Coupled Functions

Rev. 11 Kenric Nelson
Added Coupled Box-Müller Method

Rev. 10 Kenric Nelson
The compact-support domain of the CoupledNormalDistribution was producing a Power error message due 1/ $\sqrt{0}$. This was because CoupledExponential was defaulting to zero even for the reciprocal of zero. Updated CoupledNormalDistribution to be defined over the proper range for the compact-support domain. Updated CoupledExponential to evaluate to infinity for 1/0 situations. Also updated the CoupledExponentialDistribution functions.

The CoupledMultivariateDistribution function still needs attention.

Rev. 9
Add translators between the coupling (κ), risk (r), and Tsallis (q)

Rev. 8
a) Added clear and memory to each function. When the notebook is run it will clear the memory of that function. Each time the function is executed the result will be stored in memory, so that computation with the same inputs does not need to be repeated.
b) reordered coupling input to be consistent with StudentsTDistribution and related Wolfram functions

Rev. 7 Removed the parameter alpha from the definition of the coupled exponential and logarithm. This is preparation for experimenting on the proper two-parameter definitions for the functions.

Rev. 6 Added Weighted Generalized Mean

Rev. 5 Modified Coupled Product so that dimension used on outer exponent is d times Length[xList]

Rev. 4 Added Coupled Sine and Cosine Functions
Prototype initiated of Coupled Product and Coupled Sum with structure of *Mathematica* Product and Sum functions

Coupled Exponential

The variable *x* is applied to a coupled exponential function. κ controls the degree of nonlinear statistical coupling, where $\kappa = 0$ is linear domain of an exponential function. *d* and α should match the dimension and power ($x = y^{\alpha}$) of the variable *x.*

Inputs
x - variable to which coupled exponential is applied
κ - coupling parameter which modifies the coupled exponential function
d - dimension, should be equal to the dimension of x

In[∘]:=
```
ClearAll[CoupledExponential];
CoupledExponential[x_, κ_ : 0, d_ : 1] :=
 CoupledExponential[x, κ, d] =
  Which[
   κ > 0, (1 + κ x)^((1+d κ)/κ),
   -1/d ⩽ κ < 0, If[Simplify[1 + κ x] ≥ 0,
    (1 + κ x)^((1+d κ)/κ),
    If[(1 + d κ)/κ > 0, 0, ∞]
   ],
   κ == 0, Exp[x],
   True, Message[CoupledExponential::nnarg, κ]
  ]
```

In[∘]:=
```
CoupledExponential::nnarg = "Error: κ = `1/d` is not greater than -1.";
```

Coupled Logarithm

Generalization of the logarithm function, which defines smooth transition to power functions.

Inputs
x - variable to which coupled exponential is applied
κ - coupling parameter which modifies the coupled exponential function

d - dimension, should be equal to the dimension of x

In[∘]:=
```
ClearAll[CoupledLogarithm];
CoupledLogarithm[x_, κ_ : 0, d_ : 1] :=
 CoupledLogarithm[x, κ, d] =
  If[x ≥ 0,
   If[κ ≠ 0,
    1/κ (x^(κ/(1+d κ)) - 1),
    Log[x]
   ],
   Undefined
  ]
```

Coupled Gaussian - One Dimension

In[∘]:=
```
ClearAll[CoupledNormalDistribution];
CoupledNormalDistribution[μ_ : 0, σ_, κ_] :=
 CoupledNormalDistribution[μ, σ, κ] =
  Module[{x},
   ProbabilityDistribution[
    1/NormCG[σ, κ] (CoupledExponential[(x - μ)^2/σ^2, κ])^(-1/2),
    If[κ ≥ 0,
     {x, -∞, ∞},
     {x, μ - Sqrt[-σ^2/κ], μ + Sqrt[-σ^2/κ]}
    ]
   ]
  ]
```

Normalization of 1-D Coupled Gaussian

In[∘]:= NormCG[σ_, κ_] :=

$$\begin{cases} \sqrt{2\pi}\,\sigma & \kappa == 0 \\ \dfrac{\sqrt{\pi}\,\sigma\,\text{Gamma}\left[\frac{-1+\kappa}{2\kappa}\right]}{\sqrt{-\kappa}\,\text{Gamma}\left[1-\frac{1}{2\kappa}\right]} & \kappa < 0 \\ \dfrac{\sqrt{\pi}\,\sigma\,\text{Gamma}\left[\frac{1}{2\kappa}\right]}{\sqrt{\kappa}\,\text{Gamma}\left[\frac{1+\kappa}{2\kappa}\right]} & \text{True} \end{cases}$$

