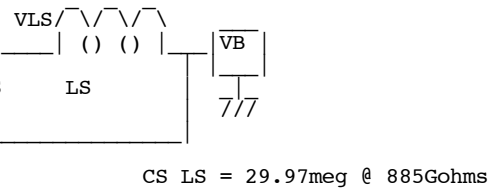
*==========**CRYSTAL_Serial_Network**==========

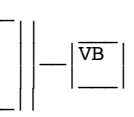
What does an AC simulation predict as the oscillation frequency for a crystal model sub-circuit being used in the serial mode? Being able to mathematically process the output of an AC analysis can provide some needed precision.

*=========**CRYSTAL_SubCircuit**==========

```
.SUBCKT    XCRYST VA     VB
 RS        VA     VCS    340
 CS        VCS    VLS    7f
 LS        VLS    VB     3.5
 CP        VA     VB     3pf
.ENDS      XCRYST
```




```
*                                   CS LS = 29.97meg @ 885Gohms
```

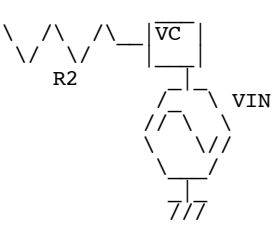



```
* Freq                 Mode  L1     C1        R1    C0      Q
* 1 MHz                fund  3.5H   0.007pf   340   3pf     64679
```

The point where the crystal is more like a zero phase short circuit is being sought.

*=========**Test_Circuit**==========

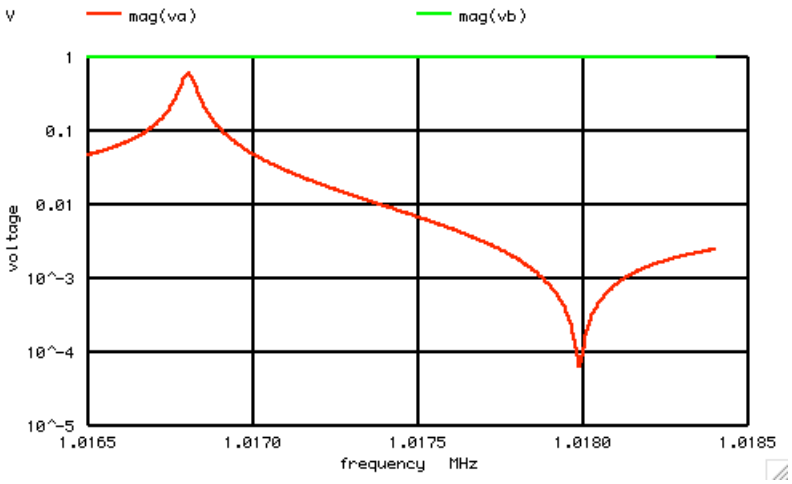
```
OneMegCrystal_serial_network
*
```

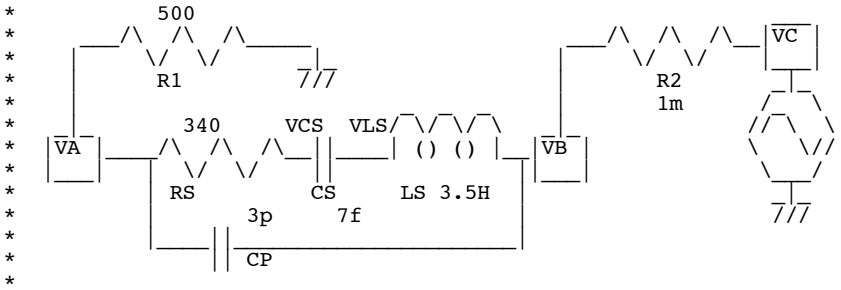


*=========**Run_AC**==========

```
.OPTIONS   GMIN=1e-18     METHOD=trap    ABSTOL=1e-18   TEMP=27 srcsteps = 1   gminsteps = 1
VIN        VC      0      DC       AC    1
XCRY1      VA      VB     XCRYST
R1         VA      0      500
R2         VC      VB     1m
.control
*AC        DECLin NUMDEC FSTART     FSTOP
ac         dec    600000 1.0165meg  1.0184meg
set        pensize = 2
plot       mag(va) mag(vb)    ylog
```

