Perplexed Computations

DANIEL LICHTBLAU
Scientific Information Group
Wolfram Research
Champaign IL 61820
danl@wolfram.com

WACHARIN WICHIRAMALA
Department of Mathematics and Computer Science
Chulalongkorn University
Bangkok, Thailand
wacharin.w@chula.ac.th

In this supplement we describe various ways to count, compute and cull the data tuples corresponding to Fourier arrangements. Actual computations were performed with *Mathematica 8* [**5**].

Counting the Data Tuples

Recall from the main article [**1**] that we want to count data tuples corresponding to the multiple intersection extension of Fourier's 17 lines problem. To count the number of distinct data tuples we use Roberts' formula [**3**] to set up a partitioning problem. We recall that formula:

$$\sum_{i=1}^{m}\left[\binom{\lambda_i}{2}-1\right]+\sum_{j=1}^{p}\binom{\mu_j}{2}=35$$

We also have from [**1**] the restrictions $3 \le \lambda_i \le 8$ and $2 \le \mu_j \le 8$ for each λ_i and μ_j. We can now set up partitioning problems wherein the lists of allowed λ and μ values are given by the binomial coefficients less 1 and binomial coefficients respectively. That is, the λ values live in the set $\{2, 5, 9, 14, 20, 27\}$ and the μ values are in $\{1, 3, 6, 10, 15, 21, 28\}$. The number of ways to partition 35 into the λ values is 19. For partitioning into the μ values there are 121 distinct ways. If we allow partitions into the union of these two sets there are 2335 possibilities. Each of these figures can be computed with *Mathematica*'s `IntegerPartitions` function. Subtracting $19 + 121$ from 2335 shows that there are 2195 data tuples arrangements that have at least one set of parallel lines and at least one multiple intersection point.

Code for above:

```
lamvals = (Binomial[#,2]-1)&/@Range[3,8]
   {2,5,9,14,20,27}
muvals = Binomial[#,2]&/@Range[2,8]
   {1,3,6,10,15,21,28}
lammuvals = Join[lamvals,muvals];
lampartitions = IntegerPartitions[35, Infinity, lamvals];
mupartitions = IntegerPartitions[35, Infinity, muvals];
lammupartitions = IntegerPartitions[35, Infinity, lammuvals];
Map[Length, {lampartitions, mupartitions, lammupartitions}]
   {19,121,2335}
```

Necessary Conditions for Realizable Arrangements

From [**1**] we recall that a necessary condition for realizability is that the sum of μ_j not exceed 17. This inequality does not allow us to work further with a simple integer partitions function. We instead resort to technology that is at the root of that functionality, to wit, generating functions. The methods we use are based on ideas from [**2**]. To recover the value of 2335 using generating functions, for example, one could form the rational function comprised of the product of reciprocals of $1-x^2, \ldots, 1-x^{27}, 1-x, \ldots, 1-x^{28}$. That is to say, we subtract from 1 the monomial given by x raised to each of the allowed λ and μ values. Multiply these terms and take the reciprocal of this product. Now consider the (formal) power series expansion of this rational function. The coefficient of the term of degree 35 in this series provides the number of integer partitions we derived above, 2335.

In order to enforce the new inequality we will alter our rational function with new variables λ and $\mu[j]$. For example, we will have denominator terms such as $1-(\lambda x)^2$ and $1-\mu[4]x^6$ since 2 and 6 are in the list of λ and μ values respectively and moreover 6 is the third such value on the μ list. This is because 2 and 6 are in the list of λ and μ values respectively and moreover 6 is the third such value on the μ list (and $\mu[4]$ is the third μ in the indexing scheme we will use).

We now explain why we treat the μ variables in such a peculiar manner. We will again create a rational function and extract the (now somewhat complicated) coefficient of the term in x^{35}. We will throw away the coefficient involving λ^{35} as it corresponds to tuples that use no parallel lines. Likewise we will throw away the terms that have no power of λ as they correspond to tuples that use no multiple intersection points. For the remaining coefficients we consider all factors of the form $\mu[j]$ perhaps raised to some power k. For each such we will multiply that exponent k times j, then sum over all these factors, and discard any terms wherein that sum exceeds 17. This enforces the new criterion. We now set λ and every μ factor to unity; they have served their purpose of demarcating just how the various components of x^{35} came about. When the dust settles we have a coefficient of 901.

