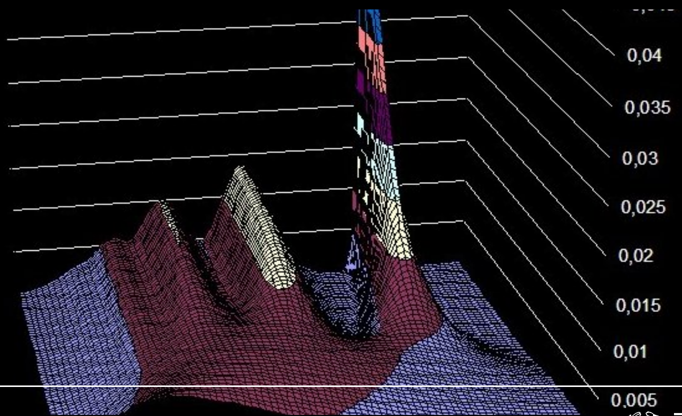
Pricing and Hedging without Fast Fourier Transform

A revisited, reliable and stable Fourier Transform Method for Affine Jump Diffusion Models



Marcello Minenna - Paolo Verzella
Quant Congress Europe – Oct. 31, 2005 - London

Syllabus of the presentation

- Review of Fourier Methods in Option Pricing
- Calibration and Performance
- Greek Behavior of New FT-Q

Syllabus of the presentation

- Review of Fourier Methods in Option Pricing
- Calibration and Performance
- Greek Behavior of New FT-Q

Review of Fourier Methods in Option Pricing – theory

European Call Maturity T Terminal Spot Price S_T

In AJD models Call Price can be expressed in a form close to the canonical Black – Scholes - Merton style

$$C_t = S_t P_1(\Theta) - Ke^{-r\tau} P_2(\Theta)$$

where

$$P_1(\Theta), P_2(\Theta) = \Pr(\ln S_T \geq \ln[K])$$

under different martingale measures

$$P_1(\Theta), P_2(\Theta) = \Pr(\ln S_T \geq \ln[K])$$

under different martingale measures

can be

determined by using the Levy's inversion formula, i.e.:

$$\Pr(\ln S_T \geq \ln[K]) = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathrm{Re}\left[\frac{e^{-i\phi \ln[K]}\widetilde{f}_j(\phi)}{i\phi}\right] d\phi$$

$$\Pr(\ln S_T \geq \ln[K]) = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathrm{Re}\left[\frac{e^{-i\phi \ln[K]}\widetilde{f}_j(\phi)}{i\phi}\right] d\phi$$

requires

a close formula for the Characteristic Function of the log – terminal price, i.e.:

$$\widetilde{f}_T(\phi) = E\left[e^{i\phi \ln S_T}\right]$$

$$\widetilde{f}_T(\phi) = E\left[e^{i\phi \ln S_T}\right]$$

has

a closed formula for AJD models

How to compute: C_t

I

Quadrature Algorithm
FT – Q for
$P_1(\Theta), P_2(\Theta)$

a

Old FT - Q

b

New FT - Q

II

Fast Fourier Transform
FFT

Algorithms Valuation Criteria

STABILITY

The algorithm is defined stable if and only if it "closes" the quadrature scheme

spanning the pricing formula on a vast area of the parameters set.

gives

a "reasonable" result different from a NaN value

Algorithms Valuation Criteria

ACCURACY

The algorithm is defined accurate if and only if

pricing the Call under the Black - Scholes - Merton model via:

- Old FT - Q
- New FT - Q
- FFT

pricing Call via standard Black – Scholes – Merton

up to 10^{-3} precision

Algorithms Valuation Criteria

SPEED

The algorithm is defined fast with respect to the results of the FFT algorithm

a set of 4100 prices along the strike

$P_1(\Theta), P_2(\Theta)$ through Quadrature Algorithm Old FT - Q

High Order Newton Cotes Algorithm

Up to 8th

$$C_t = S_t P_1(\Theta) - Ke^{-r\tau} P_2(\Theta)$$

$P_1(\Theta), P_2(\Theta)$ ⟺ through ⟺ Quadrature Algorithm Old FT - Q

Pros (+)	**Cons (-)**
ACCURACY	STABILITY
	SPEED

$P_1(\Theta), P_2(\Theta)$ ⟺ through ⟺ Quadrature Algorithm New FT - Q

In order to overcome the cited problems of Old FT – Q:

$P_1(\Theta), P_2(\Theta)$ ⟺ through ⟺ Quadrature Algorithm New FT - Q

In order to overcome the cited problems of Old FT – Q:

- Gauss - Lobatto Quadrature Algorithm
- Re-adjustment of $\widetilde{f}_T(\phi) = E\left[e^{i\phi \ln S_T}\right]$

⇕

$$C_t = S_t P_1(\Theta) - Ke^{-r\tau} P_2(\Theta)$$

$P_1(\Theta), P_2(\Theta)$ ⟺ through ⟺ Quadrature Algorithm New FT - Q

Pros (+)	**Cons (-)**
STABILITY	SPEED
ACCURACY	

C_t ⟺ through ⟺ Fast Fourier Trasform FFT

Cooley - Tukey algorithm

Applied to the equivalent formula via a recombinant FFT parameters

$$C_t = \frac{e^{-\alpha \ln K}}{\pi} \int_0^{\infty} e^{-i\phi \ln K} \tilde{f}_j(\phi) d\phi$$ for ATM

$P_1(\Theta), P_2(\Theta)$ ⟺ through ⟺ Fast Fourier Trasform FFT

Pros (+)	Cons (-)
SPEED FASTER (up to 20 times the quadrature algorithms)	**STABILITY** * The formula must be changed arbitrarily according to Option moneyness **ACCURACY** ** the recombinant FFT parameters must be changed according to the choice of the pricing models

Syllabus of the presentation

- Review of Fourier Methods in Option Pricing
- Calibration and Performance
- Greek Behaviour of New FT-Q

The Calibration Procedure and Performance

$$SSE_t = \min_{v(t),\Phi} \sum_{n=1}^{N} \left[C_{Market}(S_t) - C_{AJD}(S_t) \right]^2$$

I → Quadrature Algorithm FT - Q

- a → Old FT - Q
- b → New FT - Q

II → Fast Fourier Trasform FFT

The Calibration Procedure and Performance

$$SSE_t = \min_{v(t),\Phi} \sum_{n=1}^{N} \left[C_{Market}(S_t) - C_{AJD}(S_t)\right]^2$$

through

Quadrature Algorithm
Old FT - Q

Pros (+)

Cons (-)

STABILITY

ACCURACY

SPEED

The Calibration Procedure and Performance

$$SSE_t = \min_{v(t),\Phi} \sum_{n=1}^{N} \left[C_{Market}(S_t) - C_{AJD}(S_t)\right]^2$$

through

Fast Fourier Trasform
FFT

Pros (+)

SPEED

Cons (-)

STABILITY *

ACCURACY **

The Calibration Procedure and Performance

$$SSE_t = \min_{v(t),\Phi} \sum_{n=1}^{N} \left[C_{Market}(S_t) - C_{AJD}(S_t)\right]^2$$

through

Quadrature Algorithm
New FT - Q

Pros (+)

STABILITY

ACCURACY

SPEED

Cons (-)

The Calibration Procedure and Performance

By keeping in mind that only New FT-Q is stable and accurate, some figures on speed

Original Option Pricing Formulas are used

	Heston Model	Merton Model	BCC Model
FFT	7.26 sec.	10.54 sec.	18.33 sec.
NEW FT - Q	55.12 sec.	66.48 sec.	110.39 sec.
OLD FT – Q	390.41 sec.	454.76 sec.	722.1 sec.

