Computation for the lace expansion for lattice trees

Analysis of Section 3.3

Robert Fitzner
University of Technology Eindhoven

Abstract

In this file we performs the numerical part of the analysis of the non-backtracking lace expansion for lattice trees. All references in this version of the notebook will be to the PhD thesis of the author.

We expects as input the dimension d and the constance Γ_1,Γ_2,Γ_3, c_1,..,c_4. After choosing these quantities the used should select the menu item Evaluate-> Evaluate Notebook. In a table at the end of this document the result of the computations are shown. There it can be see whether the bootstrap with the given parameters and therefore the analysis was succesful.

We first compute bounds on the simple random walk two-point function (Section 5.2.1). Then we compute bound on the two-point function and repulsive diagrams (Section 5.1.2). We use these bounds to compute the Diagramatic bounds derived in Section 4.3. and compute the bounds used for the Analysis in Section 3.3.

11.05.2013

Input

The dimension in which we perfrom the computations

```
In[2235]:= d = 29;
```

For the bootstrap we assume that $f_i(z) \leq \Gamma_i$ with Γ_i gives as follows

```
In[2236]:= Gamma1 = 1.0331553;
           Gamma2 = 1.11706;
           Gamma3 = 1.2;
```

For the bootstrap function f_3 we use the following constants

```
In[2239]:= c1 = 0.1573;
           c2 = 0.75046;
           c3 = 0.229304;
           c4 = 9.73932;
```

Value to obtain the result in dimensions 29 and 30. (* deactivated *)

```
In[2243]:= (*Gamma1=1.0331553;
           Gamma2=1.11706;
           Gamma3=1.2;
           c1=0.1573;
           c2=0.75046;
           c3=0.229304;
           c4=9.73932;*)
```

We should als be able to obtain the result in dimension 28,27... by produce more eloborate bound on the coefficient $\Xi^{(0)}$,$\Xi^{(1)}$ and $\Xi^{(0),i}$.

Simple Random Walk integral

We compute the two-point function of the simple random walk,

$$I_{n,m}(x) = \int_{[-\pi,\pi]} e^{\mathrm{i}k\,x} \frac{\hat{D}^m(k)}{\left(1-\hat{D}(k)\right)^n} \frac{d^d k}{(2\pi)^d}$$

see Section 5.2, using that

$$I_{n,m}(x) = I_{n,(m-1)}(x) - I_{(n-1),(m-1)}(x)$$

$$I_{n,0}(x) = \frac{1}{(n-1)!} \int_0^\infty t^{n-1} \prod_{i=1}^d \int_{-\pi}^{\pi} e^{-t/d(1-\cos(k_i))} e^{\mathrm{i}k_i\,x_i} \frac{\mathrm{dk}}{2\pi} = \frac{1}{(n-1)!} \int_0^\infty t^{n-1} \prod_{i=1}^d F(t,\,d,\,|x_i|)$$

where F(t,d,n) is the modified Besselfunction. We implement the Besselfunciton and a function to compute $I_{n,0}(x)$.

```
In[2244]:= F[t_, d_, N_] := e^(-t/d) BesselI[N, t/d];
NInt[n_, d_, T_] :=
  1 / ((n - 1) !) * NIntegrate[t^(n - 1) * (F[t, d, 0])^d, {t, 0, T},
    WorkingPrecision → 40];
```

Then we define the number of n-step SRW loop as given in Section (5.2.6)-(5.2.10)

```
In[2246]:= s2 = N[2 d];
s4 = N[(d * 4! / (2 × 2) + d (d - 1) / 2 * 4!)];
s6 = N[ (d * 6! / (3! 3!) + d * (d - 1) * 6! / (2 × 2) + d (d - 1) (d - 2) / 3! * 6!)];
s8 =
  N[(d * 8! / (4! 4!) + d * (d - 1) * (8! / (3! 3!) + 8! / 2^5) + d (d - 1) (d - 2) / 2 * 8! / (2 × 2) +
     d (d - 1) (d - 2) (d - 3) / 4! * 8!)];
```

Then we compute $I_{n,0}(0)$ for n=1,2,3,4:

```
In[2250]:= I10 = NInt[1, d, ∞];
I20 = NInt[2, d, ∞];
I30 = NInt[3, d, ∞];
I40 = NInt[4, d, ∞];
```

and use $I_{n,m}(0) = I_{n,(m-1)}(0) - I_{(n-1),(m-1)}(0)$ to compute $I_{n,m}(0)$:

In[2254]:=
```
SRWTwoPointFunctionTable =
  {{n m, 0, 1, 2, 3, 4}, {0, 1, I10, I20, I30, I40}, {1, 0, 0, 0, 0, 0},
   {2, s2/(2 d)^2, 0, 0, 0, 0}, {3, 0, 0, 0, 0, 0}, {4, s4/(2 d)^4, 0, 0, 0, 0},
   {5, 0, 0, 0, 0, 0}, {6, s6/(2 d)^6, 0, 0, 0, 0}, {7, 0, 0, 0, 0, 0},
   {8, s8/(2 d)^8, 0, 0, 0, 0}, {9, 0, 0, 0, 0, 0}, {10, -1, 0, 0, 0, 0}};
For[i = 3, i < 13, i++,
 For[j = 3, j < 7, j++,
  SRWTwoPointFunctionTable[[i, j]] =
    SRWTwoPointFunctionTable[[i - 1, j]] - SRWTwoPointFunctionTable[[i - 1, j - 1]];]
]
Clear[i, j]

I11 = SRWTwoPointFunctionTable[[3, 3]];
I12 = SRWTwoPointFunctionTable[[4, 3]];
I14 = SRWTwoPointFunctionTable[[6, 3]];
I16 = SRWTwoPointFunctionTable[[8, 3]];
I16 = SRWTwoPointFunctionTable[[8, 3]];
I18 = SRWTwoPointFunctionTable[[10, 3]];
I110 = SRWTwoPointFunctionTable[[12, 3]];
I21 = SRWTwoPointFunctionTable[[3, 4]];
I22 = SRWTwoPointFunctionTable[[4, 4]];
I24 = SRWTwoPointFunctionTable[[6, 4]];
I26 = SRWTwoPointFunctionTable[[8, 4]];
I28 = SRWTwoPointFunctionTable[[10, 4]];
I210 = SRWTwoPointFunctionTable[[12, 4]];
I31 = SRWTwoPointFunctionTable[[3, 5]];
I32 = SRWTwoPointFunctionTable[[4, 5]];
I33 = SRWTwoPointFunctionTable[[5, 5]];
I34 = SRWTwoPointFunctionTable[[6, 5]];
I36 = SRWTwoPointFunctionTable[[8, 5]];
I38 = SRWTwoPointFunctionTable[[10, 5]];
I310 = SRWTwoPointFunctionTable[[12, 5]];
I42 = SRWTwoPointFunctionTable[[4, 6]];
I44 = SRWTwoPointFunctionTable[[6, 6]];
I46 = SRWTwoPointFunctionTable[[8, 6]];
I48 = SRWTwoPointFunctionTable[[8, 6]];
I410 = SRWTwoPointFunctionTable[[12, 6]];

NForm[a_] := NumberForm[N[a] , 5];
Labeled[Grid[Map[NForm, SRWTwoPointFunctionTable, {2}],
   Alignment -> {{Left, Center}, Baseline, {{2, 12}, {2, 6}} -> {"."}},
   Frame -> True, Dividers -> {{2 -> True, -1 -> True}, {2 -> True}},
   Spacings -> {1.5, {1.5, 1, {0.5}}}, ItemStyle -> {1 -> Bold, 1 -> Bold},
   Background -> {Automatic, Automatic, {{2, 12}, {2, 6}} -> GrayLevel[0.9]}],
  Style["Value of the SRW two-point function.", Bold], Top] // Text
```

Value of the SRW two−point function.

Out[2283]=

m n	0.	1.	2.	3.	4.
0.	1.	1.0182	1.0567	1.1191	1.2111
1.	0.	0.018201	0.038508	0.062411	0.091961
2.	0.017241	0.018201	0.020308	0.023902	0.02955
3.	0.	0.00095921	0.0021073	0.0035943	0.0056483
4.	0.00087642	0.00095921	0.0011481	0.001487	0.002054
5.	0.	0.000082793	0.00018887	0.00033889	0.00056707
6.	0.000072963	0.000082793	0.00010608	0.00015001	0.00022818
7.	0.	9.83×10^{-6}	0.000023286	0.000043934	0.000078168
8.	8.3558×10^{-6}	9.83×10^{-6}	0.000013456	0.000020648	0.000034234
9.	0.	1.4742×10^{-6}	3.6264×10^{-6}	7.1912×10^{-6}	0.000013586
10.	−1.	1.4742×10^{-6}	2.1522×10^{-6}	3.5647×10^{-6}	6.3948×10^{-6}

Bound on the two-point function and on repulsive diagrams

Definition of Constants

We define the constants for two setting s: we use s=i for bound on $z = z_I$ and s=o for bound on $z \epsilon (z_I, z_c)$: For $z = z_I$, we use the following relations, that are proven in Section 3.6.2.

$$\begin{gathered} z_I = \frac{1}{(2d-1)e} \\ g_{z_I} \leq e + \frac{e-1}{2d-1} \\ g^i_{z_I} \leq 1 + (g_{z_I} - 1)\frac{2d-1}{2d} \leq e \\ G_z(x) \leq g^i_{z_I} B_{z_I g^i_{z_I}}(x) \leq g^i_{z_I} \frac{2d-2}{2d-1} C(x) \\ \tilde{G}_z(x) \leq B_{z_I g^i_{z_I}}(x) \leq \frac{2d-2}{2d-1} C(x). \end{gathered} \tag{1}$$

For z ϵ (z_I, z_c) we know that

$$\begin{gathered} 2d\, z\, g^i_z < 2d\, g_z\, z < \Gamma_1 \\ g_z < e\,\Gamma_1 \text{ for all } z \geq \frac{1}{(2d-1)e} \\ g^i_z < 1 + (g_z - 1)\frac{2d-1}{2d} \end{gathered} \tag{2}$$

