ACT-2002

Méthodes numériques en actuariat

Université Laval

ACT-2002

Méthodes numériques en actuariat

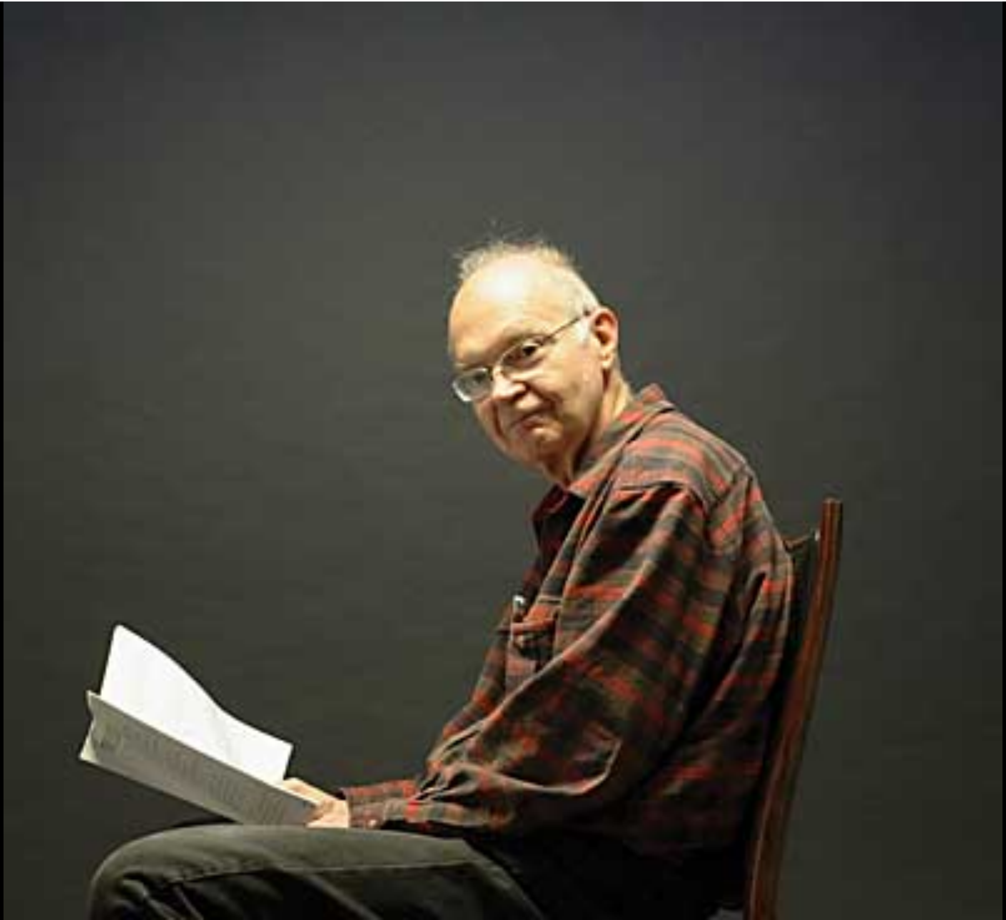
Vincent Goulet

Professeur titulaire
École d'actuariat, Université Laval

Version 2019.04

Sommaire

Simulation stochastique



Ce qu’en dit Dieu le Père (Knuth)

The moral of this story is that ***random numbers should not be generated with a method chosen at random****. Some theory should be used.*

Caractéristiques recherchées d’un générateur

1. Nombres distribués approximativement uniformément
2. Nombres approximativement indépendants
3. Période suffisamment longue
4. Facile à reproduire, mais impossible à prédire

