TI-NSPIRE CAS – MAKING MICROWORLDS

Thomas P. Dick
Mathematics Department
Oregon State University
Corvallis, OR 97331
tpdick@math.oregonstate.edu

Lewis Lum
Mathematics Department
University of Portland
Portland, OR
lum@up.edu

Question: What makes TI-Nspire CAS different in terms of look and feel?

Answer: The interface – navigation and "top level" commands are more like a computer than a graphing calculator.

On most graphing calculators, the navigation is entirely from screen to screen (example: Y= to GRAPH) or from line to line (between entry lines on the calculator screen or between menu items listed line by line). Cursor uses are limited: text cursors for insertion and deletion within a line; crosshairs for tracing a graph. For such navigation, four directional arrows (two for up/down vertical moves, two for left/right moves) are sufficient.

On TI-Nspire there is a "NavPad" that serves much like a computer mouse, with true multi-directional movement, multiple function cursor (selection pointer, marquee selection, grabbing, etc.), and "click" button that adopts most of the conventions computer users have grown accustomed to. In additon, the same top level types of commands that are common across software applications on a computer (such as Control C for copy, Control V for paste, etc.) are fully recognized. Indeed, an important first step for an experienced user of a TI-84 or TI-89 to make in transitioning to the TI-Nspire is to make a deliberate and conscious shift to a mental analogy with a computer's interface.

1) The document model – work is organized and saved into named documents (.tns files when transferred to a computer).

2) Dynamic linking - virtually any two mathematical objects can be linked (even across representational environments such as the graphic environment and the spreadsheet) so that changes in one result in immediately apparent results in another.

Items 2 and 3 above combined make TI-Nspire CAS into a "microworld maker." One can create documents that provide a constrained environment endowed with a dynamic link for exploration.

Question: What makes TI-Nspire CAS different in the calculator environment?

Answer: Entries can be created using a 2-dimensional editing palette very similar to the Equation Editor in Word.

Question: What makes TI-Nspire CAS different in the spreadsheet environment?

Answer: CAS! Imagine having spreadsheet cells inhabited by symbolic expressions whose parameters can be cell or variable references. Illustrated below is an example of what is possible in such an environment.

Column	A	B	C	D
Definition	=seq(k,k,1,5)	= x^(a#)	= d/dx(b#,x)	=d/dx(exp(2x),x,a#)
	1	x	1	2exp(2x)
	2	x^2	2x	4exp(2x)
	3	x^3	3x^2	8exp(2x)
	4	x^4	4x^3	16exp(2x)
	5	x^5	5x^4	32exp(2x)

Question: What makes TI-Nspire CAS different in the graphics environment?

Answer: Direct graphical manipulation. If one graphs a linear function $y = mx + b$ in the usual Graphs & Geometry environment on TI-Nspire one has a two-way action-consequence environment: one can rotate the graph of the line (driver is the graph) and see a resulting change in the numerical value of the slope in the equation for the line. Similarly, one can translate the graph and see a resulting change in the y-intercept of the equation. There are several families of functions that are set up for dynamic updating of their equations via direct graphical manipulation.

While it is true that there are applets that accomplish such graphical transformations, we should note that any function that is dependent on a graphically manipulable function is also dynamically updated.

Question: What makes TI-Nspire CAS different in the data analysis environment?

Answer: Far more graphical tools are available than on any other handheld. Many of the capabilities are found only in Fathom or Tinkerplots.

Question: What makes TI-Nspire CAS different in the geometry environment?

Answer: Lockable attributes. One can constrain the value of a measureable attribute by locking it. For example, one can lock the value of an angle measure in a triangle, or lock the perimeter of a rectangle. Here is a problem to consider: Suppose you lock the area of a triangle and try to move one of its vertices. What locus of points are you constrained to? (Answer: The vertex you move is constrained to two parallel lines, each parallel and equidistant from the side opposite the vertex.) Now suppose you lock the perimter of the triangle and unlock the area. What happens now when you try to move one of the vertices? (Answer: The vertex you move is constrained to an ellipse having the other two vertices as foci.)

Question: What makes TI-Nspire CAS profoundly different?

Answer: Dynamic linking! We supply a couple of general purpose microworld template examples that illustrate.

Example 1. Locus hunt
Scenario: A graphic coordinate page is already set up with a movable point P whose coordinates are displayed. If <control> < . > (period) is pressed, the current coordinates of point P are recorded in two cells of a spreadsheet and the point is "marked" on screen in the graphics page.
Example of use: Now suppose one graphs the line $y = -1$ and makes a fixed point F at (0,1) and measures the distance of point P to the line and to the point F, displaying both distances. As one moves the point P around, one can hunt for and mark positions where the two distances are the same. This makes a natural and dynamic setting for motivating the focus-direrectrix definition of a parabola.

Example 2. Slider bar
Scenario: A custom split screen page is already set up with a small horizontal window endowed with a customizable slider bar (a line segment portion of the number line whose endpoint coordinates can be specified by the user). A movable point on the segment has a dynamic coordinate displayed which can be easily linked to any parameter.
Example of use: Set the endpoint values of the slider from 0 to 3. Link the slider point coordinate to the parameter b. On the lower part of the split screen open a graphics window and graph $f1(x) = b\text{^}x$. Manipulation of the slider now dynamically updates the graph. Now, graph a second function: $f2(x) = df1(x)/dx$. Manipulation of the slider dynamically changes both graphs! Move the slider to find a value of b that makes the two graphs coincide. You get the idea!