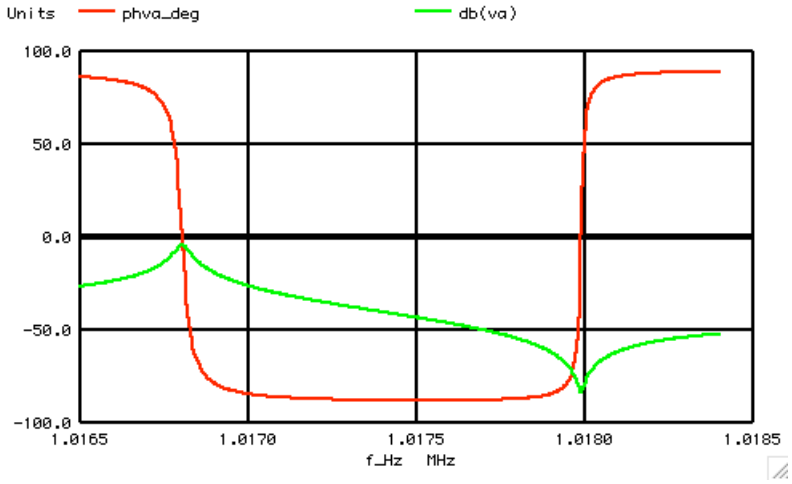
At one frequency, the impedance of the crystal will try to look like a short circuit. At another frequency it will try to look like an open circuit.





==========

One needs to look at phase as well. The plot below shows when the phase crosses zero.



==========

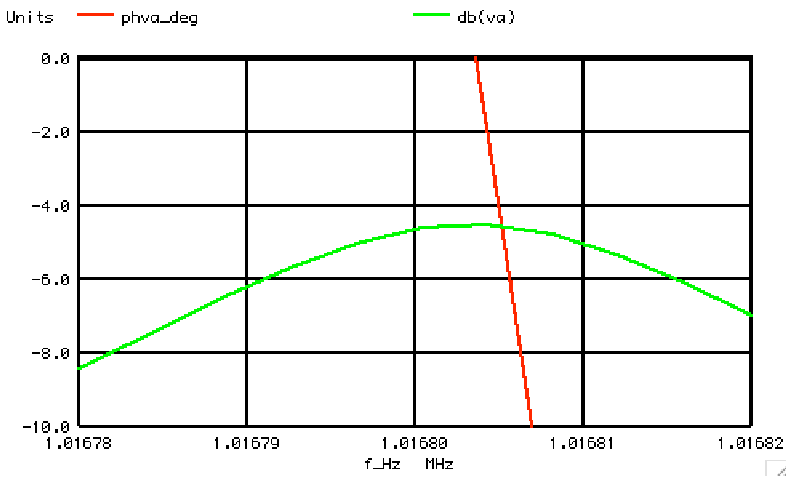
Finding the size of the output array is straight forward. It can then be used to find out where the phase magnitude is minimum.

```
*=========Process_Output_Arrays========
let       lenV= length(va)
compose   phva_deg start = 1 stop = $&lenV step =1
compose   f_Hz     start = 1 stop = $&lenV step =1
settype   frequency       f_Hz
*=========Remove_Phase_Wrap_around_If_Needed========
let       i = 0
let       phasMin=180
let       FrqPHZero = 0

repeat    $&lenV
let       phva_deg[i] = ph(va[i])

if        (abs(phva_deg[i]) < phasMin)
```

```
let      phasMin = abs(phva_deg[i])
let      FrqPHZero = frequency[i]
endif

let      f_Hz[i] = frequency[i]
let      i = i +1
end
```
===

Of course on can just gain scale up the output plot to do about the same thing.

```
*=========Find_Where_Phase_zero========
plot     phva_deg   db(va) vs f_Hz
plot     phva_deg   db(va) vs f_Hz xlimit 1.01678Meg 1.01682meg ylimit -10 0
```
===



===

But the printed out value can have more precision.

```
*=========Find_Where_Phase_zero========
print    FrqPHZero
.endc
```
===

The output frequency for the AC simulation is the following..

