Réduction de modèle

Raphaël Granger et Qinyan Yang

Mars 2025

---

1 Variables aléatoires du problème

Considérons les variables aléatoires suivantes :

$$\begin{cases} r_w \sim \mathcal{N}(0.10, 0.015) \\ r_{iw} \sim \mathcal{N}(0.05, 0.01) \\ r \sim \text{Log-}\mathcal{N}(7.71, 1.0056) \\ T_u \sim \mathcal{U}([63100, 116000]) \\ H_u \sim \mathcal{U}([1000, 1100]) \\ T_{um} \sim \mathcal{U}([6310, 11600]) \\ H_{um} \sim \mathcal{U}([900, 1000]) \\ T_{lm} \sim \mathcal{U}([631, 1160]) \\ H_{lm} \sim \mathcal{U}([800, 900]) \\ T_l \sim \mathcal{U}([63.1, 116]) \\ H_l \sim \mathcal{U}([700, 800]) \\ L \sim \mathcal{U}([1120, 1680]) \\ K_w \sim \mathcal{U}([3000, 12000]). \end{cases}$$

1.1 Loi uniforme

La valeur minimale et maximale d'une loi normale correspondent à l'intervalle définition de la variable aléatoire, et la moyenne correspond au milieu de l'intervalle.

1.2 Loi normale

Pour $X \sim \mathcal{N}(\mu, \sigma^2)$, nous avons la propriété

$$\mathbb{P}\left(\mu - 3\sigma \leq X \leq \mu + 3\sigma\right) \simeq 0,9973.$$

Ainsi pour une borne convenable, on considère $\mu - 3\sigma$ comme valeur minimum et $\mu - 3\sigma$ pour valeur maximale. La valeur moyenne est donnée par μ.

1.3 Loi lognormale

Par définition d'une loi lognormale, $X \sim \text{Log-}\mathcal{N}(\mu, \sigma^2)$ si $Y = \ln(X) \sim \mathcal{N}(\mu, \sigma^2)$.
Par analogie avec la propriété ci-dessus, on a :

$$\mathbb{P}\left(e^{\mu-3\sigma} \leq X \leq e^{\mu+3\sigma}\right) \simeq 0,9973.$$

Pour un intervalle convenable, on considère l'intervalle de confiance $\left[e^{\mu-3\sigma}, e^{\mu+3\sigma}\right]$.
La moyenne est donnée par $\mathbb{E}\left[e^{\mu+\sigma^2/2}\right]$.

Gardons ces valeurs dans des vecteurs.

```
#Valeurs minimales
val_min <- c(0.1 - 3.0*0.015, 0.05 - 3.0*0.01, exp(7.71 - 3.0*1.0056),
             63100, 1000, 6310, 900, 631, 800, 63.1, 700, 1120, 3000)

#Valeurs maximales
val_max <- c(0.1 + 3.0*0.015, 0.05 + 3.0*0.01, exp(7.71 + 3.0*1.0056),
             116000, 1100, 11600, 1000, 1160, 900, 116, 800, 1680, 12000)

#Valeurs moyennes
val_moy <- c(0.1, 0.05, exp(7.71 + (1.0056^2) / 2),
              mean(c(63100, 116000)), mean(c(1000, 1100)),
              mean(c(6310, 11600)), mean(c(900, 1000)),
              mean(c(631, 1160)), mean(c(800, 900)),
              mean(c(63.1, 116)), mean(c(700, 800)),
              mean(c(1120, 1680)), mean(c(3000, 12000)))
```

2 Propagation d'incertitudes par Monte Carlo

a.

On génère dans un premier temps un échantillon de 1000 grâce à la fonction `EchantBorehole` et appliquons à cet échantillon la fonction `borehole` afin d'obtenir les résultats de sortie.

```
echantillon <- EchantBorehole(1000)
resultats <- apply(echantillon, 1, function(x) borehole(as.numeric(x)))

cat("Moyenne : ",mean(resultats))
cat("Variance : ",var(resultats))
hist(resultats, main="Histogramme du modèle Borehole", xlab="Valeur de sortie")
```

On obtient son histogramme.

