TAMUDFEQ 1.2 Command Reference

Department of Mathematics, Texas A&M University

Monday, 06 July 2009

1. See *http://calclab.math.tamu.edu/~belmonte/TAMUDFEQ/TAMUDFEQ.html* for instructions on downloading and installing the *TAMUDFEQ* package, refreshing libraries, and using preactivated templates.
2. When starting a new problem, run the **activate** command under *atm_tamudfeq12* in the libraries portion of the catalog.
3. It is not necessary to reactivate the package on subsequent *pages* of a given problem—only if you start a *new* problem.
4. Rather than manual [re]activation, you will find preactivated templates much easier and faster to use!

COL

Description	The **col** function extracts a column from a matrix.
Access	**e.col** library shortcut in the *eigen* group
Syntax	**e.col(*M,k*)**
Input	***M***: a matrix ***k***: a positive integer column index
Output	the ***k***-th column of the matrix ***M***
Example	**e.col**$\left(\begin{bmatrix} -9 & 3 \\ 5 & -5 \end{bmatrix}, 2\right)$ returns $\begin{bmatrix} 3 \\ -5 \end{bmatrix}$.
See Also	**EIGENVECTS**, **TES**

CZPC

Description	The **czpc** function computes the zeros of a polynomial along with their respective multiplicities.
Access	**l.czpc** library shortcut in the *linear* group
Syntax	**l.czpc(*c*)**
Input	***c***: the coefficients of a polynomial in descending order (including zeros) expressed as a row vector
Output	a table (matrix) with zeros of the polynomial in the first column and respective multiplicities in the second column
Example	**l.czpc**$([1,\ -1,\ 2,\ -2,\ 1,\ -1])$returns $\begin{bmatrix} \text{"Zero"} & \text{"Mult"} \\ i & 2 \\ -i & 2 \\ 1 & 1 \end{bmatrix}$ where $i = \sqrt{-1}$.
See Also	**WRON**
Note	The acronym **czpc** stands for (**c**omplex) **z**eros from a **p**olynomial's **c**oefficients.

EIGENVECTS

Description The **eigenvects** function symbolically computes the eigenvector(s) associated with an eigenvalue of a matrix.

Access **e.eigenvects** library shortcut in the *eigen* group

Syntax **e.eigenvects(*A*,*r*)**

Input ***A***: square matrix with constant elements
r: an eigenvalue of ***A***

Output a basis for the eigenspace associated with the given eigenvalue as a matrix of column eigenvectors

Example Let $\boldsymbol{A} = \begin{bmatrix} 6 & -6 & -6 & -8 \\ 8 & -8 & -6 & -8 \\ -1 & 7 & -10 & -9 \\ -8 & 6 & 6 & 6 \end{bmatrix}$. The eigenvalues of ***A*** are $-2, -2, -1 \pm 3i$. (See the example for **evmt**.) Then

e.eigenvects(*A*, -2**)** returns $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -8 & 11 \\ 7 & -9 \end{bmatrix}$, the columns of which are eigenvectors of ***A*** associated with the eigenvalue $r = -2$. These eigenvectors form a basis of the eigenspace associated with eigenvalue $r = -2$.

Analogously, **e.eigenvects(*A*,** $-1+3i$**)** returns $\begin{bmatrix} 1 \\ 1 \\ \frac{3}{2} - \frac{1}{2}i \\ -1 \end{bmatrix}$.

See Also **EVMT**

Note This command is part of the *linalgcas* library from Texas Instruments.

EVMT

Description The **evmt** function returns a table of the eigenvalues of a square matrix and their respective algebraic multiplicities.

