NoBLE for lattice trees

Implementation of the computer-assisted proof of the NoBLE by Robert Fitzner and Remco van der Hofstad.

Robert Fitzner
Eindhoven University of Technology

Abstract

This document is the second part of the computer-assisted proof of the non-backtracking lace expansion (NoBLE). The NoBLE was created by Remco van der Hofstad and the author to prove mean-field behavior for self-avoiding walks, lattice trees (LT), lattice animals (LA) and percolation. In this file, the computations for lattice trees are performed. The technique is explained in "Generalized approach to the non-backtracking lace expansion"(I), and the bounds that we implement here are derived in "NoBLE lattice animals and trees"(II). All references in this file are to one of these two papers, which we refer to by (I) and (II).

This file is accompanied by other notebook -SRW.nb- and -General.nb-. In the SRW file, a number of simple random walk quantities are computed. In -General.nb-, general bounds, derived in (I) are implemented. Before doing computations with this file the user should first open these files, choose a dimension and once execute all lines of the file.

Then, the user is expected to choose constants Γ_i and c_μ, $c_{x,s}$ in this file. After choosing these quantities the user should select the menu item Evaluate-> Evaluate Notebook. In a table at the end of this document the results of the computations are shown: it can be seen whether the bootstrap, with the given parameters, and therefore the analysis, was successful.

The computation of the -SRW- and -General- file are independent of the values Γ_i and c, so that we need to execute these files only once, when you start the Mathematica Kernel or when you want to change the dimension under consideration.

15.03.2019

Input

Here the user should choose the constants Γ_i, c_μ and $c_{n,l,S}$ for which we try to perform the bootstrap, explained in (I), using this program/document. These values are introduced in Section 2.1, see (2.1)-(2.3) and the comments below it. For the set S used to define f_3, we are restricted to the sets {0} and $\mathbb{Z}^d \backslash \{0\}$. For the following values the bootstrap succeeds in dimension 16:

```
In[6619]:= (*The parameter choices that work in d=16 for LTs *)
Gamma1 = 1.10225007; (*Assumed bound on zg_z*)
Gamma2 = 1.335307; (* (2d-1)/(2d-2) sup_k [1-D̂(k)] Ĝ_z(k) *)
Gamma3 = 1;
cmu = 1.03578941;
GammaThree[1, 0] = 0.2913684; (*Assumed bound on sup_{x≠0} Σ_y |y|_2^2 G(y) G(x-y) *)
GammaThree[1, 1] = 7786720/100000000; (*Assumed bound on sup_{x≠0} Σ_y |y|_2^2 G(y) (D*G) (x-y) *)
GammaThree[1, 2] = 3527550/100000000;
(*Assumed bound on sup_{x≠0} Σ_y |y|_2^2 G(y) (D*D*G) (x-y) *)
GammaThree[1, 3] = 1521196/100000000; (*Assumed bound on sup_{x≠0} Σ_y |y|_2^2 G(y) (D^{*3}*G) (x-y) *)
GammaThree[2, 0] = 0.4864781; (*Assumed bound on sup_{x≠0} Σ_y |y|_2^2 G(y) (G*G) (x-y) *)
GammaThree[2, 1] = 21793600/100000000; (*Assumed bound on sup_{x≠0} Σ_y |y|_2^2 G(y) (D*G*G) (x-y) *)
GammaThree[2, 2] = 0.09178173;
(*Assumed bound on sup_{x≠0} Σ_y |y|_2^2 G(y) (D*D*G*G) (x-y) *)
GammaThree[2, 3] = 0.0422438;
(*Assumed bound on sup_{x≠0} Σ_y |y|_2^2 G(y) (D^{*3}*G*G) (x-y) *)
GammaThreeClosed[1, 6] = 363481/100000000; (*Assumed bound on Σ_y |y|_2^2 G(y) (D^{*6}*G) (-y) *)
GammaThreeClosed[2, 6] = 0.01167464;
(*Assumed bound on Σ_y |y|_2^2 G(y) (D^{*6}*G*G) (-y) *)
```

Let us explain how we found the above values: The bound $f_i(z_I)$ at z_I, which is independent of the improvement-of-bound arguments, gives a good orientation. Starting from these values we used a semi-automated procedure to find appropriate values for these constants. The basic idea is based on a fixed-point argument and works as follow: Starting from some reasonable values, we conclude bounds. Then, we use these bounds as new assumed bounds and re-do all computations again starting from theses values. We repeat this procedure until either values of the concluded bounds start to diverge and have failed or until we actually concluded bounds that were smaller than the bounds assumed in this run. This is implemented near the end of this document.

