printing the Hessian matrix at the converged value of $\boldsymbol{x}$, it is observed that $[A]$ approaches $[\boldsymbol{H}]^{-1}$.

```
      Initial function value = 1452.2619
No.       x-vector          f(x)          Deriv.
_______________________________________________

1     0.095       0.023     -2.704       1006.074
2     0.179       0.145     -5.418         37.036
3     0.508       0.145     -9.576         23.983
4     0.501       0.122     -9.656          7.004
5     0.504       0.122     -9.656          0.396
6     0.504       0.122     -9.656          0.053
7     0.504       0.122     -9.656          0.038
8     0.504       0.122     -9.656          0.028
9     0.504       0.122     -9.656          0.005
_______________________________________________

>> A
A =
    0.0091    0.0005
    0.0005    0.0033
>> inv(hessian(x,delx,n_of_var))
ans =
    0.0091    0.0005
    0.0005    0.0033
```

3.4.7 BFGS Method

In the BFGS method, the Hessian is approximated using the variable metric matrix $[A]$ given by the equation

$$[\boldsymbol{A}]_{i+1} = [\boldsymbol{A}]_i + \frac{g\nabla g^T}{\nabla g^T \Delta \boldsymbol{x}} + \frac{\nabla f(\boldsymbol{x}_i)\nabla f(\boldsymbol{x}_i)^T}{\nabla f(\boldsymbol{x}_i)^T \boldsymbol{S}_i} \tag{3.28}$$

It is important to note that whereas the matrix $[A]$ converges to the inverse of the Hessian in the DFP method, the matrix $[A]$ converges to the Hessian itself in the BFGS method. As the BFGS method needs fewer restarts as compared to the DFP method, it is more popular than the DFP method. The algorithm for the BFGS method is described in Table 3.8 and a MATLAB code of its implementation is given in *BFGS.m*.

On executing the code with a starting value of $\boldsymbol{x}$ as (–3, 2) the following output is displayed in the command window for the test problem. Again, it is observed that in the second and third iterations, search points are similar to this method as compared to DFP and the conjugate gradient methods, indicating that search directions were similar. Further, on typing A in the

TABLE 3.8

Algorithm for the BFGS Method

Step 1: Given x_i (starting value of design variable)
ε_1 (tolerance of function value from previous iteration)
ε_2 (tolerance on gradient value)
Δx (required for gradient computation)
$[A]$ (initialize to identity matrix)

Step 2: Compute $f(x_i)$ and $\nabla f(x_i)$ (function and gradient vector)
$S_i = -\nabla f(x_i)$ (search direction)
$x_{i+1} = x_i + \alpha S_i$ (update the design vector)
Minimize $f(x_{i+1})$ and determine α (use golden section method)

Step 3: Compute Δx and ∇g

$$[A]_{i+1} = [A]_i + \frac{g\nabla g^T}{\nabla g^T \Delta x} + \frac{\nabla f(x_i)\nabla f(x_i)^T}{\nabla f(x_i)^T S_i}$$

$S_{i+1} = -[[A]_{i+1}]^{-1}\nabla f(x_{i+1})$
$x_{i+2} = x_{i+1} + \alpha S_{i+1}$
Minimize $f(x_{i+2})$ and determine α (use the golden section method)
If $|f(x_{i+2}) - f(x_{i+1})| > \varepsilon_1$ or $\|\nabla f(x_{i+1})\| > \varepsilon_2$
then goto Step 3
else goto Step 4

Step 4: Converged. Print $x^* = x_{i+2}$, $f(x^*) = f(x_{i+2})$

MATLAB command prompt and then printing the Hessian matrix at the converged value of x, it is observed that $[A]$ approaches $[H]$.

```
      Initial function value = 1452.2619
No.        x-vector          f(x)          Deriv.
---------------------------------------------------
1     0.095       0.023     -2.704       1006.074
2     0.179       0.145     -5.418         37.036
3     0.508       0.145     -9.578         24.017
4     0.501       0.122     -9.655          6.900
5     0.504       0.122     -9.656          0.471
6     0.504       0.122     -9.656          0.077
7     0.504       0.122     -9.656          0.056
8     0.504       0.122     -9.656          0.040
9     0.504       0.122     -9.656          0.007
---------------------------------------------------

>> A
A =
    110.5001  -16.9997
    -16.9997  306.7238
>> hessian(x,delx,n_of_var)
ans =
    111.0981  -15.9640
    -15.9640  308.5603
```