1 Functions to calculate Numerical Derivatives and Hessian Matrix

In R, the functions in this package are made available with

```
> library("numDeriv")
>
```

The code from the vignette that generates this guide can be loaded into an editor with *edit(vignette("Guide", package="numDeriv"))*. This uses the default editor, which can be changed using *options()*.

Here are some examples of grad.

```
>  grad(sin, pi)
>   grad(sin, (0:10)*2*pi/10)
>   func0 <- function(x){ sum(sin(x))  }
>   grad(func0 , (0:10)*2*pi/10)
>   func1 <- function(x){ sin(10*x) - exp(-x) }
>   curve(func1,from=0,to=5)
>   x <- 2.04
>   numd1 <- grad(func1, x)
>   exact <- 10*cos(10*x) + exp(-x)
>   c(numd1, exact, (numd1 - exact)/exact)
>   x <- c(1:10)
>   numd1 <- grad(func1, x)
>   exact <- 10*cos(10*x) + exp(-x)
>   cbind(numd1, exact, (numd1 - exact)/exact)
```

Here are some examples of jacobian.

```
>   func2 <- function(x) c(sin(x), cos(x))
>    x <- (0:1)*2*pi
>    jacobian(func2, x)
```

Here are some examples of hessian.

```
> x <- 0.25 * pi
> hessian(sin, x)
> fun1e <- function(x) sum(exp(2*x))
> x <- c(1, 3, 5)
> hessian(fun1e, x, method.args=list(d=0.01))
```

Here are some examples of genD.

```
>     func <- function(x){c(x[1], x[1], x[2]^2)}
>     z <- genD(func, c(2,2,5))
>     z
```