

ECN 6338 Cours 7

Intégration et dérivation déterministe

William McCausland

2025-02-25

Survol du cours 7

- ▶ Exemple économique, enchères en équilibre
- ▶ Formules (ou règles) Newton-Cotes pour l'intégration univariée
 - ▶ Règle du point médian
 - ▶ Règle de trapèze
 - ▶ Règle de Simpson
- ▶ Formules (ou règles) de Gauss pour l'intégration univariée
 - ▶ Règle de Gauss-Legendre et les enchères
 - ▶ Règle de Gauss-Laguerre et l'utilité actualisée
 - ▶ Règle de Gauss-Hermite et les espérances gaussiennes
- ▶ Intégration multivariée par intégration univariée répétée (et ses limites)
- ▶ Dérivation numérique

Exemple 1, enchères en équilibre

- ▶ Considérez une vente aux enchères hollandaise (ou descendante, ou à premier prix sous pli cacheté) avec n enchéreur.
- ▶ Dans ce type de vente aux enchères, celui qui fait l'enchère la plus grande obtient l'objet, au prix égal à son enchère.
- ▶ Supposez que les valeurs privées de l'objet à vendre, v_1, v_2, \dots, v_n , sont iid avec fonction de répartition F sur $[v_{\min}, v_{\max}]$.
- ▶ En équilibre de Bayes-Nash, un enchéreur avec une valeur privée de v fait une enchère de

$$b(v) = v - \frac{\int_{v_{\min}}^v [F(x)]^{n-1} dx}{[F(v)]^{n-1}}.$$

Une fonction de répartition F pour les valeurs privées

- ▶ Supposez que $F(x) = 1 - (1 - x)^{5/2}$ sur $[v_{\min}, v_{\max}] = [0, 1]$.
- ▶ La fonction de répartition et le fonction de densité :

```
# Fonction de répartition de la valeur privée
```

```
Fv <- function(x) {1-(1-x)^2.5}
```

```
# Fonction de densité de la valeur privée
```

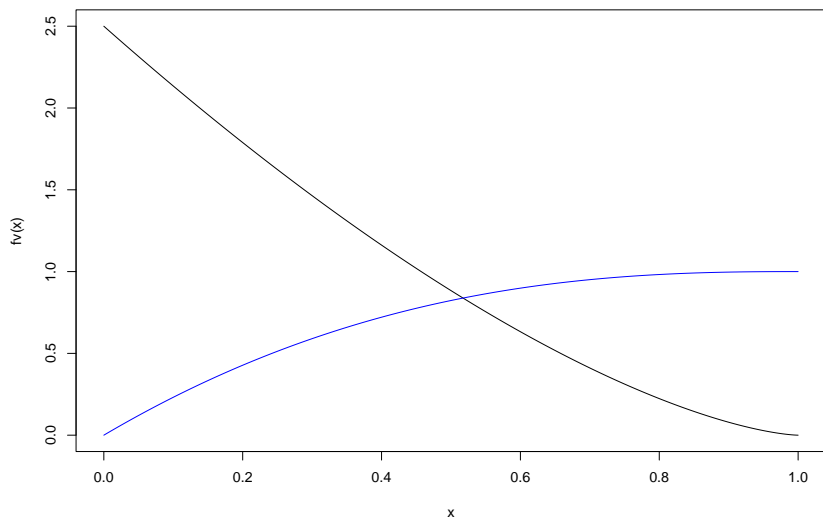
```
fv <- function(x) {2.5 * (1-x)^1.5}
```

```
# Grille de points
```

```
x <- seq(0, 1, length=1000)
```

Graphiques de F , f pour les valeurs privées

```
plot(x, fv(x), type='l')  
lines(x, Fv(x), col='blue')
```



Formules de Newton-Cotes pour l'intégration univarié

1. Règle du point médian

$$\int_a^b f(x) dx \approx (b - a) f\left(\frac{a + b}{2}\right)$$

2. Règle de trapèze

$$\int_a^b f(x) dx \approx (b - a) \frac{f(a) + f(b)}{2}$$

3. Règle de Simpson

$$\int_a^b f(x) dx \approx (b - a) \left[\frac{1}{3} \frac{f(a) + f(b)}{2} + \frac{2}{3} f\left(\frac{a + b}{2}\right) \right]$$

