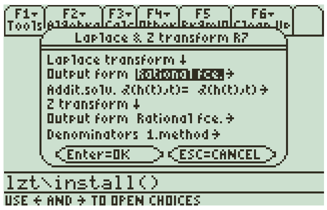
# 1. Installation

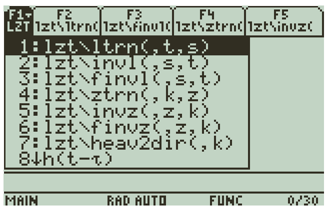
- Unpack the file *lzt.zip*
- Send the file *lztR7.89g* to your calculator by graph-link cable and software. In the calc a folder *lzt* will be created.
- Unpack the file *kerno.zip*, which be found in a directory *distributed_files*, and send *KerNO.89z to calc*. If your calculator is HW2 or HW3 version, send also *hw3patch.89z*. The HW version determine pressing F1 and A.


- Install KERNO – On calculator type *kerno()* and press EXE. If HW2 or HW3 version, install hw3patch before – type *hw3patch()*.
- Install LZT – Type *lzt\install()*.



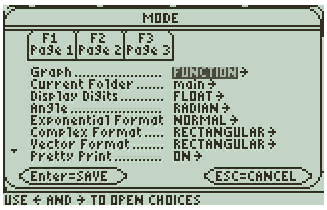

Choose whether an answer expresion are to be with a common denominator at all cost (*Rational fce.*) or as partial fractions (Partial frac.)

The Installation creates a custom menu. You can switch between custom and main menu by pressing 2nd and CUSTOM.



Some items of a mode options are change within the installation.

ANGLE is set to RADIAN
COMPLEX FORMAT is set to RECTANGULAR
EXACT/APPROX is set to AUTO

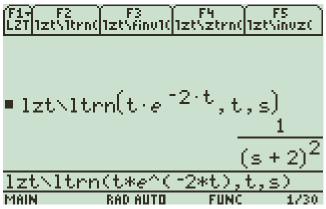


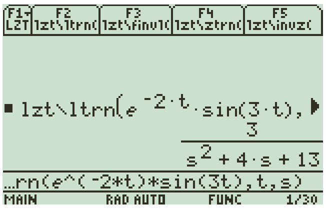

Keep these options set when you are using LZT.

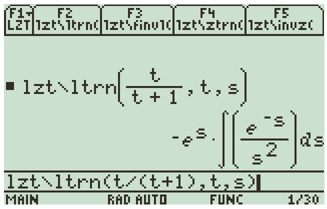
# 2. Direct Laplace transformation

***lzt \ ltrn ( f ( t ) , t , s )***

| | |
|---|---|
| ***f ( t )*** | a time dependent function you want to transform |
| ***t*** | an independent continuous variable |
| ***s*** | an independent complex variable of Laplace transform |





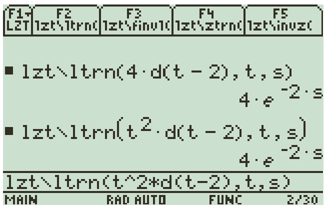


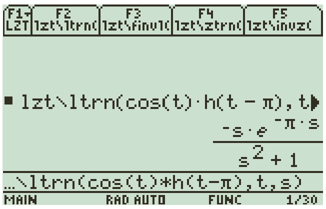

## Unit step, Dirac pulse

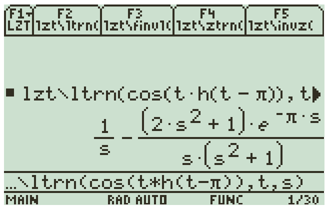
| | |
|---|---|
| $\boldsymbol{h ( t – t_0 )}$ | a unit step, $h ( t – t_0 ) = 0$ *when* $t < t_0$ *, else* $= 1$ |
| $\boldsymbol{d ( t – t_0 )}$ | a dirac pulse in time $t_0$ |

