Assignment1: Take Home Messages

- General suggestions

  - ✓ Team-working Golden Rules:

Maximum collaboration within the group, <u>zero</u> between groups!

Always review the code written by others and standardize it!

- Docs:
  - ✓ Some have nice graphical representation (often more useful than hundred of words);
  - ✓ answers are not unique: find your own;
  - ✓ Price results: in percentage or currency amount;
  - ✓ Avoid theoretical formulas when not specified by homework: go to the point!

- Codes:

  ✓ Each code has the style of the people that have programmed it: the probability (for not elementary tasks) to find the same structure in the code is zero, and also to observe the same error!

  ✓ Remember (for your opportunity) to put adequate comments in codes.

  ✓ use at the beginning of the code (if needed): clear (cancels all previously defined variables) & clc (clears the screen). Use an header for the run_ script; not a proper start:

```
%% Pricing parameters
```

  ✓ (often) better using low-level functions (when possible) more than built-in functions; it can be implemented easily in a different language

  ✓ Initialize random variable generator seed to fix results number in debugging mode (**rng**).

  ✓ Dimension vectors before a cycle with zeros or ones; use cycles only if it is absolutely needed! Otherwise exploit Matlab vectorized calculations.

How to compute a confidence interval in a MC simulation ("Error" estimation)

Consider a set of i.i.d. rvs with mean μ and variance σ^2 $\{o_i\}_{i=1,\ldots,M}$

- Estimator:
  - ✓ $\hat{o} := \frac{1}{M}\sum_{i=1}^{M} o_i$
  - ✓ is unbiased

- Variance of the estimator:
  - ✓ $\mathbb{V}[\hat{o}] = \mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}(o_i - \mu)\right]^2 = \frac{1}{M^2}\sum_{i=1}^{M}\mathbb{E}[o_i - \mu]^2 = \frac{\sigma^2}{M}$

…i.e. the central limit holds

- Estimator of the variance of the estimator, when μ is not known
  - ✓ $\hat{S} := \frac{\hat{s}_2}{M} = \frac{1}{M(M-1)}\sum_{i=1}^{M}(o_i - \hat{o})^2$ which is unbiased

…the other solution adopted…

What is a reasonable accuracy for an Equity/FX plain vanilla option?

A criteria for prices: the basis point.

(*in general lower than the average observed bid-ask spread*)

- Vectorialization:

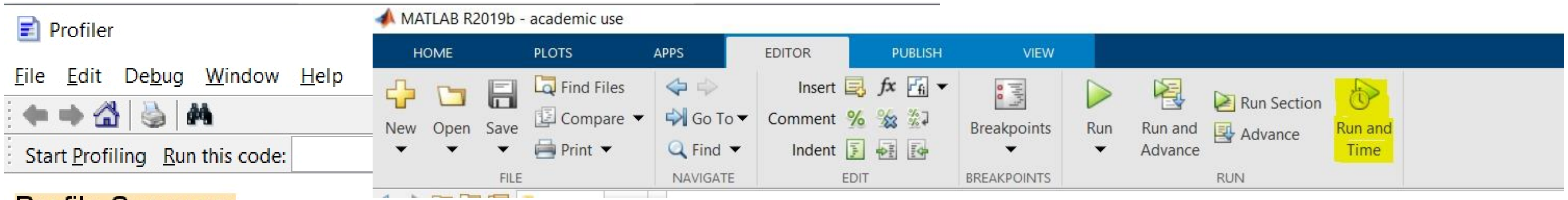
```
value = max(u.^(N:-2:-N) * F0 -K,0);
```

vs

```
value = zeros(N+1,1);
for i=0:N
   value(i)=max(u^(N-2*i)*F0 -K,0);
end
```

- MC is the most flexible technique
  - ✓ whatever payoff can be easily implemented
- computational time in Matlab
  - ✓ tic & toc.