Circuit: OneMegCrystal_series_Network
frqphzero =   1.01680e+06

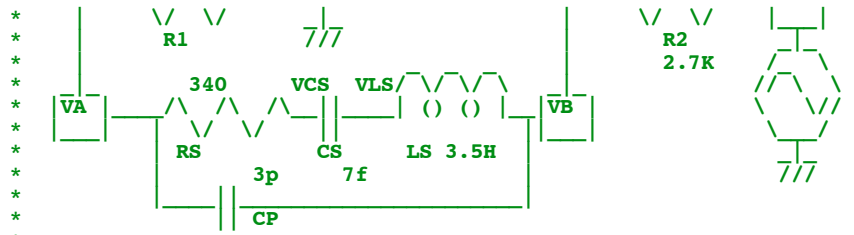
The output frequency for the free running oscillator was at the following..

Circuit: OneMeg_FreeRun_Serial_Crystal
mean(vref) =   2.42844e+00
mean(1/dt_tp) =   1.01649e+06

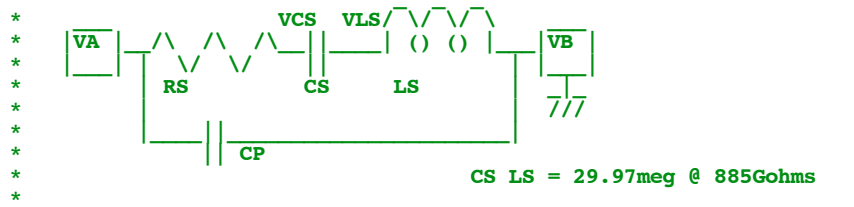
That is a 0.03% error. Tweaking the simulation conditions should be able to reduce it. Now it is easier to see all the errors.

===========Full_Netlist_For_Copy_Paste=======================

```
OneMegCrystal_series_Network
*
*           500
*     _/\  /\  /\_     __/\  /\  /\__|VC |
*    |   \/  \/   _|_ |      \/  \/    |___|
*    |       R1   ///  |        R2       _|_
*    |                 |                / _ \
*    |       ____      |               //  \ \
*   |VA|__||    ||__|VB|              \    \//
*   |___|  ||    ||  |___|             \___/
*          ||____||                       _|_
*                                         ///
*          500
*     ___/\  /\  /\_____                    ___/\  /\  /\__|VC |
```




```
* 1.017e+6
* Freq                Mode  L1     C1       R1    C0      Q
* 1 MHz               fund  3.5H   0.007pf  340   3pf     64679
*
.OPTIONS  GMIN=1e-18    METHOD=trap   ABSTOL=1e-18  TEMP=27 srcsteps = 1  gminsteps = 1
VIN       VC     0      DC      AC     1
XCRY1     VA     VB     XCRYST
R1        VA     0      500
R2        VC     VB     1m

.control
*AC       DECLin NUMDEC FSTART     FSTOP
ac        dec    600000 1.0165meg  1.0184meg
set       pensize = 2
plot      mag(va) mag(vb)    ylog

*=========Process_Output_Arrays========
let       lenV= length(va)
compose   phva_deg start = 1 stop = $&lenV step =1
compose   f_Hz     start = 1 stop = $&lenV step =1
settype   frequency       f_Hz
*=========Remove_Phase_Wrap_around_If_Needed========
let       i = 0
let       phasMin=180
let       FrqPHZero = 0
repeat    $&lenV
let       phva_deg[i] = ph(va[i])

if        (phva_deg[i] > 0)
let       phva_deg[i] = phva_deg[i]
endif

if        (abs(phva_deg[i]) < phasMin)
let       phasMin = abs(phva_deg[i])
let       FrqPHZero = frequency[i]
endif

let       f_Hz[i] = frequency[i]
let       i = i +1
end

*=========Find_Where_Phase_zero========

plot      phva_deg   db(va) vs f_Hz
plot      phva_deg   db(va) vs f_Hz xlimit 1.01678Meg 1.01682meg ylimit -10 0
print     FrqPHZero

.endc

*=========CRYSTAL_SubCircuit================================================
.SUBCKT   XCRYST VA     VB
 RS       VA     VCS    340
 CS       VCS    VLS    7f
 LS       VLS    VB     3.5
 CP       VA     VB     3pf
.ENDS     XCRYST
*
```




```
*                                  CS LS = 29.97meg @ 885Gohms
*
```


```
*
* Freq                Mode  L1     C1       R1    C0      Q
* 1 MHz               fund  3.5H   0.007pf  340   3pf     64679
```

```
*
*=======================================================================
.end

4.11.10_4.54PM
dsauersanjose@aol.com
Don Sauer
http://www.idea2ic.com/
```