Générateurs congruentiels linéaires

- Suite $x_1, x_2, \ldots$ obtenue à partir de x_0, puis

$$x_i = (ax_{i-1} + c) \bmod m, \quad i = 1, 2, \ldots,$$

  où $0 \leq x_i < m$
- Nombres sur $(0, 1)$ avec

$$u_i = \frac{x_i}{m}$$

Générateur RANDU

$$x_i = 65\,539 x_{i-1} \bmod 2^{31}, \quad i = 1, 2, \dots,$$

`revision-simulation.R`

Nombres aléatoires non uniformes

- Générer $u_1, \ldots, u_n$ d'une $U(0, 1)$
- Transformer vers nombres issus d'une loi F_X :

$$x_i = h(u_i)$$

- Transformation de variables aléatoires

Méthode de l'inverse

- Résultat clé :

$$F_X(X) \sim U(0,1)$$
$$\Updownarrow$$
$$F_X^{-1}(U) \sim F_X$$

- F_X^{-1} explicite pour quelques lois seulement
- On peut aussi résoudre numériquement

$$F_X(x) - u = 0$$

Fonctions de simulation de R

- Lois connues

```
> rgamma(3, shape = 1:3, rate = c(0.1, 2, 5))
[1] 1.5514136 1.7853782 0.9833313
```

Fonctions de simulation de R

- Lois connues

```
> rgamma(3, shape = 1:3, rate = c(0.1, 2, 5))
[1] 1.5514136 1.7853782 0.9833313
```

Gamma(1, 0.1)

Fonctions de simulation de R

- Lois connues

```
> rgamma(3, shape = 1:3, rate = c(0.1, 2, 5))
[1] 1.5514136 1.7853782 0.9833313
```

Gamma(2, 2)

Fonctions de simulation de R

- Lois connues

```
> rgamma(3, shape = 1:3, rate = c(0.1, 2, 5))
[1] 1.5514136 1.7853782 0.9833313
```

Gamma(3, 5)

Fonctions de simulation de R

- Lois connues

```
> rgamma(3, shape = 1:3, rate = c(0.1, 2, 5))
[1] 1.5514136 1.7853782 0.9833313
```

- Loi discrète quelconque

```
> sample(c(1, 3, 7, 8, 12),
+        prob = c(0.2, 0.1, 0.5, 0.05, 0.15),
+        size = 5, replace = TRUE)
[1] 1 1 7 7 7
```

Modèles actuariels — Mélanges

- Mélange discret

$$F(x) = p_1 F_1(x) + \cdots + p_n F_n(x)$$

```
> actuar::rmixture(5, probs = c(3, 5, 2),
+                  models = expression(rexp(2),
+                                      rgamma(3, 2),
+                                      rpareto(2, 3)))
[1] 0.2698414 0.4782837 2.7308532 1.5771954 2.8034498
```

- Mélange continu

$$F(x) = \int_{-\infty}^{\infty} F(x|\theta) u(\theta)\, d\theta$$

```
> rnbinom(5, size = 10, prob = rbeta(5, 2, 3))
[1] 16 29 13 16 26
```

- Distribution de la somme aléatoire

$$S = X_1 + X_2 + \cdots + X_N$$

```
> actuar::rcomppois(5, lambda = 2,
+                     rgamma(shape = 2.5, rate = 1.2))
[1]  3.758712 10.755732  2.687419  2.654180  2.387258
```

```
> actuar::rcompound(5, rnbinom(size = 2, prob = 0.8),
+                     rgamma(shape = 2.5, rate = 1.2))
[1] 0.0000000 0.8611348 0.0000000 3.7833257 0.0000000
```

Planification d'une étude de simulation

- Nous savons que
$$E[\bar{X}] = \mu$$
- Comment vérifier ce résultat par simulation ?

Planification d'une étude de simulation

- Nous savons que
$$E[\bar{X}] = \mu$$
- Comment vérifier ce résultat par simulation ?
    1. générer un échantillon $x_1, \ldots, x_n$ de la loi de X
    2. calculer $\bar{x}$
    3. répéter N fois $\rightarrow \bar{x}_1, \ldots \bar{x}_N$
    4. vérifier que la moyenne de $\bar{x}_1, \ldots \bar{x}_N$ est près de μ

Planification d'une étude de simulation

- Nous savons que
$$E[\bar{X}] = \mu$$
- Comment vérifier ce résultat par simulation ?
  1. générer un échantillon $x_1, \ldots, x_n$ de la loi de X
  2. calculer $\bar{x}$
  3. répéter N fois $\to \bar{x}_1, \ldots \bar{x}_N$
  4. vérifier que la moyenne de $\bar{x}_1, \ldots \bar{x}_N$ est près de μ
- Grand échantillon ou grand nombre de répétitions ?

