ANASEC67

Paul KOPFF

A program for the HP67 pocket computer for :

computing the cross-sectionnal characteristics of beams

What for ?

For beam bending computations, mechanical and civil engineers need some of their cross-sectionnal characteristics, that they have sometimes to compute before engaging the proper stress/strain analysis (because : whereas for classic sections of beams these characteristics are usually tabulated, when considering "special" beams, they may have to be computed by discretizing their shapes).

The simplest such characteristic is the cross-section area : which is defined as an integration of dxdy inside the (eventually complex) shape. ANASEC67 integrates also xdxdy, ydxdy, x^2dxdy, y^2dxdy and xydxdy.

How is it done ?

When the cross-section shape of a beam may be discretized as delimited by a polygon if it is not hollow, and by several polygons if it has hollows (one polygon for its external border, and one for each of its hollows) :

The only data needed are the x-y coordinates of the angles of these polygons (in any suitable orthogonal 2D coordinate system). The program should not need to enter these coordinates more than necessary (more than once). And the computations should progress as the data are entered, not after the complete description of the cross-section (because otherwise, for entering arbitrarily complex shapes, you could need more memory than the – very few – registers an HP67 provides).

So, I used computing techniques called "integral methods" : these techniques replace formulas (like dxdy, xdxdy, x^2dxdy etc.) that are to be integrated inside some area (a double integral) by other formulas that yield the same results as they are integrated along some border (a simple integral). And this simple integration can progress as the coordinates of the discretization are entered, needing only to memorize a few partial results.

The replacing formulas were designed in such a way that the external borders must be followed "counter-clock-wise" when entering data, and the hollows "clock-wise".

How does it work ?

The HP67 has 5 programmable function keys labeled A B C D E.

The E key is used to reset the registers and flags, to prepare the computer for another computation.

The A key is used to enter the pair of coordinates of a given angle of a given polygon. Type :

<coordinate x> [ENTER] <coordinate y> [A]

Having entered the coordinates of all angles of a given polygon, **the B key** is used to retrieve the coordinates of the first angle, to close the polygon and finish the current integration. Press :

[B]

Then you can do the same to enter the data of all the other polygons that may represent the hollows. After entering the data of all polygons of a given shape, **the C key** is used to tell the program that all the integrations are finished, and to display the results. Press :

[C]

The D key is used to display the results again (and again...).

At the end of the computation (after pressing key C), the registers contain following results :

R0 : the area of the cross-section :
    R0 : integration of dxdy
R1 R2 : the first order moments :
    R1 : integration of xdxdy
    R2 : integration of ydxdy
R3 R4 R5 : the second order moments
    R3 : integration of x^2dxdy
    R4 : integration of y^2dxdy
    R5 : integration of xydxdy
R6 R7 : the coordinates of the barycenter : zero-location of the first order moments
    R6 : x_G = R1 / R0
    R7 : y_G = R2 / R0
R8 R9 RA : second order moments referenced to the barycenter
    R8 : integration of $(x-x_G)^2$dxdy
    R9 : integration of $(y-y_G)^2$dxdy
    RA : integration of $(x-x_G)(y-y_G)$dxdy
RB : radial second order moment :
    RB : integration of $((x-x_G)^2+(y-y_G)^2)$dxdy
RC RD : principal second order moments : after rotation xy>uv extremal second order moments
    RC : integration of u^2dxdy
    RD : integration of v^2dxdy
RE : angle of axis uv with axis xy

In the HP41, (R0-R9)P are (R0-R9), (R0-R9)S are (R10-R19), (RA-RE) are (R20-R24), RI is R25. So when my program executes keys C or D it displays R0-R25, but R10-R19 and R25 are garbage.

The program uses (and sets/resets with key E) flags 1 and 2 for controling the computation.

Flag 0 can be set before running the progam : in this case, as data are entered they are printed on the HP97 (and the virtual printer of the emulation).

On the HP97 (and the virtual printer of the emulation) keys C and D will not only display the results but also print them.

Of course before using the program, the [DISP]<n>,[FIX],[SCI],[ENG] and [DEG],[RAD],[GRAD] should be set to the user's preferences.

This program uses exactly the 224 steps of program memory, and excercises almost all the capabilities of the HP67 pocket computer. Thus, it may also be considered a good test for the HP67 emulations.

In those times, the programs tended to use the whole available room, today it's impossible!