La règle du point médian et l'analyse de son erreur

- ▶ Supposons que $f \in C^2[a, b]$.
- ▶ Par la formule de [Taylor-Lagrange](#), pour chaque $x \in (a, b]$ il y a un $\xi \in (\frac{a+b}{2}, x)$ tel que

$$f(x) = f\left(\frac{a+b}{2}\right) + f'\left(\frac{a+b}{2}\right)\left(x - \frac{a+b}{2}\right) + \frac{1}{2}f''(\xi)\left(x - \frac{a+b}{2}\right)^2$$

- ▶ Alors il y a un $\xi \in [a, b]$ tel que

$$\begin{aligned}\int_a^b f(x) dx &= (b-a)f\left(\frac{a+b}{2}\right) + \frac{1}{6}f''(\xi) \left[\left(x - \frac{a+b}{2}\right)^3 \right]_a^b \\ &= (b-a)f\left(\frac{a+b}{2}\right) + \frac{1}{6}f''(\xi) \left[\frac{(b-a)^3}{8} - \frac{(a-b)^3}{8} \right] \\ &= (b-a)f\left(\frac{a+b}{2}\right) + \frac{(b-a)^3}{24}f''(\xi).\end{aligned}$$

- ▶ Le premier terme est l'approximation de l'intégrale par la méthode du point médian.
- ▶ Le deuxième terme est l'erreur.

La version composite de règle du point médian

- ▶ Décomposer l'intervalle $[a, b]$ en n sous-intervalles de longueur $h = \frac{b-a}{n}$.
- ▶ Les points médians des sous-intervalles sont

$$x_j = a + (j - \frac{1}{2})h, \quad j = 1, 2, \dots, n.$$

- ▶ Décomposer l'intégral par intervalle et utiliser la règle du point médian, intervalle par intervalle, donne, pour un $\xi \in [a, b]$,

$$\int_a^b f(x) dx = h \sum_{j=1}^n f(x_j) + \frac{h^2(b-a)}{24} f''(\xi).$$

La règle de trapèze

- ▶ La règle de trapèze utilise les deux points extrêmes, et le résultat est, pour un $\xi \in [a, b]$,

$$\int_a^b f(x) dx = \frac{b-a}{2}[f(a) + f(b)] - \frac{(b-a)^3}{12}f''(\xi).$$

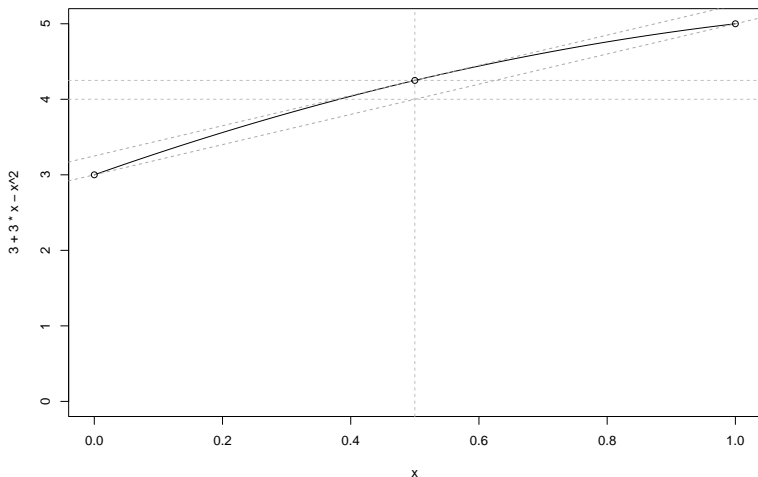
- ▶ Pour la version composite,
 - ▶ toutes les points d'évaluation, sauf deux, sont des points extrêmes de deux intervalles.
 - ▶ décomposer l'intervalle $[a, b]$ en n sous-intervalles de longueur $h = \frac{b-a}{n}$.
 - ▶ les points extrêmes sont $x_i = a + ih$, $i = 0, \dots, n$.
 - ▶ le résultat est, pour $\xi \in [a, b]$,

$$\int_a^b f(x) dx = \frac{h}{2}[f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(x_i)] - \frac{h^2(b-a)}{12}f''(\xi).$$

La règle de Simpson

- ▶ Avec le point médian et les points extrêmes, on peut utiliser la règle de Simpson : il y a un $\xi \in [a, b]$ tel que

$$\int_a^b f(x) dx = (b-a) \left[\frac{1}{3} \frac{f(a)+f(b)}{2} + \frac{2}{3} f\left(\frac{a+b}{2}\right) \right] - \frac{(b-a)^5}{2880} f^{(4)}(\xi)$$



La règle de Simpson (suite)

- ▶ La règle de Simpson simple donne l'intégral exacte d'une approximation quadratique
- ▶ La version composite donne, avec les mêmes n évaluations que la version composite de la règle de trapèze (mais $n/2$ intervalles de longueur $2h$ et n doit être pair),

$$\int_a^b f(x) dx = \frac{h}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 4f_{n-1} + f_n] \\ - \frac{h^4(b-a)}{180} f^{(4)}(\xi).$$

où $f_i \equiv f(x_i)$.

