
# setup

## overhead

## tag

In[521]:=
```
home = "ert/mercury/parse/";
Get["utility modules.m", Path → dirPack];
stamp1;
```

CreateDirectory: /Users/dantopa/Mathematica_files/io/ already exists.

CreateDirectory: /Users/dantopa/Dropbox/_mm/io/ert/ already exists.

CreateDirectory: /Users/dantopa/Dropbox/_mm/io/ert/mercury/ already exists.

General: Further output of CreateDirectory::filex will be suppressed during this calculation.

```
maximum memory: 0.312924 GB
seed file: /Users/dantopa/Mathematica_files/nb/seed 19_12.nb
user: dantopa, CPU: Xiuhcoatl,  MM v. 12.0.0 for Mac OS X x86
date: Jan 25, 2020, time: 21:04:08
nb: /Users/dantopa/Mathematica_files/nb/ert/mercury/parse/e-field-plotter.nb
```

## modules, functions, settings, ...

# 0 module library

# 1 read E fields from *.txt file

## point to file

In[ ]:=
```
dirMoM = "/Users/dantopa/Dropbox/2nd-generation/RCS-project/linux/ubuntu/";
```

## read file to scan for data sets

In[•]:=
```
strmList = Import[dirMoM <> "sphereCourse.4112.txt", "Data"]
Λ = Dimensions[strmList]
```

Out[•]=
```
{---| Run Date: January   21, 2020;    Time: 13:09:36, ,  MM SIE VIE,
  ··· 51 904 ···,                    ---| Mercury MOM Completed Sucessfully |---,
 ------------------------------------------------------------------------------
   ----}
```

large output | show less | show more | show all | set size limit...

Out[•]= {51 909}

## tag data sets: record line numbers

In[•]:=
```
(* each data set represents a unique frequency *)
```

In[•]:=
```
census = {};
Table[
  If[StringContainsQ[strmList[[k]], "  Freq   "], AppendTo[census, k]]
  , {k, Length[strmList]}];
census
m = Length[census]
```

Out[•]=
```
{485, 1000, 1515, 2030, 2545, 3060, 3575, 4090, 4605, 5120, 5635, 6150, 6665, 7180,
 7695, 8210, 8725, 9240, 9755, 10 270, 10 785, 11 300, 11 815, 12 330, 12 845,
 13 360, 13 875, 14 390, 14 905, 15 420, 15 935, 16 450, 16 965, 17 480, 17 995, 18 510,
 19 025, 19 540, 20 055, 20 570, 21 085, 21 600, 22 115, 22 630, 23 145, 23 660, 24 175,
 24 690, 25 205, 25 720, 26 235, 26 750, 27 265, 27 780, 28 295, 28 810, 29 325, 29 840,
 30 355, 30 870, 31 385, 31 900, 32 415, 32 930, 33 445, 33 960, 34 475, 34 990, 35 505,
 36 020, 36 535, 37 050, 37 565, 38 080, 38 595, 39 110, 39 625, 40 140, 40 655, 41 170,
 41 685, 42 200, 42 715, 43 230, 43 745, 44 260, 44 775, 45 290, 45 805, 46 320,
 46 835, 47 350, 47 865, 48 380, 48 895, 49 410, 49 925, 50 440, 50 955, 51 470}
```

Out[•]= 100

## collect complex field values at each frequency

### empty containers for E field values

In[•]:=
```
θθ = {};
θϕ = {};
ϕθ = {};
ϕϕ = {};
```

### advance line pointer to first data set

In[•]:=
```
myStream = OpenRead[dirMoM <> "sphereCourse.4112.txt"];
(* position pointer (line number) *)
Do[
  ReadLine[myStream]
  , {k, firstLine - 1}];
```

