ON THE SIGN CHANGES OF ψ(x)−x

Rigorous verification of the computational parts of the proof

M. Grześkowiak, J. Kaczorowski, Ł. Pańkowski, M. Radziejewski

In this notebook we verify Claims 1–5 from Section "The proof of Theorem 1" in our paper "On the sign changes of ψ(x)−x".

Calculations are based on exact integer and rational numbers and on CenteredInterval objects, through which ARB ball arithmetic is available in Mathematica. The reader may note that the code in this notebook is not based on the code in the other notebook (that describes heuristic computations). Instead we took care to write it from scratch on the basis of the calculations described in the paper. The computation has taken 6 hours (mostly spent on Claim 6) on a MacBook M1 Pro.

By our estimates several GB of free RAM should be sufficient for performing the computation. On our machine the memory used by all variables reported at the end of the computation was 2.25 GB. However, in previous versions of this notebook we noted enormous memory use, in excess of 100 GB. As a result the Mathematica kernel would exit or, on one occasion, the operating system would panic. We could only explain such memory demand by a memory leak bug. The problem no longer seems to occur with the present version of the notebook. In any case, be warned that system instability caused by insufficient memory may have disastrous consequences. If you run the computation on your machine, e.g., to verify our proof, you should use an external monitoring tool to make sure that your system is running with sufficient RAM and in good condition. If you find that there is not the case, quit Mathematica immediately, by whatever (normal or emergency) method your operating system offers. Always keep safe backup copies of important data.

Initialization

We use rigorous estimates of non-trivial zeros of the Riemann Zeta function from the Flint library that incorporates the ARB library with rigorous ball arithmetic. The imaginary parts of the zeros that we use were obtained by running the example program *zeta_zeros.c* from the Flint library with the "print-_zeros()" function replaced by:

```
void print_zeros(arb_srcptr p, const fmpz_t n, slong len, slong digits)
{
    slong i;
    fmpz_t k;
    fmpz_init_set(k, n);
```

```
    for (i = 0; i < len; i++)
    {
        arb_print(p+i);
        flint_printf("\n");
        fmpz_add_ui(k, k, 1);
    }
    fmpz_clear(k);
}
```

The program was run as follows :

```
./zeta_zeros -n 1 -count 22 -digits 1024
```

to obtain the initial zeros in higher precision and

```
./zeta_zeros -n 1 -count 10000 -digits 24
```

to obtain 10000 zeros with about 24 decimal digits precision. The results were copied to the files “zetaZeros22.txt” and “zetaZeros10000.txt”, as arguments to CenteredInterval. These files should be placed in the same directory as the present notebook, so they can be retrieved when evaluating the cell below. We also use two interval representations of π from Flint (with higher and lower precision), obtained with the following code:

```
void print_pi(slong prec)
{
    arb_t pi;
    arb_init(pi);
    arb_const_pi(pi, prec);
    arb_print(pi);
    flint_printf("\n");
}
```

7/29/24 15:19:13 In[1]:=

```
$HistoryLength = 0;(*This setting should help reduce memory use.*)
piPrec = CenteredInterval[
  4 990 052 726 770 977 142 154 679 508 178 589 889 026 388 891 990 740 340 505 428 641 215 \
   669 648 447 421 656 627 380 065 252 340 145 932 778 560 702 192 687 029 448 418 415 483 \
   188 954 586 727 357 955 236 838 483 991 113 515 699 969 191 031 129 196 856 848 330 876 \
   927 287 688 269 763 941 393 540 346 541 910 255 624 991 002 971 791 567 097 235 913 963 \
   946 153 048 394 165 591 968 593 437 619 415 761 280 434 402 460 617 709 335 929 051 394 \
   725 822 723 441 471 537 908 256 673 593 130 944 875 014 938 974 955 440 491 292 278 510 \
   804 581 427 338 029 928 540 776 104 319 872 787 747 242 672 692 358 303 125 055 316 299 \
   013 149 608 140 037 453 788 788 861 053 453 199 220 647 499 088 114 236 542 334 675 035 \
   520 389 564 613 678 933 034 002 688 158 602 236 053 046 439 470 764 530 476 520 728 537 \
   596 707 628 174 140 200 257 454 951 588 914 599 592 521 098 835 816 387 920 280 595 104 \
   690 590 499 662 564 424 656 234 943 773 885 248 132 328 562 021 350 799 214 827 268 621 \
   127 807 161 305 325 915 559 963 794 429 078 077 773 411 108 687 974 437 046 305 261 723 \
   935 679 211 260 929 550 307 843 916 995 846 682 043 220 905 771 616 446 457 676 411 836 \
   870 450 668 057 551 979 363 072 001 440 485 851 178 797 768 327 911 193 291 885 089 110 \
   179 181 432 266 964 257 177 844 093 949 635 828 223 079 277 306 912 251 336 430 677 153 \
   648 644 776 417 018 539 522 154 206 415 953 * 2 ^ - 3399, 536 870 912 * 2 ^ - 3432];
pi = CenteredInterval[1 898 976 236 818 169 290 393 997 * 2 ^ - 79, 536 870 912 * 2 ^ - 110];
initialZeros = 22;(*Number of initial zeros with higher precision.*)
numZeros = 10 000;(*Number of zeros computed with lower precision.*)
initialImZero = ReadList[StringJoin[NotebookDirectory[], "zetaZeros22.txt"]];
imZero = ReadList[StringJoin[NotebookDirectory[], "zetaZeros10000.txt"]];
```

Additional verification of the zeros of $\zeta(s)$

We check that the zeros of $\zeta(s)$ obtained from Flint agree with Mathematica's built-in function ZetaZero[k] and, in the case of the initial 22 zeros, with the estimates published by Andrew Odlyzko.

Zeros from Mathematica

7/29/24 15:20:29 In[8]:=

```
highPrecision = 1028;
Monitor[initialImZeroMMA =
   Table[N[Im[ZetaZero[k]], {Infinity, highPrecision}], {k, 1, initialZeros}],
  StringJoin["Computing the initial zeros of ξ(s): ",
   ToString[k], " of ", ToString[initialZeros]]];
passed = IntervalMemberQ[piPrec, N[Pi, {Infinity, highPrecision}]];
For[k = 1, k ≤ initialZeros, k += 1,
  passed = passed && IntervalMemberQ[initialImZero⟦k⟧, initialImZeroMMA⟦k⟧]
 ];
Print["Are the initial 22 zeros (computed with high precision) returned
    by Mathematica inside the intervals that we use? ", passed];
Parallelize[
  imZeroMMA = Table[N[Im[ZetaZero[k]], {Infinity, 28}], {k, 1, numZeros}]];
passed = IntervalMemberQ[pi, N[Pi, {Infinity, 26}]];
For[k = 1, k ≤ numZeros, k += 1,
  passed = passed && IntervalMemberQ[imZero⟦k⟧, imZeroMMA⟦k⟧]
 ];
Print["Are the initial 10000 zeros returned by
    Mathematica inside the intervals that we use? ", passed];
```

```
Are the initial 10000 zeros returned
   by Mathematica inside the intervals that we use? True
```

We needed to ask Mathematica to be slightly more precise than Flint, so Mathematica’s result is inside Flint’s interval. Actually, for higher precision and higher zeros we would need even more extra digits, because Mathematica’s built-in ZetaZero[] tends to return results with accuracy slightly below that which is requested. To see this, compare Mathematica’s output to itself, with two different precisions:

7/29/24 15:21:53 In[38]:=

```
N[Im[ZetaZero[9998]], {Infinity, 128}] - N[Im[ZetaZero[9998]], {Infinity, 140}]
```

7/29/24 15:21:57 Out[38]=

$-1.6006559355538710203061 65 \times 10^{-103}$

Zeros from Odlyzko

Zeros published by Andrew Odlyzko at https://www-users.cse.umn.edu/~odlyzko/zeta_tables/index.html are claimed to be accurate to 1000 decimal digits, but the accuracy is in fact a little higher.