possibly

***h ( p(t) ), d ( p(t) )*** where *p(t)* is a multinomial function with real roots





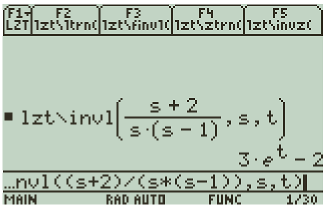


Jiri Bazant
jiri.bazant@vslib.cz
georger@razdva.cz

# 3. Inverse Laplace transformation

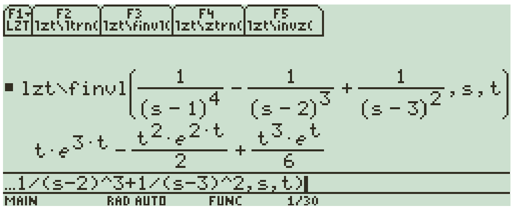
***lzt \ invl ( F ( s ) , s , t )***

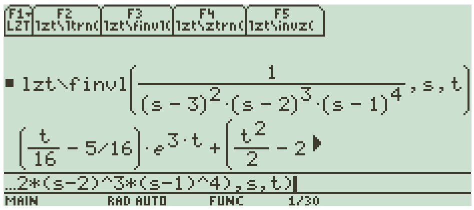
| | |
|---|---|
| ***F ( s )*** | Laplace transform of *f ( t )* |
| ***s*** | the independent complex variable of Laplace transform |
| ***t*** | the independent continuous variable |




## Inverse transformation of expressions with several multiple roots or fractions

***lzt \ finvl ( F ( s ) , s , t )***



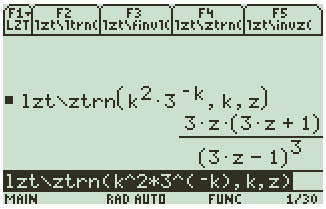

In such a case will *finvl* evaluate a result many times faster than *invl*. When the evaluation will run a long time then break it (press ON) and try the other.

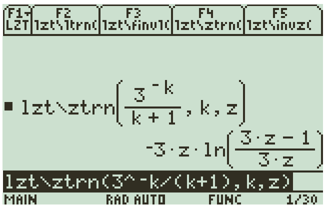

Jiri Bazant
jiri.bazant@vslib.cz
georger@razdva.cz

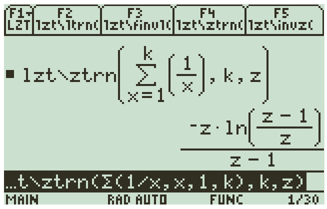
# *4. Direct Z transformation*

***lzt \ ztrn ( f ( k ) , k , z )***

| | |
|---|---|
| ***f ( k )*** | a discrete function you want to transform |
| ***k*** | an independent discrete variable |
| ***z*** | an independent complex variable of Z transform |



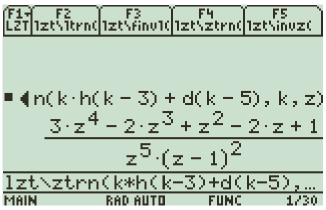




## *Unit step, Dirac pulse*

***h ( k – $k_0$ )*** the unit step, $h(k - k_0) = 0$ *when* $k < k_0$ *, else* $= 1$
***d ( k – $k_0$ )*** the dirac pulse, $d(k - k_0) = 1$ *when* $k = k_0$ *, else* $= 0$

possibly

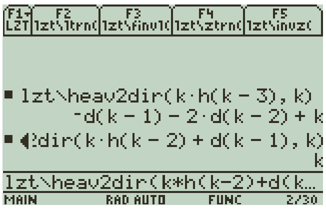
***h ( p(k) ), d ( p(k) )*** where *p(k)* is a multinomial function with real roots



## *Conversion of the Unit steps to the discrete Dirac pulses*

***lzt \ heav2dir ( f ( k ) , k )***

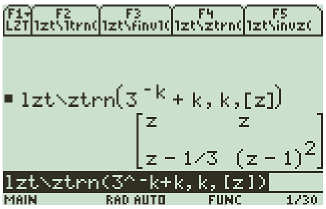
***f ( k )*** a discrete function containing one or more different unit steps




This function may simplify an answer of ***invz*** tool, when it contains more than one unit step or the unit step and dirac pulses at once.