By now, the speed of Fourier Trasform method is closer than ever to the FFT calibration time

Calibration Performances using Option Readjusted Pricing Formulas

where available

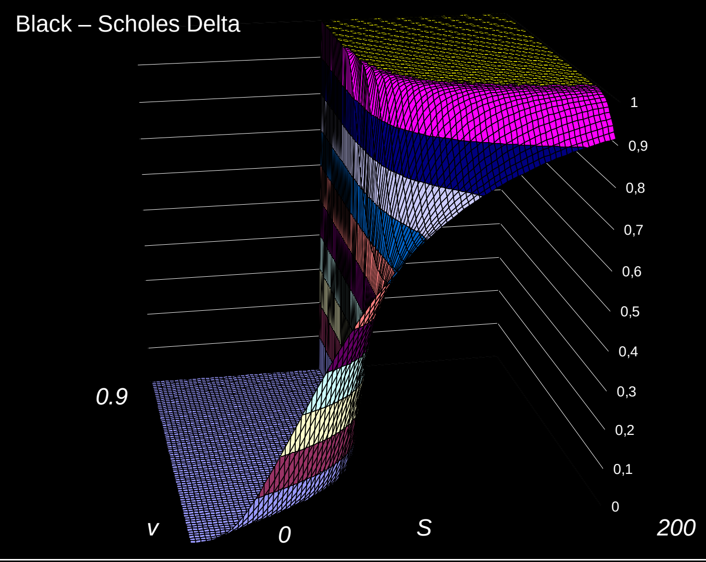
FFT	Heston Model 7.24 sec.	Merton Model 10.54 sec.	BCC Model 18.32 sec.
NEW FT - Q	Heston Model 23.13 sec.	Merton Model 66.48 sec.	BCC Model 48.7 sec.
OLD FT – Q	Heston Model 331.6 sec.	Merton Model 454.76 sec.	BCC Model 688.5 sec.

- Review of Fourier Methods in Option Pricing
- Calibration Procedure and Performance
- Greek Behaviour of New FT-Q

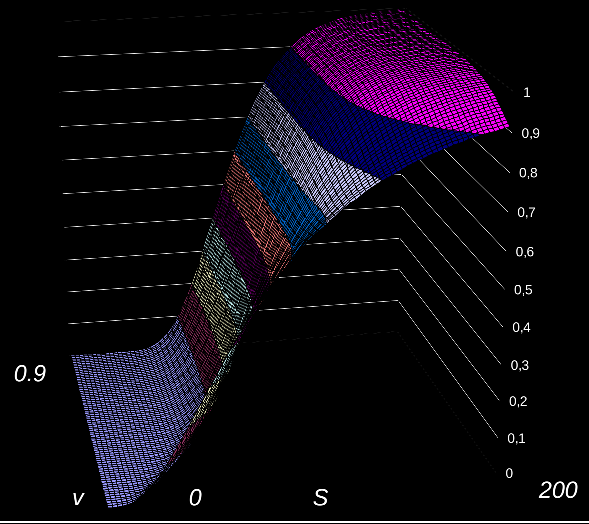
An impressive methodology to test Stability of the New FT – Quadrature algorithm is to compute Greeks

Infact, in an AJD setting the Greeks are available in closed form

So, an extended spanning of the AJD Greeks on the parameters set is useful to assess models and test Stability



Heston Delta



Lambda = -2

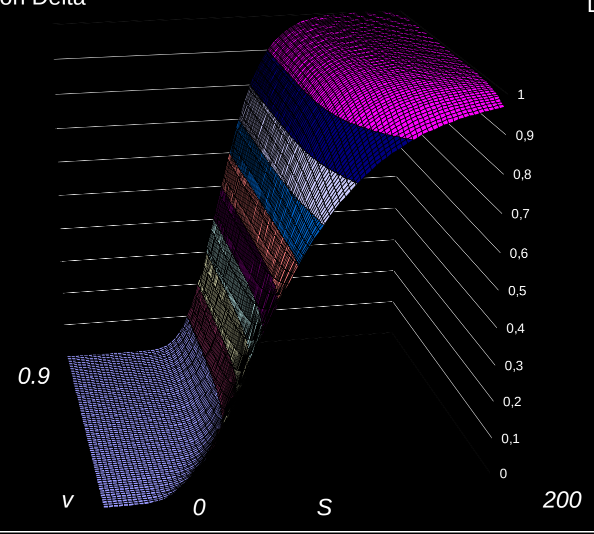
CappaV = 2

ThetaV = 0.3

EtaV = 0.1

Rho = 0

Heston Delta



Lambda = 2

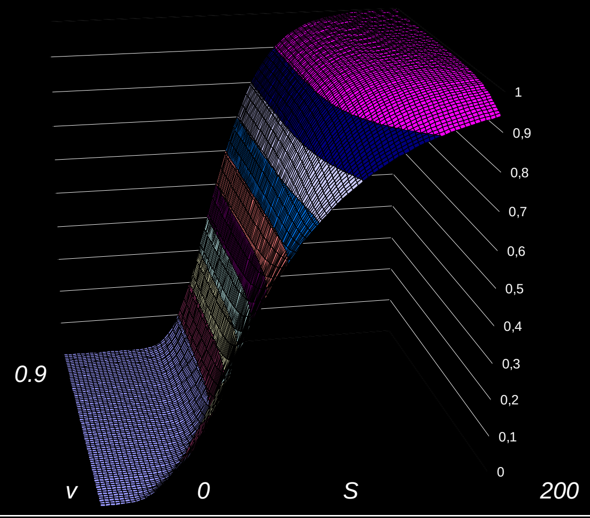
CappaV = 2

ThetaV = 0.3

EtaV = 0.1

Rho = 0

Heston Delta



Rho = -1

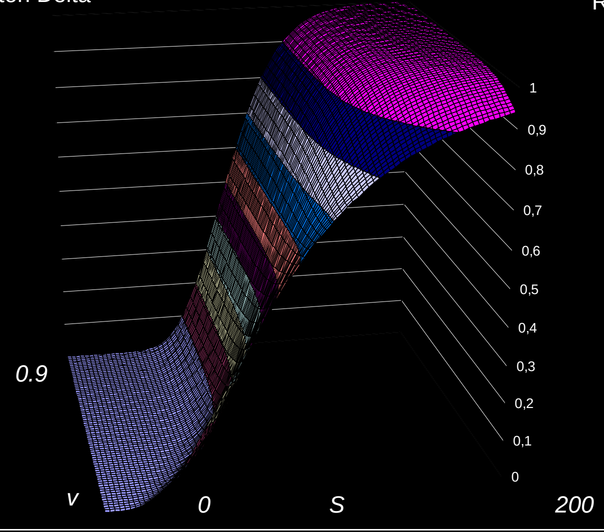
CappaV = 2

ThetaV = 0.3

EtaV = 0.1

Lambda = 0

Heston Delta



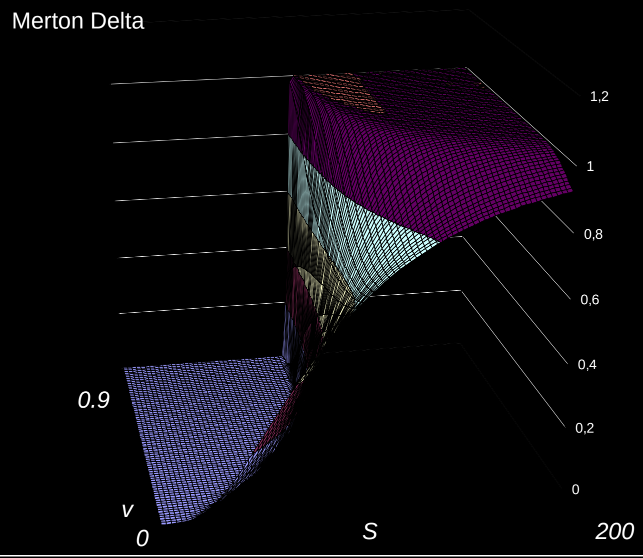
Rho = 1

CappaV = 2

ThetaV = 0.3

EtaV = 0.1

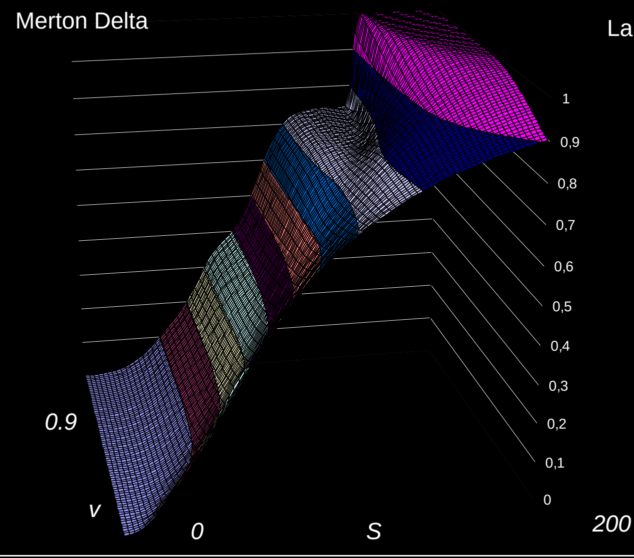
Lambda = 0



LambdaJ = 0

EtaJ = 0

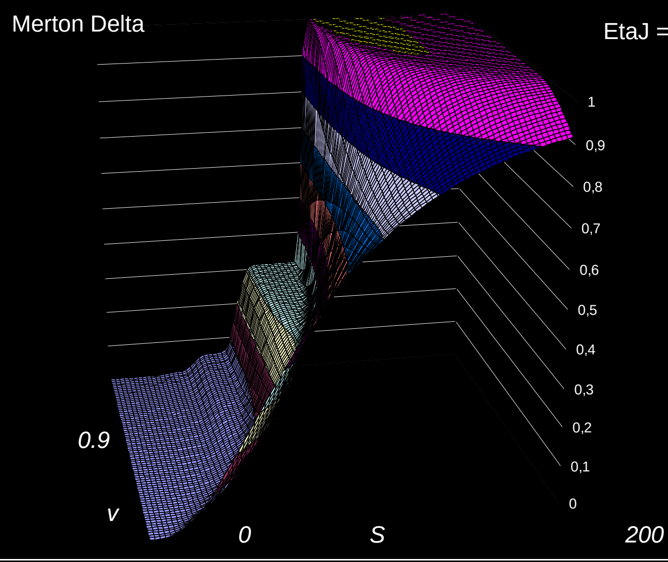
