

Package ‘GFT’

August 23, 2026

Type Package

Title Generalized Fisher Transformation of Correlation Matrices

Version 1.0.1

Description Forward and inverse generalized Fisher transformation (GFT) of correlation matrices, $\gamma = \text{vecl}(\log C)$, which maps the positive definite correlation matrices one-to-one onto the Euclidean space of dimension $n(n-1)/2$, see Archakov and Hansen (2021) [doi:10.3982/ECTA16910](https://doi.org/10.3982/ECTA16910). The inverse is computed from a variational characterization by the GFT-FP+N algorithm: a fixed-point phase in the log domain followed by a matrix-free inexact Newton phase with preconditioned conjugate gradients. Reference implementations of the plain fixed point, Broyden's method, and full Newton are included. Uses base R only.

License MIT + file LICENSE

Encoding UTF-8

Depends R ($\geq 3.5.0$)

Suggests testthat ($\geq 3.0.0$), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

URL <https://github.com/reinhardhansen/GFT>

BugReports <https://github.com/reinhardhansen/GFT/issues>

NeedsCompilation no

Author Ilya Archakov [aut],
Peter Reinhard Hansen [aut, cre]

Maintainer Peter Reinhard Hansen <hansen@unc.edu>

Repository CRAN

Date/Publication 2026-08-23 15:40:08 UTC

Contents

GFT-package	2
gft	3
inv_gft	4
inv_gft_broyden	6
inv_gft_fp	7
inv_gft_newton	8
print.gft_inv	9
vecl	10
Index	11

GFT-package

Generalized Fisher Transformation of Correlation Matrices

Description

Forward and inverse generalized Fisher transformation (GFT) of correlation matrices. The GFT maps a non-singular $n \times n$ correlation matrix C to the real vector $\gamma = \text{vecl}(\log C)$ of below-diagonal elements of the matrix logarithm of C . The map is a bijection between the set of positive definite correlation matrices and R^d with $d = n(n-1)/2$ (Archakov and Hansen, 2021), and generalizes Fisher's z -transformation, to which it reduces for $n = 2$.

Details

The forward map is computed by [gft](#). The inverse is computed from the variational characterization

$$x^*(z) = \arg \min_x \text{tr} e^{A[x;z]} - \sum_i x_i,$$

where $A[x; z]$ is symmetric with off-diagonal elements z and diagonal x , by the following solvers:

[inv_gft](#) GFT-FP+N (recommended): fixed-point phase in the log domain, then matrix-free inexact Newton via preconditioned conjugate gradients.

[inv_gft_fp](#) The Archakov-Hansen fixed point.

[inv_gft_broyden](#) Broyden's method as in Chen, Fei and Yu (2025).

[inv_gft_newton](#) Full Newton with the exact $O(n^4)$ Hessian.

The implementation is a line-faithful port of the Julia reference implementation by the same authors and uses base R only.

Author(s)

Ilya Archakov and Peter Reinhard Hansen.

Maintainer: Peter Reinhard Hansen <hansen@unc.edu>

References

- Archakov, I. and Hansen, P. R. (2021). A new parametrization of correlation matrices. *Econometrica*, **89**(4), 1699–1715. doi:[10.3982/ECTA16910](https://doi.org/10.3982/ECTA16910)
- Archakov, I. and Hansen, P. R. (2026). A variational approach to the generalized Fisher transformation of correlation matrices. Working paper.
- Chen, H., Fei, Y. and Yu, J. (2025). Multivariate stochastic volatility models based on generalized Fisher transformation. *Journal of Econometrics*, **251**, 106041.

Examples

```
C <- 0.9^abs(outer(1:5, 1:5, "-")) # Toeplitz correlation matrix
z <- gft(C)                        # forward transformation
r <- inv_gft(z)                    # inverse transformation
max(abs(r$C - C))                  # round trip at machine precision
```

gft

Generalized Fisher Transformation

Description

Computes the generalized Fisher transformation $\gamma = \text{vecl}(\log C)$: the below-diagonal elements, stacked column by column, of the matrix logarithm of a positive definite correlation matrix C .

Usage

```
gft(C)
```

Arguments

C a positive definite correlation matrix (square, symmetric, numeric). The symmetric part $(C + C')/2$ is used.

Details

The matrix logarithm is computed from the eigendecomposition of C . For $n = 2$ the transformation reduces to Fisher's classical z -transformation $z = \text{atanh}(\rho)$. The map is a bijection between the positive definite correlation matrices and $R^{n(n-1)/2}$; its inverse is computed by [inv_gft](#).

Value

A numeric vector of length $n(n - 1)/2$ containing $\text{vecl}(\log C)$.

References

- Archakov, I. and Hansen, P. R. (2021). A new parametrization of correlation matrices. *Econometrica*, **89**(4), 1699–1715. doi:[10.3982/ECTA16910](https://doi.org/10.3982/ECTA16910)

See Also

[inv_gft](#), [vecl](#), [unvecl](#).

Examples

```
# n = 2: reduces to Fisher's z-transformation
C <- matrix(c(1, 0.5, 0.5, 1), 2, 2)
all.equal(gft(C), atanh(0.5))

C <- 0.9^abs(outer(1:5, 1:5, "-"))
z <- gft(C)
max(abs(inv_gft(z)$C - C))
```

inv_gft

Inverse Generalized Fisher Transformation (GFT-FP+N)

Description

Reconstructs the unique positive definite correlation matrix C with $\text{vecl}(\log C) = z$ by the GFT-FP+N algorithm (recommended solver).

Usage

```
inv_gft(z, x0 = NULL, tol = 1e-13, maxit = 500, delta = 1,
        exact_hess = FALSE)
```

Arguments

<code>z</code>	numeric vector of length $n(n-1)/2$: the below-diagonal elements of $\log C$, stacked column by column.
<code>x0</code>	optional numeric vector of length n : starting value for the diagonal of $\log C$ (e.g. from a previous solution, for warm starts). Defaults to zero.
<code>tol</code>	convergence tolerance on $\ \text{diag}(e^A) - 1\ _\infty$.
<code>maxit</code>	maximum number of iterations.
<code>delta</code>	phase switch threshold: fixed-point steps are used while $\ \text{diag}(e^A) - 1\ _\infty > \delta$.
<code>exact_hess</code>	if TRUE, solve the Newton system with the explicit $O(n^4)$ Hessian and a Cholesky factorization instead of matrix-free conjugate gradients. Identical algorithm otherwise; mainly for benchmarking.

Details

The solution solves the strictly convex problem

$$x^*(z) = \arg \min_x \text{tr} e^{A[x;z]} - \sum_i x_i,$$

where $A[x;z]$ is symmetric with off-diagonal elements z and diagonal x . Phase 1 applies fixed-point steps $x \leftarrow x - \log \text{diag}(e^A)$ in the log domain (robust to overflow for any spectrum). Phase

2 is an inexact Newton method: the system $HS = -g$ is solved matrix-free by conjugate gradients with Jacobi preconditioner $\text{diag}(e^A)$, exact Hessian-vector products (two matrix multiplications each), and forcing tolerance $\eta = \min(1/2, \sqrt{\|g\|})$ (Eisenstat and Walker, 1996), giving superlinear convergence of order $3/2$. Steps are safeguarded by Armijo backtracking on the objective, with a fixed-point step substituted if the line search fails, and a fixed-point finisher if progress stalls at the rounding floor.

Value

An object of class "gft_inv": a list with components

x	the solution: diagonal of $\log C$.
C	the reconstructed correlation matrix $e^{A[x^*;z]}$.
iters	number of iterations.
eighs	number of eigendecompositions, the dominant $O(n^3)$ kernel.
hvs	number of Hessian-vector products.
err	final value of $\ \text{diag}(e^A) - 1\ _\infty$.
converged	logical.
hist	the error after each iteration.

References

- Archakov, I. and Hansen, P. R. (2021). A new parametrization of correlation matrices. *Econometrica*, **89**(4), 1699–1715. doi:[10.3982/ECTA16910](https://doi.org/10.3982/ECTA16910)
- Archakov, I. and Hansen, P. R. (2026). A variational approach to the generalized Fisher transformation of correlation matrices. Working paper.
- Eisenstat, S. C. and Walker, H. F. (1996). Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing*, **17**(1), 16–32.

See Also

[gft](#) for the forward map; [inv_gft_fp](#), [inv_gft_broyden](#), [inv_gft_newton](#) for the reference solvers.