Access **e.evmt** library shortcut in the *eigen* group

Syntax **e.evmt(*A*)**

Input ***A***: square matrix with constant elements

Output a table (matrix) with the eigenvalues of the matrix in the first column and their respective algebraic multiplicities in the second column

Example Let $\boldsymbol{A} = \begin{bmatrix} 6 & -6 & -6 & -8 \\ 8 & -8 & -6 & -8 \\ -1 & 7 & -10 & -9 \\ -8 & 6 & 6 & 6 \end{bmatrix}$. Then **e.evmt(*A*)** returns $\begin{bmatrix} \text{"Eigvl"} & \text{"Mult"} \\ -1+3i & 1 \\ -1-3i & 1 \\ -2 & 2 \end{bmatrix}$. The eigenvalues of ***A*** are $-2, -2, -1 \pm 3i$.

See Also **EIGENVECTS**

Note The acronym **evmt** stands for **e**igen**v**alue **m**ultiplicity **t**able.

EXPMAT

Description	The **expmat** function symbolically computes the matrix exponential function of a square matrix.
Access	**e.expmat** library shortcut in the *eigen* group
Syntax	**e.expmat(*A*)**
Input	***A***: square matrix with constant coeffcients
Output	the matrix exponential function $e^{t\mathbf{A}}$
Example	**e.expmat(**$\begin{bmatrix} -1 & 1 \\ -4 & -5 \end{bmatrix}$**)** returns $\begin{bmatrix} (2t+1)\,e^{-3t} & t\,e^{-3t} \\ -4te^{-3t} & (1-2t)\,e^{-3t} \end{bmatrix}$.
See Also	**EIGENVECTS**, **EVMT**, **KERNELBASIS**
Note	This command is part of the *linalgcas* library from Texas Instruments.

GRAIL

Description	The **grail** function returns the solution of a linear differential system via the matrix Laplace transform method.
Access	**l.grail** library shortcut in the *linear* group
Syntax	**l.grail(*A, f,* t_0**, $\boldsymbol{x}_0$**)**
Input	***A***: square matrix with constant coefficients ***f***: the [non]homogeneous part of the system; a column vector t_0: initial time value; a scalar $\boldsymbol{x}_0$: initial condition; a column vector
Output	the unique solution $\mathbf{x}$ to the initial value problem $\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{f}$, $\mathbf{x}(t_0) = \mathbf{x}_0$
Example	Let ***A***= $\begin{bmatrix} 0 & 2 \\ -1 & 3 \end{bmatrix}$, ***f***= $\begin{bmatrix} e^t \\ -e^t \end{bmatrix}$, $t_0 = 0$, and $\boldsymbol{x}_0$= $\begin{bmatrix} 5 \\ 4 \end{bmatrix}$. Then **l.grail(*A, f,* t_0, $\boldsymbol{x}_0$)** yields $\begin{bmatrix} (4t+5)\,e^t \\ (2t+4)\,e^t \end{bmatrix}$, the unique solution to $\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{f}$, $\mathbf{x}(t_0) = \mathbf{x}_0$.
See Also	**EIGENVECTS**, **EVMT**, **KERNELBASIS** (for a semiautomatic alternative method of solution)

ILAP

Description	The **ilap** function computes the inverse Laplace transform of a scalar, vector, or matrix expression.
Access	**s.ilap** library shortcut in the *Laplace* group
Syntax	**s.ilap(*F*)**
Input	***F***: a scalar, vector, or matrix expression in s
Output	$\mathscr{L}^{-1}\{\boldsymbol{F}\}$, the inverse Laplace transform of ***F***
Examples	**s.ilap**$\left(\dfrac{2s}{(s^2+1)^2} \right)$ returns $t \sin t$. **s.ilap**$\left(\begin{bmatrix} \dfrac{5s-1}{(s-1)^2} \\ \dfrac{4s-2}{(s-1)^2} \end{bmatrix} \right)$ returns $\begin{bmatrix} (4t+5)\,e^t \\ (2t+4)\,e^t \end{bmatrix}$.
See Also	**LAP**, **LTLDO**, **SYSLAP**
Note	This command is a generalization of the **ilaplace** command in a port to the TI-Nspire CAS of a special functions package written by Lars Frederiksen of Denmark; **ilap** calls **ilaplace**.