Coupled Exponential Distribution - One Dimension

The coupled exponential distribution is also known as the generalized Pareto distribution

```
In[∘]:= ClearAll[CoupledExponentialDistribution];
        CoupledExponentialDistribution[μ_ : 0, σ_, κ_] :=
         CoupledExponentialDistribution[μ, σ, κ] =
          Module[{x},
           ProbabilityDistribution[
            1/σ (CoupledExponential[(x - μ)/σ, κ, 1])^-1,
            {x, μ, If[κ ≥ 0, ∞, -σ/κ + μ]}
           ]
          ]
```

Multivariate Coupled Stretched Exponential Distribution

Status: Currently defined for $\alpha = 1$, $\kappa > 0$ and or α=2, κ>-1/d;
The compact support domain (-1/d < κ < 0) for α=2 is not fully verified due to the integration not completing.

```
In[•]:= ClearAll[MultivariateCoupledDistribution];
MultivariateCoupledDistribution[μ_, Σ_, κ_, α_] :=
 MultivariateCoupledDistribution[μ, Σ, κ, α] =
  Module[{dimMean, dimCor, x},
   (* Check that Σ is a positive
    definite matrix with dimensions equal to length of μ *)
   dimMean = Length[μ]; dimCor = Dimensions[Σ];
   If[dimCor ≠ dimMean,
    Message[MultivariateCoupledDistribution::argx, dimCor, dimMean]
   ];
   If[Not@PositiveDefiniteMatrixQ[Σ],
    Message[MultivariateCoupledDistribution::corr]
   ];
   If[α ≠ (1 || 2),
    Message[MultivariateCoupledDistribution::alpha, α]
   ];

   (* Define the distribution *)
   (* Evaluation of the input x to CoupledExponential is completed to insure
    that the definition of CoupledExponential can stay one dimensional *)
   x = Table[Symbol["$x" <> ToString@i], {i, dimMean}];
   ProbabilityDistribution[
```

$$\frac{\left(\text{CoupledExponential}\left[\left((x-\mu).\text{Inverse}[\Sigma].(x-\mu)\right)^{\frac{\alpha}{2}}, \kappa, \text{dimMean}\right]\right)^{\frac{-1}{\alpha}}}{\text{NormMultiCoupled}[\Sigma, \kappa, \alpha, \text{dimMean}]},$$

```
      (* Define the domain *)
      ##
     ] &@@
    Which[α == 1, Table[{x⟦i⟧, μ⟦i⟧, ∞}, {i, dimMean}],
     α == 2, Table[{x⟦i⟧, -∞, ∞}, {i, dimMean}]
    ]
  ]
MultivariateCoupledDistribution::argx =
  "Length of the first dimension of the matrix
    `1` does not match the length of the mean `2`.";
MultivariateCoupledDistribution::corr =
  "The correlation matrix Σ is not positive definite.";
MultivariateCoupledDistribution::alpha = "Alpha `1` is required to be 1 or 2";
```

Normalization of Multivariate Coupled Stretched Exponential

Normalization has only been completed for positive values of coupling

In[•]:=
```
ClearAll[NormMultiCoupled];
NormMultiCoupled[Σ_, κ_, α_, d_] :=
 NormMultiCoupled[Σ, κ, α, d] =
  Which[
   α == 1, Det[Σ ]^(1/2) / (1 + (-1 + d) κ),
   (*α==2, (√π Det[Σ ]^(1/2) Gamma[(1+(-1+d) κ)/(2 κ)]) / (√κ Gamma[(1+d κ)/(2 κ)])*)
   α == 2,
   Which[
    κ > 0, π^(d/2) Det[Σ ]^(1/2) ((1 + d κ)/κ^(d/2)) (Beta[1/(2 κ) + 1, d/2]/Gamma[d/2]),
    κ == 0, (2 π)^(d/2) Det[Σ ]^(1/2),
    -1/d < κ < 0, π^(d/2) Det[Σ ]^(1/2) (1/(-κ)^(d/2)) (Beta[(1+d κ)/(-2 κ) + 1, d/2]/Gamma[d/2])
   ]
  ]
```

