Computation for the lace expansion for percolation

Analysis of Section 3.3

Robert Fitzner
University of Technology Eindhoven

Abstract

In this file we performs we the computations of the analysis of the non-backtracking lace expansion for percolation. All references in this version of the notebook are to the PhD thesis of the author.

We expect as input the dimension d and the constance Γ_1,Γ_2,Γ_3, c_1,..,c_4. After choosing these quantities select the menu item Evaluate-> Evaluate Notebook. When thse computations/evaluationg are finished the results are show in tables at the end of the document. There we also show whether the bootstrap and thereby the analysis is 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 4.5. and compute the bounds used for the Analysis in Section 3.3.

10.5.2013

Input

The dimension in which we perfrom the computations

```
In[251]:= d = 38;
```

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

```
In[252]:= Gamma1 = 1.0152616; Gamma2 = 1.023744; Gamma3 = 1.2;
```

To define the bootstrap function f_3 we use the following constants

```
In[253]:= c1 = 0.2256;
          c2 = 0.6994;
          c3 = 0.074216;
          c4 = 9.12551;
```

{38, 1.01526, 1.02374, 1.2, 0.225599, 0.699387, 0.0742245, 9.12546}

Values that are working for d=38 (* deactivated*)

```
In[257]:= (*d=38;
          Gamma1=1.0152616;Gamma2=1.023744;Gamma3=1.2;
          c1=0.2256;
          c2=0.6994;
          c3=0.074216;
          c4=9.12551;*)
```

Simple Random Walk integral

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

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

see Section 5.2, using that

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

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

```
In[258]:= 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[260]:= 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[264]:= 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[408]:= 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[437]=

m n	0.	1.	2.	3.	4.
0.	1.	1.0137	1.0423	1.0877	1.1529
1.	0.	0.013707	0.02859	0.045416	0.065164
2.	0.013158	0.013707	0.014884	0.016826	0.019748
3.	0.	0.00054868	0.0011771	0.001942	0.0029221
4.	0.00051256	0.00054868	0.00062838	0.00076491	0.00098016
5.	0.	0.000036123	0.0000797	0.00013653	0.00021525
6.	0.000032837	0.000036123	0.000043578	0.000056835	0.000078712
7.	0.	3.2852×10^{-6}	7.455×10^{-6}	0.000013257	0.000021877
8.	2.9062×10^{-6}	3.2852×10^{-6}	4.1698×10^{-6}	5.8023×10^{-6}	8.6198×10^{-6}
9.	0.	3.7901×10^{-7}	8.8459×10^{-7}	1.6325×10^{-6}	2.8175×10^{-6}
10.	−1.	3.7901×10^{-7}	5.0558×10^{-7}	7.4791×10^{-7}	1.185×10^{-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 zϵ $(z_I,\ z_c)$:

Definition of key quantities

In[297]:=

```
z[i] = 1/(2 d - 1);
z[o] = Gamma1/(2 d - 1);(* Upper bound on z and thereby also on z_c *)
(*bound on the two-point function*)
VarGamma1[i] = 1;
VarGamma1[o] = Gamma1;
VarGamma2[i] = (2 d - 2)/(2 d - 1); (*G_z(x) ≤B_z(x) ≤ (2d-2)/(2d-1) C(x) *)
VarGamma2[o] = Gamma2 * (2 d - 2)/(2 d - 1); (*Ĝ_z(k) ≤ varGamma2 Ĉ(k), follows from f2*)
VarGamma3[i] = VarGamma2[i];
(*for a bound on Delta at z=
 z_I we replace G with B in x-space before going to Fourier space*)
VarGamma3[o] = Gamma3;
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 NBWs, as given explain in Section 5.1.2 we can use the number if SAW only for SAW and LT:

In[307]:=

```
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) + 6 d c3i;(*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) ) + 4 d c3i;(*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) ) + 8 d c3i + 4 d c5i;
(*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[313]:=
```
Do[
  Bound[G7i, s] = c7i (z[s])^7 + (2 d z[s])^9 * VarGamma2[s] * I110; (* G_{7,z}(e_1) *)
  Bound[G5i, s] = c5i (z[s])^5 + Bound[G7i, s]; (* G_{5,z}(e_1) *)
  Bound[G3i, s] = c3i (z[s])^3 + Bound[G5i, s]; (* G_{3,z}(e_1) *)
  Bound[G1i, s] = z[s] + Bound[G3i, s]; (* G_{1,z}(e_1) *)
  rho[Lower, s] = 1 - Bound[G1i, s];
  rho[s] = 1 - z[i] - c3i z[i]^3 (1 - z[i]) - c5i z[i]^5 (1 - z[i])^2;
  , {s, {i, o}}]
```

Then we compute $G^1_{n,z}(e_1 + e_2)$ and $G^1_{4,z}(2\, e_2)$, see (5.1.25)-(5.1.26) :

In[314]:=
```
Do[
  Bound[G8ik, s] = d/(d - 1) (2 d z[o])^8 VarGamma2[s] I110; (*G_8(e_1+e_2)*)
  Bound[G6ik, s] = c6ik z[s]^6 + VarGamma2[s] I18; (*G_6(e_1+e_2)*)
  Bound[G4ik, s] = Bound[G6ik, s] + (c4ik - 2 (2 d - 3) ) z[s]^4; (*G^1_4(e_1+e_2)*)
  Bound[G2ik, s] = Bound[G4ik, s] + (c2ik - 1) z[s]^2;(*G^1_2(e_1+e_2)*)

  , {s, {i, o}}]
```

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

In[315]:=
```
Do[
  Bound[G6, s] = Max[c6ik z[s]^6, c7i z[s]^7] + (2 d z[s])^8 VarGamma2[s] I18;
  (* Bound for sup_x G_6(x)=Max[G_7(e),G_6(e_1+e_2),sup_x G_8(x)]*)
  Bound[G4, s] = Max[c4ik z[s]^4, c5i z[s]^5] + Bound[G6, s]; (* Bound for sup_x G_4(x)*)
  Bound[G2, s] = Max[c2ik z[s]^2, c3i z[s]^3] + Bound[G4, s]; (* Bound for sup_x G_2(x)*)
  Bound[G1, s] = Max[Bound[G1i, s], Bound[G2, s]]; (* Bound for sup_x G_1(x)*)
  Bound[G4ii, s] = (2 d + 2) z[s]^4 + Bound[G6, s];(* Bound for sup_x G^1_4(2 e_1)*)
  , {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[316]:= Do[
  Bound[ClosedRepLoop, 4, s] = 2 d z[s] Bound[G3i, s];
  Bound[ClosedRepBubble, 4, s] =
   z[s]^4 (2 d c3i) + 3 z[s]^6 (2 d c5i) + 5 z[s]^8 (2 d c7i) + (2 d z[s])^10 VarGamma2[s] I110 +
    (2 d z[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 z[s]^4 (2 d c3i) + (6 + 1 - 4) (6 + 2 - 4)/2 z[s]^6 (2 d c5i) +
    (8 + 1 - 4) (8 + 2 - 4)/2 z[s]^8 (2 d c7i) + (10 - 6) (9 - 6)/2 (2 d z[s])^10 VarGamma2[s] I110 +
    6 (2 d z[s])^10 VarGamma2[s]^2 I210 + (2 d z[s])^10 VarGamma2[s]^3 I310;
  Bound[ClosedRepSquare, 4, s] =
   z[s]^4 (2 d c3i) + 10 z[s]^6 (2 d c5i) + 35 z[s]^8 (2 d c7i) +
    84 (2 d z[s])^10 VarGamma2[s] I110 + (10 - 4) (9 - 4)/2 (2 d z[s])^10 VarGamma2[s]^2 I210 +
    6 (2 d z[s])^10 VarGamma2[s]^3 I310 + (2 d z[s])^10 VarGamma2[s]^4 I410;
  Bound[ClosedRepSquare, 3, s] =
   Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s] +
    Bound[ClosedRepTriangle, 4, s] + Bound[ClosedRepSquare, 4, s];
  Bound[ClosedRepSquare, 2, s] =
   Bound[ClosedRepLoop, 4, s] + 2 Bound[ClosedRepBubble, 3, s] +
    Bound[ClosedRepTriangle, 4, s] + Bound[ClosedRepSquare, 3, s];
  , {s, {i, o}}]
```

Open repulsive diagrams

Then we define the bound on the open repulsive diagrams as in (5.3.38):

