研究内容紹介

FDTD (Finite Difference Time Domain)

1. 理論

基礎方程式より規格化された Maxwell 方程式は

$$\frac{\partial}{\partial t}\boldsymbol{B} = -\nabla \times \boldsymbol{E} \tag{1}$$

$$\frac{\partial}{\partial t}\boldsymbol{E} = \nabla \times \boldsymbol{B} - \boldsymbol{J} \tag{2}$$

(1) 式は、右辺の電場の回転微分により磁場の時間発展が求まり、(2) 式は、右辺の磁場の回転微分と電流密度から電場の時間発展を求めることができる。Maxwell 方程式を中央差分法を用いて解く方法を FDTD 法といい、数値電磁場解法として最も良く知られている方法の 1 つである。解法には Fig.1 のような空間格子点を想定する。各格子点上において各物理量が計算・記憶されることとなる。

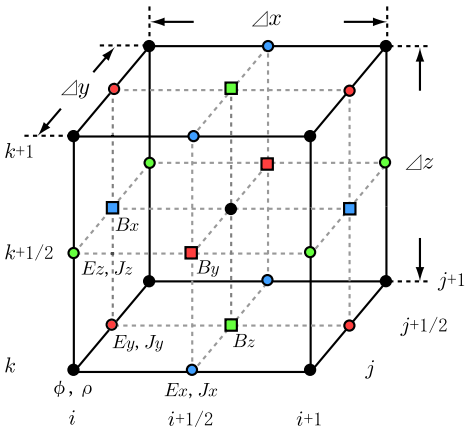

Fig. 1　空間格子点の定義

(1) 式を中央差分化すると、以下式が得られる。

$$\frac{B^{n+\frac{1}{2}}_{x(i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2})} - B^{n-\frac{1}{2}}_{x(i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2})}}{\Delta t} = \frac{E^{n}_{y(i,j+\frac{1}{2},k+1)} - E^{n}_{y(i,j+\frac{1}{2},k)}}{\Delta z} - \frac{E^{n}_{z(i,j+1,k+\frac{1}{2})} - E^{n}_{z(i,j,k+\frac{1}{2})}}{\Delta y} \tag{3}$$

$$\frac{B^{n+\frac{1}{2}}_{y(i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2})} - B^{n-\frac{1}{2}}_{y(i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2})}}{\Delta t} = \frac{E^{n}_{z(i+1,j,k+\frac{1}{2})} - E^{n}_{z(i,j,k+\frac{1}{2})}}{\Delta x} - \frac{E^{n}_{x(i,j+\frac{1}{2},k+1)} - E^{n}_{x(i,j+\frac{1}{2},k)}}{\Delta z} \tag{4}$$

$$\frac{B^{n+\frac{1}{2}}_{z(i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2})} - B^{n-\frac{1}{2}}_{z(i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2})}}{\Delta t} = \frac{E^{n}_{x(i,j+1,k+\frac{1}{2})} - E^{n}_{x(i,j,k+\frac{1}{2})}}{\Delta y} - \frac{E^{n}_{y(i+1,j,k+\frac{1}{2})} - E^{n}_{y(i,j,k+\frac{1}{2})}}{\Delta x} \tag{5}$$

同様に (2) 式を差分化すると

$$\frac{E^{n+1}_{x(i,j+\frac{1}{2},k+\frac{1}{2})} - E^{n}_{x(i,j+\frac{1}{2},k+\frac{1}{2})}}{\Delta t} = \frac{B^{n+\frac{1}{2}}_{z(i,j+1,k+\frac{1}{2})} - B^{n+\frac{1}{2}}_{z(i,j,k+\frac{1}{2})}}{\Delta y} - \frac{B^{n+\frac{1}{2}}_{y(i,j+\frac{1}{2},k+1)} - B^{n+\frac{1}{2}}_{y(i,j+\frac{1}{2},k)}}{\Delta z} - J^{n+\frac{1}{2}}_{x(i,j+\frac{1}{2},k+\frac{1}{2})} \tag{6}$$