Sandbox

To improve readability of the code, comments are added. These comments are found between the signs (* comments *). We prepare the following input field in case you want to try other values of the constants. For this, you can just remove the commenting (* / *) in the following input field:

```
In[6633]:= (*Gamma1=1.08177;
Gamma2=1.227;
Gamma3=1;
cmu=1.03105;
GammaThree[1,0]=0.12288;
GammaThree[1,1]=0.0312;
GammaThree[1,2]=0.014311;
GammaThree[1,3]=0.0055;
GammaThree[2,0]=0.1904;
GammaThree[2,1]=0.081;
GammaThree[2,2]=0.0318;
GammaThree[2,3]=0.01342;
GammaThreeClosed[1,6]=0.001183;
GammaThreeClosed[2,6]=0.00323;*)
```

Bound on the two-point function and on repulsive diagrams

In this section we use the bootstrap assumption $f_i(z) < \Gamma_i$ and the computation of -SRW.nb- to conclude bounds on the two-point function and the basic diagrams

Bound on z and g_z

We define the constants for two setting s: we use s=i for bound on $z = z_I$ and s=o for bound on $z \in (z_I, z_c)$: For $z = z_I$ we use

$$z_I = \frac{1}{(2d-1)e} \qquad g_{z_I} \le e + \frac{e-1}{2d-1} \qquad g^i_{z_I} \le 1 + (g_{z_I} - 1)\frac{2d-1}{2d} \le e$$
$$\overline{G}_z(x) \le g^i_{z_I} B_{z_I g^i_{z_I}}(x) \le g^i_{z_I} \frac{2d-2}{2d-1} C(x) \qquad \tilde{G}_z(0) = 1 \text{ and otherwise } \tilde{G}_z(x) \le B_{z_I g^i_{z_I}}(x) \le \frac{2d-2}{2d-1} C(x). \tag{1}$$

and implement this:

```
In[6634]:= z[i] = 1/((2 d - 1) Exp[1]);
gj[i] = Exp[1];
g[i] = Exp[1] + (Exp[1] - 1)/(2 d - 1);
gz[i] = g[i] × z[i];
gjz[i] = gj[i] × z[i];
VarGamma2[i] = 1;
VarGamma2b[i] = VarGamma2[i] gz[i]/gjz[i];
```

For z $\in (z_I, z_c)$, we assume that

$$2 d z g^\iota_z < 2 d g_z z < \Gamma_1, \qquad g_z < e \Gamma_1 \text{ for all } z \ge \frac{1}{(2d-1)e}, \qquad g^\iota_z < 1 + (g_z - 1)\frac{2d-1}{2d}. \tag{2}$$

```
In[6641]:= gz[o] = Gamma1/(2 d - 1);
gjz[o] = 1/cmu Gamma1/(2 d - 1);
g[o] = Exp[1] Gamma1;
gj[o] = Min[Exp[1] cmu Gamma1, 1 + (g[o] - 1) (2 d - 1)/(2 d)];
VarGamma2[o] = Gamma2;
VarGamma2b[o] = VarGamma2[o] gz[o]/gjz[o];
```

Lower bound on g_z

For some bounds we require a bound on $z < z_c$. For this we create a lower bound on g_z, which we obtain by using that we only consider $z > z_I = \frac{1}{(2d-1)e}$ and the following basic bounds:

$$g_z = \Sigma_{n=0} t_n(0) z^n \ge \Sigma^3_{n=0} (\text{lower bound on } t_n(0)) z^n_I$$
$$(n=0) \implies t_n(0) z^n_I = 1$$
$$(n=1) \implies t_1(0) z^1_I = 2d * \frac{1}{(2d-1)e} > \frac{1}{e} \tag{3}$$
$$(n=1) \implies t_2(0) z^2_I = 3d(2d-1) * \frac{1}{(2d-1)^2 e^2} = \frac{3}{2}\frac{d}{d-\frac{1}{2}}\frac{1}{e^2} > \frac{1.5}{e^2}$$

In[6647]:=
```
gLower = 1 + 1/Exp[1] + 1.5/Exp[2];
"Which is " <> TextString[NumberForm[N[gLower/g[o] * 100], 5]] <>
 "% of the upper bounds."
z[o] = gz[o]/gLower;
```

Out[6648]= Which is 52.429% of the upper bounds.

We could use even more terms then just these three and improve our bounds further to up to 70% of the upper bound. We found however that this will not reduce the dimension for which we obtain our main results.

Bounds on two-point functions

Here, we compute bounds on the two-point functions for some x, as explained in Section 5.3.2. of (I). These bounds are obtained by extracting short, explicit contributions and by bounding the longer contributions using f_2, see Section 5.3 of (I). To bound the short explicit contributions we use the values $c_j(x)$ =nrSAW[j,d,x] provided in the model-independent SRW-integral notebook, these are provided for $j \leq ComputedSteps$. The value of *ComputedSteps* is given in the -SRW.nb- file. The extraction of short contributions creates a better bound, than just applying f_2. The reason is that f_1 gives a sharper bound than f_2.

$$\tilde{G}_{m,z}(x) \leq \sum_{j=m}^{M-1} (g_z^{\iota} z)^j c_j(x) + \tilde{G}_{M,z}(x). \tag{4}$$

