NAG Toolbox for MATLAB

d03nd

1 Purpose

d03nd computes an analytic solution to the Black–Scholes equation for a certain set of option types.

2 Syntax

```
[f, theta, delta, gamma, lambda, rho, ifail] = d03nd(kopt, x, s, t,
tmat, tdpar, r, q, sigma)
```

3 Description

d03nd computes an analytic solution to the Black–Scholes equation (see Hull 1989 and Wilmott *et al.* 1995)

$$\frac{\partial f}{\partial t} + (r - q)S\frac{\partial f}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 f}{\partial S^2} = rf \tag{1}$$

$$S_{\min} < S < S_{\max}, \qquad t_{\min} < t < t_{\max}, \tag{2}$$

for the value f of a European put or call option, or an American call option with zero dividend q. In equation (1) t is time, S is the stock price, X is the exercise price, r is the risk free interest rate, q is the continuous dividend, and σ is the stock volatility. The parameters r, q and σ may be either constant, or functions of time. In the latter case their average instantaneous values over the remaining life of the option should be provided to d03nd. An auxiliary function d03ne is available to compute such averages from values at a set of discrete times.

d03nd also returns values of the Greeks

$$\Theta = \frac{\partial f}{\partial t}, \qquad \Delta = \frac{\partial f}{\partial x}, \qquad \Gamma = \frac{\partial^2 f}{\partial x^2}, \qquad \Lambda = \frac{\partial f}{\partial \sigma}, \qquad \rho = \frac{\partial f}{\partial r}.$$

Further details of the analytic solution returned are given in Section 8.1.

4 References

Hull J 1989 *Options, Futures and Other Derivative Securities* Prentice–Hall

Wilmott P, Howison S and Dewynne J 1995 *The Mathematics of Financial Derivatives* Cambridge University Press

5 Parameters

5.1 Compulsory Input Parameters

1: **kopt – int32 scalar**

Specifies the kind of option to be valued:

kopt $= 1$

A European call option.

kopt $= 2$

An American call option.

kopt $= 3$

A European put option.

Constraints:

kopt $= 1$ or 3 otherwise;
if $q = 0$, **kopt** $= 2$.

2: **x – double scalar**

The exercise price X.

Constraint: **x** ≥ 0.0.

3: **s – double scalar**

The stock price at which the option value and the Greeks should be evaluated.

Constraint: **s** ≥ 0.0.

4: **t – double scalar**

The time at which the option value and the Greeks should be evaluated.

Constraint: **t** ≥ 0.0.

5: **tmat – double scalar**

The maturity time of the option.

Constraint: **tmat** $\geq$ **t**.

6: **tdpar(3) – logical array**

Specifies whether or not various parameters are time-dependent. More precisely, r is time-dependent if **tdpar**$(1) =$ **true** and constant otherwise. Similarly, **tdpar**(2) specifies whether q is time-dependent and **tdpar**(3) specifies whether σ is time-dependent.

7: **r(*) – double array**

Note: the dimension of the array **r** must be at least 3 if **tdpar**$(1) =$ **true**, and at least 1 otherwise.

If **tdpar**$(1) =$ **false** then **r**(1) must contain the constant value of r. The remaining elements need not be set.

If **tdpar**$(1) =$ **true** then **r**(1) must contain the value of r at time **t** and **r**(2) must contain its average instantaneous value over the remaining life of the option:

$$\hat{r} = \int_{\mathbf{t}}^{\mathbf{tmat}} r(\zeta)\, d\zeta.$$

The auxiliary functiond03ne may be used to construct **r** from a set of values of r at discrete times.

8: **q(*) – double array**

Note: the dimension of the array **q** must be at least 3 if **tdpar**$(2) =$ **true**, and at least 1 otherwise.

If **tdpar**$(2) =$ **false** then **q**(1) must contain the constant value of q. The remaining elements need not be set.

If **tdpar**$(2) =$ **true** then **q**(1) must contain the constant value of q and **q**(2) must contain its average instantaneous value over the remaining life of the option:

$$\hat{q} = \int_{\mathbf{t}}^{\mathbf{tmat}} q(\zeta)\, d\zeta.$$

The auxiliary functiond03ne may be used to construct **q** from a set of values of q at discrete times.

9: **sigma**$(*)$ **– double array**

Note: the dimension of the array **sigma** must be at least 3 if **tdpar**$(3) =$ **true**, and at least 1 otherwise.