MuJ= 0



LambdaJ = 1.8 ⬆

EtaJ = 0.1

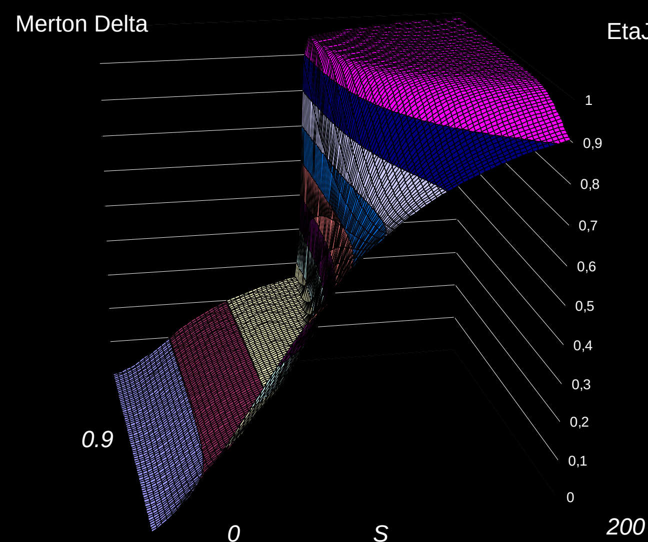
MuJ= 0.5



EtaJ = 0.1

LambdaJ = 0.5

MuJ= 0.5



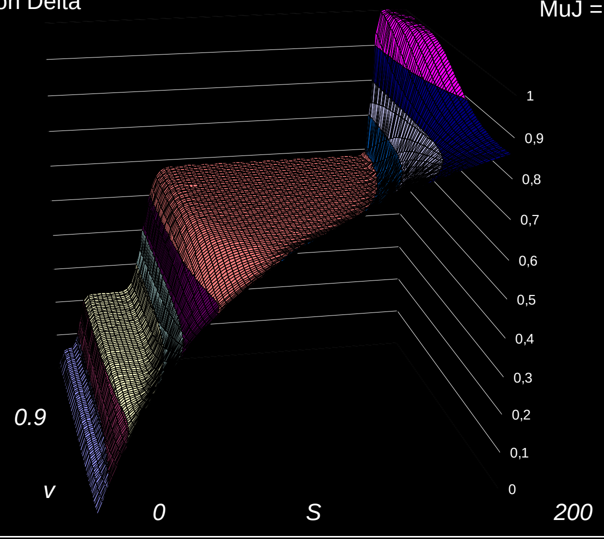
EtaJ = 0.75 ⬆

LambdaJ = 0.5

MuJ= 0.5

Merton Delta

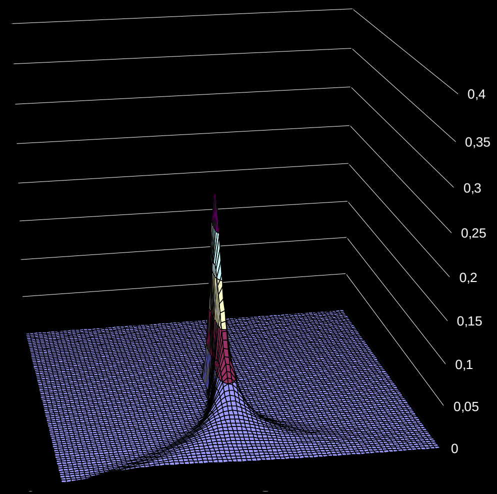
MuJ = 2.5



LambdaJ = 0.5

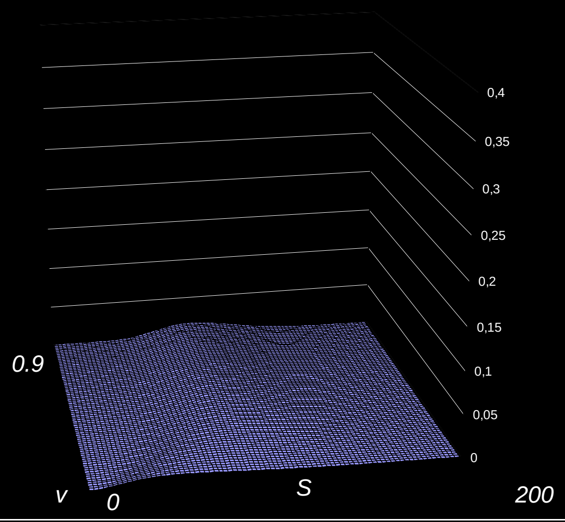
Eta= 0.1

Black – Scholes Gamma



Heston Gamma

Lambda = -2



CappaV = 2

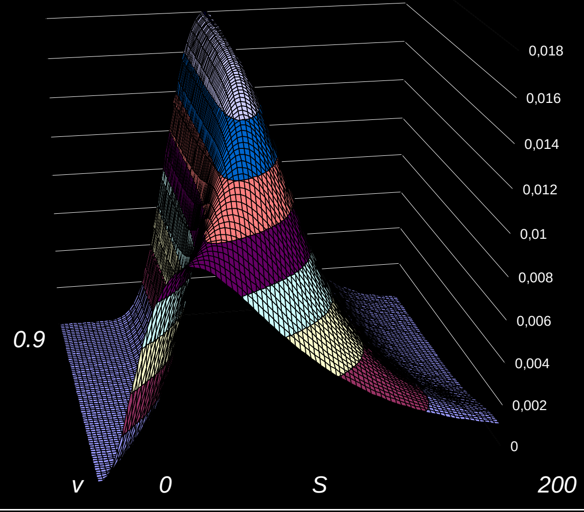
ThetaV = 0.3

EtaV = 0.1

Rho = 0

Heston Gamma

Lambda = -2



CappaV = 2

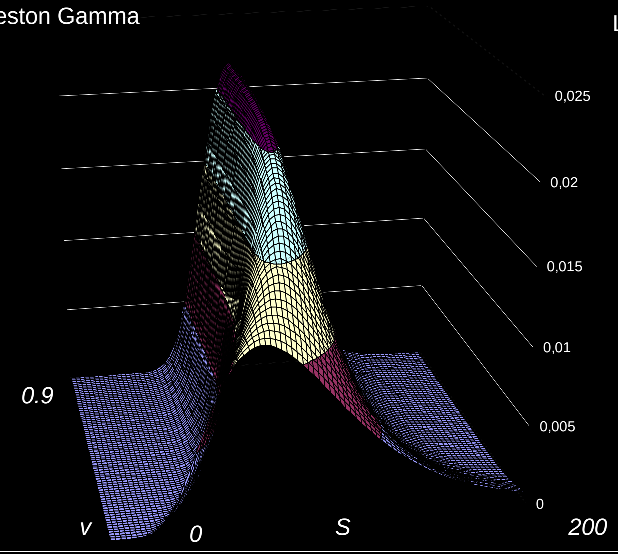