```
In[317]:= Do[
  Bound[OpenRepBubble, 1, s] =
   Max[2 c2ik z[s]^2 + 4 c4ik z[s]^4 + 6 c6ik z[s]^6,
     z[s] + 3 c3i z[s]^3 + 5 c5i z[s]^5 + 7 c7i z[s]^7] + 7 (2 d z[s])^8 VarGamma2[s] I18 +
    (2 d z[s])^8 VarGamma2[s]^2 I28;
  Bound[OpenRepBubble, 2, s] =
   Max[c2ik z[s]^2 + 3 c4ik z[s]^4 + 5 c6ik z[s]^6, 2 c3i z[s]^3 + 4 c5i z[s]^5 + 6 c7i z[s]^7] +
    6 (2 d z[s])^8 VarGamma2[s] I18 + (2 d z[s])^8 VarGamma2[s]^2 I28;
  Bound[OpenRepBubble, 3, s] =
   Max[2 c4ik z[s]^4 + 4 c6ik z[s]^6, c3i z[s]^3 + 3 c5i z[s]^5 + 5 c7i z[s]^7] +
    5 (2 d z[s])^8 VarGamma2[s] I18 + (2 d z[s])^8 VarGamma2[s]^2 I28;
  Bound[OpenRepBubble, 4, s] =
   Max[c4ik z[s]^4 + 3 c6ik z[s]^6, 2 c5i z[s]^5 + 4 c7i z[s]^7] + 4 (2 d z[s])^8 VarGamma2[s] I18 +
    (2 d z[s])^8 VarGamma2[s]^2 I28;
  Bound[OpenRepTriangle, 1, s] =
   Max[3 c2ik z[s]^2 + 10 c4ik z[s]^4 + 21 c6ik z[s]^6,
     z[s] + 6 c3i z[s]^3 + 15 c5i z[s]^5 + 28 c7i z[s]^7] +
    ((8 - 1) (7 - 1))/2 (2 d z[s])^8 VarGamma2[s] I18 + 7 (2 d z[s])^8 VarGamma2[s]^2 I28 +
    (2 d z[s])^8 VarGamma2[s]^3 I38;
  Bound[OpenRepTriangle, 2, s] =
   Max[c2ik z[s]^2 + 6 c4ik z[s]^4 + 15 c6ik z[s]^6, 3 c3i z[s]^3 + 10 c5i z[s]^5 + 21 c7i z[s]^7] +
    ((8 - 2) (7 - 2))/2 (2 d z[s])^8 VarGamma2[s] I18 + 6 (2 d z[s])^8 VarGamma2[s]^2 I28 +
    (2 d z[s])^8 VarGamma2[s]^3 I38;
  Bound[OpenRepTriangle, 3, s] =
   Max[3 c4ik z[s]^4 + 10 c6ik z[s]^6, c3i z[s]^3 + 6 c5i z[s]^5 + 15 c7i z[s]^7] +
    ((8 - 3) (7 - 3))/2 (2 d z[s])^8 VarGamma2[s] I18 + 5 (2 d z[s])^8 VarGamma2[s]^2 I28 +
    (2 d z[s])^8 VarGamma2[s]^3 I38;
  Bound[OpenRepTriangle, 4, s] =
   Max[c4ik z[s]^4 + 6 c6ik z[s]^6, 3 c5i z[s]^5 + 10 c7i z[s]^7] +
    ((8 - 4) (7 - 4))/2 (2 d z[s])^8 VarGamma2[s] I18 + 5 (2 d z[s])^8 VarGamma2[s]^2 I28 +
    (2 d z[s])^8 VarGamma2[s]^3 I38;
  Bound[OpenRepSquare, 2, s] =
   Max[c2ik z[s]^2 + 10 c4ik z[s]^4 + 35 c6ik z[s]^6, 3 c3i z[s]^3 + 20 c5i z[s]^5 + 56 c7i z[s]^7] +
    84 (2 d z[s])^8 VarGamma2[s] I18 + ((8 - 2) (7 - 2))/2 (2 d z[s])^8 VarGamma2[s]^2 I28 +
    6 (2 d z[s])^8 VarGamma2[s]^3 I38 + (2 d z[s])^8 VarGamma2[s]^4 I48;
  , {s, {i, o}}]
```

Weighted Diagrams

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[318]:= Do[
  Bound[WeightedClosedBubble, 4, s] =
   (2 d z[s])^4 VarGamma2[s] VarGamma3[s] (Varc1[s] I24 + (2 Varc2[s] + Varc3[s]) I34);
  Bound[WeightedClosedBubble, 3, s] =
   Bound[WeightedClosedBubble, 4, s] +
    2 d z[s]^3 (Bound[G3i, s] + 3 (2 d - 2) d/(d - 1) d/(d - 2) (2 d z[o])^3 VarGamma2[s] I16 +
       5 (2 d - 2) d/(d - 1) Bound[G4, s] + 9 (z[s]^3 + 3 (2 d - 2) z[s]^5 + Bound[G6, s]));

  Bound[WeightedClosedBubble, 2, s] =
   Bound[WeightedClosedBubble, 3, s] + 8 d z[s]^2 Bound[G4ii, s] +
    8 d z[s]^2 (2 d - 2) (z[s]^2 + Bound[G4ik, s]);
  Bound[WeightedClosedBubble, 0, s] =
   Bound[WeightedClosedBubble, 2, s] + 2 d z[s] Bound[G3i, s];


  Bound[WeightedOpenLine, 0, s] =
   VarGamma3[s] (Varc1[s] I10 + (2 Varc2[s] + Varc4[s]) I20);
  Bound[WeightedOpenBubble, 0, s] =
   VarGamma2[s] VarGamma3[s] (Varc1[s] I20 + (2 Varc2[s] + Varc4[s]) I30);
  Bound[WeightedOpenBubble, 1, s] =
   (2 d z[s]) VarGamma2[s] VarGamma3[s]
    (Varc1[s] Sqrt[I20 I22] + (2 Varc2[s] + Varc4[s]) Sqrt[I30 I32]);
  Bound[WeightedOpenBubble, 2, s] =
   (2 d z[s])^2 VarGamma2[s] VarGamma3[s] (Varc1[s] I22 + (2 Varc2[s] + Varc4[s]) I32);
  Bound[WeightedOpenBubble, 3, s] =
   (2 d z[s])^3 VarGamma2[s] VarGamma3[s]
    (Varc1[s] Sqrt[I22 I24] + (2 Varc2[s] + Varc4[s]) Sqrt[I32 I34]);
  , {s, {i, o}}]
```

Building Blocks

Definition of diagrams without weight

Here we define the quantities of Section 4.5.2. The element of the bounds (4.5.30)-(4.5.38). We defined the element in the thesis using bubble, triangles, square and even pentragram. The bound on these diagrams depend only on the number of two-point funtions/pieces without fix lengh and the number of fixed steps.

