Let $G(t)$ be a Gaussian process with mean 0 and kernel $k(t, t')$.

$$G(t) \sim \mathcal{GP}\big(0, k(t, t')\big) \tag{1}$$

We first discuss how to generate a sample path of $G(t)$.

1 Generating a sample path of $G(t)$

Consider a user-specified time grid:

$$\big\{t_j,\ j = 1, 2, \ldots, N\big\} \tag{2}$$

The time grid does not need to be uniform. Let

$$\mathbf{X} \equiv \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{pmatrix}, \qquad X_j \equiv G(t_j) \tag{3}$$

In our theoretical discussion, we put $\mathbf{X}$ as a column vector of N components. $\mathbf{X}$ has the multivariate normal distribution with mean 0 and covariance matrix K (an $N \times N$ positive definite matrix).

$$\mathbf{X} \sim \mathcal{N}(0,\, K), \quad K \equiv \{K_{i,j}\}, \quad K_{i,j} \equiv k(t_i, t_j) \tag{4}$$

1.1 Mathematical formulation

We carry out eigenvalue decomposition of matrix K.

$$K = QDQ^T, \qquad Q = (\mathbf{Q}_1\ \mathbf{Q}_2\ \cdots\ \mathbf{Q}_N) \tag{5}$$

where D is a diagonal matrix containing eigenvalues $\{d_j\}$ and Q an orthogonal matrix whose columns are eigenvectors $\{\mathbf{Q}_j\}$.

$$D = \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_N \end{pmatrix} \equiv \text{diag}(d_1, d_2, \ldots, d_N), \qquad K\mathbf{Q}_j = d_j\mathbf{Q}_j$$

Since K is positive definite, all eigenvalues are strictly positive in exact arithmetic (we will discuss how to deal with the effect of round-off errors in finite precision). We introduce

$$D^{\frac{-1}{2}} \equiv \text{diag}(d_1^{\frac{-1}{2}}, d_2^{\frac{-1}{2}}, \ldots, d_N^{\frac{-1}{2}})$$

Let $\mathbf{Z} \equiv (D^{\frac{-1}{2}}Q^T)\mathbf{X}$. It can be shown that $\mathbf{Z}$ has the multivariate normal distribution with mean 0 and covariance matrix $(D^{\frac{-1}{2}}Q^T)K(D^{\frac{-1}{2}}Q^T)^T = D^{\frac{-1}{2}}(Q^TKQ)D^{\frac{-1}{2}} = I$

$$\mathbf{Z} \sim \mathcal{N}(0,\, I) \tag{6}$$

That is, vector $\mathbf{Z}$ contains $i.i.d.$ standard normal random variables. To generate a sample of $\mathbf{X}$, we first generate a sample of $\mathbf{Z}$. Then we write $\mathbf{X}$ in terms of eigenvectors $\{\mathbf{Q}_j\}$ and eigenvalues $\{d_j\}$.

$$\mathbf{X} = QD^{\frac{1}{2}}\mathbf{Z} = \sum_{j=1}^{N} \underbrace{\mathbf{Q}_j}_{\text{vector}} \underbrace{\left(d_j^{\frac{1}{2}} Z_j\right)}_{\text{scalar}} \tag{7}$$

$\mathbf{X}$ is a linear combination of eigenvectors. The coefficient of $\mathbf{Q}_j$ is small for small eigenvalue d_j.

1.2 Numerical implementation in finite precision

In finite precision, due to round-off errors, very small eigenvalues suffer from loss of accuracy; some of them even become negative in computation. In numerical implementation, we keep only eigenvalues that are above a tolerance tol (for example, $tol = 10^{-12}$ for double precision). Assume that the eigenvalues in D are already sorted in ascending order. Let n_c be the smallest index such that $d_{n_c} \geq tol$.

$$n_c = \min_{d_j \geq tol} j \tag{8}$$

We neglect eigenvalues $\{d_j, 1 \leq j \leq (n_c - 1)\}$ and keep only $\{d_j, n_c \leq j \leq N\}$. We write $\mathbf{X}$ as

$$\mathbf{X} = \sum_{j=n_c}^{N} \underbrace{\mathbf{Q}_j}_{\text{vector}} \underbrace{\left(d_j^{\frac{1}{2}} Z_j\right)}_{\text{scalar}} \tag{9}$$

Notice that in numerical computation with finite precision, the number of degrees of freedom in sample path $\mathbf{X}$ is effectively reduced from N to $(N - n_c)$. For the squared exponential kernel, $(N - n_c)$ is only a small fraction of N.

2 Extending a sample path X by one time step

Consider a user-specified time grid given in (2). The time grid does not need to be uniform. Suppose a sample path of $G(t)$ is already given on this time grid. We now extend the time grid by one step and sample $G(t)$ at t_{N+1}. Let $\mathbf{X}$ in (3) denote the given sample path and let $X_* \equiv G(t_{N+1})$. Random variable $\begin{pmatrix} \mathbf{X} \\ X_* \end{pmatrix}$ has the multivariate normal distribution with mean 0 and covariance matrix $K^{(\text{ext})}$ (an $(N+1) \times (N+1)$ positive definite matrix).

$$\mathbf{X} \sim \mathcal{N}(0,\, K^{(\text{ext})}), \quad K^{(\text{ext})} \equiv \{K^{(\text{ext})}_{i,j}\}, \quad K^{(\text{ext})}_{i,j} \equiv k(t_i, t_j) \tag{10}$$