ThetaV = 0.3

EtaV = 0.1

Rho = 0

Heston Gamma

Lambda = 2 ↑



CappaV = 2

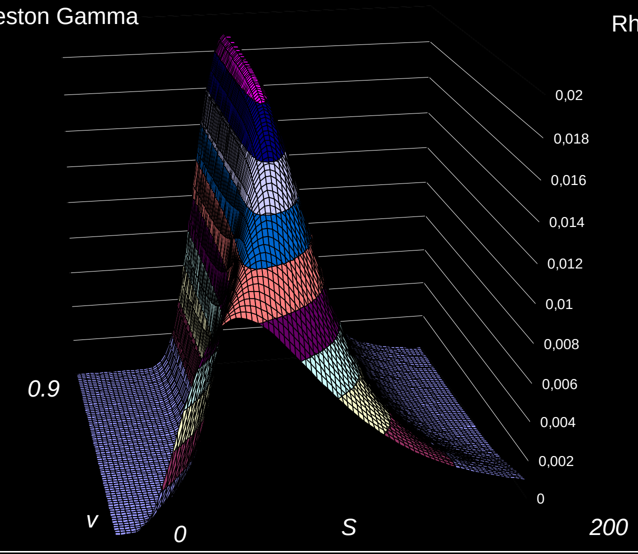
ThetaV = 0.3

EtaV = 0.1

Rho = 0

Heston Gamma

Rho = -1



CappaV = 2

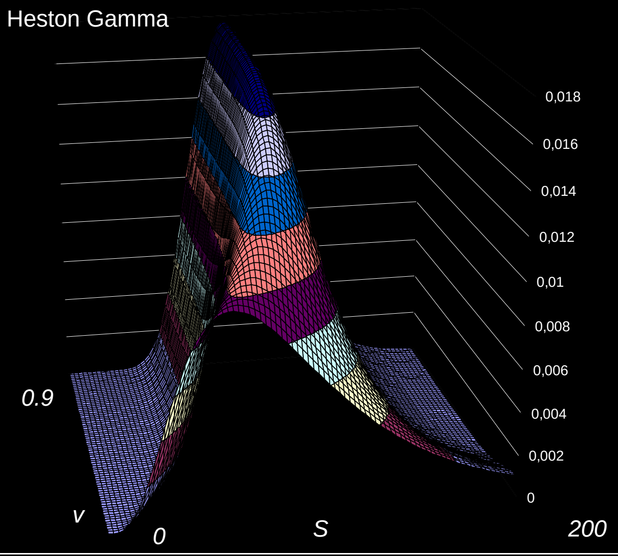
ThetaV = 0.3

EtaV = 0.1

Lambda = 0

Heston Gamma

Rho = 1



CappaV = 2

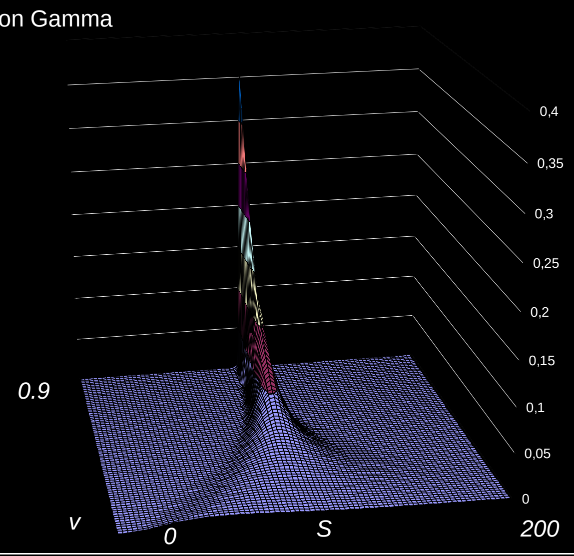
ThetaV = 0.3

EtaV = 0.1

Lambda = 0

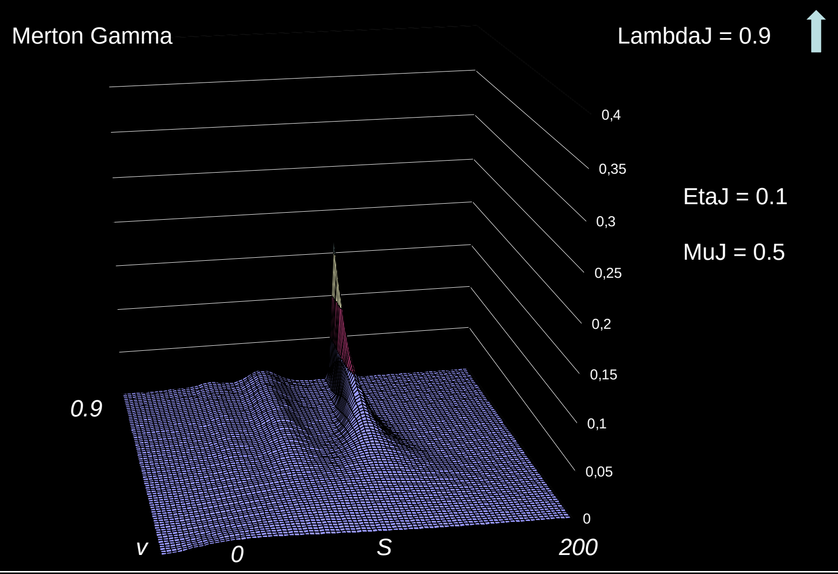
Merton Gamma

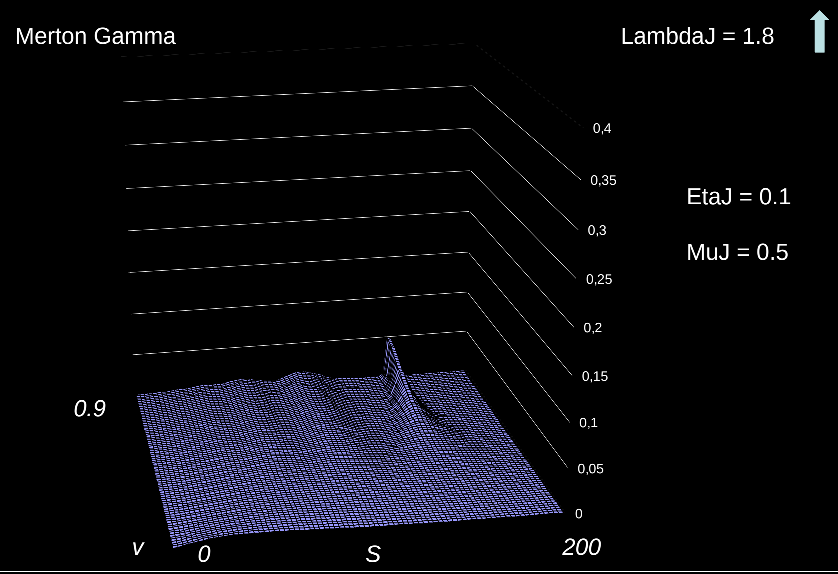
LambdaJ = 0

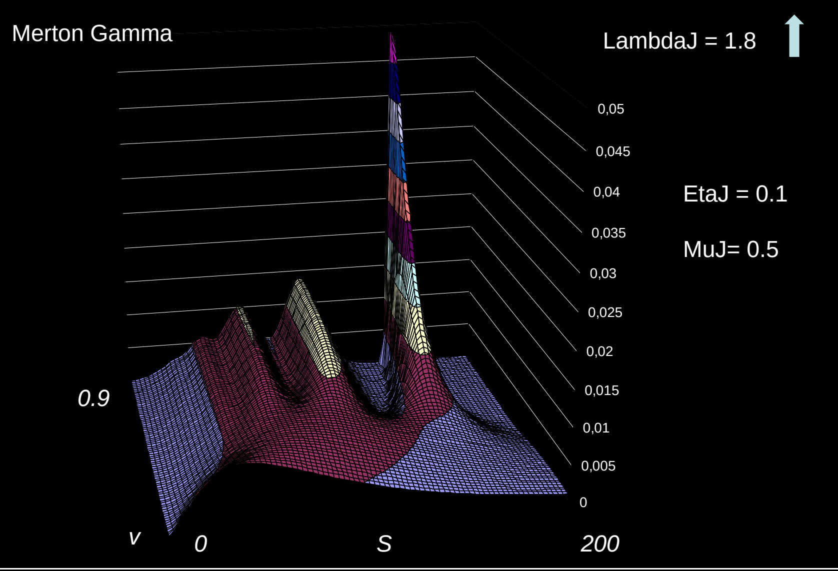


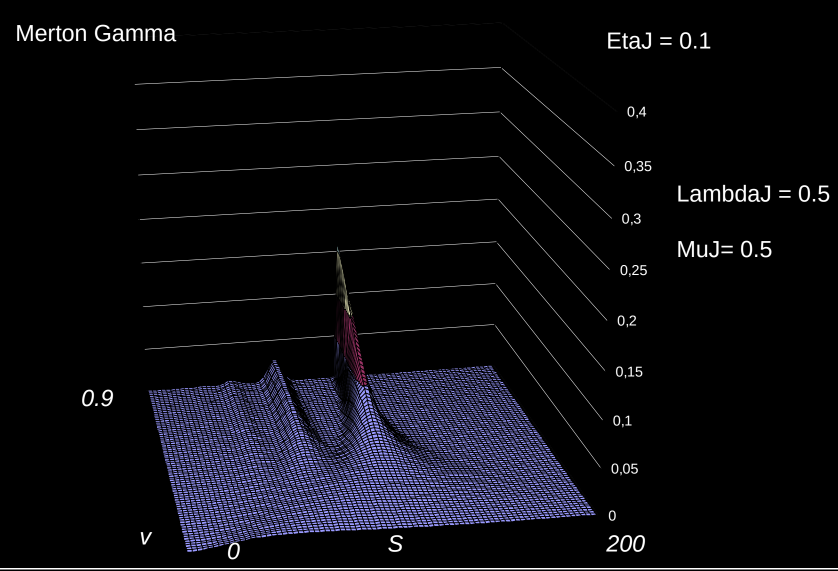
