marxer engineering & computing

www.mec.li

Mathematica und der Finanzbereich

Mathematica mit seinen Fähgikeiten als numerische und symbolische Rechenmaschine, für Monte-Carlo Simulationen, den umfassenden Plot Möglichkeiten, der Datenbankanbindung, der WEB Anbindung etc. ist auch im Finanzbereich vielseitig einsetzbar. Zusätzlich gibt es kommerzielle Packages, die speziell auf den Finanzbereich ausgerichtet sind und den Funktionsumfang in diesem Bereich erweitern.

Ganze Applikationen können in *Mathematica* geschrieben werden. Aus Platzgründen sind hier jedoch wiederum nur ganz kurze Code Snippets aufgeführt. Die ersten zwei Beispiele zeigen, wie mit verschiedenen Methoden einfach auf Daten im Web zugegriffen werden kann. Das dritte Beispiel zeigt die Definition der Black Scholes Formel. Zu beachten ist auch, wie einfach ansprechende Plots erzeugt werden können.

Aktueller Währungskurs aus dem WEB

Mathematica bietet auch die Möglichkeit via SOAP und WSDL direkt auf Angebote auf dem web zuzugreifen.

In einem ersten Schritt werden die notwendigen Services (Jlink und WebServices) installiert. Der WebService stellt die Funktion getRate zur Verfügung.

```
Needs["JLink`"]; Needs["WebServices`"];
InstallService["http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl"]
```

```
{getRate}
```

Mit einem einfachen Aufruf kann Wechselkurs USD zu Euro abgefragt werden.

```
getRate["US", "EURO"]
```

```
0.7655
```

Aktuelle Aktienpreise aus dem WEB

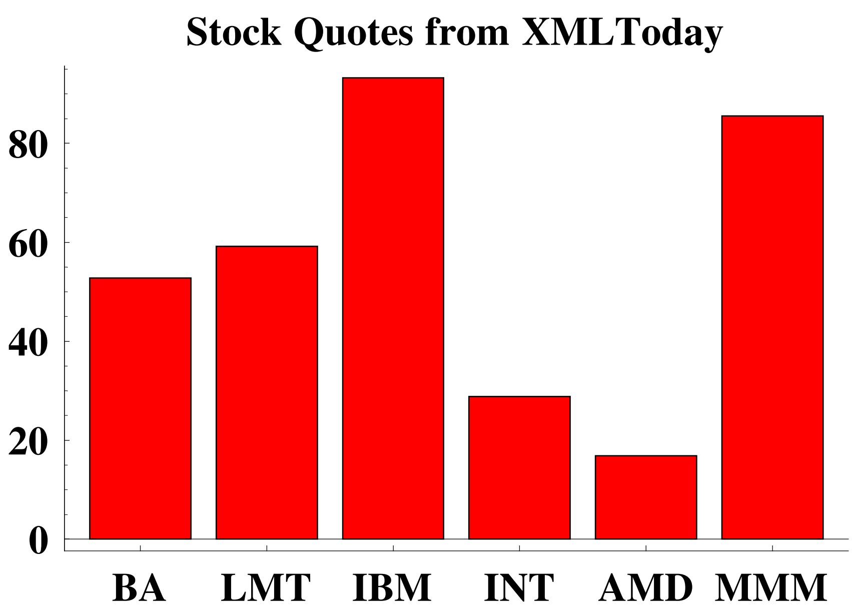
Dieses Beispiel benutzt die XMLGet Funktion von *Mathematica*, um Aktienkurse aus dem WEB zu erhalten. Mit dem "?s" Parameter kann die Auswahl der Aktien bestimmt werden.

```
xmlquotes = XML`Parser`XMLGet["http://www.xmltoday.com/
     examples/stockquote/getxmlquote.vep?s=ba+lmt+ibm+int+amd+mmm"] ;
GetQuotes[xml_] := Cases[xml, XMLElement["stock_quote", _,
     {___, XMLElement["symbol", _, {symbol_}], ___,
      XMLElement["price", {"type" -> "ask", "value" → v_}, {}], ___}] ⧴
    {symbol, ToExpression[v]}, Infinity];
quotes = GetQuotes[xmlquotes]
```

```
{{BA, 52.78}, {LMT, 59.19}, {IBM, 93.27}, {INT, 28.83}, {AMD, 16.9}, {MMM, 85.55}}
```