7/29/24 15:21:58 In[39]:=

```
initialImZeroOdl =
  {14.134725141734693790457251983562470270784257115699243175685567460149963429⋱
    809256764949010393171561012779202971548797436766142691469882254582505363 2⋱
    394471377804133812372059705496219558658602005555667258360107737002054109 8⋱
    266150754278051744259130625448197865107230493872562973832157742039521572 5⋱
    674809332140034990468034346267314420920377385487141378317356396995365428 1⋱
    130796805314916885290678208229804926433866673462332007875876179200560486 8⋱
    054356801444424651065597568665903228686510544859444320624072727032094274 5⋱
```

```
 2221304874872092412385141835146054279015244783383542545334400448793680676\
 1697300819000731393854983736215013045167269683892003917628512321285422052\
 3969133425832275335164060169763527563758969537674920336127209259991730427\
 0756830879511844534891800863008264831251691127106829105237596179774318151\
 7071354531677549515382893784903647470972701994848553220925357435790922612\
 5247736595518016975233461213977316005354125926747455725877801472609830808\
 9786007125320875093959979666606753783812148919088649772775544206565320524\
 05,
21.02203963877155499262847959389690277733434052490278175462952040358759858 6\
 0688907997136585141801514195337254736424758913838650686037313212621188216\
 2437574166925654471184407119403130672564622779261488733743555205914739713\
 2822662470789076753814440726466841906077127569834054514028439923222536788\
 2682361112892700575856532731588666042140009071151080090069720027998711017\
 5847519632216496865900574811247938691638351837234278073449023910103850457\
 5641215958399921001621834669113158721748057170315793581797724963272407699\
 2211256634415618236051804767144227146555596737812477650045558409086442916\
 9775704638165517749644524987674237036645657770483799202927066431583789323\
 8009151146858070430828784147861992007607760477484140782738907003895760433\
 2451278278637209093037972518237091808042306667383437990228251582878876176\
 1266187138296785874562376500666242078081451763697639137434059341279754969\
 7276850306200263121273830462939302565414382374433344022024800453343883072\
 8387312602306547534837868011827893175200106900560165441528110509706375932\
 28,
25.01085758014568876321379099256282181865954967255799667249654200674509209 8\
 4416442778402382245580624407504710461490557783782998515227308011881339335\
 8267168958722516981043873551292849372719199462297591267547869662885680773\
 5070039957723114023284276873669399873219586487752250099192453474976208576\
 6123345997354435583675313812659977645290374484969947911378977220661993071\
 8997232254973227163005159161921279774087660006729149830812793066702735084\
 9516001984670542469491796695225514179319665391273414521673160233737754489\
 4146417119378489574997514110658562879690076709862827218649537296323925840\
 3491387143048933588946114958624239036855617518935987873568568308927144446\
 8756375337019130417377142535868018531867896375326868632660719766920532953\
 3478506707982877118674944281439725425516531967977991272268445896927940859\
 9507227960513612021369680647653397626969177425124909525721400385588649442\
 2730332216278403670865759210329078986615602048427519273514192759701784916\
 6084411074821559128310749314226402783395134287731266441051685710163442899\
 02,
30.42487612585951321031189753058409132018156002371544018096214603699332938 9\
 3332779202905842939020891106309917115273954991176332266711863193918072259\
 5671424334115590685468136558072417349844724959319040811632315019702348484\
 1630221400985620739718392018133021868063298225719752250023746856136974712\
 4964426229779245040574906715345727886515065160832468797062817781045777722\
 5878919237386290011276030973568089049253006461289272753091944790200358938\
 9819427495511323917384271638108400499211198006924387188729695970002910005\
 4774270689081684625934838507707996560373392659163178590055839059681572073\
 0796252620549400959515892318195507003120438547291284707373793170005246046\
```

```
 9858203860095171051337905912538151203525649548068653947457306442869841989\
 0124742762009249476736375814720332208668760145726577740711967273435047923\
 4503516187981145579444869326121291441791658325190186784986764477772964821\
 5979712565041026341481014213352401333833266814485615449144877122011828407\
 0765164762211312808070237683310170970227228331540528509637318716195825137\
 81,
32.9350615877391896906623689640749034888127156035170390092800034407848156 08\
 6305510059388484961353487245496025252805975815135792377828577545060376530\
 1147268210982527271365947816607918650788117035383676547460173854812065178\
 7886596466594728787186027971658042677648544066692909393193115645508391751\
 3007902799895626566839200248741651699908688450128764042509560641448930237\
 7479705325278240994775176594460744680987406706533829095891561420480022419\
 8408375141584435237806584753908207901246070504560100102257438966836495384\
 2377042049095598981489088287256372979657214165263177857552011550777840579\
 5014772072579821404213096863779363536418594023958569909002682666321481934\
 3540824642411198675065226034566669947805384925779001837592451461820738574\
 3601378106373715610378620890222774133204177346434236286396297362844455424\
 6794670572412945536985966772759610779950109259850113657198032734505406239\
 1125982107996408750156764769596419463847304800451155728883660650892331803\
 4977150164443483206561234793403070632305551877737810242105618082026092962\
 18,
37.5861781588256712572177634807053328214055973508307932183330011136221490 89\
 6185372647303291049458238034745777477461922353799409650222639362856248220\
 4748368808900366789035453755377790319090564409937583872127563431264305150\
 7490897122077467801551204327992986651432331546548850586795060341362549259\
 9668820961532811560379952622032357848783110737761462301491055793790156771\
 6341125126537129565017674488645214983270201198449431532620803315317760746\
 4795075040560820830213633950862288220551335769555803593746481598709080074\
 0796272177164722740705738059683514846795303217276087112400673794318336706\
 7907130787878538857435773069511259743788562762143489137407599767440555578\
 5944574435456054998502810248039697965044333212466041270340603151340824120\
 3980669209295582721330805747422403795269414475603871713914177228238622773\
 3350407525960009283974067845039605557239933245709984208543930857839082705\
 8000428133328048210695855943079835422972275055349593563168097527417304301\
 2765178378868858782566744124564450930669445480799366125006791488466776116\
 35,
40.9187190121474951873981269146332543957261659627772795361613036672532805 28\
 7200712829960037198895468755036655196769454679449171364948985991756534367\
 5331050480915216875448752277922007000739719484178056445522301052855461197\
 9757902870894436727404629995409604733040969526221562845948429944927846184\
 5787877616326789854362772538625569030381839732705149501966655277103497437\
 4881669599052859103048206296364783288020522199731882565801235122140906719\
 1689558103853574341088997924759060345757218996039302094745965584290912923\
 5663707071882855846122693898158425523031107154470055342115390121180247552\
 4892024185390070008111883657554115086785606764534451142570203587461003801\
 2461701591217504665556870944439231103353127472397717165752437859003356150\
 4617157300246478469771314844670423816353203684984218574021139833549142712\
```

```
 6770287667947736204693399337830596189087702647022523444329909291375072223\
 2251216747711999592944915975281247409735591880537231769464032357313745507\
 6757717063510761817833042233227870452997287793116686793918832400230416784\
 98,
43.327073280914999519496122165406805782645668371836871446878893685521088322\
 3050536264563493710631909335548676084926268024471670123258547521235346630\
 6785286334617458228386331819781478769573446059047776859114649030017546802\
 6853027496697613091256439069665846196373054906998801694436519162899527326\
 9773312247402902902297055676432520156730365804068188556749480980178537049\
 7581536430714937276101770687867843354563190301812585120071477024107320557\
 1081343008914241615264552855237368821808322357074175925403682783202820844\
 5814028293633182547608665486774820590194315839107341956876490489700004288\
 9258721522434470005511633873189592197729287140590326873261677558413088187\
 5117328095378031093324495736965067920017652435206447838972973982815035687\
 9333910833700001892689617081051811669487471371749028348818145021241036348\
 8879382252302918527130538241663122536442258089958938099855749363680285940\
 8915387487451045248123666445118287217598562223212440326052361942860828927\
 5823558829645730213149084494081190523167790536892637612697271839643425025\
 57,
48.0051508811671597279424727494275160416868440011444251177753125198140902 16\
 4163082813303353723054009977155029777026020216265216627081359729316663878\
 7584836041574951454259439317191246640774218280793155659220339501510554415\
 9539882294586912180255358282132516378822242517229114957339065901571071341\
 6095467176156220195774300013719011808928139407455668413118972936793388848\
 4316954851088813489262277107938612544530655545258160430217317139401667621\
 9848596815386025310508650393065848664659529280290536587847109504846578979\
 8457521607470313224599286241266689136188456298675352455750858976597097002\
 4848881629220431378558854043457062763837664929361306060021454214048045407\
 5879271396728375556083478743824757976591799995228735986325938128949255347\
 1583237220000740221113625220524010126097275534601512284619421918777726973\
 9154990915090231860617311481136258931376508836671972526149786110203056722\
 7613320075475161086258162488087245079880549023530434510965723869519500307\
 6553095179526905675823924093467427293377696679845143234796517856597810125\
 85,
49.7738324776723021819167846785637240577231782996766621007819557504335116 11\
 5157392787327075074009313300707816280215611312243579991641163553140754514\
 9058991598474641116416378253783316653929728579548583391580639466325973644\
 4259036153975727355045500020189054770646843216398460410248730439673334416\
 1342965845823735611040852868361709871082539438060721579454849699494800654\
 0918439686532222948796598716534544810965172932440340523209113196215050595\
 4273947996858231814508704434773134221635876244264109437986101224081986059\
 3576100158283712613580480827333877570256578316446221984288672848119700164\
 5873088760875901736619771716993925244004000270494841883043002967932231995\
 5569628185655747157275942398769858510515539714341772965634674500030547079\
 9040139414934213133581028145037483991582551715260503387148465442689767950\
 1125393237405578331351474174485457769332302096309726741003227137015651251\
 1527718777100048072946537052793939826953074619178922511068291080412495755\
```

```
 1218458516606862288895527929277641052402584586310872653923196994729460862\
 95,
52.9703214777144606441472966088809900638250178888212247799007481403175649502\
 3041880541375878270943992988036388447753550202803052421425267597188524267\
 2950180289906415888440438135739014789887680560136071518461097272566357267\
 8982128874175943840112488882136436086401051818563432220986937961340888420\
 1154942364604552762689508527444786135051412029272085078607964997879116448\
 3349624631557376419945375193365284029438097810895657642120389690368737048\
 1919718627137671317419749396133269854256578464831198077078267156524968563\
 2681044754749000526481157760442462146512865023217484065591489114589222899\
 9709589857400374635307884636881895827912565950620145968420674343117174667\
 1164326328999484488107227524047065005394806556803123856288675947364461841\
 0658242003300237765895776548423006959264657068039752865366539653304651943\
 9064518617287094645038380888323492031442848557863853764285214083721969132\
 8838062211432628271565436562767613276667483291737345439668735430085880540\
 5462914024384564894932041398454762898224901795640662618056856317201297865\
 65,
56.4462476970633948043677594767061275527822644717166318454509698439584752802\
 2745056669030113142748523874789718788336824582795338524154404941426773315\