We define the bound on $P^{(0),b}$define in Table 4.18 as follows:

```
In[319]:= Do[
   Bound[P, 0, s] = 1 + Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s];
   Bound[P, 1, s] = 2 Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s];
   Bound[P, 2, s] = Bound[ClosedRepLoop, 4, s] +
     Bound[ClosedRepBubble, 4, s] + Bound[ClosedRepTriangle, 4, s];
   , {s, {i, o}}];
```

The bound on $A^{a,b}$define in Table 4.19 we declare the variables:

In[320]:=
```
Do[
  Bound[A, 0, 0, s] = Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s];
  Bound[A, 0, 1, s] = Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s];
  Bound[A, 0, 2, s] = Bound[ClosedRepTriangle, 4, s];
  Bound[A, 1, 0, s] =
   1/(2 d z[s]) (Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s]);
  Bound[A, 1, 1, s] =
   1/(2 d z[s]) (Bound[ClosedRepLoop, 4, s] + Bound[ClosedRepBubble, 4, s]);
  Bound[A, 1, 2, s] = Bound[ClosedRepTriangle, 4, s]/(2 d z[s]);
  Bound[A, 2, 0, s] = Bound[OpenRepBubble, 1, s];
  Bound[A, 2, 1, s] = Bound[OpenRepBubble, 2, s];
  Bound[A, 2, 2, s] = Bound[OpenRepTriangle, 3, s];
  , {s, {i, o}}];
```

We defined $A^{i,a,b}$ using diagrams what do not need to be repulsive. Follow Table 4.20 we define

In[321]:=
```
Do[
  Bound[Ai, 0, 0, s] = (2 d z[s])^4 VarGamma2[s] I14 + (2 d z[s])^4 VarGamma2[s]^2 I24;
  Bound[Ai, 0, 1, s] = (2 d z[s])^4 VarGamma2[s] I14 + (2 d z[s])^4 VarGamma2[s]^2 I24;
  Bound[Ai, 0, 2, s] = (2 d z[s])^4 VarGamma2[s]^3 I34;
  Bound[Ai, 1, 0, s] = ((2 d z[s])^4 VarGamma2[s] I14 + (2 d z[s])^4 VarGamma2[s]^2 I24)/(2 d z[s]);
  Bound[Ai, 1, 1, s] = ((2 d z[s])^4 VarGamma2[s] I14 + (2 d z[s])^4 VarGamma2[s]^2 I24)/(2 d z[s]);
  Bound[Ai, 1, 2, s] = (2 d z[s])^3 VarGamma2[s]^3 I34;
  Bound[Ai, 2, 0, s] = (2 d z[s])^2 VarGamma2[s]^2 I22;
  Bound[Ai, 2, 1, s] = (2 d z[s])^2 VarGamma2[s]^2 I22;
  Bound[Ai, 2, 2, s] = (2 d z[s])^3 VarGamma2[s]^3 Sqrt[I32 I34];
  , {s, {i, o}}];
```

Then we define $\overline{A}^{a,b}$ as in (4.5.16)-(4.5.20):

In[322]:=
```
Do[
  Do[
   Bound[Abar, a, 0, s] = Bound[Ai, a, 0, s];
   Bound[Abar, a, 1, s] = Bound[Ai, a, 1, s]/(2 d z[i]);
   , {a, {0, 1, 2}}];
  Bound[Abar, 0, 2, s] = (2 d z[s])^2 VarGamma2[s]^2 I22;
  Bound[Abar, 1, 2, s] = (2 d z[s]) VarGamma2[s]^2 I22;
  Bound[Abar, 2, 2, s] = VarGamma2[s]^2 I20;
  , {s, {i, o}}];
```

Then we define $B^{2,i,a,b}$ as in Table 4.21 and the comment after (4.5.20)

In[323]:=
```
Do[
 Do[
  Bound[B2i, a, 0, s] = 0;
  Bound[B2i, a, 1, s] = 0;
  , {a, {0, 1, 2}}];
 Bound[B2i, 0, 2, s] = Bound[ClosedRepBubble, 3, s] Bound[ClosedRepTriangle, 4, s] / (2 d z[s]) +
   Bound[ClosedRepTriangle, 4, s] Bound[OpenRepTriangle, 3, s];
 Bound[B2i, 1, 2, s] = Bound[ClosedRepBubble, 3, s] / (2 d z[s]) Bound[ClosedRepTriangle, 4, s] / (2 d z[s]) +
   Bound[ClosedRepTriangle, 4, s] Bound[OpenRepTriangle, 3, s] / (2 d z[s]);
 Bound[B2i, 2, 2, s] = Bound[ClosedRepBubble, 3, s] / (2 d z[s]) Bound[OpenRepTriangle, 3, s] +
   Bound[OpenRepBubble, 2, s] Bound[OpenRepTriangle, 4, s];
 , {s, {i, o}}];
```

Further, we define $\overline{B}^{2,i,a,b}$ as in Tables 4.22-4.23 and the comment after (4.5.20)

In[324]:=
```
Do[
 Do[
  Bound[Bbar2i, 0, a, s] = 0;
  , {a, {0, 1, 2}}];
 Bound[Bbar2i, 1, 0, s] = Bound[G3i, s] Bound[ClosedRepBubble, 4, s] +
   Bound[G2, s] Bound[ClosedRepTriangle, 4, s];
 Bound[Bbar2i, 1, 1, s] = Bound[G2, s] Bound[ClosedRepTriangle, 4, s] / (2 d z[s]);
 Bound[Bbar2i, 1, 2, s] = Bound[G3i, s] Bound[OpenRepBubble, 3, s] +
   Bound[G2, s] Bound[OpenRepTriangle, 4, s];
 Bound[Bbar2i, 2, 0, s] = Bound[ClosedRepBubble, 3, s] Bound[OpenRepBubble, 4, s] +
   Bound[ClosedRepBubble, 3, s] Bound[ClosedRepTriangle, 4, s] +
   Bound[ClosedRepTriangle, 4, s] Bound[OpenRepTriangle, 3, s];
 Bound[Bbar2i, 2, 1, s] = Bound[Bbar2i, 2, 0, s] / (2 d z[s]);
 Bound[Bbar2i, 2, 2, s] =
  Bound[G1, o] Bound[OpenRepBubble, 2, s] Bound[ClosedRepBubble, 3, s] +
   Bound[OpenRepTriangle, 3, s] Bound[ClosedRepBubble, 3, s] / (2 d z[s]) +
   Bound[G1, o] Bound[OpenRepBubble, 1, s] Bound[ClosedRepTriangle, 4, s];
 , {s, {i, o}}];
```

In[325]:=
```
Bound[OpenRepBubble, 1, o]
Bound[G1, o]
```

Out[325]= 0.0142055

Out[326]= 0.0137371

We define the bound on $P^{(N),i,b}$define in Table 4.24 as follows:

In[327]:=
```
Do[
  Bound[Piota, 0, s] = 2 d Bound[G3i, s] Bound[P, 0, s];
  Bound[Piota, 1, s] = 2 d Bound[G3i, s] Bound[P, 1, s] + 2 d (z[s] + Bound[G3i, s]);
  Bound[Piota, 2, s] = 2 d Bound[G3i, s] Bound[P, 2, s] + Bound[G3i, s] +
    Bound[ClosedRepBubble, 3, s] / z[s] +
    2 d (z[s] + Bound[G3i, s] + Bound[ClosedRepBubble, 3, s] / (2 d z[s]))
     Bound[ClosedRepBubble, 3, s];
  , {s, {i, o}}];
```

Definition of diagrams with weight

Now we first implement the bound in the diagrams $H^{1,a,b}$, $H^{2,i,a,b}$ and $H^{3,i,a,b}$ defined in (4.5.23)-(4.5.25):

