EEM-FDMのご紹介

～GPU高速計算を中心に～

株式会社EEM

2017/03/14

EEM-FDM 電磁界シミュレータ

〇差分法を用いた電磁界シミュレータ
・時間領域差分法(FDTD法)： 高周波用 (計算対象 > 波長)
・周波数領域差分法 ： 低周波用 (計算対象 < 波長)

〇用途
・準静電界(10^3Hz)から光(10^{15}Hz)まであらゆる電磁界現象に対応
・アンテナ
・EMC
・導波管
・マイクロストリップ線路、マイクロ波回路
・電波伝播
・電波吸収体、電波暗室
・メタマテリアル

〇高速化技術
・FDTD法は並列化に適しているのでCPU/GPU(クラスタ含む)により大規模計算が可能

〇動作環境
・WindowsPCからスパコンまで
・NVIDIAのグラフィックスボードを用いた超高速計算

FDTD法(Finite Difference Time Domain、時間領域差分法)

Maxwellの方程式:

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

$$\nabla \times \boldsymbol{H} = \frac{\partial \boldsymbol{D}}{\partial t} + \boldsymbol{J}$$

$$\boldsymbol{D} = \epsilon \boldsymbol{E}$$

$$\boldsymbol{B} = \mu \boldsymbol{H}$$

$$\boldsymbol{J} = \sigma \boldsymbol{E}$$

$\boldsymbol{E}$:電界
$\boldsymbol{D}$:電束密度
$\boldsymbol{H}$:磁界
$\boldsymbol{B}$:磁束密度
$\boldsymbol{J}$:電流
ϵ:誘電率
μ:透磁率
σ:導電率

↓ **D,B,J を消去**

$$\nabla \times \boldsymbol{E} = -\mu \frac{\partial \boldsymbol{H}}{\partial t}$$

$$\nabla \times \boldsymbol{H} = \epsilon \frac{\partial \boldsymbol{E}}{\partial t} + \sigma \boldsymbol{E}$$

E,H:未知数
ε,μ,σ:既知数

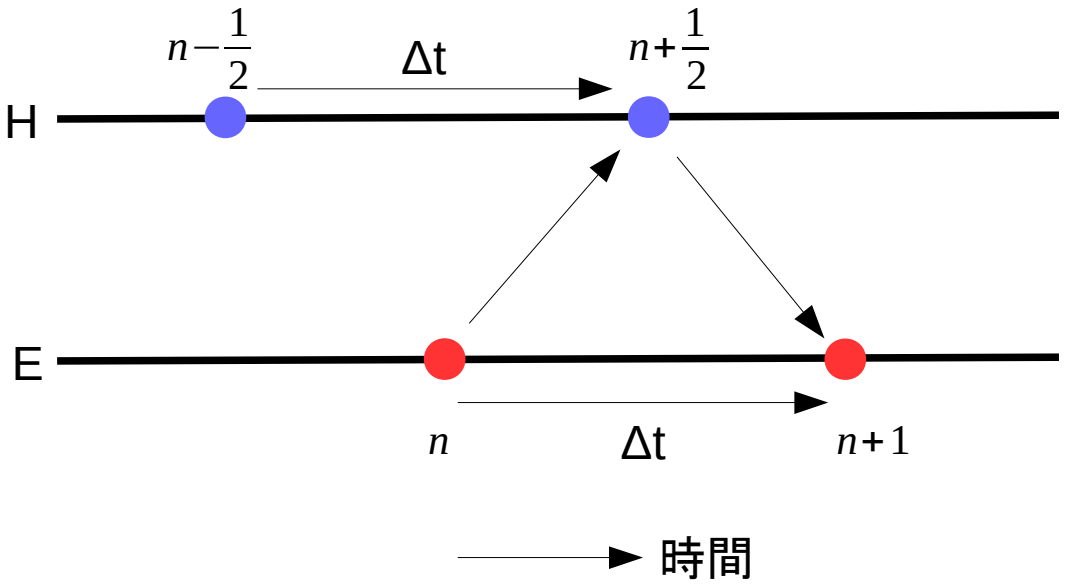
宇野亨「FDTD法による電磁界およびアンテナ解析」コロナ社, 1998

時間領域で離散化 (LeapFrog : 蛙跳びアルゴリズム)

$$Z\boldsymbol{H}^{n+\frac{1}{2}} = d_1 Z\boldsymbol{H}^{n-\frac{1}{2}} - d_2(c\Delta t)\nabla\times\boldsymbol{E}^n$$

$$\boldsymbol{E}^{n+1} = c_1\boldsymbol{E}^n + c_2(c\Delta t)\nabla\times Z\boldsymbol{H}^{n+\frac{1}{2}}$$

c_1, c_2, d_1, d_2 :
電気定数で決まる定数
(場所の関数、無次元)
Z = 120π [Ω]
c : 光速



空間領域で離散化(X成分、左辺が最新項)

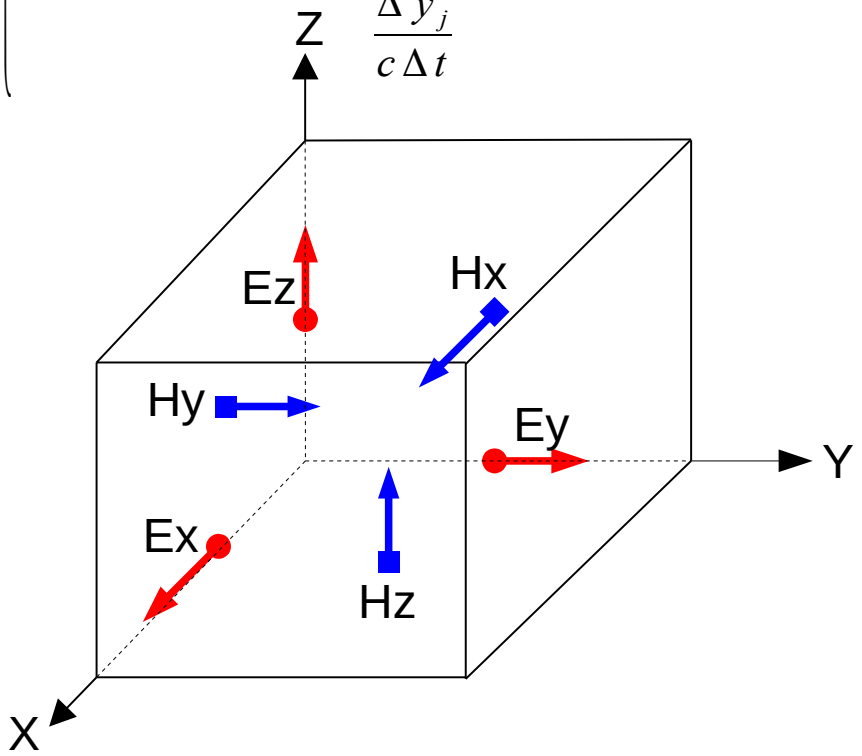
並列化に極めて適している!!

$$ZH_x^{n+\frac{1}{2}}\left(i,j+\frac{1}{2},k+\frac{1}{2}\right) = d_1\left(i,j+\frac{1}{2},k+\frac{1}{2}\right)ZH_x^{n-\frac{1}{2}}\left(i,j+\frac{1}{2},k+\frac{1}{2}\right)$$

$$-d_2\left(i,j+\frac{1}{2},k+\frac{1}{2}\right)\left\{\frac{E_z^n\left(i,j+1,k+\frac{1}{2}\right)-E_z^n\left(i,j,k+\frac{1}{2}\right)}{\frac{\Delta y_{j+\frac{1}{2}}}{c\Delta t}} - \frac{E_y^n\left(i,j+\frac{1}{2},k+1\right)-E_y^n\left(i,j+\frac{1}{2},k\right)}{\frac{\Delta z_{k+\frac{1}{2}}}{c\Delta t}}\right\}$$

$$E_x^{n+1}\left(i+\frac{1}{2},j,k\right) = c_1\left(i+\frac{1}{2},j,k\right)E_x^n\left(i+\frac{1}{2},j,k\right)$$

$$+c_2\left(i+\frac{1}{2},j,k\right)\left\{\frac{ZH_z^{n+\frac{1}{2}}\left(i+\frac{1}{2},j+\frac{1}{2},k\right)-ZH_z^{n+\frac{1}{2}}\left(i+\frac{1}{2},j-\frac{1}{2},k\right)}{\frac{\Delta y_j}{c\Delta t}} - \frac{ZH_y^{n+\frac{1}{2}}\left(i+\frac{1}{2},j,k+\frac{1}{2}\right)-ZH_y^{n+\frac{1}{2}}\left(i+\frac{1}{2},j,k-\frac{1}{2}\right)}{\frac{\Delta z_k}{c\Delta t}}\right\}$$

