NAG Toolbox

nag_specfun_opt_jumpdiff_merton_greeks (s30jb)

1 Purpose

nag_specfun_opt_jumpdiff_merton_greeks (s30jb) computes the European option price together with its sensitivities (Greeks) using the Merton jump-diffusion model.

2 Syntax

```
[p, delta, gamma, vega, theta, rho, vanna, charm, speed, colour, zomma, vomma,
ifail] = nag_specfun_opt_jumpdiff_merton_greeks(calput, x, s, t, sigma, r,
lambda, jvol, 'm', m, 'n', n)
[p, delta, gamma, vega, theta, rho, vanna, charm, speed, colour, zomma, vomma,
ifail] = s30jb(calput, x, s, t, sigma, r, lambda, jvol, 'm', m, 'n', n)
```

3 Description

nag_specfun_opt_jumpdiff_merton_greeks (s30jb) uses Merton's jump-diffusion model (Merton (1976)) to compute the price of a European option, together with the Greeks or sensitivities, which are the partial derivatives of the option price with respect to certain of the other input parameters. Merton's model assumes that the asset price is described by a Brownian motion with drift, as in the Black–Scholes–Merton case, together with a compound Poisson process to model the jumps. The corresponding stochastic differential equation is,

$$\frac{dS}{S} = (\alpha - \lambda k)dt + \hat{\sigma}dW_t + dq_t.$$

Here α is the instantaneous expected return on the asset price, S; $\hat{\sigma}^2$ is the instantaneous variance of the return when the Poisson event does not occur; dW_t is a standard Brownian motion; q_t is the independent Poisson process and $k = E[Y - 1]$ where $Y - 1$ is the random variable change in the stock price if the Poisson event occurs and E is the expectation operator over the random variable Y.

This leads to the following price for a European option (see Haug (2007))

$$P_{\text{call}} = \sum_{j=0}^{\infty} \frac{e^{-\lambda T}(\lambda T)^j}{j!} C_j\left(S, X, T, r, \sigma_j'\right),$$

where T is the time to expiry; X is the strike price; r is the annual risk-free interest rate; $C_j\left(S, X, T, r, \sigma_j'\right)$ is the Black–Scholes–Merton option pricing formula for a European call (see nag_specfun_opt_bsm_price (s30aa)).

$$\sigma_j' = \sqrt{z^2 + \delta^2\left(\frac{j}{T}\right)},$$

$$z^2 = \sigma^2 - \lambda\delta^2,$$

$$\delta^2 = \frac{\gamma\sigma^2}{\lambda},$$

where σ is the total volatility including jumps; λ is the expected number of jumps given as an average per year; γ is the proportion of the total volatility due to jumps.

The value of a put is obtained by substituting the Black–Scholes–Merton put price for $C_j\left(S, X, T, r, \sigma_j'\right)$.

The option price $P_{ij} = P(X = X_i, T = T_j)$ is computed for each strike price in a set X_i, $i = 1, 2, \ldots, m$, and for each expiry time in a set T_j, $j = 1, 2, \ldots, n$.

4 References

Haug E G (2007) *The Complete Guide to Option Pricing Formulas* (2nd Edition) McGraw-Hill

Merton R C (1976) Option pricing when underlying stock returns are discontinuous *Journal of Financial Economics* **3** 125–144

5 Parameters

5.1 Compulsory Input Parameters

1: **calput** – CHARACTER(1)

Determines whether the option is a call or a put.

calput = 'C'
A call; the holder has a right to buy.

calput = 'P'
A put; the holder has a right to sell.

Constraint: **calput** = 'C' or 'P'.

2: **x**(**m**) – REAL (KIND=nag_wp) array

x(i) must contain X_i, the ith strike price, for $i = 1, 2, \ldots,$ **m**.

Constraint: **x**(i) $\geq z$ and **x**(i) $\leq 1/z$, where $z =$ x02am(), the safe range parameter, for $i = 1, 2, \ldots,$ **m**.

3: **s** – REAL (KIND=nag_wp)

S, the price of the underlying asset.

Constraint: **s** $\geq z$ and **s** $\leq 1.0/z$, where $z =$ x02am(), the safe range parameter.

4: **t**(**n**) – REAL (KIND=nag_wp) array

t(i) must contain T_i, the ith time, in years, to expiry, for $i = 1, 2, \ldots,$ **n**.

Constraint: **t**(i) $\geq z$, where $z =$ x02am(), the safe range parameter, for $i = 1, 2, \ldots,$ **n**.

5: **sigma** – REAL (KIND=nag_wp)

σ, the annual total volatility, including jumps.