Notes pour comprendre le code pour $b(v)$

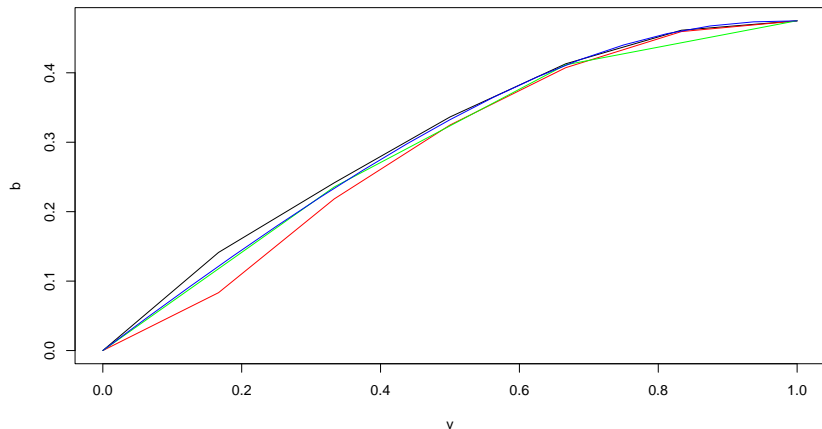
symbole	type	signification
n	entier	nombre d'intervalles
nb	entier	nombre de joueurs
F	fonction	fonction de répartition des valeurs v_i
v1	vecteur	milieux des intervalles $(\frac{1}{2n}, \frac{3}{2n}, \dots, \frac{2n-1}{2n})$
v2	vecteur	bornes des intervalles $(0, \frac{1}{n}, \frac{2}{n}, \dots, 1)$
f1, f2	vecteur	valeurs de $F(v)^{n-1}$ sur les grilles v1 et v2
v_me	vecteur	grille de valeurs de v pour évaluer $b(v)$, méthode de médiane
I_me	vecteur	l'intégrale $\int_{v_{\min}}^v F(x)^{n-1} dx$ sur la grille v_me
b_me	vecteur	valeurs de $b(v)$ sur le grille v_me
I_tr, &c	vecteur	Comme I_me, méthode de trapezoïde
I_S, &c	vecteur	Comme I_S, méthode de Simpson

Trouver la fonction d'enchère $b(v)$

```
NCbf <- function(n, nb, F) {  
  v1 <- (seq(1, n) - 0.5)/n; f1 <- F(v1)^(nb-1)  
  v2 <- seq(0, n)/n; f2 <- F(v2)^(nb-1)  
  v_me <- v_tr <- v2; b_me <- b_tr <- v2;  
  v_S <- 2*seq(0, n/2)/n; b_S <- v2[seq(1, n+1, by=2)]  
  I_me <- I_tr <- I_S <- 0  
  for (i in seq(1, n)) {  
    I_me <- I_me + f1[i]/n;  
    b_me[i+1] <- b_me[i+1] - I_me/f2[i+1]  
    I_tr <- I_tr + (f2[i] + f2[i+1]) / (2*n);  
    b_tr[i+1] <- b_tr[i+1] - I_tr/f2[i+1]  
  }  
  for (i in seq(1, n/2)) {  
    I_S <- I_S + (f2[2*i-1] + 4*f2[2*i] + f2[2*i+1])/(3*n)  
    b_S[i+1] <- b_S[i+1] - I_S/f2[2*i+1]  
  }  
  list(v_me=v_me, b_me=b_me, v_tr=v_tr, b_tr=b_tr, v_S=v_S,  
}
```

Graphiques, fonctions d'enchères

```
NC6 <- NCbf(6, 4, Fv); NC32 <- NCbf(32, 4, Fv)
plot(NC6$v_me, NC6$b_me, type='l', xlab='v', ylab='b')
lines(NC6$v_tr, NC6$b_tr, col='red')
lines(NC6$v_S, NC6$b_S, col='green')
lines(NC32$v_S, NC32$b_S, col='blue')
```



Règles gaussiennes

- ▶ Comme les règles de Newton-Cotes, les règles gaussienne donnent des approximations de la forme

$$\int_a^b w(x)f(x) dx \approx \sum_{i=1}^N \omega_i f(x_i),$$

pour une collection de paires (ω_i, x_i) .