### sweep through sets

In[•]:=
```
(* measurement sets have unique frequencies *)
set = 0;
(* number of sets is determined by user in Mercury MoM *)
While[set < numSets,
  (* read results for specific frequency *)
  tbl = Table[
    (* grab a line of data as a text string *)
    data = ReadLine[myStream];
    (* extract the four complex field value: VV, VH, HV, HH *)
    Table[
     start = α + (k - 1) gap;
     (* extract complex value for an individual field *)
     grabComplex[StringTake[data, {start, start + 2 λ + 2}]]
     , {k, efields}]
    , {j, 0, 360}];
  Print["* * * loaded efield values: set = ", set];

  (* collect complex field values *)
  AppendTo[θθ, tbl[[All, 1]]];
  AppendTo[θϕ, tbl[[All, 2]]];
  AppendTo[ϕθ, tbl[[All, 3]]];
  AppendTo[ϕϕ, tbl[[All, 4]]];

  (* scan through data file to next measurement set *)
  Do[
   ReadLine[myStream];
   (* Print[StringTake[ReadLine[myStream],10]] *)
   , {k, jump}];
  (* increment set counter *)
  set++];
```

```
* * * loaded efield values: set = 0
```

# 2 extract data

### magnitude

In[•]:=
```
magVV = Map[Abs, θθ, 1];
magVH = Map[Abs, θϕ, 1];
magHV = Map[Abs, ϕθ, 1];
magHH = Map[Abs, ϕϕ, 1];
```

### phase

```
In[•]:= argVV = Map[Arg, θθ, 1];
        argVH = Map[Arg, θϕ, 1];
        argHV = Map[Arg, ϕθ, 1];
        argHH = Map[Arg, ϕϕ, 1];
```

### re

```
In[•]:= reVV = Map[Re, θθ, 1];
        reVH = Map[Re, θϕ, 1];
        reHV = Map[Re, ϕθ, 1];
        reHH = Map[Re, ϕϕ, 1];
```

### im

```
In[•]:= imVV = Map[Im, θθ, 1];
        imVH = Map[Im, θϕ, 1];
        imHV = Map[Im, ϕθ, 1];
        imHH = Map[Im, ϕϕ, 1];
```

## 3 table of extrema

### magnitude

```
In[•]:= fVV = Flatten[magVV];
        fVH = Flatten[magVH];
        fHV = Flatten[magHV];
        fHH = Flatten[magHH];
```

```
In[•]:= extrema = TableForm[
          {{Max[fVV], Max[fVH], Max[fHV], Max[fHH]},
           {Min[fVV], Min[fVH], Min[fHV], Min[fHH]}},
          , TableHeadings → {{"max", "min"}, {"VV", "VH", "HV", "HH"}}]
```

Out[•]//TableForm=

| | VV | VH | HV | HH |
|---|---|---|---|---|
| max | 8.05482 | 6.62405 | 6.62404 | 6.63266 |
| min | 0.00518485 | $1.275 \times 10^{-6}$ | $1.27464 \times 10^{-6}$ | 0.0123255 |

## argument

```
In[•]:= fVV = Flatten[argVV];
        fVH = Flatten[argVH];
        fHV = Flatten[argHV];
        fHH = Flatten[argHH];

In[•]:= extrema = TableForm[
          {{Max[fVV], Max[fVH], Max[fHV], Max[fHH]},
           {Min[fVV], Min[fVH], Min[fHV], Min[fHH]}},
          , TableHeadings → {{"max", "min"}, {"VV", "VH", "HV", "HH"}}]
```

Out[•]//TableForm=

| | VV | VH | HV | HH |
|---|---|---|---|---|
| max | 3.14144 | 3.14148 | 3.14148 | 3.14132 |
| min | −3.14152 | −3.14153 | −3.14152 | −3.14131 |

## re

```
In[•]:= fVV = Flatten[reVV];
        fVH = Flatten[reVH];
        fHV = Flatten[reHV];
        fHH = Flatten[reHH];

In[•]:= extrema = TableForm[
          {{Max[fVV], Max[fVH], Max[fHV], Max[fHH]},
           {Min[fVV], Min[fVH], Min[fHV], Min[fHH]}},
          , TableHeadings → {{"max", "min"}, {"VV", "VH", "HV", "HH"}}]
```