$$\frac{E^{n+1}_{y(i+\frac{1}{2},j,k+\frac{1}{2})} - E^{n}_{y(i+\frac{1}{2},j,k+\frac{1}{2})}}{\Delta t} = \frac{B^{n+\frac{1}{2}}_{x(i+\frac{1}{2},j,k+1)} - B^{n+\frac{1}{2}}_{x(i+\frac{1}{2},j,k)}}{\Delta y} - \frac{B^{n+\frac{1}{2}}_{z(i+1,j,k+\frac{1}{2})} - B^{n+\frac{1}{2}}_{z(i,j,k+\frac{1}{2})}}{\Delta z} - J^{n+\frac{1}{2}}_{y(i,j+\frac{1}{2},k+\frac{1}{2})} \tag{7}$$

$$\frac{E^{n+1}_{z(i+\frac{1}{2},j+\frac{1}{2},k)} - E^{n}_{z(i+\frac{1}{2},j+\frac{1}{2},k)}}{\Delta t} = \frac{B^{n+\frac{1}{2}}_{y(i+1,j+\frac{1}{2},k)} - B^{n+\frac{1}{2}}_{y(i,j+\frac{1}{2},k)}}{\Delta x} - \frac{B^{n+\frac{1}{2}}_{x(i+\frac{1}{2},j+1,k)} - B^{n+\frac{1}{2}}_{x(i+\frac{1}{2},j,k)}}{\Delta y} - J^{n+\frac{1}{2}}_{z(i,j+\frac{1}{2},k+\frac{1}{2})} \tag{8}$$

となる。プログラム上では、1/2 などの小数をそのまま配列で用いることはできないが、1/2 → 0 としプロ

グラミングすれば良い。すると、上式群から以下の差分式を得ることができる。プログラミングしやすいように言語的な形式で表記する。

$$
\begin{aligned}
B_x^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}] &= B_x^{n-\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}] \\
&+\frac{\Delta t}{\Delta z}(E_y^n[\mathrm{i}][\mathrm{j}][\mathrm{k}] - E_y^n[\mathrm{i}][\mathrm{j}][\mathrm{k}-1]) \\
&-\frac{\Delta t}{\Delta y}(E_z^n[\mathrm{i}][\mathrm{j}][\mathrm{k}] - E_z^n[\mathrm{i}][\mathrm{j}-1][\mathrm{k}]) \qquad (9)
\end{aligned}
$$

$$
\begin{aligned}
B_y^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}] &= B_y^{n-\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}] \\
&+\frac{\Delta t}{\Delta x}(E_z^n[\mathrm{i}][\mathrm{j}][\mathrm{k}] - E_z^n[\mathrm{i}-1][\mathrm{j}][\mathrm{k}]) \\
&-\frac{\Delta t}{\Delta z}(E_x^n[\mathrm{i}][\mathrm{j}][\mathrm{k}] - E_x^n[\mathrm{i}][\mathrm{j}][\mathrm{k}-1]) \qquad (10)
\end{aligned}
$$

$$
\begin{aligned}
B_z^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}] &= B_z^{n-\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}] \\
&+\frac{\Delta t}{\Delta y}(E_x^n[\mathrm{i}][\mathrm{j}][\mathrm{k}] - E_x^n[\mathrm{i}][\mathrm{j}-1][\mathrm{k}]) \\
&-\frac{\Delta t}{\Delta x}(E_y^n[\mathrm{i}][\mathrm{j}][\mathrm{k}] - E_y^n[\mathrm{i}-1][\mathrm{j}][\mathrm{k}]) \qquad (11)
\end{aligned}
$$

