5 Oct 2011, 11:23 a.m.,*
in response to message #28 by gene wright

> Quote:
>
> to fix the memory issues with the inverse function?

Not only this!

Now there's a little confusion about the SVN updates:
4 hours ago Pauli made changes to SVN 1683 removing M.LU and re-introducing the matrix-row-operations (there was a new wp34s.op).
Now Marcus has made a new build SVN 1685 where these last changes are NOT valid (i.e. M.LU is back and row-ops not included).
And there should be also a M.DIAG (which is used in the new matrix.wp34s), but it doesn't exist in the current build.

Slowly it's going to be a mess ...

Franz

Edited: 5 Oct 2011, 11:49 a.m.

Re: We need a new calc.bin file first ;-)

Message #30 Posted by ***Marcus von Cube, Germany*** *on 5 Oct 2011, 1:52 p.m.,*
in response to message #29 by fhub

It was just a missing update (or commit). I'll fix it soon.

SVN had not warned me about the fact that my local copy wasn't up to date. It should be fixed now.

BTW, please never *complain* about inconsistencies in the SVN builds, just tell us, preferably with an email! SVN is not a release mechanism and you can expect that there are non working build checked in just to keep the team members in sync.

Edited: 5 Oct 2011, 2:03 p.m.

Re: We need a new calc.bin file first ;-)

*Message #31 Posted by **Neil Hamilton (Ottawa)** on 5 Oct 2011, 2:10 p.m.,*
in response to message #30 by Marcus von Cube, Germany

> Quote:
>
> BTW, please never *complain* about inconsistencies in the SVN builds, just tell us, preferably with an email! SVN is not a release mechanism and you can expect that there are non working build checked in just to keep the team members in sync.

Hear, hear! Very well said!

People have to realize that you and Pauli are ~9 timezones apart and SVN is used as the communication tool to pass files around -- as in any distribute development. You are *kind* enough to let us in on the bleeding edge of the development so the belly-aching and complaining is just not appropriate!

Re: We need a new calc.bin file first ;-)

*Message #32 Posted by **fhub** on 5 Oct 2011, 2:19 p.m.,*
in response to message #30 by Marcus von Cube, Germany

> Quote:
>
> BTW, please never *complain* about inconsistencies in the SVN builds, just tell us, preferably with an email!

Where did I complain? I was just informing!

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

*Message #33 Posted by **Neil Hamilton (Ottawa)** on 5 Oct 2011, 2:01 p.m.,*
in response to message #27 by Paul Dale

Wow! Pauli I just got back from my adventure to San Diego and find the matrix stuff in place!! The editor is **very** cool.

One question about power consumption... I noticed that the RPN indicator was flashing and had to remember that this meant a program was still executing(?). Does this mean that there is a larger power drain while MED is running?

Cheers...

(BTW: Thanks to all who made me feel so welcome in the HHC group. You are a genuinely nice bunch of people!)

Edit: It looks like I may have mis-attributed MED to Pauli when it looks like Marcus created it. Sorry Marcus!! Very cool tool...

Edited: 5 Oct 2011, 2:04 p.m.

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

*Message #34 Posted by **Marcus von Cube, Germany** on 5 Oct 2011, 2:05 p.m.,*
in response to message #33 by Neil Hamilton (Ottawa)

MED is mine. :-)

It's RCL that is flashing while a PSE is in progress or while a program is running, not RPN. The power drain is only slightly higher than in deep sleep, it should still be well under 1mA.

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

*Message #35 Posted by **Neil Hamilton (Ottawa)** on 5 Oct 2011, 2:14 p.m.,*
in response to message #34 by Marcus von Cube, Germany

Hi Marcus,

Correct again! It *is* RCL that is flashing.

(Looks like I got an edit in just a minute before you corrected me on the author :-)

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

*Message #36 Posted by **Bunuel66** on 6 Oct 2011, 4:36 p.m.,*
in response to message #16 by gene wright

It is interesting not to believe the result provided by HP39g or gs. Just do the experiment. On my 39gs I have checked the display: scientific with 11 figures and even tested det(AM1)-1 to see possible hidden value. In any case, The result is exactly 1. I'm probably missing something. Please give me your light ;-)

Regards

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #37 Posted by ***Gerson W. Barbosa*** *on 6 Oct 2011, 4:41 p.m.,*
in response to message #36 by Bunuel66

Something like this, perhaps?

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #38 Posted by ***gene wright*** *on 6 Oct 2011, 5:26 p.m.,*
in response to message #37 by Gerson W. Barbosa

Yes, perhaps. I have worked with the 39g+ and 39gs and simply have no reason to think they have a better way to handle numbers like this than the 49g+ or 50g have / had.