Planification d'une étude de simulation

- Nous savons que
$$E[\bar{X}] = \mu$$
- Comment vérifier ce résultat par simulation ?
  1. générer un échantillon $x_1, \ldots, x_n$ de la loi de X
  2. calculer $\bar{x}$
  3. répéter N fois $\rightarrow \bar{x}_1, \ldots \bar{x}_N$
  4. vérifier que la moyenne de $\bar{x}_1, \ldots \bar{x}_N$ est près de μ
- Grand échantillon ou grand nombre de répétitions ?
  1. n grand valide que $\bar{X} \xrightarrow{n \to \infty} \mu$ [Loi des grands nombres]
  2. N grand valide que $E[\bar{X}] = \mu$ [$\bar{X}$ sans biais pour μ]

Intégration Monte Carlo

Une idée simple (mais plein d'implications) :

$$\theta = \int_a^b h(x)\,dx$$

$$\Downarrow$$

$$\theta = \int_a^b g(x)f(x)\,dx = E[g(X)]$$

$$\Downarrow$$

$$\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} g(x_i), \quad x_1, \ldots, x_n \sim f$$

Analyse numérique

Objectifs du chapitre 10

- Constater les limites des ordinateurs pour le calcul algébrique

10 times 0.1 hardly ever makes 1.0

- Apprendre à amoindrir l'impact ou à contourner ces limites

D’où proviennent ces valeurs ?

Double (double-precision floating-point)	Double	8 bytes	-1.79769313486231570E+308 through -4.94065645841246544E-324 † for negative values; 4.94065645841246544E-324 through 1.79769313486231570E+308 † for positive values
Integer	Int32	4 bytes	-2,147,483,648 through 2,147,483,647 (signed)
Long (long integer)	Int64	8 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807 (9.2...E+18 †) (signed)

Caractéristiques obtenues

La représentation interne des nombres dans l'ordinateur permet de déduire les valeurs suivantes :

1. Plus grand nombre représentable
2. Plus petit nombre représentable
3. Plus petite valeur tel que $1+\varepsilon \neq 1$

   ```
   all.equal(target, current,
             tolerance = .Machine$double.eps ^ 0.5, scale = NULL,
             ..., check.attributes = TRUE)
   ```

4. Écart entre deux grands nombres
5. Écart entre deux petits nombres

$v = (1+i)^{-1}$

$v^{-1} = (1+i)$

$v^{-1} - 1 = i$

Arithmétique en virgule flottante

- Règles de l'arithmétique ne sont pas les mêmes !
- Cinq principes de programmation
- Coût relatif des opérations

Opération arithmétique	Coût relatif
Addition et soustraction	1,0
Multiplication	1,3
Division	3,0
Racine carrée	4,0
Logarithme	15,4

Bonnes et moins bonnes pratiques

Au lieu de faire...

```
x^(1/2)

x^(1/5)

(1/k) * x

sum(x/k)
```

Bonnes et moins bonnes pratiques

Au lieu de faire…	… faites plutôt
`x^(1/2)`	`sqrt(x)`
`x^(1/5)`	`x^0.2`
`(1/k) * x`	`x/k`
`sum(x/k)`	`sum(x)/k`

Un problème…

$$\text{résoudre } f(x) = 0$$

…trois solutions :

1. méthode de la bissection
2. méthode du point fixe
3. méthode de Newton–Raphson