 3585138991230314395097767479641842821514159045327976789706742180219023835\
 1965802772317149622477373168417648321033460697211675520831682748713648990\
 1262222505988802295235691061754578319556056254996809339748779603855421938\
 9886924898528735105142126119743109235813409901360470923914103110900943331\
 9473584228704879584556102786577358526283178106677724885875960578484157909\
 1516830553398711320111144364926784937155000470750866724249255729541317026\
 1546303183738764083693695018672303906883645055000732786798099987006561769\
 5185049360601192737525309533662000989316467491978423470677395500748300974\
 1342696859458372422014268786929451616489043921088771795888322428920997738\
 8178124689431218750153112448777217164717094786897094730480532384784727617\
 2776766067483112107631211790935473076225276136187306347951724637196712061\
 7079783174578416164171076871586600133652585271559672834254093874386792124\
 49,
59.3470440026023530796536486749922190310987728064666696981224517547468001522\
 6996298118381024870746335484385219585474392435432309183323355694184073005\
 6212173898990762113221776823974005679057204245284952644084968147273303862\
 3910280825310642382206849929802500376003172490143770767704346845812728985\
 8507450877484205692080966151785890579595230069830537243323336634723312065\
 5620493429101526059868606661904807070322117772189346119993360575793382266\
 1462457047598564940384489397956778352154200137916652305382822464848056286\
 3977729330816412758624097913914949458849921242546948106860379538264753100\
 8313688063664107660566694547444153561332964561917442336947964312653180429\
 4010965039525227559520795006567279336501268658748905330228091635965020782\
 3699559677865941612819696303852902789825080365103307453044195631883280737\
 1572169474039139487368488830073351320671805312616966326530583332127493834\
 9369740133741459513946477564549783261202953121314681877483549199357103776\
 1615170836287278993478840806711599539107232378154126962211537055772285856\
 49,
```

```
60.831778524609809844259901824524003802910090451219178257101348824808493667\
 2949205384308416703943433565698644703358868974503074794863877571512369896\
 7776718139290397255489366396360909670877412216683886903611382323813732966\
 9465456300338044996481742040533940870976129782940984400188964842502151111\
 1335393408358670066060965782823202082217315576471900534967997272830859234\
 9540264307522501844379215363719536484059777311089660786592542547816699132\
 3631938667833621762047903805963775756066867271977573019725620641617070717\
 8812019848519403992194281810377895582810135440583275681298573133845470624\
 4539956725704169790538477388529502218992292558538763926593989497317775525\
 2302791322676359672466519865985965716901132288333512324701288146549659734\
 6487335967338892329237591168704851245684367933179773704687201401755657226\
 2217261937618488808601017640213544213837209390172119611738629537275575411\
 0553487919760743047599818014270992473679191478534947036556659099682425949\
 9792854173088536996358846214128434415798293639364519211545032267683236651\
 23,
65.112544048081606660875054253183705029348149295166722405966501086675343232\
 6686853844167747844386594714258969277516620367277922125797604065267103417\
 1283933565046639547411379240773872974559426747241380152029155597066133382\
 4803985659298979057804375398625149837486331063458826984693672376060806786\
 2230890462794040102508271386380493017633384022265897398597328027557813685\
 1111710976322435129211257657486537550927042702117799091409893902356671496\
 0805022355015948341568367529956845298905085088674588430969696692059922085\
 7500229868451452688961404564647834802286322605064701766683725597371637412\
 5652648712584815267646275045030436526244129075772975393425193387957004511\
 6745923042841923035167267716672756004106536974278267866541063812451435638\
 8600898540857376548561101586374936194640003762811917719050359106347531129\
 5565025492793570208394004511746930552865641250898275959826183652469969831\
 4147407660309343740846980458044034701255293332123418613818055529567556639\
 8345321215173088521263141020101101709145780391241429281338997400866136234\
 43,
67.079810529494173714478828896522216770107144951745558874196669551694901218\
 9561969835302939750858330343055547010051348596310291518960431718158673540\
 2319454836737904008670291478098179441967844612438143836175844936243239659\
 5448051672339388685066004214583341246451635021345298388413847073026983988\
 2125899938264238700852450998975644296673551042898786207188540725885465155\
 7397345970707167627310857633493218077257913000396088644974183686871535394\
 7764759128350375674332037097265984118710070440446053451664131052421556680\
 2652357850722652744388767856266960386974820520968223744185078179438502244\
 3346678617423741976737378730996368068876529569031110880421880736675448124\
 4754126423472826840008964978766958987469469952222948151276077984728912688\
 0047881658781961721948027767838630535975211178160134229958413426046335817\
 1644732461635274593378457073043033648448128008407372455929232116261413024\
 7542832542954781778971577652485927891091400797975856467160901917331051889\
 5403576569384834692070360917708740907981969142442377934743129604651617190\
 18,
69.546401711173979252926857526554738443012474209602510157324539999663387672\
 2749104195333449331783403563537480615847867891662561228684053352270151764\
```

```
3218746616386794929305333020563595599351980343958557595114999333835451392\
5577815476563818784672545960927472810403977949465301956954113198287899441\
2753905672991271864832744271617424045807589981725965324007713291151057000\
5963350748931877639857880962727412531373070348605066644539794040423092372\
6906435472678042455592779768290482541588859320129640662501737686841432312\
9556899153220501024033603483047934941701713857948772790489315608242116544\
8412151977717797234915511215637068664397988905575321247039075738058661060\
9056618978691276590975660299730990890514703412420060488566164142937612261\
4869634923597697968639334070199501957285853017886819566252215763136466802\
7995122945624622749490791784679503965601776769920608041082232411935612382\
5002391567773007065401074770394762661359198236961182883729506234855468174\
9625326185773659999280661081460239774965292417164318324608187881348356301\
22,
72.067157674481907582522107969826168390480906621456697086683306151488407372\
3996083483635253304121745329891573665831188308364239393239648088245584888\
8390855996676745888302674866719101010189535097209841031377167227486379958\
5130686177614665616376998197197669271905364262584998184369329731220210017\
2154163954863017585480124158635196955928931519232721117911651740802773603\
1669647282433100951823695576425029578747315120388475431975347290363276086\
9531588040490159640781358830072069132830007220361644959698406819333476590\
7182841824432110585530665809073961287912633510238909400412190297500911577\
3972136661480424590673282451940522374239504901434210591685614510962039466\
6086341141283207680069777118300539680261625685207183611688073563664855004\
6348733029368913372900633109933374126695507110336609629644621044246846713\
3375110923568627214733592987546300167277187444347825424545900144210465131\
1737080519674798616116617535063002446065244825392505742634946337840098855\
1322243095295782938216077702272394727245397704091317929516827593398449294\
14,
75.704690699083933168326916762030345922811903530697400301647775301574197027\
7063236083840370218346527980499440236875710049412993774579369232403249190\
8337608653429942953940301154342853588730675266753241741687953454395704964\
1711175534056656582788719787111839068903024040573809118225198707500000640\
6037591172211976280687762314536434325622731027601407851235714067127323713\
2730061216897487596676471021538839444591114631103053505397493363056826449\
6513909943445091170178315452780762798464236397482211548344718763384623992\
3182923970854816125090190741776725754285776705594419507317270474617771281\
0677181652786676062580447322428418181138436976586364170208850182691428655\
7271757830441891394556842493793565722002357598861902102493142375562290110\
0527807090808863770943428408074977092780583880750276282870614816713452502\
2453461379665321899606042551464172526785097516858175701039881387610207413\
9793312211689814244076676024176863363672137102926551762356450984387409050\
7191863871742329170449070736629201753397701148772829715832014572141648871\
29,
77.144840068874805372682664856304637015796032449234461041765231453151139164\
2537150894082886946997377597590568744126898627394010150434487147913186850\
6720926250969244950056959854287699352594952938665551590789073219964894771\
2250820352478570207374012914861152103578929079678094268718586402341944783\
```

```
    86034670643081088090942680986988205845767919076871222668732683601467591391\
    06597219780274184138815734306541730015512005219415994953054177649034287071\
    07929705495185221781036691384931104002899087631258387479671049344254090451\
    72905371623854289453973469793840446751784565398284287399685049368194419151\
    93492637849172325367363215884466442942711922492108245169377647915815628831\
    44694853466863158562312432953236095382386761461136018427034327078557225621\
    39351383397429617090576152953122686067367349780236208012084306894188054421\
    94812645599285786304342142860013808561146024088198620632042297490278719391\
    76034947393348985146738243834709639961622368639046161611075776113366265431\
    22311795251777112531327497197699035245098151783349602469986425595902809311\
    18,
   79.3373750202493679227635928771162281906132467431200308784387204971015419321\
    67709097467745199461212410901477102377975038492265220686128115682860498921\
    06199709970777729349241193116002036546210829938880840595591024165836178951\
    86109623373707037950740704951643044756908532806669266186551732281699312941\
    82847836020286280349024764642472679972938728433263438362867949883841889441\
    91711089737165416455496333704114154400493984706855811753017890603105118291\
    65731855731353466960613390248405660126407041359995615006709455437342235341\
    03132439618116805189890290340377049721553300180805379622407616948086552991\
    24868274022835773560748385236594010125036441641139088681713752156567009341\
    46001983354395248701390872637667660234886071658616203232579740824970491611\
    98741334220532482928433255181938639760270524679170076372209117105367004061\
    07336954316681544251161987389754932964386820464175476639079150107912316631\
    87122862689108906083857585620659916980918649069071181463571299621233352771\
    14140103989229666585257078847014737385811310240528545170308782370962124021\
    31,
   82.9103808540860301831648374947706094975088805937821491465713062832359290861\
    35661907551256319233489681875525448176186010046123385280735957470697143341\
    97862880915545409209829902744647007546820535138230602538882436385604843741\
    20390670502801114963769217470821044501340563344292111265724756018589047501\
    20174821554974854721319512519347967895128797541125323948460233832938813491\
    80206848937216850204264250309056404395393371604415437958809279859505591881\
    83036514819713524408681587696372641694411047589319713170752623560820739941\
    61855883326538246938837000498648141993768593314465032640369798804147322561\
    17258868002876866036602248640131077429112590790286497528478168503567063431\
    49629953851409617643479451007896612996121015316669111268051866246294867921\
    43500602103342636277665646183136074032165621569422719831423031803156134201\
    80125413685783092636639888303766029667579048491094809757291004908265664701\
    28410208541277571026039291757877230524949394340488631725193804054302996861\
    90607760302136916777823780730716978222610617431979378446413084528326186761\
    68};
error = 0;
For[k = 1, k ≤ initialZeros, k += 1,
  error = Max[error,
    Abs[Information[initialImZero[[k]], "Center"] - initialImZeroOdl[[k]]]]
 ];
Print["Maximum difference below 10^-1022: ", Abs[error] < 10^(-1022)];
```

```
Maximum difference below 10^-1022: True
```