I'm not disputing it returns a 1, but perhaps it always does things that are flag selectable on the 50g - such as the tiny element flag and noting that the matrix had integers so the determinant must be an integer.

That's cheating. :-) At least from an accuracy perspective. After all, the actual result might work out to be 48.4802 and it rounds it to 48 because of such things. An integer, yes, but still wrong because the math inside was so bad.

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #39 Posted by ***Paul Dale*** *on 6 Oct 2011, 5:58 p.m.,*
in response to message #38 by gene wright

As Crawl wrote on the other thread, change the first element of AM1 to 58.1 and see what happens. The correct answer is 9636025.5. This should bypass the everything is an integer check.

- Pauli

Re: Interim table of results for determinant of the AM1 matrix -- should = 1 ... get your models in now!

Message #40 Posted by ***Bunuel66*** *on 7 Oct 2011, 2:16 p.m.,*
in response to message #39 by Paul Dale

Checking HP39gs with 58.1 gives det=9636025.49994, not that bad ;-)

Regards

Re: Determinants of the AM1 matrix

Message #41 Posted by ***Paul Dale*** *on 30 Sept 2011, 11:14 p.m.,*
in response to message #1 by gene wright

The 34S might get an accuracy boost if I ever get time to finish the matrix code. All the new row operations will be performed in higher precision with correct rounding.

Unfortunately, I don't think there is enough RAM to unpack the matrix and do a LU decomposition completely at the higher precision. I'll have a think though.

- Pauli

Re: Determinants of the AM1 matrix

Message #42 Posted by ***Marcus von Cube, Germany*** *on 1 Oct 2011, 4:22 a.m.,*
in response to message #41 by Paul Dale

Not regarding any pivoting, an LU decomposition should be doable in place so RAM isn't an issue here. Adding pivoting might get tricky because you need the information about row or column swaps somewhere. Maybe in a condensed format in a register (using binary)?

Re: Determinants of the AM1 matrix

Message #43 Posted by ***Paul Dale*** *on 1 Oct 2011, 4:40 a.m.,*
in response to message #42 by Marcus von Cube, Germany

I know that an LU decomposition can be done in place (if you don't mind overwriting the matrix itself) and the code I've written can do this. However, I was talking about unpacking the matrix registers into the higher precision internal format and doing the decomposition on these instead. The big advantage is greatly increased precision.

I am pretty sure there isn't enough volatile RAM to do this. 3600 bytes would be required just to store the unpacked matrix.

The space for the pivots is small -- ten integers and these can be packed into one register easily. The largest matrix that can be decomposed on the device is 10x10.

- Pauli

Re: Determinants of the AM1 matrix

Message #44 Posted by ***Werner*** *on 2 Oct 2011, 4:45 a.m.,*
in response to message #43 by Paul Dale

3600 bytes for 49 elements? That would amount to 70+ bytes per number, that is one hell of a precision.
Standard quadruple precision is only 16 bytes, delivering about 35 decimal digits of precision.
Why does it take so much memory?

Cheers, Werner

Re: Determinants of the AM1 matrix

Message #45 Posted by ***Paul Dale*** *on 2 Oct 2011, 5:19 a.m.,*
in response to message #44 by Werner

3,600 bytes is for 100 numbers (a 10x10 matrix). They are 36 bytes each in a not well packed decimal format.

The unpacked format I'm using internally is not terribly space efficient three digits in two bytes plus some overhead. All arithmetic is done in this format so any space saving measures will require packing and unpacking to operate on them.

We're running 39 digits internally. Quadruple precision reals are available and they consume 16 bytes and give 34 digits but require packing and unpacking into the internal format I'm already using. I'd rather not pay the flash space cost for these pack/unpack routines for just this function.

- Pauli

Re: Determinants of the AM1 matrix

Message #46 Posted by ***Paul Dale*** *on 1 Oct 2011, 2:52 a.m.,*
in response to message #1 by gene wright

WP 34S using the determinant command I just finished coding:

```
1.000000084391072
```

This also means I've coded the Doolittle algorithm for the LU decomposition and I should be able to use this for solving equations and find a matrix inverse in place :-)

Assuming this fits into the stack space available on the hardware. If it doesn't the answer will still be the same, but the matrix will be modified (like the 15C does).

- Pauli

Re: Determinants of the AM1 matrix

Message #47 Posted by ***Walter B*** *on 1 Oct 2011, 4:15 a.m.,*
in response to message #46 by Paul Dale

Great news! :-)

Re: Determinants of the AM1 matrix

Message #48 Posted by ***Paul Dale*** *on 1 Oct 2011, 4:42 a.m.,*
in response to message #47 by Walter B

The higher accuracy of this result should be achievable from a key stroke program by taking advantage of one of the multiply-add sequences available.

