FDTD를 이용한 마이크로스트립 안테나의 해석연구

연구책임자 이천분소 기술과 박석주
연 구 원 이천분소 기술과 박정규

요 약 문

전자파 수치해석 기법인 유한차분시간영역법(FDTD)을 소개하였고 이를 이용하여 한 개의 패치를 갖는 마이크로스트립 안테나의 전자장의 분포를 해석하기 위하여 코드작업을 하였다. Mur의 경계조건을 적용하였고 해석된 결과를 활용하여 마이크로스트립 안테나의 해석을 위한 윈도우 프로그램을 개발하였다.

SUMMARY

We introduce Electromagnetic numerical analysis, FDTD and simulate the electromagnetic field distribution of a microstrip antenna having single patch. We use Mur's absorbing boundary condition, code FDTD and develope a window program simulating the one-patch microstrip antenna.

목　　차

제1장 서론

1970년대 후반 이후로 국제적인 안테나 연구 그룹은 마이크로스트립 안테나에 대한 실험 및 이론연구를 진행해왔다. 마이크로스트립 안테나는 낮은 프로파일과 고집적회로와의 쉬운 결합성 그리고 표면모양에 그다지 구애받지 않고 응용이 가능하기 때문에 군사분야에서 항공기, 미사일 그리고 로켓같은 분야의 발전에 기여해 왔고 상업분야에서 위성이동통신, 위성방송시스템과 위치추적시스템(GPS) 그리고 리모트센싱 분야 발전에 많은 공헌을 해 왔으며 장차도 더 많은 발전이 기대된다. 이러한 마이크로스트립 안테나 시스템은 두가지 범주로 분류할수 있는데, 하나는 위성수신 안테나나 단일펄스 추적레이다와 같은 고 레벨의 파워를 방사하는 어레이 안테나와 둘째, 생체의학분야에 응용되는 안테나나 셀룰라폰 또는 GPS에 적용되는 낮은파워를 가진 송수신안테나와 같은 두범주로 나뉘어진다.
1966년 K. S. Yee가 맥스웰 방정식을 시간과 공간영역을 이산화함으로써 수치적으로 그 해를 구하는 알고리즘으로서 FDTD(유한차분시간영역법)를 제안하였다. 이 FDTD법은 모든 영역의 파를 모델링하는 도구로서 전자파를 연구하는 연구자들에게 주목을 받아왔고 여러 전자파와 관련된 여러분야에서 성공적으로 적용되었다. 또한 이 테크닉은 비 균질적인 시스템에서도 쉽게 적용할수 가 있기 때문에 마이크로스트립 안테나와 같은 구조의 안테나를 해석하는데 편리하다고 알려졌다. 본 보고서에서도 한 개의 패치를 갖는 마이크로스트립 안테나를 해석하기 위하여 FDTD를 도입하고 이것을 이용하여 마이크로스트립 안테나에서 방사되는 전자파의 분포를 해석하였다.

제2장 FDTD의 기본

1) 맥스웰방정식과 FDTD 방정식

FDTD는 다음 식 (1)과 같은 맥스웰방정식을 이산적으로 풀기위하여 두 Curl 방정식을 결합하여 만든 알고리즘이다.

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

$$\vec{\nabla} \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t} \qquad (1)$$

여기서 $\vec{J_M}$은 자기전류이다.

위의 방정식에서 $\vec{D}=\varepsilon\vec{E}$, $\vec{B}=\mu\vec{H}$, $\vec{J}=\sigma\vec{E}$ **그리고** $\vec{J_M}=\rho'\vec{H}$(ρ' **등가 자기비 저항)을 적용하여 방정식** (2)**로 변형할수 있다**.

$$\frac{\partial\vec{H}}{\partial t} = -\frac{1}{\mu}\vec{\nabla}\times\vec{E} - \rho'\vec{H}$$

$$\frac{\partial\vec{E}}{\partial t} = -\frac{1}{\varepsilon}\vec{\nabla}\times\vec{H} - \frac{\sigma}{\varepsilon}\vec{E} \qquad (2)$$

식 (2)**에** FDTD**를 적용하기 위하여** Yee**셀이라고 부르는 그림** 1**의** FDTD **격자구조를 고려해야 한다**.

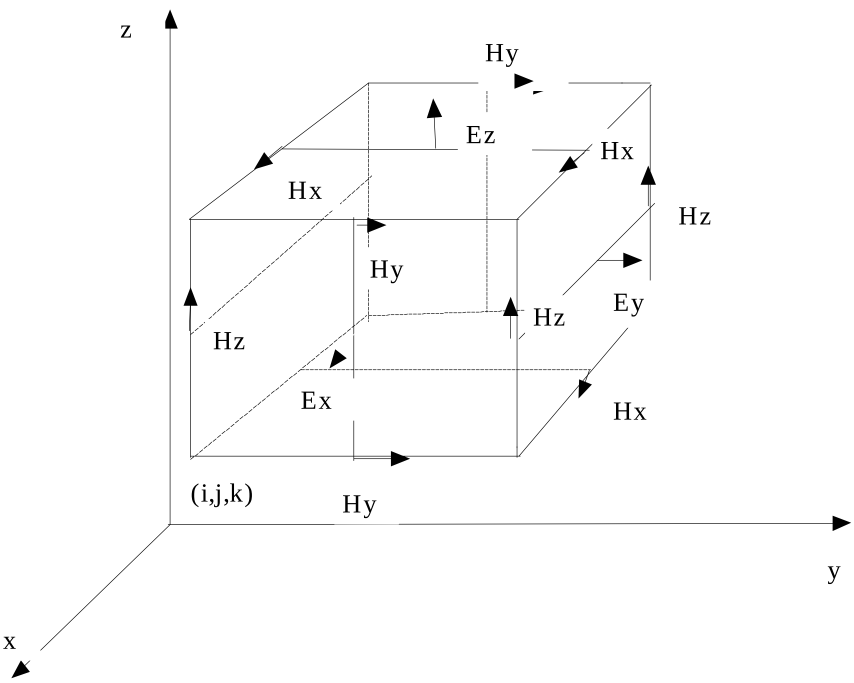


그림 1 Yee **셀**

식 (2)**를 성분별로 구분하면 맥스웰 방정식의 전기장의 방정식은 다음의 세 식이고**

$$\frac{\partial E_x}{\partial t} = -\frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial t} - \frac{\partial H_y}{\partial t}\right) - \frac{\sigma}{\varepsilon}E_x$$

$$\frac{\partial E_y}{\partial t} = -\frac{1}{\varepsilon}\left(\frac{\partial H_x}{\partial t} - \frac{\partial H_z}{\partial t}\right) - \frac{\sigma}{\varepsilon}E_y$$

$$\frac{\partial E_z}{\partial t} = -\frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial t} - \frac{\partial H_x}{\partial t}\right) - \frac{\sigma}{\varepsilon}E_z \qquad (3)$$

자기장의 방정식은 또한 다음 세 식으로 표현된다.

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_y}{\partial t} - \frac{\partial E_z}{\partial t}\right) - \frac{\rho'}{\mu}H_x$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial t} - \frac{\partial E_x}{\partial t}\right) - \frac{\rho'}{\mu}H_y$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial t} - \frac{\partial E_y}{\partial t}\right) - \frac{\rho'}{\mu}H_z \qquad (4)$$

