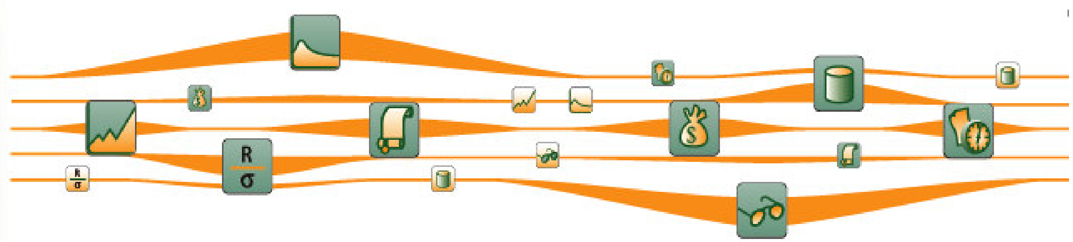
FINALYSE

Composing Solutions for Finance



Option Pricing using Numerically Evaluated Characteristic Functions

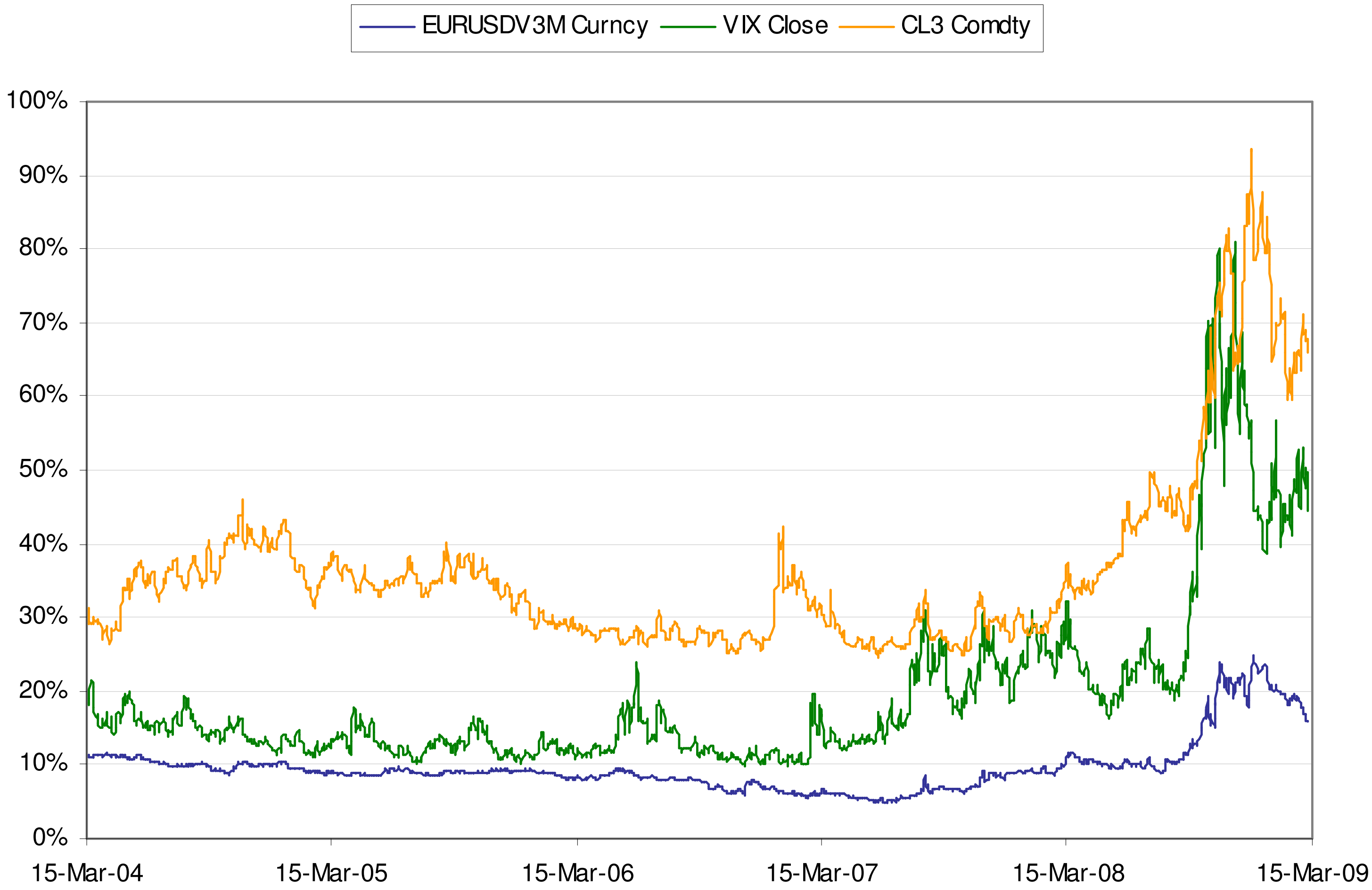
Péter Dobránszky

16th April, Paris

Agenda presentation

1. Modelling Financial Asset Price Dynamics
2. Affine Jump-Diffusion Processes
3. Solving the Riccati Equations
4. Option pricing by Fourier Inversion
5. Performance
6. Summary and Questions

Stochastic volatility

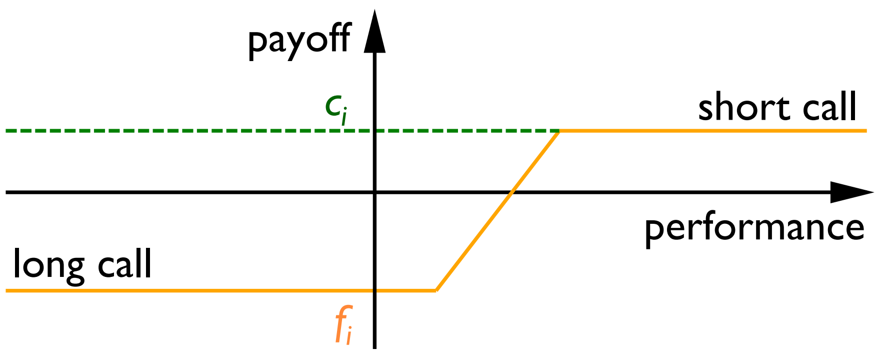


Foreign exchange, equity and energy volatility references

Cliquet spreads

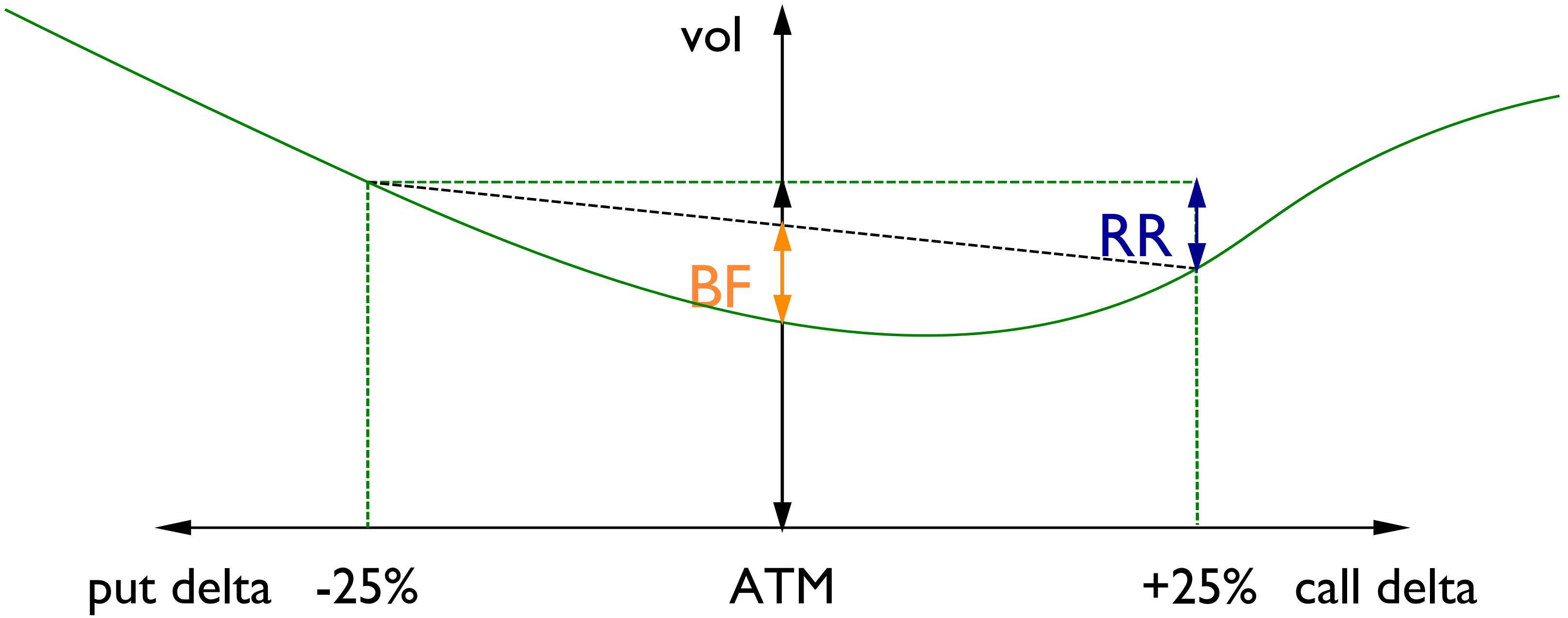
- Set of forward starting performance spread options

$$payoff = \sum_i \max\left(f_i, \min\left(c_i, \frac{S_i - S_{i-1}}{S_{i-1}}\right)\right)$$