In[328]:=
```
Do[
  Bound[H1, 0, 0, s] = Bound[WeightedClosedBubble, 0, s];
  Bound[H1, 1, 0, s] = Bound[WeightedClosedBubble, 0, s] / (2 d z[s]);
  Bound[H1, 0, 1, s] = Bound[WeightedClosedBubble, 0, s] / (2 d z[s]);
  Bound[H1, 1, 1, s] = Bound[WeightedClosedBubble, 2, s] / (2 d z[s])^2;
  Bound[H1, 2, 0, s] = Bound[WeightedOpenBubble, 0, s];
  Bound[H1, 0, 2, s] = Bound[WeightedOpenBubble, 0, s];

  Bound[H1, 2, 1, s] = Bound[WeightedOpenBubble, 1, s] / (2 d z[s]);
  Bound[H1, 1, 2, s] = Bound[WeightedOpenBubble, 1, s] / (2 d z[s]);
  Bound[H1, 2, 2, s] = Bound[WeightedOpenBubble, 0, s];

  Do[Do[
    Bound[H2, a, b, s] = Bound[H1, a, b, s];
    , {a, {0, 1, 2}}], {b, {0, 1, 2}}];
  (* for H3 we know that the unweighted path has at least length one*)
  Bound[H3, 0, 0, s] = Bound[WeightedClosedBubble, 0, s];
  Bound[H3, 1, 0, s] = Bound[WeightedClosedBubble, 2, s] / (2 d z[s]);
  Bound[H3, 0, 1, s] = Bound[WeightedClosedBubble, 2, s] / (2 d z[s]);
  Bound[H3, 1, 1, s] = Bound[WeightedClosedBubble, 3, s] / (2 d z[s])^2;
  (* to use the information that the unweighted path has at least length
   one we use Chauchy schwarz to obtain*)
  Bound[H3, 2, 0, s] = Bound[WeightedOpenBubble, 1, s];
  Bound[H3, 0, 2, s] = Bound[WeightedOpenBubble, 1, s];

  Bound[H3, 1, 2, s] = Bound[WeightedOpenBubble, 2, s] / (2 d z[o]);
  Bound[H3, 2, 1, s] = Bound[WeightedOpenBubble, 2, s] / (2 d z[o]);
  Bound[H3, 2, 2, s] = Bound[WeightedOpenBubble, 1, s];

  Clear[a, b, t];
  , {s, {i, o}}]
```

As explained in Section 4.5.6 we bound $C^{1,i,a,b}$ and $C^{2,i,a,b}$ in terms of other diagram. At this point we only implement the bound on $C^{3,i,a,b}$, see (4.5.90)-(4.5.94). For a=2,b=2 we implement the bound as describes in (4.5.97). For other a,b we improve this bound using three informations: 1.) By symmetrie can we bound the contribution $C^{3,i,a,b}$ in the same way as $C^{3,i,b,a}$. 2.) if a or b are 1 then we can use symmetrie to create an extra $\hat{D}(k)$. 3.) the complete square consists of at least four steps to improve the bounds. Using these three points to obtain the bounds:

```
In[329]:= Do[

  Bound[C3, 2, 2, s] = Bound[WeightedOpenLine, 0, s] (2 d z[s])^2 VarGamma2[s]^3
    I32 Bound[OpenRepSquare, 2, s];
  Bound[C3, 1, 2, s] = Bound[WeightedOpenLine, 0, s] (2 d z[s])^2 VarGamma2[s]^3
    I32 Bound[ClosedRepSquare, 3, s] / (2 d z[s]);
  Bound[C3, 2, 1, s] = Bound[C3, 1, 2, s];
  Bound[C3, 1, 1, s] = Bound[WeightedOpenLine, 0, s] Bound[OpenRepTriangle, 3, s]
    Bound[ClosedRepSquare, 3, s] / (2 d z[s]);
  Bound[C3, 0, 1, s] = Bound[WeightedOpenLine, 0, s] Bound[OpenRepTriangle, 2, s]
    Bound[ClosedRepSquare, 3, s] / (2 d z[s]);
  Bound[C3, 1, 0, s] = Bound[C3, 0, 1, s];
  Bound[C3, 0, 0, s] = Bound[WeightedOpenLine, 0, s] Bound[OpenRepTriangle, 2, s]
    Bound[ClosedRepSquare, 2, s];
  Bound[C3, 0, 2, s] = Bound[WeightedOpenLine, 0, s] (2 d z[s])^2 VarGamma2[s]^3
    I32 Bound[ClosedRepSquare, 2, s];
  Bound[C3, 2, 0, s] = Bound[C3, 0, 2, s];
  , {s, {i, o}}]
```

Definition of diagrams specific for initial step iota

As next we bound the term defined in (4.5.28) and bound in (4.5.100)-(4.5.102). We improve the bound stated of the first diagram of Figure 4.29, by noting that y=0 is not possible, so that the unweighted lines has always at least 2 steps.

```
In[330]:= Do[

  Bound[hi, part1, 0, s] = Bound[WeightedClosedBubble, 2, s];
  Bound[hi, part1, 1, s] = Bound[WeightedClosedBubble, 2, s] / (2 d z[s]);
  Bound[hi, part1, 2, s] = Bound[WeightedOpenBubble, 1, s];
  Do[
   Bound[hi, part2, b, s] = 2 d Bound[G3i, s] Bound[H2, 0, b, s] +
     4 d Bound[G3i, s]
      (Sum[Bound[P, a, s] Bound[H2, a, b, s], {a, 0, 2}] - Bound[H2, 0, b, s]) +
     4 d Bound[G3i, s] Sum[ Bound[H1, 0, a, s] Bound[Ai, a, b, s], {a, 0, 2}];
   Bound[hi, part3, b, s] = Bound[ClosedRepLoop, 4, s] / (2 d z[s]) Bound[H2, 1, b, s] +
     Bound[ClosedRepBubble, 3, s] / (2 d z[s]) Bound[H2, 2, b, s] +
     2 (Bound[ClosedRepLoop, 4, s] / (2 d z[s]) + Bound[ClosedRepBubble, 3, s] / (2 d z[s]))
      (Bound[Ai, 0, 0, s] Max[Bound[H2, 1, b, s], Bound[H2, 2, b, s]] +
        Bound[H1, 0, 0, s] Max[Bound[Ai, 1, b, s], Bound[Ai, 2, b, s]]);

   Bound[hi, b, s] = Sum[Bound[hi, t, b, s], {t, {part1, part2, part3}}]
   , {b, 0, 2}];
  , {s, {i, o}}]
```

Then we define the bound for $h^{i,\mathrm{II},b}$ defined in (4.5.29) and bounded in (4.5.99).

In[331]:=
```
Do[
 Do[
   Bound[hII, b, s] =
    Bound[hi, b, s] +
     2 Sum[Bound[hi, a, s] Bound[Ai, a, b, s] +
        Sum[Bound[Piota, a, s] Bound[Ai, a, c, s] Bound[H2, c, b, s], {c, 0, 2}],
       {a, 0, 2}]
   , {b, 0, 2}];
 , {s, {i, o}}]
```

Definition of vectors and matrices

Then we define the matricies we use the compute/state the bounds. First we define the matricies for which we have already computed the entries.