Code for above:

```
allpowers = 1-x^lammuvals;
rat = Times@@(1/allpowers)
```

$$1/\left(\left(1-x\right)\left(1-x^2\right)\left(1-x^3\right)\left(1-x^5\right)\left(1-x^6\right)\left(1-x^9\right)\left(1-x^{10}\right)\left(1-x^{14}\right)\left(1-x^{15}\right)\left(1-x^{20}\right)\left(1-x^{21}\right)\left(1-x^{27}\right)\left(1-x^{28}\right)\right)$$

```
SeriesCoefficient[rat,{x,0,35}]
    2335
mus = Array[μ,7,2];
lampowers2 = 1-(λx)^lamvals
```

$$\left\{1-x^2\lambda^2, 1-x^5\lambda^5, 1-x^9\lambda^9, 1-x^{14}\lambda^{14}, 1-x^{20}\lambda^{20}, 1-x^{27}\lambda^{27}\right\}$$

```
mupowers2 = 1 - mus*x^muvals
```

$$\left\{1-x\mu[2], 1-x^3\mu[3], 1-x^6\mu[4], 1-x^{10}\mu[5], 1-x^{15}\mu[6], 1-x^{21}\mu[7], 1-x^{28}\mu[8]\right\}$$

```
allpowers2 = Times@@Join[lampowers2,mupowers2];
coeff35 = Expand[SeriesCoefficient[1/allpowers2,{x,0,35}]];
rule1 = μ[j_]^k_:>μ[j*k];
rule2 = μ[j_]*μ[k_]:>μ[j+k];
f[expr_] := (expr/.rule1)/.rule2
```

```
coeff35b = (FixedPoint[f,coeff35]/.{λ^35 →0,μ[j_/;j>17]:→0});
coeff35c = coeff35b-Coefficient[coeff35b,λ,0];
coeff35c/.{λ →1,μ[_]→1}
    901
```

Finding the Actual Tuples

We now ask whether one can not only count the data tuples but actually find them. This will be useful in enforcing yet another condition necessarily satisfied by realizable tuples. This can be done with a parametrized generating function in a manner similar to what we showed above. But frankly that was enough pain for one supplementary note, so we'll go in a different direction.

We will define a variable for each allowed λ and μ value to denote how many of that value's corresponding binomial (less 1, for the λ variables) to use in a given partitioning of 35 (we will henceforth refer to both cases as a "possibly modified binomial"). We impose the constraints that they all be positive and integer valued, and that the sum of the appropriate possibly modified binomials times their corresponding variables be 35. To rule out the solutions involving either no parallel lines or no multiple intersection points we also enforce that the sum of the variables for λ and the sum of the variables for μ respectively be at least 1. In order to avoid a notational mismatch between article and supplement, we will use subscripted κ and ν variables to count the λ and μ values respectively.

This set of equality and inequality constraints gives an integer linear programming problem, customarily abbreviated ILP. We refer to [**4**] for details of how such problems might be tackled. Here we simply remark that under the hood one solves both linear equations over the integers and linear inequalities over the reals. In around four seconds on a modern desktop machine the result is computed. We have, as expected, 2195 such tuples. When we add the restriction $\sum_j j\nu_j \leq 17$ we obtain our 901 tuples in about half that time.

Code for above

```
kappas = Array[κ,6,3];
nus = Array[ν,7,2];
allvars = Join[kappas,nus];
c1 = Thread[allvars≥0];
c2 = Thread[{Total[kappas],Total[nus]}≥1];
c3 = Total[Map[#*Binomial[#[[1]],2]&,allvars]]
            -Total[kappas]==35
    2κ[3]+5κ[4]+9κ[5]+14κ[6]+20κ[7]+27κ[8]+ν[2]+3ν[3]
                +6ν[4]+10ν[5]+15ν[6]+21ν[7]+28ν[8]==35
constraints = Join[c1,c2,{c3}];
Timing[alltuples = Reduce[constraints,allvars,Integers];]
    {4.08,Null}
Length[alltuples]
    2195
c4=Total[Map[#*#[[1]]&,nus]]≤17;
Timing[muboundedtuples=Reduce[Append[constraints,c4],
            allvars,Integers];]
```

```
    {1.62,Null}
Length[muboundedtuples]
    901
```

Pruning the Tuples

There is yet another necessary condition to be met by any realizable tuple. This one is a bit more subtle than the last. We recall from our high school geometry that (Euclidean) parallel lines do not intersect. Hence any line intersecting a family of parallel lines hits them in separate points. How might this help us? We take a given data tuple. We count the number of parallel line "families," with quotes because for the present purpose we will regard each singleton (that is, nonmember of such a family) as a family in and of itself. Then the above remark shows that an intersection point can contain at most one member of each such family. Thus realizability requires that all κ variables in the tuple must be bounded by this number of families.