- Pauli

Re: Determinants of the AM1 matrix

Message #49 Posted by ***gene wright*** *on 1 Oct 2011, 9:33 a.m.,*
in response to message #48 by Paul Dale

That is a great improvement. It has almost caught up with the 9860G result. The three leaders now are:

```
Calc                         Value                Error                Ratio of error to best
Casio FX-9860G Slim          1.00000003696570     0.00000003696570
WP 34S New Pauli firmware    1.00000008439107     0.00000008439107          2.28
WP 34S RRM                   0.99999711935400     0.00000288064600          77.93
```

Re: Determinants of the AM1 matrix

Message #50 Posted by ***Paul Dale*** *on 1 Oct 2011, 8:29 p.m.,*
in response to message #49 by gene wright

You missed the highly accurate recoding which returns 1 exactly :-) We're checking if it is going to fit into available RAM or not. One solution here is to not allow 10x10 determinants and inverses -- limit it to 9x9 or 8x8 instead. I'm not too keen on this. What do folks here think about limiting the maximum invertible matrix size?

I'm sure the RRM program can be modified to achieve the same accuracy as the firmware you mention. The multiply-adds need to be recoded using complex multiplication which will take a bit of stack shuffling and a few extra steps.

- Pauli

Re: Determinants of the AM1 matrix

Message #51 Posted by ***Walter B*** *on 2 Oct 2011, 2:41 a.m.,*
in response to message #50 by Paul Dale

I can live with 9x9 as well :-)

Re: Determinants of the AM1 matrix

Message #52 Posted by ***Paul Dale*** *on 2 Oct 2011, 5:32 a.m.,*
in response to message #50 by Paul Dale

Actually, the high precision determinant is calculated as 1.0000000000000000000000000000396321014

This rounds to exactly 1 when reduced to the 16 digits stored in registers.

- Pauli

Re: Determinants of the AM1 matrix

Message #53 Posted by ***gene wright*** *on 2 Oct 2011, 7:16 a.m.,*
in response to message #52 by Paul Dale

I saw the highly accurate result. Just was not going to update the table until it was in a published flash.

9x9 ought to be good enough for anyone - Bill Gates

Re: Determinants of the AM1 matrix

Message #54 Posted by ***Paul Dale*** *on 2 Oct 2011, 7:49 a.m.,*
in response to message #53 by gene wright

Yeah, getting it to fit into RAM looks difficult to impossible. However, I've had an idea which might just work and give only slightly worse results. I'll try to code something up tomorrow and see how it turns out.

- Pauli

Re: Determinants of the AM1 matrix

Message #55 Posted by ***Marcus von Cube, Germany*** *on 2 Oct 2011, 12:49 p.m.,*
in response to message #54 by Paul Dale

I did some coding. The software still supports a 10x10 LU decomposition without disturbing the original matrix but the accuracy isn't the full 39 digits. I had to reduce it to 21 digits to make it fit. We are working on a better solution.

Re: Determinants of the AM1 matrix

Message #56 Posted by ***Paul Dale*** *on 3 Oct 2011, 5:45 a.m.,*
in response to message #53 by gene wright

Gene, time to update the 34S's determinant result, we have an image :-)

- Pauli

Re: Determinants of the AM1 matrix

Message #57 Posted by ***Paul Dale*** *on 2 Oct 2011, 7:00 p.m.,*
in response to message #52 by Paul Dale

I managed to get the high precision determinant fitting into RAM and giving a result of 1.00000000000000000000000008048808860320 which rounds to 1 exactly when returned to the stack.

Now to see how much flash this is costing us...

- Pauli

Re: Determinants of the AM1 matrix

Message #58 Posted by ***Ángel Martin*** *on 4 Oct 2011, 4:36 a.m.,*
in response to message #50 by Paul Dale

I don't know, probably as soon as you limit the size to 9x9 you'll get users needing support of higher dimensions, even if not such a precise result.

It's a trade-off, but you guys know what you're doing.

Re: Determinants of the AM1 matrix

Message #59 Posted by ***Marcus von Cube, Germany*** *on 4 Oct 2011, 4:41 a.m.,*
in response to message #58 by Ángel Martin

It's all set. Pauli managed to do an LU decomposition to 33 digits in available volatile memory for up to 10x10 elements. The trick was using the decimal128 packed format for the data.

More then 10x10 will not be possible with just 100 registers.

Re: Determinants of the AM1 matrix

Message #60 Posted by ***Paul Dale*** *on 4 Oct 2011, 4:47 a.m.,*
in response to message #59 by Marcus von Cube, Germany

There are 34 digits in a decimal128 actually. Instead of the 39 we usually use internally and the 16 in a decimal64 (our register format).

The calculated results are:

```
full precision: 1.00000000000000000000000000000396321014
decimal128:     1.00000000000000000000000000008048808860320
decimal64:      1.000000084391072
```

The first two round to exactly 1 of course.

