Package ‘invLT’

Type Package
Title Inversion of Laplace-Transformed Functions
Version 0.2.1
Date 2015-08-25
Author Christopher Barry
Maintainer Christopher Barry <cjb309@bham.ac.uk>
Description Provides two functions for the numerical inversion of Laplace-Transformed functions, returning the value of the standard (time) domain function at a specified value. The first algorithm is the first optimum contour algorithm described by Evans and Chung (2000)[1]. The second algorithm uses the Bromwich contour as per the definition of the inverse Laplace Transform. The latter is unstable for numerical inversion and mainly included for comparison or interest. There are also some additional functions provided for utility, including plotting and some simple Laplace Transform examples, for which there are known analytical solutions. Polar-cartesian conversion functions are included in this package and are used by the inversion functions.
License MIT + file LICENSE
LazyData TRUE
NeedsCompilation no
Repository CRAN
Date/Publication 2015-09-03 13:26:48

R topics documented:

BrC.r ................................................................. 2
iv.opC ............................................................... 2
ivLT.plot .......................................................... 3
L.t ................................................................. 4
opC.r ............................................................... 5
r.xy ............................................................... 6
x.rphi .............................................................. 6
**Bromwich Contour**

**Description**

The Bromwich contour with polar co-ordinates \( r \) as a function of \( \phi \)

**Usage**

\[ \text{BrC.r}(\phi, \gamma = 1) \]
\[ \text{BrC.ds_dphi}(\phi, \gamma = 1) \]

**Arguments**

- \( \phi \) value
- \( \gamma \) the Bromwich contour is a straight line and intersects the real axis at \( \gamma \)

**Functions**

- \( \text{BrC.r}: r \)
- \( \text{BrC.ds_dphi}: ds/d\phi \)

---

**Inverse Laplace Transform**

**Description**

Functionals that numerically invert a Laplace Transform.

**Usage**

\[ \text{iv.opC}(\text{FUN}, t, \text{nterms} = 31L, m = 1, \text{fail.val} = \text{NA}) \]
\[ \text{iv.opChalf}(\text{FUN}, t, \text{nterms} = 16L, m = 1, \text{fail.val} = \text{NA}) \]
\[ \text{iv.BrC}(\text{FUN}, t, \text{nterms} = 1000L, \gamma = 1) \]
Arguments

L.FUN the Laplace-Transformed function

t standard (time) domain function at which to evaluate

nterms number of terms to use in the numerical inversion (odd number safest for iv.opC, even for iv.opChalf)

m see opC.r documentation

fail.val value to return in event of failure to converge

gamma the Bromwich contour is a straight line and intersects the real axis at $\gamma$

Details

Optimum contour based on:

Evans & Chung, 2000: Laplace transform inversions using optimal contours in the complex plane


Functions

- iv.opC: inversion using the full optimum contour
- iv.opChalf: for functions which are symmetric about the real axis, it is sufficient to use half the optimum contour and half the number of subdivisions (nterms)
- iv.BrC: inversion using the Bromwich contour (the definition, but very unstable for numerical evaluation - not recommended)

Examples

```r
vals <- seq(-pi/2, pi/2, length.out = 7)
sinvals <- vapply(vals, iv.opC, complex(1), L.FUN = L.sin)
plot(vals, Re(sinvals), type = "l")
```

ivLT.plot  

*Plot Laplace Transform inversion*

Description

Plots the results of a Laplace Transform inversion at multiple time values.

Usage

```r
ivLT.plot(L.FUN, METHOD = iv.opC, tPnts = seq(-2, 5, 0.1), nterms = 100, ...)
```
Arguments

L.FUN  the Laplace-Transformed function
METHOD  inversion algorithm to use (iv.opC, iv.opChalf or iv.BrC)
tPts    time points at which to plot
nterms  number of terms to use in the numerical inversion (odd number safest for iv.opC, even for iv.opChalf)
...     graphical parameters for plot

Details

This function is useful for investigating the performance of a Laplace Transform inversion over a range of time values. Use for example with the LT functions provided in with this package (invLT).

Examples

ivLT.plot(L.tsq, iv.opC, nterms = 31L)
ivLT.plot(L.tsq, iv.opC, nterms = 1000L)
ivLT.plot(L.tsq, iv.opChalf, nterms = 16L)
ivLT.plot(L.tsq, iv.opChalf, nterms = 1000L)
ivLT.plot(L.tsq, iv.BrC, nterms = 31L)
ivLT.plot(L.tsq, iv.BrC, nterms = 1000L)

L.t

Laplace Transforms

Description

Laplace Transforms of common functions. Useful for testing out LT inversion functions and whether sufficient precision is being used.

Usage

L.t(p)
L.tsq(p)
L.exp(p)
L.cos(p)
L.sin(p)
L.H(p)

Arguments

p  Laplace domain variable (commonly called s elsewhere)
Functions

- \( L.t \): LT of \( t \)
- \( L.tsq \): LT of \( t^2 \)
- \( L.exp \): LT of \( e^{(-t)} \)
- \( L.cos \): LT of \( \cos(t) \)
- \( L.sin \): LT of \( \sin(t) \)
- \( L.H \): LT of Heaviside unit function stepping at 1: (if \( p < 1 \) 0 else 1)

---

**Description**

The optimum contour with polar co-ordinates \( r \) as a function of \( \phi \) and complex length increment with \( \phi \) (Evans and Chung, 2000)

**Usage**

\[
\text{opC.r(\phi, m = 1, t = 5)}
\]

\[
\text{opC.ds_dphi(\phi, m = 1, t = 5)}
\]

**Arguments**

- \( \phi \) value
- \( m \) width of the contour - too small and get too close to singularities on negative \( x \)-axis, too large and encounter instability for large positive \( x \)
- \( t \) standard (time) domain variable, also affects contour width

**Details**

if \( t \) is set as zero, it is changed to 5 (avoids dividing by 0)

Evans & Chung, 2000: Laplace transform inversions using optimal contours in the complex plane


**Functions**

- \( \text{opC.r: } r \)
- \( \text{opC.ds_dphi: } ds/d\phi \)
**r.xy**  
*Cartesian to Polar*

**Description**  
Conversion from cartesian to polar co-ordinates

**Usage**

\[
\begin{align*}
  r(xy(x, y)) \\
  \phi(xy(x, y))
\end{align*}
\]

**Arguments**

- \(x\)  
  x co-ordinate
- \(y\)  
  y co-ordinate

**Value**

\(r\) or \(\phi\) respectively from \(x\) and \(y\)

**Functions**

- \(r.xy\): Returns polar co-ordinate \(r\) from cartesian co-ordinates \(x\) and \(y\).
- \(\phi.xy\): Returns polar co-ordinate \(\phi\) (anti-clockwise rotation from positive x-axis) from cartesian co-ordinates \(x\) and \(y\).

**x.rphi**  
*Polar to Cartesian*

**Description**  
Conversion from polar to cartesian co-ordinates

**Usage**

\[
\begin{align*}
  x.rphi(r, \phi) \\
  y.rphi(r, \phi)
\end{align*}
\]

**Arguments**

- \(r\)  
  distance from origin
- \(\phi\)  
  anti-clockwise rotation from positive x-axis
Value

x or t respectively from r and phi

Functions

- `x.rphi`: Returns cartesian co-ordinate x from polar co-ordinates r and phi.
- `y.rphi`: Returns cartesian co-ordinate y from polar co-ordinates r and phi.
Index

BrC.ds_dphi (BrC.r), 2
BrC.r, 2

iv.BrC (iv.opC), 2
iv.opC, 2
iv.opChalf (iv.opC), 2
ivLT.plot, 3

L.cos (L.t), 4
L.exp (L.t), 4
L.H (L.t), 4
L.sin (L.t), 4
L.t, 4
L.tsq (L.t), 4

opC.ds_dphi (opC.r), 5
opC.r, 3, 5

phi.xy (r.xy), 6
plot, 4

r.xy, 6
x.rphi, 6
y.rphi (x.rphi), 6