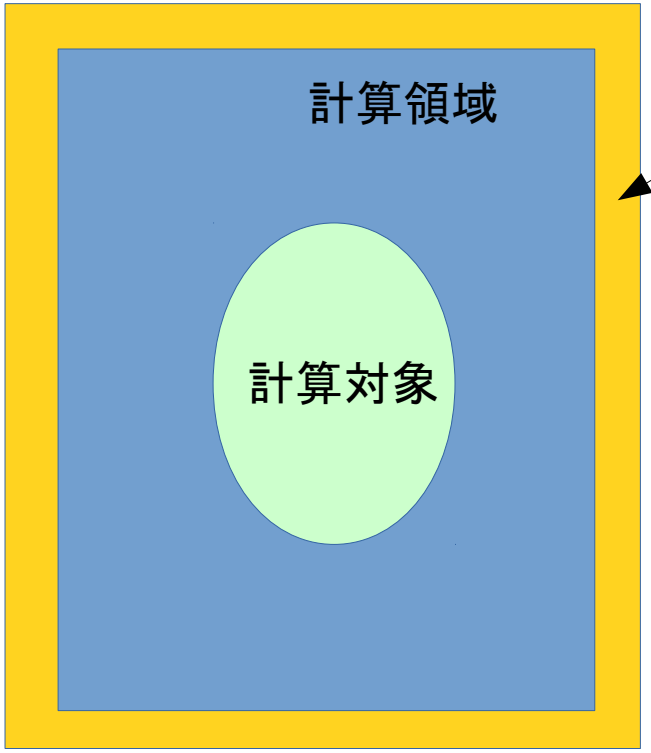


Yee格子

電界E:稜線の中心の接線方向

磁界H:面の中心の法線方向

吸収境界条件



吸収境界条件:
計算領域境界での電磁波の反射をなくす

・Mur一次：1層

・PML：5～10層
（メモリーと計算時間が少し増えるが精度が向上する）

フーリエ変換により時間領域から周波数領域に変換する

$$\boldsymbol{E}(\boldsymbol{r},\omega) = \int_0^\infty \boldsymbol{E}(\boldsymbol{r},t)e^{-j\omega t}dt$$

$$\boldsymbol{H}(\boldsymbol{r},\omega) = \int_0^\infty \boldsymbol{H}(\boldsymbol{r},t)e^{-j\omega t}dt$$

入力インピーダンス

$$Z_{\mathrm{in}}(\omega)=\frac{V_{\mathrm{in}}(\omega)}{I_{\mathrm{in}}(\omega)}$$

(VはEから、IはHから計算する)

遠方界

$$E_{\left\{\begin{smallmatrix}\theta\\ \phi\end{smallmatrix}\right\}}(r,\theta,\phi) = \frac{-jke^{-jkr}}{4\pi r}F_{\left\{\begin{smallmatrix}\theta\\ \phi\end{smallmatrix}\right\}}(\theta,\phi)$$

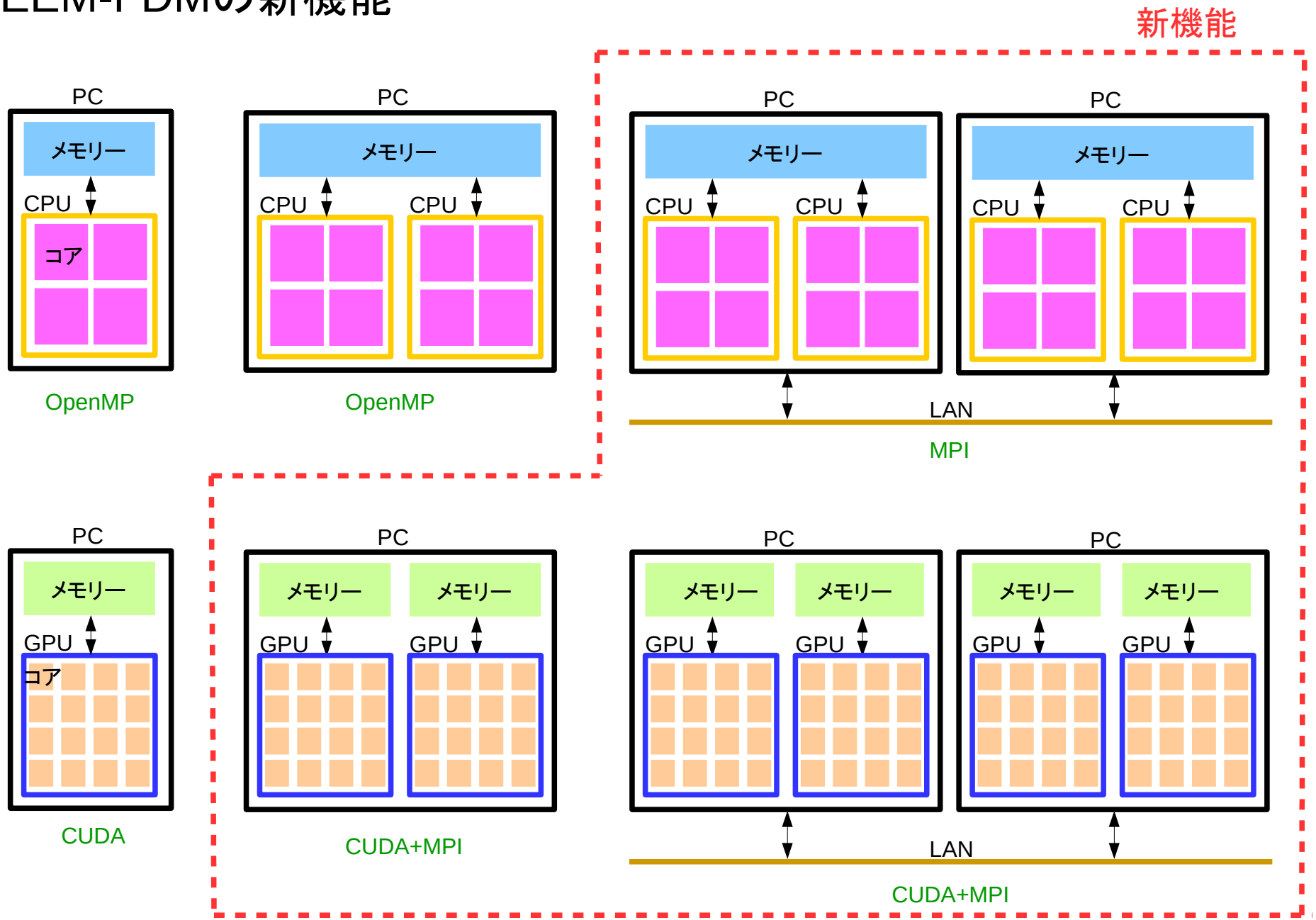
$$F_{\left\{\begin{smallmatrix}\theta\\ \phi\end{smallmatrix}\right\}}(\theta,\phi) = ZN_{\left\{\begin{smallmatrix}\theta\\ \phi\end{smallmatrix}\right\}}(\theta,\phi)\pm L_{\left\{\begin{smallmatrix}\phi\\ \theta\end{smallmatrix}\right\}}(\theta,\phi)$$

$$\boldsymbol{N}(\theta,\phi) = \int_S \hat{n}\times\boldsymbol{H}(\boldsymbol{r})\exp(jk\boldsymbol{r}\cdot\hat{r})dS$$

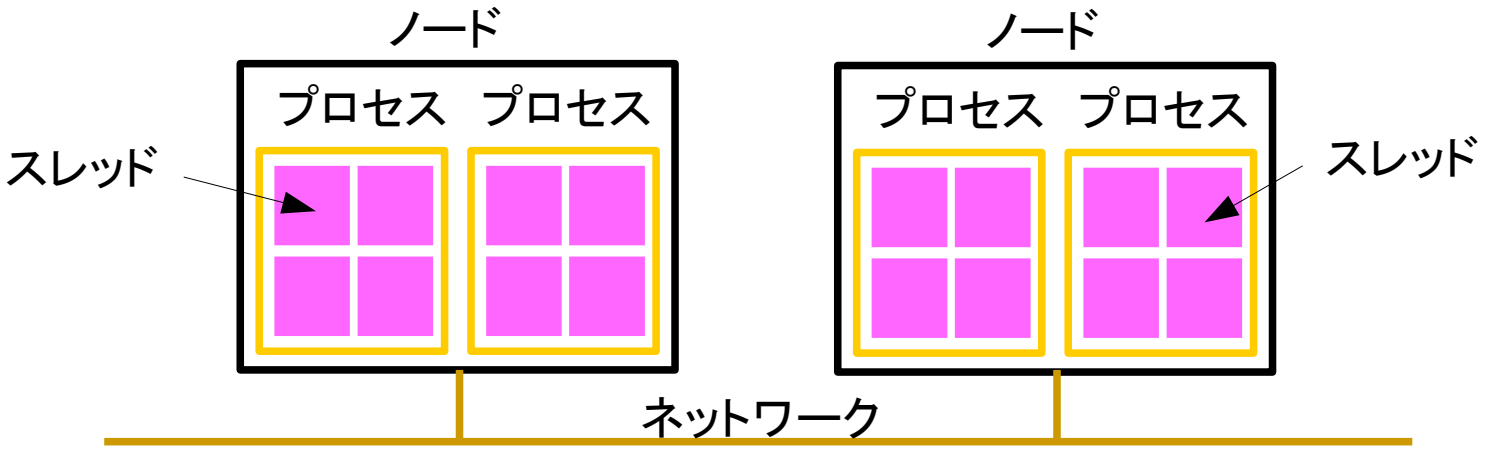
$$\boldsymbol{L}(\theta,\phi) = -\int_S \hat{n}\times\boldsymbol{E}(\boldsymbol{r})\exp(jk\boldsymbol{r}\cdot\hat{r})dS$$

(S:計算領域の境界面)

EEM-FDMの新機能


新機能
PC
メモリー
CPU
コア
OpenMP
PC
メモリー
CPU
CPU
OpenMP
PC
メモリー
CPU
CPU
PC
メモリー
CPU
CPU
LAN
MPI
PC
メモリー
GPU
コア
CUDA
PC
メモリー
メモリー
GPU
GPU
CUDA+MPI
PC
メモリー
メモリー
GPU
GPU
PC
メモリー
メモリー
GPU
GPU
LAN
CUDA+MPI

ノード、プロセス、スレッド



●ノード　：CPUとメモリー(とGPU)を持っている　：PC一台と同じ
●プロセス：独立したメモリー空間を持っている独立したプログラム
●スレッド：プロセスが管理する一連の計算処理　（コアとほぼ同義）