KERNELBASIS

Description The **kernelbasis** function returns a basis for the nullspace of a matrix.

Access **e.kernelbasis** library shortcut in the *eigen* group

Syntax **e.kernelbasis(**$\boldsymbol{M}^p$**)**

Input $\boldsymbol{M}^p$: Typically, the input is a matrix $\boldsymbol{M}=\boldsymbol{A}-r\boldsymbol{I}$ raised to a positive integer power p, where r is an eigenvalue of $\boldsymbol{A}$, a square matrix with constant elements, and $\boldsymbol{I}$ is the identity matrix. Moreover, the geometric multiplicity of r is less than its algebraic multiplicity.

Output a basis for the nullspace of $\boldsymbol{M}^p$ as a collection of [generalized] eigenvectors associated with r as referenced above

Example Let $\boldsymbol{A}=\begin{bmatrix} 42 & 90 & -40 & 0 \\ -24 & -43 & 20 & -4 \\ -9 & 0 & 2 & -9 \\ -40 & -90 & 40 & 2 \end{bmatrix}$. The eigenvalues of $\boldsymbol{A}$ are $r=-3,2,2,2$. The algebraic multiplicity of $r=2$ is thus 3, whereas its geometric multiplicity is 2 since **e.eigenvects(*A*,**2**)** yields $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & \frac{9}{4} \\ 1 & 0 \end{bmatrix}$, the columns of which are two linearly independent eigenvectors associated with $r=2$. Now for $r=2$, form $\boldsymbol{M}=\boldsymbol{A}-r\boldsymbol{I}=\begin{bmatrix} 40 & 90 & -40 & 0 \\ -24 & -45 & 20 & -4 \\ -9 & 0 & 0 & -9 \\ -40 & -90 & 40 & 0 \end{bmatrix}$. Then **e.kernelbasis(**$\boldsymbol{M}^2$**)** returns $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & \frac{9}{4} & 0 \\ 1 & 0 & 1 \end{bmatrix}$. Hence $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ is a generalized eigenvector for $r=2$.

See Also **EIGENVECTS**

Note This command is part of the *linalgcas* library from Texas Instruments.

LAP

Description The **lap** function computes the Laplace transform of a scalar, vector, or matrix expression.

Access **s.lap** library shortcut in the *Laplace* group

Syntax **s.lap(*f*)**

Input ***f***: a scalar, vector, or matrix expression in t

Output $\mathscr{L}\{\boldsymbol{f}\}$, the Laplace transform of ***f***

Example **s.lap**$(t\cdot\sin(t))$ returns $\frac{2s}{(s^2+1)^2}$.

s.lap$\left(\begin{bmatrix} (4t+5)e^t \\ (2t+4)e^t \end{bmatrix}\right)$ returns $\begin{bmatrix} \frac{5s-1}{(s-1)^2} \\ \frac{4s-2}{(s-1)^2} \end{bmatrix}$.

See Also **ILAP**, **LTLDO**, **SYSLAP**

Note This command is a generalization of the **laplace** command in a port to the TI-Nspire CAS of a special functions package written by Lars Frederiksen of Denmark; **lap** calls **laplace**.

LDO

Description	The **ldo** function constructs a linear differential operator from a row vector of its coefficients along with the independent and dependent variables in which it is expressed.
Access	**l.ldo** library shortcut in the *linear* group
Syntax	**l.ldo(*L*,*x*,*y*)**
Input	***L***: a row vector $[a_n(x), a_{n-1}(x), \ldots, a_1(x), a_0(x)]$of the coefficients (constants or functions of the independent variable) of the linear differential operator in descending order by derivative, including zero coefficients ***x***: the independent variable of the linear differential operator ***y***: the dependent variable of the linear differential operator
Output	the linear differential operator $\sum_{k=0}^{n} a_k(x)\, y^{(k)}(x)$, where $y^{(k)}(x) = \frac{d^k y}{dx^k}$ and $y^{(0)}(x) =$y(x)
Example	The command **l.ldo(**$[1,0,1,x]$, x, y**)** returns $\frac{d^3}{dx^3}(y(x)) + \frac{d}{dx}(y(x)) + xy(x)$.
See Also	**LDOEVAL**, **LTLDO**
Note	The acronym **ldo** stands for **l**inear **d**ifferential **o**perator.

