Problem: PERTURBATION

- **Find square-root of 10:**

$$\sqrt{10} = ? \tag{1}$$

- **Polynomial equation:**

$$[x^2 - 9 - \epsilon = 0] \tag{2}$$

where ϵ = 1

In [8]:
```python
import numpy as np
import scipy as sp
from sympy import *
from sympy import solve
from scipy.optimize import fsolve
import numpy.polynomial.polynomial as root
import math as mt
import pandas as pd
import matplotlib.pyplot as plt
```

Defining function and its derivative:

Perturbation-method:

- Expanding **x** in powers ϵ**.**

$$x = (x_0 + \epsilon * x_1 + \epsilon^2 * x_2 + \epsilon^3 * x_3 + \dots) \tag{3}$$

- Substituting in equation(2):

$$(x_0 + \epsilon * x_1 + \epsilon^2 * x_2 + \epsilon^3 * x_3 + \dots)^2 - 9 - \epsilon = 0 \tag{4}$$

- Seperating powers of ϵ:

$$(x_0^2 - 9) + \epsilon(2 * x_0 * x_1 - 1) + \epsilon^2(x_1^2 + 2 * x_0 * x_2) + \dots = 0 \tag{5}$$

- Since, ϵ is **arbitrary** each term must be individually equal to zero:
  - o(ϵ^0)

$$(x_0^2 - 9) = 0 \tag{6}$$

  - o(ϵ^1)

$$(2 * x_0 * x_1 - 1) = 0 \tag{7}$$

  - o(ϵ^2)

$$(x_1^2 + 2 * x_0 * x_2) = 0 \tag{8}$$

  - o(ϵ^3)

$$(2 * x_1 * x_2 + 2 * x_0 * x_3) = 0 \tag{9}$$

- o(ϵ^4)

$$(x_2^2 + 2 * x_1 * x_3 + 2 * x_0 * x_4) = 0 \tag{10}$$

- o(ϵ^5)

$$(2 * x_2 * x_3 + 2 * x_1 * x_4 + 2 * x_0 * x_5) = 0 \ldots \tag{11}$$

- Solving for x_0,x_1...

$$x_0 = 3, -3 \tag{12}$$

$$x_1 = \frac{1}{2 * x_0} = \frac{1}{6}, \frac{-1}{6} \tag{13}$$

$$x_2 = \frac{-x_1^2}{2 * x_0} = \frac{-1}{216}, \frac{1}{216} \ldots \tag{14}$$

- Similarly, solving for:

$$x_3, x_4, x_5 \ldots \tag{15}$$

  - Adding, them in **equation(3)** cummulatively.

In [17]:
```
ss1    = np.array([3, 3.16666666667, 3.162037037, 3.162294239, 3.162033465])
ext    = np.ones(len(ss))*mt.sqrt(10)
error1 = np.abs(ext-ss)
```

In [20]:
```
dict1 = {"Sol_ϵ=1":ss1,"Error":error1 }
df1 = pd.DataFrame(dict1)
ERROR1 = np.abs(ss1[len(ss1)-1]-mt.sqrt(10))

print("\nMAX_ERROR:      ", ERROR1)
print("Converged sol: ", ss1[len(ss1)-1])
print("Excat sol:     ", mt.sqrt(10))
print('\n', df1)
```

```
MAX_ERROR:      0.00024419516837959065
Converged sol:  3.162033465
Excat sol:      3.1622776601683795

     Sol_ϵ=1     Error
0  3.000000  0.162278
1  3.166667  0.004389
2  3.162037  0.000241
3  3.162294  0.000017
4  3.162033  0.000244
```

Plotting:

In [21]:
```
%matplotlib inline
%config InlineBackend.fig_format = 'svg'
```

In [22]:
```
plt.figure(figsize=(5,7))

# Solution-plot
fig1 = plt.subplot(2,1,1)
plt.plot(np.arange(0,len(ss1),1),ss1, "m--o", label ="ϵ=1")
plt.title("Solution plot", fontweight ="bold", fontsize=15, color="blue")
plt.ylabel("Root_of_10", fontsize=10,color="darkred")
plt.xlabel("No.Iterations", fontsize=10,color="darkred")
```

```python
plt.subplots_adjust(hspace=0.4)
plt.legend()

# Error-plot
plt.subplot(2,1,2)
plt.plot(np.arange(0,len(error1),1),error1,"m--o",label="ϵ=1")
plt.title("Error Plot", fontweight ="bold", fontsize=15,color = "blue")
plt.ylabel("Error", fontsize=10,color="darkred")
plt.xlabel("No.Iterations", fontsize=10,color="darkred")
plt.legend()
```

Out[22]: <matplotlib.legend.Legend at 0x1a95630da30>