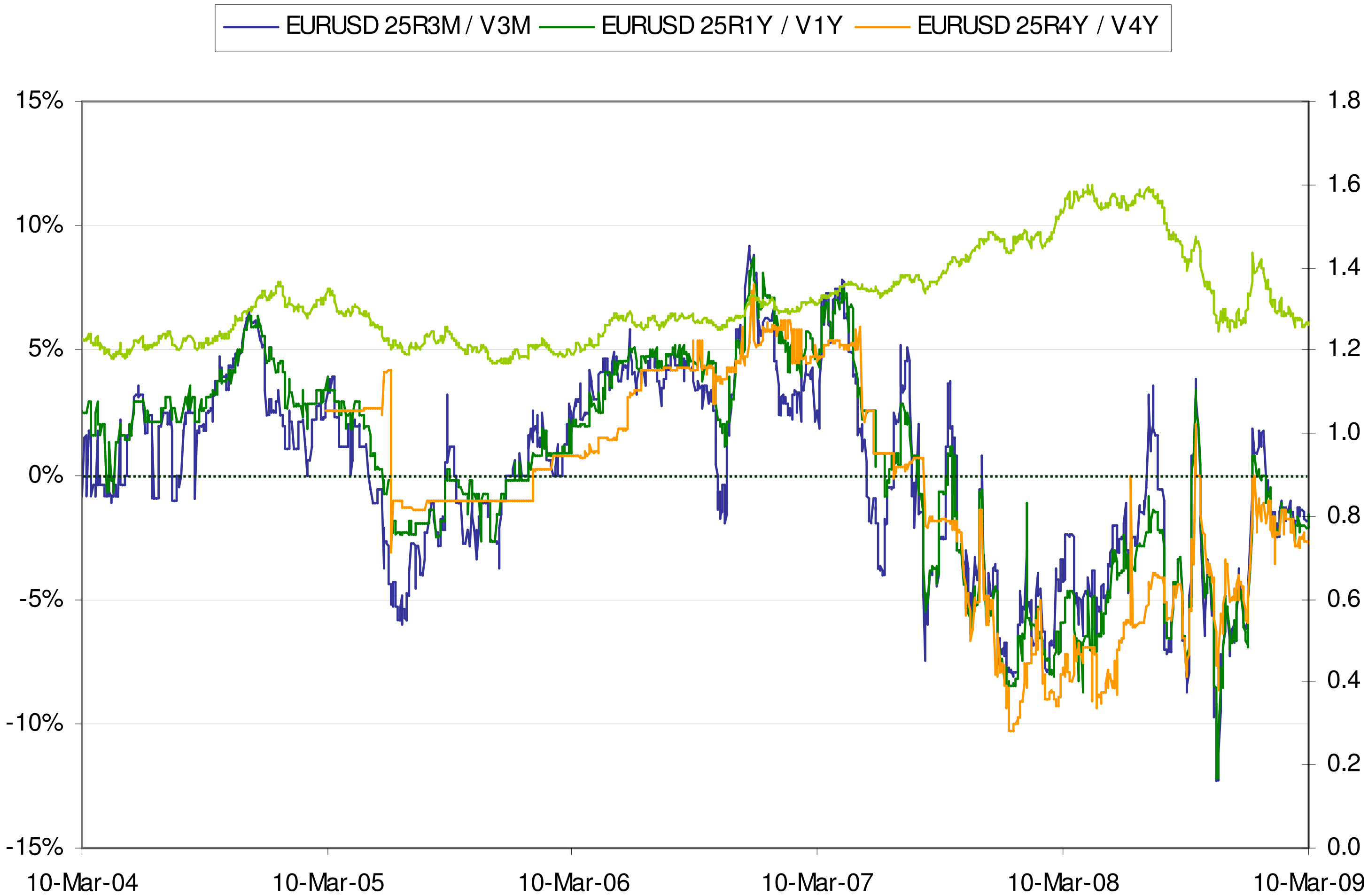


- Price not deductable from plain vanillas → we need a model
- Model should deliver forward skew → forget local volatility
- Forward start → mainly delta neutral
- Spread option → use strikes to set them vega neutral
- More or less delta and vega neutral, where is the risk then?