(3)**식과** (4)**식을** FDTD **공간으로 변형하면,**

$$E^{n+1}_{i+\frac{1}{2},j,k} = \frac{\Delta t/\varepsilon}{1+\Delta t\sigma/2\varepsilon}\left[\frac{1}{\Delta y}\left(H^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} - H^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2},k}\right) - \frac{1}{\Delta z}\left(H^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} - H^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k-\frac{1}{2}}\right)\right] + \frac{1-\Delta t\sigma/2\varepsilon}{1+\Delta t\sigma/2\varepsilon}E^{n}_{i+\frac{1}{2},j,k}$$

$$E^{n+1}_{i,j+\frac{1}{2},k} = \frac{\Delta t/\varepsilon}{1+\Delta t\sigma/2\varepsilon}\left[\frac{1}{\Delta z}\left(H^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} - H^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k-\frac{1}{2}}\right) - \frac{1}{\Delta x}\left(H^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} - H^{n+\frac{1}{2}}_{i-\frac{1}{2},j+\frac{1}{2},k}\right)\right] + \frac{1-\Delta t\sigma/2\varepsilon}{1+\Delta t\sigma/2\varepsilon}E^{n}_{i,j+\frac{1}{2},k}$$

$$E^{n+1}_{i,j,k+\frac{1}{2}} = \frac{\Delta t/\varepsilon}{1+\Delta t\sigma/2\varepsilon}\left[\frac{1}{\Delta x}\left(H^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} - H^{n+\frac{1}{2}}_{i-\frac{1}{2},j,k+\frac{1}{2}}\right) - \frac{1}{\Delta y}\left(H^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} - H^{n+\frac{1}{2}}_{i-\frac{1}{2},j,k+\frac{1}{2}}\right)\right] + \frac{1-\Delta t\sigma/2\varepsilon}{1+\Delta t\sigma/2\varepsilon}E^{n}_{i,j,k+\frac{1}{2}} \qquad (5)$$

$$H^{n+\frac{1}{2}}_{i,j-\frac{1}{2},k+\frac{1}{2}} = \frac{\Delta t/\mu}{1+\Delta t\rho'/2\mu}\left[\frac{1}{\Delta z}\left(E^{n}_{i,j+\frac{1}{2},k+1} - E^{n}_{i,j,k+\frac{1}{2}}\right) - \frac{1}{\Delta y}\left(E^{n}_{i,j+1,k+\frac{1}{2}} - E^{n}_{i,j,k+\frac{1}{2}}\right)\right] + \frac{1-\Delta t\rho'/2\mu}{1+\Delta t\rho'/2\mu}H^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}}$$

$$H^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} = \frac{\Delta t/\mu}{1+\Delta t\rho'/2\mu}\Big[\frac{1}{\Delta x}\big(E^{n}_{i+1,j,k+\frac{1}{2}} - E^{n}_{i,j,k+\frac{1}{2}}\big) - \frac{1}{\Delta z}\big(E^{n}_{i+\frac{1}{2},j,k+1} - E^{n}_{i+\frac{1}{2},j,k}\big)\Big] + \frac{1-\Delta t\rho'/2\mu}{1+\Delta t\rho'/2\mu} H^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}}$$

$$H^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{\Delta t/\mu}{1+\Delta t\rho'/2\mu}\Big[\frac{1}{\Delta y}\big(E^{n}_{i+\frac{1}{2},j+1,k} - E^{n}_{i+\frac{1}{2},j,k}\big) - \frac{1}{\Delta x}\big(E^{n}_{i+1,j+\frac{1}{2},k} - E^{n}_{i,j+\frac{1}{2},k}\big)\Big] + \frac{1-\Delta t\rho'/2\mu}{1+\Delta t\rho'/2\mu} H^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} \qquad (6)$$

식 (5)와 (6)이 우리의 시뮬레이션 연구에 필요한 방정식이다.

2) 제한조건

FDTD **방정식은 컴퓨터상에서 수치적으로 계산되기 때문에 몇가지 제한조건이 주어지게 된다. 그림** 1**의 셀의 크기를 무한히 작게 잡으면 정확한 해를 구할 수 있게 되지만 그러면 컴퓨터의 계산속도와 용량에도 문제가 될 뿐만 아니라 전자기학적인 한계가 있기 때문에 문제가 생긴다, 그렇다면 그 셀의 크기에는 어떠한 제한조건이 주어져야 할까**? **근본적인 구속조건은 고려하고 있는 시스템의 가장작은 파장보다 셀의 크기가 훨씬 더 작아야한다는 것이다. 연구에 따르면 대게 셀의 크기가 고주파에서 가장작은 파장** $\lambda_{\min}$**의 십분의 일 즉** $\lambda_{\min}/10$**보다 작게끔 결정되면 원하는 정도의 정확도를 얻는다고 한다. 또한 셀 자체는 정육면체일 필요는 없고 어떤 두변의 길이의 비가** 2:1**를 초과하지 말아야한다.**

셀의 크기가 결정되고 나면 시간 스텝의 최대 크기인 Δt**가** Courant **조건에 의하여 결정된다.** Courant **조건을 이해하기 위하여** FDTD **셀을 통과하는 평면파를 고려해보자. 한 개의 시간 스텝 안에 이 평면파의 어떤점은 한 개의 셀 이상을 통과하지 말아야한다. 왜냐하면** FDTD**는 그 파를 한 개의 시간스텝 동안에 컴퓨터의 계산상 한 셀로부터 가장 가까운 이웃의 셀로만 전파할수 있기 때문이다. 이것은 다음과 같은 식으로 표현할 수 가 있는데 이를 계산의 안정성을 위한** Courant **조건이라고 한다.**

$$c\Delta t \le 1/\sqrt{\frac{1}{(\Delta x)^2}+\frac{1}{(\Delta y)^2}+\frac{1}{(\Delta z)^2}} \qquad (7)$$

(3) 경계조건

웨이브 가이드와 같은 시스템을 제외하고는 대부분의 시스템의 경우에 있어서 파들은 경계가 없는 무한의 공간으로 방사된다. 그러나 우리가 고려하는 계산상의 FDTD공간은 필연적으로 경계가 주어지게 되어있다. 만약 그렇지 않다면 공간상의 무한대의 격자점에 대해서 계산을 수행하는 결과를 초래하기 때문에 그것은 불가능한 일이된다. 그러므로 고려하고 있는 시스템이 아무리 무한대의 공간으로 방사되는 전자기파를 대상으로 한다고 할지라도 FDTD는 반드시 한정된 공간으로 경계지워지게 된다. 이는 계산상의 경계영역에서 파의 산란이나 간섭등을 일으키게 되어 오차를 유발하게 된다. 이러한 난점을 해결하기 위하여 그림2와 같이 1981년에 Mur가 계산상의 경계면에서 파가 외부로 방사해나가도록 흡수경계 조건을 제시하였다.

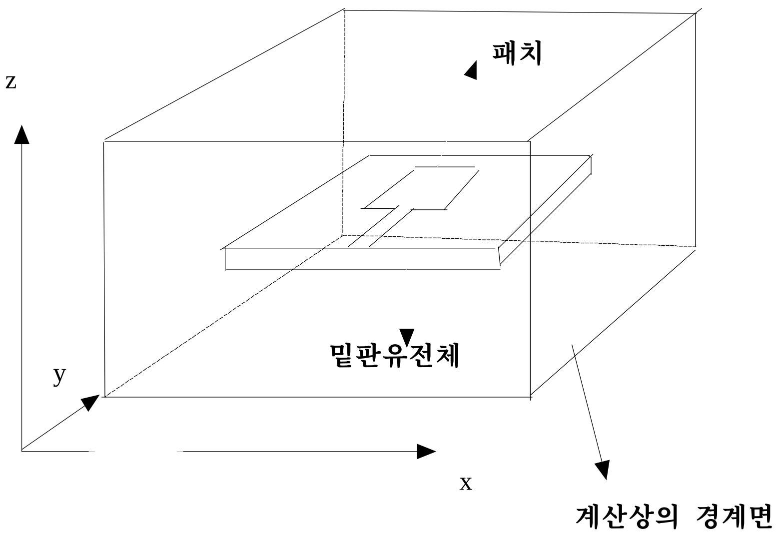


그림 2 마이크로스트립 안테나와
FDTD 계산상의 경계면

본 보고서에서도 이를 채택하여 계산을 수행하였는데 그 식은 다음과 같이 표현된다.

$$\vec{\nabla}\cdot\vec{E} - \frac{1}{c}\frac{\partial\vec{E}}{\partial t}\bigg|_{boundary} = 0 \qquad (8)$$

식 (8)을 Mur의 1차 흡수 경계조건이라고 부른다. 이 식을 FDTD로 계산이 가능하도록 이산화시키면 다음과 같다. 그림 2와 같이 계산상의 경계가 주어졌다고 할 때 x축에서의 경계조건을 만족하는 방정식은 $x=0$과 $x=l_x$에서 식 (8)을 만족해야 함으로 다음과 같다.