In the following we implement the basic quantities for the status i at z_I and the status o for z in (z_I, z_c), which will allows us to implement the bound for both (s=i,o) with the same code

In[2284]:=
```
g[i] = Exp[1] + (Exp[1] - 1)/(2 d - 1);
g[o] = Exp[1] * Gamma1;
gj[i] = Exp[1];
gj[o] = 1 + (g[o] - 1) * (2 d - 1)/(2 d);
rho[i] = gj[i]/g[i];
rho[o] = gj[o]/g[o];
twodgz[i] = 2 d 1/((2 d - 1) Exp[1]) g[i];
twodgz[o] = (2 d)/(2 d - 1) Gamma1;
twodgjz[i] = (2 d)/(2 d - 1);
twodgjz[o] = (2 d)/(2 d - 1) Gamma1;
gjz[i] = 1/(2 d - 1);
gjz[o] = 1/(2 d - 1) Gamma1;

VarGamma1[i] = (1 + (1 - Exp[-1])/(2 d - 1));
VarGamma1[o] = Gamma1;
VarGamma2[i] = rho[i] * (2 d - 2)/(2 d - 1); (*G_z(x) ≤g_z B_{zg_z}(x) ≤ g_z (2d-2)/(2d-1) C(x)*)
VarGamma2[o] = Gamma2 * (2 d - 2)/(2 d - 1); (*Ĝ_z(k) ≤ ConstantGvsC Ĉ(k), follows from f2*)
VarGamma3[i] = 1; (*We bound a weighted line by replacing the tree two-
 point function with a normal on*)
VarGamma3[o] = Gamma3; (*Ĝ_z(k) ≤ Ĉ(k)*)
Varc1[o] = c1; Varc2[o] = c2; Varc3[o] = c3; Varc4[o] = c4;
Varc1[i] = 0; Varc2[i] = 0.5; Varc3[i] = 0; Varc4[i] = 4;
```

Further, we define variables to save the number of short SAWs, as given in Section 5.1.3

In[2304]:=
```
c2ik = 2; (*c_2(e_1+e_2)*)
c4ik = 4 (2 d - 3) + 2 (2 d - 4); (*c_4(e_1+e_2)*)
c6ik = 16 + 84 (2 d - 4) + 36 (2 d - 4) (2 d - 6); (*c_4(e_1+e_2)*)

c3i = (2 d - 2); (*c_3(e_1)*)
c5i = (3 (2 d - 2 ) + 4 (2 d - 2) (2 d - 4) ); (*c_5(e_1)*)
c7i = (14 (2 d - 2 ) + 62 (2 d - 2) (2 d - 4) + 27 (2 d - 2) (2 d - 4) (2 d - 6) );
(*c_7(e_1)*)
```

Bounds on two-point function

We compute bounds as explained in Section 5.1.2. We begin by computing $G_{n,z}(e_1)$ as in (5.1.22)-(5.1.24)

In[2310]:=
```
Do[
  Bound[tG7i, s] = c7i (gjz[s])^7 + (twodgjz[s])^9 * VarGamma2[s] * I110; (* G_{7,z}(e_1) *)
  Bound[tG5i, s] = c5i (gjz[s])^5 + Bound[tG7i, s]; (* G_{5,z}(e_1) *)
  Bound[tG3i, s] = c3i (gjz[s])^3 + Bound[tG5i, s]; (* G_{3,z}(e_1) *)
  Bound[tG1i, s] = gjz[s] + Bound[tG3i, s];(* G_{1,z}(e_1) *)
  , {s, {i, o}}]
```

Then we compute $G_{n,z}(e_1 + e_2)$ and $G_{4,z}(2\,e_2)$, see (5.1.25)-(5.1.26) :

In[2311]:=
```
Do[
  Bound[tG8ik, s] = d/(d - 1) (twodgjz[o])^8 VarGamma2[s] I110; (*G_8(e_1+e_2)*)
  Bound[tG6ik, s] = c6ik gjz[s]^6 + VarGamma2[s] I18; (*G_6(e_1+e_2)*)
  Bound[tG4ik, s] = Bound[tG6ik, s] + (c4ik - 2 (2 d - 3) ) gjz[s]^4; (*G_4^1(e_1+e_2)*)
  Bound[tG2ik, s] = Bound[tG4ik, s] + (c2ik - 1) gjz[s]^2;(*G_2^1(e_1+e_2)*)
  (*Bound[tG4ii,s]=(2d+2)gjz[s]^4+Bound[tG6,s];(* Bound for sup_x G_4^1(2 e_1)*)*)
  , {s, {i, o}}]
```

We compute the supreme of the two-point function as given in (5.1.27)-(5.1.31):

In[2312]:=
```
Do[
  Bound[tG6, s] = Max[c6ik gjz[s]^6, c7i gjz[s]^7] + (twodgjz[s])^8 VarGamma2[s] I18;
  (* Bound for sup_x G_6(x)*)
  Bound[tG4, s] = Max[c4ik gjz[s]^4, c5i gjz[s]^5] + Bound[tG6, s];
  (* Bound for sup_x G_4(x)*)
  Bound[tG2, s] = Max[c2ik gjz[s]^2, c3i gjz[s]^3] + Bound[tG4, s];
  (* Bound for sup_x G_2(x)*)
  Bound[tG1, s] = Max[Bound[tG1i, s], Bound[tG2, s]];(* Bound for sup_x G_1(x)*)
  , {s, {i, o}}]
```

Closed repulsive diagrams

We define the bounds on the closed repulsive diagrams as desribed in Section 5.1.2 in (5.1.34)-(5.1.36). The bound does only depent on the total number of steps and the number of tw-point functions involved. It does not depent on the indivitual length of the pieces m_1, m_2, ... and of the orientation of the arrows.

```
In[2313]:= Do[
  Bound[ClosedRepLoop, 4, s] = twodgjz[s] Bound[tG3i, s];
  Bound[ClosedRepBubble, 4, s] =
   gjz[s]^4 (2 d c3i) + 3 gjz[s]^6 (2 d c5i) + 5 gjz[s]^8 (2 d c7i) +
    6 twodgjz[s]^10 VarGamma2[s] I110 + twodgjz[s]^10 VarGamma2[s]^2 I210;
  Bound[ClosedRepBubble, 3, s] =
   Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s];
  Bound[ClosedRepTriangle, 4, s] =
   ((4 + 1 - 4) (4 + 2 - 4))/2 gjz[s]^4 (2 d c3i) + ((6 + 1 - 4) (6 + 2 - 4))/2 gjz[s]^6 (2 d c5i) +
    ((8 + 1 - 4) (8 + 2 - 4))/2 gjz[s]^8 (2 d c7i) +
    ((10 - 4) (9 - 4))/2 twodgjz[s]^10 VarGamma2[s] I110 + 6 twodgjz[s]^10 VarGamma2[s]^2 I210 +
    twodgjz[s]^10 VarGamma2[s]^3 I310;
  Bound[ClosedRepSquare, 4, s] =
   gjz[s]^4 (2 d c3i) + 10 gjz[s]^6 (2 d c5i) + 35 gjz[s]^8 (2 d c7i) +
    84 twodgjz[s]^10 VarGamma2[s] I110 + ((10 - 4) (9 - 4))/2 twodgjz[s]^10 VarGamma2[s]^2 I210 +
    6 twodgjz[s]^10 VarGamma2[s]^3 I310 + twodgjz[s]^10 VarGamma2[s]^4 I410;
  , {s, {i, o}}]
```

Open repulsive diagrams

Then we define the bound on the open repulsive diagrams as in (5.1.38):

In[2314]:=
```
Do[
  Bound[OpenRepBubble, 2, s] =
   Max[c2ik gjz[s]^2 + 3 c4ik gjz[s]^4 + 5 c6ik gjz[s]^6,
     2 c3i gjz[s]^3 + 4 c5i gjz[s]^5 + 6 c7i gjz[s]^7] + 6 twodgjz[s]^8 VarGamma2[s] I18 +
    twodgjz[s]^8 VarGamma2[s]^2 I28;
  Bound[OpenRepBubble, 3, s] =
   Max[2 c4ik gjz[s]^4 + 4 c6ik gjz[s]^6, c3i gjz[s]^3 + 3 c5i gjz[s]^5 + 5 c7i gjz[s]^7] +
    5 twodgjz[s]^8 VarGamma2[s] I18 + twodgjz[s]^8 VarGamma2[s]^2 I28;
  Bound[OpenRepTriangle, 1, s] =
   Max[3 c2ik gjz[s]^2 + 10 c4ik gjz[s]^4 + 21 c6ik gjz[s]^6,
     gjz[s] + 6 c3i gjz[s]^3 + 15 c5i gjz[s]^5 + 28 c7i gjz[s]^7] +
    (8 - 1) (7 - 1)/2 twodgjz[s]^8 VarGamma2[s] I18 + 7 twodgjz[s]^8 VarGamma2[s]^2 I28 +
    twodgjz[s]^8 VarGamma2[s]^3 I38;
  Bound[OpenRepTriangle, 2, s] =
   Max[c2ik gjz[s]^2 + 6 c4ik gjz[s]^4 + 15 c6ik gjz[s]^6,
     3 c3i gjz[s]^3 + 10 c5i gjz[s]^5 + 21 c7i gjz[s]^7] +
    (8 - 2) (7 - 2)/2 twodgjz[s]^8 VarGamma2[s] I18 + 6 twodgjz[s]^8 VarGamma2[s]^2 I28 +
    twodgjz[s]^8 VarGamma2[s]^3 I38;
  Bound[OpenRepTriangle, 3, s] =
   Max[3 c4ik gjz[s]^4 + 10 c6ik gjz[s]^6, c3i gjz[s]^3 + 6 c5i gjz[s]^5 + 15 c7i gjz[s]^7] +
    (8 - 3) (7 - 3)/2 twodgjz[s]^8 VarGamma2[s] I18 + 5 twodgjz[s]^8 VarGamma2[s]^2 I28 +
    twodgjz[s]^8 VarGamma2[s]^3 I38;
  Bound[OpenRepSquare, 3, s] =
   Max[4 c4ik gjz[s]^4 + 20 c6ik gjz[s]^6, c3i gjz[s]^3 + 10 c5i gjz[s]^5 + 35 c7i gjz[s]^7] +
    56 twodgjz[s]^8 VarGamma2[s] I18 + (8 - 3) (7 - 3)/2 twodgjz[s]^8 VarGamma2[s]^2 I28 +
    5 twodgjz[s]^8 VarGamma2[s]^3 I38 + twodgjz[s]^8 VarGamma2[s]^4 I48;
  , {s, {i, o}}]
```

