setup

overhead

tag

In[457]:=
```
home = "rcs/fourier/latex-2/";
Get["utility modules.m", Path → dirPack];
Get["rcs-tools-01.m", Path → dirnb <> "rcs/tools/"];
Get["plot-library-01.m", Path → dirnb <> "rcs/tools/"];
Get["markers.m", Path → dirnb <> "rcs/tools/"];
stamp1;
```

CreateDirectory: /Users/dantopa/primary–repos/github/experiment–mathematica/io/ already exists.

CreateDirectory: /Users/dantopa/primary–repos/github/experiment–mathematica/io/rcs/ already exists.

CreateDirectory: /Users/dantopa/primary–repos/github/experiment–mathematica/io/rcs/fourier/ already exists.

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

```
maximum memory: 0.25106 GB
seed file: /Users/dantopa/primary-repos/github/experiment-mathematica/nb/seed 19_12.nb
user: dantopa, CPU: Xiuhcoatl,  MM v. 12.1.0 for Mac OS X x86
date: May 14, 2020, time: 08:59:02
nb: /Users/dantopa/primary-repos/github/experiment-mathematica/nb/rcs/fourier/latex-2/
 super-writer-01.nb
```

modules, functions, settings, ...

admin

In[463]:=
```
printmem
```

```
maximum memory: 0.25106 GB
```

In[464]:=
```
(* save notebook *)
NotebookSave[EvaluationNotebook[]];
```

settings

In[465]:=
```
exportFlag = True;
```

In[466]:=
```
o = {0, 0};
```

In[467]:=
```
guc = Circle[o, 1];
```

In[468]:=
```
ipad = ImagePadding → {{Automatic, 5}, {Automatic, 5}};
```

In[469]:=
```
isize = ImageSize → 5 × 72;
```

functions

substitutions

modules

In[•]:=
```
Clear[errorSpectra];
errorSpectra[v_Integer, dof_Integer] :=
 Module[{inf, two, stub, ℊ, x, y, hline, vline, ga, gb, lblx, lbly},
  stub =
   "/Users/dantopa/primary-repos/github/experiment-mathematica/io/rcs/fourier/
     search/data/norms/";
  index = ToString[pad[v]];
  (* harvest data *)
  ℊ = Import[stub <> "norm-two-" <> index <> ".dat", "Data"];
  two = First[#] & /@ ℊ;
  ℊ = Import[stub <> "norm-inf-" <> index <> ".dat", "Data"];
  inf = First[#] & /@ ℊ;
  (* inf norm *)
  lblx = "Degree of fit";
  lbly = "Infinity Norm Error";
  ga = ListLogPlot[{Range[0, 50], inf}ᵀ,
    PlotStyle → Black,
    Frame → True];
  x = dof;
  y = Log[inf[[dof + 1]]];
  hline = Line[{{-1, y}, {51, y}}];
  vline = Line[{{x, Log[0.0001]}, {x, Log[100 000]}}];
  gb = Graphics[{Gray, Opacity[0.5], hline, vline}];
  ginf = Show[{ga, gb}, ipad, FrameLabel → {lblx, lbly}];
  (* inf norm *)
  lbly = "2-Norm Error";
  ga = ListLogPlot[{Range[0, 50], two}ᵀ,
    PlotStyle → Black,
    Frame → True];
  x = dof;
  y = Log[two[[dof + 1]]];
  hline = Line[{{-1, y}, {51, y}}];
  vline = Line[{{x, Log[0.0001]}, {x, Log[100 000]}}];
  gb = Graphics[{Gray, Opacity[0.5], hline, vline}];
  gtwo = Show[{ga, gb}, ipad, FrameLabel → {lblx, lbly}];
 ]
```

significant digits

In[•]:=
```
Clear[significantDigits];
significantDigits[x_, σ_] := Module[{},
  lg = Log[10, σ];
  pow = Abs[Floor[lg] + Sign[lg]];
  y = Round[10^pow x] / 10^pow // N;
  ip = IntegerPart[y];
  fp = StringPadRight[ToString[FractionalPart[y]], pow + 3, "0"];
  Return[{fp, ip}];
 ]
```

In[•]:=
```
Clear[sigDigTableEntry];
sigDigTableEntry[x_, s_] := Module[{},
  lg = Log[10, s];
  pow = Abs[Floor[lg] + Sign[lg]];
  y = Round[10^pow x] / 10^pow // N;
  z = Round[10^pow s] / 10^pow // N;
  len = If[x < 0, 1, 0];
  zz = StringPadRight[ToString[FractionalPart[z]], pow + 2, "0"];
  Print[ToString[y], " ~pm ", zz]
 ]
```

In[•]:=
```
Clear[argon];
argon[τ_, s_] := Module[{lg, pow, seq, num, kleft},
  Print["τ = ", τ];
  Print["s = ", s];
  lg = Log[10, s];
  pow = Abs[Floor[lg] + Sign[lg]];
  sgn = If[τ < 0, "-", ""];
  kleft = Ceiling[Log[10, Abs[τ]]];
  seq = IntegerDigits[Round[τ 10^pow]];
  num = sgn;
  Do[
   num = num <> ToString[seq[[k]]];
   , {k, kleft}];
  If[kleft == 0, num = num <> "0"];
  num = num <> ".";
  Do[
   num = num <> ToString[seq[[k]]];
   , {k, kleft + 1, Length[seq]}];
  Return[num]
 ]
```

