HP Forum Archive 13

[Return to Index | Top of Index]

Eigenvalues on 42S?

*Message #1 Posted by **Steven Soto** on 25 Aug 2003, 2:11 p.m.*

Does anyone know how to find all of the eigenvalues for a 3x3 matrix on a 42s? I know you can find ONE using the solver, but I'd like to be able to find all three, like on a 48g. Can this be done with "standard" matrix operations like inverse, determinant, etc.?

Re: Eigenvalues on 42S?

*Message #2 Posted by **Patrick** on 25 Aug 2003, 2:49 p.m.,*
in response to message #1 by Steven Soto

I can't think of a direct method.

Since you are only interested in a 3x3 (not nxn) you could go back to first principles. Take an arbitrary 3x3 matrix, A =[a_{ij}], and expand the expression det(A-xI) as a cubic polynomial in x. The roots of this cubic are, of course, the eigenvalues. The coefficients of the cubic should be readily expressible in terms of determinants of various sub-matrices of A (you could probably even find this written out explicitly in a beginner's book on eigenvectors).

Write a program which computes the cubic's coefficients using the sub-matrix and determinant capabilities of the 42s, then feed the resulting cubic into the solver.

Re: Eigenvalues on 42S?

*Message #3 Posted by **Steven Soto** on 25 Aug 2003, 3:12 p.m.,*
in response to message #2 by Patrick

Exactly my problem. I already knew I could get one from the solver: I want to know how to get all three. I guess I'll have to manually factor out the one result from the solver and then put the remaining result into the quadratic equation. Any ideas on how to write a program to do this that will simply output three values?

Too late in Hungary

*Message #4 Posted by **Tizedes Csaba** on 25 Aug 2003, 5:03 p.m.,*
in response to message #3 by Steven Soto

Hi Steven!

If you can wait to next day, I will help you. Too late here, and the PC's keyboard is clicking, and my girlfriend not like that in the night.

I haven't got 42S, I'll write a little program for 32SII, what can solve ANY REAL roots of an equation. (I'm hoping... :))

A part of it is ready to run, the solver part I'll make tomorrow morning...!

Good night! Csaba

Algorithm

*Message #5 Posted by **Patrick** on 25 Aug 2003, 7:52 p.m.,*
in response to message #3 by Steven Soto

The cubic equation

$p(x) = \det(A-xI) = c_3x^3 + c_2x^2 + c_1x + c_0$

has the following coefficients:

$c_0 = \det(A)$

$c_1 = -(\text{sum of all 2x2 determinants obtained by deleting the same row and column from A})$
$= -((a_{11}a_{22} - a_{12}a_{21}) + (a_{11}a_{33} - a_{13}a_{31}) + (a_{22}a_{33} - a_{23}a_{32}))$

$c_2 = \text{sum of all 1x1 determinants obtained by deleting the same two rows and columns from A}$
$= a_{11} + a_{22} + a_{33}$

$c_3 = -1$

Now, if a cubic, p(x), has 3 real roots, then they occur between successive extrema of the cubic. If the extrema are located at $x=x_1$ and $x=x_2$, with $x_1 < x_2$, then the roots of the cubic are in the three ranges:

$(-\text{inf}, x_1)$
(x_1, x_2)
$(x_2, +\text{inf})$

You can compute x_1 and x_2 as the roots of a quadratic:

$p'(x) = 3c_3x^2 + 2c_2x + c_1$

Once you have these roots, you can feed the ranges above into the solver as guesses. Of course, you'd have to do something intelligent in place of the infinities.

Nth order solver

*Message #6 Posted by **Tizedes Csaba** on 26 Aug 2003, 5:26 a.m.,*
in response to message #3 by Steven Soto

Hi Steven,

this is a solution for your problem. This program can solve polinom-equations ANY of order (limited by mem). It's not too long, only 92 byte ;). It work fast with built-in solver, and use Horner-method:

Only one problem with it, it's solve JUST REAL ROOTS... But in the real problems, like mechanical stress calculations, the roots are reals.