EtaJ = 0

MuJ= 0


Merton Gamma
LambdaJ = 0.9
EtaJ = 0.1
MuJ = 0.5
0,4
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0
0.9
v
0
S
200


Merton Gamma
LambdaJ = 1.8
EtaJ = 0.1
MuJ = 0.5
0,4
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0
0.9
v
0
S
200


Merton Gamma
LambdaJ = 1.8
EtaJ = 0.1
MuJ= 0.5
0,05
0,045
0,04
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
0.9
v
0
S
200

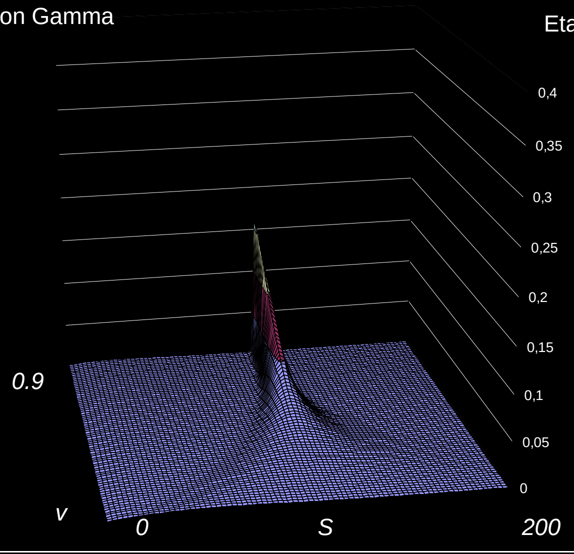

Merton Gamma
EtaJ = 0.1
LambdaJ = 0.5
MuJ= 0.5
0,4
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0
0.9
v
0
S
200