For lattice trees we defined in total three different two-point functions $G_{m,z}, \overline{G}_{m,z}, \tilde{G}_{m,z}$. The function $G_{m,z}, \overline{G}_{m,z}$ can be bounded directly using f_2. For $\tilde{G}_{m,z}$ we use an additional step

$$\tilde{G}_{m,z}(x) \leq 2\,d\,z\left(D * \overline{G}_{m-1,z}\right)(x) = 2\,d\,z\,g_z\,(D * G_{m-1,z})\,(x) \leq 2\,d\,g_z\left(2\,d\,z\,g_z^{\iota}\right)^{m-1}(D^m * G_z)\,(x) \leq 2\,d\,g_z\left(2\,d\,z\,g_z^{\iota}\right)^{m-1}\overline{\Gamma}_2\,K_{1,m}(x). \tag{5}$$

for even M. We use the following variables, that are required to be even:

In[6650]:=
```
(*Number of steps we will extract,
we have to choose M=Rsteps to apply the bound.*)
If[EvenQ[ComputedSteps],
 Explicit = ComputedSteps;
 (*the length of the polygon that we extract. It is supposed to be even*)
 RSteps = ComputedSteps + 2;,
 (*The length of the shortest loop in the remainder terms*)
 Explicit = ComputedSteps - 1;
 RSteps = ComputedSteps + 1;]
```

Bound on $G_{m,z}(e_1)$

We use $G_{m,z}(e_1) = (D * G_{m,z})\,(0)$ to compute

In[6651]:=
```
Do[
 Do[
  Bound[tG, {1}, m, s] = Sum[gjz[s]^j nrSAW[j, d, {1}], {j, m, Explicit}] +
     2 d gz[s] (2 d gjz[s])^(RSteps-2) VarGamma2[s] × Ivalue[1, RSteps, {0}];
  , {m, 1, 5}]
 , {s, {i, o}}]
```

Bound on $G_{m,z}(e_1 + e_2)$

We use $G_{m,z}(e_1 + e_2) \leq \frac{d}{d-1}\left(D^2 * G_{m,z}\right)(0)$ to compute

```
In[6652]:= Do[
  Do[
   Bound[tG, {2}, 2 m, s] = Sum[gjz[s]^j nrSAW[j, d, {2}], {j, 2 m, Explicit}] +
      d/(d - 1) 2 d gz[s] (2 d gjz[s])^(RSteps-1) VarGamma2[s] × Ivalue[1, RSteps + 2, {0}];
   , {m, 1, ComputedSteps / 2}]
  , {s, {i, o}}]
```

We are also interested in bounds on the constrained connection that does not use the lattice point e_ι. We obtain a bound on this removing some of the explicit paths using the point e_1.

```
In[6653]:= Do[
  Bound[tG, ikNotUsingi, 6, s] =
   gjz[s]^6 (nrSAW[6, d, {2}] - 9 (2 d - 3)^2 - 2 × (2 d - 4) - 2 × (2 d - 3)) +
    Bound[tG, {2}, 8, o];
  Bound[tG, ikNotUsingi, 4, s] =
   gjz[s]^4 (nrSAW[4, d, {2}] - 2 (d - 2)) + Bound[tG, ikNotUsingi, 6, s];
  Bound[tG, ikNotUsingi, 2, s] = gjz[s]^2 + Bound[tG, ikNotUsingi, 4, s];
  , {s, {i, o}}]
```

Bound on $G_{m,z}(2\,e_1)$

Now, we compute bounds on $G_{m,z}(2\,e_1)$ and $G^{1}_{m,z}(2\,e_1)$. We obtain the latter by subtracting a number of explicit contributions.

```
In[6654]:= Do[
  Do[
   Bound[tG, {0, 1}, 2 m, s] = Sum[gjz[s]^j nrSAW[j, d, {0, 1}], {j, 2 m, Explicit}] +
      2 d gz[s] (2 d gjz[s])^(RSteps-1) VarGamma2[s] × Ivalue[1, RSteps, {0}];
   , {m, 1, ComputedSteps / 2}];
  Bound[tG, twoiNotusingi, 6, s] =
   gjz[s]^6 (nrSAW[6, d, {0, 1}] - 36 (d - 2)^2) + Bound[tG, {0, 1}, 8, o];
  Bound[tG, twoiNotusingi, 4, s] = gjz[s]^4 2 (d - 1) + Bound[tG, twoiNotusingi, 6, s];
  Bound[tG, twoiNotusingi, 2, s] = Bound[tG, twoiNotusingi, 4, s];
  , {s, {i, o}}]
```

Bound on $\sup_{x\neq 0} G_{m,p}(x)$

To compute the supremum of the two-point function we use that $c_n(x)$=nrSAW[n,d,x] for $n \leq 10$ has its maximal value at $x = e_1$ or $x = e_1 + e_2$.