Examples

```
C <- 0.9^abs(outer(1:5, 1:5, "-"))
z <- gft(C)
r <- inv_gft(z)
r
max(abs(r$C - C))

# warm start from a nearby solution
z2 <- z + 0.01
r2 <- inv_gft(z2, x0 = r$x)

# an arbitrary vector in R^d is always a valid input
set.seed(1)
```

```
ra <- inv_gft(rnorm(45, sd = 2)) # n = 10
range(diag(ra$C))                # unit diagonal
min(eigen(ra$C)$values) > 0      # positive definite
```

inv_gft_broyden	<i>Inverse GFT by Broyden's Method</i>
-----------------	--

Description

Reconstructs the correlation matrix C with $\text{vecl}(\log C) = z$ by Broyden's method applied to the residual $F(x) = \log \text{diag}(e^{A[x;z]})$, as in Chen, Fei and Yu (2025). Reference implementation for benchmarking against [inv_gft](#).

Usage

```
inv_gft_broyden(z, x0 = NULL, tol = 1e-13, maxit = 500, warm = 1,
               globalized = FALSE)
```

Arguments

<code>z</code>	numeric vector of length $n(n-1)/2$.
<code>x0</code>	optional starting value of length n ; defaults to zero.
<code>tol</code>	convergence tolerance on $\ \text{diag}(e^A) - 1\ _\infty$.
<code>maxit</code>	maximum number of iterations.
<code>warm</code>	number of initial fixed-point steps before the Jacobian is formed (ignored when <code>globalized = TRUE</code>).
<code>globalized</code>	if TRUE, replace the one-step initialization with the same log-domain fixed-point phase as GFT-FP+N (fixed-point steps until $\max_i \ell_i \leq \log 2$), and only then form the Jacobian.

Details

The exact Jacobian is computed once (an $O(n^4)$ Hessian), then updated by rank-one Sherman-Morrison updates of its inverse, with one eigendecomposition per iteration and no line search. Without globalization the method can diverge for large $\|z\|$; divergence is reported gracefully via `converged = FALSE`.

Value

An object of class "gft_inv"; see [inv_gft](#) for the components. On divergence the result has `converged = FALSE` and `err = Inf`.

References

Chen, H., Fei, Y. and Yu, J. (2025). Multivariate stochastic volatility models based on generalized Fisher transformation. *Journal of Econometrics*, **251**, 106041.

See Also[inv_gft.](#)**Examples**

```

z <- gft(0.9^abs(outer(1:5, 1:5, "-")))
r <- inv_gft_broyden(z)
r$converged

```

inv_gft_fp

*Inverse GFT by the Archakov-Hansen Fixed Point***Description**

Reconstructs the correlation matrix C with $\text{vecl}(\log C) = z$ by the fixed-point iteration of Archakov and Hansen (2021), $x \leftarrow x - \log \text{diag}(e^{A[x;z]})$, evaluated in the log domain throughout.

Usage

```
inv_gft_fp(z, x0 = NULL, tol = 1e-13, maxit = 5000)
```

Arguments

<code>z</code>	numeric vector of length $n(n-1)/2$.
<code>x0</code>	optional starting value of length n ; defaults to zero.
<code>tol</code>	convergence tolerance on $\ \text{diag}(e^A) - 1\ _\infty$.
<code>maxit</code>	maximum number of iterations.

Details

Globally convergent, with one eigendecomposition per iteration. The local linear rate degrades as the spectrum of C spreads; [inv_gft](#) switches to an inexact Newton phase precisely to avoid this slowdown while retaining the fixed point's robustness.

Value

An object of class "gft_inv"; see [inv_gft](#) for the components.

References

Archakov, I. and Hansen, P. R. (2021). A new parametrization of correlation matrices. *Econometrica*, **89**(4), 1699–1715. doi:[10.3982/ECTA16910](https://doi.org/10.3982/ECTA16910)

See Also[inv_gft.](#)

Examples

```
z <- gft(0.9^abs(outer(1:5, 1:5, "-")))
r <- inv_gft_fp(z)
r$eighs
```

inv_gft_newton	<i>Inverse GFT by Full Newton</i>
----------------	-----------------------------------

Description

Reconstructs the correlation matrix C with $\text{vecl}(\log C) = z$ by a full Newton method with the exact $O(n^4)$ Hessian recomputed at every iteration, Armijo backtracking on the objective, and an optional fixed-point warm start. Reference implementation for benchmarking against [inv_gft](#).