This is the HP32SII version:

```
E01 LBL E        #prepare of 'i'
    FIX 3
    RCL N
    1
    add
    1E3
    div
    1
    add
    RND
    STO i
    0
E13 RTN

P01 LBL P        #the 'dinamical polynom'
    XEQ E
H01 LBL H          #Horner's method
    RCL mul X
    RCL add (i)
    ISG i
    GTO H
H06 RTN

X01 LBL X        #polynom solver
    SOLVE X
```

```
    GTO R          #if found root goto 'R'
    SF 0           #no real root, set flag 0
X05 RTN


D01 LBL D        #decrement polynom order,
    XEQ E
J01 LBL J          #use a little modified Horner method
    RCL mul X
    RCL add (i)
    STO (i)        #overwrite coefficients
    ISG i          #with decremented polyn.'s coefficients
    GTO J
J07 GTO X


S01 LBL S        #start-routine
    CF 0
    CF 1
    INPUT N
S05 XEQ E
I01 LBL I         #coefficient input-routine
    INPUT (i)
    ISG i
    GTO I
    FN= P
I06 GTO X


R01 LBL R       #we found a root
    VIEW X
    1
    STO sub N
    VIEW N       #number of roots, what we dont found yet
    RCL N
    x<>0?        #more root?
    GTO D        #go decrement order, and solve again
    SF 1         #no more root, and all was REAL: set flag 1
R10 RTN



'add', 'sub', 'mul' and 'div' mean 'add', 'substract', 'multiply' and 'divide' in this order.



The checksums:


E:CK=0BE8 27.5 byte
P:CK=E452  3.0 byte
H:CK=3E78  9.0 byte
X:CK=3894  7.5 byte
D:CK=11DF  3.0 byte
J:CK=C765 10.5 byte
S:CK=BB8E  7.5 byte
I:CK=32B5  9.0 byte
R:CK=E775 15.0 byte



An example:


solve this polynom with this program:



-2*x^6 -6*x^5 +82*x^4 +174*x^3 -800*x^2 -888*x +1440 =0


 A*X^6  +B*X^5 +C*X^4   +D*X^3   +E*X^2   +F*X    +G=0
```

```
In the program the coefficients are A, B, C, ... The limit of order N=12, because the 14th variable is N, and we use this variable for the order of solved polynom. If you replace it with Z, you can solve the 24

Now, start the program: [XEQ S] the calc prompting for order of polynom:

N? type 6 [R/S].

Then the program prompts for all of coefficients, type it in decrease order:

A? -2 [R/S]
B? -6 [R/S]
C? 82 [R/S]
D? 174 [R/S]
E? -800 [R/S]
F? -888 [R/S]
G? 1440 [R/S]

Then you will see the 'RUNNING/SOLVING' message, and the program stops with one of six roots: for example X=1. Press [R/S], and you'll see N=5 this mean, the polynom's got 5 root yet. Press [R/S] and repeat thi

When N=0 press [R/S], the 'flag1' anunciator set and program end. 'flag1' set mean, all of roots was real.

This polynom got 6 real roots: 1, -2, 3, -4, 5, -6, and the simplyfied form is: -2*(x-1)*(x+2)*(x-3)*(x+4)*(x-5)*(x+6)
```

I hope it help to you and anybody, who want to solve polynomials. The roots are 'more exacts' if the round off error is not a very big.

Csaba

Re: Nth order solver

*Message #7 Posted by **Steven Soto** on 26 Aug 2003, 8:01 a.m.,*
in response to message #6 by Tizedes Csaba

Tizedes,

Thanks. Very helpful and useful. I'm still amazed by the power of the pioneer series. I have used these calculators since 1988 starting with a 32s my freshman year in high school. I got a 42s as a senior in high school, and I still use that calculator today. I have a spare that I have tested, and never used which sits in a hard case without batteries. I have used this calculator all through undergraduate school and graduate school while everyone else seemed to migrate to TI's, I have yet to find something that I cannot get my 42s to do that I need it to.

And yes, this eigenvalue problem was to solve three dimensional stress states.