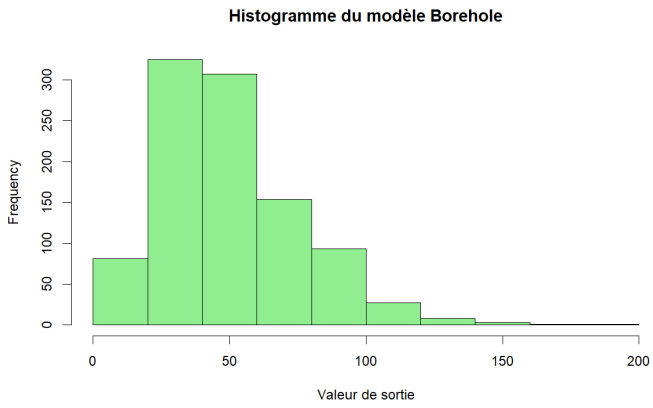

Figure 1: Histogramme du modèle Borehole sur un échantillon de taille 1000.

b.

Regardons si $\mathbb{E}[f(X)] = f(\mathbb{E}(X))$ où f est la fonction `Borehole`. Comparons donc la moyenne de la fonction sur l'échantillon `mean(resultats)` et la fonction évaluée en la moyenne des entrées `borehole(val_moy)`. Avec notre échantillon tirée, la différence est de 1.857, cette différence significative est génèré par non-linéarité de la fonction.

```
moyEchant <- mean(resultats)
borehole_moyenne <- borehole(val_moy)
diff <- abs(moyEchant - borehole_moyenne)
```

c.

Calculons le quantile d'ordre 95% de la sortie.

```
q95 <- quantile(resultats, 0.95)
```

Il en resulte que 95% des valeurs de sortie sont inférieures ou égales à `q95`.

Construisons une fonction `monte_carlo_inter` qui répète l'estimation de Monte Carlo afin de récupérer les quantiles à 2,5% et 97,5% des sorties du modèle de Borehole et obtenir un intervalle de confiance à 95%.

```
monte_carlo_inter <- function(N, iter = 10000) {
    resultats_mc_global <- numeric(iter)

    for (i in 1:iter) {
        #Nouveau échantillon
        echantillon_mc <- EchantBorehole(N)
```

```
        resultats_mc <- apply(echantillon_mc, 1, function(x) borehole(as.numeric(x)))

    resultats_mc_global[i] <- quantile(resultats_MC, 0.95)
  }

  #Calcul les quantiles à 2.5% et 97.5%
  inter_conf <- quantile(resultats_mc_global, c(0.025, 0.975))
  return(inter_conf)
}
inter_conf_95 <- monte_carlo_inter(1000, iter = 1000)
```

Les valeurs de `inter_conf_95` donnent l'intervalle de confiance à 95%. Dans notre cas, `q95` $= 100.1238$, et `inter_conf_95` $= [96.70954\,;\,107.2251]$.

d.

La fonction suivante nous donne la probabilité que le débit soit supérieur à $250\,m^3/an$. Initialisons par la même occasion l'erreur qu'on ne souhaite pas dépasser, et la taille initiale de notre échantillon.

```
debit_sup_250 <- function(N) {
  echantillon <- EchantBorehole(N)
  res <- apply(echantillon, 1, function(x) borehole(as.numeric(x)))
  return(mean(res > 250))
}
taille_init <- 2 * 10^6
taille_max <- 2 * 10^7 #arbitraire
incr <- 2 * 10^6 #pas
err_target <- 0.10
```

Il suffit ensuite d'incrémenter la taille de l'échantillon

```
probas <- c()
for (taille in seq(taille_init, taille_max, by = incr)) {
  probas <- c(probas, debit_sup_250(taille))
  if (length(probas) > 1) {
    err_relative <- abs(probas[length(probas)] -
            probas[length(probas) - 1])/ probas[length(probas) - 1]
    if (err_relative < err_target) {
      cat("Erreur relative : ", err_relative,"\n")
      cat("Taille nécéssaire : ", taille, "\n")
      break
    }
  }
}
```

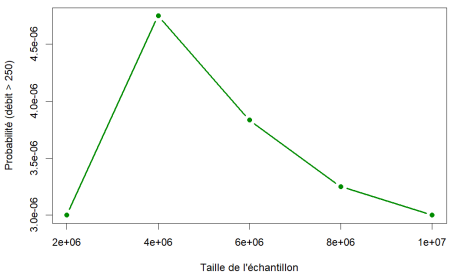

Figure 2: $\mathbb{P}(\text{débit} > 250\,m^3/an)$ selon la taille de l'échantillon.