In[•]:=
```
(*
The integral for arbitrary α does not simplify;
will need to define an explicit solution or approach
 Assuming[∞>κ>0&&∞>α>0,∫_0^∞ Refine[(CoupledExponential[(x/σ)^α,κ,d])^(-1/α)]dx]
*)
```

Coupled Probability

The Coupled Probability raises a distribution to the power
$1 - \kappa$Mult and then renormalizes the distribution. The input
The coupling term M is the multiplicative coupling. It is
 related to the source of coupling κ (small kappa) by the relationship

$$\kappa\text{Mult} = \frac{-\alpha\,\kappa}{1 + d\,\kappa}$$

where α is the power and d is the dimension of the random variable of the distribution

The function is currently written for continuous distributions,
though discrete distributions are also possible. f is required to be distribution, such that
 PDF[f, {x, x_{min}, x_{max}}] will produce a continuous distribution. The output is also a new distribution.

```
In[•]:= ClearAll[CoupledProbability];
        CoupledProbability[dist_, κMult_, x_, xmin_ : (-∞), xmax_ : ∞] :=
         CoupledProbability[dist, κMult, x, xmin, xmax] =
          If[κMult == 0, PDF[dist, x],
           FullSimplify[
                  (PDF[dist, x])^(1-κMult)
            ───────────────────────────────────
            ∫_xmin^xmax (PDF[dist, y])^(1-κMult) ⅆy
           ]
          ]
```

Coupled Entropy

Computes the coupled entropy using the coupled logarithm and coupled probability functions.
Currently limited to one-dimensional distributions
Updated Dec 17, 2020 to call the Coupled Cross-Entropy function

Inputs
p - probability distribution, limited to one-dimension
κ - coupling parameter
α - multiplicative term, default is 1
d - dimensions, default is 1

Output is the coupled entropy for a one-dimensional distribution

```
In[•]:= ClearAll[CoupledEntropy];
        CoupledEntropy[dist_, κ_, α_ : 1, d_ : 1, limits_ : {-∞, ∞}, root_ : False] :=
         CoupledEntropy[dist, κ, α, d, limits, root] =
          CoupledCrossEntropy[dist, dist, κ, α, d, limits, root]
```

Coupled Cross-Entropy

Computes the coupled cross-entropy using the coupled logarithm and coupled probability functions.
Currently limited to one-dimensional distributions

Inputs
p - probability distribution, limited to one-dimension
κ - coupling parameter
α - multiplicative term, default is 1
d - dimensions, default is 1

Output is the coupled entropy for a one-dimensional distribution

```
In[•]:= ClearAll[CoupledCrossEntropy];
CoupledCrossEntropy[distP_, distQ_,
   κ_, α_ : 1, d_ : 1, limits_ : {-∞, ∞}, root_ : False] :=
  CoupledCrossEntropy[distP, distQ, κ, α, d, limits, root] =
   If[! root,
    - ∫_{limits⟦1⟧}^{limits⟦2⟧} FullSimplify[CoupledProbability[distP, (-α κ)/(1 + d κ), x] (1/(-α))
          CoupledLogarithm[PDF[distQ, x]^-α, κ, d]] ⅆx // FullSimplify,
    NIntegrate[FullSimplify[CoupledProbability[distP, (-α κ)/(1 + d κ), x]
        CoupledLogarithm[PDF[distQ, x]^-α, κ, d]^(1/α)], {x, limits⟦1⟧, limits⟦2⟧}]
   ]
```

Coupled Divergence

Computes the coupled divergence using its relationship with Coupled Entropy and Coupled Cross-Entropy
Currently limited to one-dimensional distributions

The definition of the Coupled Divergence is notionally:
    Divergence = Cross-Entropy - Entropy
    - CoupledProbability[distP, ...] (CoupledLogarithm[distQ, ...] - CoupledLogarithm[distP, ...])
This is equivalent to:
    CoupledCrossEntropy[distP, distQ, ...] - CoupledEntropy[distP,...]
    note that for the first case the functional relationship for the generalization is the same as for $\kappa = 0$.
        And there is an alternative form in which the division of the densities is preserved but the sum is generalized

For the second def

Inputs
p - probability distribution, limited to one-dimension
κ - coupling parameter
α - multiplicative term, default is 1
d - dimensions, default is 1