Quoting the smile by delta



Stochastic skewness

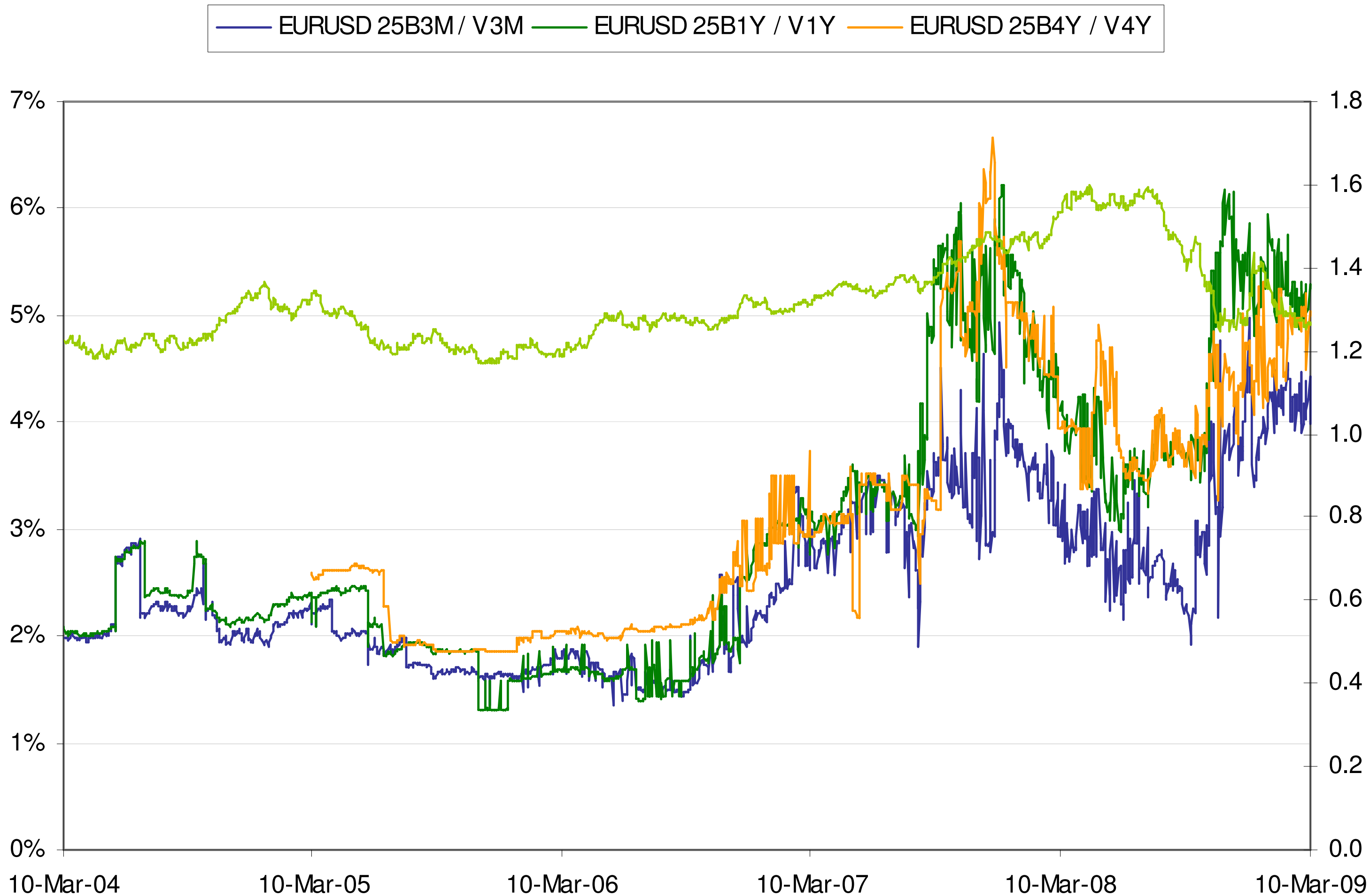


EUR/USD risk reversals over ATM volatility levels

Stochastic skew models

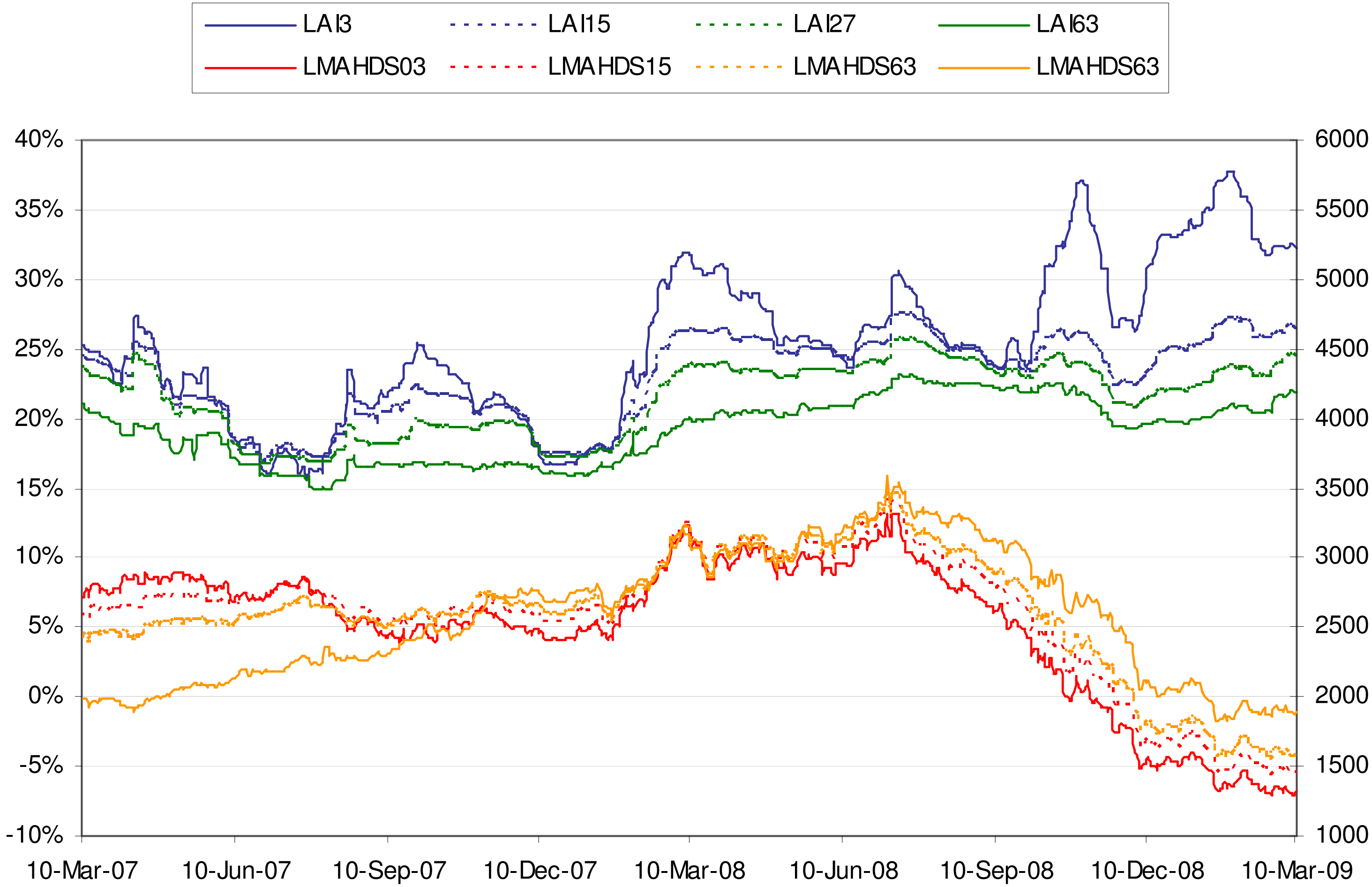
- Empirically the level and the slope of the volatility smirk fluctuate largely independently
  - Forex: distributions are usually skewed to the weaker currency, the direction of the strength, thus the sign of the skew may change
  - Equity: default expectation, risk-averseness and jump-to-default premium are stochastic, thus the level of skew may change
  - Rates: anticipated central bank actions may imply significant skew, also the sign of the skew may change
  - Commodity: upside jumps are sometime more probable than downside jumps, also the sign of the skew may change
- Focus on the stochastic correlation between asset and variance returns

Stochastic volatility of volatility



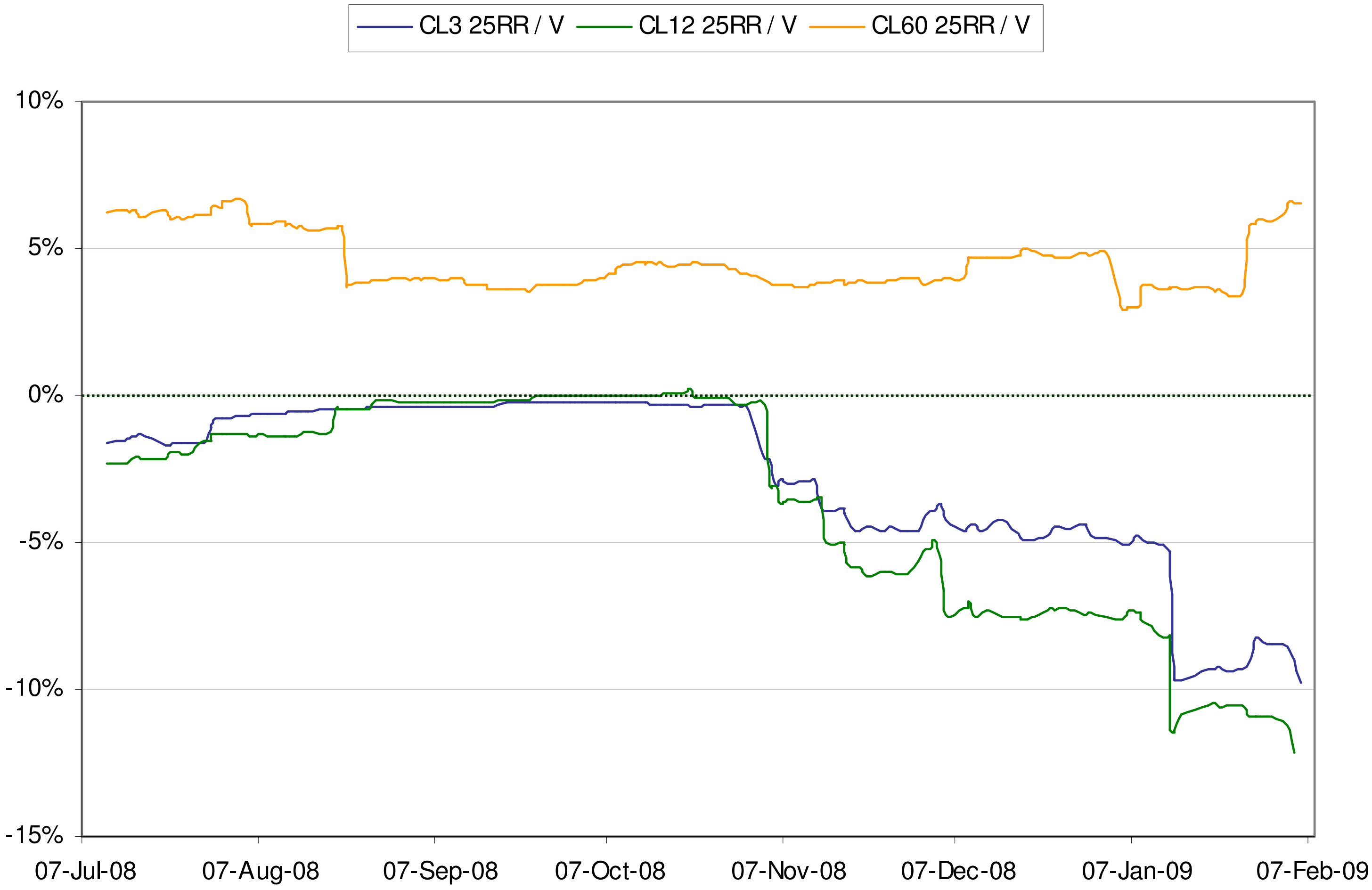
EUR/USD butterflies over ATM volatility levels

Mean-reverting asset prices



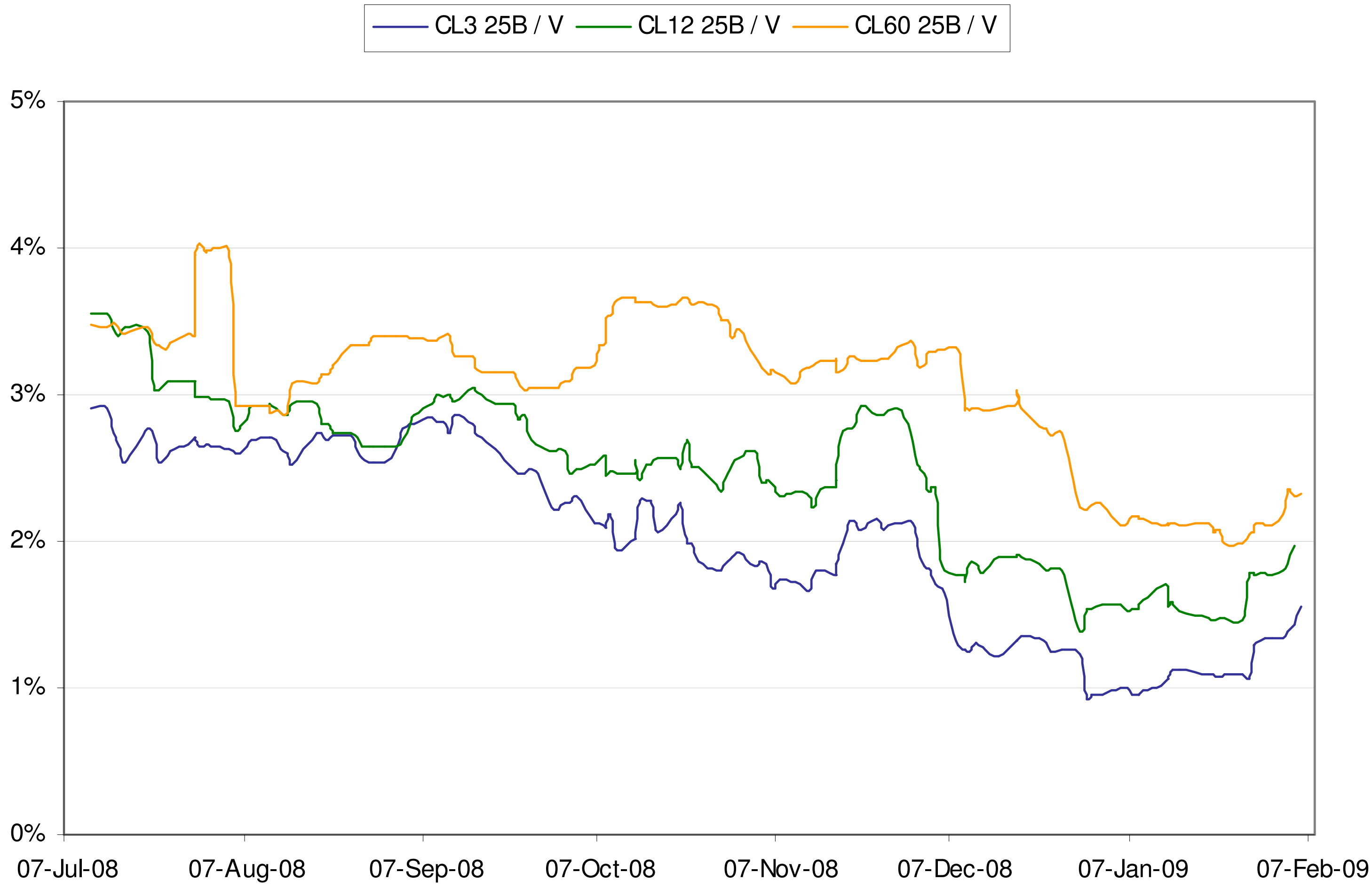
Primary aluminium (AHD) futures and implied volatilities