Weighted Diagrams

First we define weighted diagams as explained in Section 5.1., e.g. (5.1.19) and (5.1.42)-(5.1.49) we derive weighted closed diagrams

In[2315]:=
```
Do[
  Bound[WeightedClosedLine, 2, s] =
   8 d gjz[s]^2 (gjz[s]^4 (2 d - 2) * 2 + Bound[tG6, s]) +
    8 d gjz[s]^2 (2 d - 2) (gjz[s]^2 + Bound[tG4ik, s]);
  Bound[WeightedClosedBubble, 4, s] =
   twodgjz[s]^4 VarGamma2[s] VarGamma3[s] (Varc1[s] I24 + (2 Varc2[s] + Varc3[s]) I34);
  Bound[WeightedClosedBubble, 2, s] =
   Bound[WeightedClosedBubble, 4, s] + 8 d gjz[s]^2 (gjz[s]^4 (2 d - 2) * 2 + Bound[tG6, s]) +
    8 d gjz[s]^2 (2 d - 2) (gjz[s]^2 + Bound[tG4ik, s]) +
    2 d gjz[s]^3 (Bound[tG3i, s] + 3 (2 d - 2) d/(d - 1) d/(d - 2) twodgjz[s]^3 VarGamma2[s] I16 +
       5 (2 d - 2) d/(d - 1) Bound[tG6, s] +
       9 (gjz[s]^2 + 3 (2 d - 2) gjz[s]^5 + Bound[tG6, s]));

  Bound[WeightedClosedTriangle, 4, s] =
   twodgjz[s]^2 VarGamma2[s]^2 VarGamma3[s] (Varc1[s] I34 + (2 Varc2[s] + Varc3[s]) I44);
  Bound[WeightedClosedTriangle, 2, s] =
   2 Bound[WeightedClosedBubble, 4, s] + Bound[WeightedClosedTriangle, 4, s] +
    8 d gjz[s]^2 (gjz[s]^4 (2 d - 2) * 2 + Bound[tG6, s]) +
    8 d gjz[s]^2 (2 d - 2) (gjz[s]^2 + Bound[tG4ik, s]) +
    4 d gjz[s]^3 (Bound[tG3i, s] + 3 (2 d - 2) d/(d - 1) d/(d - 2) twodgjz[s]^3 VarGamma2[s] I16 +
       5 (2 d - 2) d/((d - 1)) Bound[tG6, s] + 9 (gjz[s]^2 + 3 (2 d - 2) gjz[s]^5 + Bound[tG6, s]));
  , {s, {i, o}}]
```

Then we bound the open diagram using (5.1.19), for a odd number we used Chauchy-Schwarz to obtain an improved bound

In[2316]:=
```
Do[
  Bound[WeightedOpenBubble, 0, s] =
   VarGamma2[s] VarGamma3[s] (Varc1[s] I20 + (2 Varc2[s] + Varc4[s]) I30);
  Bound[WeightedOpenBubble, 2, s] =
   twodgjz[s]^2 VarGamma2[s] VarGamma3[s] (Varc1[s] I22 + (2 Varc2[s] + Varc4[s]) I32);
  Bound[WeightedOpenBubble, 1, s] =
   twodgjz[s] VarGamma2[s] VarGamma3[s]
    √((Varc1[s] I20 + (2 Varc2[s] + Varc4[s]) I30)
       (Varc1[s] I22 + (2 Varc2[s] + Varc4[s]) I32));
  Bound[WeightedOpenBubble, 3, s] =
   twodgjz[s]^3 VarGamma2[s] VarGamma3[s]
    √((Varc1[s] I22 + (2 Varc2[s] + Varc4[s]) I32)
       (Varc1[s] I24 + (2 Varc2[s] + Varc4[s]) I34));



  Bound[WeightedOpenTriangle, 0, s] =
   VarGamma2[s]^2 VarGamma3[s] (Varc1[s] I30 + (2 Varc2[s] + Varc4[s]) I40);
  Bound[WeightedOpenTriangle, 2, s] =
   twodgjz[s]^2 VarGamma2[s]^2 VarGamma3[s] (Varc1[s] I32 + (2 Varc2[s] + Varc4[s]) I42);
  Bound[WeightedOpenTriangle, 1, s] =
   twodgjz[s] VarGamma2[s]^2 VarGamma3[s]
    √((Varc1[s] I30 + (2 Varc2[s] + Varc4[s]) I40)
       (Varc1[s] I32 + (2 Varc2[s] + Varc4[s]) I42));
  Bound[WeightedOpenTriangle, 3, s] =
   twodgjz[s]^3 VarGamma2[s]^2 VarGamma3[s]
    √((Varc1[s] I32 + (2 Varc2[s] + Varc4[s]) I42)
       (Varc1[s] I34 + (2 Varc2[s] + Varc4[s]) I44));
  , {s, {i, o}}]
```

We define the elements as given in (4.3.31)-(4.3.37) and (4.3.49)-(4.3.51)

In[2317]:=
```
Do[
  Bound[Delta, I, 0, s] = 2 twodgjz[s] Bound[tG3i, s] +
    2 Bound[WeightedClosedBubble, 2, s] + Bound[WeightedClosedTriangle, 2, s];
  Bound[Delta, I, 1, s] = Bound[tG3i, s] + Bound[WeightedClosedBubble, 2, s] / twodgjz[s] +
    Bound[WeightedClosedTriangle, 2, s] / twodgjz[s];
  Bound[Delta, I, 2, s] = Bound[WeightedOpenTriangle, 0, s];
  Bound[Delta, I, 3, s] = 2 Bound[WeightedOpenBubble, 2, s] / twodgjz[s] +
    2 Bound[WeightedOpenTriangle, 3, s] / twodgjz[s];
  Bound[Delta, I, 4, s] = Bound[WeightedOpenBubble, 1, s] +
    Bound[WeightedOpenBubble, 2, s] / twodgjz[s] + 2 Bound[WeightedOpenTriangle, 2, s] / twodgjz[s];
  Bound[Delta, I, 5, s] = Bound[WeightedClosedTriangle, 2, s] / twodgjz[s]^2;
  Bound[Delta, I, 6, s] = 2 Bound[WeightedClosedBubble, 2, s] +
    Bound[WeightedClosedTriangle, 4, s];

  Bound[Delta, II, 0, s] =
   twodgjz[s] Bound[tG3i, s] + Bound[WeightedClosedBubble, 2, s] +
    Bound[WeightedClosedTriangle, 2, s];
  Bound[Delta, II, 1, s] = Bound[WeightedClosedTriangle, 2, s] / twodgjz[s];
  Bound[Delta, II, 2, s] = Bound[WeightedOpenTriangle, 1, s];
  , {s, {i, o}}]
```

Bound on the coefficients

Bound for N=0

The bounds stated in Lemma 4.3.6:

In[2318]:=
```
Do[
  Bound[Xi, normal, 0, s] = 1 + Bound[ClosedRepBubble, 4, s]; (*Bound for Ξ*)
  Bound[Xi, iota, 0, s] = rho[s] Bound[tG1, s]; (*Bound for ∑_i Ξ^i*)
  Bound[Psii, 0, s] = rho[s]; (*Bound for Ψ^k*)
  Bound[Xi, normal, 0, Delta, 0, s] = 0;

  Bound[Xi, iota, 0, Delta, 0, s] = 0;
  Bound[Xi, iota, 0, Delta, ei, s] = 2 d rho[s] Bound[tG1, s];
  , {s, {i, o}}]
```

Bound for N ≥1

Definition of Initial Pieces (P , *P'*)

To implement the bound of N≥1 we define the boulding matricies given in (4.3.27)-(4.3.53):

In[2319]:=
```
Do[
  (* definition of first peices, inpendent of fiota*)
  Bound[P1, 0, 0, s] =
   (3 Bound[ClosedRepLoop, 4, s] + 3 Bound[ClosedRepBubble, 4, s] +
```

```
   Bound[ClosedRepTriangle, 4, s]);
Bound[P1, 0, 1, s] =
 (2 Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s] +
   Bound[ClosedRepTriangle, 4, s]);
Bound[P1, 0, 2, s] =
 (Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s] +
   Bound[ClosedRepTriangle, 4, s] + Bound[ClosedRepSquare, 4, s]);
Bound[P1, 0, -1, s] =
 (Bound[ClosedRepLoop, 4, s] + 2 Bound[ClosedRepBubble, 4, s] +
   Bound[ClosedRepTriangle, 4, s]);
Bound[P1, 0, -2, s] =
 (Bound[ClosedRepTriangle, 4, s] + Bound[ClosedRepSquare, 4, s]);;
(* definition of first peices, first step of the backbone goes to e_i*)
Do[Bound[P1, IotaStep, t, s] = Bound[P1, 0, t, s], {t, {-2, -1, 0, 1, 2}}];
Bound[P1, IotaRib, 0, s] =
 2 d (Bound[tG1i, s] Bound[P1, 0, 0, s] + Bound[ClosedRepBubble, 4, s] +
    2 Bound[tG2, s] Bound[ClosedRepLoop, 4, s] +
    Bound[tG1, s] Bound[ClosedRepBubble, 4, s] +
    Bound[tG3i, s] 1/twodgjz[s] Bound[ClosedRepBubble, 3, s] +
    (Bound[ClosedRepBubble, 4, s])/twodgjz[s] Bound[OpenRepBubble, 2, s] (*c≠x,
    d_omega=1*) + 2 Bound[OpenRepBubble, 3, s]
     (Bound[tG1i, s] + 1/twodgjz[s] Bound[ClosedRepBubble, 3, s]) (*c≠x,
    d_omega≥ 2;u≠v=x*) + Bound[OpenRepTriangle, 3, s]
     (Bound[tG1i, s] + 1/twodgjz[s] Bound[ClosedRepBubble, 3, s]) (*u≠v≠x*));
(*definition of the first piece if e_i is somewhere on the first rib*)
Bound[P1, IotaRib, 1, s] =
 2 d (Bound[tG1i, s] Bound[P1, 0, 1, s] + Bound[ClosedRepBubble, 4, s] +
    Bound[ClosedRepBubble, 4, s] + Bound[ClosedRepTriangle, 4, s] +
    1/twodgjz[s] Bound[ClosedRepBubble, 3, s] Bound[OpenRepBubble, 2, s] +
    (Bound[tG1i, s] + 1/twodgjz[s] Bound[ClosedRepBubble, 3, s])
     Bound[OpenRepTriangle, 3, s]);
Bound[P1, IotaRib, 2, s] =
 2 d (Bound[tG1i, s] Bound[P1, 0, 2, s] +
    (Bound[tG1i, s] + 1/twodgjz[s] Bound[ClosedRepBubble, 3, s])
     Bound[OpenRepSquare, 3, s]);
Bound[P1, IotaRib, -2, s] =
 2 d (Bound[tG1i, s] Bound[P1, 0, -2, s] +
```

```
    (Bound[tG1i, s] + 1/twodgjz[s] Bound[ClosedRepBubble, 3, s])
     Bound[OpenRepSquare, 3, s]);
Bound[P1, IotaRib, -1, s] =
 2 d (Bound[tG1i, s] Bound[P1, 0, -1, s] +
    Bound[tG1, s] (Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s] +
       Bound[ClosedRepTriangle, 4, s]) +
    (Bound[tG1i, s] + 1/twodgjz[s] Bound[ClosedRepBubble, 3, s])
     Bound[OpenRepTriangle, 3, s]);
, {s, {i, o}}]
```