In[6655]:=
```
maxpossible = Max[Explicit, MaxNumberOfSteps / 2];
Do[
 Bound[tG, max, maxpossible, s] =
  (2 d gjz[s])^maxpossible VarGamma2[s] × K[1, maxpossible, {1}];
 Do[
  m = maxpossible - 2 t;
  Bound[tG, max, m, s] =
   Max[nrSAW[m, d, {2}] gjz[s]^m, nrSAW[m + 1, d, {1}] gjz[s]^(m+1)] +
    Bound[tG, max, m + 2, s];
  , {t, 1, maxpossible / 2 - 1}];
 Bound[tG, max, 5, s] = Max[Bound[tG, max, 6, s], Bound[tG, {1}, 5, s]];
 Bound[tG, max, 3, s] = Max[Bound[tG, max, 4, s], Bound[tG, {1}, 3, s]];
 Bound[tG, max, 1, s] = Max[Bound[tG, max, 2, s], Bound[tG, {1}, 1, s]];
 Bound[tG, max, 0, s] = Bound[tG, max, 1, s];
 , {s, {i, o}}]
Clear[t, m, explicit, maxpossible]
```

Bound on $\frac{g_z}{g_z^\iota}$ and $\frac{g_z^\iota}{g_z}$

We require an upper on $\frac{\overline{\mu}_z}{\mu_z}$, which we conclude from $\frac{\overline{\mu}_z}{\mu_z} = \frac{z g_z}{z g_z^\iota} = \frac{g_z}{g_z^\iota} = 1 + \frac{g_z - g_z^\iota}{g_z^\iota} = 1 + \frac{1}{g_z^\iota}\, \overline{G}_z(e_1) \leq\ 1 + \tilde{G}_z(e_1)$. With Bound[tG,{1},3,s] we have an upper bound on $\tilde{G}_{3,z}(e_1)$. For the lower bound we use $\overline{G}_z(e_1) > 0$, so that $\frac{\overline{\mu}_z}{\mu_z} > 1$ and $\frac{\mu_z}{\overline{\mu}_z} < 1$.

In[6658]:=
```
Do[
 mubOverMu[s] = 1 + Bound[tG, {1}, 1, s];
 muOverMub[s] = 1;
 , {s, {i, o}}]
```

Repulsive diagrams

Now, we bound repulsive diagrams, defined in Definition 4.7 of (II) as described in Section 5.3.2 of (I). In (i) we see that the resulting bounds do no depend on the individual lengths of the pieces $m_1, m_2, \ldots$, but only of the sum of the known lengths $\sum_i m_i$. So we refer to each diagram by the number of minimal steps $\sum_i m_i$ and the number of two-point function without fixed length ($G_{m,z}(x)$ instead of $G_{\underline{m},z}(x)$). Further, we assume that only two-point functions $\tilde{G}_{m,z}$ are involved.

Closed diagrams

In[6659]:=
```
Do[
 Bound[Loop, 4, s] = 2 d gjz[s] × Bound[tG, {1}, 3, s];
 Do[
  Bound[Bubble, m, s] =
    Sum[(j + 1 - m) 2 d nrSAW[j - 1, d, {1}] gjz[s]^j, {j, m, Explicit}] +
     (RSteps - m) (2 d gjz[s])^RSteps VarGamma2b[s] × Ivalue[1, RSteps, {0}] +
     (2 d gjz[s])^RSteps VarGamma2b[s]^2 Ivalue[2, RSteps, {0}]; (* (5.40) of I*)
  , {m, 2, 8}];
 Do[
  Bound[Triangle, m, s] =
    Sum[((j + 1 - m) (j + 2 - m))/2 2 d nrSAW[j - 1, d, {1}] gjz[s]^j, {j, m, Explicit}] +

     ((RSteps - m) (RSteps - 1 - m))/2 (2 d gjz[s])^RSteps VarGamma2b[s] ×
      Ivalue[1, RSteps, {0}] + (RSteps + 1 - m) (2 d gjz[s])^RSteps VarGamma2b[s]^2
      Ivalue[2, RSteps, {0}] + (2 d gjz[s])^RSteps VarGamma2b[s]^3
      Ivalue[3, RSteps , {0}]; (* (5.41) of I*)
  , {m, 1, 5}];
 Do[
  Bound[Square, m, s] =
    Sum[((j + 1 - m) (j + 2 - m) (j + 3 - m))/6 2 d nrSAW[j - 1, d, {1}] gjz[s]^j,
      {j, m, Explicit}] + 1/6 (RSteps + 1 - m) (RSteps + 2 - m) (RSteps + 3 - m)
      (2 d gjz[s])^RSteps VarGamma2b[s] × Ivalue[1, RSteps, {0}] +
     ((RSteps - m) (RSteps - 1 - m))/2 (2 d gjz[s])^RSteps VarGamma2b[s]^2
      Ivalue[2, RSteps, {0}] + (RSteps - m) (2 d gjz[s])^RSteps VarGamma2b[s]^3
      Ivalue[3, RSteps, {0}] + gz[s]/gjz[s] (2 d gjz[s])^RSteps VarGamma2b[s]^4
      Ivalue[4, RSteps, {0}]; (* (5.42) of I*)
  , {m, 2, 5}];
 , {s, {i, o}}]
```

Open repulsive diagrams

We can bound open repulsive diagrams, in the same way as the closed diagrams. Parallel to this we also produce the bounds without extracting contributions, as it is a priori not clear which bound is better. We use the monotonicity of the SRW-integrals, see Lemma 5.1 of (I): $\sup_{x\neq 0} K_{n,l}(x) = K_{n,l}(e_1)$, to conclude that the supremums of the open diagrams is at the neighbor of the origin.

