5.1 Time-domain modeling of wire antennas by method of moments

Developing Matlab

The program for the analysis of the wire dipole by the implicit approach is from the point of the implementation demanding. Firstly it is necessary to compute the excitation Gaussian pulse modulated by a harmonic signal (5.1A.7):

```
EG=E0*4/(c*T*sqrt(pi)).*exp(-(4/T)^2*(t-t0-delt/2).*(t-t0-
delt/2)).*cos(2*pi*f0*(t-delt/2));
```

Secondly the values of the coefficients $\kappa(m, n)$ and $\kappa(m^{\pm}, n^{\pm})$ are evaluated according to the relations (5.1B.5) and (5.1B.13). Further, the distances between the discretization segments, according to (5.1B.7) and (5.1B.12), are computed. We have an optional choice for this calculation. We can use either the accurate relations (5.1B.7a) and (5.1B.12a), or the approximate relations with neglecting of the radius of the antenna wire (5.1B.7b) and (5.1B.12b), if the following condition is met: $c\Delta t >> a$.

```
Rmn=zeros(Nz+1,1);
kappa=zeros(Nz+1,1); % kappa according to (5.1B.5)
for m=0:Nz % (5.1B.13)
    hlp1=log((m+0.5)*delz + sqrt(((m+0.5)*delz)^2+a^2));
    hlp2=log((m-0.5)*delz + sqrt(((m-0.5)*delz)^2+a^2));
    kappa(m+1)=(hlp1-hlp2);
    % distance of discretization elements (5.1B.7) and (5.1B.12), optional
    choice
    % if c*delt>>a is met, it is better to use (5.1B.7b) and (5.1B.12b)
    neglecting of radius of dipole (layer B)
    % R(m+1)=sqrt(a^2+(m*delz)^2); % (5.1B.7a) a (5.1B.12a)
    R(m+1)= m*delz; % (5.1B.7b) a (5.1B.12b)
end
kappaFI=1/4/pi/ep*kappa; % kappa for scalar potential (5.1B.13)
kappa=mi/4/pi*kappa; % kappa for vector potential (5.1B.5)
```

Further, we evaluate the values of the matrix elements in (5.1B.35), i.e. $[A_{11C}(m, n)]$, $[A_{12C}(m,n)]$ according to the relations (5.1B.35) and (5.1B.22) for the vector potential and $[\varphi_{11C}(m, n)]$, $[\varphi_{12C}(m,n)]$ and $[\varphi_{13C}(m, n)]$ according to the relations (5.1B.36), (5.1B.27b) and (5.1B.29) for the scalar potential. Here it is necessary to notice that the elements of these matrixes concern currents for time $t_{Rk}>t_{k-1}$. In case of the matrixes of the vector potential $[A_{11C}(m,n)]$, $[A_{12C}(m, n)]$, the source code can be written:

```
A11C=zeros(N,N); % coefficients of vector potential (5.1B.35) and (5.1B.22)
A12C=zeros(N,N);
for m=1:N,
    for n=1:N,
        mn=abs(m-n); % consider terms at time tk>=tRk>tk-1
        if(R(mn+1)<cdelt) % distance has to be smaller than c*delt
            A11C(m,n)=A11C(m,n)+(1-R(mn+1)/cdelt)*kappa(mn+1);
            A12C(m,n)=A12C(m,n)+R(mn+1)/cdelt*kappa(mn+1);
        end
    end
end
```

In case of the matrixes of the scalar potential $[\varphi_{11C}(m, n)]$, $[\varphi_{12C}(m,n)]$ and $[\varphi_{13C}(m, n)]$ only a part of the source code for the calculation of the contribution of the first term with indexes (m^+, n^+) in (5.1B.36) will be shown. The calculation of the other terms with the indexes indexi (m^+, n^-), (m^-, n^+) and $(m^-, n$-) is analogical:

```
FI11C=zeros(N,N); % coefficients of scalar potential (5.1B.36)
FI12C=zeros(N,N); % for ++ according to (5.1B.27b) and (5.1B.29);
FI13C=zeros(N,N); % for other combinations analogically

for m=1:N,
    for n=1:N,
        mn=abs(m-n); % computing of distance indexes between elements
        mnPP=mn; % P is +; M is -
        mnPM=mn+1;
        mnMP=abs(mn-1);
        mnMM=mn;
        if(m<n)
            hlp=mnPM;
            mnPM=mnMP;
            mnMP=hlp;
        end
        % terms contribute if the following condition is met: tRk>tk-1
        if(R(mnPP+1)<cdelt)
            FI11C(m,n)=FI11C(m,n) -delt/delz/2*(1-R(mnPP+1)
            /cdelt)^2*kappaFI(mnPP+1); %(5.1B.29a)
            FI12C(m,n)=FI12C(m,n) -delt/delz/2*(1-(R(mnPP+1)
            /cdelt)^2)*kappaFI(mnPP+1); %(5.1B.29b)
            FI13C(m,n)=FI13C(m,n)-1/delz*kappaFI(mnPP+1); %(5.1B.27b)
        end
        .
        .
        .
    end
end
```

Since the inverse matrix in (5.1B.35) does not depend on time, we evaluate it before the time loop:

```
invA11CFI11C=inv(A11C+delt/delz/2*FI11C); % inverse matrix (5.1B.35)
```

Finally we implement the time loop where we compute from the known time "samples" of the vector and scalar potential the future samples of these quantities. Note that the values of currents, or the values of the integrals of currents, which do not fit to discrete time instants, are between the time instants linearly approximated. The integral of currents, which is computed according to the trapezoid rule, is stored in the variable *IS*. The meaning of the other symbols should be apparent. Again in case of the calculation of the scalar potential, the source code is shown only for the first term with indexes (m^+, n^+) (see (5.1B.34) and (5.1B.36)), the calculation of the contribution of the other terms with indexes (m^+, n^-), (m^-, n^+) and (m^-, n^-) is analogical.

```
% TIME LOOP
for k=3:Nt,
    A2=zeros(N,1); % (5.1B.19b)
    FI2=zeros(N,1); % (5.1B.36d), (5.1B.25b) and further analogically for
    +-,-+,--
    for m=1:N,
        for n=1:N,
            % computing of distance indexes between elements
            mn=abs(m-n); % vector potential
            mnPP=mn; % scalar potential
            mnPM=mn+1; % P is +; M is -
            mnMP=abs(mn-1);
            mnMM=mn;
            if(m<n)
                hlp=mnPM;
                mnPM=mnMP;
                mnMP=hlp;
            end
            % further the terms are computed if: tRk<=tk-1
            if(R(mn+1)>=cdelt) % vector potetntial (5.1B.19b)
                tk=k-R(mn+1)/cdelt;
                if(tk>1)
                    ftk=floor(tk); % linear approximation of current
                    In=(I(n,ftk+1)-I(n,ftk))*(tk-ftk)+I(n,ftk);
                    A2(m,1)=A2(m,1)+In*kappa(mn+1);
                end
            end
            if(R(mnPP+1)>=cdelt) % scalar potential ++; (5.1B.36d),
            (5.1B.25b)
                tk=k-R(mnPP+1)/cdelt;
                if(tk>1)
                    ftk=floor(tk); % linear approximation of integral of
                    current
                    ISn=(IS(n,ftk+1)-IS(n,ftk))*(tk-ftk)+IS(n,ftk);
                    FI2(m,1)=FI2(m,1)+ISn/delz*kappaFI(mnPP+1);
                end
            end
            .
            .
            .
        end
    end
    E(nap,1)=EG(k); % excitation vector is non-zero at the location of
    feeding (5.1B.7)
    % (5.1B.35)
    I(:,k)=invA11CFI11C*(delt*E-A12C*I(:,k-1)-A2+A -delt/2/delz*
    (FI12C*I(:,k-1)+FI13C*IS(:,k-1)+FI2+FI)); % integration of current with
    respect to time (trapezoid rule) (5.1B.15)
    IS(:,k)=(I(:,k)+I(:,k-1))*delt/2+IS(:,k-1); % preparation for the next
    step
    A=A11C*I(:,k)+A12C*I(:,k-1)+A2; % (5.1B.18) a (5.1B.21)
    FI=FI11C*I(:,k)+FI12C*I(:,k-1)+FI13C*IS(:,k-1)+FI2; %
    (5.1B.30)-(5.1B.33)
    % considering (5.1B.26)-(5.1B.29)
end
```