Out[•]//TableForm=

| | VV | VH | HV | HH |
|---|---|---|---|---|
| max | 3.69139 | 2.82538 | 2.82538 | 5.23855 |
| min | −6.66528 | −4.64852 | −4.64851 | −3.01551 |

## im

```
In[•]:= fVV = Flatten[imVV];
        fVH = Flatten[imVH];
        fHV = Flatten[imHV];
        fHH = Flatten[imHH];
```

```
In[•]:= extrema = TableForm[
          {{Max[fVV], Max[fVH], Max[fHV], Max[fHH]},
           {Min[fVV], Min[fVH], Min[fHV], Min[fHH]}},
          , TableHeadings → {{"max", "min"}, {"VV", "VH", "HV", "HH"}}]
```

Out[•]//TableForm=

| | VV | VH | HV | HH |
|---|---|---|---|---|
| max | 3.72418 | 4.93394 | 4.93394 | 5.6146 |
| min | −5.05535 | −3.65703 | −3.65703 | −2.88838 |

# 4 plot E fields

```
In[•]:= νticks = {{1, 100}, {25, 75}, {50, 50}, {75, 25}, {100, 1}};
        λticks = {{11.06, 1}, {71.02, 10}, {98.00, 100}};
        θticks = {{1, 0}, {91, π/4}, {181, π/2}, {271, 3 π/4}, {361, 2 π}};
        θticks = {{1, 0}, {91, 90}, {181, 180}, {271, 270}, {361, 360}};
```

## magnitude

```
In[•]:= Clear[myPlot];
        myPlot[ψ_, str_String] := Module[{g},
          g = MatrixPlot[Reverse[ψ],
            AspectRatio → 0.5,
            ColorFunction → Hue,
            FrameLabel → {{"θ", Null}, {"ν (MHz)", "λ (m)"}},
            PlotLabel → "Sphere" <> lf <> str,
            FrameTicks → {{νticks, λticks}, {θticks, None}}];
          Return[g];
         ]

In[•]:= g001 = myPlot[magVV, "Energy Magnitude (VV)"]
        g002 = myPlot[magVH, "Energy Magnitude (VH)"]
        g003 = myPlot[magHV, "Energy Magnitude (HV)"]
        g004 = myPlot[magHH, "Energy Magnitude (HH)"]
```

## argument

```
In[•]:= g101 = myPlot[argVV, "Complex Argument (VV)"]
        g102 = myPlot[argVH, "Complex Argument (VH)"]
        g103 = myPlot[argHV, "Complex Argument (HV)"]
        g104 = myPlot[argHH, "Complex Argument (HH)"]
```

### real part

```
g201 = myPlot[reVV, "Real Part (VV)"]
g202 = myPlot[reVH, "Real Part (VH)"]
g203 = myPlot[reHV, "Real Part (HV)"]
g204 = myPlot[reHH, "Real Part (HH)"]
```

### imaginary part

In[•]:=
```
g301 = myPlot[imVV, "Imaginary Part (VV)"]
g302 = myPlot[imVH, "Imaginary Part (VH)"]
g303 = myPlot[imHV, "Imaginary Part (HV)"]
g304 = myPlot[imHH, "Imaginary Part (HH)"]
```

## 5 export

### magnitude

In[•]:=
```
tresExport["VV-mag", g001];
tresExport["VH-mag", g002];
tresExport["HV-mag", g003];
tresExport["HH-mag", g004];
```

### argument

In[•]:=
```
tresExport["VV-arg", g101];
tresExport["VH-arg", g102];
tresExport["HV-arg", g103];
tresExport["HH-arg", g104];
```

### re

In[•]:=
```
tresExport["VV-re", g201];
tresExport["VH-re", g202];
tresExport["HV-re", g203];
tresExport["HH-re", g204];
```

### im

```
In[•]:= tresExport["VV-im", g301];
        tresExport["VH-im", g302];
        tresExport["HV-im", g303];
        tresExport["HH-im", g304];
```

---

## end