Matlab profiler

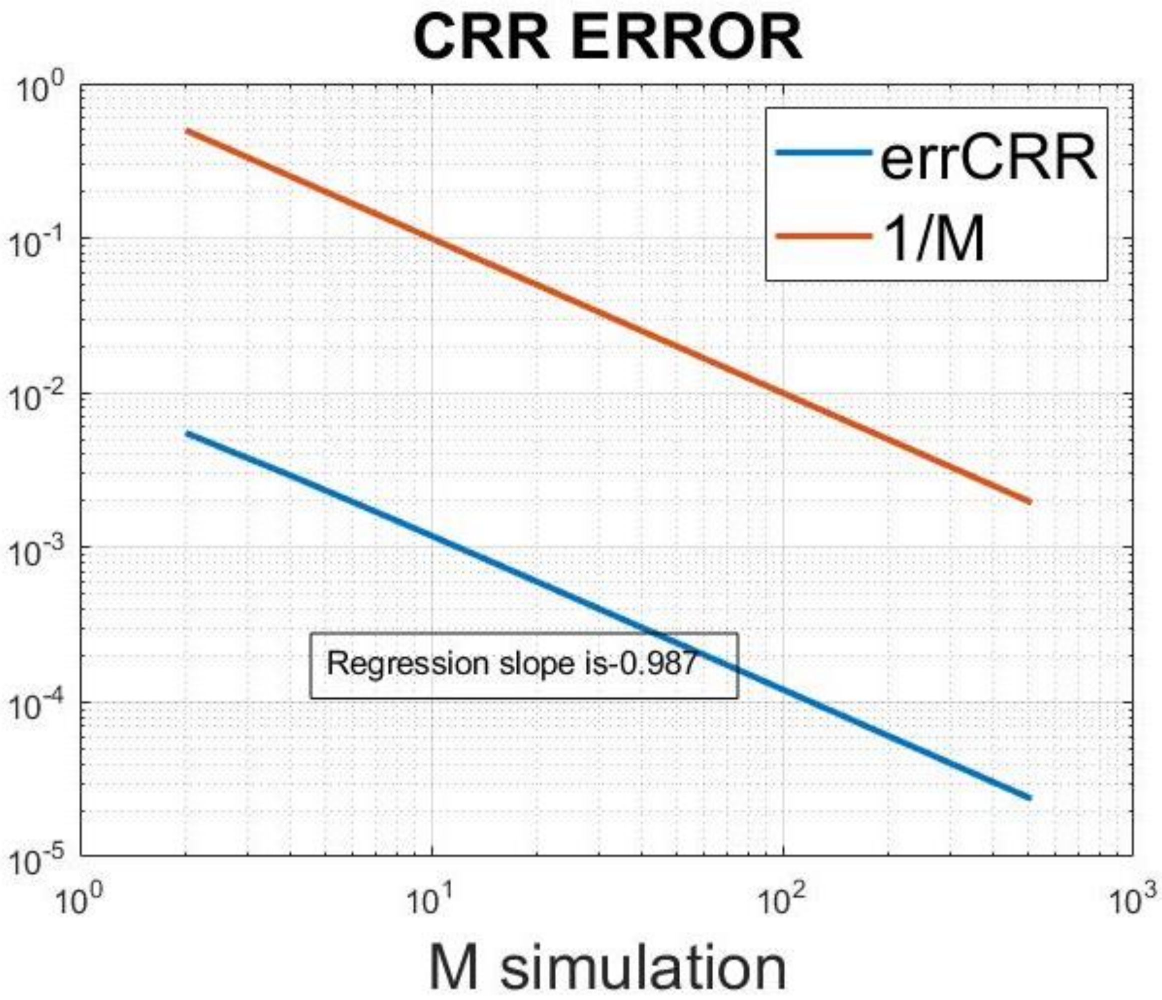


Profile Summary

Generated 25-Feb-2023 17:05:40 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
runAssign1_Group2	1	372.270 s	0.449 s	
GammaKO	4	366.169 s	0.017 s	
EuropeanOptionKOMC	50	364.377 s	74.740 s	
heaviside	50000250	288.945 s	288.945 s	
EuropeanOptionPrice	9	3.586 s	0.010 s	
EuropeanOptionMC	3	3.457 s	3.453 s	
sympref	151	2.204 s	0.076 s	
mupadmex (MEX-file)	11417	1.988 s	1.897 s	
EuropeanOptionKOCRR	100	1.713 s	0.156 s	
sympref>setSMTpref	150	1.102 s	0.085 s	
sym.sym>sym.sym	2588	0.954 s	0.074 s	
legend	6	0.932 s	0.021 s	
legend>make_legend	6	0.908 s	0.053 s	
sympref>implementedSymprefs	765	0.756 s	0.058 s	
...engine>mupadengine.evalin_internal	608	0.561 s	0.021 s	
sympref>setMuPADpref	157	0.549 s	0.017 s	
sympref>expectedPrefType	456	0.511 s	0.035 s	
sympref>symprefValidator	150	0.473 s	0.022 s	

Accuracy: Error Rescaling




```
p=polyfit(log10(M),log10(errorCRR),1); %lin regress on log-log scale
```