In[6660]:=
```
Do[
 Do[
  Bound[OpenBubblePre, m, s] =
   Min[gz[s]/gjz[s] (2 d gjz[s])^m VarGamma2[s]^2 K[2, m, {1}],
    Max[Sum[(j + 1 - m) gjz[s]^j nrSAW[j, d, {2}], {j, m, ComputedSteps}],
      Sum[(j + 1 - m) gjz[s]^j nrSAW[j, d, {1}], {j, m, ComputedSteps}]]
     + (ComputedSteps + 1 - m) (2 d gjz[s])^(ComputedSteps+1) VarGamma2b[s] ×
      K[1, ComputedSteps + 1, {1}] +
     (2 d gjz[s])^ComputedSteps+1 VarGamma2b[s]^2 K[2, ComputedSteps + 1, {1}]];
  Bound[OpenTrianglePre, m, s] =
   Min[(2 d gjz[s])^m VarGamma2b[s]^3 K[3, m, {1}],
    Max[Sum[((j + 1 - m) (j + 2 - m))/2 gjz[s]^j nrSAW[j, d, {2}], {j, m, ComputedSteps}],
      Sum[((j + 1 - m) (j + 2 - m))/2 gjz[s]^j nrSAW[j, d, {1}], {j, m, ComputedSteps}]] +
     1/2 (ComputedSteps + 1 - m) (ComputedSteps + 1 - 1 - m) (2 d gjz[s])^(ComputedSteps+1)
      VarGamma2b[s] × K[1, ComputedSteps + 1, {1}] +
     (ComputedSteps + 1 - m) (2 d gjz[s])^(ComputedSteps+1) VarGamma2b[s]^2
      K[2, ComputedSteps + 1, {1}] +
     (2 d gjz[s])^ComputedSteps+1 VarGamma2b[s]^3 K[3, ComputedSteps + 1, {1}]];
  Bound[OpenSquarePre, m, s] =
   Min[(2 d gjz[s])^m VarGamma2b[s]^4 K[4, m, {1}],
    Max[Sum[1/6 (j + 1 - m) (j + 2 - m) (j + 3 - m) gjz[s]^j nrSAW[j, d, {2}],
       {j, m, ComputedSteps}],
      Sum[1/6 (j + 1 - m) (j + 2 - m) (j + 3 - m) gjz[s]^j nrSAW[j, d, {1}],
       {j, m, ComputedSteps}]] +
     1/6 × (10 + 1 - m) × (10 + 2 - m) × (10 + 3 - m) (2 d gjz[s])^(ComputedSteps+1)
      VarGamma2b[s] × K[1, ComputedSteps + 1, {1}] +
     ((10 - m) × (10 - 1 - m))/2 (2 d gjz[s])^(ComputedSteps+1) VarGamma2b[s]^2
      K[2, ComputedSteps + 1, {1}] +
     (10 - m) (2 d gjz[s])^(ComputedSteps+1) VarGamma2b[s]^3 K[3, ComputedSteps + 1, {1}] +
     (2 d gjz[s])^(ComputedSteps+1) VarGamma2b[s]^4 K[4, ComputedSteps + 1, {1}]];
  , {m, 0, 6}];
 , {s, {i, o}}]
```

When trying to find an optimal bound we saw that the following idea improved the bounds slightly, allowing us to prove the bounds for an additional dimension. Let us show the idea in an example: a repulsive triangle with some length restrictions:

$$T_{1,0,2}(x) \leq T_{\underline{1},0,2}(x) + T_{2,0,2}(x) \leq T_{\underline{1},0,2}(x) + T_{\underline{2},0,2}(x) + T_{\underline{3},0,2}(x) + T_{4,0,2}(x)$$

In[6661]:=
```
Do[
  m = 6;
  Bound[OpenBubble, m, s] = Bound[OpenBubblePre, m, s];
  Bound[OpenTriangle, m, s] = Bound[OpenTrianglePre, m, s];
  Bound[OpenSquare, m, s] = Bound[OpenSquarePre, m, s];
  Clear[m];
  , {s, {i, o}}];
```

Using this initial step we iteratively consider smaller and smaller diagrams and in each step use the best bound possible:

In[6662]:=
```
Do[
  Do[
    m = 5 - it;
    Bound[OpenBubble, m, s] = Min[Bound[OpenBubblePre, m, s],
      Bound[tG, max, m, s] + Bound[OpenBubble, m + 1, s]];
    Bound[OpenTriangle, m, s] = Min[Bound[OpenTrianglePre, m, s],
      Bound[OpenBubble, m, s] + Bound[OpenTriangle, m + 1, s]];
    Bound[OpenSquare, m, s] = Min[Bound[OpenSquarePre, m, s],
      Bound[OpenTriangle, m, s] + Bound[OpenSquare, m + 1, s]];
    , {it, 0, 5}];
  , {s, {i, o}}];
```

Weighted Diagrams

Here we define the bounds on the weighted diagrams. As explained in Section 5.3.3 of (I) we use for $z = z_I$ a bound that is independent of our analysis. These bounds have been implemented in the accompanying notebook. To the notation: As for the unweighted diagrams, we refer to the diagrams by their number of two-point functions and number of the fixed steps on the unweighted lines.

In[6663]:=
```
Bound[WeightedBubble, 6, i] = (2 d gjz[i])^6 BoundFThreeInital[d, 1, 6, 1, {{0}}] ;
Bound[WeightedTriangle, 6, i] = (2 d gjz[i])^6 BoundFThreeInital[d, 2, 6, 1, {{0}}] ;

Do[
 Bound[WeightedOpenBubble, t, i] =
  (2 d gjz[i])^t BoundFThreeInital[d, 1, t, 1, {{1}, {2}, {0, 1}}];
 Bound[WeightedOpenTriangle, t, i] =
  (2 d gjz[i])^t BoundFThreeInital[d, 2, t, 1, {{1}, {2}, {0, 1}}];
 , {t, 0, 3}]
```

For $z \in (z_I, z_c)$ we use the bootstrap function f_3 to obtain the bonds

In[6666]:=
```
Bound[WeightedBubble, 6, o] = (2 d gjz[o])^6 GammaThreeClosed[1, 6];
Bound[WeightedTriangle, 6, o] = (2 d gjz[o])^6 GammaThreeClosed[2, 6];

Do[
 Bound[WeightedOpenBubble, t, o] = (2 d gjz[o])^t GammaThree[1, t];
 Bound[WeightedOpenTriangle, t, o] = (2 d gjz[o])^t GammaThree[2, t];
 , {t, 0, 3}]
```

As explained in Section 5.3.3. we drastically improve the bound on the closed, weighted, repulsive diagram by extracting explicit contributions, by using its repulsiveness and

$$\frac{1}{g_z}\sum_x \|x\|_2^2 \sum_{A,\, x\epsilon A} z^A \;\leq\; \frac{1}{g_z}\sum_x \|x\|_2^2\, \overline{G}_z(x)\, 2\, d\, z \left(D * \tilde{G}\right)(x) \leq \sum_x \|x\|_2^2\, G_z(x)\, 2\, d\, z \left(D * \tilde{G}\right)(x)$$

$$\frac{1}{g_z}\sum_x \|x\|_2^2 \sum_{A,\, x\epsilon A} 1_{d(0,x)>n}\, z^A \;\leq\; \sum_x \|x\|_2^2\, G_z(x)\, \frac{\left(2\, d\, z\, g_z^{\iota}\right)^n}{g_z^{\iota}} \left(D^{*n} * \tilde{G}\right)(x)$$