In[332]:=
```
Do[
 Vector[P, s] = Table[Bound[P, a, s], {a, {0, 1, 2}}];
 Vector[PNT, s] = Vector[P, s] - {1, 0, 0};
 Vector[Piota, s] = Table[Bound[Piota, a, s], {a, {0, 1, 2}}];

 Vector[h, s] = Table[Bound[H1, 0, b, s], {b, {0, 1, 2}}];
 Vector[hi, s] = Table[Bound[hi, b, s], {b, {0, 1, 2}}];
 Vector[hII, s] = Table[Bound[hII, b, s], {b, {0, 1, 2}}];

 Matrix[A, s] = Table[Bound[A, a, b, s], {a, {0, 1, 2}}, {b, {0, 1, 2}}];
 Matrix[Ai, s] = Table[Bound[Ai, a, b, s], {a, {0, 1, 2}}, {b, {0, 1, 2}}];
 Matrix[Abar, s] = Table[Bound[Abar, a, b, s], {a, {0, 1, 2}}, {b, {0, 1, 2}}];
 Matrix[B2i, s] = Table[Bound[B2i, a, b, s], {a, {0, 1, 2}}, {b, {0, 1, 2}}];
 Matrix[Bbar2i, s] = Table[Bound[Bbar2i, a, b, s], {a, {0, 1, 2}}, {b, {0, 1, 2}}];

 Matrix[C3, s] = Table[Bound[C3, a, b, s], {a, {0, 1, 2}}, {b, {0, 1, 2}}];
 Matrix[H1, s] = Table[Bound[H1, a, b, s], {a, {0, 1, 2}}, {b, {0, 1, 2}}];
 Matrix[H2, s] = Table[Bound[H2, a, b, s], {a, {0, 1, 2}}, {b, {0, 1, 2}}];
 Matrix[H3, s] = Table[Bound[H3, a, b, s], {a, {0, 1, 2}}, {b, {0, 1, 2}}];
 , {s, {i, o}}]
```

Then we define the bound for B and $\overline{B}$, see (4.5.21)-(4.5.22)

In[333]:=
```
Do[
 Matrix[B, s] = Matrix[Ai, s].Matrix[A, s] + Matrix[Ai, s] * Bound[P, 0, s] +
   Matrix[B2i, s];
 Matrix[Bbar, s] = Matrix[Ai, s].Matrix[A, s] + Matrix[Ai, s] + Matrix[Bbar2i, s];
 , {s, {i, o}}]
Clear[a, b]
```

Further, we define the bound on C^1 and C^2 as stated in (4.5.90) and (4.5.92)

In[335]:=
```
Do[
 Matrix[C1, s] =
  (2 Matrix[H2, s].Matrix[A, s] + 2 Matrix[Ai, s].Matrix[H1, s] + Matrix[H2, s]).
    Matrix[Ai, s] + Matrix[H2, s].Matrix[Bbar2i, s];
 Matrix[C2, s] =
  (2 Matrix[H3, s].Matrix[A, s] + 2 Matrix[Ai, s].Matrix[H1, s] + Matrix[H3, s]).
    Matrix[Ai, s] + 2 Matrix[C3, s] + Matrix[H3, s].Matrix[Bbar2i, s];
 , {s, {i, o}}]
Clear[a, b]
```

We compute the eigensystem of the maticies B and $\overline{B}$ to sum the bounds as explained in Section 5.3.

```
In[337]:= Do[
  EigensystemB[s] = Eigensystem[Transpose[Matrix[B, s]]];
  EigensystemBbar[s] = Eigensystem[Matrix[Bbar, s]];
  InverseProductB[s] = Inverse[Transpose[EigensystemB[s][[2]]]].Vector[P, s];
  InverseProductBForPIota[s] =
   Inverse[Transpose[EigensystemB[s][[2]]]].Vector[Piota, s];
  InverseProductBbar[s] = Inverse[Transpose[EigensystemBbar[s][[2]]]].Vector[P, s];

  Do[
   EigenVectorB[j, s] = EigensystemB[s][[2, j]] * InverseProductB[s][[j]];
   EigenVectorBbar[j, s] = EigensystemBbar[s][[2, j]] * InverseProductBbar[s][[j]];
   EigenVectorBForPIota[j, s] =
    EigensystemB[s][[2, j]] * InverseProductBForPIota[s][[j]];

   EigenValueB[j, s] = EigensystemB[s][[1, j]];
   EigenValueBbar[j, s] = EigensystemBbar[s][[1, j]];
   EigenValueBForPIota[j, s] = EigensystemB[s][[1, j]];
   , {j, 1, 3}];
  , {s, {i, o}}]
 Clear[a, b]
```

Bound on the coefficients

Bound for k=0

Then we implement the bounds stated in Lemma 4.5.1.-4.5.2 and Proposition 4.5.3. explixitely treat N=2,3 as they will also recieve special attentention when we compute the bound on Ξ(0)-Ξ(k)

```
In[339]:= Do[
  Bound[Xi, normal, 0, s] = Bound[P, 0, s] - 1;
  (* For the analysis we explicitly extract the contribution of δ0,x*)
  Bound[Xi, iota, 0, s] = Bound[G3i, s] Bound[P, 0, s];

  Bound[Xi, normal, 1, s] = Vector[P, s].Matrix[Abar, s].Vector[P, s];
  Bound[Xi, iota, 1, s] = 1/(2 d) Vector[Piota, s].Matrix[Abar, s].Vector[P, s];

  Bound[Xi, normal, 2, s] = Vector[P, s].Matrix[B, s].Matrix[Abar, s].Vector[P, s];
  Bound[Xi, iota, 2, s] =
   1/(2 d) Vector[Piota, s].Matrix[B, s].Matrix[Abar, s].Vector[P, s];

  Bound[Xi, normal, 3, s] = Vector[P, s].Matrix[B, s].Matrix[B, s].
    Matrix[Abar, s].Vector[P, s];
  Bound[Xi, iota, 3, s] =
   1/(2 d) Vector[Piota, s].Matrix[B, s].Matrix[B, s].Matrix[Abar, s].Vector[P, s];

  , {s, {i, o}}]
```

We compue the sum over all odd and even N=> 4

In[340]:=
```
Do[
 Bound[Xi, normal, EvenTail, s] =
  Sum[EigenValueB[j, s]^3/(1 - EigenValueB[j, s]^2) EigenVectorB[j, s].Matrix[Abar, s].Vector[P, s],
   {j, {1, 2, 3}}];
 Bound[Xi, normal, OddTail, s] =
  Sum[EigenValueB[j, s]^4/(1 - EigenValueB[j, s]^2) EigenVectorB[j, s].Matrix[Abar, s].Vector[P, s],
   {j, {1, 2, 3}}];
 Bound[Xi, iota, EvenTail, s] =
  1/(2 d)
   Sum[EigenValueBForPIota[j, s]^3/(1 - EigenValueBForPIota[j, s]^2)
     EigenVectorBForPIota[j, s].Matrix[Abar, s].Vector[P, s], {j, {1, 2, 3}}];
 Bound[Xi, iota,  OddTail, s] =
  1/(2 d) Sum[EigenValueBForPIota[j, s]^4/(1 -  EigenValueBForPIota[j, s]^2)
     EigenVectorBForPIota[j, s].Matrix[Abar, s].Vector[P, s], {j, {1, 2, 3}}];
 , {s, {i, o}}]
```

Then we compute the sum over all odd/even N

In[341]:=
```
Do[
 Do[
  Bound[Xi, a, Even, s] = Sum[Bound[Xi, a, t, s], {t, {0, 2, EvenTail}}];
  Bound[Xi, a, Odd, s] = Sum[Bound[Xi, a, t, s], {t, {1, 3, OddTail}}];
  Bound[Xi, a, Absolut, s] = Sum[Bound[Xi, a, t, s], {t, {Odd, Even}}];
  (*Print[Bound[Xi,a, Absolut, s]];*)
  , {a, {normal, iota}}]
 , {s, {i, o}}]
```

Bound for 0 - k

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

First we compute the terms for N=0,1,2,3 and extract therefore the contributions with trivial first and last triangle