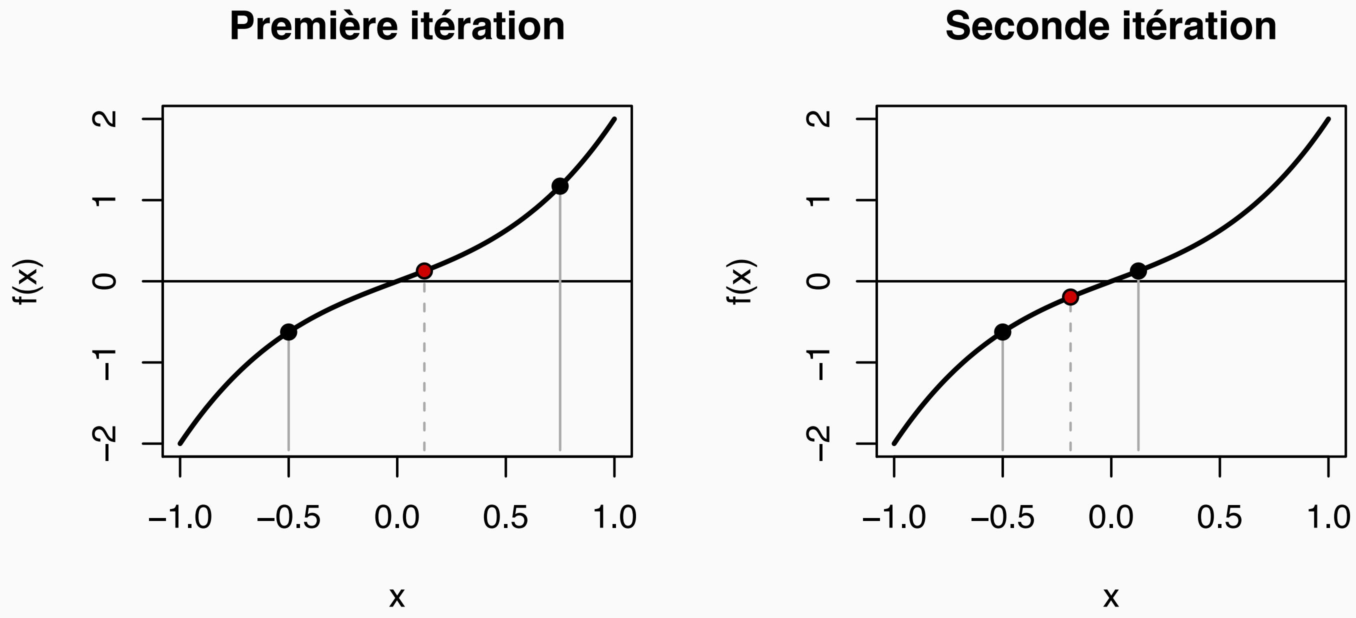
$\frac{2}{3} = 0,\bar{6}$ $\quad 0,10$

$0,\bar{6} \times 2 = 1,\bar{3}$

$0,\bar{3} \times 2 = 0,\bar{6}$

Méthode de la bissection

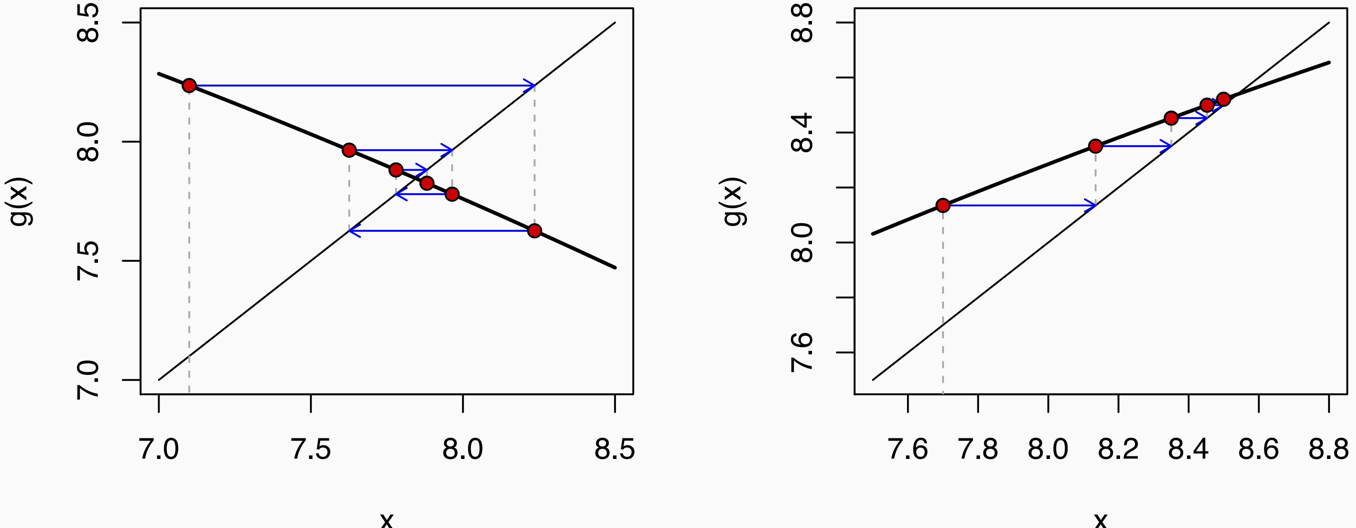
Essais successifs de part et d'autre de la solution