- ▶ Résultat théorique (Judd, théorème 7.2.1) : avec N poids et N noeuds les règles standards donnent la valeur exacte de l'intégral pour les polynômes d'ordre $2N - 1$ et des formules pour le termes résiduels.
- ▶ Différences, par rapport aux règles de Newton-Cotes :
 - ▶ les règles gaussiennes donnent les intégrales pondérées (avec $w(x)$) des polynômes globaux
 - ▶ les règles sont calculées à partir des suites de polynômes orthogonaux
 - ▶ les règles spécifient et les poids et les noeuds, pour un ordre N donné

Règles gaussiennes (suite)

- Judd donne des formules pour les intégrales suivants

$$\int_{-1}^1 f(x)(1-x^2)^{-1/2} dx \quad (\text{Gauss-Tchebyshev})$$

$$\int_{-1}^1 f(x) dx \quad (\text{Gauss-Legendre})$$

$$\int_{-\infty}^{\infty} f(x)e^{-x^2} dx \quad (\text{Gauss-Hermite})$$

$$\int_0^{\infty} f(x)e^{-x} dx \quad (\text{Gauss-Laguerre})$$

- Les deux dernières ont des applications spéciales en économie

Noeuds et poids pour l'intégration Gauss-Legendre sur $[0,1]$

```
library(mvQuad)
library(knitr)      # Pour kable
library(tidyverse)  # Pour tibble
nw <- createNIGrid(dim=1, type="GL", level=6)
rescale(nw, domain = c(0,1))
tbl <- tibble(Noeuds=nw$nodes, Poids=nw$weights)
kable(tbl)
```

Noeuds	Poids
0.03376524	0.08566225
0.16939531	0.18038079
0.38069041	0.23395697
0.61930959	0.23395697
0.83060469	0.18038079
0.96623476	0.08566225

Calcul de la fonction $b(v)$ à $v = 0.5$ et $v = 1$

```
# Fonction à intégrer
```

```
f_int <- function(v, nb) {Fv(v)^(nb-1)}
```

```
# Fonction d'enchère
```

```
b <- function(v, nb) {  
  rescale(nw, domain = c(0,v))  
  (v - quadrature(f_int, grid = nw, nb) / Fv(v)^(nb-1))  
}
```

```
c(b(0.5, 4), b(1, 4))
```

```
## [1] 0.3324346 0.4747910
```

```
c(NC32$b_S[9], NC32$b_S[17])
```

```
## [1] 0.3324358 0.4747901
```

Exemple 2, Actualisation

- ▶ Deux problèmes d'actualisation en temps continu :
 - ▶ du consommateur,

$$\int_0^{\infty} e^{-\rho t} u(c(t)) dt,$$

- ▶ et de la firme

$$\int_0^{\infty} e^{-rt} \pi(q(t)) dt.$$

- ▶ Un changement de variable $s = \rho t$, $ds = \rho dt$, de l'actualisation du consommateur donne

$$\frac{1}{\rho} \int_0^{\infty} e^{-s} u(c(s/\rho)) ds,$$

un intégral de la forme $\int_0^{\infty} e^{-x} g(x) dx$ (c.-à-d. $(a, b) = (0, \infty)$ et $w(x) = e^{-x}$), et donc disposé à approximation avec la règle Gauss-Laguerre.