■クラスタは複数のノードから成りネットワークでつながっている
■ノードは複数のプロセスを起動する　(タスクマネージャーに表示される)
■プロセスは複数のスレッドを実行する

逆に
■複数のプロセスにまたがるスレッドはない
■複数のノードにまたがるプロセスはない

◇複数のノードとプロセスが起動できるのはMPIのみ

並列化プログラミング

新機能

■OpenMP
・CPU向け
・共有メモリーモデル
・領域分割が不要
・プログラミングが容易

■MPI
・CPUクラスタ向け
・分散メモリーモデル
・領域分割が必要
・プログラミングが複雑

■CUDA
・GPU向け
・共有メモリーモデル
・領域分割が不要
・プログラミングが複雑

■CUDA+MPI
・マルチGPUとGPUクラスタ向け
・分散メモリーモデル
・領域分割が必要
・CUDAとMPIが適切に実装されていればプログラミングは比較的容易

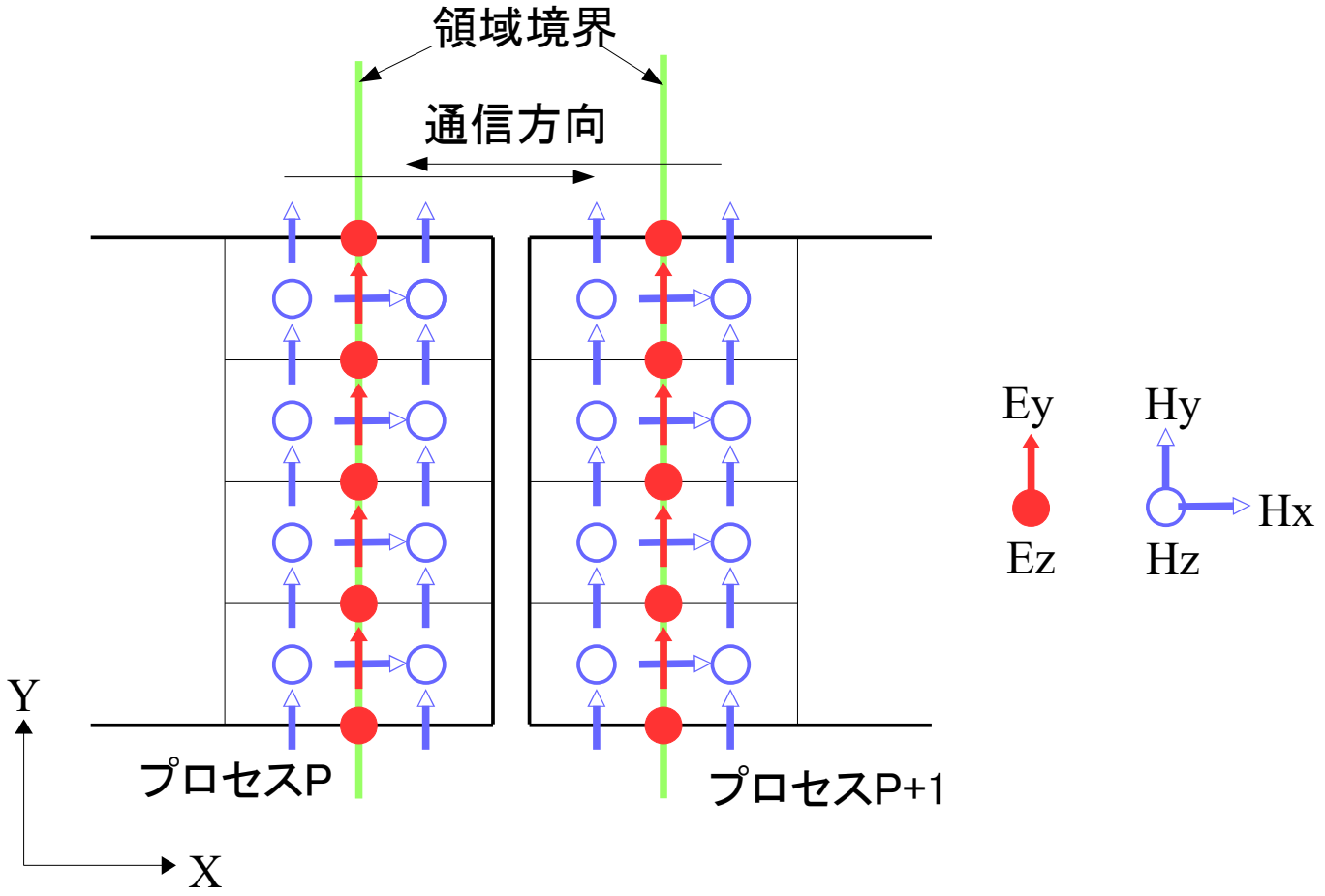
MPI : Message Passing Interface
・分散メモリー向けのプログラミング仕様
・共有メモリー環境でも動作しパフォーマンスも落ちない
→MPIで実装すればすべての環境で最高のパフォーマンスが得られる
（ただしクラスタでは高速なネットワーク環境が必要）

※開発環境はすべて無償

領域分割とMPI

○ FDTD法はアルゴリズム上、領域分割が必要になる

○ MPIの通信：
・タイムステップ毎に不足成分を隣のプロセスから通信によって取得する
・不足成分：領域境界の半セル外側のHy,Hz
・重複成分：領域境界上のEy,Ez,Hx



EEM-FDMの必要リソース

●メモリー

メモリー = (30 / プロセス数 + 24 * 周波数数) * Nx * Ny * Nz [バイト]

例えばプロセス数=1, 周波数数=1のとき、
Nx=Ny=Nz=100のとき54MB
Nx=Ny=Nz=1000のとき54GB

●ファイルサイズ

入力ファイルサイズ　：　一般に小さい

出力ファイルサイズ = (6 + 24 * 周波数数) * Nx * Ny * Nz [バイト]

使用メモリーとほぼ同じ、SSD推奨、十分な空き容量が必要

●計算時間

計算時間 ∝ Nx * Ny * Nz * タイムステップ数　/ プロセス数

グラフィックスボード(ビデオカード、GPU)

NVIDIAのグラフィックスボードがCUDAに対応
現在の世代：GTX 10XX (Pascal世代 = Compute Capability 6.X)
https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units

Model	Launch	Code name	Fab (nm)	Transistors (billion)	Die size (mm^2)	Core config[1]	Bus interface	Clock speeds			Fillrate		Memory				Supported API version				Processing power (GFLOPS)[4]			TDP (watts)	SLI HB support[5]	Release price (USD)	
								Base core clock (MHz)	Boost core clock (MHz)	Memory (MT/s)	Pixel (Boost) (GP/s)[2]	Texture (Boost) (GT/s)[3]	Size (GiB)	Bandwidth (GB/s)	Bus type	Bus width (bit)	Direct3D	OpenGL	OpenCL	Vulkan	Single precision (Boost)	Double precision (Boost)	Half Precision (Boost)[7]			MSRP	Founde Edition
GeForce GTX 1050[55]	October 25, 2016	GP107-300	14	3.3	132	640:40:32	PCIe 3.0 x16	1354	1455	7000	43.3 (46.6)	54.2 (58.2)	2	112	GDDR5	128	12.0	4.5	1.2	1.0[12]	1733 (1862)	54 (72)	27 (36)	75	No	$109	
GeForce GTX 1050 Ti[55]	October 25, 2016	GP107-400				768:48:32		1290	1392		41.3 (44.5)	61.9 (66.8)	4								1981 (2138)	62 (91)	31 (45.6)			$139	N/A
GeForce GTX 1060[56]	August 18, 2016	GP106-300	16	4.4	200	1152:72:48		1506	1708	8000 (9000)	72.3 (82)	108.4 (123)	3	192 (216)		192					3470 (3935)	108 (144)	54 (72)	120		$199	
	July 19, 2016	GP106-400				1280:80:48						120.5 (160)	6								3855 (4372)	120 (160)	60 (80)	120		$249	$299
GeForce GTX 1070[57]	June 10, 2016	GP104-200		7.2	314	1920:120:64			1683	8000	96.4 (107.7)	180.7 (202)	8	256		256					5783 (6463)	181 (228)	90 (114)	150	2-way SLI HB[58] or traditional 2/3/4-way SLI[59]	$379	$449 ($399)
GeForce GTX 1080[60]	May 27, 2016	GP104-400				2560:160:64		1607	1733	10000 (11000)	102.8 (110.9)	257.1 (277.3)		320 (352)							8228 (8873)	257 (288)	128 (144)	180		$599 ($499)	$699 ($549)
Geforce GTX 1080 Ti[61]	March 5, 2017	GP102		12	471	3584:224:88		1480[62]	1582	11000	130.2 (139.2)	331.5 (354.4)	11	484	GDDR5X	352					10608 (11339)	360 (388)	180 (194)	250		$699	$699
Nvidia Titan X[63]	August 2, 2016	GP102-400				3584:224:96		1417	1531	10000	136 (147)	317.4 (343)	12	480		384					10157 (10974)	317 (343)	159 (171)	250		N/A	$1200

FOCUSスパコン（公益財団法人計算科学振興財団)

FOCUSスパコンの構成

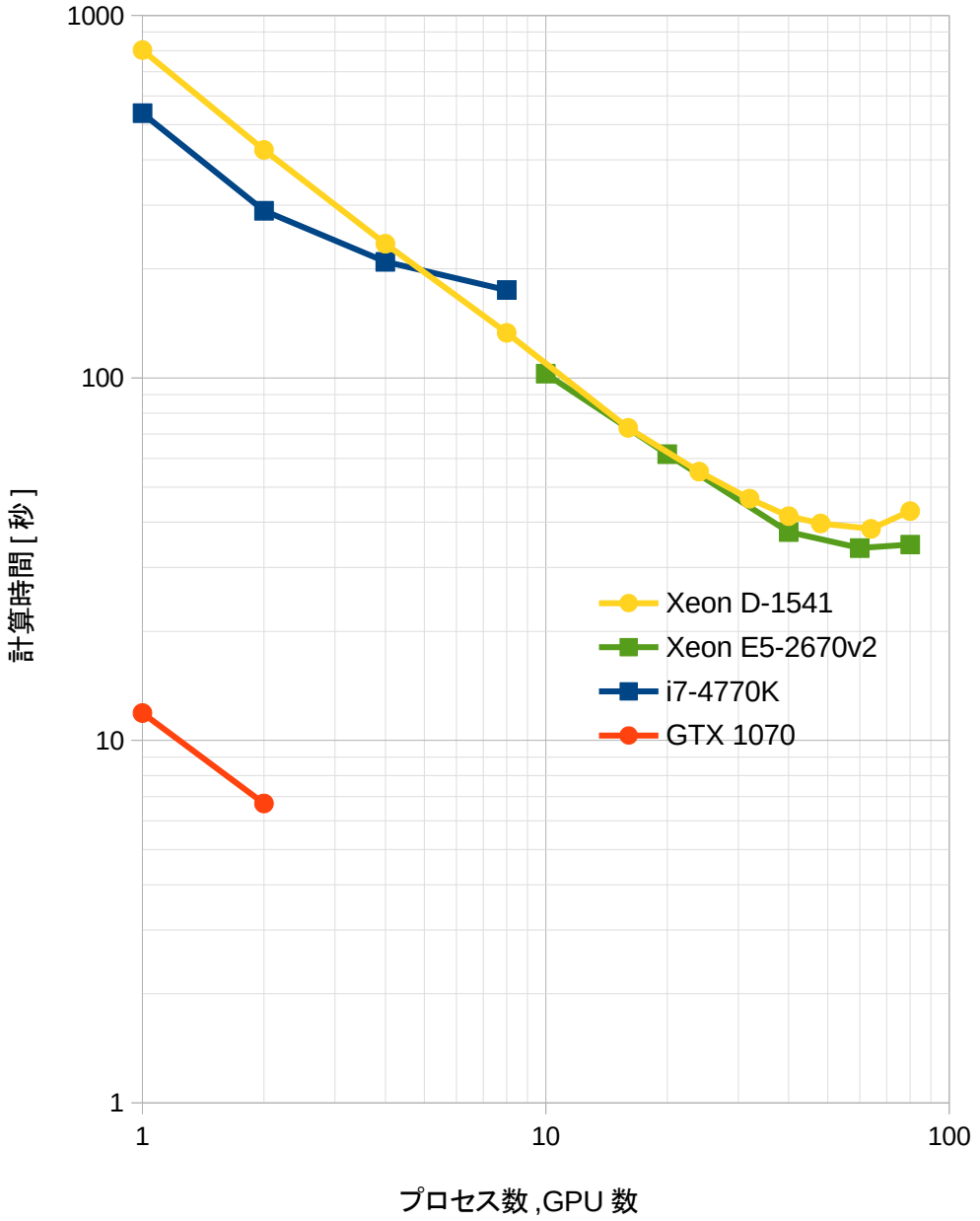
システム	CPU	コア数 (ノード当たり)	メモリー (ノード当たり)	ノード数	ノード間 ネットワーク	使用料 (1ノード1時間)
B	Xeon E7520	16	512GB	2	InfiniBand 40Gbps	100円
D	Xeon E5-2670v2	20	64GB	80	InfiniBand 56Gbps	300円
E	Xeon E5-2670v2	20	128GB	48	InfiniBand 56Gbps	300円
F	Xeon E5-2698v4	40	128GB	12	InfiniBand 56Gbps	500円
H	Xeon D-1541	8	64GB	136	Ethernet 10Gbps x 2	100円

※ アカウント料年間1万円が必要、使用料はジョブ従量制

※ 類似のサービス：Amazon Web Services (EC2) : https://aws.amazon.com/jp/

計算時間の一例

セル数=300 X 300 X 300、タイムステップ数=500



クラスタは何台(何プロセス)まで速くなるか

通信時間が計算時間と同じオーダーになると速度比が低下し始める

1タイムステップ当たりの

計算時間
= (Nx * Ny * Nz) / (300 * 300 * 300) * (500 / 500) / Np　　(i7-4770Kの場合)
= 37 * 10^{-9} * (Nx * Ny * Nz / Np)　　[秒]　　(Np:プロセス数)

通信時間
= 遅延時間 + 通信量 / 通信速度
= (2 * Ny * Nz * 4 * M)　/ (bps / 8)　　[秒]　　(遅延時間～10μsは無視できる)

通信時間 << 計算時間 となるためには

Np << Nx / 7 * 通信速度[Gbps]　　(M:両側双方向通信のためM=4としている)

結論：クラスタの有効台数はNxと通信速度に比例する

例えば 1Gbps, Nx=300 のとき Np << 40プロセス　まで (～Beowulfクラス)

※ Nxに比例する理由はX方向に領域分割しているため
→計算領域が細長いときはX方向を長手にすると有利
※ 上の評価式が成り立つ条件： Nx >> Np
・領域境界での重複計算
・吸収境界条件

クラスタの使い方

●準備作業
1. 計算に使用するすべてのコンピュータをネットワークで接続します。
(高速なネットワーク推奨)
2. すべてのコンピュータに同じアカウントとパスワードを設定します。
(パスワードなしは不可)
3. すべてのコンピュータの同じ絶対パスに4つのファイル
(fdm2_mpi.exe, fdm2g_mpi.exe, msmpi.dll, smpd.exe) をコピーします。

●計算
1. [ホスト名]と[プロセス数]を適当に設定します。
CPUクラスタでは[プロセス数]に物理コア数と論理コア数の間の数値を入力してください。
GPUクラスタでは[プロセス数]にグラフィックスボードの数を入力してください。
2. すべてのコンピュータでコマンドプロンプトで “smpd -p 8677” を実行します。
(8677はMPIのポート番号(固定))
3. [計算]ボタンをクリックすると計算を開始します。

データ作成ライブラリー

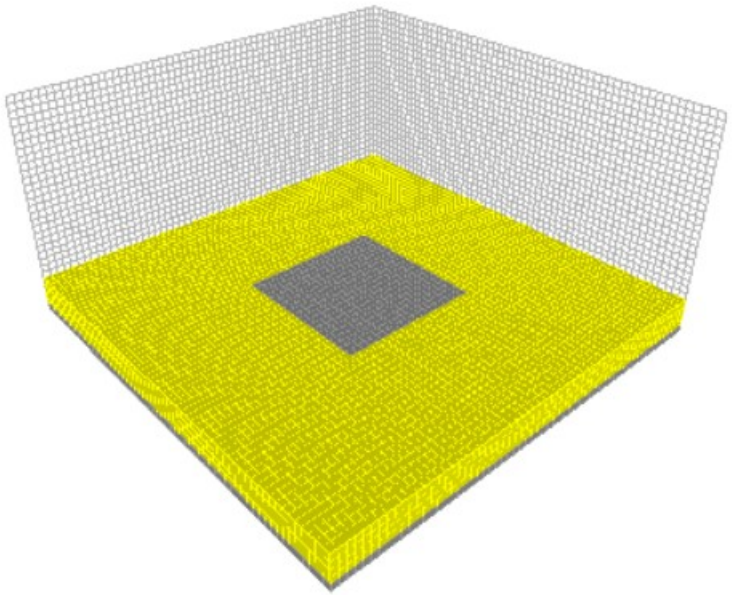
以下のようなケースではデータ作成ライブラリーが便利
(1)データの数が多く、かつ規則的なとき
(2)一つの座標を変えると同時にたくさんの変更が必要になるとき
(3)パラメーターを変えながら繰り返し計算したいとき

右に使用例を示す(ダイポールアンテナ)。
用意された関数を呼び出してデータを作成する。C言語の初歩的な知識が必要。

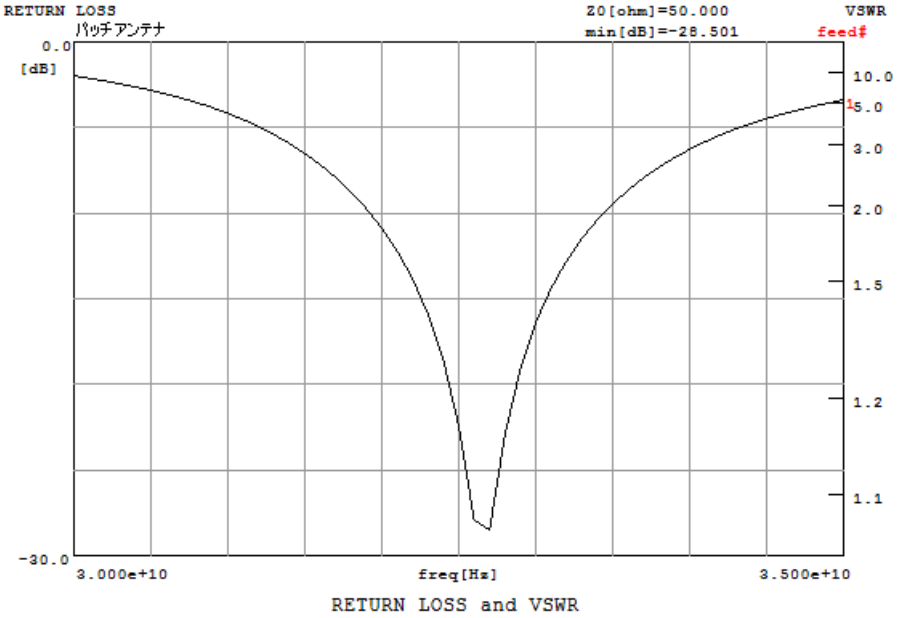
```
#include "fdm_datalib.h"

int main()
{
	// (1) initialize

	fdm_init();

	// (2) title

	fdm_title("sample1");

	// (3) domain

	fdm_domain(0);

	// (4) mesh

	fdm_xsection(2, -50e-3, +50e-3);
	fdm_xdivision(1, 20);

	fdm_ysection(2, -50e-3, +50e-3);
	fdm_ydivision(1, 20);

	fdm_zsection(4, -75e-3, -25e-3, +25e-3, +75e-3);
	fdm_zdivision(3, 10, 11, 10);

	// (5) material

	fdm_material(2.0, 0.0, 1.0, 0.0);

	// (6) geometry

	fdm_unit6(1, 1, 0e-3, 0e-3, -25e-3, 0e-3, 0e-3, 25e-3);  // dipole

	// (7) incidence

	fdm_feed(3, 0e-3, 0e-3, 0e-3, 1, 0);

	// (8) observation point

	// (9) frequency

	fdm_freq(3e9, 3e9, 0);
	fdm_freq2(2e9, 4e9, 20);

	// (10) solver

	fdm_iteration(1e-3, 2000, 100);

	// (11) output

	fdm_outdat("sample1.fdm");

	return 0;
}
```

計算例

パッチアンテナ(1/2)



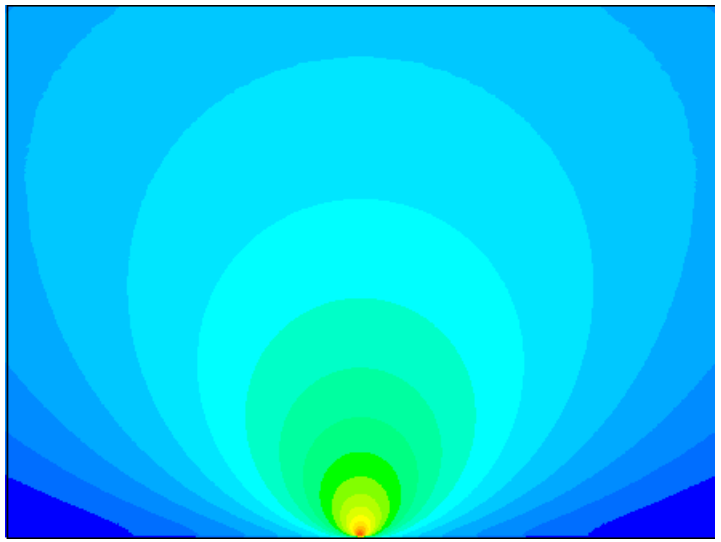
パッチアンテナ計算モデル(ミリ波用)
パッチ=2mm□、基板厚さ=0.4mm、基板比誘電率=4



反射損失周波数特性
(30～35GHz, Z0=50Ω)

パッチアンテナ(2/2)

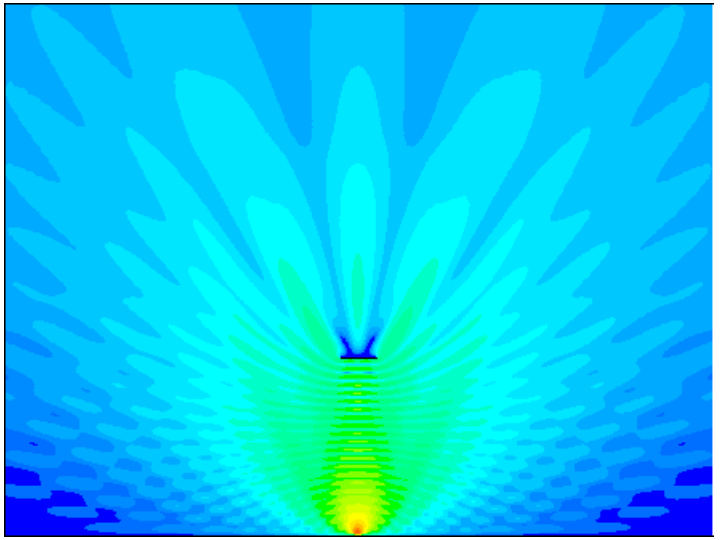
アンテナ単体



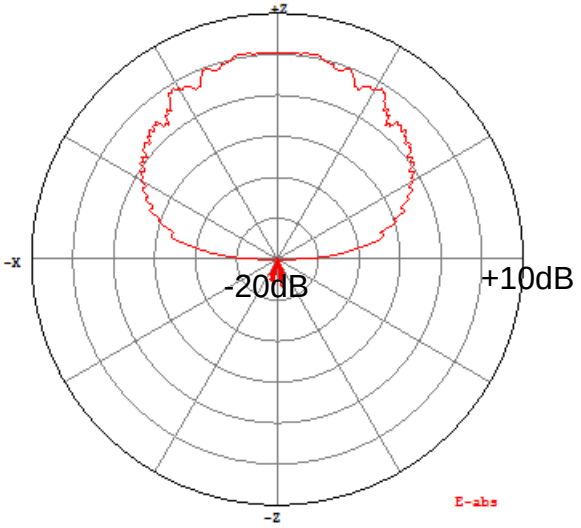
300mm

400mm

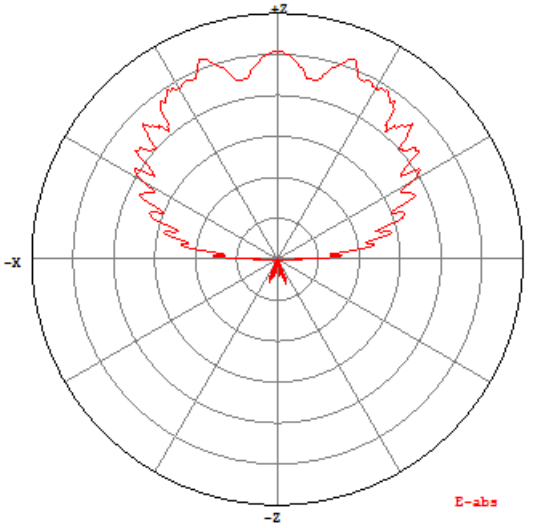
金属板の障害物



電界分布(33GHz)



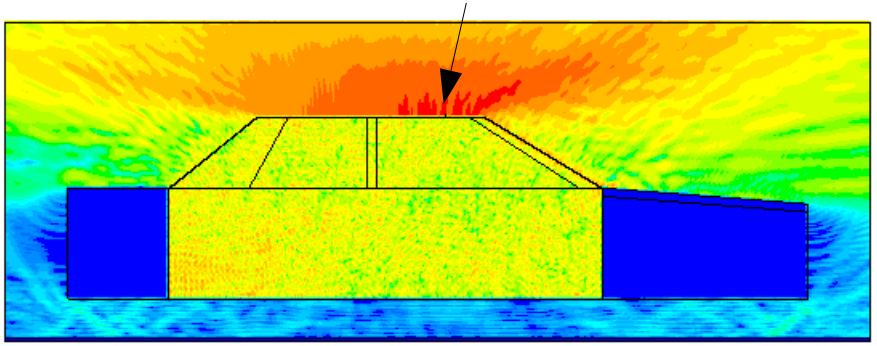
放射パターン(33GHz)



セル数 = 660 x 660 x 455 = 2.0億、タイムステップ数=7500、計算時間=14分(@2GPU)、使用メモリー=6.2GB x 2 GPU : NVIDIA GeForce GTX 1070 (8GB GDDR5) X 2 (以下同じ)