In einem zweiten Schritt werden die Kurse dargestellt

```
Needs["Graphics`Graphics`"];
BarChart[#[[2]] & /@ quotes, BarLabels → (#[[1]] & /@ quotes),
  TextStyle → {FontFamily -> "Times", FontSize → 15, FontWeight -> "Bold"},
  Frame → {True, True, False, False}, ImageSize → 600,
  PlotLabel -> "Stock Quotes from XMLToday"];
```



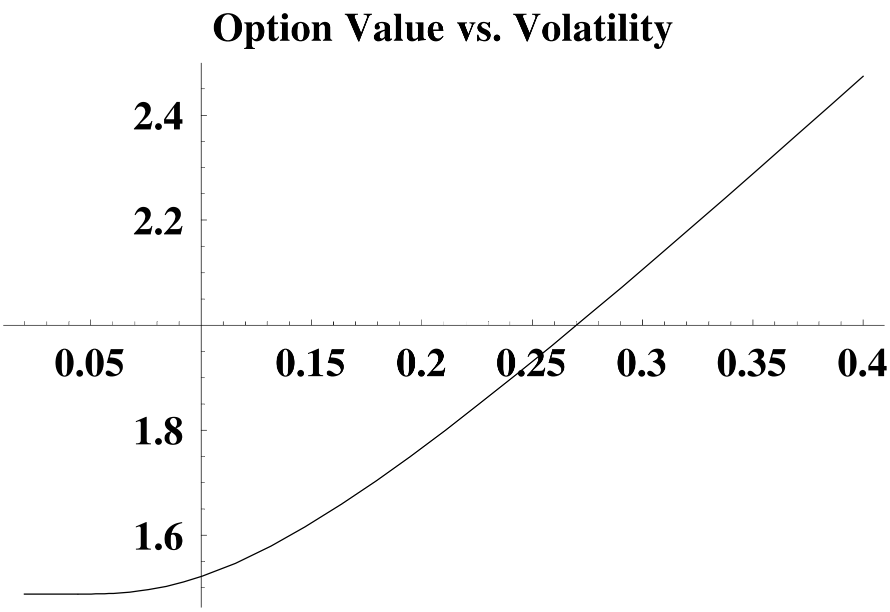
Black Scholes Modell

Im folgenden wird der einfache Fall einer vanilla European call Option (Black and Scholes, 1973) in symbolischer Weise definiert.

```
norm[z_?NumberQ] := N[0.5 Erf[z/√2] + 0.5]; norm[x_] := 1/2 (1 + Erf[x/√2]);
done[s_, σ_, k_, t_, r_, q_] := ((r - q) t + Log[s/k])/(σ √t) + (σ √t)/2;
dtwo[s_, σ_, k_, t_, r_, q_] := ((r - q) t + Log[s/k])/(σ √t) - (σ √t)/2;
BlackScholesCall[s_, k_, v_, r_, q_, t_] :=
 s Exp[-q t] norm[done[s, v, k, t, r, q]] - k Exp[-r t] norm[dtwo[s, v, k, t, r, q]];
```

Der Wert der Option wird als Funktion der Volatility geplottet mit folgenden Parametern für die Option: Strike 10, Underlying 11, Expiry 1 year, Risk-free Rate 5% continously compounded, Dividend 0.

```
Plot[
  BlackScholesCall[11, 10, vol, 0.05, 0, 1], {vol, 0.02, 0.40},
  PlotLabel -> "Option Value vs. Volatility",
  PlotRange → All,
  TextStyle → {FontFamily -> "Times", FontSize → 15, FontWeight -> "Bold"}
 ];
```



Source: "Modeling Financial Derivatives with *Mathematica*", William T. Shaw, Chapter1.nb in "http://library.wolfram.com/infocenter/Books/3631/"