One can formulate this last constraint in a way that is amenable to ILP. We instead will take our 901 explicit tuples and cull from that set any that fail to meet this new criterion. When we do so we learn that there is exactly one such failure. It contains one multiple intersection involving 6 lines, three parallel families of 3 lines each, and two parallel families comprised each of 4 lines. It is straightforward to check that this tuple meets all the requirements of the prior sections. Indeed, all lines live in bona fide parallel families ($3 \times 3 + 4 \times 2 = 17$), and the necessary sum is fulfilled:

$$\binom{6}{2} - 1 + 3 \times \binom{3}{2} + 2 \times \binom{4}{2} = 35.$$

But there are only 5 families of differently oriented lines, hence there can be no intersection point of 6 distinct lines. All remaining 900 tuples give Steiner data for realizable Fourier arrangements.

A footnote to this computation is that the code that found the 901 tuples for the extended Fourier 17 lines problem appears to have its own sense of humor: the one we managed to cull with this last constraint was 17th in the list.

Code for above:

```
paralleltotals = Map[Plus@@(#[[2,All,2]]
            *#[[2,All,1]])&,muboundedtuples];
singletontotals = 17-paralleltotals;
maxallowedmultiplicities = singletontotals
            +Map[Plus@@#[[2,All,2]]&,muboundedtuples];
maxactualmultiplicities =
            Map[Max[#[[1,All,1]]]&,muboundedtuples];
diffs = maxallowedmultiplicities-maxactualmultiplicities;
Length[Select[diffs,#≥0&]]
    900
Map[muboundedtuples[[#]]&,Position[diffs,a_/;a≤-1]]
    {{{{{6, 1}}, {{3, 3}, {4, 2}}}}}
```

Coding for the Parametrized Generalization of the Fourier 17 Problem

We have only considered the problem from the main article. It should be noted, however, that one can recast the code to handle arbitrary pairs (m, n) in place of $(17, 101)$. We provide an admittedly cryptic, but short, *Mathematica* procedure for this purpose, and illustrate on a simple example. It works by first generating a full set of data tuples such that summands are from the allowed λ and μ values. After that it prunes away according to the various criteria already described in order to obtain the realizable Fourier arrangements. This can be memory intensive if all one wants is the set of data tuples for realizable arrangements, and if that set is considerably smaller than the full set of all data tuples. That said, generating only the realizable ones via ILP tends to be slow. So we trade memory for speed.

```
SteinerData[m_,n_] := Module[{sum,bd,lams,mus,allparts,lam,mu,
           invcombo,fakes,allowedmu,allr,tuple,allpf,allmf},
sum = Binomial[m,2]-n;
If[sum<0,Return["No arrangements meet these requirements"]];
bd = Ceiling[Sqrt[2(sum+1)]+1];
lams = Table[Binomial[k,2]-1,{k,3,bd}];
mus = Table[Binomial[k,2],{k,2,bd}];
allparts = IntegerPartitions[sum;All;Join[lams;mus]];
Print[Length[allparts];" partitions"];
lam[l_] := Select[l,MemberQ[lams;#]&];
mu[l_] := Select[l,MemberQ[mus;#]&];
Do[invcombo[Binomial[k,2]-1] = k,{k,3,bd}];
Do[invcombo[Binomial[k,2]] = k,{k,2,bd}];
allowedmu = Select[allparts;Plus@@(invcombo/@mu[#])≤m&];
Print[Length[allowedmu]," possible arrangements"];
allr = Select[allowedmu;Function[p,sb=mu[p];
m-Plus@@(invcombo/@mu[p])+Length[mu[p]]≥Max[invcombo/@lam[p]]]];
Print[Length[allr]," real arrangements"];
tuple[l_] := Join[invcombo/@lam[l],{"|"},invcombo/@mu[l]];
fakes = Complement[allowedmu,allr];
If[fakes==={},Print["There are no fake arrangements"];
Print["The fake ones are ",tuple/@fakes]];
allpf = Select[allr,mu[#]=={}&];
Print[Length[allpf],
           " parallel-free arrangements: \n",tuple/@allpf];
allmf = Select[allr,lam[#]=={}&];
Print[Length[allmf],
           " multiple-free arrangements: \n",tuple/@allmf];
Print[Length[allr]-Length[allpf]-Length[allmf]," arrangements
           with multiple points and parallel families: \n",
Short[tuple/@Complement[allr,Union[allpf,allmf]],5]];
]

SteinerData[19,131]
```

```
4541 partitions
1808 possible arrangements
1808 real arrangements
There are no fake arrangements
31 parallel-free arrangements:
{{9,4,|},{8,5,3,3,|},{8,4,3,3,3,3,|},{7,7,|},{7,6,3,3,3,|},
{7,5,5,3,|},{7,5,4,3,3,3,|},{7,4,4,4,4,|},
{7,4,4,3,3,3,3,3,|},{7,3,3,3,3,3,3,3,3,3,3,|},{6,6,4,4,3,|},
{6,6,3,3,3,3,3,3,|},{6,5,5,3,3,3,3,|},{6,5,4,4,4,3,|},
{6,5,4,3,3,3,3,3,3,|},{6,4,4,4,4,3,3,3,|},
{6,4,4,3,3,3,3,3,3,3,3,|},{6,3,3,3,3,3,3,3,3,3,3,3,3,3,|},
{5,5,5,5,3,3,|},{5,5,5,4,3,3,3,3,|},{5,5,4,4,4,4,3,|},
{5,5,4,4,3,3,3,3,3,3,|},{5,5,3,3,3,3,3,3,3,3,3,3,3,|},
{5,4,4,4,4,4,3,3,3,|},{5,4,4,4,3,3,3,3,3,3,3,3,|},
{5,4,3,3,3,3,3,3,3,3,3,3,3,3,3,|},{4,4,4,4,4,4,4,4,|},
{4,4,4,4,4,4,3,3,3,3,3,|},{4,4,4,4,3,3,3,3,3,3,3,3,3,3,|},
{4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,|},
{3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,|}}
8 multiple-free arrangements:
{{|,9,3,2},{|,9,2,2,2,2},{|,8,5,2,2},{|,8,4,4},
{|,8,4,3,3},{|,7,6,3,2},{|,7,5,4,3},{|,6,6,5}}
1769 arrangements with multiple points and parallel
families:
{{9,3,|,3},{7,6,|,4},{8,|,5,3},{7,|,5,5},{7,4,|,6},
⟨⟨1760⟩⟩,
{4,3,3,3,3,3,3,3,3,3,3,3,3,3,|,2,2,2,2,2,2,2,2,2},
{3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,|,2,2,2,2,2,2},
{3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,|,3,2,2,2,2,2,2,2},
{3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,|,2,2,2,2,2,2,2,2}}
```

