

---

## setup

overhead

tag

```
In[ ]:= home = "rcs/fourier/search/";
Get["utility modules.m", Path → dirPack];
Get["rcs-tools-01.m", Path → dirnb <> "rcs/tools/"];
Get["latex-tools-01.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: 0.424271 GB

seed file: /Users/dantopa/primary-repos/github/experiment-mathematica/nb/seed 19_12.nb

user: dantopa, CPU: Xihcoatl, MM v. 12.1.0 for Mac OS X x86

date: May 13, 2020, time: 09:28:50

nb: /Users/dantopa/primary-repos/github/experiment-mathematica/nb/rcs/fourier/search/
nelson-scanner-05.nb
```

modules, functions, settings, ...

---

## import

```
In[ ]:=  $\sigma$  = Import[dirDataLocker <> sciaccarcs];
Dimensions[ $\sigma$ ]
 $\lambda$  = Length[ $\sigma$ ]

Out[ ]:= {28, 361}

Out[ ]:= 28
```

---

## summary report

```
In[ ]:=  $\omega$  = 50;
mesh = Range[-180, 180];

file = dirData <> "summary-log.dat";
 $\psi$  = OpenWrite[file, PageWidth →  $\infty$ ];
```

```

tagSource[ψ, ""]
τ = 5;
markers = {};
Do[
  knu = nu - 2;
  b = σ[[knu]];
  brange = (Max[b] - Min[b]);
  showBrange =  $\frac{\text{Round}[1000 \text{ brange}]}{1000}$  // N;
  $tick;
  (* result stubs *)
  criteriaA = "condition NOT for satisfied for degree <= "<>ToString[ω];
  criteriaB = criteriaA;
  criteriaC = criteriaA;
  satisfiedA = False;
  satisfiedB = False;
  satisfiedC = False;
  marks = {nu, -1, -1, -1};
  Do[
    $ttick;
    (* linear system *)
    fileName = "nu="<>ToString[nu]<>"-d="<>pad[d];
    Clear[basis];
    basis[θ_] := Table[Cos[k θ], {k, 0, d}];
    (* build linear system *)
    A = BuildAFourierCos[mesh, d];
    (* least squares solution *)
    x = LeastSquares[A, b];
    (* error analysis *)
    errorN[A, x, b];
    (* criteriaA analysis *)
    If[satisfiedA, Goto[B]];
    sd = StandardDeviation[residual];
    If[3 sd ≤ τ,
      satisfiedA = True;
      marks[[2]] = d;
      dA = ToString[d];
      showrsd = ToString[ $\frac{\text{Round}[1000 \text{ rsd}]}{1000}$  // N];
      criteriaA = "Satisfied at d = "<>ToString[d]<>
        " where standard deviation of the residuals = "<>showrsd;
    ];
  ];

```

```

(* criteriaB analysis *)
Label[B];
If[satisfiedB, Goto[C]];
solutionPts = x.basis[ $\theta$ ] /.  $\theta \rightarrow \text{mesh } \frac{\pi}{180}$ ;
dataRange = Max[solutionPts] - Min[solutionPts];
dataRange =  $\frac{\text{Round}[1000 \text{ dataRange}]}{1000}$  // N;
nelson =  $\frac{\text{Round}[1000 (\text{Max}[b] - \text{Max}[\text{solutionPts}])]}{1000}$  // N;
If[nelson ≤ 10,
  satisfiedB = True;
  marks[[3]] = d;
  dB = ToString[d];
  showNelson = ToString[nelson];
  criteriaB = "Satisfied at d = "<>
    ToString[d]<>" where the amplitude difference = "<>showNelson;
];
(* criteriaC analysis *)
Label[C];
If[satisfiedC, Goto[D]];
residualRange = Max[residual] - Min[residual];
rsd = StandardDeviation[residual];
trsd = 3 rsd;
showResidualRange =  $\frac{\text{Round}[1000 \text{ residualRange}]}{1000}$  // N;
showrsd =  $\frac{\text{Round}[1000 \text{ rsd}]}{1000}$  // N;
If[residualRange ≤  $\tau$ ,
  satisfiedC = True;
  marks[[4]] = d;
  dC = ToString[d];
  sResidualRange = ToString[showResidualRange];
  criteriaC = "Satisfied at d = "<>ToString[d]<>
    " where variation in residuals = "<>sResidualRange;
];
Label[D];
If[satisfiedA ∧ satisfiedB ∧ satisfiedC, markers = AppendTo[markers, marks];
  Break[]];
, {d,  $\theta$ ,  $\omega$ };
Write[ $\psi$ , ""];
Write[ $\psi$ , "Summary for nu = ", pad[nu], ", variation in data = ", showBrange];
Write[ $\psi$ , "Criteria 1: (3 sigma) 97.5% of |residuals| ≤ 5: ", criteriaA];

```

```

Write[ψ, "Criteria 2: 90% of amplitude: ", criteriaB];
Write[ψ, "Criteria 3: |residuals| <= 5:", criteriaC];
ttiempo["frequency = ", nu, " MHz"];
, {nu, 3, 30}]
tiempo["full sweep"]
Close[ψ];
(* edit[file]; *)
(* write markers file *)
file = dirData<>"markers-log.dat";
ψ = OpenWrite[file, PageWidth → ∞];
tagSource[ψ, ""];
Do[
  Write[ψ, markers[[k]]]
  , {k, Length[markers]}]
Close[ψ];
(* edit[file]; *)

```