$$\left.\frac{\partial E}{\partial x}-\frac{1}{c}\frac{\partial E}{\partial t}\right|_{x=0}=0$$

$$\left.\frac{\partial E}{\partial x}-\frac{1}{c}\frac{\partial E}{\partial t}\right|_{x=l_x}=0 \qquad (9)$$

이를 이산화하면 각각의 결과는 (10)식이 된다.

$$E^{n+1}_{0,j,k}=E^{n}_{1,j,k}+\frac{c\Delta t-\Delta x}{c\Delta t+\Delta x}\left(E^{n+1}_{1,j,k}-E^{n}_{0,j,k}\right)$$

$$E^{n+1}_{l_x,j,k}=E^{n}_{l_{x-1},j,k}+\frac{c\Delta t-\Delta x}{c\Delta t+\Delta x}\left(E^{n+1}_{l_{x-1},j,k}-E^{n}_{l_x,j,k}\right) \qquad (10)$$

y축과 z축도 마찬가지의 과정을 거치면 식 (11)과 (12)를 얻게된다.

$$E^{n+1}_{i,0,k}=E^{n}_{i,1,k}+\frac{c\Delta t-\Delta x}{c\Delta t+\Delta x}\left(E^{n+1}_{i,1,k}-E^{n}_{i,o,k}\right)$$

$$E^{n+1}_{i,l_{y,k}}=E^{n}_{i,l_{y-1},k}+\frac{c\Delta t-\Delta x}{c\Delta t+\Delta x}\left(E^{n+1}_{i,l_{y-1},k}-E^{n}_{i,l_y,k}\right) \qquad (11)$$

$$E^{n+1}_{i,j,o}=E^{n}_{i,j,1}+\frac{c\Delta t-\Delta x}{c\Delta t+\Delta x}\left(E^{n+1}_{i,j,1}-E^{n}_{i,j,0}\right)$$

$$E^{n+1}_{i,j,l_z}=E^{n}_{i,j,l_{z-1}}+\frac{c\Delta t-\Delta x}{c\Delta t+\Delta x}\left(E^{n+1}_{i,j,l_{z-1}}-E^{n}_{i,j,l_z}\right) \qquad (12)$$

FDTD 방정식 (5), (6)과 Mur의 1차 경계조건 (10), (11), (12)가 시뮬레이션 계산에 필요한 모든 식을 망라한 것이다.

제3장 FDTD를 이용한 마이크로스트립 안테나의 모의실험

(1) 마이크로스트립 안테나의 구조

그림 3과 같은 구조를 갖는 마이크로스트립 안테나를 고려하였다. 패치의 크기는 $25\Delta x \times 25\Delta y$(단위 mm)이며 계산하고자 하는 FDTD 셀의 크기는 $45\Delta x \times 65\Delta y \times 10\Delta z$로 하였다. 또한 아래판의 상대 유전율은 전형적으로 $\varepsilon_r = 2.5$를 선택하여 계산하였다.

셀의 크기는 변화를 주어가면서 계산을 수행하여였는데 여기서는 $\Delta x \times \Delta y \times \Delta z$가 각각 mm단위로 $0.2666mm \times 0.3188mm \times 0.1600mm$이다.

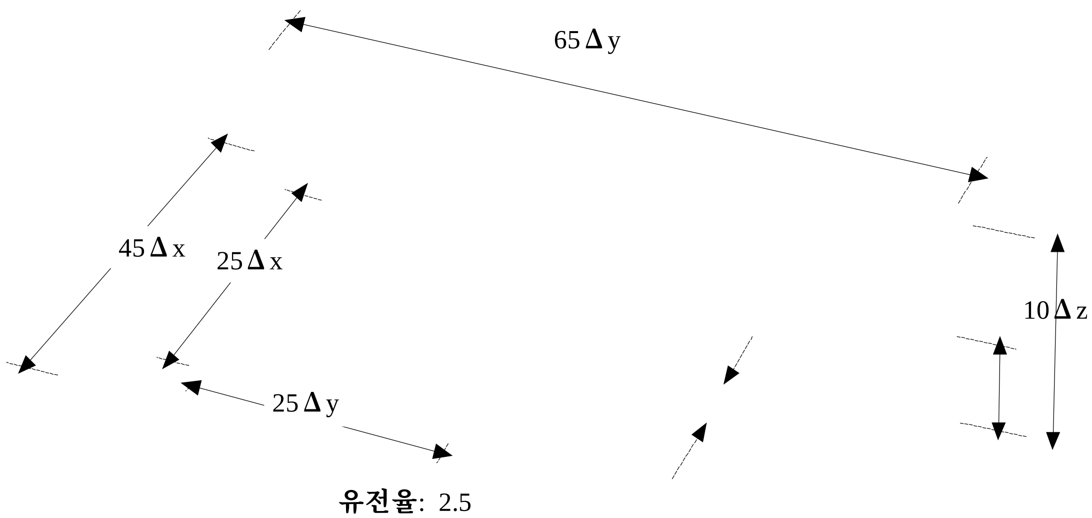


그림3 : 마이크로스트립 안테나의 구조

2) 입사파의 구현

FDTD는 광대역 특성의 분석이 가능함으로 모든 주파수를 포함하는 입력 신

호원으로서 가우시안 펄스를 사용하였다. 식 (13)의 가우시안 펄스는 한번의 신호로 광대역 응답을 얻을수 있는데 시간영역의 함수를 푸리에 변환을 함으로서 주파수 특성을 파악할수 가 있다.

$$g(t, x) = \exp[-(t - t_0 - \frac{x - x_0}{v})^2 / T^2] \qquad (13)$$

위의 식에서 입력부의 한 지점에서만 가우시안 입력펄스를 가하기 때문에 위치에 대한 의존성은 없으므로 시간에만 의존하는 방정식 (14)의 가우시안 펄스를 단일 패치의 마이크로스트립 안테나의 입력단에 입력함으로써 시뮬레이션을 수행하였다.

$$g(t) = \exp[-(t - t_0)^2 / T^2] \qquad (14)$$

이것의 주파수 특성은 식 (14)를 푸리에 변환함으로써 얻어지고 다음과 같다.

$$G(f) \propto \exp[-\pi^2 T^2 f^2]$$

우리의 시뮬레이션 연구에서는 그림 4와 같이 전기장의 z성분을 입력하였고 FDTD 계산 시간간격 $\Delta t = 0.341 \times 10^{-12}\,\mathrm{sec}$로 하였으며 $T = 34\Delta t$로 정하였다.

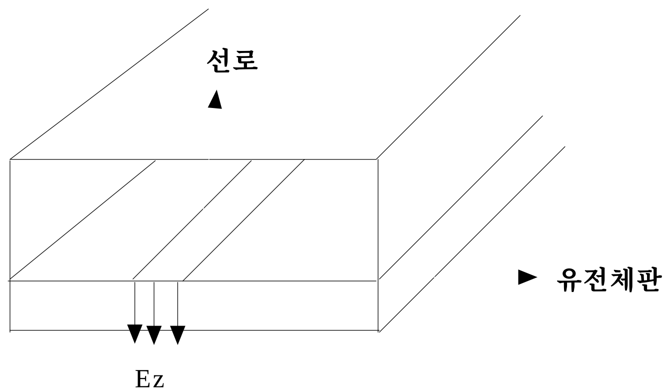


그림4 : 입력전기장과 성분

3) 시뮬레이션 결과

전자장의 z성분을 시뮬레이션하였다. 시간에 따른 분포가 그림 (5), (6), (7)에서 보여지는데 계산의 전체시간은 $3000\Delta t = 1023 nsec$로 하영 수행하였다. 포트란 프로그램은 마이크로선사의 유닉스로 실행하였는데 세 시간여의 계산시간이 걸렸다.

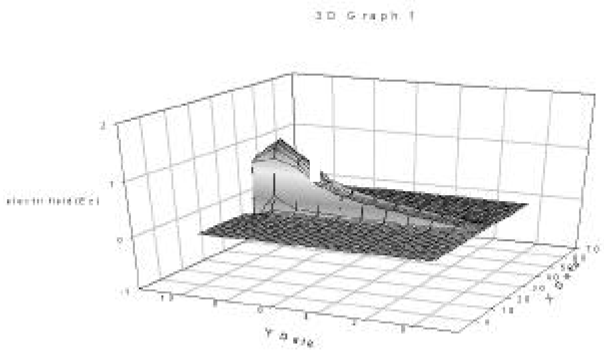


그림5:전기장의 z성분의 분포(1)

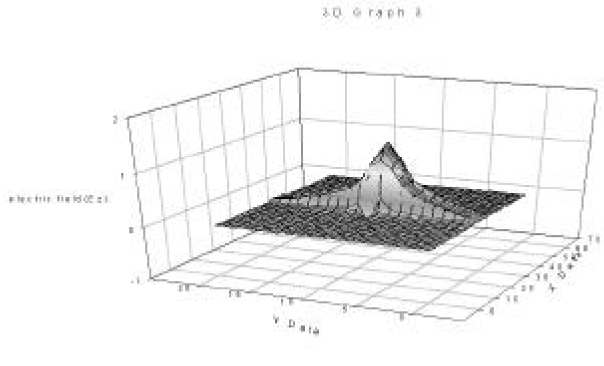


그림6:**전기장의** z**성분의 분포**(2)

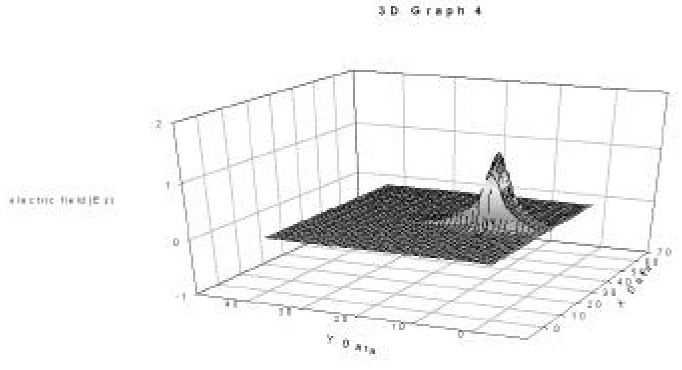


그림7:**전기장의** z**성분의 분포**(3)

4) 마이크로스트립 안테나 해석을 위한 윈도우 프로그램

위에서 해석한 결과를 토대로 사용자 인터페이스가 편리한 윈도우용 해석 프로그램을 제작하였다. FDTD 해석 포트란 코드를 Visual C++코드로 변환하여 한 개의 패치를 갖는 마이크로스트립 안테나의 전자장 분포를 시뮬레이션하여 디스플레이 하도록 제작하였다. 시간의 경과에 따라 전자장의 분포의 변화를 연속적으로 보여주도록 되어있으며 패치의 크기와 유전율을 입력하도록 되어있다. 그림 (8)은 해석용프로그램의 메인부이고 그림 (9), (10)은 프로그램을 구동하여 변화하는 상황을 보여준다. 여기서는 선로상에서 전자장의 시간적 변화를 보여준다.

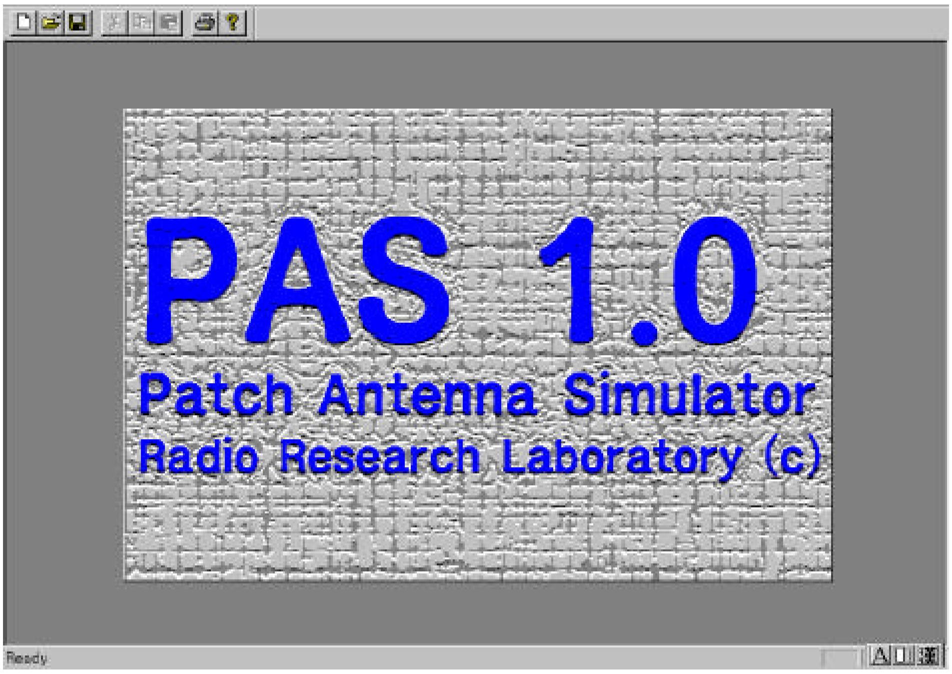
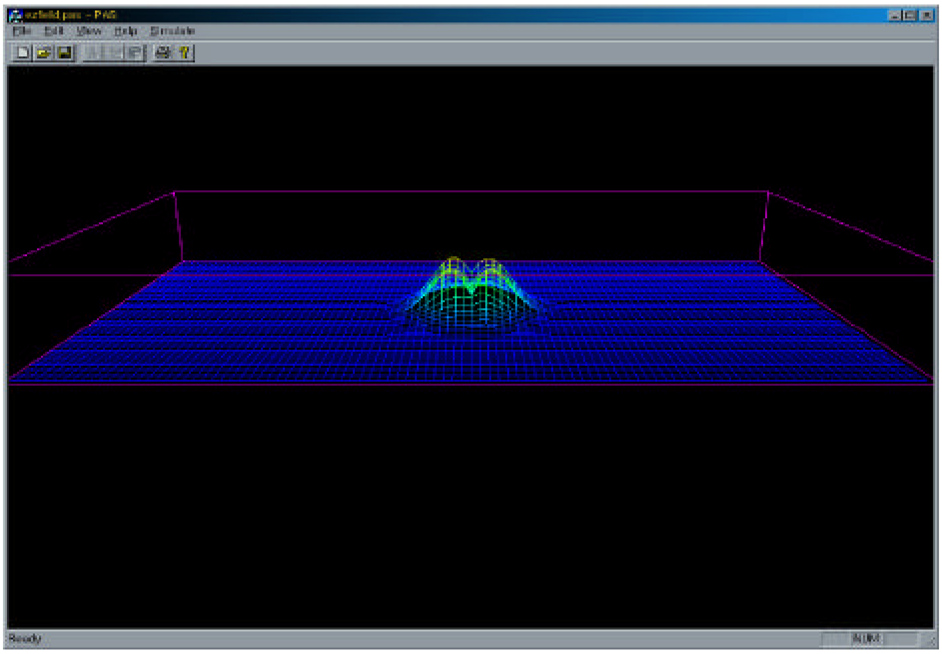


그림8:윈도우용 해석프로그램의 메인부



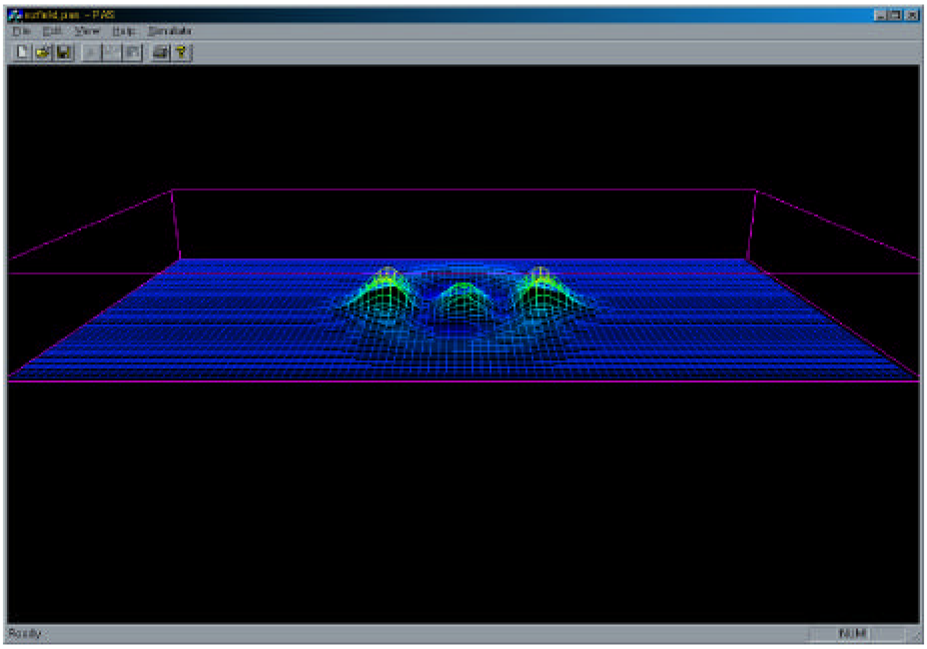


그림9, 10:**윈도우용 해석프로그램으로 구현한 전송선로위에서의 전자장분포의 변화 모습**

제4장 결론

전자파 수치해석의 FDTD를 이용하여 한 개의 패치를 갖는 마이크로스트립 안테나의 전자장 분포를 해석하였다. 이 방법은 마이크로스트립 안테나를 해석하는데 비교적 간단하면서도 효과적인 테크닉이다. 그것은 안테나의 여러 가지 특성들 예를들면 입력 임피던스나 요구된 주파수 영역에서의 방사 패턴 등에 대한 정보를 제공할수 있다. 또한 패치위에서 전자장이나 전류의 분포를 비주얼화 할수 있다는 점이 안테나의 특성에 대한 물리적인 통찰을 준다는 장점이 있다. 본 연구에서는 전통적인 FDTD 방법을 이용한 모델링으로 한 개의 패치를 갖는 마이크로스트립 안테나의 광대역 특성을 분석하였다. 또한 FDTD 코드작업에서 얻어진 결과를 가지고 윈도우용 해석 프로그램을 제작하였다. 이 연구를 통하여 기본적인 FDTD 해석방법을 구현하였으며 마이크로스트립 안테나에 대한 특성 이해를 달성하였다. 실험과의 비교연구와 계산속도의 효율성 개선 그리고 다중패치 안테나에 대한 코드적용 등앞으로의 남은 과제라 하겠다.

* 참고문헌

(1) K. S. Yee "Numerical solutions of initial boundary value problems involving Maxwell's equations isotropic media" IEEE Trans. Antennas Propagat. vol. AP-14, pp. 302-307, May 1966.

(2) Raymond J. Luebbers, Karl S. Kunz, Michael Schneider, Forrest Hunsberger "A finite-difference time domain near zone to far zone transformation" IEEE TRANS. ON ANTENNAS AND PROPAGATION, vol 39, NO.4 pp. 429-433, April 1991

(3) G. Mur, "Absorbing boundary conditions for the finite difference approximation of the time domain electromagnetic field equation" IEEE Trans. Elec. Comp. , vol EMC-32 No. 4. pp.1073-1077 Oct. 1981.

(4) Kai Fong Lee, Wei Chen "Advance in Microstrip and Printed Antennas" JOHN WIELEY & SONS, INC.

(5) Karls, Kunz, raymond j. Luebbers "The Finite Difference time Domain Method for Electromagnetics" CRC Press

*FDTD 해석 포트란 소스코드

```
      program fdtdmicro
      integer h1,h2,opoint,mtl,wtl,ctl
      parameter(nx=60,ny=100,nz=13)
      parameter(dx=0.389e-3,dy=0.4e-3,dz=0.265e-3)
      parameter(h1=4,h2=7,js=9,opoint=40)
      parameter(mtl=27,wtl=6,ctl=mtl+wtl/2,med=2)
      parameter(ntmax=3000,modmax=100)
      parameter(pie=3.14159265359,c=2.997925e8)
      parameter(e0=8.854185e-12,u0=1.2566371e-6)
      parameter(Rs=50)
      dimension   Ex(0:nx,0:ny,0:nz),
     &            Ey(0:nx,0:ny,0:nz),
     &            Ez(0:nx,0:ny,0:nz),
     &            Hx(0:nx,0:ny,0:nz),
     &            Hy(0:nx,0:ny,0:nz),
     &            Hz(0:nx,0:ny,0:nz),
     &            RE(med),ER(med),Vt(ntmax)
                  ! ER : dielectric constant
                       ! med=1 : DIELECTRIC MATERIAL
                       ! med=2 : FREE SPACE

c arrays to store 2 previous time steps on each side
c array format : Xyt(n, ny, nz)
c       n = 0 x=0 plane (absorbing boundary)
c       n = 1 x=x1 plane
c       n = 2 x=Lx-1 plane
c       n = 3 x=Lx plane (absorbing boundary)
c       t = 0 or 1 to save previous 2 time steps
c Xzt(n, ny, nz) - same but for tangential z component
      dimension Xy0(0:3,0:ny,0:nz)
     &          ,Xz0(0:3,0:ny,0:nz)
     &          ,Yx0(0:nx,0:3,0:nz)
     &          ,Yz0(0:nx,0:3,0:nz)
     &          ,Zx0(0:nx,0:ny,0:3)
     &          ,Zy0(0:nx,0:ny,0:3)
      dimension Xy1(0:3,0:ny,0:nz)
     &          ,Xz1(0:3,0:ny,0:nz)
     &          ,Yx1(0:nx,0:3,0:nz)
     &          ,Yz1(0:nx,0:3,0:nz)
     &          ,Zx1(0:nx,0:ny,0:3)
     &          ,Zy1(0:nx,0:ny,0:3)

      data ER/2.5, 1.0/

c      dt=c*SQRT(1/dx**2+1/dy**2+1/dz**2)
      dt=0.341e-12
      if(dt.gt.c*SQRT(1/dx**2+1/dy**2+1/dz**2)) then
```

```
      print *, 'reset the time'
      endif
      t1=34*dt
      t0=3.*t1
c       tg=t0+4.*t1
      rh=dt/u0
      do m=1,med
      RE(m)=dt/(e0*ER(m))
      enddo

c initialization
      do 10 i=0,nx
      do 10 j=0,ny
      do 10 k=0,nz
      Ex(i,j,k)=0.
      Ey(i,j,k)=0.
      Ez(i,j,k)=0.
      Hx(i,j,k)=0.
      Hy(i,j,k)=0.
      Hz(i,j,k)=0.
10    continue

      do nt=1,ntmax
      Vt(nt)=0.0
      enddo
      print *, 'initialization completed'

      t=0.0
          open(1,file='ezfild.dat',status='unknown')
      open(0,file='v.dat',status='unknown')


      do 30 nt=1, ntmax   ! TIME LOOP
      print *, 'nt=', nt ,Ez(ctl,js,h2-1), Ez(ctl,js+10,h2-1)
c source input
      if(t.le.2*t0) then
          psource=exp(-(t-t0)**2./t1**2.)
      do 332 k=0,nz-1
      do 332 i=0,nx
          if(k.ge.h1.and.k.lt.h2) then
          if(i.ge.mtl.and.i.le.mtl+wtl) then
      Ez(i,js,k)=
c hard source
          &psource
c include the source resistence
     &+((Hy(i,js,k)-Hy(i-1,js,k))*dy
     & +(Hx(i,js-1,k)-Hx(i,js,k))*dx)*Rs/dz
c soft source
     &+Ez(i,js,k)

           else
```

```
          Ez(i,js,k)=0.
          endif
          elseif(k.ge.h2) then
          Ez(i,js,k)=0.0
          endif
332   continue
          endif

c Hx step
      do 600 i=0,nx
      do 600 j=0,ny-1
      do 600 k=0,nz-1
      Hx(i,j,k)=Hx(i,j,k)
     &-rh*((Ez(i,j+1,k)-Ez(i,j,k))/dy
     &     -(Ey(i,j,k+1)-Ey(i,j,k))/dz)
600   continue
c Hy step
      do 700 i=0,nx-1
      do 700 j=0,ny
      do 700 k=0,nz-1
      Hy(i,j,k)=Hy(i,j,k)
     &-rh*((Ex(i,j,k+1)-Ex(i,j,k))/dz
     &     -(Ez(i+1,j,k)-Ez(i,j,k))/dx)
700   continue
c Hz step
      do 800 i=0,nx-1
      do 800 j=0,ny-1
      do 800 k=0,nz
      Hz(i,j,k)=Hz(i,j,k)
     &-rh*((Ey(i+1,j,k)-Ey(i,j,k))/dx
     &     -(Ex(i,j+1,k)-Ex(i,j,k))/dy)
800   continue

c Ex step
      do 41 i=1, nx-1
      do 41 j=1, ny-1
      do 41 k=0,h1
      m=2
      Ex(i,j,k)=Ex(i,j,k)
     &+RE(m)*((Hz(i,j,k)-Hz(i,j-1,k))/dy
     &          -(Hy(i,j,k)-Hy(i,j,k-1))/dz)
41    continue
      do 42 i=1, nx-1
      do 42 j=1, ny-1
      do 42 k=h1,h2
      m=1
      Ex(i,j,k)=Ex(i,j,k)
     &+RE(m)*((Hz(i,j,k)-Hz(i,j-1,k))/dy
     &          -(Hy(i,j,k)-Hy(i,j,k-1))/dz)
42    continue
      do 43 i=1, nx-1
```

```
      do 43 j=1, ny-1
      do 43 k=h2,nz-1
      m=2
      Ex(i,j,k)=Ex(i,j,k)
     &+RE(m)*((Hz(i,j,k)-Hz(i,j-1,k))/dy
     &        -(Hy(i,j,k)-Hy(i,j,k-1))/dz)
43    continue
c Ey step
      do 51 i=1, nx-1
      do 51 j=1, ny-1
      do 51 k=0,h1
      m=2
      Ey(i,j,k)=Ey(i,j,k)
     &+RE(m)*((Hx(i,j,k)-Hx(i,j,k-1))/dz
     &        -(Hz(i,j,k)-Hz(i-1,j,k))/dx)
51    continue
      do 52 i=1, nx-1
      do 52 j=1, ny-1
      do 52 k=h1,h2
      m=1
      Ey(i,j,k)=Ey(i,j,k)
     &+RE(m)*((Hx(i,j,k)-Hx(i,j,k-1))/dz
     &        -(Hz(i,j,k)-Hz(i-1,j,k))/dx)
52    continue
      do 53 i=1, nx-1
      do 53 j=1, ny-1
      do 53 k=h2,nz-1
      m=2
      Ey(i,j,k)=Ey(i,j,k)
     &+RE(m)*((Hx(i,j,k)-Hx(i,j,k-1))/dz
     &        -(Hz(i,j,k)-Hz(i-1,j,k))/dx)
53    continue
c Ez step
      do 61 i=1, nx-1
      do 61 j=1, ny-1
      do 61 k=0,h1
      m=2
      Ez(i,j,k)=Ez(i,j,k)
     &+RE(m)*((Hy(i,j,k)-Hy(i-1,j,k))/dx
     &        -(Hx(i,j,k)-Hx(i,j-1,k))/dy)
61    continue
      do 62 i=1, nx-1
      do 62 j=1, ny-1
      do 62 k=h1,h2
      m=1
      Ez(i,j,k)=Ez(i,j,k)
     &+RE(m)*((Hy(i,j,k)-Hy(i-1,j,k))/dx
     &        -(Hx(i,j,k)-Hx(i,j-1,k))/dy)
62    continue
      do 63 i=1, nx-1
      do 63 j=1, ny-1
```

```
      do 63 k=h2,nz-1