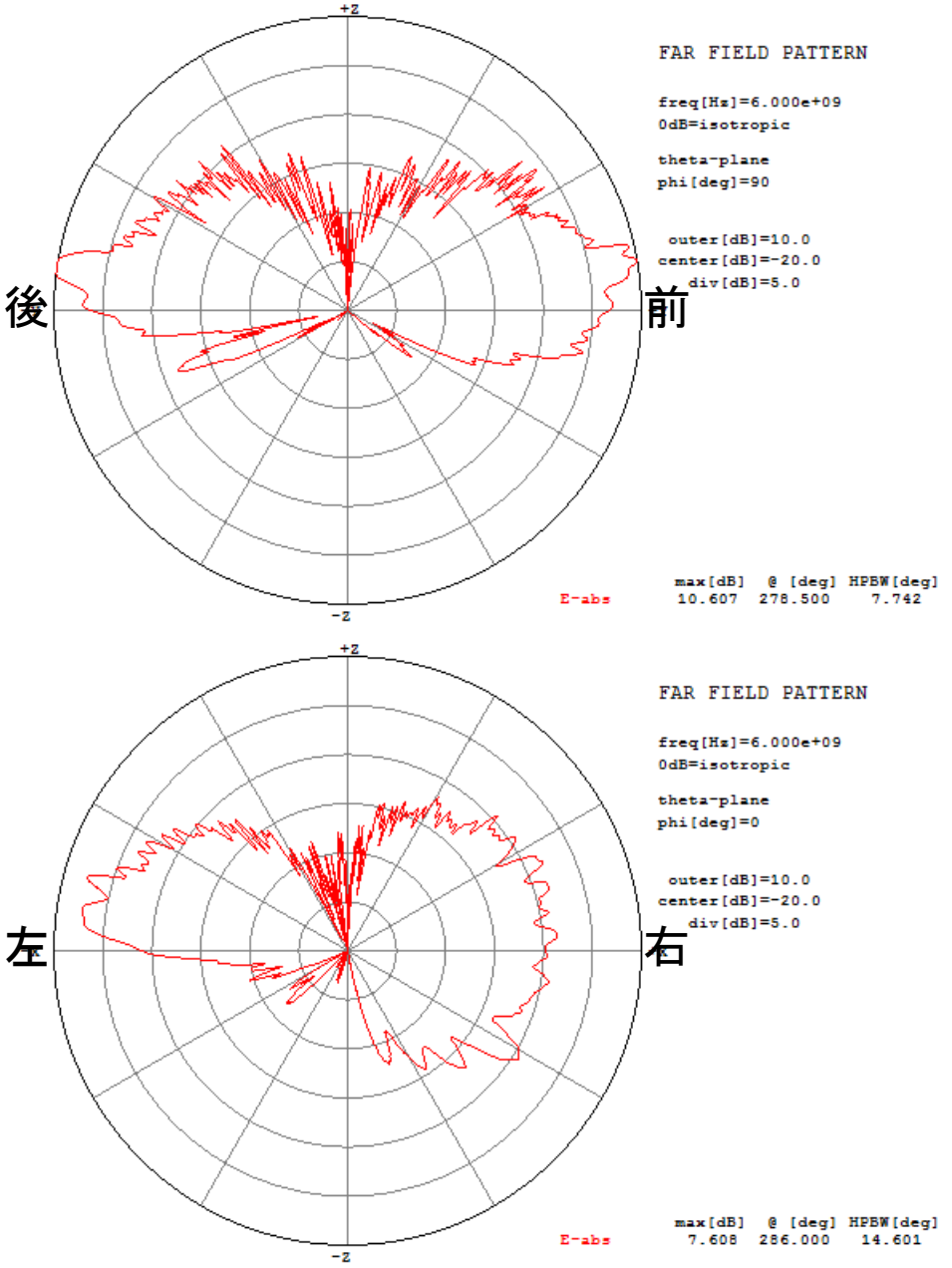
車載アンテナ

アンテナ(垂直モノポール、右端)



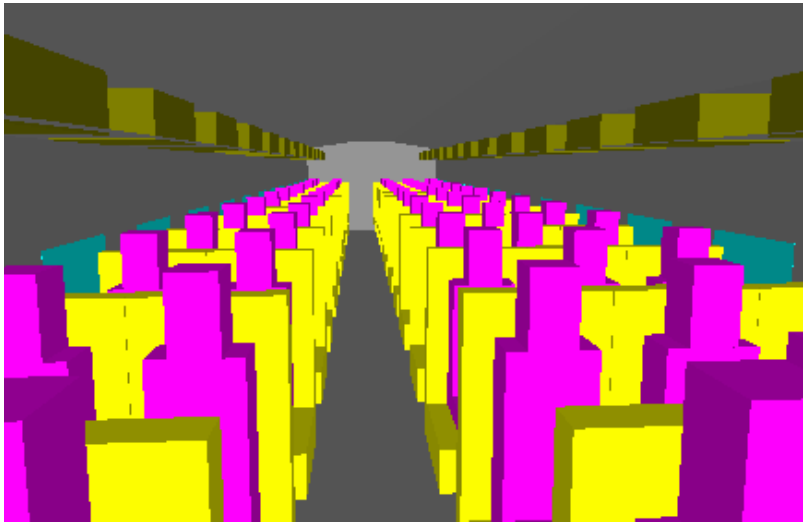
電界分布(6GHz、中心面)

セル数 = 440 x 1100 x 404 = 2.0億
タイムステップ数 = 2300
計算時間 = 5分 (@2GPU)
使用メモリー = 6.2GB x 2

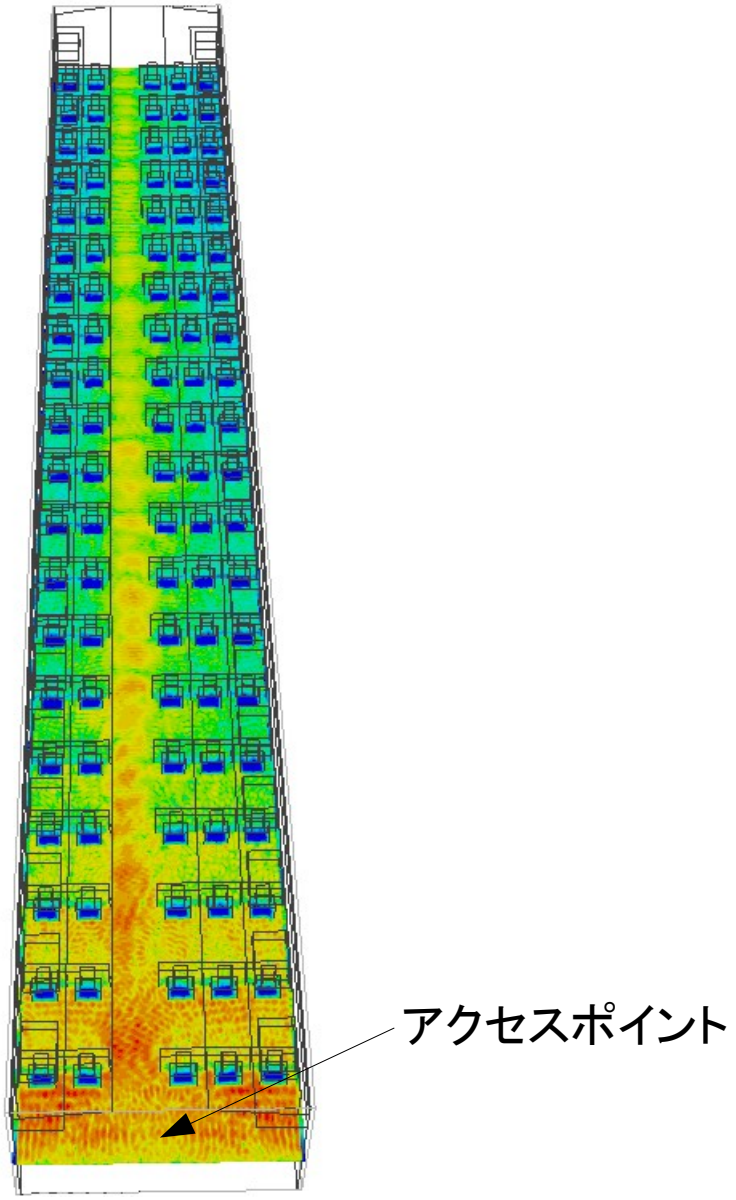


放射パターン(6GHz)

新幹線

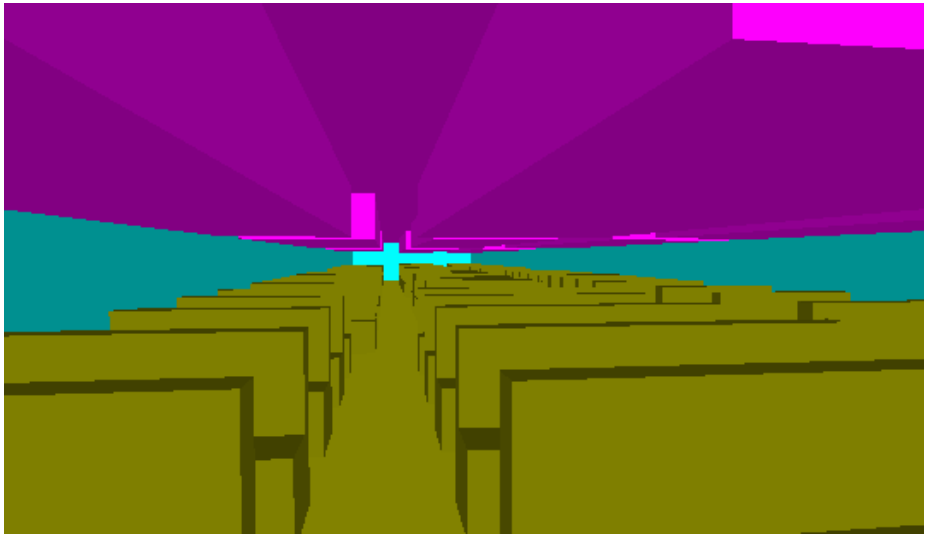


セル数 = 2141 x 322 x 217 = 1.5億
タイムステップ数 = 6700
計算時間 = 10分 (@2GPU)
使用メモリー = 4.0GB x 2

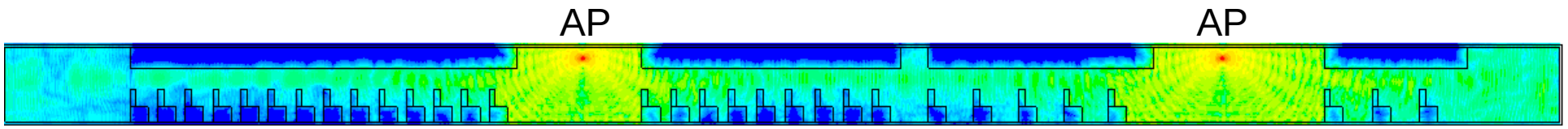


電界分布(2.45GHz, 床上80cm)