Usage

```
inv_gft_newton(z, x0 = NULL, tol = 1e-13, maxit = 500, warm = 1,
               safeguard = TRUE)
```

Arguments

<code>z</code>	numeric vector of length $n(n-1)/2$.
<code>x0</code>	optional starting value of length n ; defaults to zero.
<code>tol</code>	convergence tolerance on $\ \text{diag}(e^A) - 1\ _\infty$.
<code>maxit</code>	maximum number of iterations.
<code>warm</code>	number of initial fixed-point steps.
<code>safeguard</code>	logical. If TRUE (the default), apply the two rounding-floor safeguards described below. If FALSE, reproduce the published comparator with only the Armijo line search added, which may stagnate short of <code>tol</code> .

Details

Each iteration forms the exact Hessian in $O(n^4)$ time and solves the Newton system by Cholesky factorization; if the factorization fails or the line search cannot make progress, a fixed-point step is substituted. [inv_gft](#) obtains the same fast local convergence at $O(n^3)$ cost per iteration by solving the Newton system inexactly and matrix-free.

Near the solution the objective $f = \text{tr}(e^A) - 1'x$ is flat to within cancellation, so the Armijo condition carries no information: the line search exhausts all of its halvings, the fixed-point fallback is taken, and the iteration can lock at an error several orders of magnitude above the attainable floor. With `safeguard = TRUE` two further rules prevent this. The full Newton step is taken untested once the predicted decrease falls below the floating-point resolution of f , and a persistent lack of progress below 10^{-9} hands the iteration to the contractive fixed point.

`safeguard = FALSE` disables both and reproduces the comparator benchmarked in the reference below, namely the method of Chen, Fei and Yu (2025) with only the Armijo line search added. In that mode converged can be FALSE on a small fraction of otherwise well conditioned problems; this is the documented behaviour of that variant, not a defect of the implementation.

Value

An object of class "gft_inv"; see [inv_gft](#) for the components.

References

Archakov, I. and Hansen, P. R. (2026). A variational approach to the generalized Fisher transformation of correlation matrices. Working paper.

See Also

[inv_gft.](#)

Examples

```
z <- gft(0.9^abs(outer(1:5, 1:5, "-")))
r <- inv_gft_newton(z)
r$converged
```

print.gft_inv

Print an Inverse GFT Result

Description

Compact display of an object of class "gft_inv" as returned by [inv_gft](#) and the other inverse-GFT solvers.

Usage

```
## S3 method for class 'gft_inv'
print(x, ...)
```

Arguments

x an object of class "gft_inv".
 ... ignored.

Value

x, invisibly.

Examples

```
r <- inv_gft(gft(0.9^abs(outer(1:5, 1:5, "-"))))
print(r)
```

vecl

*Stack and Unstack Below-Diagonal Elements***Description**

vecl stacks the below-diagonal elements of a square matrix column by column. unvec1 is its right inverse on symmetric matrices with zero diagonal: it forms the symmetric matrix with zero diagonal and below-diagonal elements z.

Usage

```
vecl(M)
unvec1(z)
```

Arguments

M a square matrix.
 z a numeric vector of length $n(n-1)/2$ for some integer n .

Value

vecl returns a numeric vector of length $n(n-1)/2$. unvec1 returns an $n \times n$ symmetric numeric matrix with zero diagonal.

Examples

```
M <- matrix(1:9, 3, 3)
vecl(M)                    # c(2, 3, 6)
unvec1(c(0.1, 0.2, 0.3))
```

Index

- * **array**
 - vecl, [10](#)
- * **multivariate**
 - gft, [3](#)
 - inv_gft, [4](#)
 - inv_gft_broyden, [6](#)
 - inv_gft_fp, [7](#)
 - inv_gft_newton, [8](#)
- * **optimize**
 - inv_gft, [4](#)
 - inv_gft_broyden, [6](#)
 - inv_gft_fp, [7](#)
 - inv_gft_newton, [8](#)
- * **package**
 - GFT-package, [2](#)
- * **print**
 - print.gft_inv, [9](#)

GFT (GFT-package), [2](#)
gft, [2](#), [3](#), [5](#)
GFT-package, [2](#)

inv_gft, [2–4](#), [4](#), [6–9](#)
inv_gft_broyden, [2](#), [5](#), [6](#)
inv_gft_fp, [2](#), [5](#), [7](#)
inv_gft_newton, [2](#), [5](#), [8](#)

print.gft_inv, [9](#)

unvecl, [4](#)
unvecl (vecl), [10](#)

vecl, [4](#), [10](#)