電場の差分式は

$$
\begin{aligned}
E_x^{n+1}[\mathrm{i}][\mathrm{j}][\mathrm{k}] &= E_x^{n-1}[\mathrm{i}][\mathrm{j}][\mathrm{k}] \\
&+\frac{\Delta t}{\Delta y}(B_z^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}+1][\mathrm{k}] - B_z^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}]) \\
&-\frac{\Delta t}{\Delta z}(B_y^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}+1] - B_y^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}]) \\
&-\Delta t\, J_x^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}] \qquad (12)
\end{aligned}
$$

$$
\begin{aligned}
E_y^{n+1}[\mathrm{i}][\mathrm{j}][\mathrm{k}] &= E_y^{n-1}[\mathrm{i}][\mathrm{j}][\mathrm{k}] \\
&+\frac{\Delta t}{\Delta z}(B_x^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}+1] - B_x^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}]) \\
&-\frac{\Delta t}{\Delta x}(B_z^{n+\frac{1}{2}}[\mathrm{i}+1][\mathrm{j}][\mathrm{k}] - B_z^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}]) \\
&-\Delta t\, J_y^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}] \qquad (13)
\end{aligned}
$$

$$
\begin{aligned}
E_z^{n+1}[\mathrm{i}][\mathrm{j}][\mathrm{k}] &= E_z^{n-1}[\mathrm{i}][\mathrm{j}][\mathrm{k}] \\
&+\frac{\Delta t}{\Delta x}(B_y^{n+\frac{1}{2}}[\mathrm{i}+1][\mathrm{j}][\mathrm{k}] - B_y^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}]) \\
&-\frac{\Delta t}{\Delta y}(B_x^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}+1][\mathrm{k}] - B_x^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}]) \\
&-\Delta t\, J_z^{n+\frac{1}{2}}[\mathrm{i}][\mathrm{j}][\mathrm{k}] \qquad (14)
\end{aligned}
$$

となる。プログラムには上式の差分式を用いればよい。

2. 格子点設計

3 次元時では、2 次元時と比べ、スタッガード格子が複雑になるため、格子点宣言や計算範囲の指定が難解になる。そこで、ここでは 3 次元格子点設計について考える。まず、最初に格子点宣言に述べよう。ここで (imx,jmx,kmx) は、(x, y, z) 方向の**セル**数を表している。

プログラム言語で電磁場の配列宣言する場合は、

```
/* C 言語 */
double Bx[imx+1][jmx+2][kmx+2];
double By[imx+2][jmx+1][kmx+2];
double Bz[imx+2][jmx+2][kmx+1];

double Ex[imx+2][jmx+1][kmx+1];
double Ey[imx+1][jmx+2][kmx+1];
double Ez[imx+1][jmx+1][kmx+2];
```

とすればよく、これで電磁場の各成分ごとに (imx,jmx,kmx) 個のセルを宣言したことになる。全セル数は tg = imx × jmx × kmx となる。

Fig.2 と Fig.3 にモデル化した格子点概念図を示す。青色が x 成分を、赤色が y 成分を、緑色が z 成分をそれぞれ意味しており、ズレている方向を示唆している。電場 $\boldsymbol{E}$ が成分方向に 1 つ余計に格子点が多いのに対し、磁場 $\boldsymbol{B}$ は成分方向以外の成分が 1 つ余計に必要となる。このように、各成分ごとに宣言する格子点数が異なるのは、スタッガード格子点を想定しているためである。ただし、電磁場の各成分のセル数は (imx,jmx,kmx) で宣言される。どうして、宣言する格子点数が異なるのに、同数のセル数が確保されるのかについては、実際に格子点設計図* を描いて見ると良い。