      m=2
      Ez(i,j,k)=Ez(i,j,k)
     &+RE(m)*((Hy(i,j,k)-Hy(i-1,j,k))/dx
     &       -(Hx(i,j,k)-Hx(i,j-1,k))/dy)
63    continue

c boundary condition for conductors & boundary layer of two medium
      do 73 i=1, nx-1
      do 73 j=1, ny-1
      Ex(i,j,h1)=0.0
      Ey(i,j,h1)=0.0
      if(i.ge.mtl.and.i.lt.mtl+wtl) then
      Ex(i,j,h2)=0.0
      Ey(i,j,h2)=0.0
      elseif(i.lt.mtl.or.i.gt.mtl+wtl) then
      Ex(i,j,h2)=Ex(i,j,h2)+2.0*dt/((ER(1)+ER(2))*E0)
     &*((Hz(i,j,h2)-Hz(i,j-1,h2))/dy
c     & -(Hy(i,j,h2)-Hy(i,j,h2-1))/dz)
     & -(Hy(i,j,h2)-Hy(i-1,j,h2))/dz)
      Ey(i,j,h2)=Ey(i,j,h2)+2.0*dt/((ER(1)+ER(2))*E0)
c     &*((Hx(i,j,h2)-Hx(i,j,h2-1))/dz
     &*((Hx(i,j,h2)-Hx(i,j-1,h2))/dz
     & -(Hz(i,j,h2)-Hz(i-1,j,h2))/dx)
          else
         Ey(i,j,h2)=0.0
      endif
73    continue

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCC  MUR'S  2ND ORDER ABC  CCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c The following need only be done once
      if (nt.eq.1) then

c constants in calculations of surface boundary values
      cx1=(c*dt-dx)/(c*dt+dx)
      cx2=(2*dx)/(c*dt+dx)
      cx3=cx2*(c*dt)**2/(4*dy*dy)
      cx4=cx2*(c*dt)**2/(4*dz*dz)
      cy1=(c*dt-dy)/(c*dt+dy)
      cy2=(2*dy)/(c*dt+dy)
      cy3=cy2*(c*dt)**2/(4*dx*dx)
      cy4=cy2*(c*dt)**2/(4*dz*dz)
      cz1=(c*dt-dz)/(c*dt+dz)
      cz2=(2*dz)/(c*dt+dz)
      cz3=cz2*(c*dt)**2/(4*dx*dx)
      cz4=cz2*(c*dt)**2/(4*dy*dy)
         cxy=(c*dt-0.5*sqrt(dx**2+dy**2))
         &/(c*dt+0.5*sqrt(dx**2+dy**2))
```

```
        cxz=(c*dt- 0.5*sqrt(dx**2+dz**2))
        &/(c*dt+0.5*sqrt(dx**2+dz**2))
        cyz=(c*dt- 0.5*sqrt(dz**2+dy**2))
        &/(c*dt+0.5*sqrt(dz**2+dy**2))
c  initialization
      do  2  k=0,nz
      do  2  j=0,ny
      Xy0(0,j,k)=0.
      Xy0(1,j,k)=0.
      Xy0(2,j,k)=0.
      Xy0(3,j,k)=0.
      Xy1(0,j,k)=0.
      Xy1(1,j,k)=0.
      Xy1(2,j,k)=0.
      Xy1(3,j,k)=0.
      Xz0(0,j,k)=0.
      Xz0(1,j,k)=0.
      Xz0(2,j,k)=0.
      Xz0(3,j,k)=0.
      Xz1(0,j,k)=0.
      Xz1(1,j,k)=0.
      Xz1(2,j,k)=0.
      Xz1(3,j,k)=0.
2     continue
      do  4  k=0,nz
      do  4  i=0,nx
      Yx0(i,0,k)=0.
      Yx0(i,1,k)=0.
      Yx0(i,2,k)=0.
      Yx0(i,3,k)=0.
      Yx1(i,0,k)=0.
      Yx1(i,1,k)=0.
      Yx1(i,2,k)=0.
      Yx1(i,3,k)=0.
      Yz0(i,0,k)=0.
      Yz0(i,1,k)=0.
      Yz0(i,2,k)=0.
      Yz0(i,3,k)=0.
      Yz1(i,0,k)=0.
      Yz1(i,1,k)=0.
      Yz1(i,2,k)=0.
      Yz1(i,3,k)=0.
4     continue
      do  6  j=0,ny
      do  6  i=0,nx
      Zx0(i,j,0)=0.
```

```
      Zx0(i,j,1)=0.
      Zx0(i,j,2)=0.
      Zx0(i,j,3)=0.
      Zx1(i,j,0)=0.
      Zx1(i,j,1)=0.
      Zx1(i,j,2)=0.
      Zx1(i,j,3)=0.
      Zy0(i,j,0)=0.
      Zy0(i,j,1)=0.
      Zy0(i,j,2)=0.
      Zy0(i,j,3)=0.
      Zy1(i,j,0)=0.
      Zy1(i,j,1)=0.
      Zy1(i,j,2)=0.
      Zy1(i,j,3)=0.
6     continue
      endif
c tangential E fields on the ABS
      do 110 j=1,ny-1
      do 110 k=1,nz-1
c x=0 boundary
      Ey(0,j,k)=-Xy0(1,j,k)
     &+cx1*(Ey(1,j,k)+Xy0(0,j,k))
     &+cx2*(Xy1(0,j,k)+Xy1(1,j,k))
     &+cx3*(Xy1(0,j+1,k)-2*Xy1(0,j,k)+Xy1(0,j-1,k)
     &      +Xy1(1,j+1,k)-2*Xy1(1,j,k)+Xy1(1,j-1,k))
     &+cx4*(Xy1(0,j,k+1)-2*Xy1(0,j,k)+Xy1(0,j,k-1)
     &      +Xy1(1,j,k+1)-2*Xy1(1,j,k)+Xy1(1,j,k-1))
      Ez(0,j,k)=-Xz0(1,j,k)
     &+cx1*(Ez(1,j,k)+Xz0(0,j,k))
     &+cx2*(Xz1(0,j,k)+Xz1(1,j,k))
     &+cx3*(Xz1(0,j+1,k)-2*Xz1(0,j,k)+Xz1(0,j-1,k)
     &      +Xz1(1,j+1,k)-2*Xz1(1,j,k)+Xz1(1,j-1,k))
     &+cx4*(Xz1(0,j,k+1)-2*Xz1(0,j,k)+Xz1(0,j,k-1)
     &      +Xz1(1,j,k+1)-2*Xz1(1,j,k)+Xz1(1,j,k-1))
c x=Lx boundary
      Ey(nx,j,k)=-Xy0(2,j,k)
     &+cx1*(Ey(nx-1,j,k)+Xy0(3,j,k))
     &+cx2*(Xy1(3,j,k)+Xy1(2,j,k))
     &+cx3*(Xy1(3,j+1,k)-2*Xy1(3,j,k)+Xy1(3,j-1,k)
     &      +Xy1(2,j+1,k)-2*Xy1(2,j,k)+Xy1(2,j-1,k))
     &+cx4*(Xy1(3,j,k+1)-2*Xy1(3,j,k)+Xy1(3,j,k-1)
     &      +Xy1(2,j,k+1)-2*Xy1(2,j,k)+Xy1(2,j,k-1))
      Ez(nx,j,k)=-Xz0(2,j,k)
     &+cx1*(Ez(nx-1,j,k)+Xz0(3,j,k))
     &+cx2*(Xz1(3,j,k)+Xz1(2,j,k))
```

```
     &+cx3*(Xz1(3,j+1,k)-2*Xz1(3,j,k)+Xz1(3,j-1,k)
     &       +Xz1(2,j+1,k)-2*Xz1(2,j,k)+Xz1(2,j-1,k))
     &+cx4*(Xz1(3,j,k+1)-2*Xz1(3,j,k)+Xz1(3,j,k-1)
     &       +Xz1(2,j,k+1)-2*Xz1(2,j,k)+Xz1(2,j,k-1))
110   continue
        do 210 i=1,nx-1
        do 210 k=1,nz-1
c y=0 boundary
      Ex(i,0,k)=-Yx0(i,1,k)
     &+cy1*(Ex(i,1,k)+Yx0(i,0,k))
     &+cy2*(Yx1(i,0,k)+Yx1(i,1,k))
     &+cy3*(Yx1(i+1,0,k)-2*Yx1(i,0,k)+Yx1(i-1,0,k)
     &       +Yx1(i+1,1,k)-2*Yx1(i,1,k)+Yx1(i-1,1,k))
     &+cy4*(Yx1(i,0,k+1)-2*Yx1(i,0,k)+Yx1(i,0,k-1)
     &       +Yx1(i,1,k+1)-2*Yx1(i,1,k)+Yx1(i,1,k-1))
      Ez(i,0,k)=-Yz0(i,1,k)
     &+cy1*(Ez(i,1,k)+Yz0(i,0,k))
     &+cy2*(Yz1(i,0,k)+Yz1(i,1,k))
     &+cy3*(Yz1(i+1,0,k)-2*Yz1(i,0,k)+Yz1(i-1,0,k)
     &       +Yz1(i+1,1,k)-2*Yz1(i,1,k)+Yz1(i-1,1,k))
     &+cy4*(Yz1(i,0,k+1)-2*Yz1(i,0,k)+Yz1(i,0,k-1)
     &       +Yz1(i,1,k+1)-2*Yz1(i,1,k)+Yz1(i,1,k-1))
c y=Ly boundary
      Ex(i,ny,k)=-Yx0(i,2,k)
     &+cy1*(Ex(i,ny-1,k)+Yx0(i,3,k))
     &+cy2*(Yx1(i,2,k)+Yx1(i,3,k))
     &+cy3*(Yx1(i+1,3,k)-2*Yx1(i,3,k)+Yx1(i-1,3,k)
     &       +Yx1(i+1,2,k)-2*Yx1(i,2,k)+Yx1(i-1,2,k))
     &+cy4*(Yx1(i,3,k+1)-2*Yx1(i,3,k)+Yx1(i,3,k-1)
     &       +Yx1(i,2,k+1)-2*Yx1(i,2,k)+Yx1(i,2,k-1))
      Ez(i,ny,k)=-Yz0(i,2,k)
     &+cy1*(Ez(i,ny-1,k)+Yz0(i,3,k))