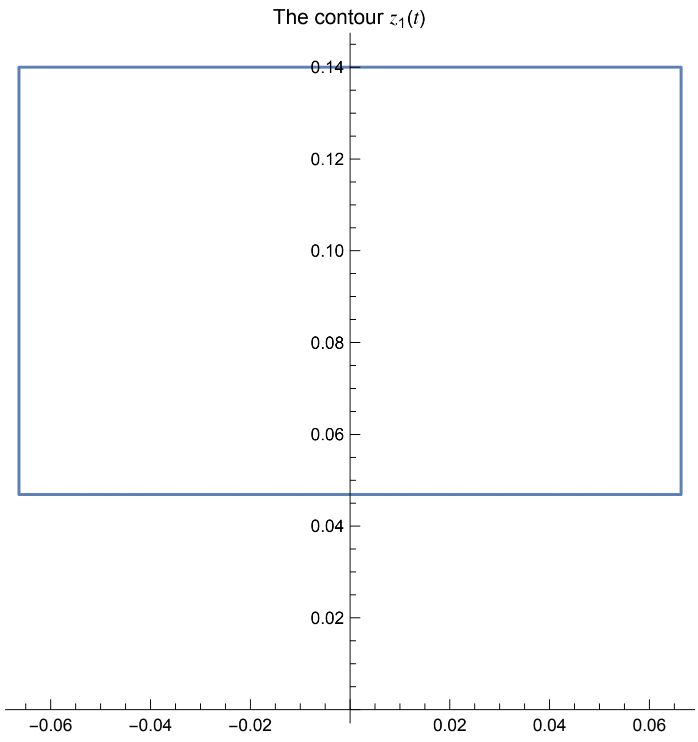
Claim 1

We have $\left|\alpha_j - \alpha_{j-1}\right| < \pi$ and $\left|z_1(t_j) - z_1(t_{j-1})\right| < \frac{\pi}{\omega_N}$ for $j = 1, \ldots, m$.