Term structured stochastic skewness



WTI light sweet crude oil (CL) risk reversals over ATM volatility levels

Volatility - smile - maturity relationship



WTI light sweet crude oil (CL) butterflies over ATM volatility levels

Commodity modelling requirements

- Mean-reversion in asset prices – short-term, long-term
  - Stochastic convenience yield
  - Decreasing volatility term structure
- Multi-factor stochastic volatility – short-term, long-term
  - Volatility smile also on long-term
  - Unspanned stochastic volatility (cannot model the skew changes)
  - Equilibrium volatility level is stochastic also
- Jumps
  - Discontinuous asset path
  - Closer futures jump larger than longer futures
- Stochastic mean-reverting jump frequency
  - Stochastic implied volatility skew
  - Reduce the need for stochastic volatility

Agenda presentation

1. Modelling Financial Asset Price Dynamics
2. Affine Jump-Diffusion Processes
3. Solving the Riccati Equations
4. Option pricing by Fourier Inversion
5. Performance
6. Summary and Questions

Affine jump-diffusion models

$$d\mathbf{X}_t = \left(K_0 + K_1 \cdot \mathbf{X}_t\right)dt + \sigma\left(\mathbf{X}_t, t\right)d\mathbf{W}_t^{\mathbb{Q}} + d\mathbf{J}_t$$

$$\left(\sigma\left(\mathbf{X}_t, t\right)\sigma\left(\mathbf{X}_t, t\right)^T\right)_{ij} = H_{0ij} + H_{1ij} \cdot \mathbf{X}_t$$

$$\Lambda_t = l_0 + l_1 \cdot \mathbf{X}_t$$

$$\theta_\nu\left(u\right) = \int \exp\left(u \cdot z\right)d\nu\left(z\right)$$

$$H_{0ij}, l_0 \in \mathbb{R}, K_0, H_{1ij}, l_1 \in \mathbb{R}^N, K_1 \in \mathbb{R}^{N\times N}$$

Affine transform

$$\psi^{\mathbf{X}}\left(u,\mathbf{X}_t,t,T\right)= E^{\mathbb{Q}}\left[e^{-\int_t^T(\rho_0+\rho_1\cdot\mathbf{X}_u)du+u\cdot\mathbf{X}_T}\middle|\mathcal{F}_t\right]= e^{\alpha(t)+\beta(t)\cdot\mathbf{X}_t}$$

with alpha and beta satisfying the following complex-valued matrix Riccati equations

$$\frac{d\beta(t)}{dt}=\rho_1-K_1^T\beta(t)-\frac{1}{2}\beta(t)^T H_1\beta(t)-l_1\left(\theta\left(\beta(t)\right)-1\right)$$
$$\frac{d\alpha(t)}{dt}=\rho_0-K_0^T\beta(t)-\frac{1}{2}\beta(t)^T H_0\beta(t)-l_0\left(\theta\left(\beta(t)\right)-1\right)$$

with boundary conditions

$$\beta(T)=u,\quad \alpha(T)=0$$

Affine extended transform

$$\phi^{X}\left(v,u,\mathbf{X}_{t},t,T\right)=E^{\mathbb{Q}}\left[v\mathbf{X}_{T}\cdot e^{-\int_{t}^{T}\left(\rho_{0}+\rho_{1}\cdot\mathbf{X}_{u}\right)du+u\cdot\mathbf{X}_{T}}\middle|\mathcal{F}_{t}\right]=$$
$$=\psi^{X}\left(u,\mathbf{X}_{t},t,T\right)\cdot\left(A\left(t\right)+B\left(t\right)\mathbf{X}_{t}\right)$$

with A and B satisfying the following complex-valued matrix Riccati equations

$$\frac{dB\left(t\right)}{dt}=K_{1}^{T}B\left(t\right)+\beta\left(t\right)^{T}H_{1}B\left(t\right)+l_{1}\nabla\theta\left(\beta\left(t\right)\right)B\left(t\right)$$
$$\frac{dA\left(t\right)}{dt}=K_{0}^{T}B\left(t\right)+\beta\left(t\right)^{T}H_{0}B\left(t\right)+l_{0}\nabla\theta\left(\beta\left(t\right)\right)B\left(t\right)$$

with boundary conditions

$$B\left(T\right)=v,\quad A\left(T\right)=0$$

Affine characteristic of log-returns

$$S_t = e^{a+b\mathbf{X}_t}$$

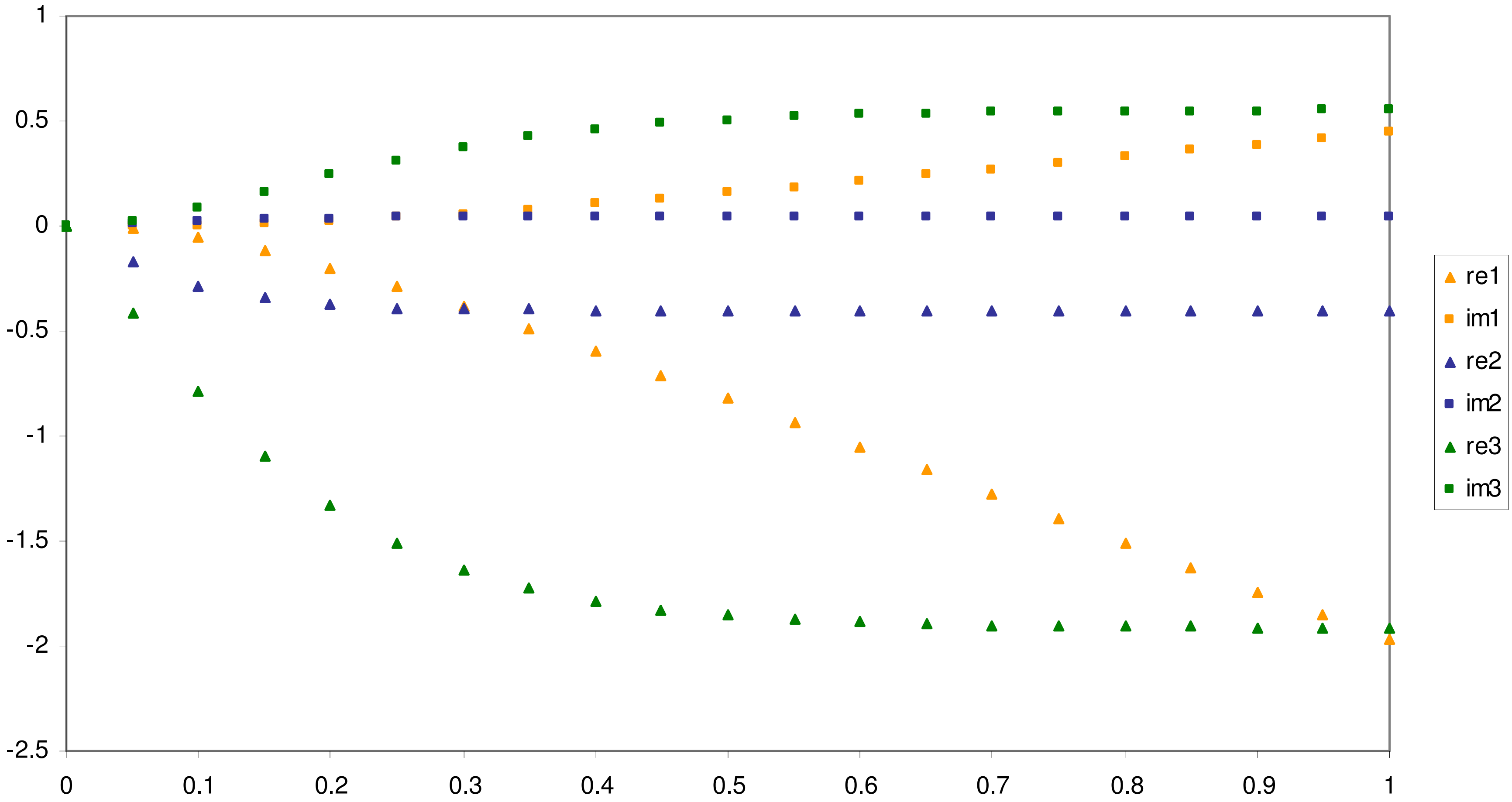
$$\varphi_{S_T}(u) = E^{\mathbb{Q}}\left[e^{iu(a+b\mathbf{X}_t)} \middle| \mathcal{F}_t \right] = e^{iua+\alpha(iub,t)+\beta(iub,t)\cdot\mathbf{X}_t}$$

- How to price vanilla options?
  - Specify the underlying affine jump-diffusion process by SDE
  - Translate SDE into Riccati equations to be solved
  - Solve the ODE either analytically or numerically
  - Use FFT or direct integration as Fourier inversion to calculate option prices