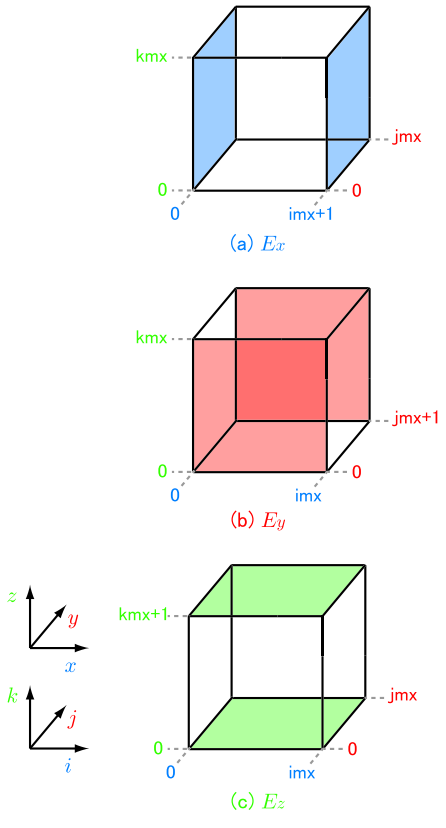

Fig. 2　電場の配列宣言の範囲

*格子点設計図 (2 次元) : docs/xpoints.eps

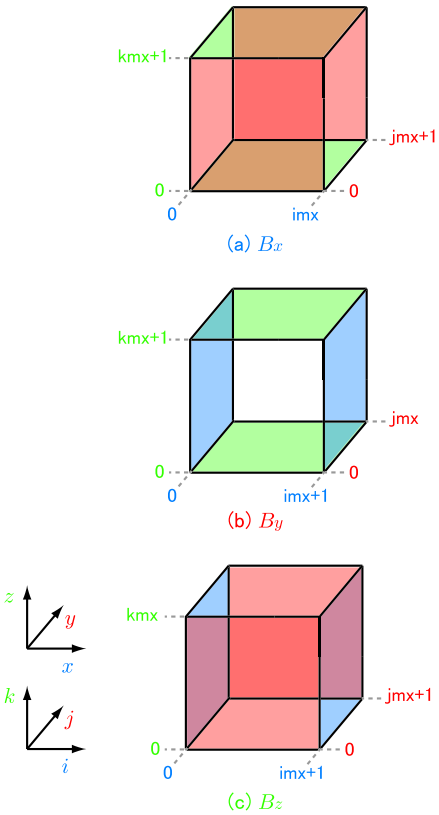

Fig. 3　磁場の配列宣言範囲

3. 計算範囲

次に実際に計算を実行する際の計算範囲を示す。計算範囲は、格子点設計の仕様や用いる差分式、境界条件等により異なる場合がある。しかし、どの場合にせよ、境界を除いたすべての格子点が計算されていること、存在しない格子点が計算に含まれない等の条件が満たされている必要がある。

磁場計算範囲

```
/* Bx */
for(i=0;i<=imx;i++)
  for(j=1;j<=jmx;j++)
    for(k=1;k<=kmx;k++)
/* By */
for(i=1;i<=imx;i++)
  for(j=0;j<=jmx;j++)
    for(k=1;k<=kmx;k++)
/* Bz */
for(i=1;i<=imx;i++)
  for(j=1;j<=jmx;j++)
    for(k=0;k<=kmx;k++)
```

電場計算範囲

```
/* Ex */
for(i=1;i<=imx;i++)
  for(j=0;j<=jmx;j++)
    for(k=0;k<=kmx;k++)
/* Ey */
for(i=0;i<=imx;i++)
  for(j=1;j<=jmx;j++)
    for(k=0;k<=kmx;k++)
/* Ez */
for(i=0;i<=imx;i++)
  for(j=0;j<=jmx;j++)
    for(k=1;k<=kmx;k++)
```

計算範囲の終端値は、電磁場の各成分とも (imx,jmx, kmx) となっているが、初期値がそれぞれ異なっている。これは、格子点がそれぞれズレていることに起因する。計算には上記のループを採用すればよい。ループ内で計算されない格子点は、境界条件によって、値が更新 (反射の場合は保持) されることになる。