As a rule, we try to represent all the data as it appears in the paper. Unfortunately, we are forced by Mathematica to use 1-based array indexing. For γ_n, ρ_n and ω_n we use functions to simulate 0-based arrays, but for the arrays z1 and alpha we need to remember that indices are off by 1. As Mathematica reserves all names starting with capital letters, objects named with capital letters in the paper are preceded with c, e.g., cN for “capital N”.

7/29/24 15:22:07 In[43]:=

```
cN = 21;
gamma[n_] := imZero[[n + 1]];
rho[n_] := 1 / 2 + I * gamma[n];
omega[n_] := gamma[n] - gamma[0];
cFNeta[z_] := 1 - Sum[Abs[rho[0] / rho[n]] Exp[I omega[n] z], {n, 1, cN}];
f[z_] := -Sum[I omega[n] Abs[rho[0] / rho[n]] Exp[I omega[n] z], {n, 1, cN}];
delta = 132 737 / 10^6;
y0 = 468 918 / 10^7;
y1 = 14 / 100;
m = 12 000;
z1 = Join[
   Table[4 t delta + y0 I, {t, 0, 1 / 8, 1 / m}],
   Table[delta / 2 + (y0 + 4 (t - 1 / 8) (y1 - y0)) I, {t, 1 / 8 + 1 / m, 3 / 8, 1 / m}],
   Table[-4 (t - 1 / 2) delta + y1 I, {t, 3 / 8 + 1 / m, 5 / 8, 1 / m}],
   Table[-delta / 2 + (y1 + 4 (t - 5 / 8) (y0 - y1)) I, {t, 5 / 8 + 1 / m, 7 / 8, 1 / m}],
   Table[4 (t - 1) delta + y0 I, {t, 7 / 8 + 1 / m, 1, 1 / m}]
  ]; (* i.e. z1[[1]],...,z[[m+1]] represent z1(t0),...,z1(tm).
*)
(* The argument function, denoted as α(t) in the paper,
is called argFun[t] here. *)
argFunBreakpoints = {0, 1 / 32, 1 / 8, 3 / 16, 3 / 8, 1 / 2};
argFunValues =
  {0, 489 819 / 10^6, 185 802 / 10^5, 204 829 / 10^5, 290 189 / 10^5, pi};
argFun[t_] :=
  Simplify[Piecewise[Table[{(argFunValues[[j]] * (argFunBreakpoints[[j + 1]] - t) +
          argFunValues[[j + 1]] * (t - argFunBreakpoints[[j]])) /
        (argFunBreakpoints[[j + 1]] - argFunBreakpoints[[j]]),
      argFunBreakpoints[[j]] ≤ t ≤ argFunBreakpoints[[j + 1]]}, {j, 5}]]];
alpha = Join[
   Table[-pi + argFun[(j - 1 / 2) / m], {j, 1, m / 2}],
   Table[pi - argFun[(m + 1 / 2 - j) / m], {j, m / 2 + 1, m}]
  ];
alpha = Join[{alpha[[m]] - 2 pi}, alpha, {2 pi + alpha[[1]]}];
(* i.e. alpha[[j+1]] represents αj in the paper. *)
Print[ComplexListPlot[z1, AspectRatio → 1, PlotLabel → "The contour z1(t)"]];
Print["Claim 1: ", Max[Table[Abs[alpha[[j + 1]] - alpha[[j]]], {j, 1, m}]] < pi &&
  Max[Table[Abs[z1[[j + 1]] - z1[[j]]], {j, 1, m}]] < pi / omega[cN]];
```




```
Claim 1: True
```

Claim 2

We have $u_j > 0$ for $j = 1, \ldots, m$.

7/29/24 15:22:11 In[60]:=

```
b = Table[(1 / 2) Abs[z1[[j]] - z1[[j + 1]]], {j, 1, m}];
xPrime = Table[Min[Re[z1[[j]]], Re[z1[[j + 1]]]], {j, 1, m}]; (* x'_j *)
xBis = Table[Max[Re[z1[[j]]], Re[z1[[j + 1]]]], {j, 1, m}];
(* x''_j – we use the Latin "bis" instead of "double prime" *)
y = Table[Min[Im[z1[[j]]], Im[z1[[j + 1]]]], {j, 1, m}];
cM = Table[Sum[
    Abs[rho[0] / rho[n]] omega[n] ^2 Exp[-omega[n] × y[[j]]], {n, 1, cN}], {j, 1, m}];
cNPrime = 9998;
cT1 = (gamma[cNPrime] + gamma[cNPrime + 1]) / 2;
cRN[y_] := Sum[Abs[rho[0] / rho[n]] Exp[-omega[n] y], {n, cN + 1, cNPrime}] +
   Exp[(gamma[0] - cT1) y] * (Abs[rho[0]] / cT1)
    (Log[cT1] / (2 pi y) + 4 Log[cT1] + 2 / (cT1 y));
yUnique = Union[y];
Monitor[cRNLookup =
   Association[Table[yUnique[[j]] → cRN[yUnique[[j]]], {j, 1, Length[yUnique]}]],
  StringJoin["Precomputing R̃_N(y): ",
   ToString[j], " of ", ToString[Length[yUnique]], "."]];
Monitor[u = Table[Min[
       Re[cFNeta[z1[[j]]] Exp[-I alpha[[j + 1]]]] +
        Min[0, Re[b[[j]] × f[z1[[j]]] Exp[-I alpha[[j + 1]]]]],
       Re[cFNeta[z1[[j + 1]]] Exp[-I alpha[[j + 1]]]] +
        Min[0, -Re[b[[j]] × f[z1[[j + 1]]] Exp[-I alpha[[j + 1]]]]]
      ] - b[[j]] ^2 × cM[[j]] / 2 - cRNLookup[y[[j]]],
    {j, 1, m}],
  StringJoin["Computing u_j: ", ToString[j], " of ", ToString[m], "."]];
Print["Claim 2: ", Min[u] > 0];
```

```
Claim 2: True
```

Claim 3

We have $\beta''_{n,j} - \beta'_{n,j} < \pi$ for $n = 1, \ldots, N$ and $j = 1, \ldots, m$.