Definition of the intermediate pieces (A $\overline{A}$)

In[2320]:=
```
Do[
 (*definition of itermediate peices, where one shared edges is counted*)
 Do[Bound[A, 0, a, s] = Bound[P1, 0, a, s], {a, {-2, -1, 0, 1, 2}}];
 Bound[A, 1, 0, s] =
  Bound[tG3i, s] +
   1/twodgjz[s] (Bound[ClosedRepLoop, 4, s] + 2 Bound[ClosedRepBubble, 4, s] +
      Bound[ClosedRepTriangle, 4, s]);
 Bound[A, 2, 0, s] = Bound[OpenRepBubble, 2, s] + Bound[OpenRepTriangle, 2, s];

 Bound[A, 1, 1, s] =
  1/twodgjz[s] (Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s] +
     Bound[ClosedRepTriangle, 4, s]);
 Bound[A, 1, 2, s] = Bound[ClosedRepTriangle, 4, s]/twodgjz[s];
 Bound[A, 2, 1, s] = Bound[OpenRepTriangle, 2, s];
 Bound[A, 2, 2, s] = Bound[OpenRepSquare, 3, s];

 Do[
  Bound[A, -t1, 0, s] = Bound[A, t1, 0, s];
  , {t1, 1, 2}];
 Clear[t1];

 (*definition of itermediate peices, where both shared edges are not counted*)
 Do[Bound[Abar, a, 0, s] = gj[s] Bound[A, a, 0, s], {a, {-2, -1, 0, 1, 2}}];
 (*(4.3.25)*)
 Do[Bound[Abar, 0, a, s] = gj[s] Bound[A, a, 0, s], {a, {-2, -1, 0, 1, 2}}];
 (*(4.3.26)*)
 Bound[Abar, 1, 1, s] =
  gj[s]
   1/(twodgjz[s])^2 (Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s] +
      Bound[ClosedRepTriangle, 4, s]);
 Bound[Abar, 1, 2, s] = gj[s] Bound[OpenRepTriangle, 2, s]/twodgjz[s];
 Bound[Abar, 2, 1, s] = Bound[Abar, 1, 2, s];
 Bound[Abar, 2, 2, s] = gj[s] Bound[OpenRepTriangle, 1, s];


 (*Using Symmetrie we define the other ones*)
 Do[Do[
   Do[
    Bound[t, a, -b, s] = Bound[t, a, b, s];
    Bound[t, -a, b, s] = Bound[t, a, b, s];
    Bound[t, -a, -b, s] = Bound[t, a, b, s];
    , {a, {1, 2}}], {b, {1, 2}}], {t, {A, Abar}}];
 (*the more complex pieces*)
 , {s, {i, o}}];
```

Definition of the Delta entries

In[2321]:=
```
Do[
 Bound[Delta, start, 2, s] = Bound[Delta, I, 2, s];
 Bound[Delta, start, 1, s] = Bound[Delta, I, 1, s];
 Bound[Delta, start, 0, s] = Bound[Delta, I, 0, s];
 Bound[Delta, start, -1, s] = Bound[Delta, I, 1, s];
 Bound[Delta, start, -2, s] = Bound[Delta, I, 2, s];

 Bound[Delta, end, 2, s] = Bound[Delta, I, 4, s];
 Bound[Delta, end, 1, s] = Bound[Delta, I, 3, s];
 Bound[Delta, end, 0, s] = Bound[Delta, I, 0, s];
 Bound[Delta, end, -1, s] = Bound[Delta, I, 1, s];
 Bound[Delta, end, -2, s] = Bound[Delta, I, 2, s];

 Bound[Delta, 0, 0, s] = Bound[Delta, I, 0, s];
 Bound[Delta, 1, 0, s] = Bound[Delta, I, 3, s];
 Bound[Delta, 0, -1, s] = Bound[Delta, I, 1, s];
 Bound[Delta, -1, 0, s] = Bound[Delta, I, 1, s];
 Bound[Delta, 0, 1, s] = Bound[Delta, I, 1, s];
 Bound[Delta, -1, 1, s] = Bound[Delta, I, 5, s];
 Bound[Delta, -1, -1, s] = Bound[Delta, I, 5, s];
 Bound[Delta, 1, 1, s] = Bound[Delta, I, 6, s];
 Bound[Delta, 1, -1, s] = Bound[Delta, I, 6, s];

 Do[
  Bound[Delta, -2, t, s] = Bound[Delta, I, 2, s];
  Bound[Delta, 2, t, s] = 2 Bound[Delta, I, 2, s];
  , {t, -2, 2}];
 Bound[Delta, -1, 2, s] = Bound[Delta, I, 2, s];
 Bound[Delta, -1, -2, s] = Bound[Delta, I, 2, s];
 Bound[Delta, 1, 2, s] = 2 Bound[Delta, I, 2, s];
 Bound[Delta, 1, -2, s] = 2 Bound[Delta, I, 2, s];
 Bound[Delta, 0, 2, s] = Bound[Delta, I, 2, s];
 Bound[Delta, 0, -2, s] = Bound[Delta, I, 2, s];
 Do[
  Bound[Delta, iotaI, t, s] = Bound[Delta, II, Abs[t], s] + Bound[Delta, I, Abs[t], s] +
    Bound[tG1i, s] Bound[Delta, 0, t, s] + Bound[tG3i, s] Bound[Delta, -1, t, s] +
    Bound[ClosedRepBubble, 4, s] / twodgjz[s] Bound[Delta, -2, t, s];
  Bound[Delta, iotaII, t, s] =
   Bound[Delta, II, Abs[t], s] + Bound[Delta, I, Abs[t], s] +
    Bound[tG1i, s] Bound[Delta, 0, t, s] + 2 Bound[tG3i, s] Bound[Delta, -1, t, s] +
    2 Bound[ClosedRepBubble, 4, s] / twodgjz[s] Bound[Delta, -2, t, s] +
    2 twodgjz[s] Bound[P1, IotaRib, t, s];
  , {t, -2, 2}];
 , {s, {i, o}}]
```

Definition of the vectors and matices

We condition on the length of the backone and indentify whether the backbone is on the top or bottom of the diagram.

– the backbone in on the bottom, d (u, v) ≥ 2.
– the backbone in on the bottom, d (u, v) = 1.
u = v
– the backbone in on the top, d (u, v) = 1.
– the backbone in on the top, d (u, v) ≥ 2.

In[2322]:=
```
Do[
  VectorP1[normal, s] = Table[Bound[P1, 0, r - 3, s], {r, 1, 5}];
  VectorP1[iota, s] = Table[Bound[P1, IotaStep, r - 3, s] + Bound[P1, IotaRib, r - 3, s],
    {r, 1, 5}];

  MatrixA[s] = Table[Bound[A, r - 3, t - 3, s], {t, 1, 5}, {r, 1, 5}];
  MatrixAbar[s] = Table[Bound[Abar, r - 3, t - 3, s], {t, 1, 5}, {r, 1, 5}];
  VectorAbar[s] = Table[Bound[Abar, r - 3, 0, s], {r, 1, 5}];

  VectorDelta[start, s] = Table[Bound[Delta, start, r - 3, s], {r, 1, 5}];
  VectorDelta[end, s] = Table[Bound[Delta, end, r - 3, s], {r, 1, 5}];
  VectorDelta[iotaI, s] = Table[Bound[Delta, iotaI, r - 3, s], {r, 1, 5}];
  VectorDelta[iotaII, s] = Table[Bound[Delta, iotaII, r - 3, s], {r, 1, 5}];
  MatrixDelta[s] = Table[Bound[Delta, r - 3, t - 3, s], {t, 1, 5}, {r, 1, 5}];
  , {s, {i, o}}]
```

To compute the geometric sum over matricies we compute a representation of $P^{(1)}$ and $P^{(1)\iota}$ by eigenvalue of the matricies A:

In[2323]:=
```
Do[
  EigenA[s] = Eigensystem[Transpose[MatrixA[s]]];
  InverseProduct[normal, s] =
   Inverse[Transpose[EigenA[s][[2]]]].VectorP1[normal, s];
  InverseProduct[iota, s] = Inverse[Transpose[EigenA[s][[2]]]].VectorP1[iota, s];
  Do[
   EigenVector[normal, j, s] = EigenA[s][[2, j]] * InverseProduct[normal, s][[j]];
   EigenVector[iota, j, s] = EigenA[s][[2, j]] * InverseProduct[iota, s][[j]];
   EigenValue[j, s] = EigenA[s][[1, j]];
   , {j, 1, 5}]
  , {s, {i, o}}]
```