Constraint: **sigma** > 0.0.

6: **r** – REAL (KIND=nag_wp)

r, the annual risk-free interest rate, continuously compounded. Note that a rate of 5% should be entered as 0.05.

Constraint: **r** ≥ 0.0.

7: **lambda** – REAL (KIND=nag_wp)

λ, the number of expected jumps per year.

Constraint: **lambda** > 0.0.

8: **jvol** – REAL (KIND=nag_wp)

The proportion of the total volatility associated with jumps.

Constraint: $0.0 \leq \mathbf{jvol} < 1.0$.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the dimension of the array **x**.

The number of strike prices to be used.

Constraint: $\mathbf{m} \geq 1$.

2: **n** – INTEGER

Default: the dimension of the array **t**.

The number of times to expiry to be used.

Constraint: $\mathbf{n} \geq 1$.

5.3 Output Parameters

1: $\mathbf{p}(ldp, \mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

$\mathbf{p}(i, j)$ contains P_{ij}, the option price evaluated for the strike price $\mathbf{x}_i$ at expiry $\mathbf{t}_j$ for $i = 1, 2, \ldots, \mathbf{m}$ and $j = 1, 2, \ldots, \mathbf{n}$.

2: **delta**$(ldp, \mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

The leading $\mathbf{m} \times \mathbf{n}$ part of the array **delta** contains the sensitivity, $\frac{\partial P}{\partial S}$, of the option price to change in the price of the underlying asset.

3: **gamma**$(ldp, \mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

The leading $\mathbf{m} \times \mathbf{n}$ part of the array **gamma** contains the sensitivity, $\frac{\partial^2 P}{\partial S^2}$, of **delta** to change in the price of the underlying asset.

4: **vega**$(ldp, \mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

vega(i, j), contains the first-order Greek measuring the sensitivity of the option price P_{ij} to change in the volatility of the underlying asset, i.e., $\frac{\partial P_{ij}}{\partial \sigma}$, for $i = 1, 2, \ldots, \mathbf{m}$ and $j = 1, 2, \ldots, \mathbf{n}$.

5: **theta**$(ldp, \mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

theta(i, j), contains the first-order Greek measuring the sensitivity of the option price P_{ij} to change in time, i.e., $-\frac{\partial P_{ij}}{\partial T}$, for $i = 1, 2, \ldots, \mathbf{m}$ and $j = 1, 2, \ldots, \mathbf{n}$, where $b = r - q$.

6: **rho**$(ldp, \mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

rho(i,j), contains the first-order Greek measuring the sensitivity of the option price P_{ij} to change in the annual risk-free interest rate, i.e., $-\frac{\partial P_{ij}}{\partial r}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

7: **vanna**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

vanna(i,j), contains the second-order Greek measuring the sensitivity of the first-order Greek Δ_{ij} to change in the volatility of the asset price, i.e., $-\frac{\partial \Delta_{ij}}{\partial T} = -\frac{\partial^2 P_{ij}}{\partial S \partial \sigma}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

8: **charm**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

charm(i,j), contains the second-order Greek measuring the sensitivity of the first-order Greek Δ_{ij} to change in the time, i.e., $-\frac{\partial \Delta_{ij}}{\partial T} = -\frac{\partial^2 P_{ij}}{\partial S \partial T}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

9: **speed**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

speed(i,j), contains the third-order Greek measuring the sensitivity of the second-order Greek Γ_{ij} to change in the price of the underlying asset, i.e., $-\frac{\partial \Gamma_{ij}}{\partial S} = -\frac{\partial^3 P_{ij}}{\partial S^3}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

10: **colour**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

colour(i,j), contains the third-order Greek measuring the sensitivity of the second-order Greek Γ_{ij} to change in the time, i.e., $-\frac{\partial \Gamma_{ij}}{\partial T} = -\frac{\partial^3 P_{ij}}{\partial S \partial T}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

11: **zomma**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

zomma(i,j), contains the third-order Greek measuring the sensitivity of the second-order Greek Γ_{ij} to change in the volatility of the underlying asset, i.e., $-\frac{\partial \Gamma_{ij}}{\partial \sigma} = -\frac{\partial^3 P_{ij}}{\partial S^2 \partial \sigma}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

12: **vomma**$(ldp,\mathbf{n})$ – REAL (KIND=nag_wp) array

$ldp = \mathbf{m}$.

vomma(i,j), contains the second-order Greek measuring the sensitivity of the first-order Greek Δ_{ij} to change in the volatility of the underlying asset, i.e., $-\frac{\partial \Delta_{ij}}{\partial \sigma} = -\frac{\partial^2 P_{ij}}{\partial \sigma^2}$, for $i = 1,2,\ldots,\mathbf{m}$ and $j = 1,2,\ldots,\mathbf{n}$.