-Steven

Pocket calculators forever

*Message #8 Posted by **Tizedes Csaba** on 27 Aug 2003, 2:09 a.m.,*
in response to message #7 by Steven Soto

Steven,

don't mention it! I made it with pleasure! :) My friends said I'm crazy for that what haven't got more keys than 50 and LCD resolution less than 160x160 with 1 colour. :)

And it's true!

Csaba

I am adding and subtracting I'm controlling and composing

Re: Eigenvalues on 42S?

*Message #9 Posted by **Valentin Albillo** on 26 Aug 2003, 10:57 a.m.,*
in response to message #1 by Steven Soto

If your matrix is symmetric (which is usually the case in many real-life engineering problems), the eigenvalues will all be real, and Jacobi's method will produce them all at once, to full precision. Jacobi's method uses 2x2 rotation matrices applied iteratively to any arbitrary symmetric NxN matrix, till it reduces it to a diagonal matrix. The eigenvalues are then precisely the diagonal elements, and the corresponding eigenvectors are produced as well as a side effect.

This algorithm is very straightforward, always converges without requiring any input from the user other than the NxN matrix itself, and converges very quickly, producing accurate results (great stability as well). Further, the transformation is made *in-place*, thus requiring a minimum amount of memory.

I wrote such a program for the barebones HP-41C, which should run unchanged, as is, in any HP42S. It was published in the PPC Journal, circa 1980 or so, a little search will find it. It is only 100+ program steps, so you can key it in effortlessly.

Else (or also), have a look at this excellent paper (PDF format):

Eigenvalues of a symmetric matrix: Jacobi's method

which includes a full, detailed description of the method, as well as a program listing.

Best regards from V.

Re: Eigenvalues on 42S?

*Message #10 Posted by **hugh** on 26 Aug 2003, 12:06 p.m.,*
in response to message #9 by Valentin Albillo

the method is also coded for the 15c in the advanced functions handbook page 148.

however, i do like csaba's program as a good programmatic use of the solve feature.

something ive been doing recently is the same problem the other way around. ive been finding it much better to use an eigenvalue approach to polynomial roots rather than the reverse. you also get the complex roots a lot easier this way too.

Re: Eigenvalues on 42S?

*Message #11 Posted by **Veli-Pekka Nousiainen** on 26 Aug 2003, 10:45 p.m.,*
in response to message #10 by hugh

"for the 15c in the advanced functions handbook page 148"

Can you send it here or to me directly?

VPN

Re: Eigenvalues on 42S?

*Message #12 Posted by **Veli-Pekka Nousiainen** on 26 Aug 2003, 10:47 p.m.,*
in response to message #9 by Valentin Albillo

"I wrote such a program for the barebones HP-41C"

Can you send it here or to me directly?

VPN

Re: Eigenvalues on 42S?

*Message #13 Posted by **Valentin Albillo** on 27 Aug 2003, 4:29 a.m.,*

in response to message #12 by Veli-Pekka Nousiainen

Thanks for your interest re my "Eigenvalues for an NxN symmetrical matrix: The Jacobi Method" article and program for the barebones HP-41C .

I initially thought it was published in "PPC Journal", V7 or V8, but a cursory search in the PDF pages available online failed to locate it, which means it was published in "PPC Technical Notes" instead, which alas, has no online presence as far as I can tell.

If you've got the first 10 issues or so of "PPC Technical Notes" (circa 1980 or 1981), you'll find it there for sure.

I remember it was heavily optimized, took just 100+ steps or so (thus fitting in a single card, maybe even in just one side of a card), and could handle very large matrices as it made use of the fact that the matrix was symmetrical, so it only needed to store N*(N+1)/2 elements (i.e: only 120 elements for a 15x15 matrix) instead of N*N (i.e: 225 elements for said 15x15 matrix). The N eigenvalues were found all at once, simultaneously (not one by one), and very quickly.

Best regards from V.

Edited: 27 Aug 2003, 4:39 a.m.

[Return to Index | Top of Index]

GTO Go back to the main exhibit hall