Bound for k=0

Now we first implement the bound on the absolute values of the coefficients stated in Lemma 4.3.7,4.3.8 and Proposition 4.3.9

In[2324]:=
```
Do[
  Bound[Xi, normal, 0, s] = 1;
  Bound[Xi, iota, 0, s] = rho[s] Bound[tG1i, s];

  Bound[Xi, normal, 1, s] = rho[s] Bound[P1, 0, 0, s];
  Bound[Xi, iota, 1, s] = rho[s]/(2 d) Bound[P1, IotaStep, 0, s] +
    rho[s]/(2 d) Bound[P1, IotaRib, 0, s];
  factor[normal] = rho[s];
  factor[iota] = rho[s]/(2 d);
  Do[
   Bound[Xi, t, 2, s] = factor[t] VectorP1[t, s].VectorAbar[s];
   Bound[Xi, t, 3, s] =
    factor[t] VectorP1[t, s].MatrixAbar[s].VectorP1[normal, s];
   Bound[Xi, t, EvenTail, s] =
    Bound[Xi, t, 2, s] +
     Abs[
      factor[t]
       Sum[EigenVector[t, j, s].MatrixAbar[s].VectorP1[normal, s] /
           ((1 - EigenValue[j, s]^2) EigenValue[j, s]), {j, 1, 5}]];
   Bound[Xi, t, OddTail, s] =
    Bound[Xi, t, 3, s] +
     Abs[
      factor[t]
       Sum[EigenVector[t, j, s].MatrixAbar[s].VectorP1[normal, s] /
           ((1 - EigenValue[j, s]^2) EigenValue[j, s]^2), {j, 1, 5}]];
   , {t, {normal, iota}}];
  Bound[Xi, normal, Even, s] = Bound[Xi, normal, EvenTail, s];
  (* recall here that extract the contribtion of Ξ^(0)(x)=δ_{0,x} in the analysis.*)
  Bound[Xi, iota, Even, s] = Bound[Xi, iota, 0, s] + Bound[Xi, iota, EvenTail, s];

  Do[
   Bound[Xi, t, Odd, s] = Bound[Xi, t, 1, s] + Bound[Xi, t, OddTail, s];
   Bound[Xi, t, Absolut, s] = Bound[Xi, t, Odd, s] + Bound[Xi, t, Even, s];
   , {t, {normal, iota}}];
  , {s, {i, o}}];
```

Bounds for $\hat{\Xi}^{(N)}(0)$-$\hat{\Xi}^{(N)}(k)$ for N=0,1,2,3

We now compute the bound as given in Lemma Lemma 4.3.7,4.3.8 and Proposition 4.3.9

In[2325]:=
```
Do[
 Bound[Xi, iota, 1, Delta, 0, s] = rho[s] Bound[Delta, iotaI, 0, s];
 Bound[Xi, iota, 1, Delta, ei, s] = rho[s] Bound[Delta, iotaII, 0, s];
 Bound[Xi, normal, 1, Delta, 0, s] = rho[s] Bound[Delta, I, 0, s];
 , {s, {i, o}}]
```

Bound for N=2

In[2326]:=
```
Do[
  Bound[Xi, normal, 2, Delta, 0, s] =
   2 rho[s] (VectorP1[normal, s].VectorDelta[end, s] +
      VectorDelta[start, s].VectorP1[normal, s]);
  Bound[Xi, iota, 2, Delta, 0, s] =
   2 rho[s]
    (VectorP1[iota, s].VectorDelta[end, s] +
      VectorDelta[iotaI, s].VectorP1[normal, s]);
  Bound[Xi, iota, 2, Delta, ei, s] =
   2 rho[s]
    (VectorP1[iota, s].VectorDelta[end, s] +
      VectorDelta[iotaII, s].VectorP1[normal, s]);
  , {s, {i, o}}]
```

Bound for N=3

In[2327]:=
```
Do[
  Bound[Xi, normal, 3, Delta, 0, s] =
   3 rho[s] VectorDelta[start, s].MatrixA[s].VectorP1[normal, s] +
    3 rho[s] VectorP1[normal, s].MatrixA[s].VectorDelta[end, s] +
    3 rho[s] VectorP1[normal, s].MatrixDelta[s].VectorP1[normal, s];
  Bound[Xi, iota, 3, Delta, 0, s] =
   3 rho[s] VectorDelta[iotaI, s].MatrixA[s].VectorP1[normal, s] +
    3 rho[s] VectorP1[iota, s].MatrixA[s].VectorDelta[end, s] +
    3 rho[s] VectorP1[iota, s].MatrixDelta[s].VectorP1[normal, s];
  Bound[Xi, iota, 3, Delta, ei, s] =
   3 rho[s] VectorDelta[iotaII, s].MatrixA[s].VectorP1[normal, s] +
    3 rho[s] VectorP1[iota, s].MatrixA[s].VectorDelta[end, s] +
    3 rho[s] VectorP1[iota, s].MatrixDelta[s].VectorP1[normal, s];
  , {s, {i, o}}]
```

Bounds for $\hat{\Xi}(0)$-$\hat{\Xi}(k)$ for N≥4

Bounds for $\hat{\Xi}(0)$-$\hat{\Xi}(k)$ for N>=4

We compute the sum over the bound of Proposition 4.3.9 over even N using the technique of Section 5.3.

In[2328]:=
```
Do[
   Bound[Xi, normal, EvenTail, Delta, 0, s] =
    Abs[
     rho[s] 2
      (Sum[VectorDelta[start, s].EigenVector[normal, j, s] EigenValue[j, s]^2
           (1/(1 - EigenValue[j, s]^2)^2 + 1/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
        Sum[EigenVector[normal, j, s].VectorDelta[end, s] EigenValue[j, s]^2
           (1/(1 - EigenValue[j, s]^2)^2 + 1/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
        Sum[(EigenVector[normal, j, s] EigenValue[j, s]^3) / (1 - EigenValue[j, s]^2)^2,
           {j, 1, 5}].Sum[EigenVector[normal, j, s]/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}] +
        Sum[(EigenVector[normal, j, s] EigenValue[j, s]^2) / (1 - EigenValue[j, s]^2)^2,
           {j, 1, 5}].Sum[(EigenVector[normal, j, s] EigenValue[j, s]) /
```

```
        (1 - EigenValue[j, s]^2)^2, {j, 1, 5}])];

Bound[Xi, iota, EvenTail, Delta, 0, s] =
 Abs[
  rho[s] 2
   (Sum[VectorDelta[iotaI, s].EigenVector[normal, j, s] EigenValue[j, s]^2
       (1/(1 - EigenValue[j, s]^2)^2 + 1/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
     Sum[EigenVector[iota, j, s].VectorDelta[end, s] EigenValue[j, s]^2
       (1/(1 - EigenValue[j, s]^2)^2 + 1/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
     Sum[(EigenVector[iota, j, s] EigenValue[j, s]^3)/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}].
      Sum[EigenVector[normal, j, s]/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}] +
     Sum[(EigenVector[iota, j, s] EigenValue[j, s]^2)/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}].
      Sum[(EigenVector[normal, j, s] EigenValue[j, s]) / (1 - EigenValue[j, s]^2)^2,
       {j, 1, 5}])];

Bound[Xi, iota, EvenTail, Delta, ei, s] =
 Abs[
  rho[s] 2
   (Sum[VectorDelta[iotaII, s].EigenVector[normal, j, s] EigenValue[j, s]^2
       (1/(1 - EigenValue[j, s]^2)^2 + 1/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
     Sum[EigenVector[iota, j, s].VectorDelta[end, s] EigenValue[j, s]^2
       (1/(1 - EigenValue[j, s]^2)^2 + 1/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
     Sum[(EigenVector[iota, j, s] EigenValue[j, s]^3)/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}].
      Sum[EigenVector[normal, j, s]/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}] +
     Sum[(EigenVector[iota, j, s] EigenValue[j, s]^2)/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}].
      Sum[(EigenVector[normal, j, s] EigenValue[j, s]) / (1 - EigenValue[j, s]^2)^2,
       {j, 1, 5}])];
 {s, {i, 2}}];
```

Now we compute the sum over odd N

In[2329]:=
```
Do[
  Bound[Xi, normal, OddTail, Delta, 0, s] =
   Abs[
    rho[s]
     (2 Sum[VectorDelta[start, s].EigenVector[normal, j, s] EigenValue[j, s]^2
           (1/(1 - EigenValue[j, s]^2)^2 + 2/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
      2 Sum[EigenVector[normal, j, s].VectorDelta[end, s] EigenValue[j, s]^2
           (1/(1 - EigenValue[j, s]^2)^2 + 2/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
      2 Sum[(EigenVector[normal, j, s] EigenValue[j, s]^3) / (1 - EigenValue[j, s]^2)^2,
          {j, 1, 5}].Sum[EigenVector[normal, j, s]/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}] -
      Sum[(EigenVector[normal, j, s] EigenValue[j, s]^2) / (1 - EigenValue[j, s]^2)^2,
         {j, 1, 5}].Sum[(EigenVector[normal, j, s] EigenValue[j, s]) /
          (1 - EigenValue[j, s]^2)^2, {j, 1, 5}])];
  Bound[Xi, iota, OddTail, Delta, 0, s] =
   Abs[
    rho[s]
     (2 Sum[VectorDelta[iotaI, s].EigenVector[normal, j, s] EigenValue[j, s]^2
           (1/(1 - EigenValue[j, s]^2)^2 + 2/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
      2 Sum[EigenVector[iota, j, s].VectorDelta[end, s] EigenValue[j, s]^2
           (1/(1 - EigenValue[j, s]^2)^2 + 2/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
      2 Sum[(EigenVector[iota, j, s] EigenValue[j, s]^3) / (1 - EigenValue[j, s]^2)^2,
          {j, 1, 5}].Sum[EigenVector[normal, j, s]/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}] -
      Sum[(EigenVector[iota, j, s] EigenValue[j, s]^2)/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}].
       Sum[(EigenVector[normal, j, s] EigenValue[j, s]) / (1 - EigenValue[j, s]^2)^2,
         {j, 1, 5}])];

  Bound[Xi, iota, OddTail, Delta, ei, s] =
   Abs[
    rho[s]
```

```
(2 Sum[VectorDelta[iotaII, s].EigenVector[normal, j, s] EigenValue[j, s]^2
      (1/(1 - EigenValue[j, s]^2)^2 + 2/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
   2 Sum[EigenVector[iota, j, s].VectorDelta[end, s] EigenValue[j, s]^2
      (1/(1 - EigenValue[j, s]^2)^2 + 2/(1 - EigenValue[j, s]^2)), {j, 1, 5}] +
   2 Sum[(EigenVector[iota, j, s] EigenValue[j, s]^3) / (1 - EigenValue[j, s]^2)^2,
      {j, 1, 5}].Sum[EigenVector[normal, j, s]/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}] -
   Sum[EigenVector[iota, j, s] EigenValue[j, s]^2/(1 - EigenValue[j, s]^2)^2, {j, 1, 5}].
    Sum[(EigenVector[normal, j, s] EigenValue[j, s]) / (1 - EigenValue[j, s]^2)^2,
     {j, 1, 5}]) ];
 , {s, {i, o}}];
```