In[342]:=

```
Do[
 Bound[Xi, normal, 0, Delta, 0, s] = Bound[WeightedClosedBubble, 0, s];
 Bound[Xi, normal, 1, Delta, 0, s] =
  Bound[H2, 0, 0, s] + 4 Sum[Bound[H3, 0, b, s] Bound[Ai, 0, b, s], {b, 0, 2}] +
   6 Vector[h, s].Matrix[Ai, s].Vector[PNT, s] +
   3 Vector[PNT, s].Matrix[H2, s].Vector[PNT, s];
 (*for this bound we use the spatial symmetrie of the diagrams,
 when one triangle it trivial to reduce the factor 8 stated in (4.5.47)
  to 4 (sinis terms cancel, so we can split the cosin without creating
    the factor 2)*)
 Bound[Xi, normal, 2, Delta, 0, s] =
  4 (Vector[h, s].Matrix[Bbar, s].Matrix[Ai, s].Vector[PNT, s] +
      Vector[PNT, s].Matrix[B, s].
       (Matrix[H3, s].Vector[PNT, s] + Matrix[Ai, s].Vector[h, s]) +
      Vector[PNT, s].Matrix[C1, s].Vector[PNT, s]) +
   6 (Vector[h, s].Matrix[Bbar, s].Matrix[Ai, s].{1, 0, 0} +
      Vector[PNT, s].Matrix[B, s].Matrix[H2, s].{1, 0, 0} +
      Vector[PNT, s].Matrix[C1, s].{1, 0, 0}) +
   2 ({1, 0, 0}.Matrix[B, s].Matrix[H2, s].{1, 0, 0} +
      {1, 0, 0}.Matrix[C1, s].{1, 0, 0});
 Bound[Xi, normal, 3, Delta, 0, s] =
  5 (Vector[h, s].Matrix[Bbar, s].Matrix[Bbar, s].Matrix[Ai, s].Vector[PNT, s] +
      Vector[PNT, s].Matrix[B, s].Matrix[B, s].
       (Matrix[H2, s].Vector[PNT, s] + Matrix[Ai, s].Vector[h, s]) +
      Vector[PNT, s].(Matrix[C1, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C2, s]).
       Vector[PNT, s]) +
   4
    ({1, 0, 0}.Matrix[B, s].Matrix[B, s].
       (Matrix[H2, s].Vector[PNT, s] + Matrix[Ai, s].Vector[h, s]) +
      {1, 0, 0}.(Matrix[C1, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C2, s]).
       Vector[PNT, s]) +
   4 (Vector[h, s].Matrix[Bbar, s].Matrix[Bbar, s].Matrix[Ai, s].{1, 0, 0} +
      Vector[PNT, s].Matrix[B, s].Matrix[B, s].(Matrix[H2, s].{1, 0, 0}) +
      Vector[PNT, s].(Matrix[C1, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C2, s]).
       {1, 0, 0}) +
   3 ({1, 0, 0}.Matrix[B, s].Matrix[B, s].Matrix[H2, s].{1, 0, 0} +
      {1, 0, 0}.(Matrix[C1, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C2, s]).
       {1, 0, 0});
 , {s, {i, o}}]
```

Then we compute the sum over the remaining N

In[343]:=

```
Do[
 Do[
  v[j] = EigenVectorB[j, s];
  vb[j] = EigenVectorBbar[j, s];
  e[j] = EigenValueB[j, s];
  eb[j] = EigenValueBbar[j, s];
  , {j, {1, 2, 3}}];
 Bound[Xi, normal, EvenTail, Delta, 0, s] =
  Vector[h, s].Matrix[Ai, s].Sum[vb[j] * eb[j]^3 (2/(1 - eb[j]^2)^2 + 4/(1 - eb[j]^2)),
```

```
{j, {1, 2, 3}}] + Sum[e[j]^3 (2/(1 - e[j]^2)^2 + 4/(1 - e[j]^2)) * v[j], {j, {1, 2, 3}}].
   (Matrix[H3, s].Vector[P, s] + Matrix[Ai, s].Vector[h, s]) +
  2 Sum[v[j] 1/(1 - e[j]^2), {j, {1, 2, 3}}].Matrix[C1, s].
   Sum[vb[j] 1/(1 - eb[j]^2)^2, {j, {1, 2, 3}}] +
  2 Sum[v[j] 1/(1 - e[j]^2)^2, {j, {1, 2, 3}}].Matrix[C1, s].
   Sum[vb[j] 1/(1 - eb[j]^2), {j, {1, 2, 3}}] - 4 Vector[P, s].Matrix[C1, s].Vector[P, s] +
  2 Sum[v[j] e[j]/(1 - e[j]^2), {j, {1, 2, 3}}].Matrix[C2, s].
   Sum[vb[j] eb[j] (1/(1 - eb[j]^2)^2 + 1/(1 - eb[j]^2)), {j, {1, 2, 3}}] +
  2 Sum[v[j] e[j]/(1 - e[j]^2)^2, {j, {1, 2, 3}}].Matrix[C1, s].
   Sum[vb[j] eb[j] 1/(1 - eb[j]^2), {j, {1, 2, 3}}];

Bound[Xi, normal, OddTail, Delta, 0, s] =
 Vector[h, s].Matrix[Ai, s].Sum[vb[j] * eb[j]^4 (2/(1 - eb[j]^2)^2 + 5/(1 - eb[j]^2)),
    {j, {1, 2, 3}}] + Sum[e[j]^4 (2/(1 - e[j]^2)^2 + 5/(1 - e[j]^2)) * v[j], {j, {1, 2, 3}}].
   (Matrix[H3, s].Vector[P, s] + Matrix[Ai, s].Vector[h, s]) +
  Sum[v[j] 2/(1 - e[j]^2)^2, {j, {1, 2, 3}}].
   (Matrix[C2, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C1, s]).
   Sum[vb[j] 1/(1 - eb[j]^2), {j, {1, 2, 3}}] +
  Sum[v[j] 1/(1 - e[j]^2), {j, {1, 2, 3}}].
   (Matrix[C2, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C1, s]).
   Sum[vb[j] (2/(1 - eb[j]^2)^2 + 1/(1 - eb[j]^2)), {j, {1, 2, 3}}] +
  5 Vector[P, s].(Matrix[C2, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C1, s]).
    Vector[P, s];

, {s, {i, o}}]
```

Bounds for $\sum_i \hat{\Xi}^i(0)-\hat{\Xi}^i(k)$

First we compute the terms for N=0,1,2,3 and therefore extract the contributions with trivial first and last triangle

In[344]:=
```
Do[
 Bound[Xi, iota, 0, Delta, ei, s] =
  2 d (2 d - 1) z[s] Bound[G3i, s]^2 +
   (2 d - 1) Bound[G3i, s] Bound[WeightedClosedBubble, 2, s];
 Bound[Xi, iota, 0, Delta, 0, s] =
  Bound[Xi, iota, 0, Delta, ei, s] + 2 d Bound[Xi, iota, 0, s];
 (* we can use symmetrie to cancel the sin terms an obtain this bound*)

 Bound[Xi, iota, 1, Delta, ei, s] =
  Bound[hi, 0, s] + Vector[hi, s].Vector[PNT, s] +
   Vector[Piota, s].Matrix[Ai, s].Vector[h, s];
 Bound[Xi, iota, 1, Delta, 0, s] =
  Bound[Xi, iota, 1, Delta, ei, s] + 2 d Bound[Xi, iota, 1, s];

 Bound[Xi, iota, 2, Delta, ei, s] =
  2 (Vector[hII, s].Matrix[Bbar, s].{1, 0, 0} +
      Vector[Piota, s].Matrix[B, s].Matrix[H3, s].{1, 0, 0}) +
   3 (Vector[hII, s].Matrix[Bbar, s].Vector[PNT, s] +
      Vector[Piota, s].Matrix[B, s].
       (Matrix[H3, s].Vector[PNT, s] + Matrix[Ai, s].Vector[h, s]));
 Bound[Xi, iota, 2, Delta, 0, s] =
  3 (Vector[hII, s].Matrix[Bbar, s].{1, 0, 0} +
      Vector[Piota, s].Matrix[B, s].Matrix[H3, s].{1, 0, 0}) +
   4 (Vector[hII, s].Matrix[Bbar, s].Vector[PNT, s] +
      Vector[Piota, s].Matrix[B, s].
       (Matrix[H3, s].Vector[PNT, s] + Matrix[Ai, s].Vector[h, s])) +
   3 Vector[Piota, s].Matrix[B, s].Matrix[Abar, s].{1, 0, 0} +
   4 Vector[Piota, s].Matrix[B, s].Matrix[Abar, s].Vector[PNT, s] ;

 Bound[Xi, iota, 3, Delta, ei, s] =
  3 (Vector[hII, s].Matrix[Bbar, s].Matrix[Bbar, s].{1, 0, 0} +
      Vector[Piota, s].Matrix[B, s].Matrix[B, s].Matrix[H3, s].{1, 0, 0}) +
   3 Vector[Piota, s].Matrix[B, s].Matrix[C2, s].{1, 0, 0} +
   4 (Vector[hII, s].Matrix[Bbar, s].Matrix[Bbar, s].Vector[PNT, s] +
      Vector[Piota, s].Matrix[B, s].Matrix[B, s].
       (Matrix[H3, s].Vector[PNT, s] + Matrix[Ai, s].Vector[h, s])) +
   4 Vector[Piota, s].Matrix[B, s].Matrix[C2, s].Vector[PNT, s];
 Bound[Xi, iota, 3, Delta, 0, s] =
  4 (Vector[hII, s].Matrix[Bbar, s].Matrix[Bbar, s].{1, 0, 0} +
      Vector[Piota, s].Matrix[B, s].Matrix[B, s].Matrix[H3, s].{1, 0, 0}) +
   4 Vector[Piota, s].Matrix[B, s].Matrix[C2, s].{1, 0, 0} +
   4 Vector[Piota, s].Matrix[B, s].Matrix[B, s].Matrix[Abar, s].{1, 0, 0} +
   5 (Vector[hII, s].Matrix[Bbar, s].Matrix[Bbar, s].Vector[PNT, s] +
      Vector[Piota, s].Matrix[B, s].Matrix[B, s].
       (Matrix[H3, s].Vector[PNT, s] + Matrix[Ai, s].Vector[h, s])) +
   5 Vector[Piota, s].Matrix[B, s].Matrix[C2, s].Vector[PNT, s] +
   5 Vector[Piota, s].Matrix[B, s].Matrix[B, s].Matrix[Abar, s].Vector[PNT, s];
 , {s, {i, o}}]
```