Merton Gamma

EtaJ = 0.75 ↑

LambdaJ = 0.5

MuJ= 0.5

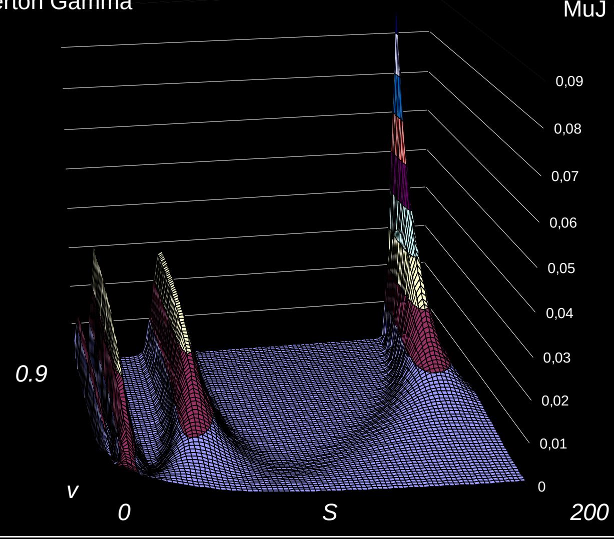


Merton Gamma

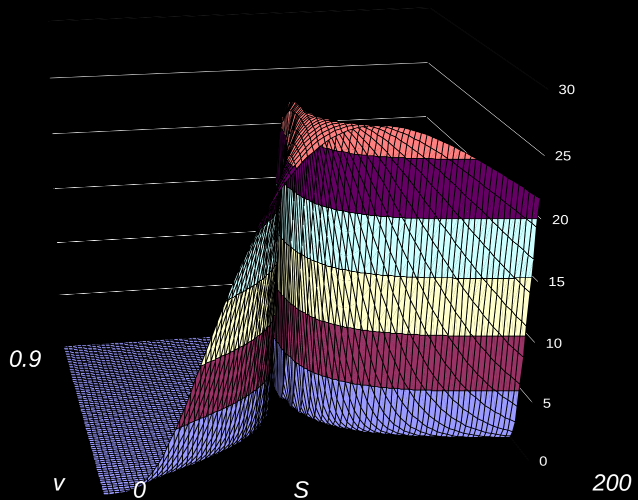
MuJ = 2.5 ↑

LambdaJ = 0.5

EtaJ= 0.1



Black – Scholes Vega



Heston Vega

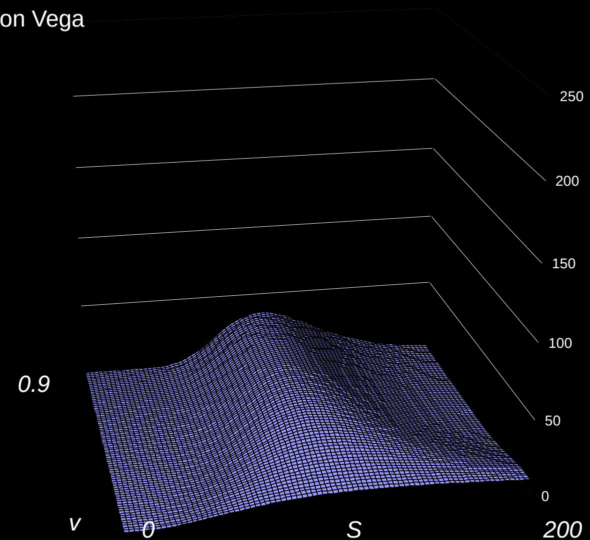
Lambda = -2

CappaV = 2

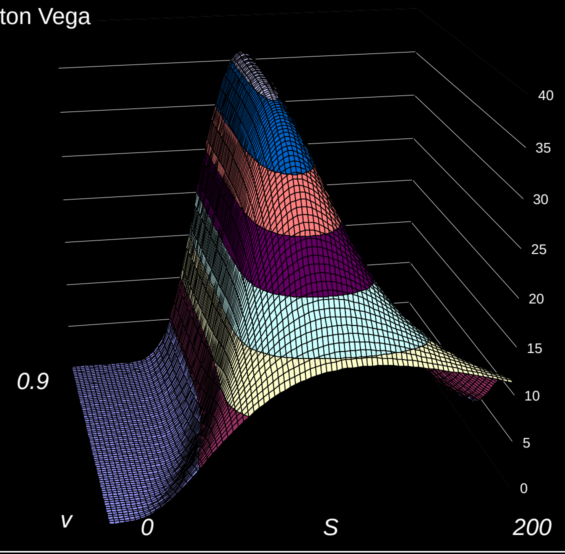
ThetaV = 0.3

EtaV = 0.1

Rho = 0



Heston Vega



Lambda = -2

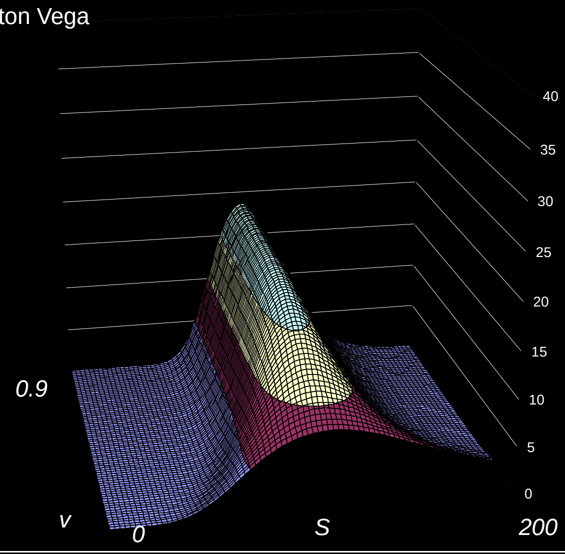
CappaV = 2

ThetaV = 0.3

EtaV = 0.1

Rho = 0

Heston Vega



Lambda = 2

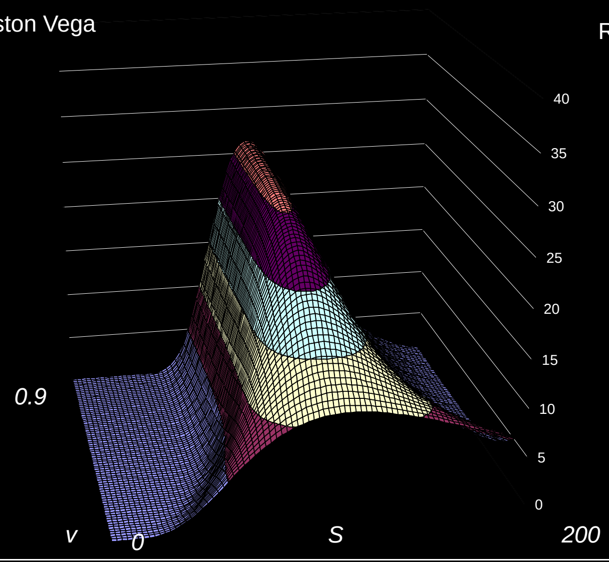
CappaV = 2

ThetaV = 0.3

EtaV = 0.1

Rho = 0

Heston Vega



Rho = -1

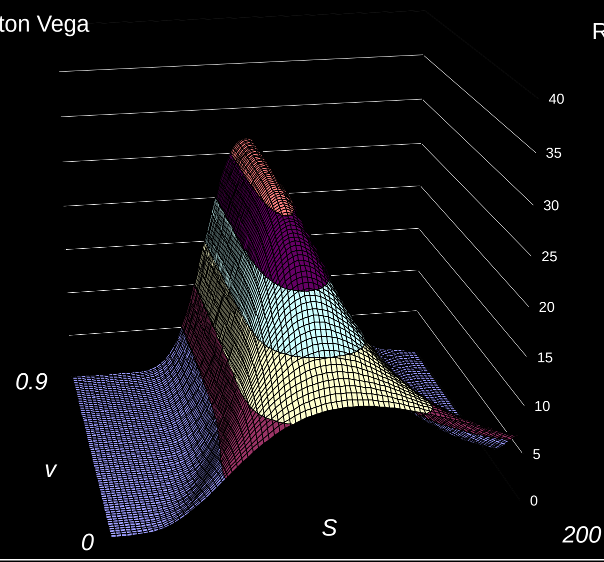