LDOEVAL

Description	The **ldoeval** program displays the result obtained when a linear differential operator acts on a given function or collection of functions.
Access	**l.ldoeval** library shortcut in the *linear* group
Syntax	**l.ldoeval(*f*,*L*,*x*,*y*)**
Input	***f***: a scalar expression or a row vector of expressions in the independent variable ***L***: a row vector $[a_n(x), a_{n-1}(x), \ldots, a_1(x), a_0(x)]$ of the coefficients (constants or functions of the independent variable) of the linear differential operator in descending order by derivative, including zero coefficients; more typically, a variable containing said row vector ***x***: the independent variable of the linear differential operator ***y***: the dependent variable of the linear differential operator
Output	The scalar or vector expression $\sum_{k=0}^{n} a_k(x)\, f^{(k)}(x)$, obtained by substituting f for y in the linear differential operator $\sum_{k=0}^{n} a_k(x)\, y^{(k)}(x)$, is displayed.
Example	The command **l.ldo(**$[1,0,1,x]$, x, y**)→L** followed by the command **l.ldoeval(**$\sin(x)$, **L**, x, y**)** displays $x\sin x$. Similarly, the command **l.ldo(**$[1,-1,4,-4]$, x, y**)→L** followed by the command **l.ldoeval(**$[e^x, \cos(2x), \sin(2x)]$, **L**, x, y**)** displays $[0,0,0]$.
See Also	**LDO**, **LTLDO**
Note	The acronym **ldo** stands for **l**inear **d**ifferential **o**perator.

LTLDO

Description	The **ltldo** function returns the Laplace transform of a linear differential operator with constant coefficients and associated initial conditions at $t = 0$.
Access	**s.ltldo** library shortcut in the *Laplace* group
Syntax	**s.ltldo(*d,i*)**
Input	***d***: a row vector of the constant coefficients of a linear differential operator in descending order by derivative, including zero coefficients ***i***: initial conditions at $t = 0$ of an n-th order initial value problem in ascending order by derivative, expressed as a row vector
Output	the Laplace transform of the linear differential operator followed by substitution of the initial conditions
Examples	Given the linear differential operator $y''(t) + 5y'(t) + 6y(t)$ with initial conditions $y(0) = 2$ and $y'(0) = 3$, the command **s.ltldo(**$[1,5,6]$, $[2,3]$**)** yields $\psi s^2 + (5\psi - 2)s + 6\psi - 13$ or $\left(s^2 + 5s + 6\right)\psi - 2s - 13$, which stands for $\left(s^2 + 5s + 6\right)Y(s) - 2s - 13$, the Laplace transform of the linear differential operator followed by substitution of the initial conditions. Here $\psi = Y(s) = \mathscr{L}(y(t))$.
See Also	**ILAP**, **LAP**, **SYSLAP**
Note	The acronym **ltldo** stands for **L**aplace **t**ransform of a **l**inear **d**ifferential **o**perator.

ROOF

Description	The **roof** function returns a second linearly independent solution to a homogeneous linear second-order ordinary differential equation using a **r**eduction **of** **o**rder **f**ormula (hence the acronym).
Access	**l.roof** library shortcut in the *linear* group
Syntax	**l.roof(*f,p,x*)**
Input	***f***: a solution to a given homogeneous linear second-order ordinary differential equation that is in standard linear form (SLF), $y'' + py' + qy = 0$ ***p***: the coefficient of the first derivative of the dependent variable in the SLF of the differential equation ***x***: the independent variable in the differential equation
Output	a second linearly independent solution to the differential equation, given by the reduction of order formula $y = f \int \frac{e^{-\int p\,dx}}{f^2}\,dx$
Examples	Given that $y = e^x$ is a solution of the differential equation $y'' - 2y' + y = 0$, the command **l.roof(**e^x**,**-2**,**x**)** yields xe^x, a second linearly independent solution.
See Also	**CZPC**, **GRAIL**

SYSLAP

Description The **syslap** function computes the solution of a [system of n-th order] linear differential equation(s) with constant coefficients and initial conditions via Laplace transform methods.