Summation of the Delta Bounds

```
In[2330]:= Do[
  Do[
   Bound[Xi, t, Even, Delta, 0, s] =
    Bound[Xi, t, 0, Delta, 0, s] + Bound[Xi, t, 2, Delta, 0, s] +
     Bound[Xi, t, EvenTail, Delta, 0, s];
   Bound[Xi, t, Odd, Delta, 0, s] =
    Bound[Xi, t, 1, Delta, 0, s] + Bound[Xi, t, 3, Delta, 0, s] +
     Bound[Xi, t, EvenTail, Delta, 0, s];
   Bound[Xi, t, Absolut, Delta, 0, s] =
    Bound[Xi, t, Odd, Delta, 0, s] + Bound[Xi, t, Even, Delta, 0, s];
   , {t, {normal, iota}}];

  Bound[Xi, iota, Even, Delta, ei, s] =
   Bound[Xi, iota, 0, Delta, ei, s] + Bound[Xi, iota, 2, Delta, ei, s] +
    Bound[Xi, iota, EvenTail, Delta, ei, s];
  Bound[Xi, iota, Odd, Delta, ei, s] =
   Bound[Xi, iota, 1, Delta, ei, s] + Bound[Xi, iota, 3, Delta, ei, s] +
    Bound[Xi, iota, OddTail, Delta, ei, s];
  Bound[Xi, iota, Absolut, Delta, ei, s] =
   Bound[Xi, iota, Odd, Delta, ei, s] + Bound[Xi, iota, Even, Delta, ei, s];
  , {s, {i, o}}]
```

Computation of constants of Proposition 3.3.1

In this section we perform the computations of Section 3.4. to obtain the constances stated in Proposition 3.3.1:

$$\begin{gathered}\sum_x |F(x)| \leq K_F = \text{Bound[KF]}\\ \text{Bound[KPhi, 1]} = \underline{K}_\Phi \leq \hat{\Phi}(0) \leq \overline{K}_\Phi = \text{Bound[KPhi, 2]}\\ \text{Bound[KPhiabs, 1]} = \underline{K}_{|\Phi|} \leq \sum_x \left|\Phi(x)\right| \leq \overline{K}_{|\Phi|} = \text{Bound[KPhiabs, 2]}\\ \sum_{x\neq 0} |\Phi_z(x)| \leq K_{|\Phi|} = \text{Bound[KPhiWithoutZero]}\end{gathered} \tag{3}$$

$$\sum_x F(x)[1-\cos(k\,x)] \geq K_{\text{Lower}}\left[1-\hat{D}(k)\right]$$
$$\sum_x \left|F(x)\right|[1-\cos(k\,x)] \leq K_{\Delta\text{F}}\left[1-\hat{D}(k)\right] \quad (4)$$
$$\sum_x \left|\Phi_z(x)\right|[1-\cos(k\,x)] \leq K_{\Delta\Phi}\left[1-\hat{D}(k)\right]$$

Bound on absolute value K_F and K_Φ

In[2331]:=
```
Do[
 alpha[s] = twodgjz[s] / (2 d);
 baralpha[s] = twodgz[s] / (2 d);
 Bound[KPsi, s] = rho[s] + (2 d - 2)/(2 d) Bound[Xi, normal, Absolut, s];
 Bound[KF, s] =
  (2 d baralpha[s]) / (1 - alpha[s] - (2 d - 2) alpha[s] Bound[Xi, iota, Absolut, s])
   Bound[KPsi, s];
 Bound[KPhiup, s] = 1 + Bound[Xi, normal, Even, s] +
   Bound[Xi, iota, Absolut, s] Bound[KF, s];
 Bound[KPhidown, s] = 1 - Bound[Xi, normal, Odd, s] -
   Bound[Xi, iota, Absolut, s] Bound[KF, s];
 Bound[KPhiabsup, s] = 1 + Bound[Xi, normal, Absolut, s] +
   Bound[Xi, iota, Absolut, s] Bound[KF, s];
 Bound[KPhiabsdown, s] = 1 - Bound[Xi, normal, Absolut, s] -
   Bound[Xi, iota, Absolut, s] Bound[KF, s];
 Bound[KPhiWithoutZero, s] =
  Bound[Xi, normal, Absolut, s] + Bound[Xi, iota, Absolut, s] Bound[KF, s];
 , {s, {i, o}}]
```

Bounds on differences

Next we implement the computation of Section 3.4.3. First the differences of F_1 and Φ_1, lines (3.4.26), (3.4.27), (3.4.29)

```
In[2332]:= Bound[DifferencefF, Part1, Lower, i] =
    2 d baralpha[i]
    ---------------
    1 - alpha[i]^2
    (1 - Bound[tG1i, i] - Bound[Xi, normal, Odd, Delta, 0, i] - Bound[Xi, normal, Odd, i] -
      alpha[i] Bound[Xi, normal, Even, Delta, 0, i]);
  Bound[DifferencefF, Part1, Absolut, i] =
    2 d baralpha[i]
    ---------------
    1 - alpha[i]^2
    (rho[i] + (1 + alpha[i]) Bound[Xi, normal, Absolut, Delta, 0, i] +
      Bound[Xi, normal, Absolut, i]);
  Bound[DifferencefF, Part1, Lower, o] =
    Min[2 d baralpha[i] / (1 - alpha[i]^2), 2 d baralpha[o] / (1 - alpha[o]^2)]
    (1 - Bound[tG1i, o] - Bound[Xi, normal, Odd, Delta, 0, o] - Bound[Xi, normal, Odd, o] -
      alpha[o] Bound[Xi, normal, Even, Delta, 0, o]);
  Bound[DifferencefF, Part1, Absolut, o] =
    Max[2 d baralpha[i] / (1 - alpha[i]^2), 2 d baralpha[o] / (1 - alpha[o]^2)]
    (rho[i] + (1 + alpha[o]) Bound[Xi, normal, Absolut, Delta, 0, o] +
      Bound[Xi, normal, Absolut, o]);
  Do[
   Bound[KDeltaPhi, Part1, s] = Bound[Xi, normal, Absolut, Delta, 0, s] +
      baralpha[s]
      --------------
      1 - alpha[s]^2
      (2 d Bound[Xi, normal, Absolut, Delta, 0, s] Bound[Xi, iota, Absolut, s] +
        (1 + Bound[Xi, normal, Absolut, s]) Bound[Xi, iota, Absolut, Delta, ei, s] +
        2 d alpha[s] Bound[Xi, normal, Absolut, Delta, 0, s]
         Bound[Xi, iota, Absolut, s] +
        alpha[s] (1 + Bound[Xi, normal, Absolut, s])
         Bound[Xi, iota, Absolut, Delta, 0, s]);
   , {s, {i, o}}]
```

Then the differences of F_2 and Φ_2: For the bound in (3.4.30), (3.4.32) and (3.4.35)

```
In[2337]:= Do[
  Bound[DifferencefF, Part2, Lower, s] =
   -((2 d baralpha[s])^2/(1 - alpha[s]^2))
     (Bound[Xi, normal, Odd, Delta, 0, s] Bound[Xi, iota, Odd, s] +
       Bound[Xi, normal, Even, Delta, 0, s] Bound[Xi, iota, Even, s])
    - (2 d baralpha[s]^2/(1 - alpha[s]^2)) (rho[s] + (2 d - 2)/(2 d) Bound[Xi, normal, Even, s])
     ( Bound[Xi, iota, Even, Delta, ei, s] + 2 d Bound[Xi, iota, Even, s] +
       alpha[s]^2 Bound[Xi, iota, Even, Delta, 0, s] +
       alpha[s] (Bound[Xi, iota, Odd, Delta, ei, s] + Bound[Xi, iota, Odd, Delta, 0, s] +
          2 d Bound[Xi, iota, Odd, s])) -
    (2 d baralpha[s]^2/(1 - alpha[s]^2)) ((2 d - 2)/(2 d) Bound[Xi, normal, Odd, s])
     ( Bound[Xi, iota, Odd, Delta, ei, s] + 2 d Bound[Xi, iota, Odd, s] +
       alpha[s]^2 Bound[Xi, iota, Odd, Delta, 0, s] +
       alpha[s] (Bound[Xi, iota, Even, Delta, ei, s] +
          Bound[Xi, iota, Even, Delta, 0, s] + 2 d Bound[Xi, iota, Even, s]));
  Bound[DifferencefF, Part2, Absolut, s] =
   ((2 d baralpha[s])^2/(1 - alpha[s]^2)) Bound[Xi, normal, Absolut, Delta, 0, s]
     Bound[Xi, iota, Absolut, s]
    + (2 d baralpha[s]^2/((1 - alpha[s]^2) (1 + alpha[s]))) (rho[s] + (2 d - 2)/(2 d) Bound[Xi, normal, Absolut, s])
     ( Bound[Xi, iota, Absolut, Delta, ei, s] + 2 d Bound[Xi, iota, Absolut, s] +
       alpha[s] Bound[Xi, iota, Absolut, Delta, 0, s]);
  Bound[KDeltaPhi, Part2, s] =
   (2 d baralpha[s]^2/(1 - alpha[s]^2))
    (2 d Bound[Xi, normal, Absolut, Delta, 0, s] Bound[Xi, iota, Absolut, s]^2 +
      2 (1 + Bound[Xi, normal, Absolut, s])
       (Bound[Xi, iota, Absolut, s]/(1 + alpha[s]))
        (Bound[Xi, iota, Absolut, Delta, ei, s] +
          alpha[s] Bound[Xi, iota, Absolut, Delta, 0, s]));
  , {s, {i, o}}]
```