In[345]:=
```
Do[
 Do[
  v[j] = EigenVectorBForPIota[j, s];
  vb[j] = EigenVectorBbar[j, s];
  e[j] = EigenValueBForPIota[j, s];
```

```
 eb[j] = EigenValueBbar[j, s];
 , {j, {1, 2, 3}}];
Bound[Xi, iota, EvenTail, Delta, 0, s] =
 Vector[hII, s].Sum[vb[j] * eb[j]^3 (2/(1 - eb[j]^2)^2 + 3/(1 - eb[j]^2)), {j, {1, 2, 3}}] +
  Sum[e[j]^3 (2/(1 - e[j]^2)^2 + 3/(1 - e[j]^2)) * v[j], {j, {1, 2, 3}}].
   (Matrix[H3, s].Vector[P, s] + Matrix[Ai, s].Vector[h, s]) +
  Sum[v[j] (2 e[j])/(1 - e[j]^2)^2, {j, {1, 2, 3}}].
   (Matrix[C2, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C1, s]).
    Sum[vb[j] 1/(1 - eb[j]^2), {j, {1, 2, 3}}] +
  Sum[v[j] e[j]/(1 - e[j]^2)^2, {j, {1, 2, 3}}].
   (Matrix[C2, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C1, s]).
    Sum[vb[j] (2/(1 - eb[j]^2)^2 + 1/(1 - eb[j]^2)), {j, {1, 2, 3}}];
Bound[Xi, iota, EvenTail, Delta, ei, s] =
 Vector[hII, s].Sum[vb[j] * eb[j]^3 (2/(1 - eb[j]^2)^2 + 4/(1 - eb[j]^2)), {j, {1, 2, 3}}] +
  Sum[e[j]^3 (2/(1 - e[j]^2)^2 + 4/(1 - e[j]^2)) * v[j], {j, {1, 2, 3}}].
   (Matrix[H3, s].Vector[P, s] + Matrix[Ai, s].Vector[h, s]) +
  Sum[v[j] (2 e[j])/(1 - e[j]^2)^2, {j, {1, 2, 3}}].
   (Matrix[C2, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C1, s]).
    Sum[vb[j] 1/(1 - eb[j]^2), {j, {1, 2, 3}}] +
  Sum[v[j] e[j]/(1 - e[j]^2)^2, {j, {1, 2, 3}}].
   (Matrix[C2, s].Matrix[Bbar, s] + Matrix[B, s].Matrix[C1, s]).
    Sum[vb[j] (2/(1 - eb[j]^2)^2 + 2/(1 - eb[j]^2)), {j, {1, 2, 3}}] +
  Sum[e[j]^3 (2/(1 - e[j]^2)^2 + 4/(1 - e[j]^2)) * v[j], {j, {1, 2, 3}}].Matrix[Ai, s].
    Vector[P, s];

Bound[Xi, iota, OddTail, Delta, 0, s] =
 Vector[hII, s].Sum[vb[j] * eb[j]^4 (2/(1 - eb[j]^2)^2 + 4/(1 - eb[j]^2)), {j, {1, 2, 3}}] +
```

```
Sum[e[j]^4 (2/(1 - e[j]^2)^2 + 4/(1 - e[j]^2)) * v[j], {j, {1, 2, 3}}].
  (Matrix[H3, s].Vector[P, s] + Matrix[Ai, s].Vector[h, s]) +
 Sum[v[j] (2 e[j])/(1 - e[j]^2)^2, {j, {1, 2, 3}}].Matrix[C2, s].
  Sum[vb[j] 1/(1 - eb[j]^2), {j, {1, 2, 3}}] +
 Sum[v[j] e[j]/(1 - e[j]^2), {j, {1, 2, 3}}].Matrix[C2, s].
  Sum[vb[j] 2/(1 - eb[j]^2)^2, {j, {1, 2, 3}}] -
 4 Vector[Piota, s].Matrix[B, s].Matrix[C2, s].Vector[P, s] +
 Sum[v[j] (2 e[j]^2)/(1 - e[j]^2)^2, {j, {1, 2, 3}}].Matrix[C1, s].
  Sum[vb[j] 1/(1 - eb[j]^2), {j, {1, 2, 3}}] +
 Sum[v[j] (2 e[j]^2)/(1 - e[j]^2), {j, {1, 2, 3}}].Matrix[C1, s].
  Sum[vb[j] (2/(1 - eb[j]^2)^2 + 2/(1 - eb[j]^2)), {j, {1, 2, 3}}];
Bound[Xi, iota, OddTail, Delta, ei, s] =
 Vector[hII, s].Sum[vb[j] * eb[j]^4 (2/(1 - eb[j]^2)^2 + 5/(1 - eb[j]^2)), {j, {1, 2, 3}}] +
 Sum[e[j]^4 (2/(1 - e[j]^2)^2 + 5/(1 - e[j]^2)) * v[j], {j, {1, 2, 3}}].
  (Matrix[H3, s].Vector[P, s] + Matrix[Ai, s].Vector[h, s]) +
 Sum[v[j] (2 e[j])/(1 - e[j]^2)^2, {j, {1, 2, 3}}].Matrix[C2, s].
  Sum[vb[j] 1/(1 - eb[j]^2), {j, {1, 2, 3}}] +
 Sum[v[j] e[j]/(1 - e[j]^2), {j, {1, 2, 3}}].Matrix[C2, s].
  Sum[vb[j] 2/(1 - eb[j]^2)^2, {j, {1, 2, 3}}] +
 Sum[v[j] (2 e[j]^2)/(1 - e[j]^2)^2, {j, {1, 2, 3}}].Matrix[C1, s].
  Sum[vb[j] 1/(1 - eb[j]^2), {j, {1, 2, 3}}] +
 Sum[v[j] (2 e[j]^2)/(1 - e[j]^2), {j, {1, 2, 3}}].Matrix[C1, s].
```

```
Sum[vb[j] (2/(1 - eb[j]^2)^2 + 3/(1 - eb[j]^2)), {j, {1, 2, 3}}] +
 Sum[e[j]^3 (2/(1 - e[j]^2)^2 + 5/(1 - e[j]^2)) * v[j], {j, {1, 2, 3}}].Matrix[Ai, s].
  Vector[P, s];
, {s, {i, o}}]
```