Output is the coupled entropy for a one-dimensional distribution

```
In[•]:= ClearAll[CoupledDivergence];
        CoupledDivergence[distP_, distQ_,
           κ_, α_ : 1, d_ : 1, limits_ : {-∞, ∞}, root_ : False] :=
          CoupledDivergence[distP, distQ, κ, α, d, limits, root] =
           CoupledCrossEntropy[distP, distQ, κ, α, d, limits, root] -
            CoupledEntropy[distP, κ, α, d, limits, root]
```

Coupled Product

The Coupled Product is a generalization of the product function, which raises each input to a power and after combining the inputs, takes the root. It is defined based on the sum of arguments of the coupled exponential:

$$\prod_{i=1\otimes_{\alpha,\kappa,d_i}}^{N} x_i = \mathrm{Exp}_{\alpha,\kappa,D}\left[\sum_{i=1}^{N} \mathrm{Log}_{\alpha,\kappa,d_i}[x_i]\right];\quad D = \sum_{i=1}^{N} d_i$$
$$= \left(\sum_{i=1}^{N} x_i^{\frac{\alpha\kappa}{1+d_i\kappa}} - (N-1)\right)^{\frac{1+D\kappa}{\alpha\kappa}};\quad D = \sum_{i=1}^{N} d_i$$

Inputs
xList - list of arguments to be combined by the coupled product
κ - coupling parameter
α - multiplicative term, default is 2 which is associated with coupled Gaussian distributions
d - dimension, currently defined as a constant, but planned to also except of list the same dimensions as xList; version 5 modified the dimensions of the outer exponent to be d times Length[xList]. Thus the inputs are assumed to be the same dimension and the output dimension is sum of the input dimensions.

Output - coupled product of xList

Improvements Planned:
The dimension is defined as a constant. The dimension of each coupled logarithm should be controllable. The dimension of the coupled exponential should be the sum of the dimensions used for each coupled logarithm.

The CoupledProductPrototype is intended to allow the input of arguments to be combined match the structure of the built in *Mathematica* Product function. However; more extensive pattern matching is required to process the variety of inputs possible using this structure.

Old version: CoupledProductOld works but does not build upon the coupled exponential and coupled logarithm functions. It does compute the output dimension as a sum of the input dimensions.

```
In[∘]:= Clear[CoupledProduct];
        CoupledProduct[xList_, κ_, α_ : 2, d_ : 1] :=
          CoupledProduct[xList, κ, α, d] =
           CoupledExponential[Total[CoupledLogarithm[#^-α, κ, d] & /@ xList]^(-1/α),
            κ, d Length[xList]];
        (* Not used
           CoupledProductPrototype[f_,index_,κ_,α_:2,d_:1]:=
          CoupledExponential[Sum[CoupledLogarithm[f^-α,κ,d],index]^(-1/α),
           κ,α,d];
        *)
```

```
In[∘]:= (* Not used
          CoupledProductOld[xList_,κ_,α_,dList_]:=
         Module[{
            κMultList = Table[(-α κ)/(1+dList[[i]] κ),{i,Length[dList]}],
            κMultOutput = (-α κ)/(1+Sum[dList[[i]],{i,Length[dList]}] κ)
           },
           If[Length[dList]≠1,
            (Sum[(xList[[i]])^κMultList[[i]],{i,Length[xList]}]
                -(Length[xList]-1))^(1/κMultOutput),
            (Sum[(xList[[i]])^κMultList[[1]],{i,Length[xList]}]
                -(Length[xList]-1))^(1/κMultList[[1]])
           ]
          ]/;Length[xList]>1&&Length[dList]==Length[dList]
        *)
```

Coupled Sum

For two inputs the coupled sum is x + y + (-α κ) x y
For more than two inputs, the solution is determined using recursion. The last element of the list is extracted as one of the variables, and the other variable calls the coupled sum without the last element of the list.
The input is a list of variables to be operated on, the nonlinear coupling κ, and the power α affecting the variable x^{α}. The operator currently assumes each variable in the list is one-dimensional.

Improvement Planned
A simpler specification of the function uses the fold function. Use of alpha not equal to 1 still in evaluation. Earlier plan no longer necessary:
 CoupledSumPrototype will define the coupled sum in terms of the product of arguments of the coupled logarithm. The input is intended to follow the structure of the *Mathematica* Sum function, but requires additional pattern matching of input types to be fully functional.

