In **EH3D,** we have to solve:

$$\begin{pmatrix} L - i\omega M_\sigma & -i\omega M_\sigma G \\ -D M_\sigma & -D M_\sigma G \end{pmatrix} \begin{pmatrix} A \\ \phi \end{pmatrix} = \begin{pmatrix} i\omega q \\ D q \end{pmatrix}$$

$$L = C_e M_\mu^{-1} C_f - G \mu^{-1} D$$
$$\omega = 2\pi f$$

G is the gradient operator, D is the divergence, C_f is the curl from cell faces to edges, and C_e is the curl from cell edges to faces. q is the source.

M_σ is the diagonal matrix consisting of average conductivities on cell faces.

$$M_\sigma = \left(\frac{v_{i-1}\sigma_{i-1}^{-1} + v_i \sigma_i^{-1}}{v_{i-1} + v_i} \right)^{-1}$$

M_μ is the diagonal matrix consisting of average permeabilities (μ) on cell edges.

$$M_\mu = \left(\frac{v_{i-1,j-1}\mu_{i-1,j-1} + v_{i-1,j}\mu_{i-1,j} + v_{i,j-1}\mu_{i,j-1} + v_{i,j}\mu_{i,j}}{v_{i-1,j-1} + v_{i-1,j} + v_{i,j-1} + v_{i,j}} \right)$$

The complex conductivity σ is:

$$\sigma = \sigma_r - (\sigma_i + \omega \epsilon_0) i$$
$$\epsilon_0 = 8.85418781762 \times 10^{-12}$$

$$\mu = \mu_0 (1+\kappa) = \mu_0 \mu_r \ , \quad \mu_0 = 4\pi 10^{-7}$$

Then, to get *E* and *H* (and optionally *J*) :

$$E = A + G\phi$$
$$H = \frac{1}{i\omega} M_\mu^{-1} C_f E$$
$$J = M_\sigma E$$

The matrix system gets solved using *bicgstab*.

$$A x = b$$

The solver stops when the maximum number of iterations is reached or when the residual is less than the user-specified tolerance.

$$\frac{\| Ax - b \|}{\| b \|} < \varepsilon$$

Preconditioners

$$A\ x\ =\ b$$

$$\begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

The following preconditioners are used.

0 SSOR
L = lower diagonal matrix of *A* $\times$ 1/square root of the diagonal.
U = 1/square root of the diagonal $\times$ upper diagonal matrix of *A*.
Solve *(LU) x = b*

1 ILU
Incomplete LU of the real values of A_1 and A_4 .

3 Full complex ILU (**NEW**)
Incomplete LU of the entire matrix.

4 BLUGS (**NEW**)
Get ILU of A_1
Get ILU of A_4
Solve for x_2: $A_4\ x_2 = b_2$
Solve for x_1: $A_1\ x_1 = b_1\ -\ A_2\ x_2$
Note that this is the slowest preconditioner, however, it can reduce the residual in fewer iterations.

Changes to EH3D

- Two new preconditioners.
    1. Full complex ILU (preconditioner type = 3)
    2. BLUGS (preconditioner type = 4)
- When `NO_IMAG_COND` is specified instead of imaginary conductivity in the input file, only real conductivity will be used (no $(\sigma_i + \omega\epsilon_0)\ i$ term).
- EH3D checks every 5 iterations to see if a file `EH3D_PAUSE` exists in the current directory. If the file is found, EH3D pauses, and the user is given an option to quit EH3D or restart with a different preconditioner. If the “Quit” option is chosen, the solution corresponding to the smallest residual is written out and the program exits. (If the program is restarted, don't forget to delete this file, otherwise, it'll pause again after 5 iterations.)
- A file `bicgstab_out.txt` is output. It contains the residual after each iteration of *bicgstab*.

July 2007

The input file to EH3D, "`eh3d.inp`" has a new format:

```
10.          ! frequency
mesh.txt     ! mesh file
1block.con   ! real conductivity file | halfspace value
0.0          ! imaginary conductivity file | halfspace value
0.0          ! susceptibility file | halfspace value
trx.txt      ! transmitter file
0.01 1e-08 1000   ! droptol, tol, iter
1            ! preconditioner type
0            ! 0=single, 1=double precision output
0            ! 0=binary, 1=ascii output
1 1 0        ! 1=output  E, H or J files
```

Trasmitter file

The first line of the transmitter file has a label indicating the type of transmitter.
Valid values for the label are:
`TRX_ORIG  TRX_MAGNETIC_DIPOLE  TRX_LOOP  TRX_LINES  TRX_SECONDARY  TRX_COR_SRC`

The format of the rest of the file depends on what the label is.

```
TRX_MAGNETIC_DIPOLE
x y z  theta alpha  m
```
To have loop parallel to x-y plane, set `theta=alpha`=0 .

```
TRX_LOOP
x y z  radius  theta alpha  I
```
To have loop parallel to x-y plane, set `theta=alpha`=0 .

```
TRX_LINES
n
x1 y1 z1
x2 y2 z2
:
xn yn zn
```

```
TRX_ORIG
n
x1 y1 z1
x2 y2 z2
:
xn yn zn
```

```
TRX_SECONDARY | TRX_COR_SRC
background real conductivity file
background imaginary conductivity file
E primary E file
J primary J file
H primary H file
```

At least one of the field files should be supplied. *E* and *J* files cannot be supplied at the same time.
When *J* is supplied, primary *E* is calculated: $E_p = M_{\sigma b}^{-1} J_p$

When only *E* (or *J*) is supplied, primary *H* is calculated: $H_p = \frac{1}{i\omega} M_\mu^{-1} C_f E_p$

When only *H* is supplied, primary *E* is calculated: $E_p = M_{\sigma b}^{-1} C_e H_p$

When the header is `TRX_SECONDARY`, the source term is: $s = \left(M_\sigma - M_{\sigma b}\right) E_p$
The system is then solved for secondary fields.

When the header is `TRX_COR_SRC`, the source term is: $s = C_e H_p - M_{\sigma b} E_p$

Dec. 2007

The code was modified to read in the model file with imaginary susceptibilities. It is entered in the input file after the real susceptibility file.

```
0.0     ! real susceptibility file | halfspace value | null
0.0     ! imaginary susceptibility file | halfspace value | null
```

The complex permeability is then calculated.
$\mu = \mu_0 \left((1+\kappa_r) + i\kappa_i \right) = \mu_0 \mu_r$

Feb. 2008

The code can now handle multiple frequencies, and interpolate data at receiver locations.

The first line of the input file can have a single frequency, and the code will behave as before. Or, several frequencies can be specified:
`N_FREQ` *n*
f1 f2 ... fn

For example, to run the code for frequencies of 10Hz, 50Hz, and 100Hz, the first two lines of the input file should be:
```
N_FREQ 3
10.  50.  100.
```

If the last line of the input file is a three-column receiver file, the fields will be interpolated at these locations and output to a file `eh3d_recv.txt`. This is the equivalent of running **ehpoints**. If the interpolated values are not required, set this field to `NONE`.

Code is parallelized with OpenMP. When more than one frequency is specified, they'll be solved in parallel on multi-cpu (core) systems. To set the number of threads to use (*p*), the environment variable `OMP_NUM_THREADS` needs to be modified. The default is the number of processors available.

When using Windows. (Make sure there are no spaces around the "=" sign.)
`set OMP_NUM_THREADS=`*p*

When using Linux csh:
`setenv OMP_NUM_THREADS` *p*

When *p* is set to 1, the systems will be solved sequentially.

There should now be a model file for each frequency. The following is the input file "eh3d.inp" format:

```
N_FREQ nf        ! nf is the # of frequencies
f1 f2 ... fnf    ! frequencies
mesh.txt         ! mesh file
r_con1 r_con2 ... r_connf  ! real      conductivity   files | halfspace values
i_con1 i_con2 ... i_connf  ! imaginary conductivity   files | halfspace values
r_sus1 r_sus2 ... r_susnf  ! real      susceptibility files | halfspace values
i_sus1 i_sus2 ... i_susnf  ! imaginary susceptibility files | halfspace values
trx.txt              ! transmitter file
0.01 1e-10 1000      ! droptol, tol, iter
3                    ! preconditioner type
0                    ! 0=single, 1=double precision output
0                    ! 0=binary, 1=ascii output
1 0 0                ! 1=output  E, H or J files
receiver file | NONE
```

If the same file needs to be repeated several times, "n*" can be used. For example, if the same conductivity file is used for 5 frequencies, instead of entering it 5 times, the following can be entered:

```
5*model.con
```

Or, for a constant halfspace of 0, the following can be entered:

```
5*0.0
```