

# 境界条件

## 1. 吸収境界条件

レーザー・プラズマ相互作用のシミュレーションでは、プラズマ表面で反射したレーザー光が再び境界に振れ、シミュレーション領域に戻って来ることがないようにするため、吸収境界条件がよく用いられる。

Tab. 1 吸収境界条件の分類

type	名前	精度	難易度
D	Mur	中	簡単
D	Higdon	良	簡単
M	Berenger	最良	複雑

考案されている吸収境界には、上記 (Tab.1) のようなものがある。ここで、type D とは境界上で反射波が存在しないという近似的な微分方程式から導出されたものである。D は Differential-based absorbing boundary condition の略である。次に type M は Material-based absorbing boundary condition の略であり、境界に存在しない仮想的な媒質を置き、反射波を減衰させるという発想から導かれたものである。ここでは、簡単かつ最もポピュラーな Mur 1 次吸収境界条件について述べていく。

## 2. Mur1 次

$x = 0$  の境界に  $x$  の正方向から  $E_z$  成分をもつ平面波が垂直に入射した場合、伝搬速度を  $v$  とすると平面波は、

$$E_z = E_z(x + vt) \quad (1)$$

と書け、次の伝搬方程式を満たす。ここで、 $y, z$  座標は関係しないため省略する。

$$\frac{\partial E_z}{\partial x} - \frac{1}{v} \frac{\partial E_z}{\partial t} = 0 \quad (2)$$

$x = 0$  の境界で完全に反射がない場合は、境界上でも (2) 式を保ったまま伝搬するはずである。これを FDTD 法の形式に定式化するために、時間差分をとると、

$$\frac{\partial E_z}{\partial t} = \frac{E_z^n - E_z^{n-1}}{\Delta t} = v \frac{\partial E_z^{n-1/2}}{\partial x} \quad (3)$$

右辺の  $x$  についての微分を  $x = \Delta x/2$  で行くと、

$$\frac{E_z^n(1/2) - E_z^{n-1}(1/2)}{\Delta t} = v \frac{E_z^{n-1/2}(1) - E_z^{n-1/2}}{\Delta t} \quad (4)$$

となる。左辺の  $E_z^n(1/2)$  や右辺の  $E_z^{n-1/2}$  は FDTD 法では存在しないので、前後の平均値

$$E_z^n(1/2) = \frac{E_z^n(0) + E_z^n(1)}{2} \quad (5)$$

$$E_z^{n-1/2}(1) = \frac{E_z^n(1) + E_z^{n-1}(1)}{2} \quad (6)$$

$$E_z^{n-1/2}(0) = \frac{E_z^n(0) + E_z^{n-1}(0)}{2} \quad (7)$$

を採用し (4) 式に代入し、 $E_z^n(0)$  について解くと

$$E_z^n(0, j) = E_z^{n-1}(1, j) + X \left\{ E_z^n(1, j) - E_z^{n-1}(0, j) \right\} \quad (8)$$

$$X = \frac{v\Delta t - \Delta x}{v\Delta t + \Delta x} \quad (9)$$

となる。これを  $x = 0$ 、 $x = L_x$  にそれぞれ境界条件を適用することで電磁波を吸収できる。例えば、シミュレーションで直線偏向レーザー (伝搬方向が  $x$ 、電場成分  $E_y$ 、磁場成分  $B_z$ ) を照射し、 $x$  方向に Mur1 次吸収境界条件を適用する場合、2 次元なら、境界  $x$  に垂直な電磁場成分  $E_y, B_z$  に境界条件を適用する。 $E_x$  に適用する必要はない。また、Mur1 次吸収境界条件では垂直入射する電磁場しか吸収されないが、2 次の Mur 吸収境界条件では、斜め入射の電磁場も吸収することができる。

## 3. 粒子境界条件

PIC-Simulation では、粒子と場の境界条件は別々に与えられる。レーザー・プラズマ相互作用のシミュレーションでは、粒子は反射境界、場は吸収境界条件を適用することがしばしば行われる。場に吸収境界が適用されているのに対し、粒子に反射境界が適用される理由は、シミュレーション領域内の電荷中性を保つためである。粒子に対し、吸収境界を適用するとまず先に電子が境界に触れ、領域の電荷中性が破られることになる。また、レーザーの非線形効果によって、特に電子が強い加速と加熱を受けプラズマ壁から飛び出す、生成されるシース電場によってほとんどの