Exercice 5, chapitre 7 de Judd

- L'exercice concerne le problème d'actualisation du consommateur, pour

$$c(t) = 1 - e^{-\lambda t}, \quad u(c) = -e^{-ac}.$$

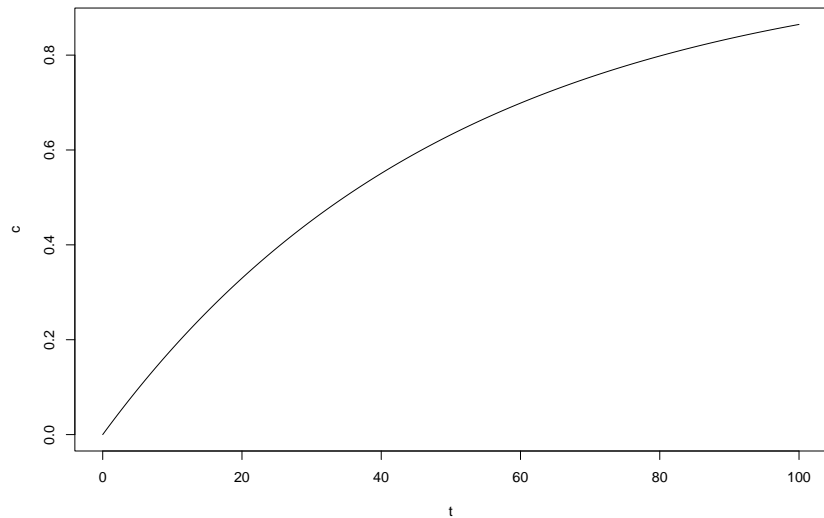
Calculs pour des graphiques d'illustration :

```
# Une configuration possible des paramètres
lambda <- 0.02; rho <- 0.04; a <- 0.5

t <- seq(0, 100, by=0.1) # Temps, en ans
c <- 1 - exp(-lambda*t)  # Chemin de consommation
u_inst <- -exp(-a*c)      # Utilité instantanée ...
u_actu <- exp(-rho*t) * u_inst # ... et actualisée
```

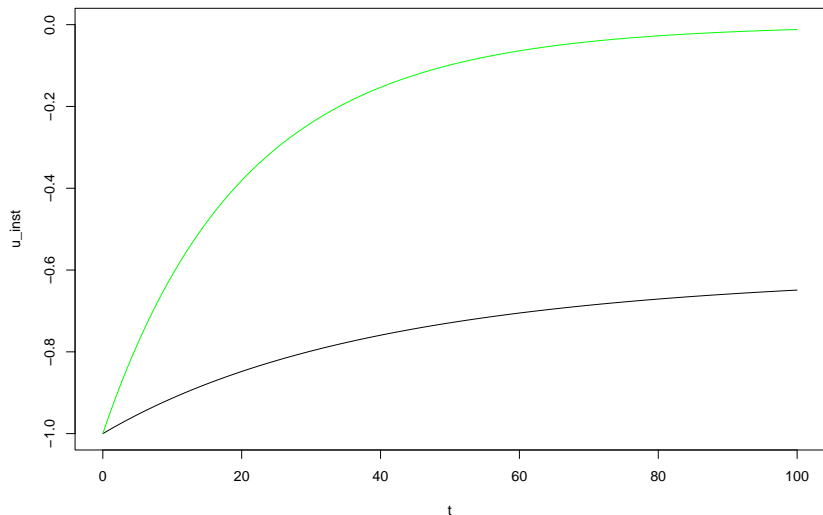
Chemin de consommation

```
plot(t, c, type='l')
```



Utilité instantanée, avant et après actualisation

```
plot(t, u_inst, type='l', ylim=c(-1,0))  
lines(t, u_actu, col='green')
```



Une fonction pour l'intégration Gauss-Laguerre

Après le changement de variables $s = \rho t$, on a une fonction $u_s(s)$ à intégrer par rapport à $w(s) = e^{-s}$:

```
u_s <- function(s, lambda, rho, a)
{
  c <- 1 - exp(-(lambda/rho) * s)
  u_s <- -exp(-a*c)/rho
}
```

Noeuds et poids pour l'intégration Gauss-Laguerre

```
nw <- createNIGrid(d=1, type='GLa', level=5)
tbl <- tibble(Noeuds=nw$nodes, Poids=nw$weights)
kable(tbl)
```

Noeuds	Poids
0.2635603	5.217556e-01
1.4134031	3.986668e-01
3.5964258	7.594245e-02
7.0858100	3.611759e-03
12.6408008	2.336997e-05

Valeurs de l'intégral pour $N = 3, 5, 10$

```
nw <- createNIGrid(d=1, type='GLa', level=3)
I3 = quadrature(u_s, grid=nw, lambda, rho, a)
I3
```

```
## [1] -21.29571
```

```
nw <- createNIGrid(d=1, type='GLa', level=5)
I5 = quadrature(u_s, grid=nw, lambda, rho, a)
I5
```

```
## [1] -21.30583
```

```
nw <- createNIGrid(d=1, type='GLa', level=10)
I10 = quadrature(u_s, grid=nw, lambda, rho, a)
I10
```

```
## [1] -21.30613
```

