Comparing results of harmonic pre-stressed simulation

Laura Del Tin, 2006, email: ldeltin@deis.unibo.it

Loading functions

You need to change the path to the right one

```
<< path/Post4MOR.m
```

Setting directory

You may need this to your current directory

```
SetDirectory["."]
```

Making output matrix from EMAT file

```
ComputeCurrentOutput[matE_, matC_, nvolt_Integer] :=
 Module[{mask1, mask2, outVec, output1, maskDirichlet, maskDelete, out, output},
  mask1 = Table[1, {Length[matC]}];
  mask2 = Table[1, {nvolt}];
  outVec = mask2.Take[matC, nvolt];
  output1 = outVec.matE;
  maskDirichlet = mask1.matC;
  maskDelete =
    (i = 0; out = {}; Scan[(++i; If[# == 1, AppendTo[out, {i}]]) &, maskDirichlet]; out);
  output = Delete[output1, maskDelete];
  {SparseArray[{output}], outVec.output1}
 ]

matE = Import["emat.E", "MTX"]
matC = Import["emat.C", "MTX"]
```

```
SparseArray[<440>, {5985, 5985}]
```

```
SparseArray[<468>, {468, 5985}]
```

We assume that useful voltages are at at the beginning
Number of voltages is equal to numberr of trans element created by emtgen macro,
which can be read in the _emtgen.out file

```
nvolt = 55
```

```
55
```

Compute C matrix for current output and displacement current due to applied alternate voltage (negleting the multiplying factor $i2\pi\omega$):

```
{matCadd, voltcontrib} = ComputeCurrentOutput[matE, matC, nvolt]
```

```
{SparseArray[<55>, {1, 5517}], 0.000723302}
```

Making system matrices from FULL file after harmonic simulation

omega=0 => Re[tangent stiffness matrix]=K

```
matK = Import["mat0.K", "MTX"]
```

```
SparseArray[<328800>, {5517, 5517}]
```

omega=1e13

```
mat1e13re = Import["mat1e13.K", "MTX"]
```

```
SparseArray[<274813>, {5517, 5517}]
```

Compute M:
mat1e13=K-ω^2M

```
matM = (matK - mat1e13re) / (1*^26)
```

```
SparseArray[<328800>, {5517, 5517}]
```

Create the complete full system and add current d.o.f. to mechanical d.o.f.

```
matB = Import["mat0.B", "MTX"]
matC = Import["mat0.C", "MTX"]
matCnames = Import["mat0.C.names", "list"]
```

```
SparseArray[<55>, {5517, 2}]
```

```
SparseArray[<1>, {1, 5517}]
```

```
{centerUY}
```

```
sysfull = MakeDynamicSystem[matM, Null, matK, matB, matC, matCnames, Null]
```

```
DynamicSystem[{5517,2,1}, ...]
```

```
sys2out = AddOutputs[sysfull, matCadd, {"current"}]
```

```
DynamicSystem[{5517,2,2}, ...]
```

```
WriteSystem[sys2out, "full"]
```

Testing reduced system

Loading the reduced system:

```
sys = ReadSystem["mor"]
```

```
DynamicSystem[{30,2,2}, ...]
```

Loading ANSYS results:

```
ansys = ReadResult["results.txt"]
```

```
- SimulationResult -
```

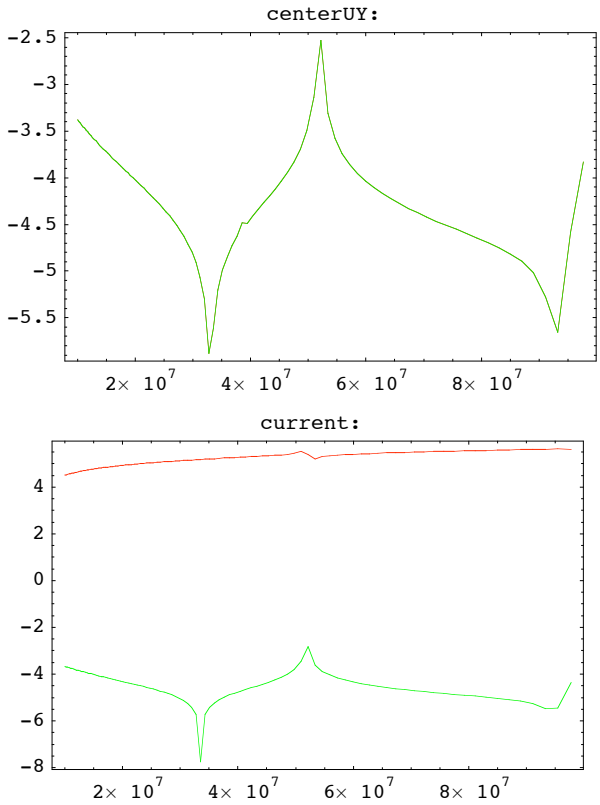
Make harmonic simulation of reduced system:

```
res = HarmonicSolution[XSeries[ansys], sys]
```

```
- SimulationResult -
```

Compare results:

```
PlotResult[{ansys, res}, FunctionY → (Log10[Abs[#]] &),
  PlotStyle → {RGBColor [1, 0, 0], RGBColor[0, 1, 0]}];
```

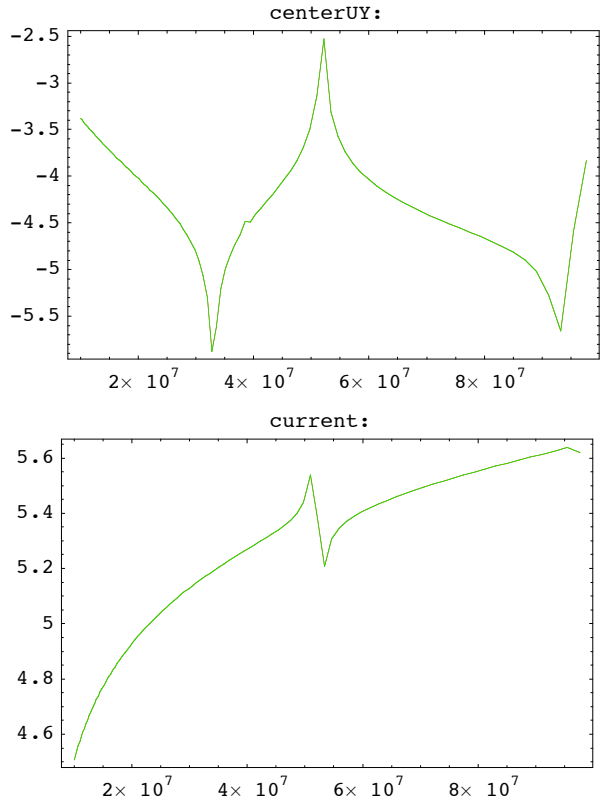


It can be seen that current result is not correct. The computed values needs in fact still to be summed with the dispacement current previously computed, and $i2\pi\omega$. This is done by the following function:

```
resNew = TransformResult[res, YNames[res],
  TransformFunction → {"current", Function[{y, freq}, 2 * Pi * freq * (y + voltcontrib)]}]
```

```
- SimulationResult -
```

```
PlotResult[{ansys, resNew}, FunctionY → (Log10[Abs[#]] &),
  PlotStyle → {RGBColor [1, 0, 0], RGBColor[0, 1, 0]}];
```



Relative percentage error:

```
PlotResult[Difference[ansys, resNew, ErrorFunction → ((Abs[#1] - Abs[#2]) / Abs[#1] &)],
  FunctionX → Log10, PlotStyle → {RGBColor[0, 1, 0]}];
```