In[•]:=
```
Clear[boron];
boron[τ_, s_] := Module[{lg, pow, seq, num, kleft},
  lg = Log[10, s];
  pow = Abs[Floor[lg] + Sign[lg]];
  sgn = If[τ < 0, "-", ""];
  kleft = Ceiling[Log[10, Abs[τ]]];
  seq = IntegerDigits[Round[τ 10^pow]];
  num = sgn;
  Do[
   num = num <> ToString[seq[[k]]];
   , {k, kleft}];
  If[kleft == 0, num = num <> "0"];
  num = num <> ".";
  Do[
   num = num <> ToString[seq[[k]]];
   , {k, kleft + 1, Length[seq]}];
  Return[num]
 ]
```

```
In[•]:= Clear[carbon];
carbon[τ_, s_] := Module[{lg, pow, λ},
  lg = Log[10, s];
  pow = Abs[Floor[lg] + Sign[lg]];
  x = ToString[Round[τ 10^pow] / 10^pow // N];
  λ = StringLength[x];
  positionDecimalPoint = First[First[StringPosition[x, "."]]];
  digitsAterDecimal = λ - positionDecimalPoint;
  precisionDesired = pow;
  deficit = precisionDesired - digitsAterDecimal;
  If[deficit > 0, x = StringPadRight[x, StringLength[x] + deficit, "0"]];
  Return[x]
 ]
```

solution function in latex

```
In[•]:= Clear[myfunction];
myfunction[tbl_, v_] := Module[{λ, breaks, fcn, freq, num, sgn, term},
    breaks = 4 Range[9] + 2;
    λ = Length[tbl];
    fcn = "f_{" <> ToString[v] <> "}(~alpha) &= " <> tbl[[1]];
    Do[
     num = tbl[[k]];
     sgn = " + ";
     If[StringTake[num, 1] == "-", sgn = " - ";
      num = StringDrop[num, 1]];
     If[k == 2, freq = "", freq = ToString[k - 1]];
     If[Length[Position[breaks, k]] > 0, sgn = "\\ & " <> sgn];
     term = sgn <> num <> " ~cos(" <> freq <> "~alpha)";
     fcn = fcn <> term;
     , {k, 2, λ}];
    Return[fcn];
   ];
```

tagger

assemble plots

```
In[•]:= tab = "    ";
        ttab = tab <> tab;
        tttab = tab <> ttab;
        ttttab = tab <> tttab;
```

markers

```
In[•]:= markerLocker =
          "/Users/dantopa/primary-repos/github/experiment-mathematica/io/rcs/fourier/panic
            /data/markers-naked/";
```

```
In[•]:= markers = Import[markerLocker <> "markers-log-" <> elev <> ".txt", "CSV"];
```

```
In[•]:= locker =
          "/Users/dantopa/primary-repos/github/experiment-mathematica/io/rcs/fourier/latex
            -2/data/";
```

```
In[•]:= angle = 80;
        elev = "0p" <> pad[angle, 3];
        nameElev = "-" <> elev;

        tbl = Table[
            carbon[amps[[l]], errs[[l]]]
            , {l, Length[amps]}];
```

secList

In[•]:=
```
myList = StringReplace[secList, "~" → FromCharacterCode[92]];
myList = StringReplace[myList, "@" → FromCharacterCode[34]];
myList
```

Out[•]=
```
\input{./sections/"ssec nu=03-d=04"}
\input{./sections/"ssec nu=04-d=06"}
\input{./sections/"ssec nu=05-d=05"}
\input{./sections/"ssec nu=06-d=08"}
\input{./sections/"ssec nu=07-d=11"}
\input{./sections/"ssec nu=08-d=10"}
\input{./sections/"ssec nu=09-d=10"}
\input{./sections/"ssec nu=10-d=12"}
\input{./sections/"ssec nu=11-d=12"}
\input{./sections/"ssec nu=12-d=16"}
\input{./sections/"ssec nu=13-d=18"}
\input{./sections/"ssec nu=14-d=20"}
\input{./sections/"ssec nu=15-d=22"}
\input{./sections/"ssec nu=16-d=22"}
\input{./sections/"ssec nu=17-d=26"}
\input{./sections/"ssec nu=18-d=26"}
\input{./sections/"ssec nu=19-d=28"}
\input{./sections/"ssec nu=20-d=25"}
\input{./sections/"ssec nu=21-d=25"}
\input{./sections/"ssec nu=22-d=26"}
\input{./sections/"ssec nu=23-d=28"}
\input{./sections/"ssec nu=24-d=28"}
\input{./sections/"ssec nu=25-d=28"}
\input{./sections/"ssec nu=26-d=28"}
\input{./sections/"ssec nu=27-d=30"}
\input{./sections/"ssec nu=28-d=34"}
\input{./sections/"ssec nu=29-d=36"}
\input{./sections/"ssec nu=30-d=40"}
```

In[•]:=
```
edit[dirData <> "fixer.zsh"]
```

```
Removing quotes from
 /Users/dantopa/primary-repos/github/experiment-mathematica/io/rcs/fourier/latex/data/
   fixer.zsh
```

Export: Cannot infer format of file fixer.zsh.

In[•]:=
```
cleaner[dirData <> "fixer.zsh", dirData <> "fixerC.zsh"]
```

```
file name =
 /Users/dantopa/primary-repos/github/experiment-mathematica/io/rcs/fourier/latex/data/
   fixer.zsh
```

--- Export: Cannot infer format of file fixerC.zsh.

---

end