Finally, we compute the differences of $F_3$and Φ_3, lines (4.4.37) and (4.4.38)

In[2338]:=
```
Do[
 tmp = 1 / (1 - (2 d alpha[s] Bound[Xi,iota,Absolut,s])/(1-alpha[s]));
 Bound[DifferencefF, Part3, Absolut, s] =
  Bound[Xi, normal, Absolut, Delta, 0, s] (2 d baralpha[s])/(1 - alpha[s])^3
    (2 d alpha[s] Bound[Xi, iota, Absolut, s])^2 tmp +
   (1 + Bound[Xi, normal, Absolut, s]) (baralpha[s] (2 d alpha[s])^2)/((1 - alpha[s])^2 (1 - alpha[s]^2))
    Bound[Xi, iota, Absolut, s] tmp^2
    (Bound[Xi, iota, Absolut, Delta, ei, s] +
      alpha[s] Bound[Xi, iota, Absolut, Delta, 0, s]) +
   (1 + Bound[Xi, normal, Absolut, s]) (baralpha[s] (2 d alpha[s])^2)/((1 - alpha[s])^2 (1 - alpha[s]^2))
    Bound[Xi, iota, Absolut, s] tmp
    (Bound[Xi, iota, Absolut, Delta, ei, s] +
      alpha[s] Bound[Xi, iota, Absolut, Delta, 0, s] +
      2 d Bound[Xi, iota, Absolut, s]);

 Bound[DifferencefF, Part3, Lower, s] = -Bound[DifferencefF, Part3, Absolut, s];
 Bound[KDeltaPhi, Part3, s] =
  Bound[Xi, normal, Absolut, Delta, 0, s] (2 d baralpha[s] Bound[Xi, iota, Absolut, s])/(1 - alpha[s])^3
    (2 d alpha[s] Bound[Xi, iota, Absolut, s])^2 tmp +
   (1 + Bound[Xi, normal, Absolut, s])
    (baralpha[s] (2 d alpha[s] Bound[Xi, iota, Absolut, s])^2) /
     ((1 - alpha[s])^2 (1 - alpha[s]^2)) (tmp^2 + tmp)
    1/(1 + alpha[s]) (Bound[Xi, iota, Absolut, Delta, ei, s] +
      alpha[s] Bound[Xi, iota, Absolut, Delta, 0, s]);

 Bound[KDeltaFLower, s] =
  1 / (Bound[DifferencefF, Part1, Lower, s] + Bound[DifferencefF, Part2, Lower, s] +
     Bound[DifferencefF, Part3, Lower, s]);
 Bound[KDeltaF, s] = Bound[DifferencefF, Part1, Absolut, s] +
   Bound[DifferencefF, Part2, Absolut, s] + Bound[DifferencefF, Part3, Absolut, s];
 Bound[KDeltaPhi, s] = Bound[KDeltaPhi, Part1, s] + Bound[KDeltaPhi, Part2, s] +
   Bound[KDeltaPhi, Part3, s];
 Clear[tmp];
 , {s, {i, o}}]
```

Check of the sufficient condition

Now we can compute whether P(γ,Γ,z) is satisfied, see Definition 3.3.2.

In[2339]:=
```
Do[
  NoBLEBoundF1[s] = (1 + (2 d-2)/(2 d-1) Gamma1 Bound[Xi, iota, Even, s]) / (rho[s] - (2 d-2)/(2 d) Bound[Xi, normal, Odd, s]);
  NoBLEBoundF2[s] = (2 d - 2)/(2 d - 1) Bound[KPhiabsup, s] Bound[KDeltaFLower, s];
  , {s, {i, o}}]
```

and we compute for f_3

In[2340]:=
```
Do[
  NoBLEBoundF3[Part1, s] = 1/(2 c1) Bound[KDeltaFLower, s] Bound[KDeltaPhi, s];
  NoBLEBoundF3[Part2, s] = 1/(2 c2) Bound[KPhiabsup, s] Bound[KDeltaF, s]
    Bound[KDeltaFLower, s]^2;
  NoBLEBoundF3[Part3, s] =
   2 Bound[KDeltaFLower, s]^2 / c3
    √(Bound[KDeltaF, s] Bound[KDeltaPhi, s] Bound[KPhiWithoutZero, s] Bound[KF, s]);
  NoBLEBoundF3[Part4, s] =
   2 Bound[KDeltaFLower, s]^2 / c4
    (2 Bound[KPhiabsup, s] Bound[KDeltaF, s]^2 Bound[KDeltaFLower, s] +
      √(Bound[KDeltaF, s] Bound[KDeltaPhi, s] Bound[KPhiWithoutZero, s]
         Bound[KF, s]));
  NoBLEBoundF3[s] = Max[NoBLEBoundF3[Part1, s], NoBLEBoundF3[Part2, s],
    NoBLEBoundF3[Part3, s], NoBLEBoundF3[Part4, s]];
  , {s, {i, o}}]
```

We finally check

In[2341]:=
```
Do[
  Succes[f1, s] = NoBLEBoundF1[s] < Gamma1;
  Succes[f2, s] = NoBLEBoundF2[s] < Gamma2;
  Succes[f3, s] = NoBLEBoundF3[s] < Gamma3;
  Succes[s] = Succes[f1, s] && Succes[f2, s] && Succes[f3, s];
  , {s, {i, o}}]
Succes[overall] = Succes[i] && Succes[o];
```

Result

The overall result

The statement that the bootstrap was succesful is

In[2343]:=
```
Succes[overall]
```

Out[2343]=
```
True
```

If this succedes than the analysis of Section 3.3 can be used to proved mean-field behavoior for LT.

Assuming the bootstrap was succesful we prove the following bounds: The infrared bound in (3.3.12) and (3.3.13) hold with

In[2344]:=
```
(2d-2)/(2d-1) Gamma2(* ≥ G_z(k)[1-D̂(k)]*)
Max[Bound[KDeltaFLower, o], 1]
(* Nominator in (4.3.13) *)
```

Out[2344]= 1.09746

Out[2345]= 1.11069

Further, we have proven that $g_{z_c} z_c$ is smaller than

In[2346]:=
```
1/(2d-1) Gamma1
```

Out[2346]= 0.0181255

and that g_{z_c} smaller than

In[2347]:=
```
Gamma1 * Exp[1]
```

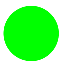
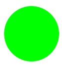
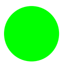
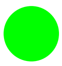
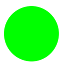
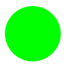
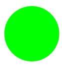
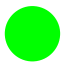
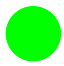
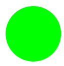
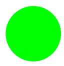
Out[2347]= 2.80841

The improvement of bounds

In[2348]:=
```
bubbles = {Graphics[{Green, Disk[{0, 0}, 0.8]}, ImageSize -> 30],
   Graphics[{Red, Disk[{0, 0}, 0.8]}, ImageSize -> 40]};
tableClassicCheck =
  {{Bounds, Init - f1, Init - f2, Init1 - f3, Init2 - f3, Init3 - f3, Init4 - f3,
    f1, f2, f31, f32, f33, f34}, {Gamma, Gamma1, Gamma2, Gamma3 , Gamma3 ,
    Gamma3 , Gamma3 , Gamma1, Gamma2, Gamma3, Gamma3 , Gamma3 , Gamma3 },
   {Bounds, NoBLEBoundF1[i], NoBLEBoundF2[i], NoBLEBoundF3[Part1, i] ,
    NoBLEBoundF3[Part2, i] , NoBLEBoundF3[Part3, i], NoBLEBoundF3[Part4, i] ,
    NoBLEBoundF1[o], NoBLEBoundF2[o], NoBLEBoundF3[Part1, o] ,
    NoBLEBoundF3[Part2, o] , NoBLEBoundF3[Part3, o] , NoBLEBoundF3[Part4, o] },
   { check,
    If[NoBLEBoundF1[i] < Gamma1, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF2[i] < Gamma2, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF3[Part1, i] < Gamma3, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF3[Part2, i] < Gamma3, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF3[Part3, i] < Gamma3, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF3[Part4, i] < Gamma3, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF1[o] < Gamma1, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF2[o] < Gamma2, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF3[Part1, o] < Gamma3, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF3[Part2, o] < Gamma3, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF3[Part3, o] < Gamma3, bubbles[[1]], bubbles[[2]]],
    If[NoBLEBoundF3[Part4, o] < Gamma3, bubbles[[1]], bubbles[[2]]],},
   {Required, NoBLEBoundF1[i], NoBLEBoundF2[i], NoBLEBoundF3[Part1, i] / Gamma3 ,
    NoBLEBoundF3[Part2, i]  / Gamma3, NoBLEBoundF3[Part3, i] * c3 / Gamma3 ,
    NoBLEBoundF3[Part4, i] * c4  / Gamma3, NoBLEBoundF1[o], NoBLEBoundF2[o],
    NoBLEBoundF3[Part1, o] * c1 / Gamma3 , NoBLEBoundF3[Part2, o]  * c2 / Gamma3,
    NoBLEBoundF3[Part3, o] * c3 / Gamma3, NoBLEBoundF3[Part4, o] * c4 / Gamma3}};
Labeled[Grid[tableClassicCheck, Alignment → {Center}, Frame → True,
   Dividers → {{2 → True, -1 → True}, {2 → True}}, ItemStyle → {1 → Bold, 1 → Bold},
   Background → {Automatic, Automatic, {{2, 2}, {2, 15}} → GrayLevel[0.7]}],
  Style["Result for Dimension " Text[d], Bold], Top] // Text
```

Out[2350]=

Result for Dimension 29

Boun⋱ ds	**Init – f_1**	**Init – f_2**	**Init1 – f_3**	**Init2 – f_3**	**Init3 – f_3**	**Init4 – f_3**	**f_1**	**f_2**	**f_{31}**	**f_{32}**	**f_{33}**	**f_{34}**
Gam⋱ ma	1.033⋱ 16	1.117⋱ 06	1.2	1.2	1.2	1.2	1.033⋱ 16	1.117⋱ 06	1.2	1.2	1.2	1.2
Boun⋱ ds	1.031⋱ 92	1.0578	0.331⋱ 929	0.863⋱ 446	0.496⋱ 682	0.652⋱ 354	1.033⋱ 15	1.117⋱ 05	1.199⋱ 85	1.199⋱ 97	1.199⋱ 93	1.199⋱ 99
check												
Requ⋱ ire⋱ d	1.031⋱ 92	1.0578	0.276⋱ 608	0.719⋱ 538	0.094⋱ 909⋱ 3	5.294⋱ 57	1.033⋱ 15	1.117⋱ 05	0.157⋱ 281	0.750⋱ 444	0.229⋱ 291	9.739⋱ 21

In the following we give a semi-automate proceedure to find appropiate value for the constants $\Gamma_i$and c_i.

