

VTI vs TiEmu
Implicit Differentiation
for the TI-92+

September 15, 2022.

We have an odd problem when emulating a TI-92+ on a PC. Eventually, we realize the Voyage 200 is similar, but capable of running a newer calculator operating system, Advanced Mathematics Software (AMS) 3.10. For the TI-92+ machines, TI stopped developing at AMS 2.09. One fine point of difference between the two is that AMS 2.09 did not include an ability to perform implicit differentiation. So, we learn this, and we decide we wish to switch over to emulating the Voyage 200. Yes?

Well, not necessarily. Emulating the V200 would require abandoning Rusty Wagner's classic emulator, Virtual Texas Instruments (VTI). Rusty's emulator can emulate the TI-92+ with AMS 2.09, but it cannot emulate the Voyage 200. Switching over to using AMS 3.10 requires using TiEmu.

I appreciate TiEmu, but it has a problem: As of this date, it is still impossible to install the TiEmu emulator without triggering virus messages from security programs. Webroot SecureAnywhere aggressively deletes the TiEmu installer, immediately. People insist the stern warning messages are "false positives." Some say we should shut down all protection when running the TiEmu installer.

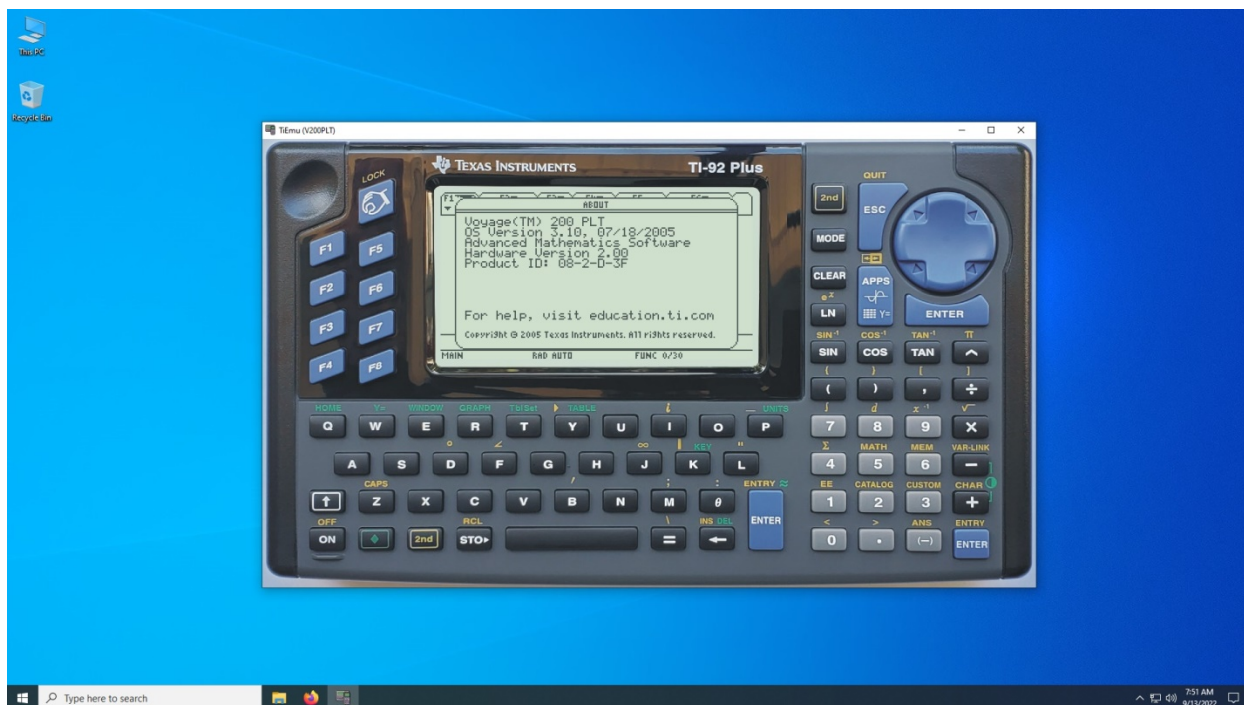
For me, that is simply too high a price to pay for the option of using implicit differentiation. So, what are the other available options? How can we differentiate implicitly, but safely?