Agenda presentation

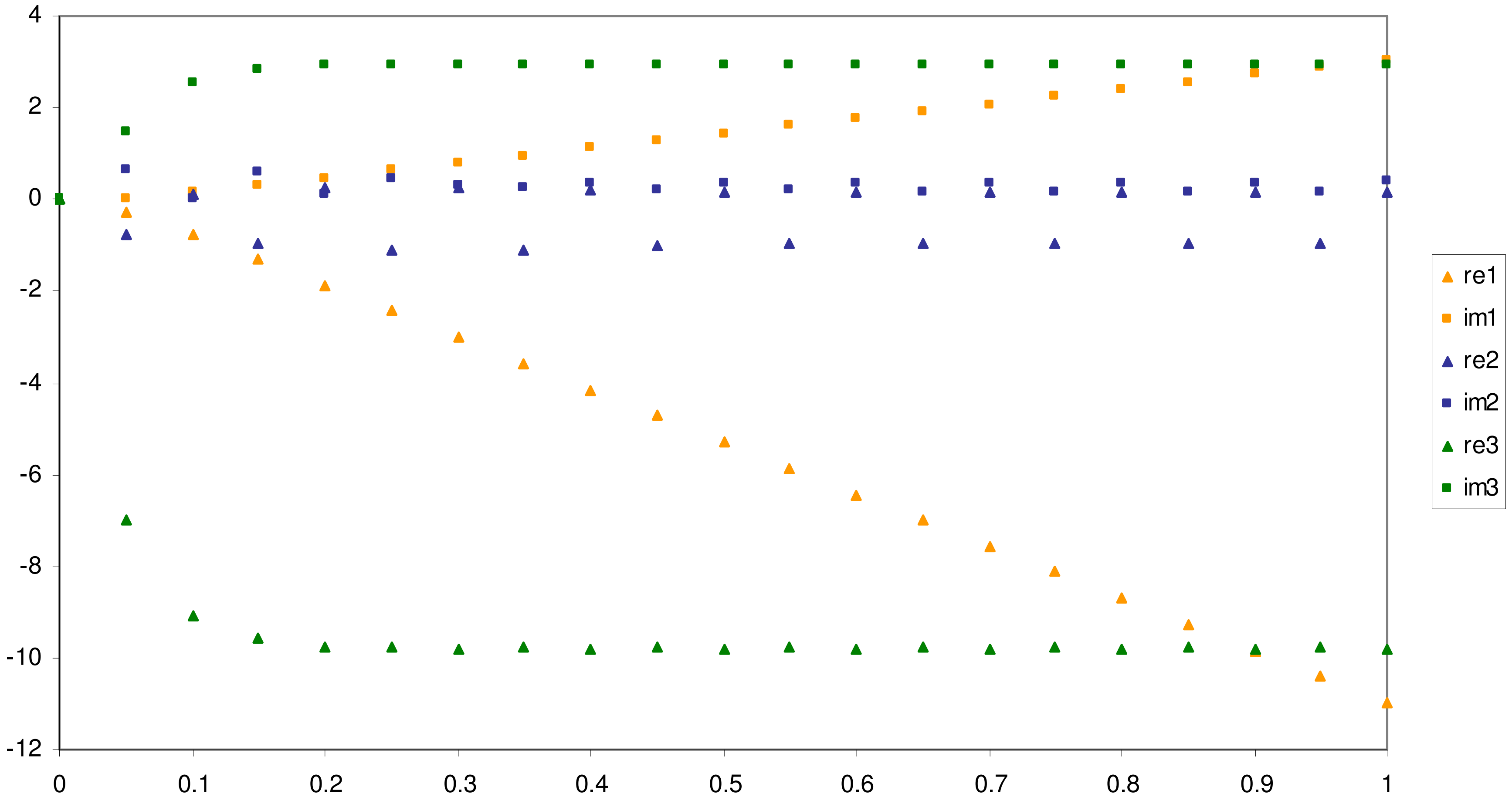
1. Modelling Financial Asset Price Dynamics
2. Affine Jump-Diffusion Processes
3. Solving the Riccati Equations
4. Option pricing by Fourier Inversion
5. Performance
6. Summary and Questions

Classical 4th order explicit Runge-Kutta



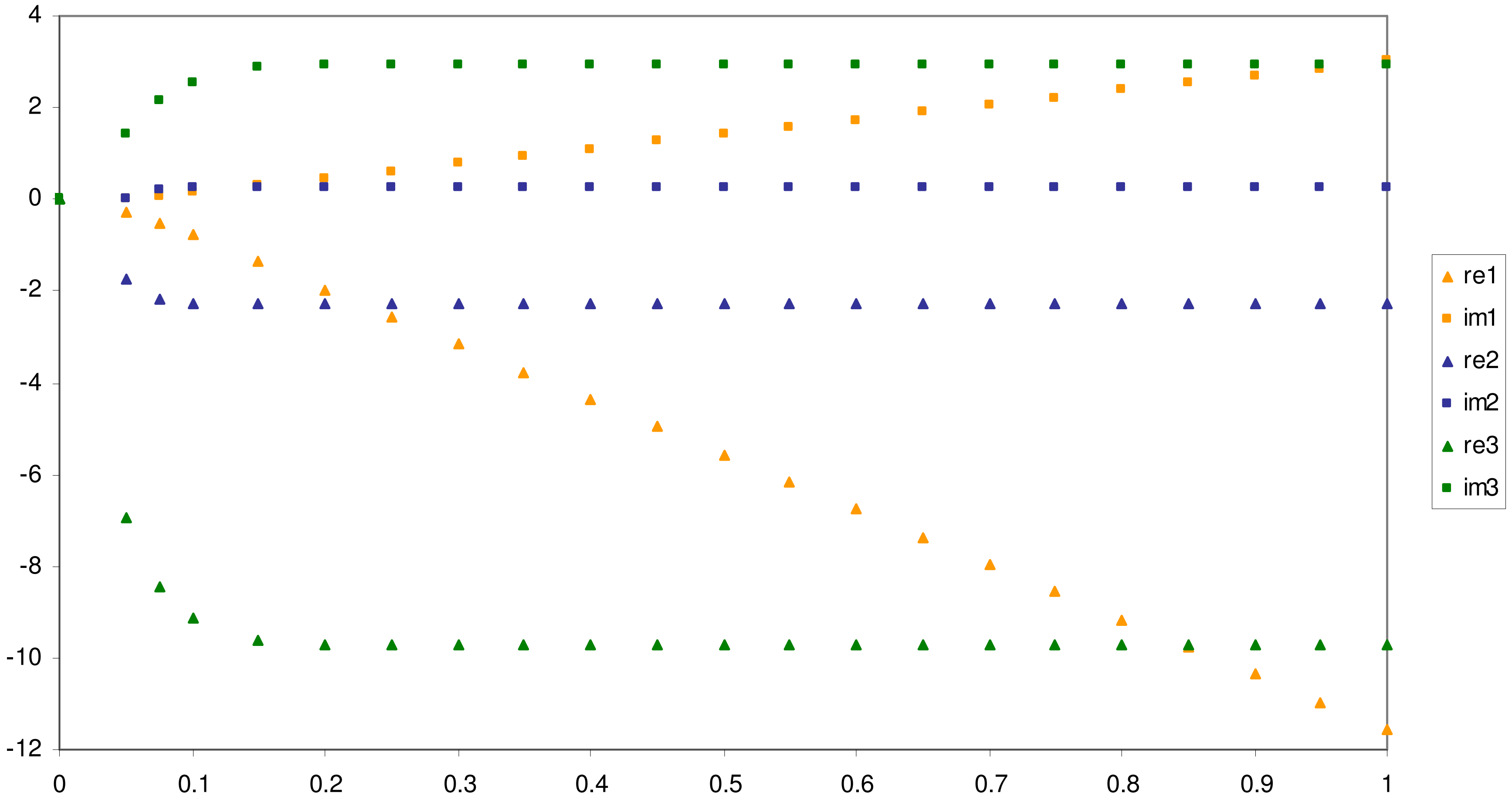
Trolle-Schwartz, u = 20, number of time steps = 20