Méthode du point fixe

Résoudre

$$x = g(x)$$



Méthode de Newton-Raphson

- Problème de recherche de racine
- Solution est le point de convergence de

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})},$$

Astuce Ripley

- Problème d'optimisation lorsque paramètres strictement positifs
- Idée : estimer le logarithme des paramètres :

$$\theta \in \mathbb{R}^+ \quad \Leftrightarrow \quad \tilde{\theta} = \ln \theta \in \mathbb{R}$$

- Par exemple, plutôt que de travailler avec

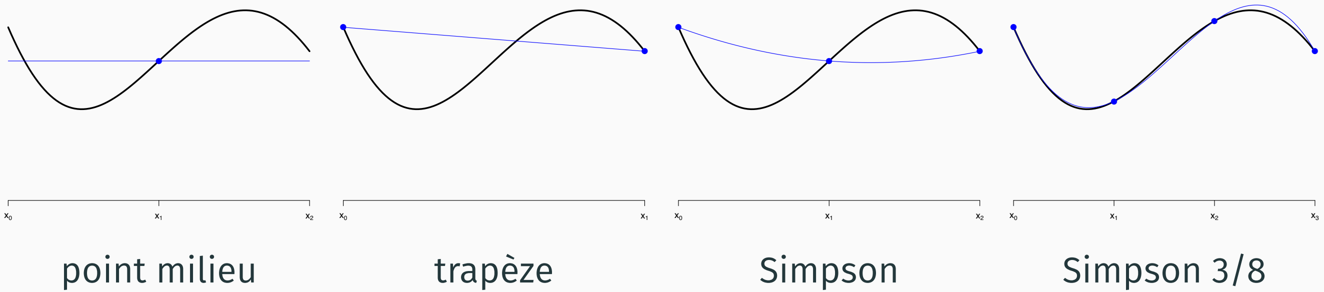
$$f(x; \alpha, \lambda) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}$$

  on prend

$$f(x; \tilde{\alpha}, \tilde{\lambda}) = \frac{(e^{\tilde{\lambda}})^{e^{\tilde{\alpha}}}}{\Gamma(e^{\tilde{\alpha}})} x^{e^{\tilde{\alpha}}-1} e^{-e^{\tilde{\lambda}} x}$$

Intégration numérique

- Idée maîtresse



- Clé : polynômes d'interpolation de Lagrange

Algèbre linéaire

Branchez-vous à

socrative.com

Student login

Salle de classe ACT2002

Un théorème pour les gouverner tous

Soit $\mathbf{A}$ une matrice $n \times n$. Les énoncés suivants sont équivalents.