We make no claims about realizability of these data tuples. We believe it is a challenging problem to show, in any automated fashion whether a given data tuple is or is not realizable as a Fourier arrangement.

Concluding Remarks

We have shown that there are at most 900 data tuples that might be realizable, i.e., that might be the Steiner data of Fourier arrangements. The authors for [**1**] claim that they once showed the existence of 900 such distinct arrangements, and that those formed a complete set. Even allowing for the possibility that their checks were not exhaustive, at minimum they proved that there are at least 900 arrangements having $m \geq 1$ and $p \geq 1$. So we can safely conclude that there are exactly 900 Steiner tuples.

A modest remaining detail would be to (re)verify that each of the 900 tuples is the Steiner data for a Fourier arrangement. This appears to be a daunting task, because no necessary-and-sufficient condition for such an arrangement is known. The verifications described in [**1**] were done 30 years ago, and at their advanced age authors GA and JW are reluctant to undertake to repeat their earlier investigations. The present authors are about 30 years younger. DL has no idea how to go about the task, and sees this as evidence that computer technology has caused younger mathematicians to lose important skills. WW has ideas of how to do the verifications, but no time to pursue them. He politely refrains from mentioning that DL's skill level was never all that high.

Acknowledgments The first author thanks Jack Wetzel for considering the remote possibility that he might know how to approach these computations, and for providing the background material as motivation. We thank our ever watchful editor, Walter Stromquist, for doing the computational verifications independently (using a spreadsheet!! But don't ask us how.) They indicated, among other things, that an earlier formulation by DL had a flaw. We also thank him for stating explicitly the final necessary condition that reduced the list of 901 (all the way down) to 900 tuples.

REFERENCES

1. G. L. Alexanderson and J. E. Wetzel, Perplexities related to Fourier's 17 line problem, *Math. Mag.* **85** (2012) 3–12.
2. P. Flajolet and R. Sedgewick, *Analytic Combinatorics,* Cambridge University Press, 2009.
3. S. Roberts, On the figure formed by the intercepts of a system of straight lines in a plane, and on analogous relations in space of three dimensions, *Proc. London Math. Soc.* **19** (1889) 405–422.
4. A. Schrijver, *Theory of Linear and Integer Programming,* John Wiley and Sons, 1998.
5. Wolfram Research, Champaign, Illinois, *Mathematica 8*. `http://www.wolfram.com`