We bound the unweighted connections $\left(G^{*n} * D^{*l}\right)(x)$ using $K_{n,l}$ (x), defined in (3.36) of (I). We compute $K_{n,l}(x)$ for $l \leq rem$, see below, and some points close to the origin. For x for which we have not computed $K_{n,l}(x)$ we use a monotonicity argument to bound $K_{n,l}(x)$, in our case K (2 e_1) > K (x) for all relevant x. This monotonicity is implied in Lemma 5.1 of (I).

```
In[6669]:= Do[
  explicit = Min[MaxNumberOfSteps / 2 - 2, ComputedSteps];
  rem = explicit + 1;
  LongContributions[point_] = (2 d gz[s]) (2 d gjz[s])^(rem-1) K[1, rem, point];
  (*  and The points x for which we have not computed K(x). We bound
    its value by K(2e1)>K(x).*)
  point5Remainder = {{1, 1}, {0, 0, 1}, {0, 0, 0, 0, 1}, {1, 0, 0, 1}, {0, 1, 1},
    {2, 0, 1}, {1, 2}, {3, 1}};
  point4Remainder = {{0, 1}, {1, 0, 1}, {0, 0, 0, 1}, {0, 2}, {2, 1}};

  Bound[WeightedBubble, 5, s] =
   Bound[WeightedBubble, 6, s] +
    1
    ──────
    gLower
     (gjz[s]^5 nrSAW[5, d, {1}] *
        (Sum[ nrSAW[r, d, {1}] gjz[s]^r, {r, 1, explicit}] + LongContributions[{1}]) +
      3 gjz[s]^5 nrSAW[5, d, {3}] *
        (Sum[ nrSAW[r, d, {3}] gjz[s]^r, {r, 1, explicit}] + LongContributions[{2}]) +
      5 gjz[s]^5 nrSAW[5, d, {5}] *
        (Sum[ nrSAW[r, d, {5}] gjz[s]^r, {r, 1, explicit}] + LongContributions[{2}]) +
      25 gjz[s]^5
       Sum[nrSAW[5, d, v] * (Sum[ nrSAW[r, d, v] gjz[s]^r, {r, 1, explicit}] +
           LongContributions[{0, 1}]), {v, point5Remainder}]);

  Bound[WeightedBubble, 4, s] =
   Bound[WeightedBubble, 5, s] +
    1
    ──────
    gLower
     (2 gjz[s]^4 nrSAW[4, d, {2}] *
        (Sum[ nrSAW[r, d, {2}] gjz[s]^r, {r, 1, explicit}] + LongContributions[{2}]) +
      4 gjz[s]^4 nrSAW[4, d, {4}] *
        (Sum[ nrSAW[r, d, {4}] gjz[s]^r, {r, 1, explicit}] +
          LongContributions[{2}]) +
      10 Sum[gjz[s]^4 nrSAW[4, d, v] *
          (Sum[ nrSAW[r, d, v] gjz[s]^r, {r, 1, explicit}] +
           LongContributions[{0, 1}]), {v, point4Remainder}]);

  Bound[WeightedBubble, 3, s] =
```

```
    Bound[WeightedBubble, 4, s] +
      1
    ------
    gLower
     (gjz[s]^3 nrSAW[3, d, {1}] *
        (Sum[ nrSAW[r, d, {1}] gjz[s]^r, {r, 1, explicit}] + LongContributions[{1}]) +
       5 gjz[s]^3 nrSAW[3, d, {1, 1}] *
        (Sum[ nrSAW[r, d, {1, 1}] gjz[s]^r, {r, 1, explicit}] +
          LongContributions[{0, 1}]) +
       9 gjz[s]^3 nrSAW[3, d, {0, 0, 1}] *
        (Sum[ nrSAW[r, d, {0, 0, 1}] gjz[s]^r, {r, 1, explicit}] +
          LongContributions[{0, 1}]) +
       3 gjz[s]^3 nrSAW[3, d, {3}] *
        (Sum[ nrSAW[r, d, {3}] gjz[s]^r, {r, 1, explicit}] +
          LongContributions[{2}]));

   Bound[WeightedBubble, 2, s] =
    Bound[WeightedBubble, 3, s] +
      1
    ------ (2 d * 4 * gjz[s]^2 Bound[tG, twoiNotusingi, 4, s] +
    gLower
       2 * 2 * gjz[s]^2 2 d (2 d - 2) Bound[tG, ikNotUsingi, 2, s]);
   Bound[WeightedBubble, 1, s] =
    Bound[WeightedBubble, 2, s] + 2 d z[s] × Bound[tG, {1}, 3, s];
   Bound[WeightedBubble, 0, s] = Bound[WeightedBubble, 1, s];

   Bound[WeightedTriangle, 5, s] =
    Bound[WeightedBubble, 5, s] + Bound[WeightedTriangle, 6, s];
   Bound[WeightedTriangle, 4, s] =
    Bound[WeightedBubble, 4, s] + Bound[WeightedTriangle, 5, s];
   Bound[WeightedTriangle, 3, s] =
    Bound[WeightedBubble, 3, s] + Bound[WeightedTriangle, 4, s];
   Bound[WeightedTriangle, 2, s] =
    Bound[WeightedBubble, 2, s] + Bound[WeightedTriangle, 3, s];
   Bound[WeightedTriangle, 1, s] =
    Bound[WeightedBubble, 1, s] + Bound[WeightedTriangle, 2, s];
   Bound[WeightedTriangle, 0, s] =
    Bound[WeightedBubble, 0, s] + Bound[WeightedTriangle, 1, s];
   (*remove the auxiliary variables from the memory*)
   Clear[point4Remainder, point5Remainder, LongContributions];
   s
   , {s, {i, o}}]
```