In[∘]:=

```
In[∘]:= Clear[CoupledSum];
CoupledSum[xList_, κ_, α_] :=
 CoupledSum[xList, κ, α] =
  If[Length[xList] == 2,
    First[xList] + Last[xList] + (-α κ) First[xList] × Last[xList],
    (*Else when xList > 2*)
    CoupledSum[Drop[xList, -1], κ, α] +
     Last[xList] +
     (-α κ) CoupledSum[Drop[xList, -1], κ, α] × Last[xList]
   ] /; Length[xList] > 1 (* only execute if length > 1 *)

CoupledSumPrototype[list_, κ_, α_ : 1] :=
  Fold[(#1^α + #2^α + κ #1^α #2^α)^(1/α) &, list];
```

Coupled Sine and Cosine

```
In[∘]:= ClearAll[CoupledSin];
CoupledSin[x_, κ_, α_, d_] :=
 CoupledSin[x, κ, α, d] =
  (CoupledExponential[ⅈ x, κ, d]^(-1/α) -
      CoupledExponential[-ⅈ x, κ, d]^(-1/α)) / 2 ⅈ

In[∘]:= ClearAll[CoupledCos];
CoupledCos[x_, κ_, α_, d_] :=
 CoupledCos[x, κ, α, d] =
  (CoupledExponential[ⅈ x, κ, d]^(-1/α) +
     CoupledExponential[-ⅈ x, κ, α, d]^(-1/α)) / 2
```

Weighted Generalized Mean

Computes the weighted generalized mean of a list. For a 2-D list the weighted generalized mean of each column is computed. The weights are normalized, thus weights equal to one correspond to the generalized mean

$$Y = \left(\sum_{i=1}^{N} \frac{W_i}{\sum_{i=1}^{N} W_i} X_i^p\right)^{\frac{1}{p}}$$

Inputs:
X - a one or two dimensional list of variables
W - scalar, vector, or matrix of weights which modify the contribution of each input X_i
p - scalar, power for the generalized mean

p = -1 is the harmonic mean
p = 0 is the geometric mean
p = 1 is the arithmetic mean
p = 2 is the root mean square

Outputs:
Y - scalar or vector of weighted generalized mean; for 2-D input mean of the first dimension is computed

```
In[•]:= ClearAll[WeightedGeneralizedMean];
WeightedGeneralizedMean[X_, p_, W_ : 1] :=
 WeightedGeneralizedMean[X, p, W] =
  Module[{n, m, i, j, normW, sumW, XArrayDepth, WArrayDepth, Y},

   XArrayDepth = ArrayDepth[X];
   Which[
    XArrayDepth == 1, {n} = Dimensions[X],
    XArrayDepth == 2, {n, m} = Dimensions[X],
    XArrayDepth > 2, WeightedGeneralizedMean::argx =
     "Warning: Inputs X expected to have ArrayDepth of 2"
   ];

   If[Length[p] > 0, WeightedGeneralizedMean::argx =
     "Warning: Input p expected to be a scalar"
   ];
   Assert[True];
   (* Check dimensions of W and initialize normW
    expand scalar to matrix size of X
    expand column vector into n rows
    or set message warning about mismatch in size*)
   WArrayDepth = ArrayDepth[W];
   If[XArrayDepth == 1,
    Which[WArrayDepth == 0, normW = Table[W, n],
     WArrayDepth == 1, normW = W,
     WArrayDepth > 1, WeightedGeneralizedMean::argx =
      "Warning: Input W expected to have same or less depth than X"
    ],
    (*Else if XArrayDepth == 2 *)
    Which[WArrayDepth == 0, normW = Table[W, {n}, {m}],
      WArrayDepth == 1, normW = Table[W, m] // Transpose,
      WArrayDepth == 2, normW = W,
      WArrayDepth > 2, WeightedGeneralizedMean::argx =
       "Warning: Input W expected to have ArrayDepth of 1 or 2"
     ];
];
If[Dimensions[normW] ≠ Dimensions[X], WeightedGeneralizedMean::argx =
  "Dimensions of X and resized W do not match"];

If[XArrayDepth == 1, Module[{},
  Assert[True];
  sumW = $\sum_{i=1}^{n} normW_{[[i]]}$;
  normW = $\frac{normW}{sumW}$;
  Quiet[If[p ≠ 0,
    Check[Y = $\left(\sum_{i=1}^{n} normW_{[[i]]} X_{[[i]]}^{p}\right)^{\frac{1}{p}}$ // N,
     Y = If[p > 0, Max[X], Min[X]]
    ],

    Y = $\prod_{i=1}^{n} X_{[[i]]}^{normW_{[[i]]}}$ // N
   ]]
 ],
 Module[{},
  Y = Table[0, m];

  (* Loop column and rows to compute weighted generalized mean *)
  For[j = 1, j ≤ m, j++,
   sumW = $\sum_{i=1}^{n} normW_{[[i,j]]}$;
   normW[[All,j]] = $\frac{normW_{[[All,j]]}}{sumW}$;
   Quiet[If[p ≠ 0,
     Check[Y[[j]] = $\left(\sum_{i=1}^{n} normW_{[[i,j]]} X_{[[i,j]]}^{p}\right)^{\frac{1}{p}}$ // N,
      Y[[j]] = If[p > 0, Max[X[[All,j]]], Min[X[[All,j]]]]
     ],
     Y[[j]] = $\prod_{i=1}^{n} X_{[[i,j]]}^{normW_{[[i,j]]}}$ // N
    ]]
  ]
 ]
];
(*Print[sumW];*)
Assert[True];
```

```
    N[Y, 5] (* Set output of module *)
    (*{normW,XArrayDepth,WArrayDepth} *) (* Monitor Variables *)
   ];
```

