# Fractional Gaussian Noise (FGN)

## Parametrization

The (stationary) FGN (Gaussian) process has correlation function at lag $k$

$$\rho(k) = |k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}$$

where $H$ is the Hurst parameter or self-similarity parameter, which we assume to be

$$1/2 \leq H < 1.$$

so the process has long range properties for $H > 1/2$. The locations of the the process is fixed to $1, 2, \ldots, n$, where $n$ is the dimension of the finite representation of the FGN process.

## Hyperparameters

The marginal precision, $\tau$, of the process is represented as

$$\tau = \exp(\theta_1)$$

The Hurst parameter $H$ is represented as

$$H = \frac{1}{2} + \frac{1}{2}\frac{\exp(\theta_2)}{1+\exp(\theta_2)}$$

and the prior is defined on $\theta = (\theta_1, \theta_2)$.

## Specification

The FGN model is specified as

```
f(<whatever>, model="fgn", order=<order>, hyper = <hyper>)
```

The parmeter `order` gives the order of the Markov approximation. Currently, only `order=3` is implemented.

### Hyperparameter spesification and default values for `model="fgn"`

**doc** `Fractional Gaussian noise model`

**hyper**

- **theta1**
  - **hyperid** `13101`
  - **name** `log precision`
  - **short.name** `prec`
  - **prior** `pc.prec`
  - **param** `3 0.01`
  - **initial** `1`
  - **fixed** `FALSE`
  - **to.theta** `function(x) log(x)`
  - **from.theta** `function(x) exp(x)`
- **theta2**

**hyperid** `13102`
**name** `logit H`
**short.name** `H`
**prior** `pcfgnh`
**param** `0.9 0.1`
**initial** `2`
**fixed** `FALSE`
**to.theta** `function(x) log((2 * x - 1) / (2 * (1 - x)))`
**from.theta** `function(x) 0.5 + 0.5 * exp(x) / (1 + exp(x))`

**constr** `FALSE`

**nrow.ncol** `FALSE`

**augmented** `TRUE`

**aug.factor** `5`

**aug.constr** `1`

**n.div.by**

**n.required** `FALSE`

**set.default.values** `TRUE`

**order.default** `4`

**order.defined** `3 4`

**pdf** `fgn`

**Hyperparameter spesification and default values for** `model="fgn2"`

**doc** `Fractional Gaussian noise model (alt 2)`

**hyper**

**theta1**

**hyperid** `13111`
**name** `log precision`
**short.name** `prec`
**prior** `pc.prec`
**param** `3 0.01`
**initial** `1`
**fixed** `FALSE`
**to.theta** `function(x) log(x)`
**from.theta** `function(x) exp(x)`

**theta2**

**hyperid** `13112`
**name** `logit H`
**short.name** `H`

**prior** `pcfgnh`

**param** `0.9 0.1`

**initial** `2`

**fixed** `FALSE`

**to.theta** `function(x) log((2 * x - 1) / (2 * (1 - x)))`

**from.theta** `function(x) 0.5 + 0.5 * exp(x) / (1 + exp(x))`

**constr** `FALSE`

**nrow.ncol** `FALSE`

**augmented** `TRUE`

**aug.factor** `4`

**aug.constr** `1`

**n.div.by**

**n.required** `FALSE`

**set.default.values** `TRUE`

**order.default** `4`

**order.defined** `3 4`

**pdf** `fgn`

## Example

```
library(FGN)
n = 1000
H = 0.77
y = SimulateFGN(n, H)
y = y - mean(y)
r = inla(y ~ -1 + f(idx, model="fgn"),
         data = data.frame(y, idx=1:n),
         control.family =list(hyper = list(prec = list(initial = 12, fixed=TRUE))))
print(c(MLE=FitFGN(y, demean=TRUE)$H,
        Post.mean=r$summary.hyperpar[2,"mean"],
        Post.mode=r$summary.hyperpar[2,"mode"]))
```

## Notes

In the example above, then the `f(idx,model="fgn")` object will expand into a Gaussian of length `(order + 1)*n`. The first $n$ elements is the FGN model (which is of interrest), then there are `order` vector of AR1 processes each of length $n$, and the sum of these AR1 processes is used to represent the FGN.

Another alternative, is `f(idx,model="fgn2")` object will expand into a Gaussian of length `order*n`, which are the cummulative sums of the the `order` vector of AR1 processes each of length $n$. If `order==3`, with weighted AR1 processes (and with the given precision), $x$, $xx$ and $xxx$, then `model="fgn2"` return the vector $(x + xx + xxx, xx + xxx, xxx)$ where $\phi_x < \phi_{xx} < \phi_{xxx}$.

The PC-prior for $H$ take two arguments $(U, \alpha)$ where $\text{Prob}(U < H < 1) = \alpha$.