

5. 行列関連

5.1 行列の加算と 2 次元配列

行列の加算を素直に表現したプログラムは、以下のようなになるでしょう。

```
void addMat(int N1, int N2, double A[][3], double B[][3], double C[][3]) {
    for(int i = 0; i < N1; i++)
        for(int j = 0; j < N2; j++) C[i][j] = A[i][j] + B[i][j];
}
```

上の例では列数を固定的に指定してありますが、C ではここに変数名を記述することができません。列数を省略するとコンパイルエラーになります。そこで、次のように 1 次元配列で代用するのが一般的です。

```
void addMat2(int N1, int N2, double A[], double B[], double C[]) {
    for(int i = 0, ii = 0; i < N1; i++, ii += N2)
        for(int j = 0; j < N2; j++) C[ii+j] = A[ii+j] + B[ii+j];
}
```

すなわち、A[i][j] を A[ii+j]と読み替えて記述するわけです。もともと、この部分は同一箇所の加算ですから、次のように簡単に記述してもよいでしょう。

```
void addMat3(int N1, int N2, double A[], double B[], double C[]) {
    for(int i = 0; i < N1*N2; i++) C[i] = A[i] + B[i];
}
```

どうしても 2 次元配列の形で記述したい場合、ポインタ配列を使います。

```
void addMat4(int N1, int N2, double *A[], double *B[], *double C[]) {
    for(int i=0; i < N1; i++) for(int j=0; j < N2; j++) C[i][j] = A[i][j]+B[i][j];
}
```

ただし、<stdarg.h>をインクルードし、初期設定する関数を用意しておきましょう。

```
double *setArray(int N, ...) { /* ... は可変リストであることを示す */
    double *B; B=(double *)calloc(N, sizeof(double));
    va_list ap; va_start(ap, N); /* N の後が可変リストの開始 */
    for(int i=0; i < N; i++) B[i] = va_arg(ap, double); /* 順次引数を取り込む */
    va_end(ap); /* これを忘れないこと */
    return B;
}
```

初期設定では、たとえば次のように呼び出します。

```
double *B[3];
B[0]=setArray(3, 1.0, 2.0, 3.0);
B[1]=setArray(3, 4.0, 5.0, 6.0);
B[2]=setArray(3, 7.0, 8.0, 9.0);
```