Exemple 3, Espérance par rapport à une loi gaussienne

- Problème : évaluer $E[f(Y)]$ pour $Y \sim N(\mu, \sigma^2)$.
- L'intégral est

$$E[f(Y)] = (2\pi\sigma^2)^{-1/2} \int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}(y-\mu)^2} f(y) dy.$$

- Un changement de variables $x = (y - \mu)/(\sqrt{2}\sigma)$,
 $dx = dy/(\sqrt{2}\sigma)$ donne

$$E[f(Y)] = \pi^{-1/2} \int_{-\infty}^{\infty} e^{-x^2} f(\mu + \sqrt{2}\sigma x) dx,$$

un intégral de la forme $\int_{-\infty}^{\infty} e^{-x^2} g(x) dx$
(c.-à-d. $(a, b) = (-\infty, \infty)$ et $w(x) = e^{-x^2}$), et donc disposé à approximation avec la règle Gauss-Hermite.

Exemple, Section 7.6 de Judd

- Problème du choix de portefeuille :

$$\max_{\omega} E[u(R(W - \omega) + \omega e^Z)],$$

où

- W est la richesse au moment de la décision, au début de la période, en dollars;
- ω est le montant, en dollars, placé en un actif avec risque;
- $W - \omega$ est le montant, en dollars, placé en un actif sans risque;
- R est le rendement brut, non-aléatoire, de l'actif sans risque pendant la période de l'analyse;
- $Z \in N(\mu, \sigma^2)$ est le log-rendement aléatoire de l'actif avec risque;
- $u(c) = c^{1+\gamma}/(1+\gamma)$.

Une fonction pour l'intégration Gauss-Hermite

Après le changement de variables $x = (z - \mu)/(\sqrt{2}\sigma)$, on a une fonction $u_x(x)$ à intégrer par rapport à $w(x) = e^{-x^2}$:

```
u_x <- function(x, omega, gamma, mu, sigma, R, W)
{
  c = R*(W-omega) + omega*exp(mu + sqrt(2)*sigma*x)
  u_x = pi^(-0.5) * c^(1+gamma)/(1+gamma)
}
```

Dans un premier temps, on considère le problème du calcul de l'intégral suivant pour ω fixe :

$$U = \int_{-\infty}^{\infty} e^{-x^2} u_x(x) dx.$$

Noeuds et poids pour l'intégration Gauss-Hermite

```
library(gaussquad)
regle <- hermite.h.quadrature.rules(7)[[7]]
tbl <- tibble(Noeuds=regle$x, Poids=regle$w)
kable(tbl)
```

Noeuds	Poids
2.6519614	0.0009718
1.6735516	0.0545156
0.8162879	0.4256073
0.0000000	0.8102646
-0.8162879	0.4256073
-1.6735516	0.0545156
-2.6519614	0.0009718

Une fonction pour évaluer l'utilité U comme fonction de ω

```
Eu_x <- function(omega, gamma, mu, sigma, R, W, regle)
{
  hermite.h.quadrature(u_x, regle, -Inf, Inf, weighted=T,
                      omega, gamma, mu, sigma, R, W)
}

Eu_x_minus <- function(omega, gamma, mu, sigma, R, W, regle)
{
  -Eu_x(omega, gamma, mu, sigma, R, W, regle)
}

# Valeurs de omega et des paramètres pour
# l'exemple, pages 262-263
Eu_x(omega=1, gamma=-0.5, mu=0.15, sigma=0.25, R=1, W=2, regle)

## [1] 2.958743
```

U pour plusieurs valeurs de ω

```
omega <- array(seq(0, 2, by=0.4))  
gamma = -5; mu=0.2; sigma=0.3; R=1.1; W=2  
U <- apply(omega, 1, Eu_x, gamma, mu, sigma, R, 2, regle)  
tbl <- tibble(omega=omega, U=U)  
kable(tbl)
```

omega	U
0.0	-0.0106721
0.4	-0.0098145
0.8	-0.0097168
1.2	-0.0102605
1.6	-0.0116135
2.0	-0.0144237