航空機

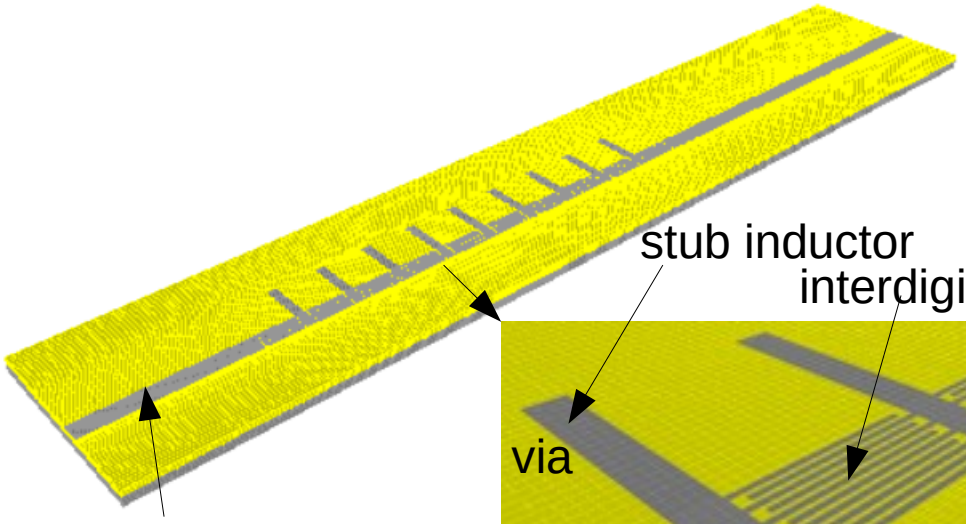


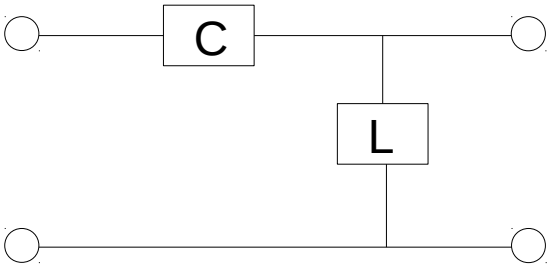
セル数 = 400 x 2560 x 135 = 1.4億
タイムステップ数 = 5000
計算時間 = 9分 (@2GPU)
使用メモリー = 3.8GB x 2



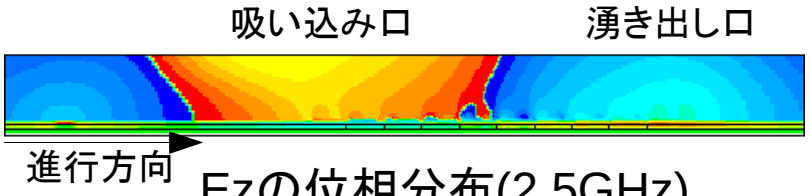
電界分布(1.5GHz)

メタマテリアル

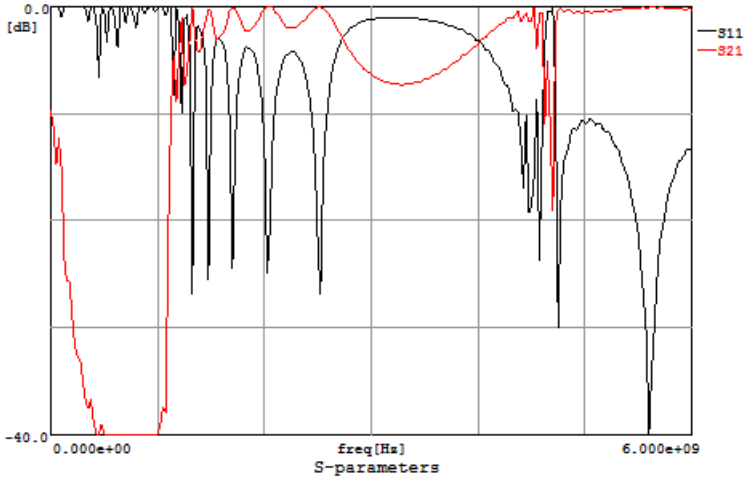




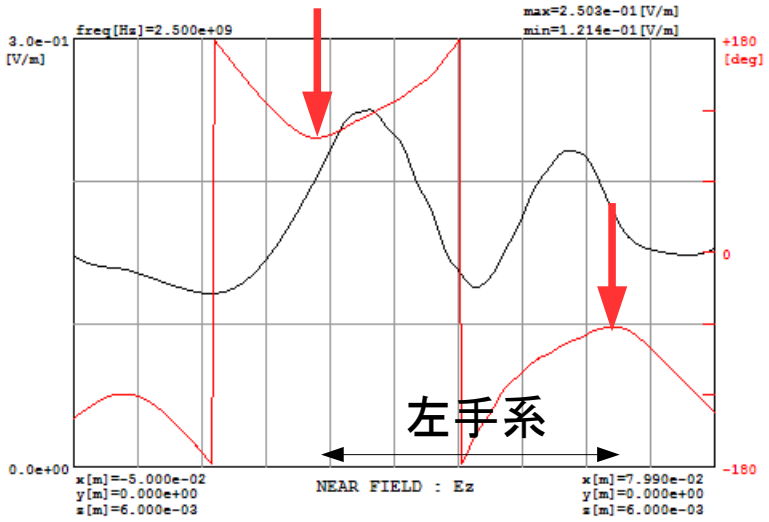
メタマテリアルの等価回路



Ezの位相分布(2.5GHz)



S11,S21の周波数特性(0～6GHz)

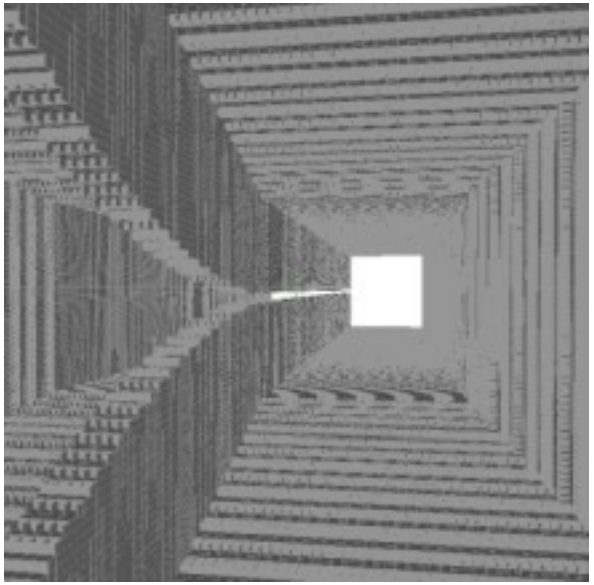


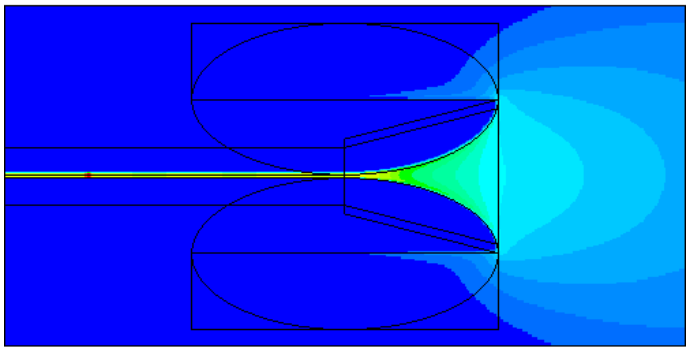
Ezの位相分布(2.5GHz,Z=6mm)

BBE(Backfire-Broadside-Endfire)アンテナ:
周波数でビームの向きが変わるアンテナ

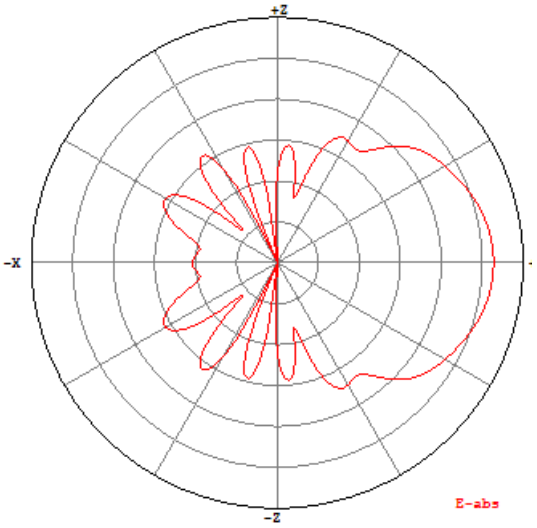
C.Caloz and T.Itoh, Electromagnetic Metamaterials: Transmission line theory and Microwave Applications, Wiley Interscience, 2006

ダブルリッジホーンアンテナ

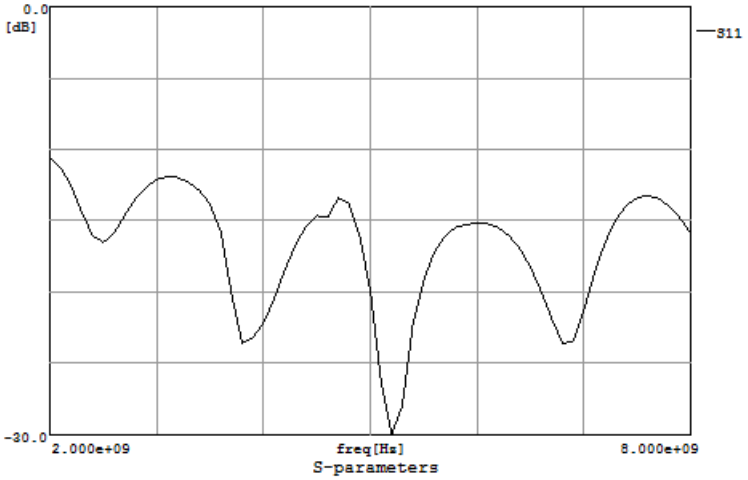




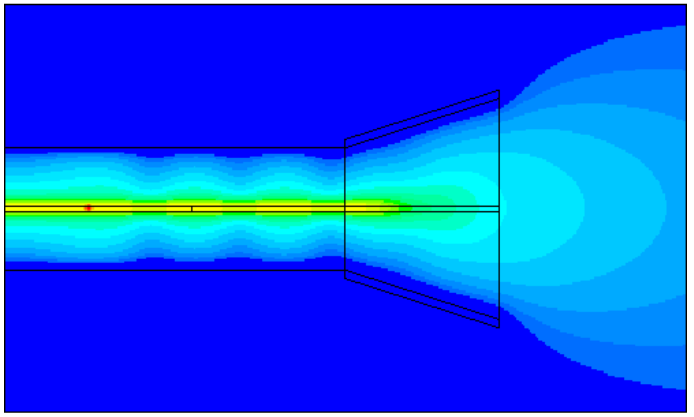
E面電界分布(3GHz)



