

# Package ‘diffMeshGP’

January 28, 2026

**Type** Package

**Title** Multi-Fidelity Computer Experiments Using the Tuo-Wu-Yu Model

**Version** 0.1.1

**Date** 2026-01-25

**Description** This R function implements the nonstationary Kriging model proposed by Tuo, Wu and Yu (2014) <[DOI:10.1080/00401706.2013.842935](https://doi.org/10.1080/00401706.2013.842935)> for analyzing multi-fidelity computer outputs. This function computes the maximum likelihood estimates for the model parameters as well as the predictive means and variances of the exact solution.

**License** GPL (>= 2)

**Encoding** UTF-8

**Imports** stats

**RoxygenNote** 7.3.0

**NeedsCompilation** no

**Author** Wenjia Wang [aut, cre],  
Rui Tuo [aut],  
C. F. Jeff Wu [aut]

**Maintainer** Wenjia Wang <[1992wangwenjia@gmail.com](mailto:1992wangwenjia@gmail.com)>

**Repository** CRAN

**Date/Publication** 2026-01-28 19:20:02 UTC

## Contents

GPdiffMesh . . . . .	2
Index	5

## Description

This R function implements the nonstationary Kriging model proposed by Tuo, Wu and Yu (2014) for analyzing multi-fidelity computer outputs. This function computes the maximum likelihood estimates for the model parameters as well as the predictive means and variances of the exact solution (i.e., the conceptually highest fidelity).

## Usage

```
GPdiffMesh(x,X,meshT,Y,regFunX = function(x){ return(0*matrix(x[,1]))},
           regFunT = function(x){ return(1*matrix(x[,1]))},
           phi1 = 1,
           sigma12 = 1,
           sigma22 = 1,
           phi2 = 1,
           mybeta = FALSE,
           l = 4)
```

## Arguments

- |                                      |   |
|--------------------------------------|---|
| x                                    | An $(n_0 \times d)$ matrix of control points on which the values of the exact solution will be predicted. Each row represents one input point.  |
| X                                    | An $(n \times d)$ matrix of control variables.  |
| meshT                                | An $(n \times 1)$ mesh density matrix. A same row of X and meshT represents the inputs from one trial of the computer experiment.   |
| Y                                    | An $(n \times 1)$ computer output matrix corresponding to (X, meshT).   |
| regFunX                              | A scalar or a vector-valued regression function $f_1(x)$ . The default value is $f_1(x) = 1$ . See Details below.   |
| regFunT                              | A scalar or a vector-valued regression function $f_2(x, t)$ . The default value is $f_2(x, t) = t$ . See Details below.   |
| phi1, sigma12, sigma22, phi2, mybeta | Initial values of the parameters for the maximum likelihood estimation. The default values are $\phi_1 = 1, \sigma_{12} = 1, \sigma_{22} = 1, \phi_2 = 1, \text{mybeta} = 1$ . See Details below. |
| l                                    | The value of parameter $l$ in the covariance function of the nonstationary Kriging model. (We used fixed $l$ in this R function.) The default value is $l = 4$ . See Details below.               |

## Details

This R function implements the nonstationary Kriging model proposed by Tuo, Wu and Yu (2014) for the modeling and analysis of multi-fidelity computer experiments. Denote an input-output pair from the computer simulation by  $(x, t, y)$ , where  $x$  is the vector of input variables,  $t$  is the mesh density, and  $y$  is the corresponding computer output. Tuo, Wu and Yu (2014) use the following Gaussian process model:

$$y = \eta(x, t) = \eta(x, 0) + \delta(x, t),$$

where  $\eta(x, 0)$  and  $\delta(x, t)$  are realizations of two mutually independent Gaussian processes  $V(x)$  and  $Z(x, t)$ . Assume

$$E(V(x)) = f_1^T(x)\beta_1, E(Z(x, t)) = f_2^T(x, t)\beta_2,$$

where  $f_1(x)$  can be set in `regFunX`, and  $f_2(x, t)$  can be set in `regFunT`. Both  $f_1(x)$  and  $f_2(x, t)$  can be vector-valued function.  $f_2(x, t)$  should satisfy for any  $x$ ,  $\lim_{t \rightarrow 0} f_2(x, t) = 0$ . The default functions are  $f_1(x) = 1$  and  $f_2(x, t) = t$ . Assume the covariance of  $V(x)$  is

$$\text{Cov}(V(x_1), V(x_2)) = \sigma_1^2 e^{-\phi_1^2 \|x_1 - x_2\|_2^2},$$

and the covariance of  $Z(x, t)$  is

$$\text{Cov}(Z(x_1, t_1), Z(x_2, t_2)) = \sigma_2^2 \min(t_1, t_2)^l e^{-\phi_2^2 \|x_1 - x_2\|_2^2},$$

where  $l$  is a fixed parameter and is not estimated in this function.

The goal of the Tuo-Wu-Yu model is to predict for the *exactsolution*  $\varphi(\cdot) := \eta(\cdot, 0)$ . For a set of control points  $x = (x_1, \dots, x_{n_0})$ , this R function predicts  $\eta(x_i, 0) = \varphi(x_i)$  for  $i = 1, \dots, n_0$ .

This R function uses maximum likelihood method to estimate the model parameters. Nelder-Mead method is used to maximize the likelihood function. The solution may depend on the choice of initial values. Users may specify the initial values or use the default values.

## Value

<code>outy</code>	An $(n_0 \times 1)$ matrix of predictive means corresponding to control points. A same row of <code>x</code> and <code>outy</code> represents a pair of one control point and corresponding predictive mean.
<code>sigy</code>	An $(n_0 \times 1)$ matrix of predictive variances corresponding to control points. A same row of <code>x</code> and <code>sigy</code> represents a pair of one control point and corresponding predictive variances.
<code>estipar</code>	A list of parameter estimates.

## Examples

```
hig02 <- function(s)
{
  #The test function is from [2].
  y <- s*sin(s) / 10
  return(y)
}
myX <- matrix(c(seq(from = 0, to = 10, by = 1),
  seq(from = 0, to = 10, by = 1)), ncol = 2)
myy <- hig02(matrix(myX[,1]))
myT <- matrix(c(0.01, 0.5, 0.01, 0.02, 0.02, 0.01, 0.01, 0.02, 0.002,
```

```

    0.003,0.03))
myregf <- function(x){
  return(x)
}
myregfn <- function(s){
  return(cbind((matrix(s[,1])^2*matrix(s[,2])),
    (matrix(s[,1])*matrix(s[,2]))))
}
#Here s=cbind(t,x), where x is a matrix of input variables and
#   t is the corresponding mesh density matrix.
x <- matrix(c(seq(from = 0,to = 10, by = 0.1),
  seq(from = 0,to = 10, by = 0.1)),ncol = 2)
myploty <- hig02(matrix(x[,1]))
y <- GPdiffMesh(x, myX, myT, myy, regFunX = myregf, regFunT = myregfn)
# The regression function is  $\beta_0 + \beta_1 x + (t^2x, tx)' \beta_2$ 
y$outy
y$sigy
y$estipar
plot(x[,1], myploty,"l")
lines(x[,1],y$outy, type="o", pch=22, lty=2, col="red")

```

# Index

GPdiffMesh, [2](#)