## *An Answer with separated numerators and denominators of partial fractions*

***lzt \ ztrn ( f ( k ) , k , [z] )***

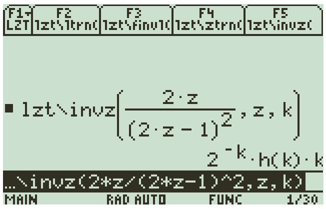


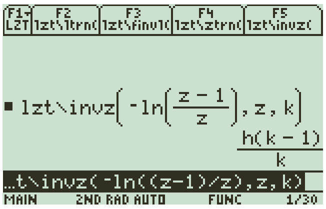

A result is represented by matrix. First row contains numerators and second row contains denominators. Each column represents a partial fraction and whole result is given by a sum of all partial fractions. This is a way how to prevent TI89 software from an attempt to simplify the result expression in each step which may be too slow.

# *5. Inverse Z transformation*

***lzt \ invz ( F ( z ) , z , k )***

| | |
|---|---|
| ***F ( z )*** | Z transform of *f ( k )* |
| ***z*** | the independent complex variable of Z transform |
| ***k*** | the independent discrete variable |






***finvz*** works just as ***finvl***. See the section 3.

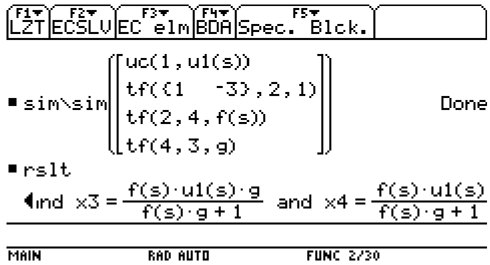
# *6. FAQ*

**Q: What to do when an error “Invalid program reference” occurred**

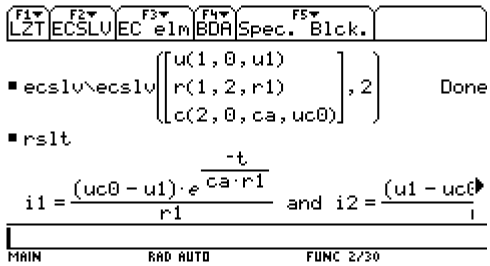
**A:** TI-89 operation system v2.0 or higher contains a restriction that disallow assembly programs return an answer. LZT use the assembly programs. There are patches and kernels that break this and others restrictions. In a folder “\distributed files” are two. First is KERNO by Greg Dietsche and second is PREOS by Patrick Pelissier. You must install one to LZT may work. See the documentation included. There are always new versions of the operation system and now is here a new version of TI-89 named Titanium. May become that LZT with these versions of kernels will not work. In such a case look for new version or other patch that break the restriction of the operation system. Here are some of places where to look.

www.ticalc.org
http://www.ticalc.org/pub/89/asm/shells/date.html
http://calc.gregd.org/
http://www.tigen.org/kevin.kofler/ti89prog/

# *7. Shells for Laplace and Z transform*




If you deal with automation or simulation of dynamic systems then may be useful for you a program for calculation block diagram algebra**.** In a zip-archive “\LZT_shell\bda.zip” is the program named **Block diagram algebra** simulates like MATLAB Simulink but symbolically. It doesn’t have a GUI, the diagram description is entered by a command line.




**Electric circuit solver** is a program for symbolic simulation of electric circuits, specially suited for dynamic process in RLC circuits. The circuit description is entered by the command line. Install it from “\LZT_shell\ecslv.zip”.

**You should know that this programs needs Laplace and Z transformation to be installed otherwise they can’t fully works.**