The reason there are 39 digits in the decimal128 result is that I'm still doing the accumulations and multiplications using the 39 digit numbers. The 34 digit ones are just for storing the LU decomposition of the matrix as it is being calculated.

- Pauli

Re: Determinants of the AM1 matrix

Message #61 Posted by ***Werner*** *on 4 Oct 2011, 5:54 a.m.,*
in response to message #60 by Paul Dale

I would say even 16 digits (and 39 for the inner products, reminiscent of 'floating-2') is plenty for solving 10x10 systems.
It is better to get an indication that your matrix condition is bad than to get a too-good-to-be-true result...

Cheers, Werner

Re: Determinants of the AM1 matrix

Message #62 Posted by ***Paul Dale*** *on 1 Oct 2011, 5:37 a.m.,*
in response to message #46 by Paul Dale

I've tried the same problem converting the matrix to internal precision and performing all operations at this precision. The determinant calculation gives a result of 1.000000000000000 when rounded back to register precision. Now to see if we can squeeze this version of the code into the hardware constraints -- I'm not confident it is possible.

- Pauli

Re: Determinants of the AM1 matrix

Message #63 Posted by ***Dieter*** *on 2 Oct 2011, 9:39 a.m.,*
in response to message #1 by gene wright

Very interesting - please take a closer look at this:

> Quote:
>
> ```
> HP-15C              1.080204421
> HP-35s              1.00282960115
> ```

The 15C uses its internal 13-digit precision and (supposedly) a state-of-the-art algorithm. Nevertheless, the better result here comes from the 35s which has to rely on the 12 digits available for user programs. Either the 35s matrix algorithm is absolutely great or this result is a positive exception and the 35s is otherwise outperformed by the 15C. What do you think?

Dieter

Re: Determinants of the AM1 matrix

Message #64 Posted by ***Palmer O. Hanson, Jr.*** *on 2 Oct 2011, 11:27 p.m.,*
in response to message #63 by Dieter

HP-15C 1.080204421

HP-35s 1.00282960115

TI-59 0.9983600987185

TI-95 1.000676708424

> Quote:
>
> The 15C uses its internal 13-digit precision and (supposedly) a state-of-the-art algorithm. Nevertheless, the better result here comes from the 35s which has to rely on the 12 digits available for user programs. Either the 35s matrix algorithm is absolutely great or this result is a positive exception and the 35s is otherwise outperformed by the 15C. What do you think?

The TI-59 and TI-95 both use 13 digit precision but the TI-59 truncates rather than rounds to the 13th digit. In addition, the TI-59 has the multiplication anomaly. But, both machines beat the HP-15C result. Kahan's "Mathematics Written in Sand" paper gave yhe following illustration of the capability of the HP-15C:

> Quote:
>
> ... For example, let E be a multiple of the notorious Hilbert matrix E(i,j) = 360360/(i+j-1) for 1<= i,j <= 8. The constant 360360 ensures that every element of E is an integer, hence exact, and 8x8 is as large a matrix as fits in the calculator. In under 90 seconds it gets the inverse of E correct to roughly three sig. dec., three more than are expected ... ... This extra accuracy is no accident with ill-conditioned matries like E, prone to systemic concellation, but is due to the extra-precise accumulation of scalar products to 13 sig. dec. during matrix operations.

My recollection (I will have to search for the results) is that the ML-02 program in the TI-59 Master Library module returns an inverse which is correct to approximately five significant digits. What do you think of that?

Re: Determinants of the AM1 matrix

Message #65 Posted by ***Werner*** *on 3 Oct 2011, 2:11 a.m.,*
in response to message #64 by Palmer O. Hanson, Jr.

The condition number for AM1 is about 9e10, meaning you stand to lose 11 digits and thus can't trust your result at all on a 10-digit calculator. Depending on the algorithm used, however, you might get lucky.
LU-factorisation can be performed resulting in either a unit lower triangular matrix L or a unit upper triangular matrix U. The 42S does the former. If you calculate the determinant of AM1 on a 42S, you get:

```
    0.970960198039
```

By no means an exceptional result, seeing that it uses 12 digits and accumulates dot products with the 15 digit internal precision. It indeed loses 11 digits in the process. Now calculate the determinant of TRANS(AM1), which amounts to simulating the L*unitU factorisation:

```
    0.999403205613
```

All of a sudden, we have a result that is 2 digits better. That's what happened in the 15C as well, I gather. It uses 10-digit arithmetic with 13-digit inner products, but uses a different order of operations. Anyone willing to calculate DET(TRANS(AM1)) on a 15C and post the results?

Cheers, Werner

[Return to Index | Top of Index]

Go back to the main exhibit hall