Classical 4th order explicit Runge-Kutta



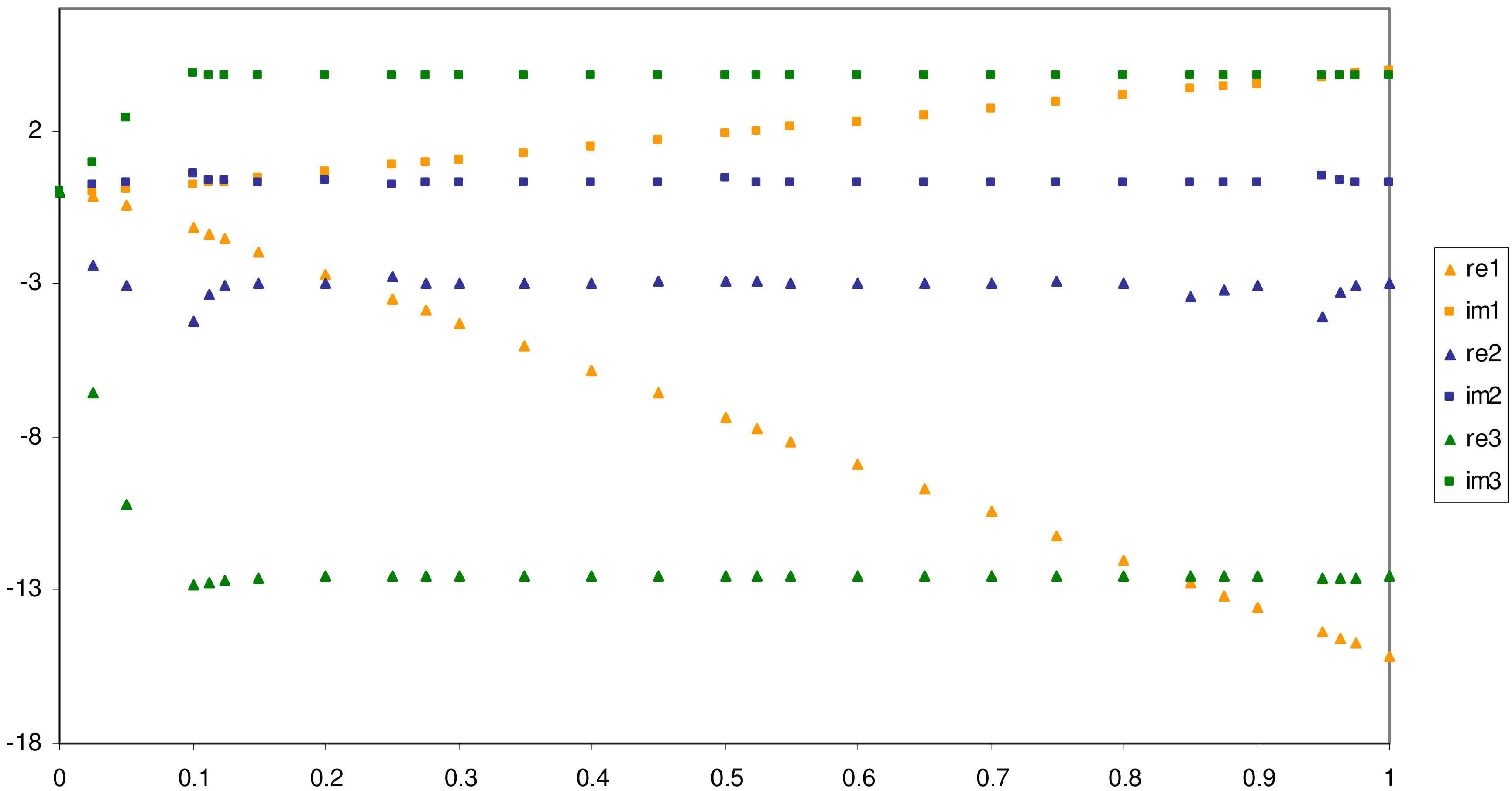
Trolle-Schwartz, u = 92, number of time steps = 20

Adaptive 4th and 5th order explicit Runge-Kutta



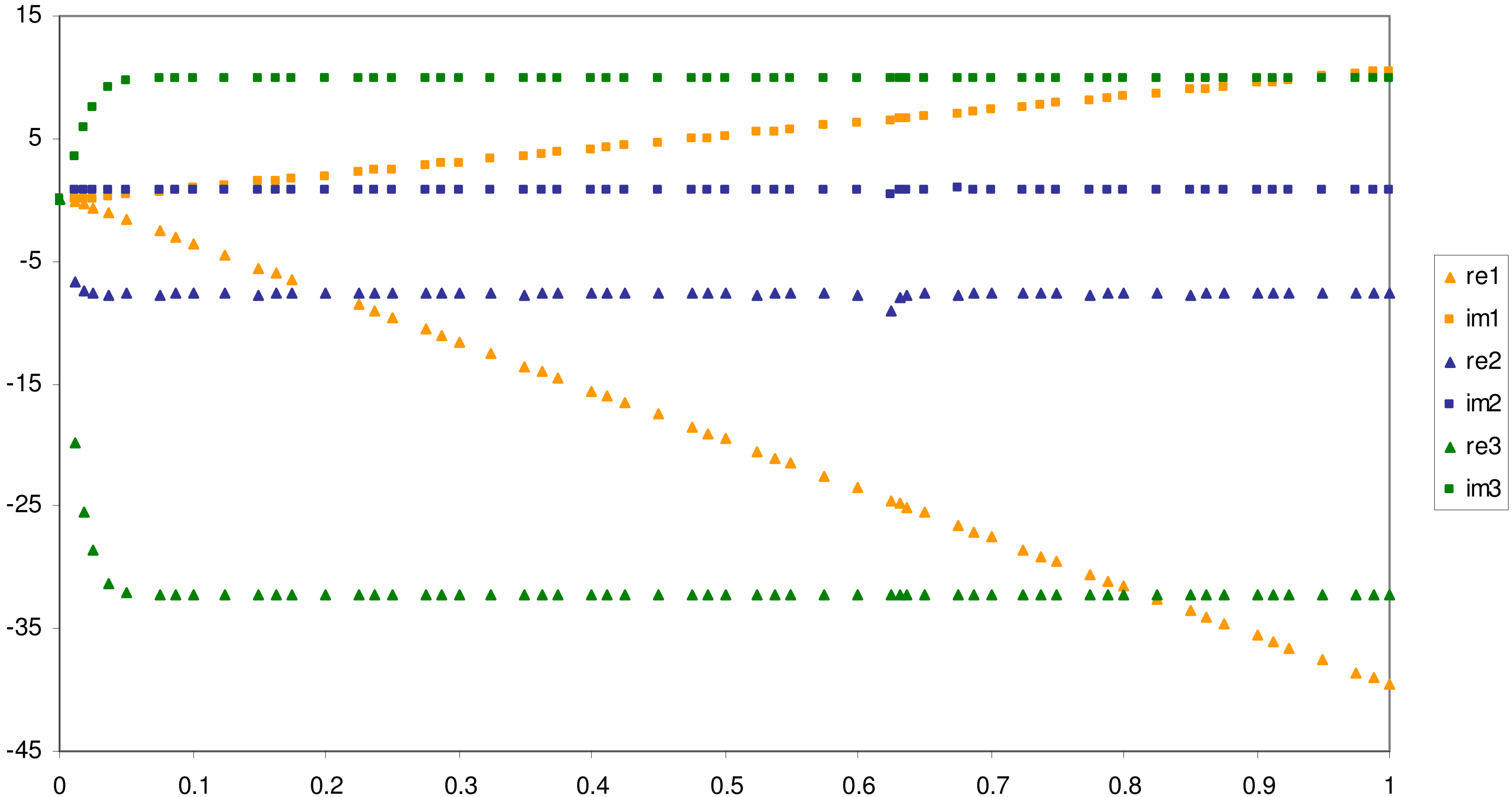
Trolle-Schwartz, u = 92, initial number of steps = 20, final number of steps = 21

Adaptive 4th and 5th order explicit Runge-Kutta



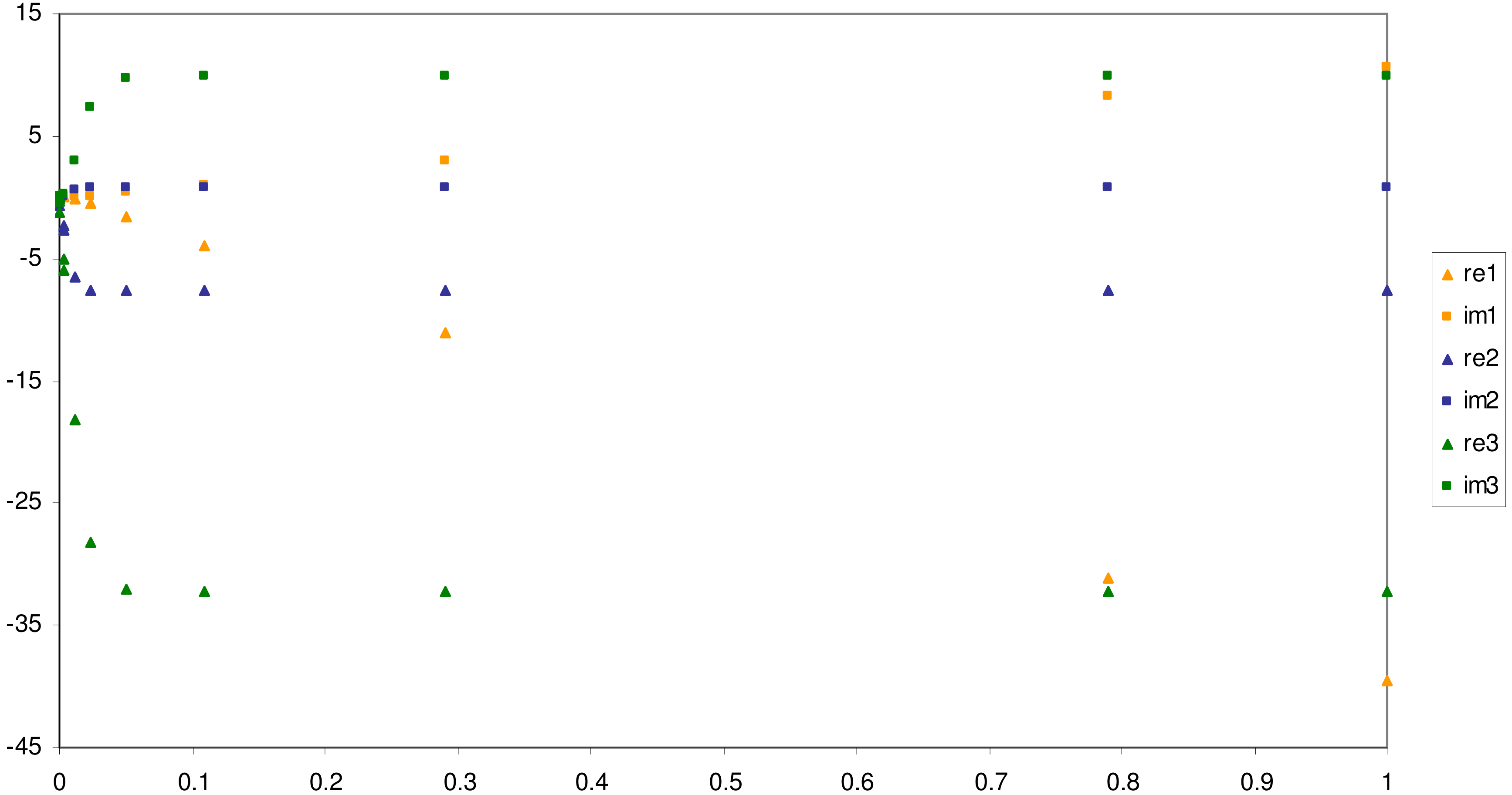
Trolle-Schwartz, u = 118, initial number of steps = 20, final number of steps = 28