Variable transformations

risk and coupling Domain Dual

```
In[•]:= riskDomainDual[r_, α_ : 2, d_ : 1] := riskDomainDual[r, α, d] =
         -α r
        ------- ;
        α + d r
```

```
In[•]:= couplingDomainDual[κ_, α_ : 2, d_ : 1] := couplingDomainDual[κ, α, d] =
          - κ
        ------- ;
        1 + d κ
```

Coupling to Risk and Risk to Coupling

```
In[•]:= couplingToRisk[κ_, α_ : 2, d_ : 1] := couplingToRisk[κ, α, d] =
         -α κ
        ------- ;
        1 + d κ
```

```
In[•]:= riskToCoupling[r_, α_ : 2, d_ : 1] := riskToCoupling[r, α, d] =
           -r
        -------- ;
        d r  + α
```

RiskQDual

```
In[•]:= riskQDual[rORq_] := 1 - rORq;
```

qToCoupling and couplingToq

```
In[•]:= couplingToq[κ_, α_ : 2, d_ : 1] := couplingToq[κ, α, d] =
          1 - couplingToRisk[κ, α, d];
```

```
In[•]:= qToCoupling[q_, α_ : 2, d_ : 1] := qToCoupling[q, α, d] =
          - (1 - q)
        -------------
        α + d (1 - q)
```

Coupled Box-Müller Method

The coupled Box-Muller algorithm than has the following procedure.

1) Draw two uniform random variables U1 and U2 over the range 0 to 1.
2) Apply the generalized transformation $(Z_1, Z_2) = T(U_1, U_2)$ where T is

$$Z_1 = \sqrt{\ln_\kappa(U_1^{-2})}\ \cos(2\,\pi\,U_2)$$

$$Z_2 = \sqrt{\ln_\kappa(U_2^{-2})}\ \sin(2\,\pi\,U_1).$$

In this case the dimension of the logarithm is zero and thus the variable U does not need to be raised to a power.

3) (Z_1, Z_2) forms a two-dimensional coupled Gaussian distribution with $\mu_1 = \mu_2 = 0$ and

$\Sigma^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. For a set of independent random variables only one of the two variates should be selected, since although uncorrelated (no cross terms in the correlation matrix) the variates are dependent since the joint distribution cannot be factored.

4) The location and scale for the coupled Gaussian variates are added to Z by $X = \mu + \sigma Z$.

In[•]:=

```
Clear[CoupledVariate];
CoupledVariate[μ_ : 0, σ_ : 1, κ_, n_ : 1] := Module[
   {UniformVariates = RandomReal[{0, 1}, 2 n], CoupledNormalVariates},
   CoupledNormalVariates =
    If[n == 1,
     √CoupledLogarithm[UniformVariates[[1]]^-2, κ, 0] Cos[2 π UniformVariates[[-1]]],
     Table[
      √CoupledLogarithm[UniformVariates[[i]]^-2, κ, 0] Cos[2 π UniformVariates[[-i]]],
      {i, n}
     ]
    ];
   μ + σ CoupledNormalVariates
  ];
```