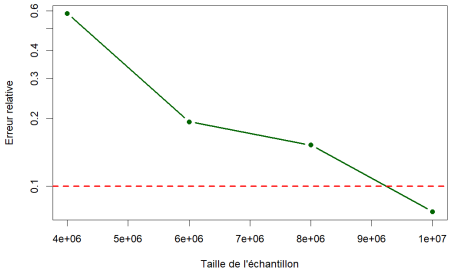

Figure 3: Erreur relative en fonction de l'échantillon.

Lorsque l'échantillon atteint une taille de 10^7, l'erreur relative est bien en dessous du seuil toléré. La taille de l'échantillon nécéssaire pour ce seuil de tolérance est de 10^7.

3 Analyse de sensibilité

a. Méthode de Morris

Nous utilisons la fonction `morris()` du package 'sensitivity' pour effectuer une analyse de sensibilité de Morris. Cette fonction évalue efficacement l'impact de chaque variable d'entrée sur la sortie du modèle, tout en maîtrisant le nombre total d'évaluations de la fonction, comme exigé par les contraintes de coût du problème. Nous définissons la fonction du modèle comme `borehole()`, et les noms des variables d'entrée comme `colnames(EchantBorehole(1))`. Nous réglons le paramètre de répétition `r` à 10, ce qui implique que les variations de chaque variable d'entrée seront évaluées 10 fois.

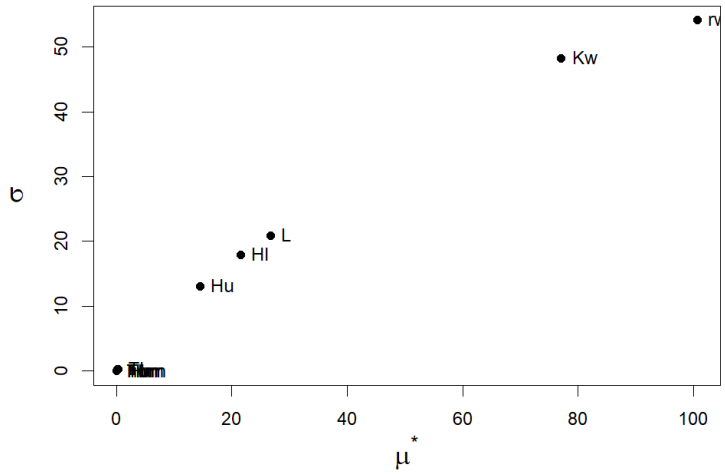

Figure 4: Analyse de sensibilité de Morris

L'analyse de sensibilité de Morris nous permet de comprendre la portée des variables d'entrée en fonction de leur impact sur la sortie. L'axe des abscisses modélise l'effet d'une variable sur la sortie, alors que celle des ordonnées indique l'intéraction avec les autres variables. Toutes les variables ont un comportement liénaire sauf Kw et rw qui ont donc des fortes intéractions avec les autres variables et impactent grandement la sortie.

Au total, l'analyse de sensibilité de Morris ne nécessite que 56 évaluations de la fonction, loin de la limite de 100 exigée par le problème. Cela illustre la capacité de la méthode de Morris à évaluer les impacts des variables tout en contrôlant efficacement les coûts de calcul. Les variables à fixer sont non influentes, qui sont riw, r, Tu, Tum, Hum, Tlm, Hlm, Tl.

b. Indices basés sur la corrélation/régression

i.

À partir d'un échantillon Monte Carlo de taille 100 on calcule le débit d'eau. Nous avons tracé les scatterplots suivants:

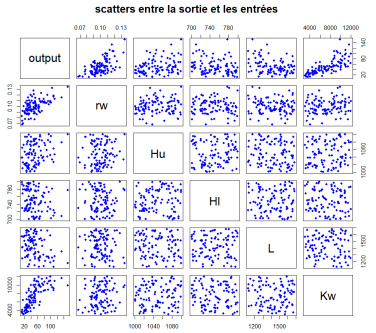

Figure 5: Scatterplots entre la sortie et les entrées sur un échantillon de 100.