CappaV = 2

ThetaV = 0.3

EtaV = 0.1

Lambda = 0

Heston Vega



Rho = 1

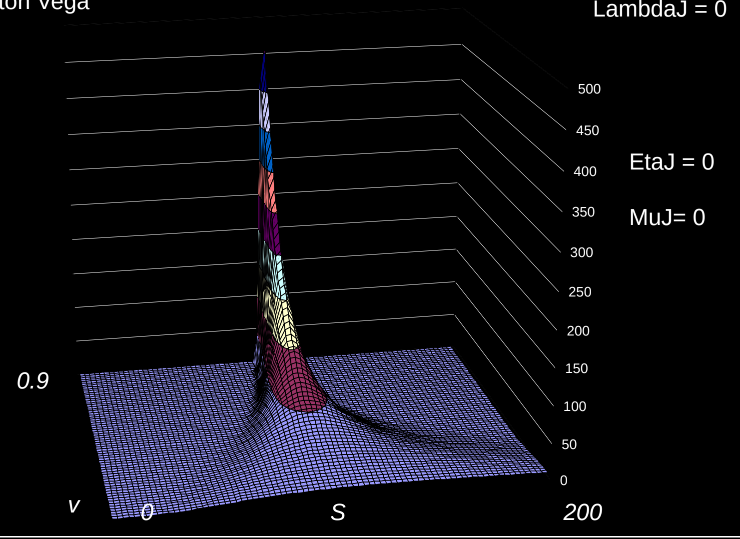
CappaV = 2

ThetaV = 0.3

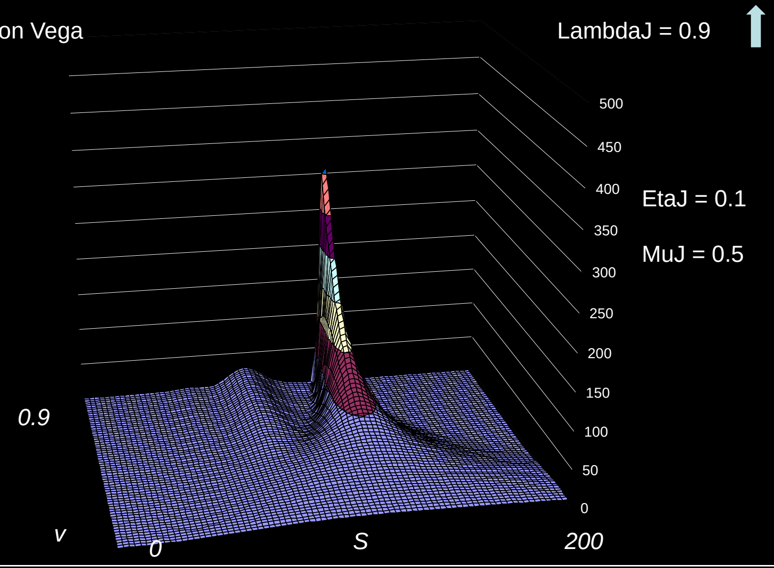
EtaV = 0.1

Lambda = 0

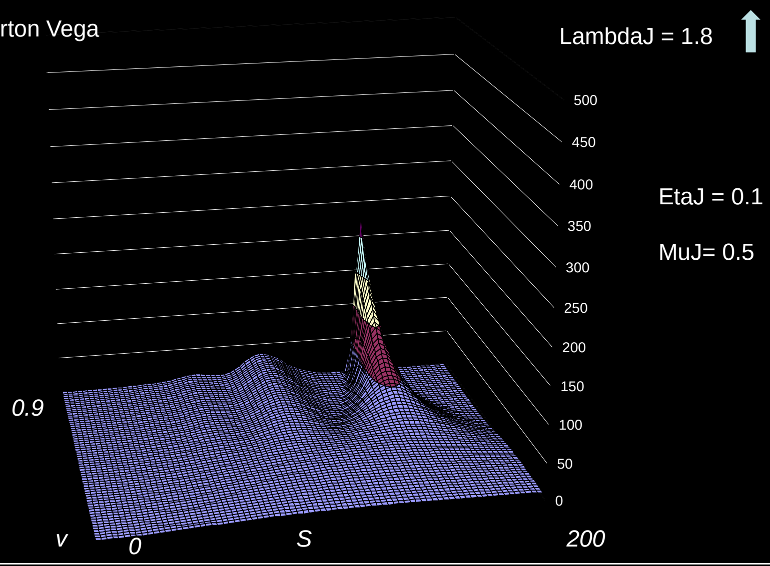
Merton Vega



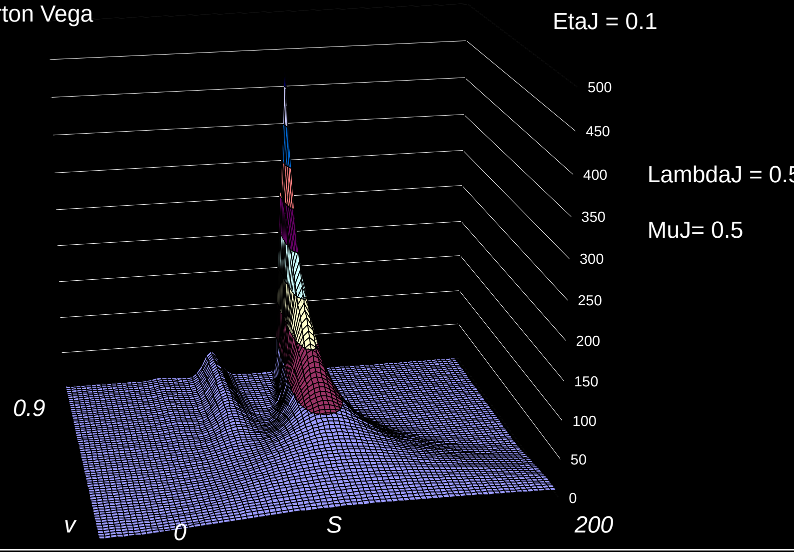
Merton Vega

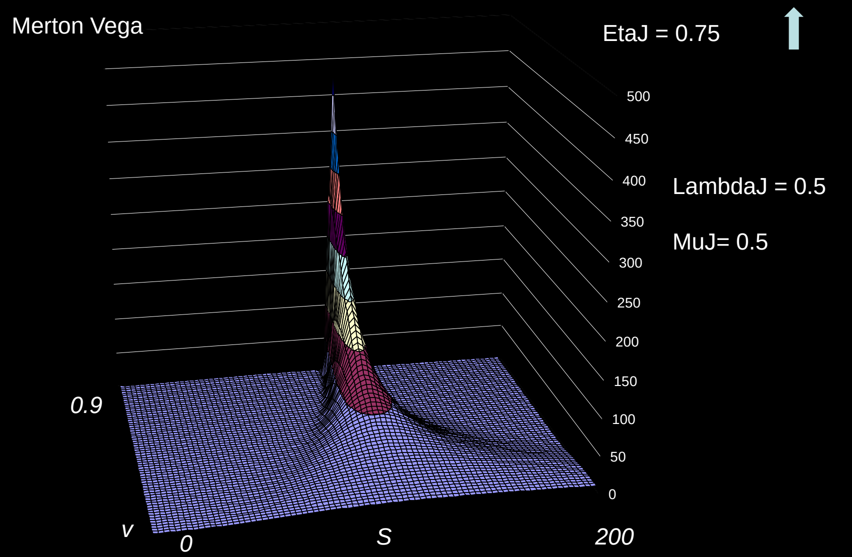


Merton Vega

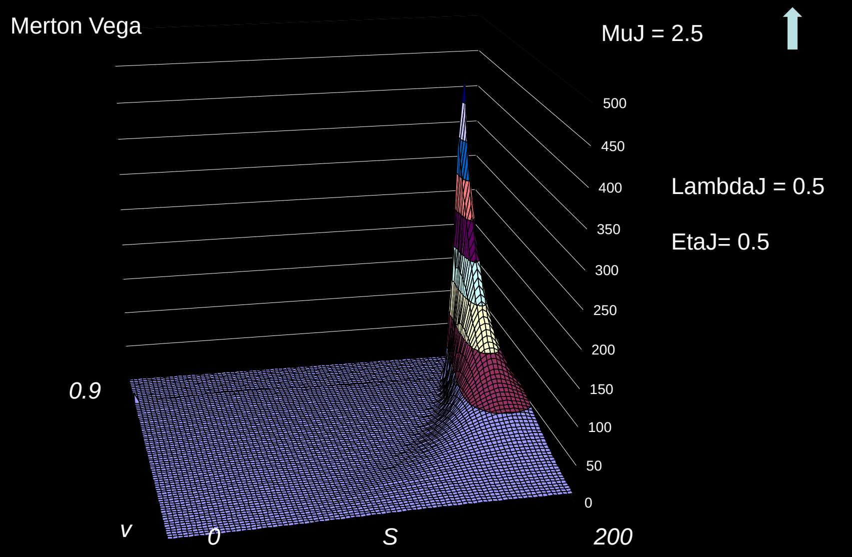


Merton Vega





Merton Vega
EtaJ = 0.75
500
450
400
350
300
250
200
150
100
50
0
LambdaJ = 0.5
MuJ= 0.5
0.9
v
0
S
200



Merton Vega
MuJ = 2.5
500
450
400
350
300
250
200
150
100
50
0
LambdaJ = 0.5
EtaJ= 0.5
0.9
v
0
S
200