This handout gives a pseudocode description of an algorithm for creating fractal terrains using midpoint displacement and successive random additions. Slightly modified from code provided by Richard F. Voss as found in *The Science of Fractal Images* by Barnsley, Devaney, Mandelbrot, Peitgen, Saupe, and Voss, pages 100–101.

```
MidPointFM2D (X, maxlevel, sigma, H, seed)

Arguments    X[][]       2D array of size (N+1)^2
             maxlevel    maximal number of recursions.  N = 2^maxlevel
             sigma       initial standard deviation
             seed        seed value for random number generator

Variables    i, N, stage
             delta              standard deviation for current level
             x, y, y0, D, d     indexing variables

Functions    f3(delta, x0, x1, x2) = (x0+x1+x2)/3 + delta * Gauss()
             f4(delta, x0, x1, x2, x3) = (x0+x1+x2+x3)/4 + delta *
                Gauss()


BEGIN
    InitGauss(seed)
    N = 2^maxlevel

    /* set the initial random corners */
    delta := sigma
    X[0][0] := delta * Gauss()
    X[0][N] := delta * Gauss()
    X[N][0] := delta * Gauss()
    X[N][N] := delta * Gauss()
    D := N
    d := N/2
    FOR stage :=1 TO maxlevel DO
        /* going from grid type I to type II */
        delta := delta * power(0.5, 0.5*H)

        /* interpolate and offset mid points */
        FOR x:=d TO N-d STEP D DO
            FOR y:= d TO N-d STEP D DO
                X[x][y]:= f4(delta, X[x+d][y+d], X[x+d][y-d],
                                    X[x-d][y+d], X[x-d][y-d])
            END FOR
        END FOR
```

```
/* displace existing points */
FOR x:=0 TO N STEP D DO
    FOR y:= 0 TO N STEP D DO
        X[x][y] := X[x][y] + delta * Gauss()
    END FOR
END FOR

/* going from grid type II to type I */
delta := delta * power(0.5, 0.5*H)

/* interpolate and offset mid points at boundary */
FOR x:= d TO N-d STEP D DO
    X[x][0] := f3(delta, X[x+d][0], X[x-d][0], X[x][d])
    X[x][N] := f3(delta, X[x+d][N], X[x-d][N], X[x][N-d])
    X[0][x] := f3(delta, X[0][x+d], X[0][x-d], X[d][x])
    X[N][x] := f3(delta, X[N][x+d], X[N][x-d], X[N-d][x])
END FOR

/* interpolate and offset mid points in interior */
FOR x:= d TO N-d STEP D DO
    FOR y:= D TO N-d STEP D DO
        X[x][y] := f4(delta, X[x][y+d], X[x][y-d],
                             X[x+d][y], X[x-d][y])
    END FOR
END FOR

FOR x:= D TO N-d STEP D DO
    FOR y:= d TO N-d STEP D DO
        X[x][y] := f4(delta, X[x][y+d], X[x][y-d],
                             X[x+d][y], X[x-d][y])
    END FOR
END FOR

/* displace existing points */
FOR x:= 0 TO N STEP D DO
    FOR y:= 0 TO N STEP D DO
        X[x][y] := X[x][y] + delta * Gauss()
    END FOR
END FOR
FOR x:= d TO N-d STEP D DO
    FOR y:= d TO N-d STEP D DO
        X[x][y] := X[x][y] + delta * Gauss()
    END FOR
END FOR
```

```
        /* prepare for next level */
        D := D/2
        d := d/2
    END FOR
END
```