電子が再びプラズマ中に引き戻されるため、ほとんどの場合において粒子の反射境界条件の適用は物理的に妥当であるといえる。

#### 4. 3次元境界条件の適用

2次元から3次元の拡張で最も難解であるのが、境界条件の適用である。ここでは、 $x$ 方向に吸収境界条件、 $yz$ 方向に周期境界条件を与える場合について述べていく。まず、 $yz$ 方向の周期境界条件の場合を見てみる。

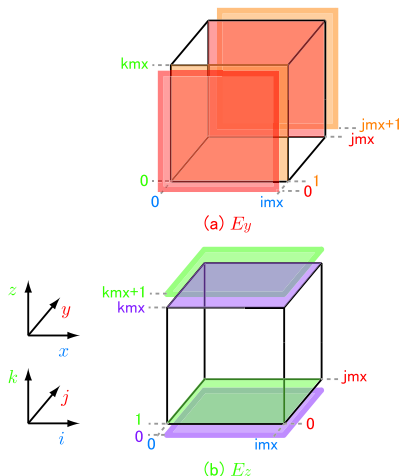


Fig. 1 電場の周期境界条件の適用範囲

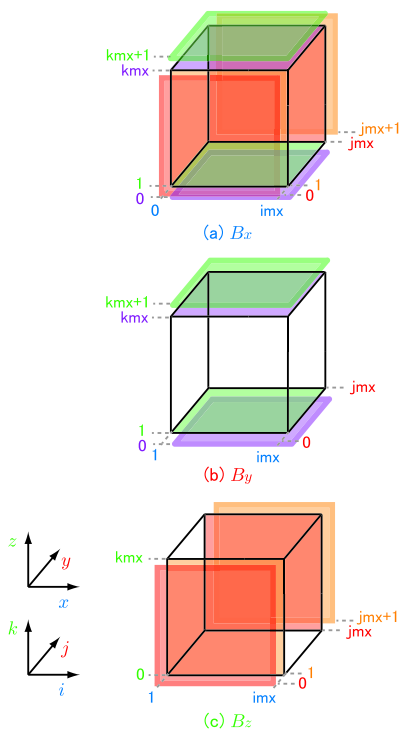


Fig. 2 磁場の周期境界条件の適用範囲

Fig.1とFig.2に概念図を示す。シミュレーション領域(黒枠)の外側に位置する境界面( $y$ 方向なら0と $jmx+1$ 、 $z$ 方向なら0と $kmx+1$ )は、電磁場の数値計

算によって値が更新されない格子点である。これらの格子点を対になる(例えば、 $y$ 方向の格子点なら、0には $jmx$ が $jmx+1$ には1がそれぞれ対応する)シミュレーション領域(黒枠)上の格子点を用いて更新するのが、周期境界条件である。電場 $E_x$ は $yz$ 方向にズレていないため、特に処理をする必要はない。

次に $x$ 方向の吸収境界条件を見てみる。Fig.3とFig.4に概念図を示す。Mur1次吸収境界条件の場合は、境界面での格子点と隣り合うシミュレーション領域(黒枠)上の格子点が必要となる。例えば、左側の吸収境界の場合には、 $x$ 方向の0と1の格子点が必要となる。 $x$ 方向の吸収境界条件には、 $E_x$ や $B_x$ の境界に平行な成分は処理をする必要はない。境界に垂直な $yz$ 成分に(8)式を適用する。