Access **s.syslap** library shortcut in the *Laplace* group

Syntax **s.syslap(*P, f,* t_0*, i*)**

Input ***P***: matrix of coefficients of linear differential operator(s)
f: the [non]homogeneous part(s) of the linear differential equation(s)
t_0: initial time value
i: row(s) of initial conditions in ascending order by dependent variable(s)

Output the solution of the initial value problem specified by the data

Example The command **s.syslap(**$[1,5]$, $2u(t-3)$, 0, 1**)** yields the solution to the initial value problem

$$y' + 5y = 2u(t-3), \quad y(0) = 1$$

where u is the Heaviside unit step function; namely,

$$y(t) = e^{-5t}\left(u(t) - \frac{2e^{15}u(t-3)}{5}\right) + \frac{2u(t-3)}{5}.$$

Similarly, **s.syslap(**$[1,5,6]$, $e^t\cos(2t)$, 0, $[2,3]$**)** yields the solution to the initial value problem

$$y'' + 5y' + 6y = e^t\cos 2t, \quad y(0) = 2, \quad y'(0) = 3;$$

namely,

$$y(t) = \tfrac{2}{65}e^t\cos 2t + \tfrac{7}{130}e^t\sin 2t + \tfrac{114}{13}e^{-2t} - \tfrac{34}{5}e^{-3t}.$$

See Also **ILAP**, **LAP**, **LTLDO**

Notes An example of using **syslap** to solve a system of linear differential equations with initial conditions is given in a streaming video. See *http://calclab.math.tamu.edu/~belmonte/TAMUDFEQ/TAMUDFEQ.html* for details.

TES

Description The **tes** function returns a “**t**runcated **e**xponential **s**eries,” as specified in the **Output** block below.

Access **e.tes** library shortcut in the *eigen* group

Syntax **e.tes(*A*,*r*,*w*)** or **e.tes(*A*,*r*,*I*)**

Input ***A***: an $n \times n$ square matrix with constant elements
r: an eigenvalue of ***A*** whose geometric multiplicity is less than its algebraic multiplicity
w: a generalized eigenvector associated with r
I: the $n \times n$ identity matrix

Output The command returns either

$$e^{rt}\left(\sum_{k=0}^{n-1}\frac{t^k}{k!}\left(\mathbf{A}-r\mathbf{I}\right)^k\right)\mathbf{w},$$

a solution of $\mathbf{x}' = \mathbf{A}\mathbf{x}$, or

$$e^{rt}\left(\sum_{k=0}^{n-1}\frac{t^k}{k!}\left(\mathbf{A}-r\mathbf{I}\right)^k\right)\mathbf{I},$$

the latter of which gives the matrix exponential function $e^{t\mathbf{A}}$ in the case that r is the lone eigenvalue of ***A***.

Examples Let ***A***= $\begin{bmatrix} 42 & 90 & -40 & 0 \\ -24 & -43 & 20 & -4 \\ -9 & 0 & 2 & -9 \\ -40 & -90 & 40 & 2 \end{bmatrix}$. In the example for the **kernelbasis** command, $\mathbf{w} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ is seen to be a generalized eigenvector for the eigenvalue $r = 2$ of ***A***. Then **e.tes(*A*,** 2**, *w*)** yields $\begin{bmatrix} 0 \\ -4te^{2t} \\ -9te^{2t} \\ e^{2t} \end{bmatrix}$, a linearly independent solution of $\mathbf{x}' = \mathbf{A}\mathbf{x}$.

