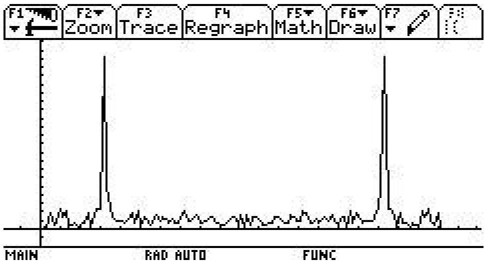
FFT v1.2 (Flash app)

Fast Fourier Transform for the TI-89/92+

Author: Bhuvanesh Bhatt (bbhatt1@towson.edu)

Last updated: October 7, 2001



Description:

This program computes the radix-2 FFT (or inverse FFT) of a list or matrix, using the signal-processing convention for the FFT by default:

FFT = $\sum_{j=1}^{n} \frac{u_j}{e^{-2p\cdot \boldsymbol{i}\cdot(j-1)\cdot(k-1)/n}}$ Inverse FFT = $\frac{1}{n}\cdot\sum_{k=1}^{n} v_k \cdot e^{2p\cdot \boldsymbol{i}\cdot(j-1)\cdot(k-1)/n}$

The user can also specify two additional conventions.

This Flash application is freeware – it can be sent to and run on any TI-89 or TI-92 Plus calculator. It was my submission for TI's Flash app contest for 2001.

Syntax:

FFT**.**FFT(list[,direction[,convention]])

If there is no variable called FFT in the current folder, you can simply use FFT(...)

Evaluating FFT() shows a help message in the status line. The Catalog FlashApps screen also shows a help message.

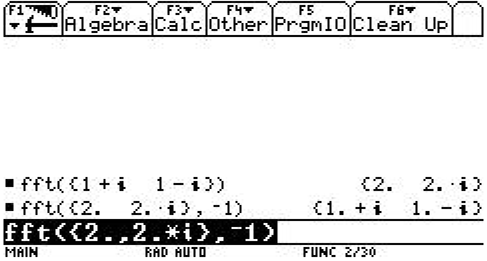
Notes:

- *list* can have complex numbers, but it must be numeric, and it also cannot have numbers like ***e***. It can be a list or matrix.
- The length of *list* (or if it is a matrix, both dimensions) must be a power of 2.
- *Direction* positive/negative: Forward/Inverse FFT
- *Convention* can be one of the following:
  - "s" or "signal" (signal processing, default)
  - "p" or "physics" (physical sciences – forward and inverse transforms are exchanged and multiplied by sqrt(dim(*list*))^(-sign(*direction*))
  - "d" or "data" (data analysis – forward and inverse transforms are exchanged)
- To abort a calculation, press [ON]

Examples:

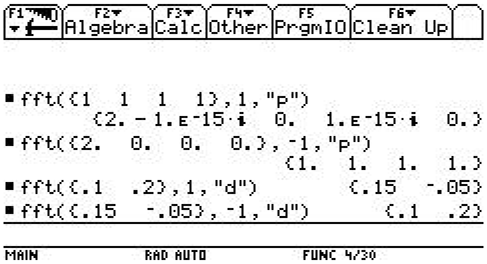
FFT({1+***i***,1–***i***}) returns the forward FFT {2, 2·***i***}
FFT({2, 2·***i***},-1) returns the inverse FFT {1+***i***,1–***i***}

Here is a screenshot of the above example:



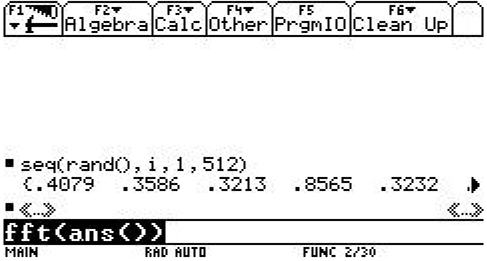
```
■ fft({1 + i   1 - i})              {2.   2.·i}
■ fft({2.   2.·i}, -1)        {1. + i   1. - i}
fft({2.,2.*i},-1)
MAIN          RAD AUTO          FUNC 2/30
```

This shows how to use the other conventions:



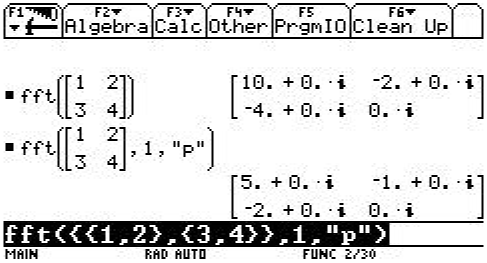
```
■ fft({1   1   1   1},1,"p")
      {2. - 1.E-15·i   0.   1.E-15·i   0.}
■ fft({2.   0.   0.   0.}, -1,"p")
                        {1.   1.   1.   1.}
■ fft({.1   .2},1,"d")           {.15   -.05}
■ fft({.15   -.05}, -1,"d")         {.1   .2}

MAIN          RAD AUTO          FUNC 4/30
```

This shows the FFT of a somewhat large list. The result is too large and is therefore not displayed by the AMS:



```
■ seq(rand(), i, 1, 512)
 {.4079   .3586   .3213   .8565   .3232   ▸
■ «...»                                  «...»
fft(ans())
MAIN          RAD AUTO          FUNC 2/30
```

This shows the FFT of a matrix, using two conventions:



```
      ⎛⎡1  2⎤⎞      ⎡10. + 0.·i   -2. + 0.·i⎤
■ fft ⎝⎣3  4⎦⎠      ⎣-4. + 0.·i   0.·i      ⎦
      ⎛⎡1  2⎤        ⎞
■ fft ⎝⎣3  4⎦, 1, "p"⎠
                    ⎡5. + 0.·i    -1. + 0.·i⎤
                    ⎣-2. + 0.·i   0.·i      ⎦
fft({{1,2},{3,4}},1,"p")
MAIN          RAD AUTO          FUNC 2/30
```

The title screenshot shows the amplitude plot with two symmetric peaks of a slightly noised periodic signal.

Thanks to:

- E.W.
- Mike Grass
- Texas Instruments, especially Heidi and S.H.
- The TIGCC team

What's New:

- The user can choose two additional common conventions for the FFT
- The *direction* argument is now optional (if you want to do a default forward FFT)
- FFT of a matrix is now supported (but it probably needs some optimization)
- Fixed a memory leak that used to occur when the calculation was aborted

Future plans:

- Convolution and correlation
- Allow ***e*** and **π** in input list/matrix
- Remove restriction on length/dimensions of list/matrix
- This program is *very* fast compared to a TI-Basic implementation I tried, especially for large lists. For example, the FFT of a 512-element array of floating-point random numbers took about 24 seconds (excluding result display time) on a HW2 TI-92+ with AMS v2.05. Still, I will try to speed it up as much as I can.

For updates and other programs:

http://triton.towson.edu/users/bbhatt1/ti/

Disclaimer:

In no event shall I (Bhuvanesh Bhatt) or Texas Instruments (TI) be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the use of the software and/or documentation. Moreover, I shall not be liable for any claim of any kind whatsoever against their use by any other party. Anyone can distribute this program provided it remains free and is not modified.