Résolution du problème de portefeuille

```
optimise(Eu_x_minus, interval=c(0,2),  
         gamma, mu, sigma, R, W, regle)
```

```
## $minimum
```

```
## [1] 0.6592695
```

```
##
```

```
## $objective
```

```
## [1] 0.00967836
```

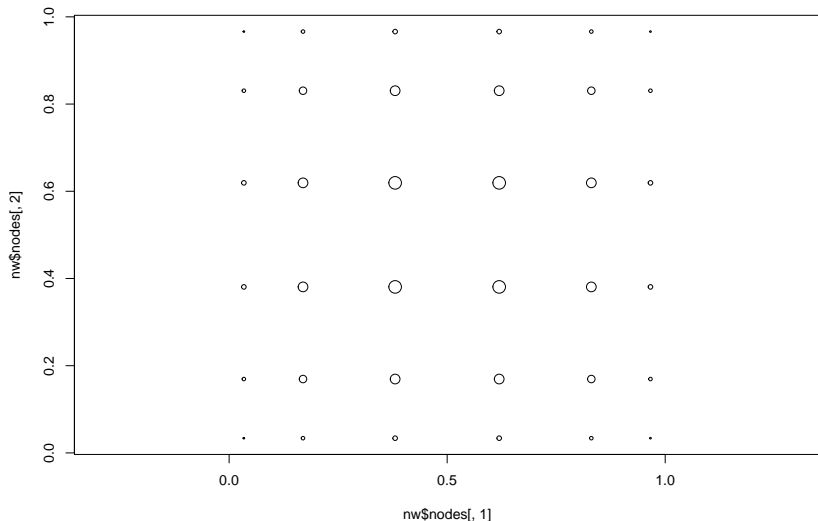

Règles de produit pour l'intégration multivariée

- ▶ L'intégration multivariée est difficile.
- ▶ Les règles de produit sont des solutions de force brute qui marche quand
 - ▶ la région d'intégration est un espace produit, et
 - ▶ la dimension est basse (2-5)
- ▶ Illustration de l'intégration sur $[-1, 1]^d$:
 - ▶ prendre une règle univarié pour l'intégration sur $[-1, 1]$, avec pairs (w_i, x_i) , $i = 1, \dots, m$,
 - ▶ former l'approximation

$$\int_{[-1,1]^d} f(x) dx \approx \sum_{i_1=1}^m \cdots \sum_{i_d=1}^m \omega_{i_1} \omega_{i_2} \cdots \omega_{i_d} f(x_{i_1}, x_{i_2}, \dots, x_{i_d})$$

Intégration sur $[0, 1]^2$: noeuds, poids d'un règle de produit

```
nw <- createNIGrid(dim=2, type = 'GLe', level=6)  
plot(nw$nodes[,1], nw$nodes[,2], cex=36*nw$weights, asp=1)
```



Commentaires sur l'intégration multivariée

- ▶ On peut toujours construire des règles de produit mais la computation coûte.
- ▶ Avec les pairs (w_i, x_i) , $i = 1, \dots, m$, d'une règle exacte pour les polynômes d'ordre $2m - 1$, on a l'approximation

$$\int_{[-1,1]^d} f(x) dx \approx \sum_{i_1=1}^m \cdots \sum_{i_d=1}^m \omega_{i_1} \omega_{i_2} \cdots \omega_{i_d} f(x_{i_1}, x_{i_2}, \dots, x_{i_d})$$

exacte pour les polynômes multivariés jusqu'à l'ordre $2m - 1$, dimension par dimension.

- ▶ m^d points pour d dimensions!
- ▶ Le théorème de Mysovskikh promet l'intégration exacte des polynôme d'ordre *total* m avec un nombre comparable de points, mais ce n'est pas constructif.
- ▶ Il y a des formules pour les hyper-rectangles, les hypersphères, les simplexes, etc.

Dérivation numérique

La formule de **Taylor-Lagrange** pour une expansion autour de x donne (pour un $\xi \in (x, x + h)$)

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(\xi),$$

Cependant, avec la formule bilatérale et l'expansion autour de x , on obtient (pour un $\xi \in (x, x + h)$)

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6}f'''(\xi).$$

Pour la formule bilatérale, h optimal est $h^* = (3\epsilon/M_3)^{1/3}$, qui donne une erreur maximale de $2\epsilon^{2/3}M_3^{1/3}9^{1/3}$, où ϵ est l'erreur maximale de l'évaluation de $f(x)$ et M_3 majore $|f'''|$ près de x .