- $\mathbf{A}$ est inversible
- $\det(\mathbf{A}) \neq 0$
- $\text{rang}(\mathbf{A}) = n$
- La seule solution de $\mathbf{Ax} = \mathbf{0}$ est la solution triviale
- Le système d'équations $\mathbf{Ax} = \mathbf{b}$ a une solution pour tout vecteur $\mathbf{b}$ et celle-ci est unique
- Les lignes de $\mathbf{A}$ sont linéairement indépendantes
- Les lignes de $\mathbf{A}$ forment une base de $\mathbb{R}^n$
- $\lambda = 0$ n'est pas une valeur propre de $\mathbf{A}$

Valeur et vecteur propre : définition

- Valeur λ tel que

$$\mathbf{Ax} = \lambda \mathbf{x}$$

$$\Updownarrow$$

$$(\lambda \mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{0}$$

  pour $\mathbf{x} \neq \mathbf{0}$

- $\mathbf{x}$ est le vecteur propre correspondant à λ

Diagonalisation

Soit **A** une matrice $n \times n$. Les énoncés suivants sont équivalents.

1. **A** est diagonalisable.
2. **A** possède n vecteurs propres linéairement indépendants.

De plus, si **P** est la matrice qui diagonalise **A**, alors

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{P} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

Probabilité de ruine

- Sous certaines hypothèses,

$$\begin{aligned}\psi(u) &= 1 - F(u) \\ &= \pi e^{\mathbf{T}u}\mathbf{e}\end{aligned}$$

  où

$$e^{\mathbf{M}} = \sum_{n=0}^{\infty} \frac{\mathbf{M}^n}{n!}$$

- F est la fonction de répartition de la distribution *phase-type* avec vecteur de probabilités initiales π et matrice de transition $\mathbf{T}$

Exponentielle d'une matrice

- Si $\mathbf{A} = \operatorname{diag}(a_1, \ldots, a_k)$ est *diagonale*

$$e^{\mathbf{A}} = \operatorname{diag}(e^{a_1}, \ldots, e^{a_k})$$

- Si $\mathbf{M}$ est *diagonalisable*, on peut écrire

$$\mathbf{A} = \mathbf{P}^{-1}\mathbf{M}\mathbf{P} = \operatorname{diag}(\lambda_1, \ldots, \lambda_k)$$