If **tdpar**$(3) =$ **false** then **sigma**(1) must contain the constant value of σ. The remaining elements need not be set.

If **tdpar**$(3) =$ **true** then **sigma**(1) must contain the value of σ at time **t**, **sigma**(2) the average instantaneous value $\hat{\sigma}$, and **sigma**(3) the second-order average $\bar{\sigma}$, where:

$$\hat{\sigma} = \int_{\mathbf{t}}^{\mathbf{tmat}} \sigma(\zeta)\, d\zeta,$$

$$\bar{\sigma} = \left(\int_{\mathbf{t}}^{\mathbf{tmat}} \sigma^2(\zeta)\, d\zeta \right)^{1/2}.$$

The auxiliary function d03ne may be used to compute **sigma** from a set of values at discrete times.

Constraints:

if **tdpar**$(3) =$ **false**, **sigma**$(1) > 0.0$;
if **tdpar**$(3) =$ **true**, **sigma**$(i) > 0.0$, for $i = 1, 2, 3$.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **f – double scalar**

The value f of the option at the stock price **s** and time **t**.

2: **theta – double scalar**
3: **delta – double scalar**
4: **gamma – double scalar**
5: **lambda – double scalar**
6: **rho – double scalar**

The values of various Greeks at the stock price **s** and time **t**, as follows:

$$\mathbf{theta} = \Theta = \frac{\partial f}{\partial t}, \quad \mathbf{delta} = \Delta = \frac{\partial f}{\partial \mathbf{s}}, \quad \mathbf{gamma} = \Gamma = \frac{\partial^2 f}{\partial \mathbf{s}^2},$$
$$\mathbf{lambda} = \Lambda = \frac{\partial f}{\partial \sigma}, \quad \mathbf{rho} = \rho = \frac{\partial f}{\partial r}.$$

7: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **kopt** < 1,
or **kopt** > 3,
or **kopt** $= 2$ when $q \neq 0$,
or **x** < 0.0,
or **s** < 0.0,
or **t** < 0.0,
or **tmat** $<$ **t**,
or **sigma**$(1) \leq 0.0$, with **tdpar**$(3) =$ **false**,
or **sigma**$(i) \leq 0.0$, with **tdpar**$(3) =$ **true**, for some $i = 1$, 2 or 3.

7 Accuracy

Given accurate values of **r**, **q** and **sigma** no further approximations are made in the evaluation of the Black–Scholes analytic formulae, and the results should therefore be within machine accuracy. The values of **r**, **q** and **sigma** returned from d03ne are exact for polynomials of degree up to 3.

8 Further Comments

8.1 Algorithmic Details

The Black–Scholes analytic formulae are used to compute the solution. For a European call option these are as follows:

$$f = Se^{-\hat{q}(T-t)}N(d_1) - Xe^{-\hat{r}(T-t)}N(d_2)$$

where

$$d_1 = \frac{\log(S/X) + \left(\hat{r} - \hat{q} + \bar{\sigma}^2/2\right)(T-t)}{\bar{\sigma}\sqrt{T-t}},$$

$$d_2 = \frac{\log(S/X) + \left(\hat{r} - \hat{q} - \bar{\sigma}^2/2\right)(T-t)}{\bar{\sigma}\sqrt{T-t}} = d_1 - \bar{\sigma}\sqrt{T-t},$$

$N(x)$ is the cumulative Normal distribution function and $N'(x)$ is its derivative

$$N(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\zeta^2/2} d\zeta,$$
$$N'(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

The functions $\hat{q}$, $\hat{r}$, $\hat{\sigma}$ and $\bar{\sigma}$ are average values of q, r and σ over the time to maturity:

$$\hat{q} = \frac{1}{T-t}\int_t^T q(\zeta) d\zeta,$$
$$\hat{r} = \frac{1}{T-t}\int_t^T r(\zeta) d\zeta,$$
$$\hat{\sigma} = \frac{1}{T-t}\int_t^T \sigma(\zeta) d\zeta,$$
$$\bar{\sigma} = \left(\frac{1}{T-t}\int_t^T \sigma^2(\zeta) d\zeta\right)^{1/2}.$$

