Interpolation with Generalized Vandermonde Matrices

In [4]:

```
#keep
import numpy as np
import numpy.linalg as la
import matplotlib.pyplot as pt
%matplotlib inline
```

Take a function and its derivative

In [9]:

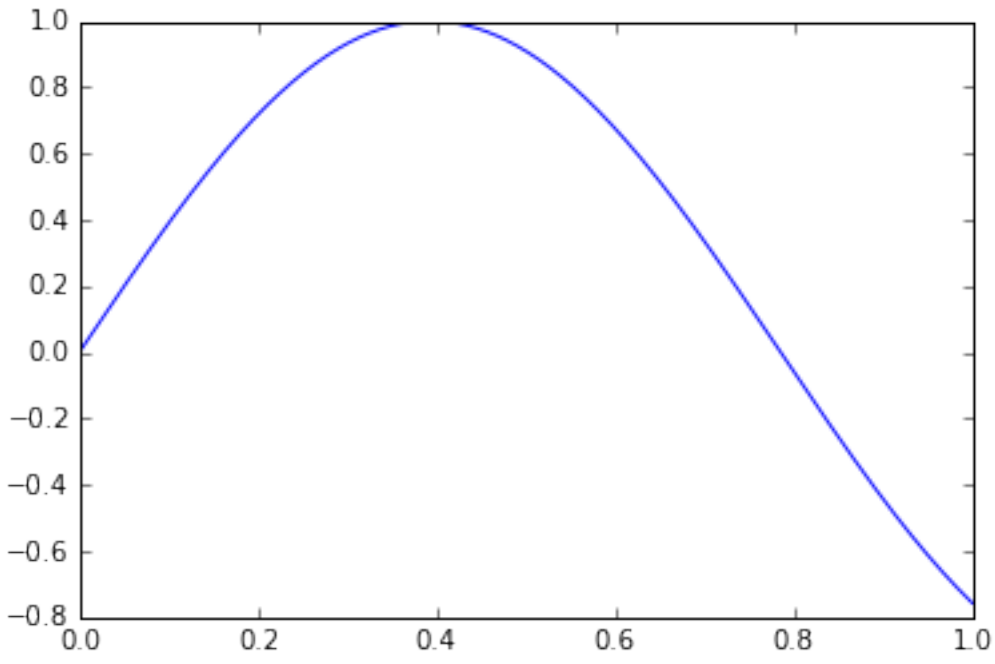
```
def f(x):
    return np.sin(4*x)
def df(x):
    return 4*np.cos(4*x)
```

In [10]:

```
x = np.linspace(0, 1, 1000)
pt.plot(x, f(x))
```

Out[10]:

```
[<matplotlib.lines.Line2D at 0x10d8aac50>]
```



Fix some parameters:

In [13]:

```
degree = 2
nodes = np.linspace(0, 1, degree+1)
print(nodes)
```

```
[ 0.   0.5  1. ]
```

Build the Vandermonde matrix:

In [22]:

```
V = np.array([
    nodes**i
    for i in range(degree+1)
]).T
print(V)
```

```
[[ 1.    0.    0.  ]
 [ 1.    0.5   0.25]
 [ 1.    1.    1.  ]]
```

Now find the interpolation coefficients as `coeffs`:

In [23]:

```
coeffs = la.solve(V, f(nodes))
```

In [24]:

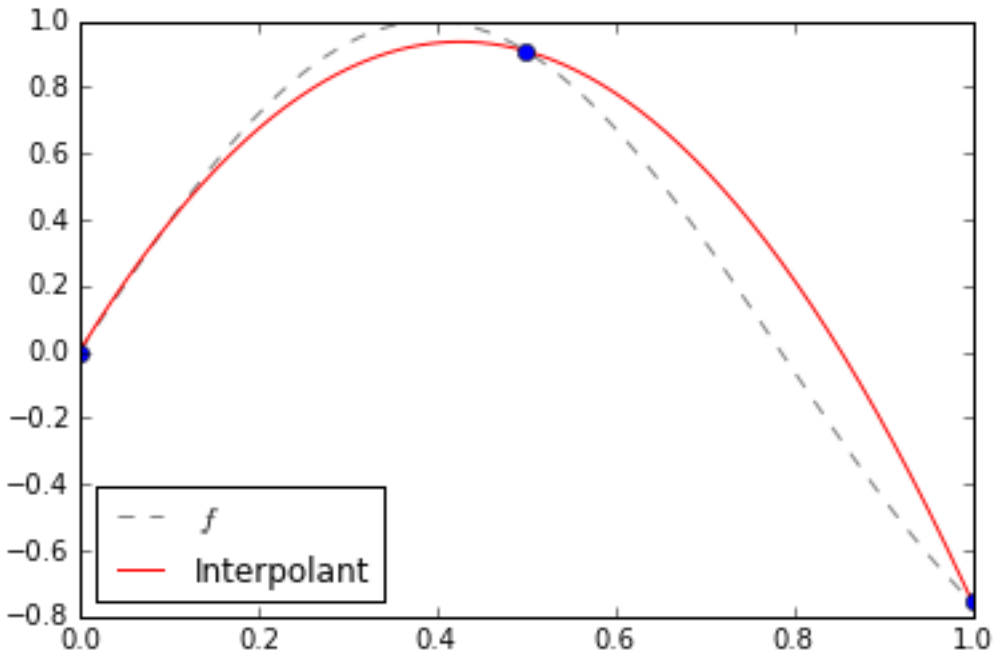
```
interp = 0*x
for i in range(degree+1):
    interp += coeffs[i] * x**i
```

In [25]:

```
pt.plot(x, f(x), "--", color="gray", label="$f$")
pt.plot(x, interp, color="red", label="Interpolant")
pt.plot(nodes, f(nodes), "o")
pt.legend(loc="best")
```

Out[25]:

```
<matplotlib.legend.Legend at 0x10dbf5780>
```



In [ ]: