Interactive geometry for the HP49

Bernard Parisse

14 juin 2017

1 Introduction, software licence.

This document describes the HP49 ROM system update, with an interactive analytic geometry software. The main fonctionnalities of this module are

- Creation of geometrical objects, lines, segments, circles, curves (function graphs and parametric curves). These objects may be approximate objects (like for most geometry software) or exact, in that case computation are done using the computer algebra system, the system may for example return $\sqrt{2}$ as a circle radius (instead of 1.414...). All objects may be constructed from within the geometry application or using command from the history or inside a program.
- The geometry interactive application handles figures, i.e. a collection of geometrical objects with connections. Dynamic geometry is implemented, it is possible to move points with the arrow keys and see how the figure changes.
- All constructed objects within the interface are stored in user variables, they may be used from the history for analytic geometry computation.
- Conversely, one can program geometry functions like usual functions and use these new functions inside the interactive application. All geometrical constructions correspond to a commandname which is inspired by the `xcas` software : `http://www-fourier.ujf-grenoble.fr/~parisse/giac.html`

Software License :

The geometry software application is licensed under the General Public License (GPL) version 2 as published by the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA, in conjonction with the LGPL HP49 CAS release (Lesser General Public License version 2), cf.
`http://www-fourier.ujf-grenoble.fr/~parisse`
Note that the (L)GPL license do not apply to this documentation

2 Installation

Get a ROM update software at `http://www.hpcalc.org` for Windows or Linux. Follow the installation instructions but replace the ROM file by
`ftp://www-fourier.ujf-grenoble.fr/pub/xcas/hpcas/Vger.flash`
For example under Linux, get `hp49flash.tgz`, compile it and run in a Terminal the command :
`./upgrade Vger.flash 0`
(replace 0 by 1, 2 or 3 depending on the serial port where you connect the calculator) and follow the instructions on the screen.

3 A first construction

We will construct a triangle from three random points and it's circonscircle.

1. Type `GEO` (type the alpha key twice to lock alpha mode) to run the geometry application.
2. You can suppress axes by typing `F6` (Config) followed by `Remove axes`
3. Type `F1` (File) then select `New`, give a variable name for the construction, e.g. `CIRCON`
4. Type `F2` (Add) select `Points` then `aleatoire` (random), enter 3 point names `A B C` (note that the keyboard is locked in alpha mode). The 3 random points should appear on the screen.
5. Type `F2` (Add) select `Lines` then `mediatrice`, you should see the commandline with `mediatrice()`
   the cursor should be after the open parenthesis, give the name of the two A and B points, separated by a comma, you should now have in the commandline `mediatrice(A,B)`
   type `ENTER`, now give the label name of the mediatrice (this name will be displayed on the screen) and the variable name of the mediatrice, e.g. `:c:c`. If you don't want to display the label on the screen, type `::c`. Note that you must give a variable name for the mediatrice.
6. Do the same for the mediatrices `b` (points `A` and `C`) and `c` (points `B` and `C`).
7. Type again `F2`, select `Points` then `inter`, type the variable name of the two mediatrices, e.g. `a` and `b`, give a label and variable name e.g. `:O:O`.
8. Type again `F2`, then `Curves`, then `cercle`, give as arguments the center `O` and one of the 3 points, e.g. `A`, give a label and variable name for the circle, e.g. `:S:S`.
9. Type `F1` then `Save` to save the `CIRCON` construction.
10. Now we will move one of the 3 points and see how the figure changes. Type `F4` (Move), select `A`, `B` or `C` and move the point with the arrow keys (use the shifts or/and alpha before an arrow key to move faster or slower). Type F1 (Ok) or F6 (Cancel) to finish the move, in the first case you accept the final position, in the second case you come back to the initial position.

11. To leave the application, type `F1` then `Quit`.
12. You are now back in history, where you can run geometry command like for example `affixe(A)` or `rayon(S)` (radius of the circle). You can select these commands from the Mesures softmenu.
13. To run again the construction, type again `GEO`, then `F1` then `Load` and select the figure `CIRCON`.
14. You can also edit the whole construction from the F4 menu, `Edit figure` submenu. Outside the geometry environment, you can edit the variable.

4 The geometry environment

From history, type `GEO` to enter the interactive geometry environment. The main application menu is activated by typing one of the 6 keys F1 to F6. The arrow keys and +/- keys let you modify the visualisation window. Some other key shortcuts gve you direct access to submenu items.