European Barrier Up&In Call

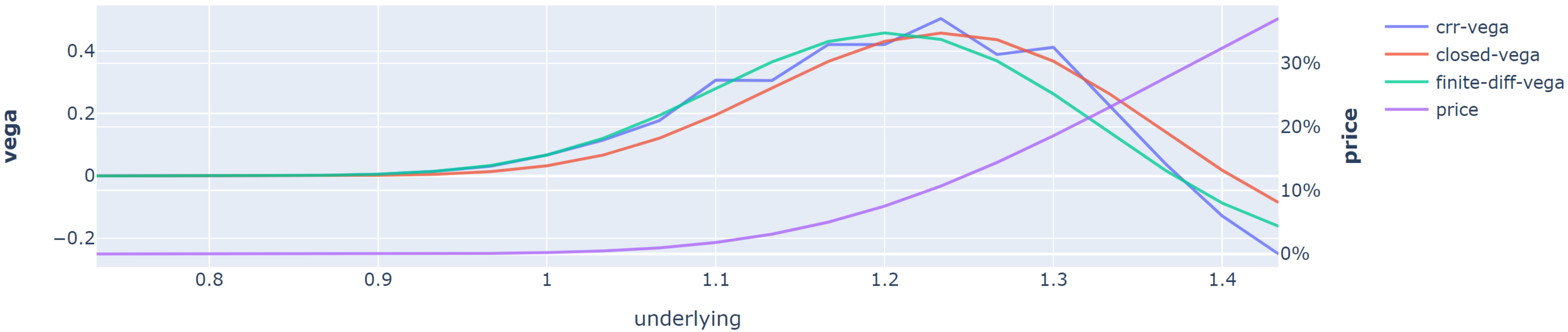
$N_{CRR} = 1000$

$I_{S0} = 100$

$\Delta\sigma = 2\%$

Vega: Digital Risk

European Barrier Up&In Call



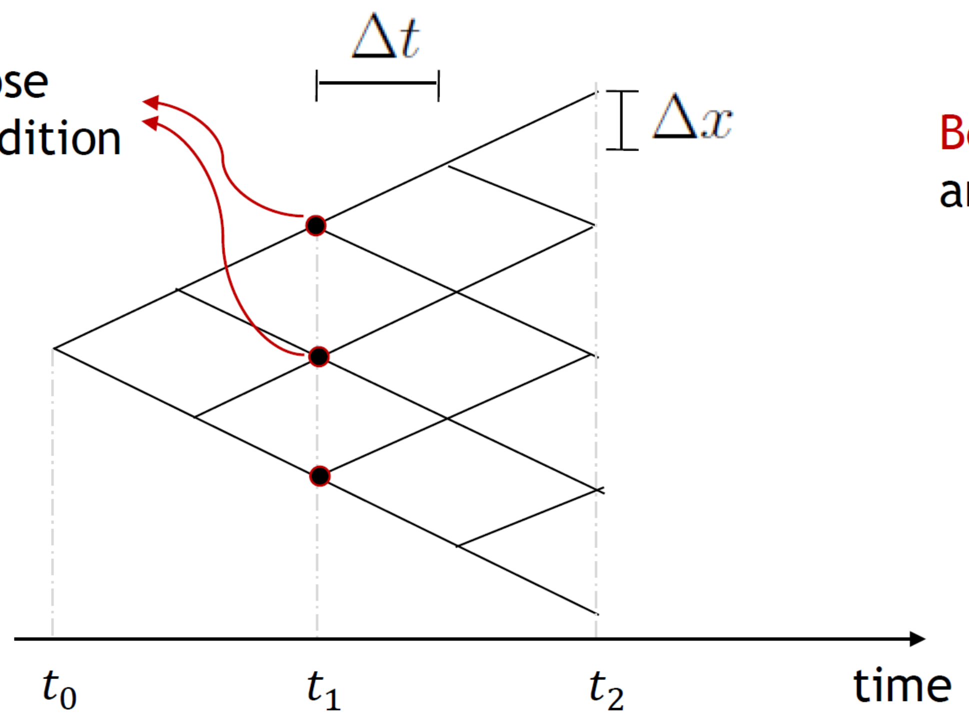
$N_{CRR} = 1000$

$I_{S0} = 25$

$\Delta\sigma = 2\%$

Discrete Time (CRR): How to use it in a Bermudan option

points where impose
the additional condition



Bermudan option $C_B(t_0, t_2)$ with an additional exercise date in t_1

going backwards, in each point of t_1 one should impose the additional condition

$$\boxed{C_B(t_1, t_2) = max[S(t_1) - K, C(t_1, t_2)]}$$

with $\quad F(t_1, t_2) = S(t_1)\, e^{(r-q)(t_2 - t_1)} \quad$ in presence of a dividend yield

Accuracy: Antithetic Variables

Antithetic Variables: a simple recipe to increment accuracy in MC

Use the Gaussian r.v. twice: the drawn variable and with opposite sign.

```
Z  = randn(N,1);
F  = F0*exp(-0.5*sigma^2*T + sigma*sqrt(T)*Z);
F_ = F0*exp(-0.5*sigma^2*T - sigma*sqrt(T)*Z);
simulatedPrice = ones(2,N);