We write the covariance matrix $K^{(\text{ext})}$ as

$$K^{(\text{ext})} = \begin{pmatrix} K & K_* \\ K_*^T & K_{**} \end{pmatrix} \tag{11}$$

$$K_{i,j} = k(t_i, t_j), \quad 1 \le (i,j) \le N$$
$$(K_*)_i = k(t_i, t_{N+1}), \quad 1 \le i \le N$$
$$K_{**} = k(t_{N+1}, t_{N+1})$$

where K is $N \times N$, K_* is $N \times 1$, and K_{**} is 1×1. The conditional distribution of X_* given $\mathbf{X}$ is

$$(X_* \big| \mathbf{X}) \sim \mathcal{N}(\mu,\, d) \tag{12}$$
$$\mu = K_*^T K^{-1} \mathbf{X}, \quad d = K_{**} - K_*^T K^{-1} K_*$$

In general, $(X_* \big| \mathbf{X})$ has a normal distribution with a mean and a variance depending on the time grid and on the sample path. We discuss the cases of two kernels separately.

2.1 Exponential kernel

The exponential kernel has the expression:

$$K(t_i, t_j) = \exp(-\frac{|t_i - t_j|}{\tau}) \tag{13}$$

where τ is the given time scale. It can be shown that the Gaussian process corresponding to the exponential kernel is a Markov process (an Ornstein-Uhlenbeck process). The conditional distribution $(X_* \big| \mathbf{X})$ depends only on X_N; it is not affected by $(X_1, X_2, \ldots, X_{N-1})$. It has the expression below.

$$(X_* \big| \mathbf{X}) \sim \mathcal{N}(\mu,\, d) \tag{14}$$
$$\mu = X_N \exp\Big(\frac{-(t_{N+1} - t_N)}{\tau}\Big), \quad d = 1 - \exp\Big(\frac{-2(t_{N+1} - t_N)}{\tau}\Big)$$

An Ornstein-Uhlenbeck process $G(t)$ is only continuous in time t, but is not differentiable.

2.2 Squared exponential kernel

The squared exponential kernel (also called the RBF kernel in machine learning) has the expression:

$$K(t_i, t_j) = \exp(-\frac{|t_i - t_j|^2}{2\tau^2}) \tag{15}$$

$G(t)$ generated from the squared exponential kernel is differentiable in time t. We use the general formula (12) to generate $(X_* \big| \mathbf{X})$. We need to work with K^{-1}, which, for the squared exponential kernel, is practically singular in finite precision. We carry out eigenvalue decomposition of matrix K as shown in equation

(5). Based on the decomposition $K = QDQ^T$, we write K^{-1} as

$$K^{-1} = QD^{-1}Q^T \tag{16}$$

In numerical eigenvalue decomposition, small eigenvalues have low or no accuracy. In matrix inversion, small inaccurate eigenvalues will be catastrophic. We need to be more aggressive in weeding out small eigenvalues. We set a tolerance $tol = 10^{-8}$ or larger. We find the pseudo inverse based on eigenvalues that are above tol. Let n_c be the smallest index such that $d_{n_c} \geq tol$ defined in (8). We neglect the subspace of eigenvectors associated with eigenvalues below tol. We write out the pseudo inverse of K for the subspace of eigenvectors associated with eigenvalues above tol.

$$\begin{aligned} K^\dagger &= QD^\dagger Q^T, \qquad D^\dagger = \text{diag}(0, \ldots, 0, d_{n_c}^{-1}, d_{(n_c+1)}^{-1}, \ldots, d_N^{-1}) \\ &= \tilde{Q}\tilde{D}^{-1}\tilde{Q}^T \end{aligned} \tag{17}$$

$$\tilde{Q} \equiv \underbrace{\left(\mathbf{Q}_{n_c}\ \mathbf{Q}_{(n_c+1)}\ \cdots\ \mathbf{Q}_N\right)}_{\text{an } N\times(N-n_c) \text{ matrix}},$$

$$\tilde{D} = \underbrace{\text{diag}(d_{n_c}, d_{(n_c+1)}, \ldots, d_N)}_{\text{an } (N-n_c)\times(N-n_c) \text{ diagonal matrix}}$$

The geometric meaning of pseudo inverse is demonstrated in the property below

$$(K^\dagger K)\mathbf{U} = (KK^\dagger)\mathbf{U} = \begin{cases} \mathbf{U}, & \text{for } \mathbf{U} \in \text{span}\left\{\mathbf{Q}_{n_c}, \mathbf{Q}_{(n_c+1)}, \cdots, \mathbf{Q}_N\right\} \\ 0, & \text{for } \mathbf{U} \in \text{span}\left\{\mathbf{Q}_1, \cdots, \mathbf{Q}_{(n_c-1)}\right\} \end{cases} \tag{18}$$

With the numerical pseudo inverse (17), we compute the mean and the variance given in (12).

$$(X_*|\mathbf{X}) \sim \mathcal{N}(\mu,\, d)$$

$$\mu = K_*^T K^{-1}\mathbf{X} = (\tilde{Q}^T K_*)^T \tilde{D}^{-1}(\tilde{Q}^T\mathbf{X}) \tag{19}$$

$$d = K_{**} - K_*^T K^{-1} K_* = K_{**} - (\tilde{Q}^T K_*)^T \tilde{D}^{-1}(\tilde{Q}^T K_*) \tag{20}$$

where both $(\tilde{Q}^T K_*)$ and $(\tilde{Q}^T\mathbf{X})$ are column vector of size $(N - n_c)$.

Remark: For a time grid with non-uniform time steps, matrices K, K_*, $\tilde{Q}$, and $\tilde{D}$ need to be recalculated at every time step.