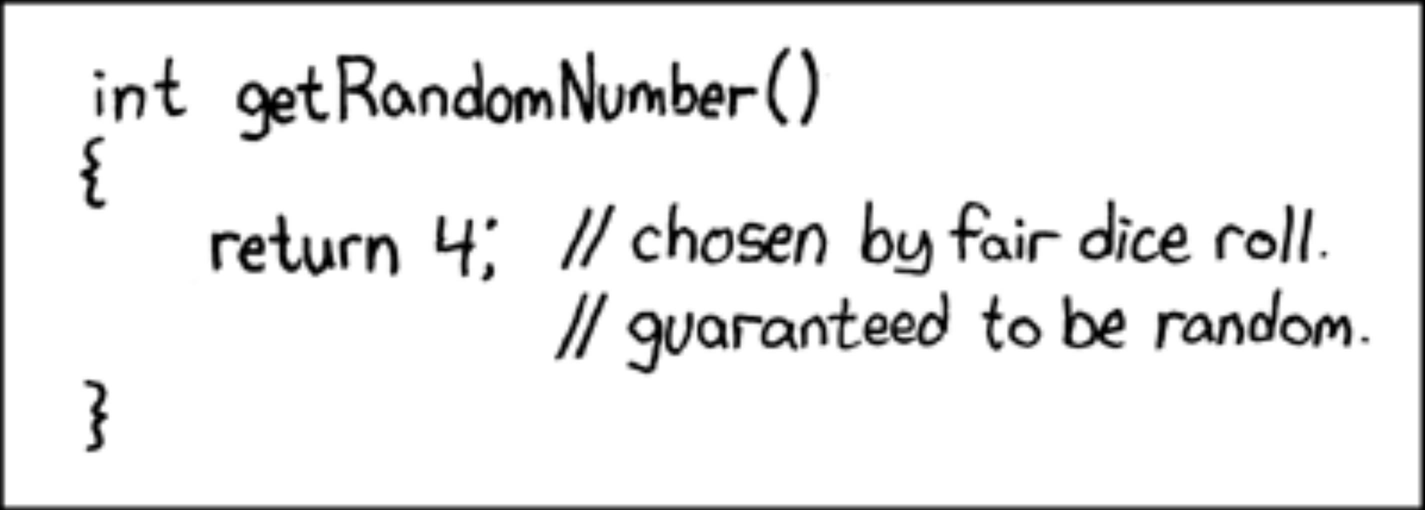
$$\Updownarrow$$

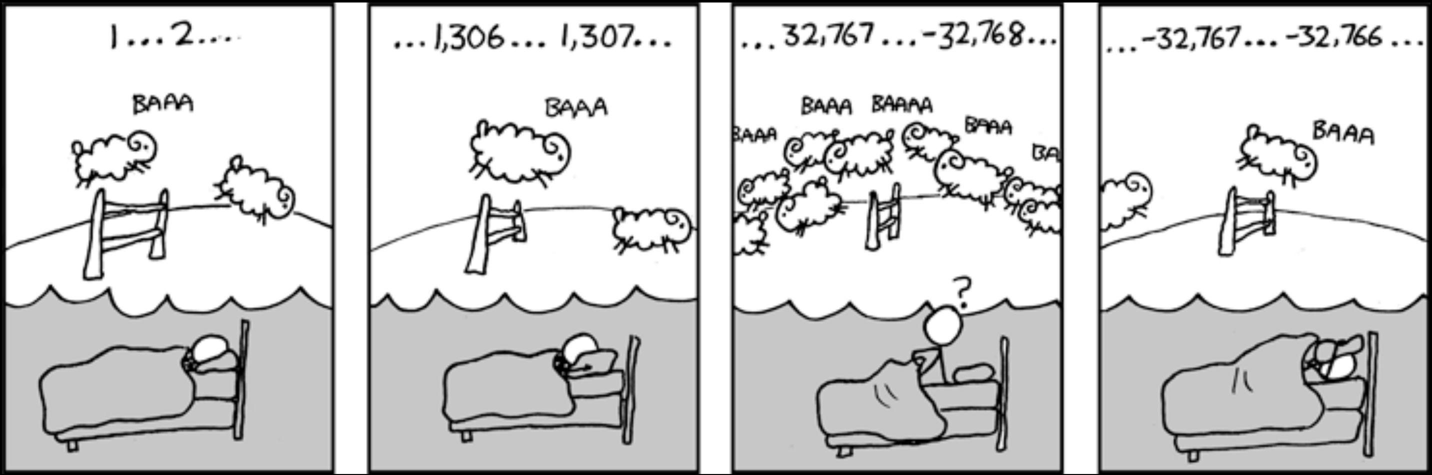
$$\mathbf{M} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1} = \mathbf{P}\operatorname{diag}(\lambda_1, \ldots, \lambda_k)\mathbf{P}^{-1}$$

et donc

$$\begin{aligned} e^{\mathbf{M}} &= \mathbf{P}e^{\mathbf{A}}\mathbf{P}^{-1} \\ &= \mathbf{P}\operatorname{diag}(e^{\lambda_1}, \ldots, e^{\lambda_k})\mathbf{P}^{-1} \end{aligned}$$

Validation finale des apprentissages


int getRandomNumber()
{
return 4; // chosen by fair dice roll.
// guaranteed to be random.
}


1...2...
BAAA
...1,306... 1,307...
BAAA
...32,767...-32,768...
BAAA
BAAA
BAAAA
BAAA
?
...-32,767...-32,766...
BAAA

$$\begin{bmatrix} \cos 90^\circ & \sin 90^\circ \\ -\sin 90^\circ & \cos 90^\circ \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} a_2 & a_1 \end{bmatrix}$$

Ce document a été produit par le système de mise en page XƎLaTeX avec la classe **beamer** et le thème Metropolis. Les titres et le texte sont composés en Fira Sans, les mathématiques en Arev Math et le code informatique en Fira Mono. Les icônes proviennent de la police Font Awesome. Les graphiques ont été réalisés avec R.