The Greeks are then calculated as follows:

$$
\begin{aligned}
\Delta &= \frac{\partial f}{\partial S} = e^{-\hat{q}(T-t)}N(d_1) + \frac{Se^{-\hat{q}(T-t)}N'(d_1) - Xe^{-\hat{r}(T-t)}N'(d_2)}{\bar{\sigma}S\sqrt{T-t}}, \\
\Gamma &= \frac{\partial^2 f}{\partial S^2} = \frac{Se^{-\hat{q}(T-t)}N'(d_1) + Xe^{-\hat{r}(T-t)}N'(d_2)}{\bar{\sigma}S^2\sqrt{T-t}} + \frac{Se^{-\hat{q}(T-t)}N'(d_1) - Xe^{-\hat{r}(T-t)}N'(d_2)}{\bar{\sigma}^2S^2(T-t)}, \\
\Theta &= \frac{\partial f}{\partial t} = rf + (q-r)S\Delta - \frac{\sigma^2 S^2}{2}\Gamma, \\
\Lambda &= \frac{\partial f}{\partial \sigma} = \left(\frac{Xd_1e^{-\hat{r}(T-t)}N'(d_2) - Sd_2e^{-\hat{q}(T-t)}N'(d_1)}{\bar{\sigma}^2}\right)\hat{\sigma}, \\
\rho &= \frac{\partial f}{\partial r} = X(T-t)e^{-\hat{r}(T-t)}N(d_2) + \frac{\left(Se^{-\hat{q}(T-t)}N'(d_1) - Xe^{-\hat{r}(T-t)}N'(d_2)\right)\sqrt{T-t}}{\bar{\sigma}}.
\end{aligned}
$$

Note: that Θ is obtained from substitution of other Greeks in the Black–Scholes partial differential equation, rather than differentiation of f. The values of q, r and σ appearing in its definition are the instantaneous values, not the averages. Note also that both the first-order average $\hat{\sigma}$ and the second-order average $\bar{\sigma}$ appear in the expression for Λ. This results from the fact that Λ is the derivative of f with respect to σ, not $\hat{\sigma}$.

For a European put option the equivalent equations are:

$$
\begin{aligned}
f &= Xe^{-\hat{r}(T-t)}N(-d_2) - Se^{-\hat{q}(T-t)}N(-d_1), \\
\Delta &= \frac{\partial f}{\partial S} = -e^{-\hat{q}(T-t)}N(-d_1) + \frac{Se^{-\hat{q}(T-t)}N'(-d_1) - Xe^{-\hat{r}(T-t)}N'(-d_2)}{\bar{\sigma}S\sqrt{T-t}}, \\
\Gamma &= \frac{\partial^2 f}{\partial S^2} = \frac{Xe^{-\hat{r}(T-t)}N'(-d_2) + Se^{-\hat{q}(T-t)}N'(-d_1)}{\bar{\sigma}S^2\sqrt{T-t}} + \frac{Xe^{-\hat{r}(T-t)}N''(-d_2) - Se^{-\hat{q}(T-t)}N''(-d_1)}{\bar{\sigma}^2S^2(T-t)}, \\
\Theta &= \frac{\partial f}{\partial t} = rf + (q-r)S\Delta - \frac{\sigma^2 S^2}{2}\Gamma, \\
\Lambda &= \frac{\partial f}{\partial \sigma} = \left(\frac{Xd_1e^{-\hat{r}(T-t)}N'(-d_2) - Sd_2e^{-\hat{q}(T-t)}N'(-d_1)}{\bar{\sigma}^2}\right)\hat{\sigma}, \\
\rho &= \frac{\partial f}{\partial r} = -X(T-t)e^{-\hat{r}(T-t)}N(-d_2) + \frac{\left(Se^{-\hat{q}(T-t)}N'(-d_1) - Xe^{-\hat{r}(T-t)}N'(-d_2)\right)\sqrt{T-t}}{\hat{\sigma}}.
\end{aligned}
$$

The analytic solution for an American call option with $q = 0$ is identical to that for a European call, since early exercise is never optimal in this case. For all other cases no analytic solution is known.

9 Example

```
kopt = int32(2);
x = 50;
s = 0;
t = 0;
tmat = 0.4166667;
```

```
tdpar = [false; false; false];
r = [0.1];
q = [0];
sigma = [0.4];
[f, theta, delta, gamma, lambda, rho, ifail] = ...
    d03nd(kopt, x, s, t, tmat, tdpar, r, q, sigma)
```

```
f =
     0
theta =
     0
delta =
     0
gamma =
     0
lambda =
     0
rho =
     0
ifail =
           0
```