Adaptive 4th and 5th order explicit Runge-Kutta



Trolle-Schwartz, u = 300, initial number of steps = 20, final number of steps = 59

Adaptive 3rd and 4th order implicit Rosenbrock



Trolle-Schwartz, u = 300, final number of steps = 12

Solve the ODE numerically

- As we use quadratures to integrate functions, use quadratures to integrate differential equations
  - Numerical Recipes in C, Chapter 16
- Explicit
  - Classical 4th order Runge-Kutta method
    - ✓ Fixed stepsize, moderate precision, 4 evaluations / step
  - Variable stepsize Bulirsch-Stoer method
    - ✓ High precision with extrapolation, good for heavy function evaluations
  - Adaptive stepsize 4th and 5th order Runge-Kutta method
    - ✓ 6 evaluations / step, adaptive stepsize
    - ✓ Weights: Runge-Kutta-Fehlberg, Cash-Karp
- Implicit
  - 3rd and 4th order Rosenbrock method
    - ✓ 1 function and derivatives evaluation / step + 1 LU decomposition + 4 back substitution
    - ✓ Weights: Kaps-Rentrop, Shampine

Solve the ODE numerically (cont.)

- Affine asset pricing models
  - Both ODE and its derivatives are closed-form
  - Polynomial form, only basic operations (+,*)
  - Dimension of the differential equation is low, <10
  - Usually stiff problem for high value of *u*

- Implicit Rosenbrock method with Shampine weights
- Minimum stepsize = initial step = 1 day
- Maximum 200 integration steps (convergence test)
- Control measure for adaptive stepsize control
  - Accept or reject the last step
  - Decide about the size of the next step

Control measure

- Calculate the largest absolute error between the 4th and the 5th order estimations – take both the real and imaginary parts
- Take the largest increment (y_i-y_{i-1}) from the last step as tolerance
- Normalize both the absolute error and the tolerance by time (x)
- Calculate proportion of tolerance / error

- If largest error is zero → accept the step
  - But, never step next more than 5 times bigger (even then we can reach 10 years in 6 steps starting with a 1 day initial step)
- If proportion bigger than 1 → accept the step
  - New step = 95% * old step * (proportion ^ 1/5)
  - Expand with lower exponent, 95% for conservativeness
- If proportion smaller than 1 → reject the step
  - New step = 95% * old step * (proportion ^ 1/4)
  - Shrink with larger exponent, 95% for conservativeness

Agenda presentation

Characteristic functions

- In probability theory it is the continuous Fourier transformation of the probability density function

$$\varphi_X(u) = E\left[e^{iuX}\right] = \int_{-\infty}^{\infty} e^{iux} f_X(x)\,dx$$

- Probability density function is the continuous inverse Fourier transformation of the characteristic function

$$f_X(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{iux}\,\overline{\varphi_X(u)}\,du$$

- For independent random variables

$$\varphi_{X+Y}(u) = E\left[e^{iu(X+Y)}\right] = E\left[e^{iuX}e^{iuY}\right] = E\left[e^{iuX}\right]\cdot E\left[e^{iuY}\right] = \varphi_X(u)\cdot\varphi_Y(u)$$

Pricing using characteristic functions

- Long call

$$c_T(K) = e^{-rT} F_T \int_k^{\infty} \left(e^x - e^k\right) q_T(x)\,dx$$

- Make an adjustment for later purposes

$$c_T(K) = e^{-rT} F_T e^{-\alpha k} \int_k^{\infty} \left(e^{x+\alpha k} - e^{k+\alpha k}\right) q_T(x)\,dx$$

- Apply the Fourier and then the inverse Fourier transform

$$c_T(K) = e^{-rT} F_T \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{\infty} e^{-ivk} \int_{-\infty}^{\infty} e^{ivk} \int_k^{\infty} \left(e^{x+\alpha k} - e^{k+\alpha k}\right) q_T(x)\,dx\,dk\,dv =$$

$$= e^{-rT} F_T \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{\infty} e^{-ivk} \psi_T(v)\,dv = e^{-rT} F_T \boxed{\frac{e^{-\alpha k}}{\pi} \int_0^{\infty} e^{-ivk}} \psi_T(v)\,dv$$

technique

Pricing using characteristic functions

$$\psi_T(v) = \int_{-\infty}^{\infty} e^{ivk} \int_{k}^{\infty} \left(e^{x+\alpha k} - e^{k+\alpha k}\right) q_T(x)\,dx\,dk =$$

$$= \int_{-\infty}^{\infty} q_T(x) \boxed{\int_{-\infty}^{x} \left(e^{x+\alpha k} - e^{k+\alpha k}\right)} e^{ivk}\,dk\,dx =$$

$$= \int_{-\infty}^{\infty} q_T(x) \frac{e^{(\alpha+1+iv)x}}{\alpha^2+\alpha-v^2+i(2\alpha+1)v}\,dx =$$

$$= \frac{1}{\alpha^2+\alpha-v^2+i(2\alpha+1)v} \int_{-\infty}^{\infty} e^{(\alpha+1+iv)x} q_T(x)\,dx =$$

$$= \boxed{\frac{1}{\alpha^2+\alpha-v^2+i(2\alpha+1)v}}\,\varphi_T\left(v-(\alpha+1)i\right)$$

payoff — process

FFT based option pricing (Carr-Madan)

$$c_T(K) = e^{-rT} F_T \boxed{\frac{e^{-\alpha k}}{\pi} \int_0^\infty e^{-ivk}} \boxed{\frac{1}{\alpha^2 + \alpha - v^2 + i(2\alpha+1)v}} \varphi_T\left(v - (\alpha+1)i\right) dv$$

technique — payoff — process

- v = ft→GetV(); // Grid in the integration space
- k = ft→GetK(); // Grid in the log-strike space
- data = payoff→GetU(v); // Get the input parameter for the CF
- cf→FromUToPhi(data); // Evaluated CF
- payoff→FromPhiToPsi(v, data); // Apply the payoff
- ft→FromPsiToIntegrand(v, data); // Get the integrand
- ft→Weightening(data); // Numerical trick for DFT
- ft→Transform(data); // Discrete Fourier transformation
- payoff→ModifyBack(k, data); // Reverse the adjustment
- ft→Interpolate(data, logStrike); // Interpolate the vector

Direct integration based option pricing

- From Jim Gatheral's book:

$$c_T(K) = e^{-rT} F_T \left(1 - \frac{e^{k/2}}{\pi} \int_0^\infty \frac{dv}{v^2 + 1/4} \mathrm{Re}\left[e^{-ivk} \varphi_T \left(v - i/2 \right) \right] \right)$$

- Advantages
  - No need anymore for equal grid steps
  - Pricing error can be targeted (eg. set to 0.1 vega in calibrations)
- Use adaptive quadratures like the adaptive Simpson method
  - Adaptive upper bound (I start with upper bound = 62.5)
- Caching if several strikes are computed at the same time
  - Vectorized version of the adaptive Simpson method

Control variate for Fourier inversion

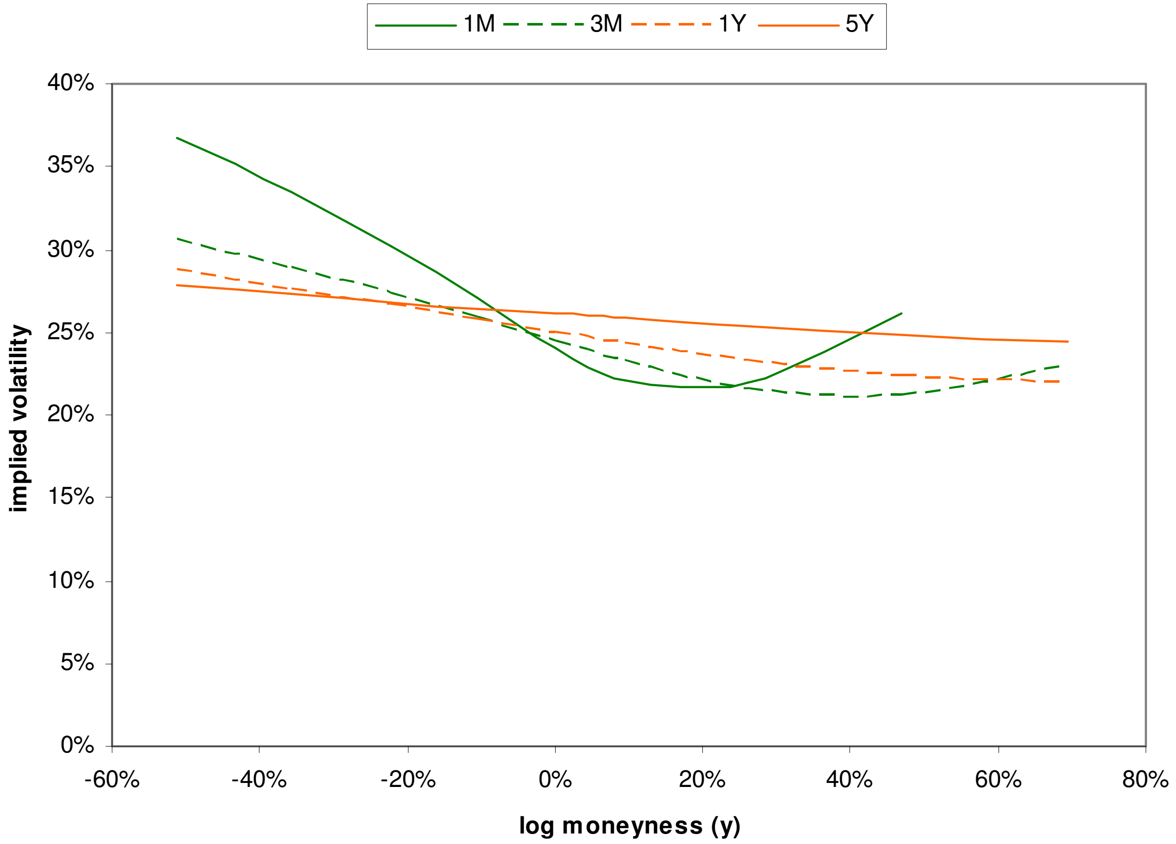
$$c_T(K) = c_T^{BS}(K) + c_T(K) - c_T^{BS}(K) =$$