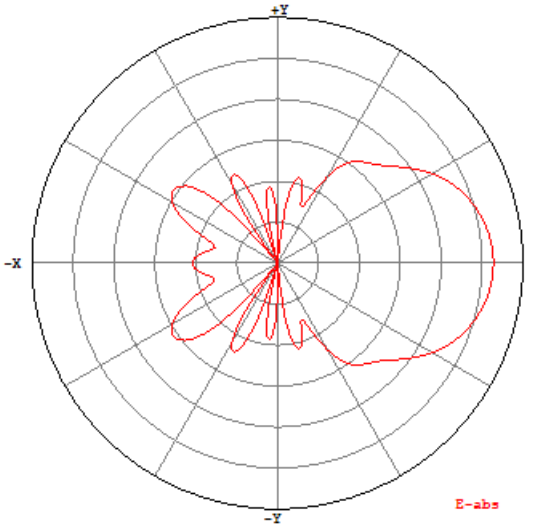
E面放射パターン(3GHz)



S11周波数特性(2～8GHz)

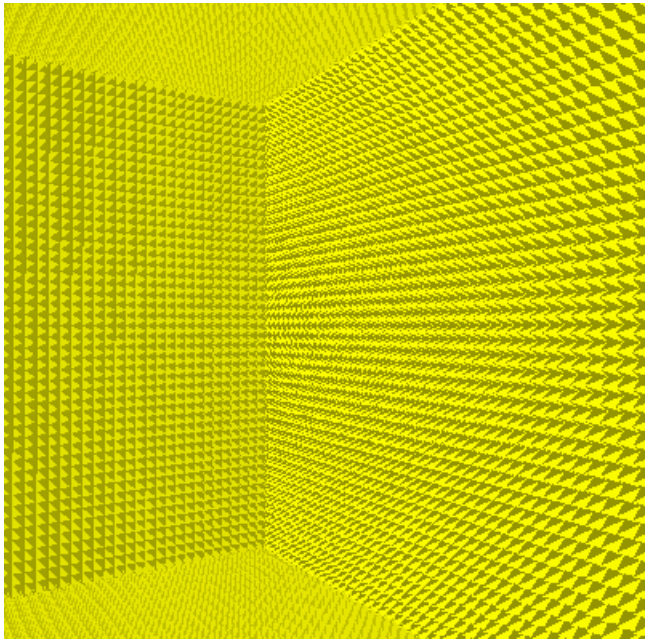


H面電界分布(3GHz)



H面放射パターン(3GHz)

電波暗室(1/2)



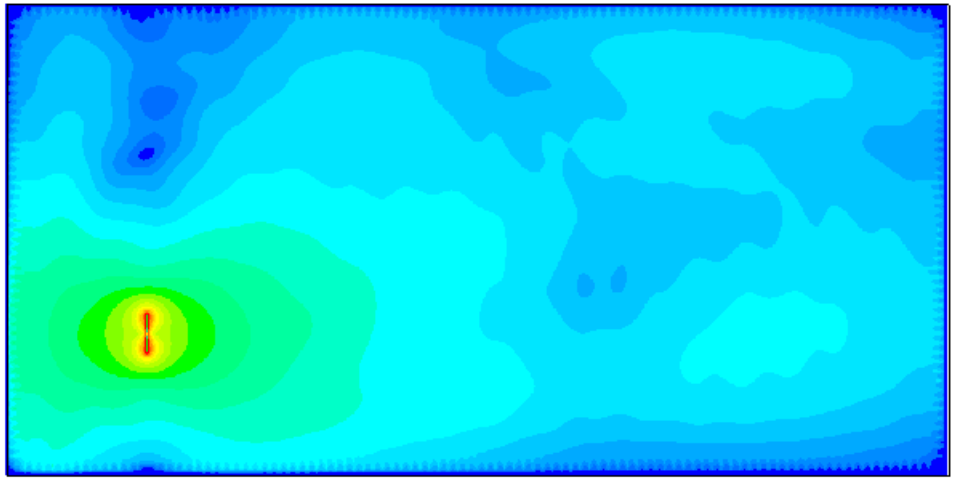
電波暗室の大きさ = 10m x 5m x 5m
セル数 = 1000 x 500 x 500 = 2.5億
(セルサイズ=1cm)
使用メモリー = 6.6GB x 2

(1) 周波数 = 300MHz
タイムステップ数 = 2900
計算時間 = 8分 (@2GPU)

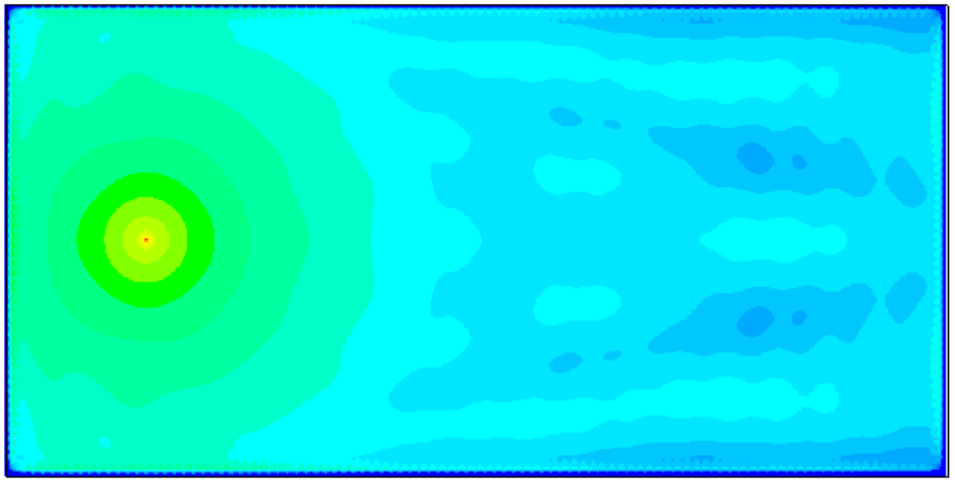
(2) 周波数 = 1GHz
タイムステップ数 = 1900
計算時間 = 6分 (@2GPU)

電波暗室(2/2)

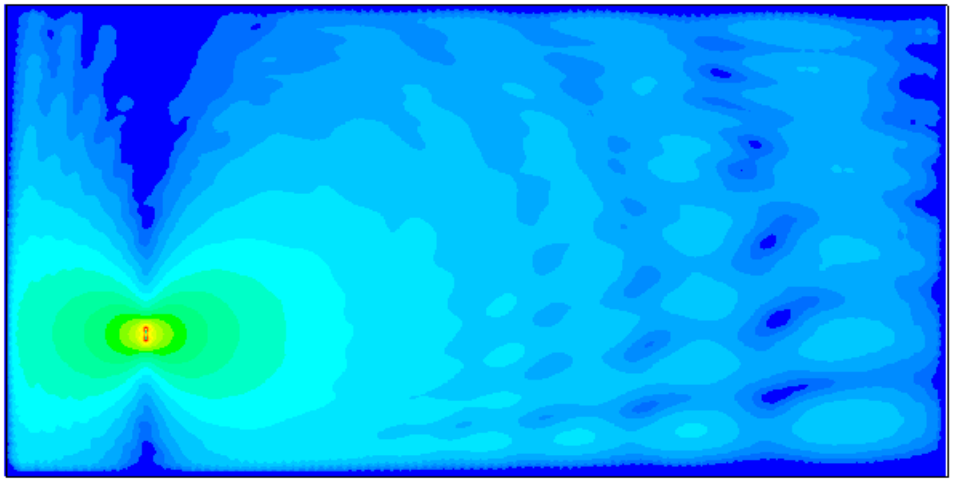
電波暗室内の電界分布



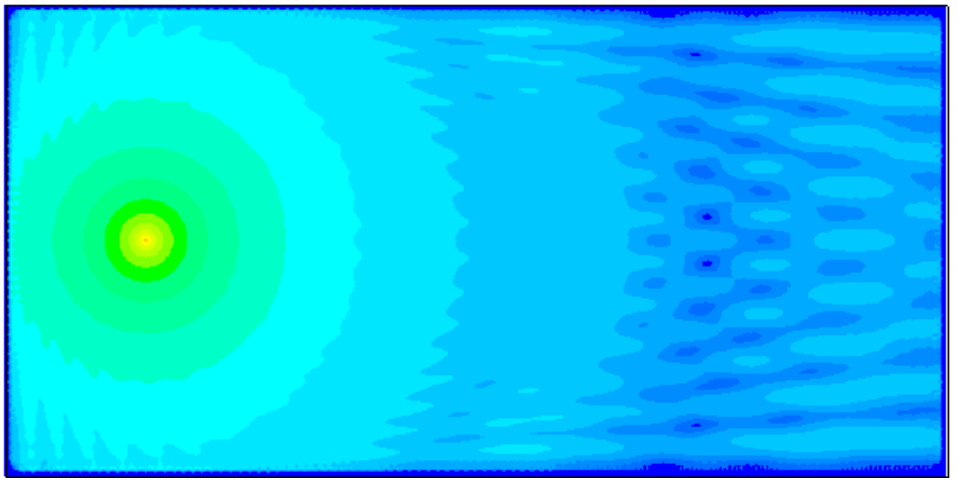
300MHz 垂直面



300MHz 水平面 床上1.5m



1GHz 垂直面



1GHz 水平面 床上1.5m