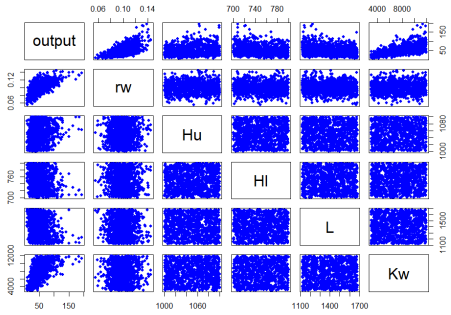

Figure 6: Scatterplots entre la sortie et les entrées sur un échantillon de 1000.

C'est plus visible avec un échantillon de 1000.

Les points impliquant les variables `Hu`, `Hl`, `L` sont dispersées de manières aléatoire, il n'y a pas de dépendance apparente entre la sortie et ces variables. Cependant, les variables `rw` et `Kw` présentent des dépendances.

- Pour la variable `rw` : Le graphique de dispersion indique une relation linéaire évidente entre `rw` et le débit. Lorsque `rw` augmente, le débit diminue de manière marquée, ce qui identifie rw comme un facteur clé influençant le débit d'eau.
- Pour `Kw` : Une corrélation positive est illustrée entre `Kw` et le débit. Une augmentation de Kw entraîne une augmentation significative du débit, soulignant l'importance de `Kw` sur le débit.

ii.

Nous avons défini un modèle de régression linéaire `lm_model`.

```
lm_model <- lm(output_filt ~ ., data = data.frame(input_filt))
```

Cela permet de récupérer les valeurs des coefficients de régression normalisées (SRC^2) :

rw	**Hu**	**Hl**	**L**	**Kw**
0.508	0.353	0.06	0.041	0.038

`rw` et `Hu` sont les paramètres avec le plus d'impact qui nécéssite un meilleur contrôle afin optimiser le débit d'eau. Leur impact significatif s'explique par leur rôle direct dans les termes logarithmiques et linéaires de la fonction `borehole()`.

Les autres variables (`Hl`, **L**, `Kw`) peuvent être fixées à leurs valeurs moyennes dans les analyses ultérieures pour simplifier le modèle sans perte majeure de précision.

L'analyse de Sobol est qualitative et permet de quantifier les effets non linéaires et les intéractions entre varaibles, ici `rw` et `Hu`.

c. Indices de Sobol

En utilisant la fonction `borehole()` comme le modèle, la taille d'échantillon de 2000, et 60 échantillons de bootstrap pour faire l'analyse de Sobol:

```
sobol_ind <- sobol2007(
  model = borehole,
  X1 = EchantBorehole(2000),
  X2 = EchantBorehole(2000),
  nboot = 60)
plot(sobol_ind, main = "Sobol Indices")
```

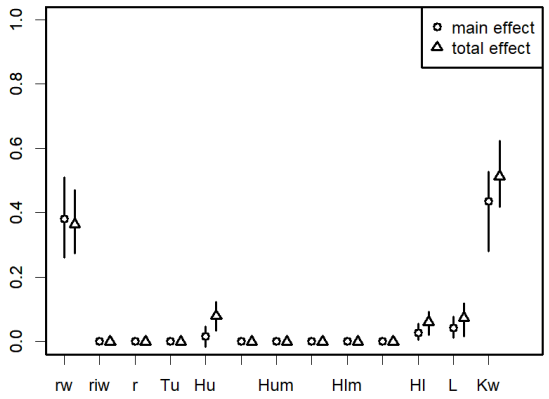

Figure 7: Indices de Sobol des variables.

Cette figure reflète les effets principaux et les effets totaux de chaque variable, révélant ainsi que `rw` et **Kw** sont les paramètres déterminants du modèle (valeur d'effet > 0.3). Leur indice total sont aussi conséquentes, indiquant que ces deux variables intéragissent avec les autres. Les autres variables ont une faible influence sur le modèle.