7/29/24 15:32:16 In[72]:=

```
(*We have to define our own rigorous floor and ceiling functions,
as Mathematica does not seem to support Floor[CenteredInterval[...]].*)
floor[ball_] := Module[
   {x = Information[ball, "Center"], n},
   n = Floor[x];
   If[n ≤ ball,
    Return[n]
   ];
   Print["Indeterminate floor of ", ball, "."];
   Return[Floor[ball]]
  ];
```

```
ceiling[ball_] := Module[
   {x = Information[ball, "Center"], n},
   n = Ceiling[x];
   If[n ≥ ball,
    Return[n]
   ];
   Print["Indeterminate ceiling of ", ball, "."];
   Return[Ceiling[ball]]
  ];
norm[ball_] := Min[ball - floor[ball], ceiling[ball] - ball];
Module[{var = ""},
  Monitor[
    var = "β′n,j";
    betaPrime = Table[Table[
       pi + omega[n] × xPrime[[j]] - alpha[[j + 1]],
       {j, 1, m}], {n, 1, cN}];
    var = "β″n,j";
    betaBis = Table[Table[
       pi + omega[n] × xBis[[j]] - alpha[[j + 1]],
       {j, 1, m}], {n, 1, cN}];
    var = "φ′n,j";
    phiPrime = Table[Table[
       Piecewise[{
          {0, ceiling[betaPrime[[n]][[j]] / (2 pi)] ≤ floor[betaBis[[n]][[j]] / (2 pi)]},
          {2 pi Min[norm[betaPrime[[n]][[j]] / (2 pi)], norm[betaBis[[n]][[j]] / (2 pi)]],
           ceiling[1 / 2 + betaPrime[[n]][[j]] / (2 pi)] ≤
            floor[1 / 2 + betaBis[[n]][[j]] / (2 pi)]}
         },
        2 pi Min[norm[betaPrime[[n]][[j]] / (2 pi)], norm[betaBis[[n]][[j]] / (2 pi)]]
       ],
       {j, 1, m}], {n, 1, cN}];
    var = "φ″n,j";
    phiBis = Table[Table[
       Piecewise[{
          {2 pi Max[norm[betaPrime[[n]][[j]] / (2 pi)], norm[betaBis[[n]][[j]] / (2 pi)]],
           ceiling[betaPrime[[n]][[j]] / (2 pi)] ≤ floor[betaBis[[n]][[j]] / (2 pi)]},
          {pi, ceiling[1 / 2 + betaPrime[[n]][[j]] / (2 pi)] ≤
            floor[1 / 2 + betaBis[[n]][[j]] / (2 pi)]}
         },
        2 pi Max[norm[betaPrime[[n]][[j]] / (2 pi)], norm[betaBis[[n]][[j]] / (2 pi)]]
       ],
       {j, 1, m}], {n, 1, cN}];
    ,
    StringJoin["Computing ", var, " for n = ", ToString[n], "."]];
```

```
 ];
Print["Claim 3: ",
  Max[Table[Table[betaBis[[n]][[j]] - betaPrime[[n]][[j]], {j, 1, m}], {n, 1, cN}]] < pi];
Claim 3: True
```

Claim 4

There exists a matrix $C=\left(c_{k,j}\right)_{\substack{1\le k\le N\\1\le j\le N}}$ with integer coefficients such that $\sum_{k=1}^{N}\left|\sum_{j=1}^{N} c_{k,j}\,u_{j,n}\right| < 1.66\cdot 10^{-13}$ for $n=1,\ldots,N$.

We perform an adapted LLL algorithm, described as Algorithm 1 in the paper. Some of the notation is as in Algorithm 1 and Proposition 2. We write inputU for u, vec for v, coeffsBound for M and coeffs for c here, because the names u, v, M and c also appear in the main part of the proof of Theorem 1. First we use standard Mathematica high-precision numbers to perform the algorithm (find the coefficients). Then rigorous intervals are used to verify the claim.

7/29/24 15:33:09 In[77]:=

```
theta = Table[(initialImZeroMMA[[n + 1]] - initialImZeroMMA[[1]]) / (2 π), {n, 1, cN}];
e = Table[Table[If[k == n, 1, 0], {k, 1, cN}], {n, 1, cN}];
inputU = Table[e[[n]] - theta * ((e[[n]].theta) / (theta.theta)), {n, 1, cN}];
δ = 1 / 4;
coeffsBound = 10^300;
vec = inputU;
coeffs = Table[Table[If[k == l, 1, 0], {k, 1, cN}], {l, 1, cN}];
mu = Table[Table[0, {k, 1, l - 1}], {l, 1, cN}];
(*This is just to set up the table structure.*)
vStar = vec; (*To set up the table structure.*)
cB = Table[0, {k, 1, cN}];
update[l_, m_] :=
  Module[{i, j},
   For[i = 1, i < l, i++,
    For[j = l, j ≤ m, j++,
     mu[[j]][[i]] = (vec[[j]].vStar[[i]]) / cB[[i]]
    ]
   ];
   For[i = l, i ≤ cN, i++,
    vStar[[i]] = vec[[i]] - Sum[mu[[i]][[j]] * vStar[[j]], {j, 1, i - 1}];
    cB[[i]] = vStar[[i]].vStar[[i]];
    For[j = i + 1, j ≤ cN, j++,
     mu[[j]][[i]] = (vec[[j]].vStar[[i]]) / cB[[i]]
    ]
   ];
  ];
update[1, cN];
k = 2;
Until[k == cN + 1,
```

```
   For[j = k - 1, j ≥ 1, j--,
    q = Round[mu[[k]][[j]]];
    If[Max[Abs[coeffs[[k]] - q coeffs[[j]]]] > coeffsBound,
     Return[]
    ];
    coeffs[[k]] -= q coeffs[[j]];
    vec[[k]] = Sum[coeffs[[k]][[j]] * inputU[[j]], {j, 1, cN}];
    update[k, k]
   ];
   If [cB[[k]] ≥ (δ - mu[[k]][[k - 1]] ^ 2) cB[[k - 1]],
    k += 1
    ,
    {coeffs[[k - 1]], coeffs[[k]]} = {coeffs[[k]], coeffs[[k - 1]]};
    {vec[[k - 1]], vec[[k]]} = {vec[[k]], vec[[k - 1]]};
    update[k - 1, k];
    k = Max[2, k - 1]
   ]
  ];
d = Sum[Abs[vec[[k]]], {k, 1, cN}];
```

Print["max $\sum_{k=1}^{N}\left|\sum_{j=1}^{N} c_{k,j}\, u_{j,n}\right|$ estimate: ", N[Max[d], 10]];

```
thetaBalls =
   Table[(initialImZero[[n + 1]] - initialImZero[[1]]) / (2 piPrec), {n, 1, cN}];
inputUBalls = Table[
    e[[n]] - thetaBalls * ((e[[n]].thetaBalls) / (thetaBalls.thetaBalls)), {n, 1, cN}];
vecBalls = Table[Sum[coeffs[[k]][[j]] * inputUBalls[[j]], {j, 1, cN}], {k, 1, cN}];
Print["Claim 4: ", Max[Sum[Abs[vecBalls[[k]]], {k, 1, cN}]] < 166 * 10 ^ (-15)];
```

max $\sum_{k=1}^{N}\left|\sum_{j=1}^{N} c_{k,j}\, u_{j,n}\right|$ estimate: $1.655336986\times 10^{-13}$

Claim 4: True

Claim 5

Claim 5 in the paper comprises the preliminary part "There exist $x_{n,k}$ satisfying $0 \leq x_{n,1} \leq \ldots \leq x_{n,\ell} \leq 1/2 - \epsilon$ and $w_n(\epsilon + x_{n,k}) \leq k/\ell$ for $k = 1, \ldots, \ell$, $n = 1, \ldots, N$", and the final estimate:

$$\frac{2}{\delta}\cdot 2^N \sum_{k_1+\ldots+k_N\leq \ell}\prod_{n=1}^{N}(x_{n,k_n} - x_{n,k_n-1}) \geq \frac{1}{60}$$

The pre-computed values of $x_{n,k}$, of which we claim that they exist, are supplied in external files:

```
x1.txt, ..., x21.txt
```

that should be placed in the same directory as the present notebook (and should be available from the same source as this notebook). If they are unavailable (or in the wrong directory), you should receive an error message. If the file contents is incorrect, the behaviour is hard to predict — Mathematica may run for hours processing the input that it did not understand (keeping it in symbolic form).

The reader need not recreate those files to verify the proof. It is quite sufficient to check that the values supplied in our files satisfy the claims.