One thing to do is to run TiEmu inside a "virtual" Windows PC. Use VMware, or VirtualBox, or any of the others. Create a cute regression. We make an emulator run an emulator, but it works. With TiEmu running safely inside a virtual PC, any malware included in the setup cannot escape, and it cannot affect the "real" PC.

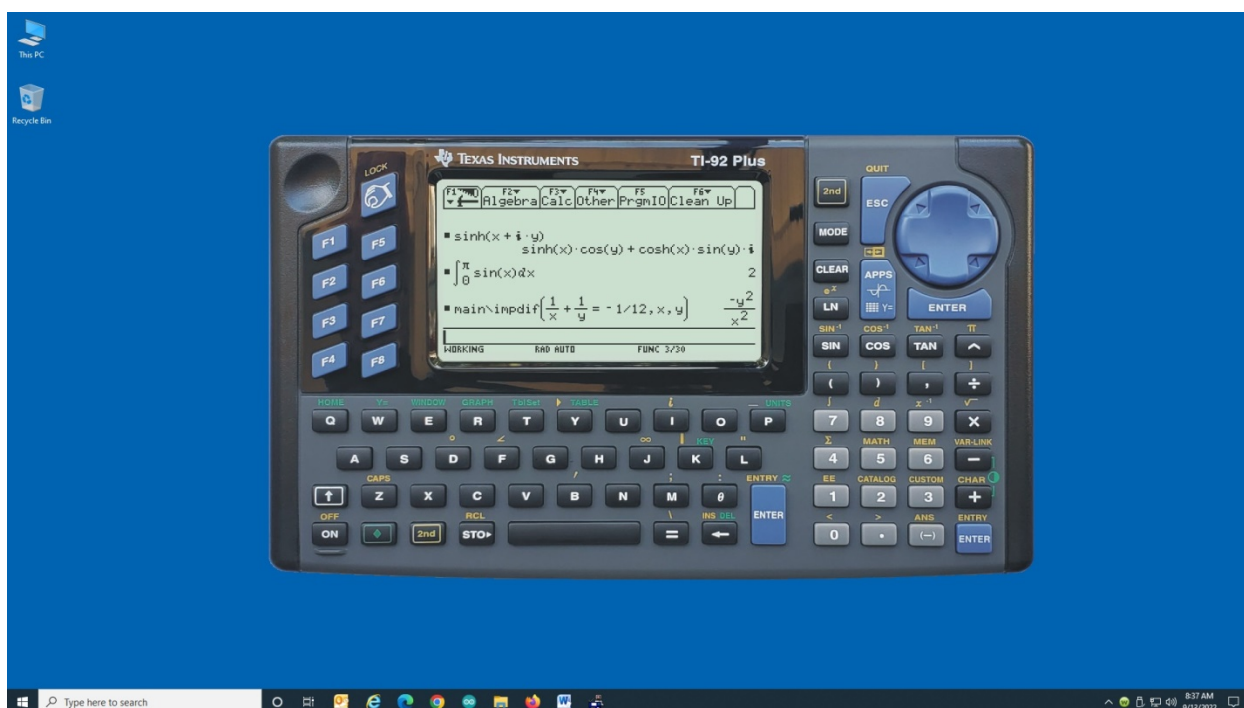
I want that implicit differentiation function to work on my emulation of the TI-92+, but I want it running on my real PC, not inside a virtual PC. I want it to be convenient, and quick to activate. Yet, I will not switch over to running TiEmu directly on my real PC for speedy access and convenience, not until after the TiEmu programmers bring in a team of exorcists, and apply some holy water.

In a moment, I will show how to add an implicit differentiation feature to AMS 2.09, which neatly avoids the problem. When installed, it works exactly like the implicit differentiation function in AMS 3.10, but it runs on a TI-92+ which can be perfectly emulated by Rusty Wagner's VTI, all without triggering any virus warning alarm sirens. Before we look at how to make that happen, first I would like you to see something beautiful that VTI can do, something TiEmu cannot do (as of yet).