Before beginning a construction, select exact or approx mode depending on what you want to do with your construction. If you want to move points and see the resulting effect (dynamic geometry), select approx mode. Exact mode is more appropriate if you plan to do analytic geometry (including computation of geometric loci). Check also that the `X`, `Y` and `t` variables are free.

In the following subsections, commandnames are written in `typewriter` font.

4.1 File menu (F1)

Use it to load/save constructions.
- New (shortcut N) : make a new construction, give a variable name
- Save (shortcut STO) : save the current construction
- Save as (shortcut right shift STO) : save the construction under a new variable name
- Load (shortcut left shift STO) : load a construction from current directory
- Info : display the current construction in text mode
- Help : display a short help on the environment and list the keyboard shortcuts
- Quit : leave the environment

4.2 Add Menu

Add a new object to the construction. Geometric object have a variable name et a non mandatory label name (the label if present is displayed on the screen), the input format for such an object is `:label:variable_name`.

4.2.1 Points (shortcut P)

- aleatoire (shortcut R) : make some random points. You must give the variable names of the points. They are created with the same label as the variable name.

- `point` : enter the 2 coordinates then the label/variablename. Note that if you are using formal coordinates (like `point(a,b)` where `a` and `b` have no value assigned), check that these variables are real (see REALASSUME) or that the real variables flag is checked (MODES key, then FLAG, go to the last flag 128)
- `milieu` : midpoint of 2 points, enter the two points names then the label/name of the midpoint.
  It's synonym `barycentre` may be used from the command menu (cf. infra) to compute the isobarycenter of a list of points, like this
  `<< { A B C } 1 barycentre >>`
  It may also be used to compute the barycenter of a list of points and a list of coefficients, like this
  `<< { A B C } { 1 2 3 } barycentre >>`
- `centre` : center of a circle.
  The synonym `sommets` may be used to get the list of vertices of a polygon. For example, to get the third vertex of a polygon named `Q`, type
  `<< Q centre 3 GET>>`
- `inter` : intersection of 2 geometric objects. For 2 lines returns one point (undef if parallel), for a line and a circle or for 2 circle, returns 2 points (twice the same if tangent, twice undef if not secant) géométriques. Other intersections return a list of points. If both objects are parametric curves, one of them must have a rational parametrization to get it's cartesian equation by resultant computation.
- `element` : make a point on a line, circle or curve.
- `zxpnt` : make a point given it's complex affix.

4.2.2 Lines (shortcut L)

- `segment` : from 2 points or complex numbers
- `droite` : like segment
- `parallele` or `tangent` : draws the parallel through a point to a line or the tangent at one point to a curve
- `perpendiculaire` : draw the normal through a point to a line
- `mediatrice` : médiatrix of 2 points
- `mediane` : line from point 1, to the midpoint of points 2 and 3
- `bissectrice` : bissectrice from point 1 of the triangle made by the 3 points
- `hauteur` : normal from point 1 to the line point2-point3
- `eqxpnt` : make a line or a circle given a cartesian equation in X and Y

4.2.3 Circles, curves (shortcut S)

- `cercle` : circle given by center (point or complex number) and radius (a real) or a point on the circle
- circonscrit : circle passing through 3 points.
  There is no corresponding command, it is translated to a call to `cercle` with the list of points as argument 1 and 0 as argument 2
  `<< { A B C } 0 cercle >>`
- inscrit : cercle inside and tangent to the triangle of vertices the 3 points.

There is no corresponding command, it is translated to a call to `cercle` with the list of points as argument 1 and 1 as argument 2
`<< { A B C } 1 cercle >>`
Note that the computation of this circle in exact mode requires simplification of expressions with square roots and may therefore be quite long.

- `plot` : draws the graph of an expression in X. Note that the argument is an expression, not a function. For example `plot(X^2-1)`.
- `paramplot` : draws the graph of a parametric curve given by 2 expressions $X(t)$ and $Y(t)$. Note that the arguments are expressions, not functions. Note also that $X(t)$ and $Y(t)$ may be complex numbers, the systems draws $X(t) + iY(t)$. It is therefore easy to draw a parametric curve if you know it's affixe $Z(t)$, take $Z(t)$ as argument 1 and 0 as argument 2. If argument 2 is 0 and argument 1 is a point, it will also work. If argument 2 is 0 and argument 1 is a line, then the enveloppe of the lines will be drawn. Examples :
  `paramplot(COS(2*t),COS(3*t)`
  `paramplot(EXP(i*t),0)`
  `paramplot(eqxpnt(X*SIN(t)+Y*COS(t)-1),0)`
- `lieu` : locus of a point or enveloppe of lines. The system asks for the point on a line, circle or curve, then for the depending point (or line for enveloppes). It computes the parametric equation of the depending object.
  Computation time may be quite long.
  Since the computation of a locus with the first point on a line or circle is done with rational parametrization, simplifications may occur in exact mode, it is therefore recommanded to be in exact mode for locus computation (switching to exact mode just before locus computation is not sufficient, you must make the entire construction in exact mode. If it is already done in approx mode, go back to history and run `XQ` on the construction variable name).
  If you want to use the `lieu` command from history, note that it takes 3 arguments, the figure name being argument 1.