7/29/24 15:44:04 In[97]:=

```
l = 16 000;(*The resolution for
 approximating the weights or penalty functions.*)
SetOptions[$FrontEndSession, DynamicEvaluationTimeout → Infinity];
v[ϕ_, ψ_] := Piecewise[{{Cos[ϕ] - Cos[ϕ + ψ], ϕ + ψ ≤ pi}}, Cos[ϕ] + 1];
epsilon = 2 * 166 * 10^(-15);
verifyFormat[x_] :=
  Module[{k},
   If[
    Length[x] == l,
    For[k = 1, k ≤ l, k += 1,
     If[Head[x⟦k⟧] ≠ Rational,
      Return[False]
     ]
    ];
    True,
    False
   ]
  ];
passed = True;
For[n = 1, n ≤ cN, n += 1,
  memu = MemoryInUse[];
  c = Table[
    Abs[rho[0] / rho[n]] / (u⟦j⟧ Exp[omega[n] × y⟦j⟧] ×
       Piecewise[{
         {Cos[phiBis⟦n⟧⟦j⟧ / 2 - phiPrime⟦n⟧⟦j⟧ / 2], phiBis⟦n⟧⟦j⟧ ≤ pi / 2},
         {1, phiPrime⟦n⟧⟦j⟧ ≥ pi / 2}
        }, Cos[pi / 4 - phiPrime⟦n⟧⟦j⟧ / 2]
       ]),
    {j, 1, m}];
  c = Join[{Max[Table[
       (Abs[rho[0] / rho[n]] / (u⟦j⟧ Exp[omega[n] × y⟦j⟧])) ×
        Piecewise[{
          {0, phiBis⟦n⟧⟦j⟧ < pi / 2},
          {0, phiPrime⟦n⟧⟦j⟧ > pi / 2}
         }, 1 / Cos[pi / 4 - phiPrime⟦n⟧⟦j⟧ / 2]
        ],
       {j, 1, m}]]}, c, c];
  phi = Join[{pi / 2}, phiPrime⟦n⟧, phiBis⟦n⟧];
  x = ReadList[StringJoin[NotebookDirectory[], "x", ToString[n], ".txt"]];
  If[verifyFormat[x],
   diff = Table[Piecewise[
      {{x⟦k⟧ - x⟦k - 1⟧, k ≠ 1}},
```

```
        x⟦1⟧
       ], {k, 1, l}]; (*The list of lists of differences x_{n,k}-x_{n,k-1}.*)
     passed = passed && (Min[diff] ≥ 0 && Max[x] ≤ 1 / 2 - epsilon);
     Monitor[
      For[j = 1, j ≤ Length[c], j += 1,
        maxk = Min[l, ceiling[l c⟦j⟧ (Cos[phi⟦j⟧] + 1)]];
        For[k = 1, k ≤ maxk, k += 1,
         passed = passed && (c⟦j⟧ × v[phi⟦j⟧, 2 pi (epsilon + x⟦k⟧)] ≤ k / l);
        ];
       ];
      ,
      Column[{StringJoin[
         "Checking ∀_k c_{n,j} max (v (φ'_{n,j}, 2 π (ϵ + x_{n,k})), v (φ''_{n,j}, 2 π (ϵ + x_{n,k}))) for n = ",
         ToString[n], ", j = ", ToString[j], " of ", ToString[Length[c]], "."],
        StringJoin["Memory in use: ", ToString[N[MemoryInUse[] / 2^30,
           {Infinity, 3}]], " GB by all data in the current session, ",
         ToString[N[(Max[0, MemoryInUse[] - memu]) / 2^30, 3]],
         " GB by the check for the current n."]}]];
     ,
     Print["Problem reading the file ",
       StringJoin[NotebookDirectory[], "x", ToString[n], ".txt"]];
    ];
    Clear[c, phi, x];
   ];
  Print["Claim 5, preliminary checks: ", passed];
  diff = Table[{}, {n, 1, cN}];
  (*The variable to store the differences x_{n,k}-x_{n,k-1}.*)
  For[n = 1, n ≤ cN, n += 1,
    x = ReadList[StringJoin[NotebookDirectory[], "x", ToString[n], ".txt"]];
    If[verifyFormat[x],
     diff⟦n⟧ = Table[Piecewise[
         {{x⟦k⟧ - x⟦k - 1⟧, k ≠ 1}},
         x⟦1⟧
        ], {k, 1, l}];
     ,
     Print["Problem reading the file ",
      StringJoin[NotebookDirectory[], "x", ToString[n], ".txt"]];
     Break[];
    ];
   ];
  If[n > cN, last = diff⟦cN⟧;
    For[n = cN - 1, n ≥ 1, n -= 1,
```

```
    Print["The computation of (Sum_{k_n+…+k_N=S} Prod_{i=n}^{N} (x_{i,k_i} - x_{i,k_i-1}))^l_{S=1} for n = ",
     n, " started with ", N[MemoryInUse[] / 2^30, {Infinity, 3}],
     " GB of memory in use."];
    Parallelize[
     conv = Table[Sum[diff[[n]][[i]] × last[[k - i]], {i, 1, k - 1}], {k, 1, l}]];
    last = conv;
   ];
   final = (2 / delta) 2^cN Sum[last[[k]], {k, 1, l}];
   Print["Claim 5, final estimate: ", N[final, 5]];
   Print["Claim 5, final estimate > 1/60: ", final > 1 / 60];
  ];
StringJoin["Memory in use at the end of the computation: ",
 ToString[N[MemoryInUse[] / 2^30, {Infinity, 3}]], " GB."]
```

Claim 5, preliminary checks: True

The computation of $\left(\sum_{k_n+\ldots+k_N=S} \prod_{i=n}^{N} \left(x_{i,k_i} - x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 20 started with 2.22 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S} \prod_{i=n}^{N} \left(x_{i,k_i} - x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 19 started with 2.22 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S} \prod_{i=n}^{N} \left(x_{i,k_i} - x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 18 started with 2.22 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S} \prod_{i=n}^{N} \left(x_{i,k_i} - x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 17 started with 2.22 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S} \prod_{i=n}^{N} \left(x_{i,k_i} - x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 16 started with 2.22 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S} \prod_{i=n}^{N} \left(x_{i,k_i} - x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 15 started with 2.22 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S} \prod_{i=n}^{N} \left(x_{i,k_i} - x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 14 started with 2.22 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 13 started with 2.22 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 12 started with 2.22 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 11 started with 2.23 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 10 started with 2.23 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 9 started with 2.23 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 8 started with 2.23 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 7 started with 2.23 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 6 started with 2.24 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 5 started with 2.24 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 4 started with 2.24 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 3 started with 2.24 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n = 2 started with 2.24 GB of memory in use.

The computation of $\left(\sum_{k_n+\ldots+k_N=S}\prod_{i=n}^{N}\left(x_{i,k_i}-x_{i,k_i-1}\right)\right)_{S=1}^{l}$ for n =

1 started with 2.25 GB of memory in use.

Claim 5, final estimate: 0.016676

Claim 5, final estimate > 1/60: True

7/29/24 21:18:18 Out[108]=

Memory in use at the end of the computation: 2.25 GB.