On the next page, the top picture shows TiEmu emulating a Voyage 200. It looks exactly like a TI-92+ because I prefer the visual appearance of a TI-92+. I made my V200 emulation "look" like a TI-92+, but it is actually running V200 firmware using AMS 3.10. In the picture, TiEmu is operating safely, because it is trapped inside an isolated Windows virtual machine. Any malware present cannot escape while making the picture. Look closely at the picture. What do you see?



In the second picture, VTI is emulating a TI-92+ running AMS 2.09. The calculator is using AMS 2.09, and performing an implicit differentiation. This is done by using a user-defined “impdif” function. Note that the calculator image is “borderless.” Borderless emulation is a feature not yet supported by TiEmu.



So, how did this happen? In the ticalc archives, I noticed several people tried making similar implicit differentiation programs for AMS 2.09, long ago. I tried each of them, carefully. Ultimately, I decided to build on what I learned, and make one that works exactly like the “impdif” function in AMS 3.10.

The heart of the problem is that TI published an incomplete solution here:

<https://education.ti.com/en/customer-support/knowledge-base/other-graphing/product-usage/12059>

The knowledge-base article in the above link shows “impdif” defined this way:

```
Define impdif(uu,xx,yy)=-d(uu,xx)/d(uu,yy)
```

If entering this definition, first use the MODE button and set the current working directory to MAIN. The method shown in this knowledge-base article only works with expressions that are set equal to zero. With a bit of care, a better “Define” statement can be created. It’s just slightly longer to enter:

```
Define impdif(uu,xx,yy)=-d(right(uu)-left(uu),xx)/(d(right(uu)-left(uu),yy))
```

On my AMS 2.09 equipped TI-92+ (both real and emulated) I created a “UTILITY” folder in the “MAIN” folder. In that UTILITY folder, I created a small program named "impdifdp" which translates into "implicit differentiation definition program." It is an installer. It creates an “impdif” function which is stored in a function variable in the “main” folder. The program looks like this:

```
:impdifdp()  
:Prgm  
:setFold(main)  
:Define impdif(uu,xx,yy)=-d(right(uu)-left(uu),xx)/(d(right(uu)-  
left(uu),yy))  
:EndPrgm
```

The TI-92+ keyboard has a 2nd function “**d**” placed above the "8" key for normal explicit differentiation. When typing the above program into the program editor, in each of the two places where you see the characters “**d**(“ which mean “derivative” you should NOT type a letter “d” followed by a “(” character. Instead, type the 2nd function key, followed by the “8” key. This shifted keystroke generates a special derivative symbol, along with an opening parenthesis in the program code.

Instead of typing the program in directly, it is much more convenient to simply run VTI with a TI-92+ emulation, and then drag and drop the “IMPDIFDP.9XP” file onto the screen of the emulated calculator. If it does not already exist, VTI will automatically create the UTILITY folder, and place the program inside. Run that program once on your TI-92+ to install the “impdif” function definition. On your real calculator, you should archive the “impdifdp” program variable. That way, if you lose battery power you won’t lose the ability to activate the function. Archived files survive power outages.

For convenience, I also created a separate hotkey program named "kbdprgm8" in the "MAIN" folder. Pressing the [green diamond] key then "8" runs the program. It prompts for an implicit function, and then returns the derivative. The hotkey is easy to remember, because it is located on the same key that is normally used for explicit derivatives.

Another program, "MAIN\kbdprgm3" is included in this package. It defines a custom menu with "impdif" added to the Math menu associated with function key F2. This [green diamond] then "3" hotkey combination is also easy to remember, because it is located on the button with a "CUSTOM" label. The [2nd] function of that button is ordinarily used for toggling the custom menu on and off.

Other items on the custom menu allow for conveniently entering symbols used for common unit conversions, and quick access to some of my most frequently used math functions.

So, now the good old TI-92+ (both real and virtual) includes implicit differentiation, and the emulated version on the PC has a nice borderless appearance. It looks exactly like using a perfectly new calculator, freshly unboxed.

Side note: If you would like to use the TI-92+ skin shown in this document, it is maintained here:

<https://www.ticalc.org/archives/files/fileinfo/478/47807.html>

Have fun with all of it,

Bill Triplett
20220915

Copyright © September 13, 2022 by Bill Triplett

Any person may distribute this file freely, but may not modify or separate parts of the content or remove the copyright notice. No person or business entity may sell the content in whole or in part without written permission from the author.