NAG Toolbox

nag_specfun_opt_jumpdiff_merton_price (s30ja)

1 Purpose

nag_specfun_opt_jumpdiff_merton_price (s30ja) computes the European option price using the Merton jump-diffusion model.

2 Syntax

```
[p, ifail] = nag_specfun_opt_jumpdiff_merton_price(calput, x, s, t, sigma, r,
lambda, jvol, 'm', m, 'n', n)
```

```
[p, ifail] = s30ja(calput, x, s, t, sigma, r, lambda, jvol, 'm', m, 'n', n)
```

3 Description

nag_specfun_opt_jumpdiff_merton_price (s30ja) uses Merton's jump-diffusion model (Merton (1976)) to compute the price of a European option. This 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)).

$$\begin{aligned}
\sigma'_j &= \sqrt{z^2 + \delta^2\left(\tfrac{j}{T}\right)}, \\
z^2 &= \sigma^2 - \lambda\delta^2, \\
\delta^2 &= \tfrac{\gamma\sigma^2}{\lambda},
\end{aligned}$$

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 $P_{ij} = P\left(X = X_i, T = T_j\right)$ 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 $i = 1, 2, \dots,$ **m**.

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

Constraint: **t**$(i) \geq z$, where $z =$ x02am(), the safe range parameter, for $i = 1, 2, \dots,$ **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: $0.0 \leq$ **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: **ifail** – INTEGER

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

$\mathbf{ifail} = 1$

On entry, $\mathbf{calput} = \langle value \rangle$ was an illegal value.

$\mathbf{ifail} = 2$

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

$\mathbf{ifail} = 3$

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

$\mathbf{ifail} = 4$

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

$\mathbf{ifail} = 5$

Constraint: $\mathbf{s} \geq \langle value \rangle$ and $\mathbf{s} \leq \langle value \rangle$.

$\mathbf{ifail} = 6$

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

$\mathbf{ifail} = 7$

Constraint: $\mathbf{sigma} > 0.0$.

$\mathbf{ifail} = 8$

Constraint: $\mathbf{r} \geq 0.0$.

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 a European call with jumps. The time to expiry is 3 months, the stock price is 45 and the strike price is 55. The number of jumps per year is 3 and the percentage of the total volatility due to jumps is 40%. The risk-free interest rate is 10% per year and the total volatility is 25% per year.

9.1 Program Text

```
function s30ja_example

fprintf('s30ja example results\n\n');

put = 'C';
lambda = 3;
s = 45;
sigma = 0.25;
r = 0.1;
jvol = 0.4;
x = [55.0];
t = [0.25];

[p, ifail] = s30ja( ...
                    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('   Strike    Expiry   Option Price\n');
for i=1:1
  for j=1:1
    fprintf('%9.4f %9.4f %9.4f\n', x(i), t(j), p(i,j));
  end
end
```

9.2 Program Results

```
     s30ja example results

Merton Jump-Diffusion Model
 European Call :
  Spot       =     45.0000
  Volatility =      0.2500
  Rate       =      0.1000
  Jumps      =      3.0000
  Jump Vol   =      0.4000

   Strike    Expiry   Option Price
  55.0000    0.2500    0.2417
```