Computation of $x_{n,k}$ (optional, overwrites files)

This subsection is here only for information on how we obtained the precomputed values of $x_{n,k}$. It is not necessary to re-evaluate it even if you wish to confirm the validity of the proof. If you do run it, you will be overwriting files named

```
x1.txt, ..., x21.txt
```

in the directory containing the present notebook. We advise against it, as any malfunction, bug, or mistake, may result in data loss. Accordingly, all the code in this section was commented out. If you do run it, keep an eye on the memory availability (see the introductory part of this notebook), as we did observe excessive memory use when evaluating the xCompute[] function.

```
(*xCompute[n_]:=
  Module[{t=AbsoluteTime[],memu=MemoryInUse[],var,c,phi,x,maxk,j,k},
   Monitor[
    var="c_{n,j}";
    c=Table[
      Abs[rho[0]/rho[n]]/(u⟦j⟧Exp[omega[n]y⟦j⟧]
         Piecewise[{
           {Cos[phiBis⟦n⟧⟦j⟧/2-phiPrime⟦n⟧⟦j⟧/2],phiBis⟦n⟧⟦j⟧≤pi/2},
           {1,phiPrime⟦n⟧⟦j⟧≥pi/2}
          },Cos[pi/4-phiPrime⟦n⟧⟦j⟧/2]
         ]),
      {j,1,m}];
    var="c_{n,0}";
    c=Join[{Max[Table[
          (Abs[rho[0]/rho[n]]/(u⟦j⟧Exp[omega[n]y⟦j⟧]))
           Piecewise[{
             {0,phiBis⟦n⟧⟦j⟧<pi/2},
             {0,phiPrime⟦n⟧⟦j⟧>pi/2}
            },1/Cos[pi/4-phiPrime⟦n⟧⟦j⟧/2]
           ],
          {j,1,m}]]},c,c];
    ,
    StringJoin["Computing ",var,
     " for n = ",ToString[n],", j = ",ToString[j],"."]
```

```
  ];
  phi=Join[{pi/2},phiPrime[[n]],phiBis[[n]]];
  Monitor[
   var="x_{n,k}";
   x=Table[1/2-(3/2)epsilon,{k,1,l}];
   For[j=1,j≤Length[c],j+=1,
    maxk=Min[l,ceiling[l c[[j]](Cos[phi[[j]]]+1)]];
    For[k=1,k≤maxk,k+=1,
     If[c[[j]](1+Cos[phi[[j]]])>k/l,
      x[[k]]=Min[x[[k]],Information[(ArcCos[Cos[phi[[j]]]-k/(l c[[j]])]-phi[[j]])/
             (2 pi),"Bounds"][[1]]-(3/2)epsilon];
     ]
    ];
   ];
   ,
   Column[{StringJoin["Computing ",var," for n = ",ToString[n],
      ", j = ",ToString[j]," of ",ToString[Length[c]],"."],StringJoin[
      "Memory in use: ",ToString[N[MemoryInUse[]/2^30,{Infinity,3}]],
      " GB by all data in the current session, ",
      ToString[N[(MemoryInUse[]-memu)/2^30,{Infinity,3}]],
      " GB by the computation for the current n."]}]];
  Export[StringJoin[NotebookDirectory[],"x",ToString[n],".txt"],x];
  If[x==ReadList[StringJoin[NotebookDirectory[],"x",ToString[n],".txt"]],
   Print["The values of x_{n,k} for n=",n,
     " were saved to the file x",n,".txt in the notebook's directory."];,
   Print["The values of x_{n,k} for n=",n," were saved to the file x",
     n,".txt, but could not be read back correctly."];
  ];
  Clear[c,phi,x];
  Print["This computation (for n=",
   n,") has taken ",Ceiling[(AbsoluteTime[]-t)/60],
   " minute(s) and increased the memory in use by about ",
   N[(MemoryInUse[]-memu)/2^30,3]," GB"];
 ];*)
```

```
(*xCompute[1];*)
```

The values of $x_{n,k}$ for n=1 were saved to the file x1.txt in the notebook's directory.

This computation (for n=1) has taken 64
minute(s) and increased the memory in use by about 2.15 GB

```
(*xCompute[2];*)
```

The values of $x_{n,k}$ for n=2 were saved to the file x2.txt in the notebook's directory.

This computation (for n=2) has taken 62
minute(s) and increased the memory in use by about 1.73 GB

(*xCompute[3];*)

The values of $x_{n,k}$ for n=3 were saved to the file x3.txt in the notebook's directory.

This computation (for n=3) has taken 35
minute(s) and increased the memory in use by about 1.15 GB

(*xCompute[4];*)

The values of $x_{n,k}$ for n=4 were saved to the file x4.txt in the notebook's directory.

This computation (for n=4) has taken 29
minute(s) and increased the memory in use by about 0.970 GB

(*xCompute[5];*)

The values of $x_{n,k}$ for n=5 were saved to the file x5.txt in the notebook's directory.

This computation (for n=5) has taken 21
minute(s) and increased the memory in use by about 0.726 GB

(*xCompute[6];*)

The values of $x_{n,k}$ for n=6 were saved to the file x6.txt in the notebook's directory.

This computation (for n=6) has taken 16
minute(s) and increased the memory in use by about 0.569 GB

(*xCompute[7];*)

The values of $x_{n,k}$ for n=7 were saved to the file x7.txt in the notebook's directory.

This computation (for n=7) has taken 13
minute(s) and increased the memory in use by about 0.460 GB

(*xCompute[8];*)

The values of $x_{n,k}$ for n=8 were saved to the file x8.txt in the notebook's directory.

This computation (for n=8) has taken 9
minute(s) and increased the memory in use by about 0.324 GB

(*xCompute[9];*)

The values of $x_{n,k}$ for n=9 were saved to the file x9.txt in the notebook's directory.

This computation (for n=9) has taken 8
minute(s) and increased the memory in use by about 0.253 GB

(*xCompute[10];*)

The values of $x_{n,k}$ for n=10 were saved to the file x10.txt in the notebook's directory.

This computation (for n=10) has taken 6
minute(s) and increased the memory in use by about 0.193 GB

(*xCompute[11];*)

```
The values of x_{n,k} for n=11 were saved to the file x11.txt in the notebook's directory.

This computation (for n=11) has taken 5
  minute(s) and increased the memory in use by about 0.143 GB
```

(*xCompute[12];*)

```
The values of x_{n,k} for n=12 were saved to the file x12.txt in the notebook's directory.

This computation (for n=12) has taken 4
  minute(s) and increased the memory in use by about 0.111 GB
```

(*xCompute[13];*)

```
The values of x_{n,k} for n=13 were saved to the file x13.txt in the notebook's directory.

This computation (for n=13) has taken 3
  minute(s) and increased the memory in use by about 0.0974 GB
```

(*xCompute[14];*)

```
The values of x_{n,k} for n=14 were saved to the file x14.txt in the notebook's directory.

This computation (for n=14) has taken 2
  minute(s) and increased the memory in use by about 0.0666 GB
```

(*xCompute[15];*)

```
The values of x_{n,k} for n=15 were saved to the file x15.txt in the notebook's directory.

This computation (for n=15) has taken 2
  minute(s) and increased the memory in use by about 0.0558 GB
```

(*xCompute[16];*)

```
The values of x_{n,k} for n=16 were saved to the file x16.txt in the notebook's directory.

This computation (for n=16) has taken 2
  minute(s) and increased the memory in use by about 0.0448 GB
```

(*xCompute[17];*)

```
The values of x_{n,k} for n=17 were saved to the file x17.txt in the notebook's directory.

This computation (for n=17) has taken 2
  minute(s) and increased the memory in use by about 0.0357 GB
```

(*xCompute[18];*)

```
The values of x_{n,k} for n=18 were saved to the file x18.txt in the notebook's directory.

This computation (for n=18) has taken 1
  minute(s) and increased the memory in use by about 0.0259 GB
```

(*xCompute[19];*)

```
The values of x_{n,k} for n=19 were saved to the file x19.txt in the notebook's directory.

This computation (for n=19) has taken 1
  minute(s) and increased the memory in use by about 0.0229 GB
```

(*xCompute[20];*)

The values of $x_{n,k}$ for n=20 were saved to the file x20.txt in the notebook's directory.

This computation (for n=20) has taken 1 minute(s) and increased the memory in use by about 0.0189 GB

```
(*xCompute[21];*)
```

The values of $x_{n,k}$ for n=21 were saved to the file x21.txt in the notebook's directory.

This computation (for n=21) has taken 1 minute(s) and increased the memory in use by about 0.0141 GB