     &+cy2*(Yz1(i,2,k)+Yz1(i,3,k))
     &+cy3*(Yz1(i+1,3,k)-2*Yz1(i,3,k)+Yz1(i-1,3,k)
     &       +Yz1(i+1,2,k)-2*Yz1(i,2,k)+Yz1(i-1,2,k))
     &+cy4*(Yz1(i,3,k+1)-2*Yz1(i,3,k)+Yz1(i,3,k-1)
     &       +Yz1(i,2,k+1)-2*Yz1(i,2,k)+Yz1(i,2,k-1))
210   continue
      do 310 i=1,nx-1
      do 310 j=1,ny-1
c z=0 boundary
      Ex(i,j,0)=-Zx0(i,j,1)
     &+cz1*(Ex(i,j,1)+Zx0(i,j,0))
     &+cz2*(Zx1(i,j,0)+Zx1(i,j,1))
     &+cz3*(Zx1(i+1,j,0)-2*Zx1(i,j,0)+Zx1(i-1,j,0)
     &       +Zx1(i+1,j,1)-2*Zx1(i,j,1)+Zx1(i-1,j,1))
     &+cz4*(Zx1(i,j+1,0)-2*Zx1(i,j,0)+Zx1(i,j-1,0)
```

```
     &         +Zx1(i,j+1,1)-2*Zx1(i,j,1)+Zx1(i,j-1,1))
      Ey(i,j,0)=-Zy0(i,j,1)
     &+cz1*(Ey(i,j,1)+Zy0(i,j,0))
     &+cz2*(Zy1(i,j,0)+Zy1(i,j,1))
     &+cz3*(Zy1(i+1,j,0)-2*Zy1(i,j,0)+Zy1(i-1,j,0)
     &         +Zy1(i+1,j,1)-2*Zy1(i,j,1)+Zy1(i-1,j,1))
     &+cz4*(Zy1(i,j+1,0)-2*Zy1(i,j,0)+Zy1(i,j-1,0)
     &         +Zy1(i,j+1,1)-2*Zy1(i,j,1)+Zy1(i,j-1,1))
c z=Lz boundary
      Ex(i,j,nz)=-Zx0(i,j,2)
     &+cz1*(Ex(i,j,nz-1)+Zx0(i,j,3))
     &+cz2*(Zx1(i,j,2)+Zx1(i,j,3))
     &+cz3*(Zx1(i+1,j,3)-2*Zx1(i,j,3)+Zx1(i-1,j,3)
     &         +Zx1(i+1,j,2)-2*Zx1(i,j,2)+Zx1(i-1,j,2))
     &+cz4*(Zx1(i,j+1,3)-2*Zx1(i,j,3)+Zx1(i,j-1,3)
     &         +Zx1(i,j+1,2)-2*Zx1(i,j,2)+Zx1(i,j-1,2))
      Ey(i,j,nz)=-Zy0(i,j,2)
     &+cz1*(Ey(i,j,nz-1)+Zy0(i,j,3))
     &+cz2*(Zy1(i,j,2)+Zy1(i,j,3))
     &+cz3*(Zy1(i+1,j,3)-2*Zy1(i,j,3)+Zy1(i-1,j,3)
     &         +Zy1(i+1,j,2)-2*Zy1(i,j,2)+Zy1(i-1,j,2))
     &+cz4*(Zy1(i,j+1,3)-2*Zy1(i,j,3)+Zy1(i,j-1,3)
     &         +Zy1(i,j+1,2)-2*Zy1(i,j,2)+Zy1(i,j-1,2))
310   continue

c edges
c parallel to x-axis
      do i=0,nx-1
      Ex(i,0,0)=0.5*(Zx1(i,1,0)+Yx1(i,0,1))
          &+cyz*(0.5*(Ex(i,1,0)+Ex(i,0,1))-Yx1(i,0,0))
      Ex(i,0,nz)=0.5*(Zx1(i,1,3)+Yx1(i,0,nz-1))
          &+cyz*(0.5*(Ex(i,0,nz-1)+Ex(i,1,nz))-Yx1(i,0,nz))
          Ex(i,ny,0)=0.5*(Zx1(i,ny-1,0)+Yx1(i,3,1))
          &+cyz*(0.5*(Ex(i,ny-1,0)+Ex(i,ny,1))-Yx1(i,3,0))
          Ex(i,ny,nz)=0.5*(Zx1(i,ny-1,3)+Yx1(i,3,nz-1))
          &+cyz*(0.5*(Ex(i,ny,nz-1)+Ex(i,ny-1,nz))-Yx1(i,3,nz))
      enddo
c parallel to y-axis
      do j=0,ny-1
      Ey(0,j,0)=0.5*(Zy1(1,j,0)+Xy1(0,j,1))
          &+cxz*(0.5*(Ey(0,j,1)+Ey(1,j,0))-Xy1(0,j,0))
          Ey(0,j,nz)=0.5*(Zy1(1,j,3)+Xy1(0,j,nz-1))
          &+cxz*(0.5*(Ey(1,j,nz)+Ey(0,j,nz-1))-Xy1(0,j,nz))
          Ey(nx,j,0)=0.5*(Zy1(nx-1,j,0)+Xy1(3,j,1))
          &+cxz*(0.5*(Ey(nx-1,j,0)+Ey(nx,j,1))-Xy1(3,j,0))
          Ey(nx,j,nz)=0.5*(Zy1(nx-1,j,3)+Xy1(3,j,nz-1))
          &+cxz*(0.5*(Ey(nx-1,j,nz)+Ey(nx,j,nz-1))-Xy1(3,j,nz))
      enddo
```

```
c parallel to z-axis
      do k=0,nz-1
      Ez(0,0,k)=0.5*(Xz1(0,1,k)+Yz1(1,0,k))
     &+cxy*(0.5*(Ez(0,1,k)+Ez(1,0,k))-Yz1(0,0,k))
     Ez(nx,0,k)=0.5*(Xz1(3,1,k)+Yz1(nx-1,0,k))
     &+cxy*(0.5*(Ez(nx-1,0,k)+Ez(nx,1,k))-Yz1(nx,0,k))
     Ez(0,ny,k)=0.5*(Xz1(0,ny-1,k)+Yz1(1,3,k))
     &+cxy*(0.5*(Ez(0,ny-1,k)+Ez(1,ny,k))-Yz1(0,3,k))
     Ez(nx,ny,k)=0.5*(Xz1(3,ny-1,k)+Yz1(nx-1,3,k))
     &+cxy*(0.5*(Ez(nx,ny-1,k)+Ez(nx-1,ny,k))-Yz1(nx,3,k))
      enddo

c Now that we've calulated the n+1 time step value on all surfaces,
c copy these into the arrays that store the previous time steps :
      do 400 j=0,ny
      do 400 k=0,nz
      Xy0(0,j,k)=Xy1(0,j,k)
      Xy0(1,j,k)=Xy1(1,j,k)
      Xy0(2,j,k)=Xy1(2,j,k)
      Xy0(3,j,k)=Xy1(3,j,k)
      Xy1(0,j,k)=Ey(0,j,k)
      Xy1(1,j,k)=Ey(1,j,k)
      Xy1(2,j,k)=Ey(nx-1,j,k)
      Xy1(3,j,k)=Ey(nx,j,k)
      Xz0(0,j,k)=Xz1(0,j,k)
      Xz0(1,j,k)=Xz1(1,j,k)
      Xz0(2,j,k)=Xz1(2,j,k)
      Xz0(3,j,k)=Xz1(3,j,k)
      Xz1(0,j,k)=Ez(0,j,k)
      Xz1(1,j,k)=Ez(1,j,k)
      Xz1(2,j,k)=Ez(nx-1,j,k)
      Xz1(3,j,k)=Ez(nx,j,k)
400   continue

      do 410 i=0,nx
      do 410 k=0,nz
      Yx0(i,0,k)=Yx1(i,0,k)
      Yx0(i,1,k)=Yx1(i,1,k)
      Yx0(i,2,k)=Yx1(i,2,k)
      Yx0(i,3,k)=Yx1(i,3,k)
      Yx1(i,0,k)=Ex(i,0,k)
      Yx1(i,1,k)=Ex(i,1,k)
      Yx1(i,2,k)=Ex(i,ny-1,k)
      Yx1(i,3,k)=Ex(i,ny,k)
      Yz0(i,0,k)=Yz1(i,0,k)
      Yz0(i,1,k)=Yz1(i,1,k)
      Yz0(i,2,k)=Yz1(i,2,k)
```

```
      Yz0(i,3,k)=Yz1(i,3,k)
      Yz1(i,0,k)=Ez(i,0,k)
      Yz1(i,1,k)=Ez(i,1,k)
      Yz1(i,2,k)=Ez(i,ny-1,k)
      Yz1(i,3,k)=Ez(i,ny,k)
410   continue

      do 420 i=0,nx
      do 420 j=0,ny
      Zx0(i,j,0)=Zx1(i,j,0)
      Zx0(i,j,1)=Zx1(i,j,1)
      Zx0(i,j,2)=Zx1(i,j,2)
      Zx0(i,j,3)=Zx1(i,j,3)
      Zx1(i,j,0)=Ex(i,j,0)
      Zx1(i,j,1)=Ex(i,j,1)
      Zx1(i,j,2)=Ex(i,j,nz-1)
      Zx1(i,j,3)=Ex(i,j,nz)
      Zy0(i,j,0)=Zy1(i,j,0)
      Zy0(i,j,1)=Zy1(i,j,1)
      Zy0(i,j,2)=Zy1(i,j,2)
      Zy0(i,j,3)=Zy1(i,j,3)
      Zy1(i,j,0)=Ey(i,j,0)
      Zy1(i,j,1)=Ey(i,j,1)
      Zy1(i,j,2)=Ey(i,j,nz-1)
      Zy1(i,j,3)=Ey(i,j,nz)
420   continue


c          !!! TOTAL VOLTAGE Vt(nt) (INPUT+REFLECTION)
      do 99 k=h1+1,h2
      Vt(nt)=Vt(nt)+Ez(ctl,opoint,k)
99    continue
         write(0,130) nt, Vt(nt)

      do 66 j=0,ny
      do 66 i=0,nx
      if(mod(nt,modmax).eq.0) then
      write(1,150) nt,i, j, Ez(i,j,h2-1)
      endif
66    continue


      t=t+dt
30    continue       ! nt (time) LOOP

c     ****************************************************
130   format(i7,f20.10)
140   format(4i7,f20.10)
150   format(3i7, f20.10)
130   format(i7,f20.10)

      close(0)
```

```
close(1)

stop
end
```