4.2.4 Triangle, quadrilateres

- `triangle` : triangle given 3 vertices
- isocèle : isocele triangle given by 2 points and an angle
  Commandname is also `triangle`
- equilateral : like isocele but the angle argument is $\pi/3$
- `quadrilatere` : polygon with 4 points as vertices
- parallelogramme : parallelogram given by 3 points. The system asks for a label/variablename for point 4.
  Commandname is `quadrilatere` with 0 as argument 4.
- carre : direct square given by 2 points. The systems asks for label/variablename of points 3 and 4. Commandname is `quadrilatere` with $\pi/2$ as argument 3 and 1 as argument 4.
- losange : losange given by 2 points, asks for points 3 and 4 label/name. Commande is `quadrilatere` with angle as argument 2 and 1 as argument 4.
- rectangle : rectangle given by 2 points and a length ratio, asks for points 3 and 4

label/names. Command is `quadrilatere` with $\pi/2$ as argument 3 and length ratio as argument 4.

For polygons with more than 4 vertices, use either `zxpnt` with the vector of affixes of the vertices as argument, or `droite` with argument 1 the list of vertices and argument 2 the binary integer `#0`.

4.2.5 Transformations

- `translation` : argument 1 is a segment or the affix of a vector. Argument 2 is the object to translate.
- `symetrie` : central symmetry if argument 1 is a point, or orthogonal symmetry if it's a line. Argument 2 is the object to transform.
- `rotation` : rotation given by the center (argument 1) and the angle (argument 2). Argument 3 is the object to transform.
- `homothetie` : homothetie given by the center (argument 1) and a ratio (argument 2). Argument 3 is the object to transform. If the ratio is a complex number, it's a similitude.
- `projection` : projection on a line (argument 1) of argument 2 (must be a point).
- `inversion` : inversion of centre argument 1.

You can define a transformation with fixed parameter in the history, like this :
`DEFINE(rA(X)=rotation(A,pi/3,X)`
which defines the rotation r_A of centre A, angle $pi/3$, that you can use later from command submenu (e.g. `rA(B)` will compute the image of B by the rotation r_A).

4.2.6 Legend

Use it to displays the result of a formula evaluation in the screen at a given position (a point position or at any complex number). For example, to display the abscisse of a point M at the point of coordinates $(2, 1)$, type `abscisse(M)` (you can get abscisse from the Mesures softkey menu, deselect the alpha mode by pressing the alpha key, press F6, press the abscisse softkey, then type the alpha and M keys), then `(2,1)` (you could also type a pointname here), then type the "fixed" part of the message which will be displayed, e.g. `"Mx="`.

4.2.7 Save formula

Allows to save a list of computed expressions in a variable when one moves a point (dynamic geometry). Each move makes the computation happen, and the results are added to the variable (previous computation are not replaced). You can study the accumulated numeric values later from history or from another application.

4.2.8 Command

Defines a geometric object by a command, which is either an algebraic object or a RPN program. For example :

`<< A B C D E A 6 ->LIST #0 droite >>`
draws a closed polygon with 5 vertex.

All HP49 commands may be used, the constraint is that the command does not take any argument from the stack and returns exactly one argument.

4.3 Remove (F3)

Select an object to be removed from the construction.

4.4 Move (F4)

Select a point to be moved with the arrow keys, or edit the whole construction in text mode.

If you select moving a point, you will have to wait a few seconds before moving begins (the system compute the fixed and moving part of the figures). If the selected point is on a curve, moving is constrainted on the curve. Use the alpha and shift before the arrow keys to move slower or faster. If you hit the arrow keys fast, the figure may not be fully redrawn before the next keystroke is taken in account. Type F1 (Ok) to accept the new position or F6 (Cancel) to go back to the initial position. Note that the first move may ask you to switch to approx mode if you were in exact mode, answer yes.

It is not possible to move a point when a geometric locus depends on it. You must edit the figure and change the point coordinates by hand. If the locus is defined before the point, moving the point is allowed.

