Electromagnetic inverse scattering problem

Sofranac **Martina**

17.07.2009.

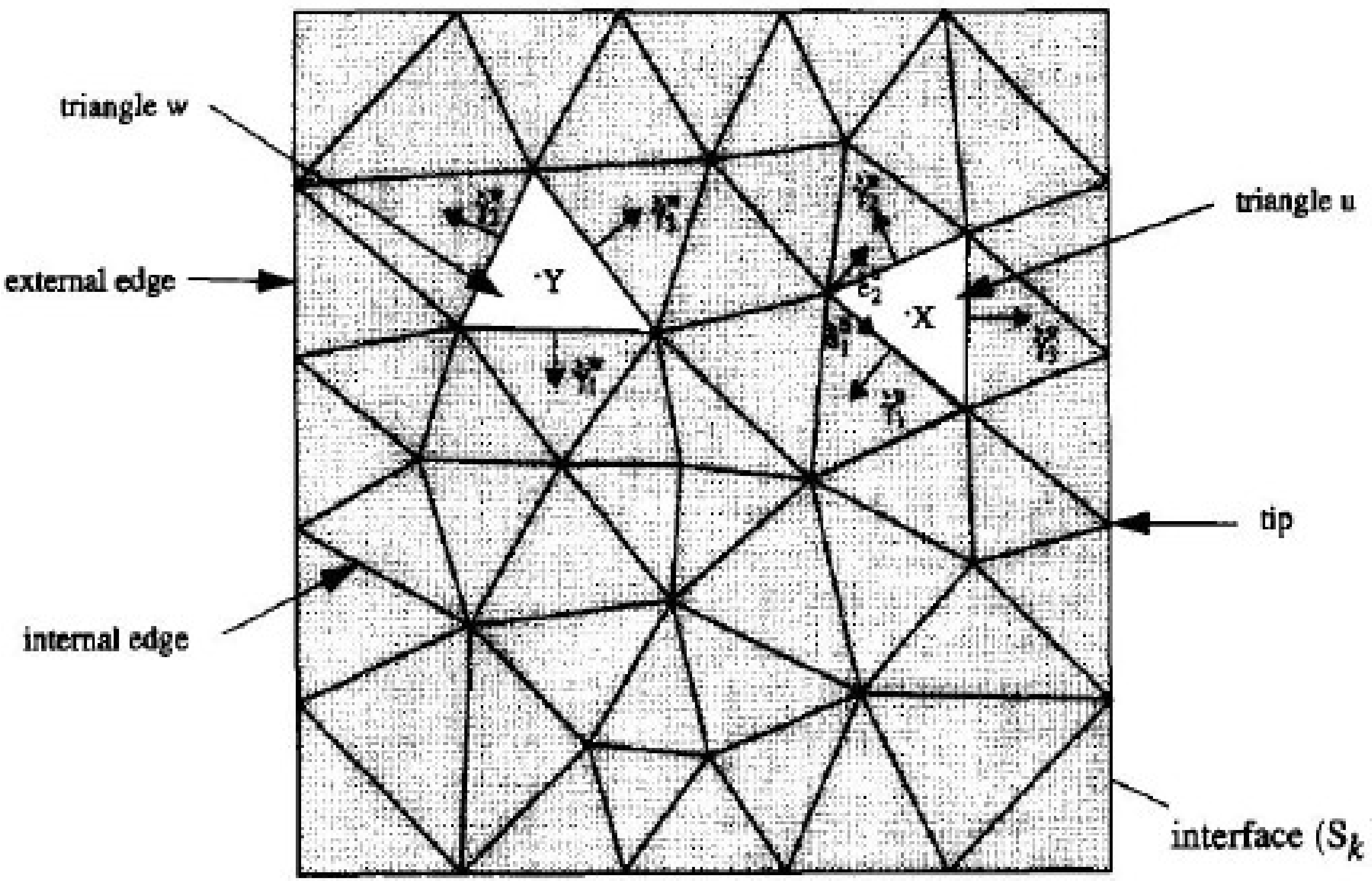
Green Representation for Electromagnetic field :

$$\vec{E}_k(x) = \vec{E}_k^i + i\omega\mu \oint\limits_{(S_k)} G(x,y)\vec{j}_k(y)ds(y) - \frac{1}{i\omega\epsilon} \oint\limits_{(S_k)} grad_x G(x,y) div_{S_k}\vec{j}_k(y)ds(y) - rot \oint\limits_{(S_k)} G(x,y)\vec{m}_k(y)ds(y)$$

$$\vec{H}_k(x) = \vec{H}_k^i + i\omega\epsilon \oint\limits_{(S_k)} G(x,y)\vec{m}_k(y)ds(y) - \frac{1}{i\omega\mu} \oint\limits_{(S_k)} grad_x G(x,y) div_{S_k}\vec{m}_k(y)ds(y) - rot \oint\limits_{(S_k)} G(x,y)\vec{j}_k(y)ds(y)$$

Boundary conditions:

$$\vec{n}_k \times \vec{E}_k + \vec{n}_l \times \vec{E}_l = 0 = \vec{m}_k + \vec{m}_l$$
$$\vec{n}_k \times \vec{H}_k + \vec{n}_l \times \vec{H}_l = 0 = \vec{j}_k + \vec{j}_l$$



triangle w
external edge
internal edge
triangle u
tip
interface (S_k)
Y
X

Rumsey's reaction concept

- The reaction concept between $\{\vec{j}_l, \vec{m}_l\}$ and $\{\vec{j}^t_l, \vec{m}^t_l\}$ is defined by:

$$\boxed{R_{S_l}(\{\vec{j}_l, \vec{m}_l\}, \{\vec{j}^t_l, \vec{m}^t_l\}) = \oint_{S_l} \vec{E}_l(x) \cdot \vec{j}^t_l(x) - \vec{m}^t_l(x) \cdot \vec{H}_l(x)\, ds^t(x)}$$

- This concept can be applied to $\{\vec{E}^i_l, \vec{H}^i_l\}$ if sources are present in Ω_l:

$$\boxed{R_{S_l}(\{\vec{j}^a_l, \vec{m}^a_l\}, \{\vec{j}^t_l, \vec{m}^t_l\}) = \oint_{S_l} \vec{E}^i_l(x) \cdot \vec{j}^t_l(x) - \vec{m}^t_l(x) \cdot \vec{H}^i_l(x)\, ds^t(x)}$$

Variational formulation of integral equations

- Using the boundary conditions, the total reaction on all domains is expressed as:

$$\boxed{\sum_{l=1}^{N} R_{S_l}(\{\vec{j}_l, \vec{m}_l\}, \{\vec{j}^t_l, \vec{m}^t_l\}) + R_{S_l}(\{\vec{j}^a_l, \vec{m}^a_l\}, \{\vec{j}^t_l, \vec{m}^t_l\}) = 0}$$

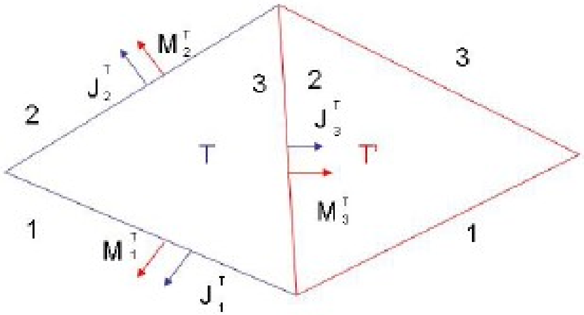
Variational formulation of our problem

$$\sum_{l=1}^{N}\mu_{rl}Q_{Sl}(S_t,j_l,j_l^t)+\frac{k_l^2}{\mu_{rl}}Q_{sl}(S_t,m_l,m_l^t)-P_{sl}(S_t,j_l,m_l^t)-P_{sl}(S_t,m_l,j_l^t)=-\sum_{l=1}^{N}\oint_{Sl}(E_l^i(x)j_l^t-H_l^i(x)m_l^t)ds(x)$$

where P_{sl} and Q_{sl} are deffined as :

$$Q_{Sl}(S_t,j_l,j_l^t)=\oint_{S^t}\oint_{S^l}G(k,x,y)(j(y)j^t(x)-\frac{1}{k_l^2}div_S j(y)div_{S^t}j^t(x))ds(x)ds^t(y)$$

$$P_{Sl}(S_t,j_t,m_l^t)=\oint_{S^t}\oint_{S^l}[grad(G(k,x,y)\times j(y))]p^t(x)ds(y)ds^t(x)$$



$$-J_2^{T'} = J_3^T = J_k$$

Currents of the triangle T can be expressed as it is written:

$$\overrightarrow{j^T} = \overrightarrow{j_1^T} \cdot J_1^T + \overrightarrow{j_2^T} \cdot J_2^T + \overrightarrow{j_3^T} \cdot J_3^T$$

$$\overrightarrow{m^T} = M_1^T \cdot \overrightarrow{m_1^T} + M_2^T \cdot \overrightarrow{m_2^T} + M_3^T \cdot \overrightarrow{m_3^T}$$

Bilinear form is tranformed into descrete one :

$$\sum_{l=1}^{N}\sum_{u=1}^{NE_1}\sum_{w=1}^{NE_1}\mu_{rl}A_l^{uw}(\vec{j_l}^{u},\vec{j_l}^{tw})+\frac{k_l^2}{\mu_{rl}}A_l^{uw}(\vec{p_l}^{u},\vec{p_l}^{tw})-B_l^{uw}(\vec{j_l}^{u},\vec{j_l}^{tw})-B_l^{uw}(\vec{p_l}^{u},\vec{p_l}^{tw})=\sum_{l=1}^{N}\sum_{w=1}^{NE_1}C_l^{w}(\vec{E_l}^{i},\vec{H_l}^{i},\vec{j_l}^{tw},\vec{p_l}^{tw})$$

Matrix Formulation of problem : Ax=C

Integral solver in comparison with Differential solvers :

Matrix A is dense , contains N*N block

Much more storage space

More CPU time

Hermite interpolation

Hermite(Osculating) polinomials are generalization of both the Taylor and Langrangian polynomials. If we have given n+1 points x_0, x_1, ... , x_n and nonnegative integers m_0, ... Hermite polynomial approximating a function f is ploynomial of at least degree of $m_i$at point $x_i$where f belongs to $C^m(a, b)$ and m= max$\{m_0, ...m_n\}$ and x_i belongs to [a,b] for every i = 0, ... n.

The degree of this osculating polynomial will be at most :

$M = \sum_i m_i + n.$

The number of conditions to be satisfied is $\sum_i m_i + (n + 1)$ and a polynomial of degree M has M+1 coefficients that has to satisfy these conditions.

By definition, if we have n+1 distinct numbers in range [a,b] and $m_i$are nonnegative integers associated with x_i, so for each i = 0, ... , n .

If m= $\underset{0 \le i \le n}{max}\, m_i$ and f$\in C^m[a, b]$

so Hermite polynomial approximating f is the polynomial P of least degree m such that

$\frac{d^k P(x_i)}{dx^k} = \frac{d^k f(x_i)}{dx^k}$ for each i =0, 1, ... , n and k=0,1, ... , m_i.

If the function f belongs to $C^1[a,b]$and x_0, x_1, ... , x_n that are in range [a,b] are distinct , tha unique polynomial of least degree matching with f and f' at x_0, x_1, ... , x_n is the polynomial of degree at most M=2n+1 given as :

$$H_{2n+1}(x) = \sum_j f(x_j) H_{n,j}(x) + \sum_j f'(x_j) \hat{H}_{n,j}(x)$$

where :

$$H_{n,j}(x) = [1 - 2(x - x_j) L'_{n,j}(x_j)] L^2_{n,j}(x)$$

and

$$\hat{H}_{n,j}(x) = (x - x_j) L^2_{n,j}(x)$$

here $L_{n,j}(x)$denotes Langrange coefficient polynomial of degree n that si defined by formula :

$$L_{n,k}(x) = \frac{(x - x_0)....(x - x_{k-1})(x - x_{k+1})....(x - x_n)}{(x_k - x_0)....(x_k - x_{k-1})(x_k - x_{k+1})....(x_k - x_n)}$$

k-th divided difference with respect to $x_i, x_{i+1}, ..., x_{i+k}$ will be :

$$f[x_i, x_{i+1, ...,} x_{i+k}] = \frac{f[x_{i+1}, ..., x_{i+k}] - f[x_i, x_{i+1} ..., x_{i+k-1}]}{x_{i+k} - x_i}$$

Using Newton divided differences we can simplify programming of interpolation

$$E(\varepsilon) = c_5\varepsilon^5 + c_4\varepsilon^4 + c_3\varepsilon^3 + c_2\varepsilon^2 + c_1\varepsilon + c_0$$

Newton divided differences

x	f(x)	First divided differnces	Second divided differences	Third divided differences
x_0	$f[x_0]$			
		$f[x_0, x_1] = \frac{f[x_1]-f[x_0]}{x_1-x_0}$		
x_1	$f[x_1]$		$f[x_0, x_1, x_2] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$	
		$f[x_1, x_2] = \frac{f[x_2]-f[x_1]}{x_2-x_1}$		$f[x_0,x_1,x_2,x_3] = \frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$
x_2	$f[x_2]$		$f[x_1, x_2, x_3] = \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	
		$f[x_2, x_3] = \frac{f[x_3]-f[x_2]}{x_3-x_2}$		$f[x_1,x_2,x_3,x_4] = \frac{f[x_2,x_3,x_4]-f[x_1,x_2,x_3]}{x_4-x_1}$
x_3	$f[x_3]$		$f[x_2, x_3, x_4] = \frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2}$	
		$f[x_3, x_4] = \frac{f[x_4]-f[x_3]}{x_4-x_3}$		$f[x_0,x_1,x_2,x_3] = \frac{f[x_3,x_4,x_5]-f[x_2,x_3,x_4]}{x_5-x_2}$
x_4	$f[x_4]$		$f[x_3, x_4, x_5] = \frac{f[x_4,x_5]-f[x_3,x_4]}{x_5-x_3}$	
		$f[x_4, x_5] = \frac{f[x_5]-f[x_4]}{x_5-x_4}$		
x_5	$f[x_5]$			

z	f(z)	First divided differnces	Second divided differences
$z_0 = x_0$	$f[z_0] = f[x_0]$		
		$f[z_0, z_1] = f'(x_0)$	
$z_1 = x_0$	$f[z_1] = f[x_0]$		$f[z_0, z_1, z_2] = \frac{f[z_1,z_2]-f[z_0,z_1]}{z_2-z_0}$
		$f[z_1, z_2] = \frac{f[z_2]-f[z_1]}{z_2-z_1}$	
$z_2 = x_1$	$f[z_2] = f[x_1]$		$f[z_1, z_2, z_3] = \frac{f[z_2,z_3]-f[z_1,z_2]}{z_3-z_1}$
		$f[z_2, z_3] = f'(x_1)$	
$z_3 = x_1$	$f[z_3] = f[x_1]$		$f[z_2, z_3, z_4] = \frac{f[z_3,z_4]-f[z_2,z_3]}{z_4-z_2}$
		$f[z_3, z_4] = \frac{f[z_4]-f[z_3]}{z_4-z_3}$	
$z_4 = x_2$	$f[z_4] = f[x_2]$		$f[z_3, z_4, z_5] = \frac{f[z_4,z_5]-f[z_3,z_4]}{z_5-z3}$
		$f[z_4, z_5] = f'(x_2)$	
$z_5 = x_2$	$f[z_5] = f[x_2]$		

$$H_5 = f[z_0] + \sum_{k=1}^{2n+1} f[z_0, ..., z_k](x - z_0)(x - z1)...(x - z_{k-1})$$

$$H(x) = Q_{0,0} + (x - x_0)Q_{11} + Q_{2,2}(x - x_0)^2 + Q_{3,3}(x - x_0)(x - x_1) + Q_{4,4}(x - x_0)^2(x - x_1)^2 + Q_{5,5}(x - x_2)(x - x_0)^2(x - x_1)^2$$

This polynomial is transformed in following form:

$$E(\varepsilon) = c_5\varepsilon^5 + c_4\varepsilon^4 + c_3\varepsilon^3 + c_2\varepsilon^2 + c_1\varepsilon + c_0$$

Searching for all possible values for which P(ϵ)=0

Searching for zeros is equivavelent to problem where we want to calculate eigenvalues of companion matrix :

$$A = \begin{pmatrix} -\frac{c_4}{c_5} & -\frac{c_3}{c_5} & -\frac{c_2}{c_5} & -\frac{c_1}{c_5} & -\frac{c'}{c_5} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

QR method for computing eigenvalues

- Gram-Schmidt
- Given rotation
- Householder transformations
- …

Classical Gram -Shmidt

$$u_1 = a_1 \text{ and } q_1 = \frac{u_1}{||u_1||}$$

$$u_k = a_k - \sum_{j=1}^{k-1} proj_{u_j} a_k \qquad q_k = \frac{u_k}{||u_k||}$$

In k-th step we first decompose A and then we calculate A in next iteration

$$A^k = Q * R$$

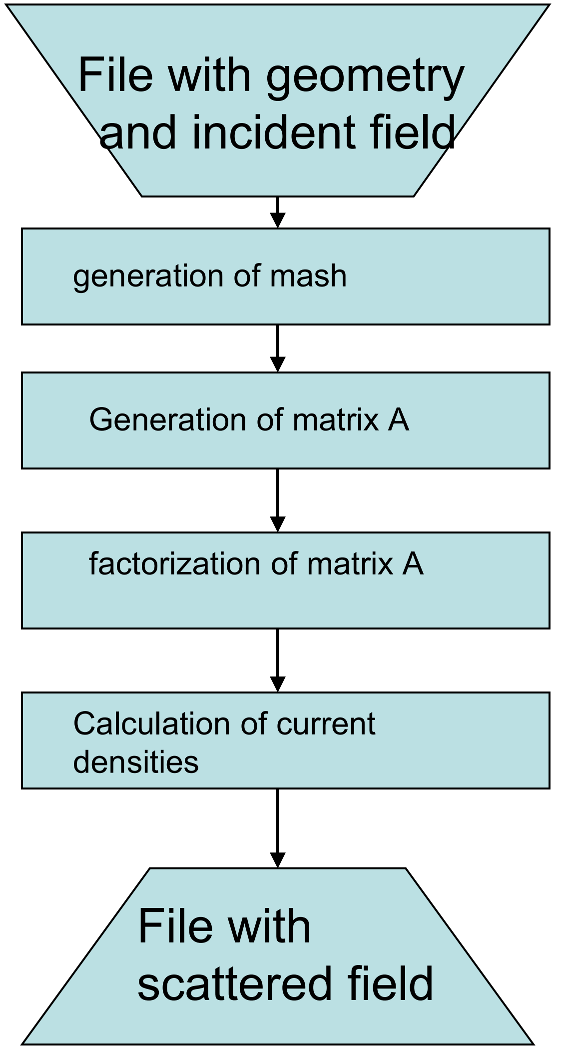
$$A^{k+1} = R * Q$$

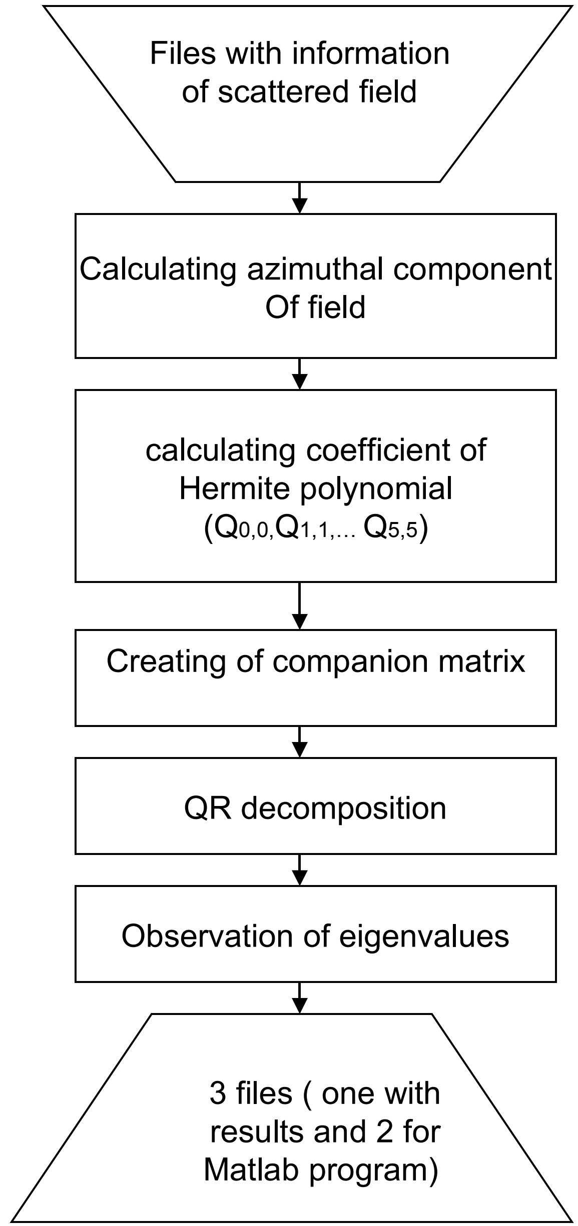
After certain number of iteration we will have convergence

Modified Gram-Schmidt

```
do k=1,5

   r(k,k)=sqrt(dot_product(A(1:5,k),A(1:5,k)))
   q(1:5, k) = A(1:5, k) / r(k,k);

            do j = k+1,5
            r(k, j) = dot_product(q(1:5, k), A(1:5, j));
            A(1:5, j) = A(1:5, j) - r(k, j) * q(1:5, k);
            end do

 end do
```

Structure of SR3D




Files with information of scattered field
Calculating azimuthal component Of field
calculating coefficient of Hermite polynomial ($Q_{0,0}$,$Q_{1,1}$,... $Q_{5,5}$)
Creating of companion matrix
QR decomposition
Observation of eigenvalues
3 files (one with results and 2 for Matlab program)

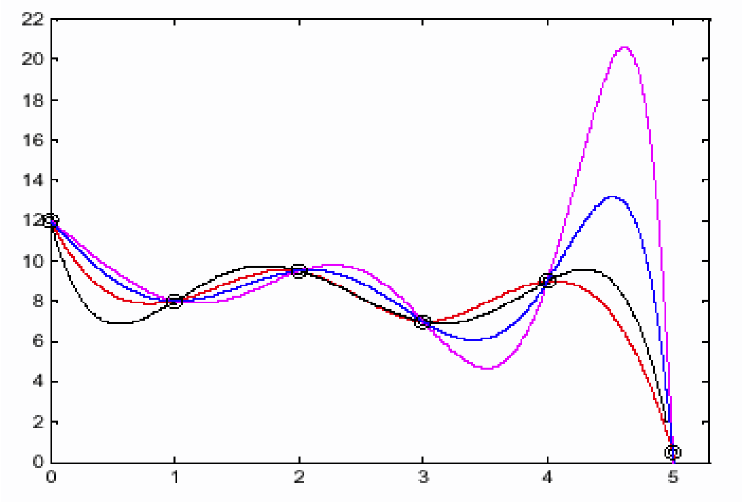
Observation of eigenvalues

We have to check if we have 2 X 2 blocks and to flag corresponding elements

Checking if real eigenvalues are in corresponding range

If its not flaged and its in range (min ϵ, max ϵ) it is written in file results.txt

Various ways for aproximation of function



If only values of function are known:

Least-square method,
Polynomial and trigonometric functions,
Taylor series and Chebyshev approximations,
Piecewise polynomial functions,
Splines, cubic splines, B-splines functions,
Rational functions, Padé approximations,
Thiele interpolations,
FFT, Neural networks, Genetic algorithms
and so on.

If we have data about values and derivatives of function:

Splines, cubic splines, B-splines ...
Piecewise polynomial functions
Neural networks, Genetic algorithms