For Sum over all N

```
In[346]:= Do[
  Do[
   Bound[Xi, a, Even, Delta, 0, s] =
    Sum[Bound[Xi, a, t, Delta, 0, s], {t, {0, 2, EvenTail}}];
   Bound[Xi, a, Odd, Delta, 0, s] =
    Sum[Bound[Xi, a, t, Delta, 0, s], {t, {1, 3, OddTail}}];
   Bound[Xi, a, Absolut, Delta, 0, s] =
    Sum[Bound[Xi, a, t, Delta, 0, s], {t, {Odd, Even}}];
   , {a, {normal, iota}}];

  Bound[Xi, iota, Even, Delta, ei, s] =
   Sum[Bound[Xi, iota, t, Delta, ei, s], {t, {0, 2, EvenTail}}];
  Bound[Xi, iota, Odd, Delta, ei, s] =
   Sum[Bound[Xi, iota, t, Delta, ei, s], {t, {1, 3, OddTail}}];
  Bound[Xi, iota, Absolut, Delta, ei, s] =
   Sum[Bound[Xi, iota, t, Delta, ei, s], {t, {Odd, Even}}];
  , {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:

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

$$\begin{gathered}\sum_x F(x)[1-\cos(k\,x)] \geq K_{\text{Lower}}\left[1-\hat{D}(k)\right]\\ \sum_x \left|F(x)\right|[1-\cos(k\,x)] \leq K_{\Delta \text{F}}\left[1-\hat{D}(k)\right]\\ \sum_x \left|\Phi_z(x)\right|[1-\cos(k\,x)] \leq K_{\Delta\Phi}\left[1-\hat{D}(k)\right]\end{gathered} \tag{2}$$

Bound on absolute value K_F and K_Φ

In[347]:=
```
Do[
 alpha[s] = z[s] rho[s];
 baralpha[s] = z[s];
 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

As 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[348]:=
```
Do[
 Bound[DifferencefF, Part1, Lower, s] =
  Min[(2 d baralpha[i])/(1 - alpha[i]^2), (2 d baralpha[s])/(1 - alpha[s]^2)]
   (rho[Lower, s] - Bound[Xi, normal, Odd, Delta, 0, s] - Bound[Xi, normal, Odd, s] -
     alpha[o] Bound[Xi, normal, Even, Delta, 0, s]);
 Bound[DifferencefF, Part1, Absolut, s] =
  Max[(2 d baralpha[i])/(1 - alpha[i]^2), (2 d baralpha[s])/(1 - alpha[s]^2)]
   (rho[s] + (1 + alpha[s]) Bound[Xi, normal, Absolut, Delta, 0, s] +
     Bound[Xi, normal, Absolut, s]);
 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[349]:= 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[350]:= 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 $P(\gamma,\Gamma,z)$ is satisfied, see Definition 3.3.2.

In[351]:=
```
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[352]:=
```
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[353]:=
```
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[355]:=
```
Succes[overall]
```

Out[355]=
```
True
```

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

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

In[356]:=
```
(2 d - 2)/(2 d - 1) Gamma2(* ≥ G_z(k) [1-D̂(k)] *)
Max[Bound[KDeltaFLower, o], 1]
(* Nominator in (4.3.13) *)
```

Out[356]= 1.01009

Out[357]= 1.03546

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

In[358]:=
```
1/(2 d - 1) Gamma1
```

Out[358]= 0.0135368

The improvement of bounds

In[359]:=
```
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[361]=

Result for Dimension 38

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	1.015⋱26	1.023⋱74	1.2	1.2	1.2	1.2	1.015⋱26	1.023⋱74	1.2	1.2	1.2	1.2
Boun⋱ds	1.015⋱13	1.001⋱43	0.406⋱327	0.872⋱91	0.559⋱433	0.648⋱425	1.015⋱26	1.023⋱74	1.199⋱97	1.199⋱97	1.199⋱99	1.2
check												
Requ⋱ire⋱d	1.015⋱13	1.001⋱43	0.338⋱606	0.727⋱425	0.034⋱599⋱1	4.931⋱01	1.015⋱26	1.023⋱74	0.225⋱594	0.699⋱383	0.074⋱215⋱7	9.125⋱51

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 in 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.

```
(*d=38;
{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}*)
```

Print out of the computed bounds in the coefficients

In[371]:=
```
Do[
 MethodeFourTable[s] = {{Quantity, ΞZero, ΞOne, ΞTwo, ΞThree, ΞEvenTail, ΞOddTail},
    {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 Ξ̂ι], 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[ Ξι (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[Ξι (1 - cos (k (x - eι)))] , 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 zI 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
```

Out[374]=

Bound on coefficients at z_I in Dimension 38

Quantity	Ξ^{Zero}	Ξ^{One}	Ξ^{Two}	Ξ^{Three}	$\Xi^{EvenTail}$	$\Xi^{OddTail}$
Bound for $\hat{\Xi}$	0.00041992	0.00122413	0.0000147095	1.06125×10^{-7}	8.49607×10^{-10}	6.63532×10^{-12}
Bound for $\hat{\Xi}^{\iota}$	0.000190877	0.0000228947	2.36926×10^{-7}	1.75471×10^{-9}	1.3958×10^{-11}	1.0917×10^{-13}
$(1-\cos kx)\,\hat{\Xi}$	0.00174721	0.00434495	0.170197	0.00161496	0.0000379002	0.00248981
$(1-\cos kx)\,\Xi^{\iota}$	0.0145317	0.0128212	0.0374024	0.000440648	0.0000150137	0.000021449
$\Xi^{\iota}\,(1-\cos k\,(x-e_{\iota}))$	0.0000250022	0.0110812	0.0249137	0.00033018	0.0000180227	0.000162039

Out[375]=

Bound on coefficients in Dimension 38

Quantity	Ξ^{Zero}	Ξ^{One}	Ξ^{Two}	Ξ^{Three}	$\Xi^{EvenTail}$	$\Xi^{OddTail}$
Bound for $\hat{\Xi}$	0.00044864	0.0013487	0.0000180453	1.44312×10^{-7}	1.28054×10^{-9}	1.10869×10^{-11}
Bound for $\hat{\Xi}^{\iota}$	0.00020033	0.0000254733	2.92344×10^{-7}	2.40101×10^{-9}	2.11678×10^{-11}	1.8354×10^{-13}
$(1-\cos kx)\,\hat{\Xi}$	0.00274677	0.0107585	0.494344	0.00521691	0.000135368	0.00808369
$(1-\cos kx)\,\Xi^{\iota}$	0.0152663	0.032988	0.110892	0.00145826	0.0000543616	0.0000772693
$\Xi^{\iota}\,(1-\cos k\,(x-e_{\iota}))$	0.000041251	0.031052	0.0739334	0.0010935	0.0000652432	0.000547875

Out[376]=

Bound on the constants of Proposition 3.3.1 in Dimension 38

Quantity	KF	KPhi	DELTAFLower	DELTAFAbsolut	DELTAPhi
Bound for	1.0307	1.0007	1.03546	1.56233	0.522886

Out[377]=

Bound on the constants of Proposition 3.3.1 in Dimension 38

Quantity	DELTA⋱FLow⋱er	2	3	DELTA⋱FAbs⋱olut	2	3	DELTA⋱Phi	2	3
Bound for	0.967105	−0.0013⋱473⋱9	-7.92019×10^{-7}	1.56049	0.00183⋱722	7.92019×10^{-7}	0.522885	7.06203×10^{-7}	1.64358×10^{-10}