# Numerics in Applied Mathematical Finance (with R)

Maxim Litvak

2017-03-06

# Outline

1. Outline
2. Basic numerics
3. Numerics in Stochastics
4. Application case

# Outline

- Numerical derivatives
- Matrix decomposition
- Pseudorandom number generators
- Generation of correlated random numbers
- Computation of quantile functions
- Constrained distributions

# Numerical derivative - I

## 2 point symmetric derivative

$f'_{num}(x) = \frac{f(x+h)-f(x-h)}{2h}$

## Both advantage and disadvantage of symmetry:

More derivatives exist (e.g. modulus function), but some of them you may not want to exist

## Question: how to set h?

Set h depending on application cases.

# Numerical derivative - II

## Code example

```
num.deriv <- function(f, x, h = 1e-05)
{
  return((f(x + h) - f(x - h))/(2*h))
}

print(num.deriv(sqrt, 4, .1))
print(num.deriv(sqrt, 4, .01))

[1] 0.2500195
[1] 0.2500002
```

# Numerical derivative - III

## 5 point derivative

If 2-point derivative behaves badly, try more precision

$$f'_{num} = \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12h}$$

# Pseudo-Random Numbers Generators (PRNG) - I

## Importance of reproducibility of results

### Sounds like a paradox: random numbers must be reproducible

Helps to spot the effects of other factors, e.g. you need to calculate impact of the new pricing algorithm and eliminate effect of randomness.

## Setting seeds

- "Whatever one sows, that will he also reap"
- Input: one number (called "seed"), output: the sequence of numbers that repeats only after very big period
- Period of currently popular Mersenne Twister is $2^{19937} - 1$
- RAND() function in Excel2003 has period of $10^{13}$
- However, there are still legacy systems in use with small period (e.g. 40 mln)

# Pseudo-Random Numbers Generators (PRNG) - II

## Code example

```
loss.dist <- function(seed, N)
{
  set.seed(seed)
  return(runif(N))
}

print(loss.dist(seed=1, N=4))
print(loss.dist(seed=1, N=4))  # same seed - same sequence
print(loss.dist(seed=10, N=4)) # different seed - different se

[1] 0.2655087 0.3721239 0.5728534 0.9082078
[1] 0.2655087 0.3721239 0.5728534 0.9082078
[1] 0.5074782 0.3067685 0.4269077 0.6931021
```

# Semidefinite matrix decomposition and eigenvalues

## Example

Assume that the default rates in different industries are correlated. The corresponding correlation matrix is positive semidefinite.

## Reason?

- In order to draw the correlated random variables we need to decompose the correlation matrix

## Possibilities

### Cholesky decomposition

$\Sigma = LL^T$ where L is low-triangular

### Eigendecomposition

$\Sigma = (Q\Lambda^{\frac{1}{2}})(Q\Lambda^{\frac{1}{2}})^T$ where $\Lambda$ is matrix with eigenvalues on diagonal and 0

# Correlated numbers generation

## Practical scenario:

There's a correlation matrix given. However, an expert sets some of the negative correlations to 0 (reality check). We need to know if the adjusted matrix is still positive semidefinite.

## Approach

The smallest eigenvalue must be positive.

## Code example

```
R <- matrix(c(1,.5,.5,1), nrow = 2)
print(min(eigen(R)$value))

          [,1]      [,2]
[1,] 1.0000000 0.4792674
[2,] 0.4792674 1.0000000
```

# Correlated numbers generation - II

## Refresher: fact from the probability theory

Let $\xi \in \Phi_{0_n, I_n}$ and $\Sigma = AA^T$ Then $A\xi \in \Phi_{0_n, \Sigma}$

## Code example

```
R <- matrix(c(1,.5,.5,1), nrow = 2)
EG <- eigen(R)
mx <- EG$vectors %*% diag(sqrt(EG$values))
V <- matrix(rnorm(1000), nrow = 2)
print(cor(t(mx%*%V)))
```

# Computation of quantile functions - I

## Given

- Given: F() - cdf, probability $y$
- Find: quantile $x$, s.t. $\mathrm{y} = \mathrm{F}(\mathrm{x})$

## No closed form solution examples

- Normal distribution (not even cdf is given in elementary functions!)
- Gamma distribution

## Quantile function given

If the quantile function is given, it's better to use its Taylor expansion

## Example

Normal cdf is implemented in practice as a piecewise Taylor polynom, i.e. with coefficients varying on different intervals.

# Computation of quantile functions - II

## Problem

Many algorithms require an interval to be defined, however, the quantile function are often defined on unconstrained intervals.

## Example

- Find a quantile for Gamma distribution
- Problem: the right end of domain is unconstrained, additionally, the root finding algorithm might not converge in the tail
- Solution: use Chebyshev's inequality to constrain the domain

# Computation of quantile functions - III

## Application

### Inequality

$P(|X-\mu| \geq 10\sigma) = 0.01$

### In numbers

- Default rate 2%, $\theta = 1$
- $P(|X - 0.02| \geq 10 \times 0.02) = 0.01$
- Thus, right bound 0.22. If x is bigger than 0.22, then set it hard to 0.99 (if precision in the tail is not important)

## Inequality application

$P(|X-\mu| \geq 10\sigma) = 0.01$

# Computation of quantile functions - IV

## Problem

Find quantile of Gamma distribution using uniroot procedure

## Solution

```
pg <- function(x) pgamma(x, 0.02, 1) - 0.95
uniroot(pg, c(0,0.22))$root
# check: qgamma(0.95, 0.02, 1)

[1] 0.04592691
```

# Case - Constrained distribution

- Given: normal distribution is used to simulate the collateral value
- Data: the mean and standard deviation are used from historical observations
- Proposal: set the negative outcomes to 0

# Case - Constrained distribution

## Problem

- the mean and standard deviation would shift after cutting
- i.e. we need to figure out new parameters, s.t. they would give the historical mean and deviation after truncation

## Solution

- calculate new mean and standard deviations
- luckily, the mean and deviation have a close-form solution
- $\hat{\mu} = \mu + \eta$
- $\hat{\sigma}^2 = \sigma^2 - \mu\eta + \eta^2$
- where $\eta = \sigma \frac{\phi(-\frac{\mu}{\sigma})}{1 - \Phi(-\frac{\mu}{\sigma})}$
- However, no straight-forward way to invert (and we have a system of two equations - one for mean and one for deviation)

# Case - Constrained distribution

## Numerical inversion of a system of equations

- One of possible options is to minimize the sum of error squares
- $(\hat{\mu}(\mu, \sigma) - \mu_0)^2 + (\hat{\sigma}(\mu, \sigma) - \sigma_0)^2 \rightarrow \min_{\mu,\sigma}$
- require a multi-dimensional optimization (e.g. gradient descent)

# Case - Constrained distribution

**Some conclusions: a simple (and meaningful) requirement led to:**

- Mathematical calculations (still feasible)
- Multivariate optimization (with some numerical tinkering)

**Homework if desired**

- How the equations would change if the collateral can be at most its notional (i.e. right-bound)