13: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **calput** $= \langle value \rangle$ was an illegal value.

ifail $= 2$

Constraint: **m** ≥ 1.

ifail $= 3$

Constraint: **n** ≥ 1.

ifail $= 4$

Constraint: $\mathbf{x}(i) \geq \langle value \rangle$ and $\mathbf{x}(i) \leq \langle value \rangle$.

ifail $= 5$

Constraint: **s** $\geq \langle value \rangle$ and **s** $\leq \langle value \rangle$.

ifail $= 6$

Constraint: $\mathbf{t}(i) \geq \langle value \rangle$.

ifail $= 7$

Constraint: **sigma** > 0.0.

ifail $= 8$

Constraint: **r** ≥ 0.0.

ifail $= 9$

Constraint: **lambda** > 0.0.

ifail $= 10$

Constraint: **jvol** ≥ 0.0 and **jvol** < 1.0.

ifail $= 12$

Constraint: $ldp \geq$ **m**.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

The accuracy of the output is dependent on the accuracy of the cumulative Normal distribution function, Φ, occurring in C_j. This is evaluated using a rational Chebyshev expansion, chosen so that the maximum relative error in the expansion is of the order of the ***machine precision*** (see nag_specfun_cdf_normal (s15ab) and nag_specfun_erfc_real (s15ad)). An accuracy close to ***machine precision*** can generally be expected.

8 Further Comments

None.

9 Example

This example computes the price of two European calls with jumps. The time to expiry is 6 months, the stock price is 100 and strike prices are 80 and 90 respectively. The number of jumps per year is 5 and the percentage of the total volatility due to jumps is 25%. The risk-free interest rate is 8% per year while the total volatility is 25% per year.

9.1 Program Text

```
     function s30jb_example

fprintf('s30jb example results\n\n');

put    = 'C';
lambda = 5;
s      = 100.0;
sigma  = 0.25;
r      = 0.08;
jvol   = 0.25;
x      = [80.0, 90.0];
t      = [0.5];

[p, delta, gamma, vega,   theta, rho, ...
    vanna, charm, speed, colour, zomma, vomma, ifail] = ...
    s30jb(...
          put, x, s, t, sigma, r, lambda, jvol);

fprintf('\nMerton Jump-Diffusion Model\n European Call :\n');
fprintf('  Spot       =   %9.4f\n', s);
fprintf('  Volatility =   %9.4f\n', sigma);
fprintf('  Rate       =   %9.4f\n', r);
fprintf('  Jumps      =   %9.4f\n', lambda);
fprintf('  Jump Vol   =   %9.4f\n\n', jvol);

fprintf(' Time to Expiry : %8.4f\n', t(1));

fprintf('%8s%9s%9s%9s%9s%9s%9s\n','Strike','Price','Delta','Gamma',...
        'Vega','Theta','Rho');
for i=1:2
  fprintf('%8.4f%9.4f%9.4f%9.4f%9.4f%9.4f%9.4f\n', x(i), p(i,1), ...
          delta(i,1), gamma(i,1), vega(i,1), theta(i,1), rho(i,1));
end

fprintf('\n%26s%9s%9s%9s%9s%9s\n','Vanna','Charm','Speed','Colour',...
        'Zomma','Vomma');
for i=1:2
  fprintf('%17s%9.4f%9.4f%9.4f%9.4f%9.4f%9.4f\n', ' ', vanna(i,1), ...
          charm(i,1), speed(i,1), colour(i,1), zomma(i,1), vomma(i,1));
end
```

9.2 Program Results

```
     s30jb example results

Merton Jump-Diffusion Model
 European Call :
  Spot       =    100.0000
  Volatility =      0.2500
  Rate       =      0.0800
  Jumps      =      5.0000
  Jump Vol   =      0.2500

 Time to Expiry :   0.5000
  Strike    Price    Delta    Gamma     Vega    Theta      Rho
 80.0000  23.6090   0.9431   0.0064   8.1206  -7.6718  35.3480
```

```
 90.0000   15.4193    0.8203    0.0149   18.5256   -9.9695   33.3037

                      Vanna     Charm     Speed    Colour     Zomma     Vomma
                    -0.6334    0.1080   -0.0006   -0.0035    0.0315   70.6824
                    -0.7726    0.0770   -0.0009    0.0109   -0.0186   49.7161
```