Initially we guess a good value for the constant and make a first computation. Then we deactivate there initial definition at the top of the document and use the code below. We recompile the document a number of times and hope that the algorithm below converges to a fixed point.

In[2351]:=
```
(*d=29;
{d,Gamma1,Gamma2,Gamma3,c1,c2,c3,c4}
 Gamma1=NoBLGamma1=NoBLEBoundF1[o]+0.000001;
Gamma2=NoBLEBoundF2[o]+0.000001;
c1=NoBLEBoundF3[Part1,o]*c1/Gamma3+0.00001;
c2=NoBLEBoundF3[Part2,o]*c2/Gamma3+0.00001;
c3=NoBLEBoundF3[Part3,o]*c3/Gamma3+0.00001;
c4=NoBLEBoundF3[Part4,o]*c4/Gamma3+0.0001;
{d,Gamma1,Gamma2,Gamma3,c1,c2,c3,c4}*)
```

{32,1.02945,1.06828,1.2,0.0689682,0.587369,0.122415,6.22451}

{30,1.03181,1.09078,1.2,0.106636,0.658126,0.170443,7.66509}

{29,1.03316,1.11704,1.2,0.157265,0.750406,0.229273,9.73817}

Print out of the computed bounds in the coefficients

In[2352]:=
```
Do[
 MethodeFourTable[s] = {{Quantity, Ξ^Zero, Ξ^One, Ξ^Two, Ξ^Three, Ξ^EvenTail, Ξ^OddTail},
    {Text[Bound for Ξ̂], Bound[Xi, normal, 0, s], Bound[Xi, normal, 1, s] ,
     Bound[Xi, normal, 2, s], Bound[Xi, normal, 3, s], Bound[Xi, normal, EvenTail, s],
     Bound[Xi, normal, OddTail, s]},
    {Text[Bound for Ξ̂^ι], Bound[Xi, iota, 0, s], Bound[Xi, iota, 1, s] ,
     Bound[Xi, iota, 2, s], Bound[Xi, iota, 3, s], Bound[Xi, iota, EvenTail, s],
     Bound[Xi, iota, OddTail, s]},
    {Text[Ξ̂ (1 - cos (kx))], Bound[Xi, normal, 0, Delta, 0, s],
     Bound[Xi, normal, 1, Delta, 0, s], Bound[Xi, normal, 2, Delta, 0, s],
     Bound[Xi, normal, 3, Delta, 0, s], Bound[Xi, normal, EvenTail, Delta, 0, s],
     Bound[Xi, normal, OddTail, Delta, 0, s]},
    {Text[ Ξ^ι (1 - cos (kx))], Bound[Xi, iota, 0, Delta, 0, s],
     Bound[Xi, iota, 1, Delta, 0, s], Bound[Xi, iota, 2, Delta, 0, s],
     Bound[Xi, iota, 3, Delta, 0, s], Bound[Xi, iota, EvenTail, Delta, 0, s],
     Bound[Xi, iota, OddTail, Delta, 0, s]},
    {Text[Ξ^ι (1 - cos (k (x - e_ι)))] , Bound[Xi, iota, 0, Delta, ei, s],
     Bound[Xi, iota, 1, Delta, ei, s] , Bound[Xi, iota, 2, Delta, ei, s],
     Bound[Xi, iota, 3, Delta, ei, s], Bound[Xi, iota, EvenTail, Delta, ei, s],
     Bound[Xi, iota, OddTail, Delta, ei, s]}};
 , {s, {i, o}}]
MethodeFourTablePart1 = {{Quantity, KF, KPhi, DELTAFLower, DELTAFAbsolut, DELTAPhi},
   {Bound for , Bound[KF, o], Bound[KPhiup, o], Bound[KDeltaFLower, o],
    Bound[KDeltaF, o], Bound[KDeltaPhi, o]}};
MethodeFourTablePart2 =
  {{Quantity, DELTAFLower, 2, 3, DELTAFAbsolut, 2, 3, DELTAPhi, 2, 3},
   {Bound for , Bound[DifferencefF, Part1, Lower, o],
    Bound[DifferencefF, Part2, Lower, o], Bound[DifferencefF, Part3, Lower, o],
    Bound[DifferencefF, Part1, Absolut, o], Bound[DifferencefF, Part2, Absolut, o],
    Bound[DifferencefF, Part3, Absolut, o], Bound[KDeltaPhi, Part1, o],
    Bound[KDeltaPhi, Part2, o], Bound[KDeltaPhi, Part3, o]}};

Labeled[Grid[MethodeFourTable[i], Alignment → {Center}, Frame → True,
   Dividers → {{2 → True, -1 → True}, {2 → True}}, ItemStyle → {1 → Bold, 1 → Bold},
   Background → {{None}, {GrayLevel[0.9]}, {None}}],
  Style["Bound on coefficients at z_I in Dimension " Text[d], Bold], Top] // Text
Labeled[Grid[MethodeFourTable[o], Alignment → {Center}, Frame → True,
   Dividers → {{2 → True, -1 → True}, {2 → True}}, ItemStyle → {1 → Bold, 1 → Bold},
   Background → {{None}, {GrayLevel[0.9]}, {None}}],
  Style["Bound on coefficients in Dimension " Text[d], Bold], Top] // Text

Labeled[Grid[MethodeFourTablePart1, Alignment → {Center}, Frame → True,
   Dividers → {{2 → True, -1 → True}, {2 → True}}, ItemStyle → {1 → Bold, 1 → Bold},
   Background → {{None}, {GrayLevel[0.9]}, {None}}],
  Style["Bound on the constants of Proposition 3.3.1 in Dimension " Text[d], Bold],
  Top] // Text
Labeled[Grid[MethodeFourTablePart2, Alignment → {Center}, Frame → True,
   Dividers → {{2 → True, -1 → True}, {2 → True}}, ItemStyle → {1 → Bold, 1 → Bold},
   Background → {{None}, {GrayLevel[0.9]}, {None}}],
  Style["Bound on the constants of Proposition 3.3.1 in Dimension " Text[d], Bold],
  Top] // Text
```

Bound on coefficients at z_I in Dimension 29

Out[2355]=

Quantity	Ξ^{Zero}	Ξ^{One}	Ξ^{Two}	Ξ^{Three}	$\Xi^{EvenTail}$	$\Xi^{OddTail}$
Bound for $\hat{\Xi}$	1	0.00266084	0.0000562718	9.73869×10^{-7}	0.0000562804	9.73936×10^{-7}
Bound for $\hat{\Xi}^{\iota}$	0.0176746	0.000519668	0.0000124557	1.21361×10^{-7}	0.0000124577	1.21376×10^{-7}
$(1-\cos kx)\,\hat{\Xi}$	0	0.012852	0.0589629	0.00202497	9.32556×10^{-6}	0.0000139882
$(1-\cos kx)\,\Xi^{\iota}$	0	0.0249083	0.160094	0.0177925	0.0000548181	0.0000822259
$\Xi^{\iota}\,(1-\cos k\,(x-e_{\iota}))$	1.02512	0.0830255	0.161012	0.0178035	0.000054914	0.0000823696

Bound on coefficients in Dimension 29

Out[2356]=

Quantity	Ξ^{Zero}	Ξ^{One}	Ξ^{Two}	Ξ^{Three}	$\Xi^{EvenTail}$	$\Xi^{OddTail}$
Bound for $\hat{\Xi}$	1	0.00311422	0.0000802795	1.63684×10^{-6}	0.0000802975	1.63701×10^{-6}
Bound for $\hat{\Xi}^{\iota}$	0.0182831	0.000613387	0.0000176217	2.09508×10^{-7}	0.0000176256	2.09546×10^{-7}
$(1-\cos kx)\,\hat{\Xi}$	0	0.0289464	0.255857	0.0107572	0.0000628946	0.0000943404
$(1-\cos kx)\,\Xi^{\iota}$	0	0.0607706	0.703223	0.0928051	0.000355265	0.000532889
$\Xi^{\iota}\,(1-\cos k\,(x-e_{\iota}))$	1.06042	0.137684	0.704741	0.0928269	0.000355505	0.000533248

Bound on the constants of Proposition 3.3.1 in Dimension 29

Out[2357]=

Quantity	KF	KPhi	DELTAFLower	DELTAFAbsolut	DELTAPhi
Bound for	1.08329	1.02057	1.11069	1.42618	0.339853

Bound on the constants of Proposition 3.3.1 in Dimension 29

Out[2358]=

Quantity	DELTA⋱ FLow⋱ er	2	3	DELTA⋱ FAbs⋱ olut	2	3	DELTA⋱ Phi	2	3
Bound for	0.961212	−0.0586⋱ 633	−0.0022⋱ 108⋱ 3	1.36004	0.06392⋱ 55	0.00221⋱ 083	0.338274	0.00154⋱ 653	0.00003⋱ 29124