simulatedPrice(1,:) = B * max(0, F - K);
simulatedPrice(2,:) = B * max(0, F_ - K);
mean_price = mean(mean(simulatedPrice,1));
```

Main advantages:

- uses 2N variables with N drawings;
- $Var\left[\frac{X+Y}{2}\right] = \frac{1}{2}\{Var[X] + Cov(X,Y)\} < Var[X]$, w.r.t. Cov(X,Y) < 0
- the “sperimental” distribution is automatically symmetric

Numerical Gamma: the impact of seed

- Fixing the seed in MonteCarlo:

$$C^{MC}(F_0+h) = B(0,T)\,\frac{1}{M}\sum_{i=1}^{M} f(F_0+h;\ \boldsymbol{g_i})$$

$$C^{MC}(F_0) = B(0,T)\,\frac{1}{M}\sum_{i=1}^{M} f(F_0;\ \boldsymbol{g_i})$$

$$\Gamma^{num}(F_0) = \frac{1}{h^2}\{C^{MC}(F_0+h) - 2C^{MC}(F_0) + C^{MC}(F_0-h)\}$$

Then:

$$\Gamma^{num}(F_0) = \frac{1}{h^2}\,B(0,T)\,\frac{1}{M}\sum_{i=1}^{M}\{f(F_0+h;\ \boldsymbol{g_i}) - 2f(F_0;\ \boldsymbol{g_i}) + f(F_0-h;\ \boldsymbol{g_i})\}$$

$$\Gamma^{num}(F_0) = \frac{1}{M}\sum_{i=1}^{M}\{\Gamma^{exact}(F_0;\boldsymbol{g_i}) + o(h)\} = \frac{1}{M}\sum_{i=1}^{M}\Gamma^{exact}(F_0;\boldsymbol{g_i}) + o(h)$$

Bermudan vs European