$$= c_T^{BS}(K) + e^{-rT} F_T \left(1 - \frac{e^{k/2}}{\pi} \int_0^\infty \frac{dv}{v^2 + 1/4} \mathrm{Re}\left[e^{-ivk} \left(\varphi_T(v - i/2) - \varphi_T^{BS}(v - i/2) \right) \right] \right)$$

$$\sigma^{BS} = \sqrt{V - M^2} = \sqrt{\left[-\mathrm{Re}\, \varphi_T''(0) \right] - \left[\mathrm{Im}\, \varphi_T'(0) \right]^2}$$

- Calculate CF derivatives numerically (eps = 1e-5)
- Better convergence achieved both for FFT and direct integration

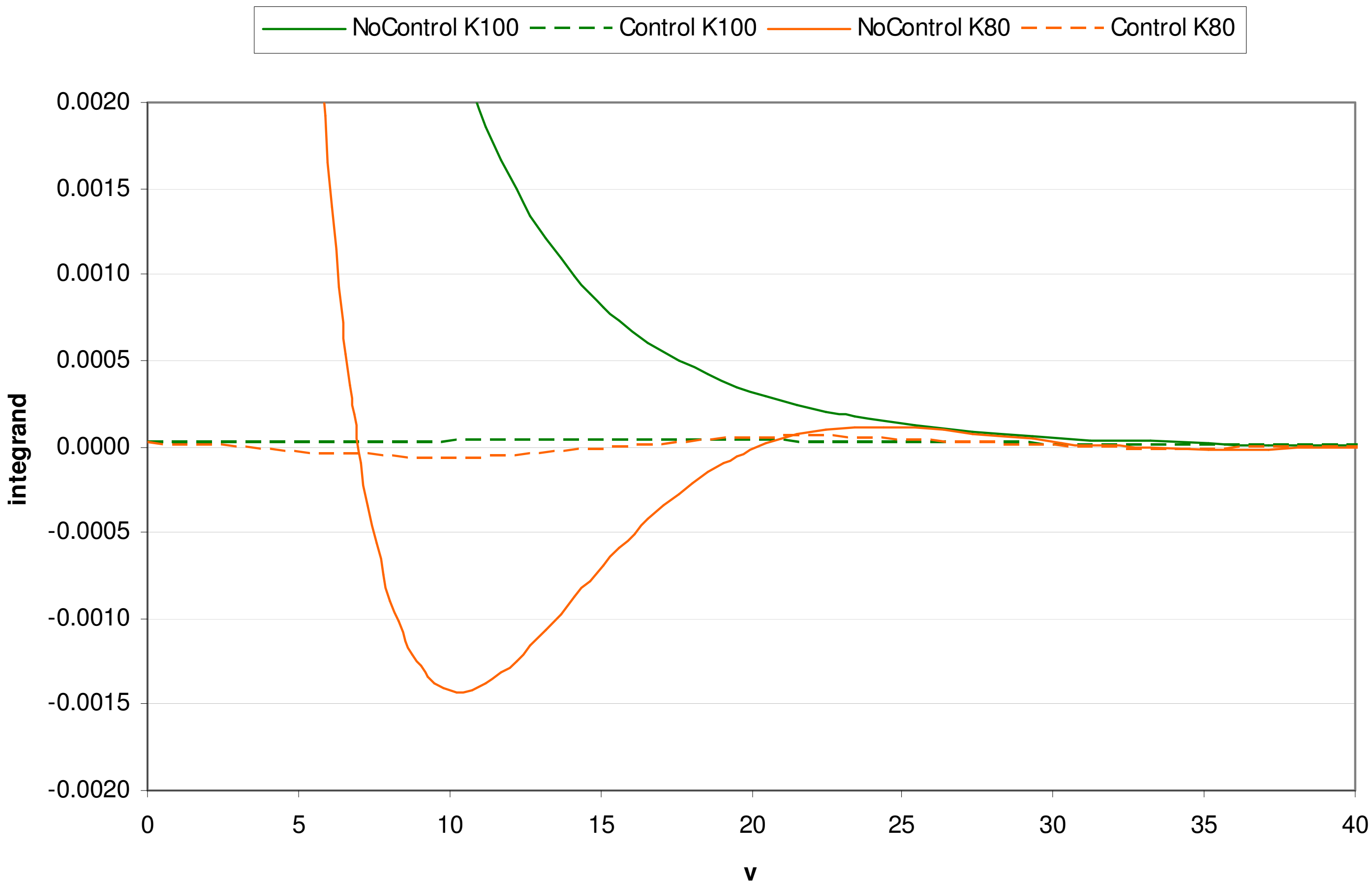
Bates = Heston + jumps



Var0 = 0.04, VarInf = 0.06, Kappa = 0.6, VolOfVol = 0.2, Rho = -0.5, JumpFreq = 5, JumpMean = -0.04, JumpVol = 0.05

For one month the standard deviation is 26%

Control variate for Fourier inversion



K = 80%: NoControl - 269 fun.eval., Control - 49 fun.eval.

K = 100%: NoControl - 265 fun.eval., Control - 21 fun.eval.

Agenda presentation

Performance measurement

- Our objective: use the technique for global calibrations
  - 5 reasonable strikes and 6 maturities per valuation dates
  - Calibrate with pricing precision of 0.1 vega (bid-ask spread ≈ ±vega)

1. Consider the Bates model with the earlier parameterization
2. Choose six tenors = {1W, 1M, 3M, 1Y, 2Y, 5Y}
3. Choose five strikes = {0.1Δ, 0.25Δ, 0.5Δ, 0.75Δ, 0.9Δ}
4. Calculate BS implied volatilities per delta per tenor
5. Calculate moneyness for each tenor
6. Calculate the 0.1 $vega_{BS}$ as targeted precision for each node

7. Measure the time to price vanilla options on the mesh (30 nodes)

Performance results

written in C++, executed on Intel 2Ghz laptop	No control variate	With control variate
Carr-Madan FFT (4096, $\alpha = 1.5$) Analytic CF	43 ms	49 ms
Carr-Madan FFT (4096, $\alpha = 1.5$) Numerical CF	301 ms	306 ms
Direct integration Analytic CF	1.05 ms	0.35 ms
Direct integration Numerical CF	6.91 ms	2.55 ms

- Control variate makes FFT slightly slower, but much more precise
- Direct integration is much faster than FFT!
- Option pricing using numerically evaluated characteristic functions is slower than using analytical ones, but not in magnitudes! (< 10 times)
- Control variate makes direct integration even faster

Logarithm of the Bates CF

$$d = \text{sqrt}\left(\left(A + B\cdot u\right)^2 + C\cdot\left(i + u\right)\cdot u\right)$$
$$e = D + E\cdot u - d$$
$$f = \exp\left(F\cdot d\right)$$
$$g = \frac{e}{D + E\cdot u + d}$$
$$h = g\cdot f$$

$$\log\varphi = F\cdot\left(G\cdot e - 2\cdot\log\frac{1-h}{1-g}\right) + H\cdot e\cdot\frac{1-f}{1-h} + \\ + I\cdot u + J\cdot\left(\exp\left(\left(K + L\cdot u\right)\cdot u\right) - 1\right)$$

- Costly: complex sqrt, complex exp (2 times), complex log

Riccati equations for the Bates CF

$$A = F \cdot u + G \cdot \left(\exp\left(\left(H + I \cdot u \right) \cdot u \right) - 1 \right)$$
$$C = \left(J + 0.5 \cdot u \right) \cdot u$$
$$D = K + L \cdot u$$

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \qquad f = \frac{dy}{dx} = \begin{bmatrix} A + B \cdot y_2 \\ C + \left(D + E \cdot y_2 \right) \cdot y_2 \end{bmatrix}$$

$$\frac{df}{dx} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \frac{df}{dy} = \begin{bmatrix} 0 & 0 \\ B & D + 2 \cdot E \end{bmatrix}$$

- Only one exponential per *u* in case of Bates (no exp in case of Heston)
- Polynomial Riccati equations and derivatives

Agenda presentation

Summary

- Solve the ODEs either analytically or numerically
  - Solving numerically, use control measure to apply adaptive stepsize methods
- The ODEs may become stiff for high value of *u*
  - Solving a stiff problem needs more time
  - Use implicit schemes to solve the ODEs
  - Even in case of jumps the derivatives have polynomial form, thus also the Jacobian is polynomial
- Pricing by Fourier inversion
  - Avoid using high *u* → use direct integration rather than FFT
  - Use the control variate technique

- Numerical solution for ODEs are competitive with analytical solutions

- Use LAPACK, never use the STL complex class in VC++, catch floating point exceptions and handle them, use Volodymyr Myrnyy's FFT implementation with C++ template metaprogramming (vs. FFTW)

Questions

- Contact details:

  Péter Dobránszky
  peter@dobranszky.com

- Further details:

  http://peter.dobranszky.com