If you edit the figure, you will see a list of lists, each geometrical object is a list made of 2 objects, the computation formula to get the object (e.g. `'mediatrice(A,B)'`) and the object name (format `:label:variable_name`). For example `:c:c` would display `c`, `::c` or `c` will not display a label).

4.5 Pict (F5)

Gives access to the graphic environment of the HP49. If you hit the F2 key (X,Y), moving the cursor will display the position of the cursor which can help locate the geometric objects. Press ON to go back to the geometry environment.

4.6 Config (F6)

Configure application.

- Exact : exact mode. All computation will be done with the CAS system if possible. Answers may be used to make a mathematical proof. Some computations may be slow.
- Approx : approx mode. Computations are faster but roundoff occurs. You can not use these answers to make mathematical proofs, only guesses.
- Xminmax, Yminmax : define the visualisation window. Also used to define the limit value of t for parametric curves (Indep-low and high) as well as the step value of t between 2 successives evaluations.

– Tick : defines the tickmark on the axis (H-Tick and V-Tick) and allows to choose the unit as pixel or not.
– Zoom in, zoom out : multiplies or divides by 1.414
– Show axes, remove axes
– English or Francais : choose another language

4.7 Other keys

The arrow keys allows to change the visualisation window. The +/- keys are shortcuts for zoom in/out (use alpha and shifts before to modify the zoom factor). The shortcuts list may be displayed by typing F1 (File) then Help.

5 Examples

5.1 A locus

Let C be a circle, A a fixed point on the circle, M a moving point on C. Let $MAPN$ be a direct square of vertices M, A. What is the locus of N when M moves on C ?

– Check that you are in exact mode. Type N to make a new figure and give it a name, e.g. `LIEU`
– We begin by drawing the circle of center the origin and radius 1 : F2 then Circles then cercle (shortcut `S` then cercle), give a name `:C:C`.
– Put A at $(1, 0)$: F2 Points point (shortcut `1`) then `1,0` then the name `:A:A`.
– Now make a moving point M on the cercle : F2 Points element (shortcut `P` element), then `C,1`, then the pointname `:M:M`.
– Make the square : F2 Polygone carre, give the first M and second pointname A, then the square name e.g. `::Q` (we don't need to display the square name), then the names of points 3 and 4 `:P:P` and `:N:N`.
– Eventually, we construct the locus of N : F2 Circles and curves (shortcut `S`) then lieu, then the pointname on the circle `M` then the depending pointname `N`, and the locus name `:L:L`.

It seems the locus is a circle arc, what is it really ? Go back to history (F1 Quit), then type
`eqxpnt(L) STO> l`
which gives the cartesian equation of the locus. It's a circle equation. If you want to get the center of the circle, type
`centre(eqxpnt(l))`
(replace centre by `rayon` for the radius).

Note that the locus is the whole circle, not a circle arc. The system shows only a circle arc because of the rational parametrization of the circle C with limit values for the parameter. Therefore a small part of the circle locus will never be displayed. You can reduce the ârt that is not drawn by changing the Indep-Low and Indep-High before drawing the locus (shortcut `T`).

5.2 An enveloppe locus

Given a point F and a circle C, M a moving point on C, N the mediatrix of F and M. What is the enveloppe of the mediatrices N when M moves on C ?

- Check that you are in exact mode. Type N to make a new figure and give it a name, e.g.`ENVEL`.
- Draw a circle C with the origin as center and radius 2 :
  F2 Circles, curves (shortcut `S`), select cercle, then type `0,2`, then `:C:C`.
- Take for example 1 and 1 for coordinates of F :
  F2 Points point (shortcut `1`) then `1,1` then `:F:F`.
- Now M,
  type `P` select element, type `C,1` then `:M:M`.
- Make the mediatrix
  type F2 lines (shortcut `L`) then `F,M` then `:N:N`.
- Construct the enveloppe
  F2 Curves (shortcut `S`) then lieu, type `M` then `N`, give a name to the locus e.g. `:L:L`.

It takes around 30 seconds to the HP49 to compute the enveloppe. Type + to zoom in and the arrow keys to see the enveloppe better. It looks like it is an ellipse, is it really ? Go back to the hitory (F1 Quit), let's compute the cartesian equation of the locus
`eqxpnt(L)`
After around 10 seconds, you get a 2nd order equation in X, Y, the enveloppe is therefore a conic. Moreover the quadratic part of the equation is definite positive, it is therefore an ellipsis. Computing the axis of the ellipsis may be done by diagonalization of the matrix of the quadratic part (the current version of the system has no built-in instruction to do it, use `AXQ` adn `EGV`).