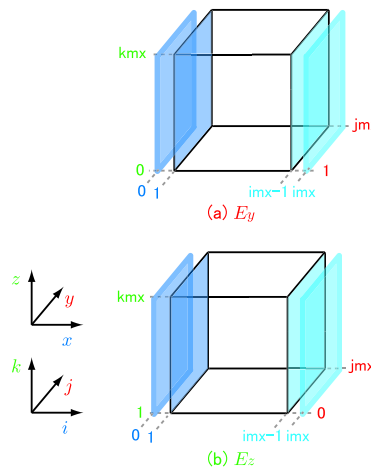


Fig. 3 電場の周期吸収条件の適用範囲

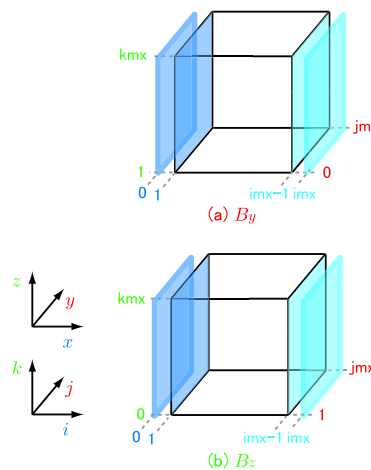


Fig. 4 磁場の周期吸収条件の適用範囲

※後述するプログラム中において、表記されている $dt\_dx(ddt)$ は(9)式にあたる関数である。 $ddt$ は計算の際に用いた時間刻み幅である。これらの関数は自身で定義する必要がある。また、 $tmp$ は前時間の値が記憶されている。 $imm$ は $imx-1$ を、 $imp$ は $imx+1$ をそれぞれ意味している。 $jmm, kmm, jmp, kmp$ も同様である。

```

/* Ey & Ez cycle boundary conditions */
for(i=0;i<=imx;i++)
{
  for(j=0;j<=jmx;j++)
  {
    Ez[i][j][ 0] = Ez[i][j][kmx];
    Ez[i][j][kmp] = Ez[i][j][ 1];
  }
  for(k=0;k<=kmx;k++)
  {
    Ey[i][ 0][k] = Ey[i][jmx][k];
    Ey[i][jmp][k] = Ey[i][ 1][k];
  }
}

/* Bx cycle boundary conditions */
for(i=0;i<=imx;i++)
{
  for(j=1;j<=jmx;j++)
  {
    Bx[i][j][kmp] = Bx[i][j][ 1];
    Bx[i][j][ 0] = Bx[i][j][kmx];
  }
  for(k=1;k<=kmx;k++)
  {
    Bx[i][ 0][k] = Bx[i][jmx][k];
    Bx[i][jmp][k] = Bx[i][ 1][k];
  }
}

/* By cycle boundary conditions */
for(i=1;i<=imx;i++)
{
  for(j=0;j<=jmx;j++)
  {
    By[i][j][kmp] = By[i][j][ 1];
    By[i][j][ 0] = By[i][j][kmx];
  }
}

/* Bz cycle boundary conditions */
for(i=1;i<=imx;i++)
{
  for(k=0;k<=kmx;k++)
  {
    Bz[i][jmp][k] = Bz[i][ 1][k];
    Bz[i][ 0][k] = Bz[i][jmx][k];
  }
}

```

```

/* Ey absorption boundary conditions */
for(k=0;k<=kmx;k++)
for(j=1;j<=jmx;j++)
{
  Ey[ 0][j][k] = tmp[ 1][j][k]+dt_dx(ddt)
    *(Ey[ 1][j][k]-tmp[ 0][j][k]);
  Ey[imx][j][k] = tmp[imm][j][k]+dt_dx(ddt)
    *(Ey[imm][j][k]-tmp[imx][j][k]);
}

/* Ez absorption boundary conditions */
for(k=1;k<=kmx;k++)
for(j=0;j<=jmx;j++)
{
  Ez[ 0][j][k] = tmp[ 1][j][k]+dt_dx(ddt)
    *(Ez[ 1][j][k]-tmp[ 0][j][k]);
  Ez[imx][j][k]=tmp[imm][j][k]+dt_dx(ddt)
    *(Ez[imm][j][k]-tmp[imx][j][k]);
}

/* By absorption boundary conditions */
for(k=1;k<=kmx;k++)
for(j=0;j<=jmx;j++)
{
  By[ 0][j][k] = tmp[ 1][j][k]+dt_dx(ddt)
    *(By[ 1][j][k]-tmp[ 0][j][k]);
  By[imx][j][k] = tmp[imm][j][k]+dt_dx(ddt)
    *(By[imm][j][k]-tmp[imx][j][k]);
}

/* Bz absorption boundary conditions */
for(k=0;k<=kmx;k++)
for(j=1;j<=jmx;j++)
{
  Bz[ 0][j][k] = tmp[ 1][j][k]+dt_dx(ddt)
    *(Bz[ 1][j][k]-tmp[ 0][j][k]);
  Bz[imx][j][k] = tmp[imm][j][k]+dt_dx(ddt)
    *(Bz[imm][j][k]-tmp[imx][j][k]);
}

```