Similary, the matrix ***A***= $\begin{bmatrix} -1 & 1 \\ -4 & -5 \end{bmatrix}$ has the lone eigenvalue $r = -3$ with algebraic multiplicity 2 (for example, via **evmt**). Then **e.tes(*A*,** -3**, *I*)** where ***I***= $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ yields $e^{t\mathbf{A}} = \begin{bmatrix} (2t+1)\,e^{-3t} & te^{-3t} \\ -4te^{-3t} & (1-2t)\,e^{-3t} \end{bmatrix}$, as verified by **expmat**.

See Also **EIGENVECTS**

U2U

Description The **u2u** program converts an expression in t to one in x for graphing purposes.

Access **s.u2u** library shortcut in the *Laplace* group

Syntax **s.u2u(*e*,*“f”*)**

Input ***e***: a scalar or vector expression in t, possibly involving Heaviside unit step functions
“f”: the name of the function (a string) which will contain the converted expression

Output The expression ***e*** has all occurrences of the variable t changed to x and function references to $u()$ changed to $\hat{u}()$. The command then defines the Heaviside unit step function $\hat{u}(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$.

Examples The command **s.syslap(**$[1,5]$, $2u(t-3)$, 0, 1**)** yields the solution to the initial value problem

$$y' + 5y = 2u(t-3), \quad y(0) = 1$$

where u is the Heaviside unit step function; namely,

$$e^{-5t}\left(u(t) - \frac{2e^{15}u(t-3)}{5}\right) + \frac{2u(t-3)}{5}.$$

Then **s.u2u(Ans,“f”)** defines the function

$$f(x) = e^{-5x}\left(\hat{u}(x) - \frac{2e^{15}\hat{u}(x-3)}{5}\right) + \frac{2\hat{u}(x-3)}{5}$$

where $\hat{u}(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$. One may then use the Graphs and Geometry application to graph the solution by referencing $f(x)$.

See Also **GRAIL**, **SYSLAP**

WRON

Description The **wron** function computes the Wronskian matrix of a fundamental solution set of an n-th order linear differential equation.

Access **l.wron** library shortcut in the *linear* group

Syntax **l.wron(*v*,*x*)**

Input ***v***: a row vector whose elements constitute the fundamental solution set
x: the independent variable in which is ***v*** expressed

Output the Wronskian matrix consisting of the derivatives $\dfrac{d^{(k)}\mathbf{v}}{dx^k}$, $k = 0, 1, \ldots n-1$, as row vectors (zeroth derivative: ***v***)

Example The command **l.wron(**$\left[e^{3x}, xe^{3x}, \cos(2x), \sin(2x)\right]$, x**)** returns

$$\begin{bmatrix} e^{3x} & xe^{3x} & \cos 2x & \sin 2x \\ 3e^{3x} & (3x+1)e^{3x} & -2\sin 2x & 2\cos 2x \\ 9e^{3x} & (9x+6)e^{3x} & -4\cos 2x & -4\sin 2x \\ 27e^{3x} & (27x+27)e^{3x} & 8\sin 2x & -8\cos 2x \end{bmatrix}.$$

See Also **CZPC**

[Please turn the page for the last command!]

ABOUT

Description	The α**bout** program displays TAMUDFEQ credits.
Access	**s.**α**bout** library shortcut in the *Laplace* group
Syntax	**s.**α**bout()**
Input	(none)
Output	display of TAMUDFEQ credits
Example	The command returns the following display. TAMUDFEQ 1.2 Monday, 06 July 2009 Differential Equations package for TI-Nspire CAS © 2009, Art Belmonte; under GPLv3, 2007 EMAIL: Art.Belmonte@math.tamu.edu
See Also	Information regarding the package is comprised of this command reference document and online streaming videos.
Note	The first letter of the command is the lowercase Greek letter alpha (α). This causes the command to appear last in both the *Laplace* pop-up menu as well as the doubly-linked variables menu for easy access. (The command has nothing to do with Laplace transforms. It's just that there was an open slot in the menu in which to place the command.)