Building Blocks

Blocks without weight

In the following we implement the bound on the coefficient $A^{m,l}$ defined in Appendix C.1

```
In[6670]:= Do[
    Bound[A, 0, 0, s] = Bound[Loop, 4, s] + 2 Bound[Bubble, 3, s] +
```

```
  Bound[Triangle, 3, s];
(*Table 2*)
Bound[A, 0, 1, s] = Bound[Loop, 4, s] + 2 Bound[Bubble, 3, s] +
  Bound[Triangle, 4, s];
(*Table 3*)
Bound[A, 0, 2, s] = Bound[Bubble, 3, s] + 2 Bound[Triangle, 4, s] +
  Bound[Square, 5, s];
(*Table 4*)
Bound[A, 0, -1, s] = Bound[Loop, 4, s] (*d0,x=1,w=y*) + Bound[Bubble, 4, s]
  (*d0,x=1,0≠w≠y*) + Bound[Triangle, 3, s] (*d0,x>2*); (*Table 5*)
Bound[A, 0, -2, s] = Bound[Triangle, 4, s] (*d0,x=1→w≠0*) +
  Bound[Triangle, 4, s] (*d0,x≥2,w=0*) + Bound[Square, 5, s] (*d0,x≥2,w≠0*);
(*Table 6*)
```

```
Bound[A, -1, 0, s] = Bound[Bubble, 3, s]/(2 d gjz[s]) + Bound[Triangle, 3, s]/(2 d gjz[s]);
Bound[A, -1, -1, s] = Bound[Triangle, 3, s]/(2 d gjz[s]);
Bound[A, -1, -2, s] = Bound[Square, 4, s]/(2 d gjz[s]);
Bound[A, -1, 1, s] = Bound[Triangle, 3, s]/(2 d gjz[s]);
Bound[A, -1, 2, s] = Bound[Square, 4, s]/(2 d gjz[s]);

Bound[A, -2, 0, s] = Bound[OpenBubble, 2, s] + Bound[OpenTriangle, 2, s];
Bound[A, -2, 1, s] = Bound[OpenTriangle, 2, s];
Bound[A, -2, -1, s] = Bound[OpenTriangle, 2, s];
Bound[A, -2, 2, s] = Bound[OpenSquare, 3, s];
Bound[A, -2, -2, s] = Bound[OpenSquare, 3, s];
Do[Do[
  Bound[A, a, b, s] = Bound[A, -a, b, s];
  , {a, {1, 2}}], {b, {-2, -1, 0, 1, 2}}];

Do[
 Bound[Abar, b, 0, s] = Bound[A, b, 0, s];
 Bound[Abar, 0, b, s] = Bound[Abar, b, 0, s]
 , {b, -2, 2}];
Do[
 Bound[Abar, b, 1, s] = Bound[Abar, 0, 1, s]/(2 d gjz[s]);
 Bound[Abar, b, -1, s] = Bound[Abar, 0, -1, s]/(2 d gjz[s]);
 , {b, {-1, 1}}];
```

```
Do[Do[
  Bound[Abar, a, b, s] = (2 d gjz[s])^1 VarGamma2[s]^3 K[3, 1, {0}];
  , {a, {-2, 2}}], {b, {-2, 2}}];

Do[Do[
  Bound[Abar, a, b, s] = (2 d gjz[s])^1 VarGamma2[s]^3 K[3, 2, {1}];
  Bound[Abar, b, a, s] = (2 d gjz[s])^1 VarGamma2[s]^3 K[3, 2, {1}];
  , {a, {-1, 1}}], {b, {-2, 2}}];
, {s, {i, o}}];
```

Blocks with weight

Here we implement the element stated in Appendix C.2.

```
In[6671]:= Do[
  Bound[C, 0, 0, s] = 2 Bound[WeightedBubble, 2, s] + Bound[WeightedTriangle, 2, s];
  Bound[C, 0, -1, s] = Bound[WeightedBubble, 2, s] / (2 d gjz[s]) + Bound[WeightedTriangle, 3, s] / (2 d gjz[s]);
  Bound[C, -1, 0, s] = Bound[C, 0, -1, s];
  Bound[C, 0, 1, s] = Bound[C, 0, -1, s];
  Bound[C, 1, 0, s] = (*d0,x=1,v=w≠ x*)
   (4 * gjz[s] × Bound[tG, twoiNotusingi, 4, s] +
      2 * gjz[s] (2 d - 2) Bound[tG, ikNotUsingi, 2, s]) + (*d0,x=1,v≠w≠ x,
    split weight*) 2 Bound[WeightedBubble, 3, s] / (2 d gjz[s]) + (*d0,x≥ 2,split weight*)
    2 Bound[WeightedTriangle, 3, s] / (2 d gjz[s]);
  Bound[C, -1, -1, s] = (*v=w=y*)
   Max[4 * Bound[tG, twoiNotusingi, 4, s], 2 * Bound[tG, ikNotUsingi, 2, s]] +
    (*v≠w=y or v=w≠y*) 2 Bound[WeightedBubble, 3, s] / (2 d gjz[s])^2 + (*v≠w≠y*)
    Bound[WeightedTriangle, 4, s] / (2 d gjz[s])^2;
  Bound[C, -1, 1, s] = Bound[C, -1, -1, s];
  Bound[C, 1, -1, s] = 2 Bound[WeightedBubble, 3, s] / (2 d gjz[s])^2 +
    4 Bound[WeightedTriangle, 4, s] / (2 d gjz[s])^2;
  Bound[C, 1, 1, s] = Bound[C, 1, -1, s];
  (*the bad bounds*)
  Bound[C, 0, 2, s] = Bound[WeightedOpenTriangle, 0, s];
  Bound[C, 0, -2, s] = Bound[WeightedOpenTriangle, 0, s];
  Do[
   Bound[C, -2, t, s] = Bound[WeightedOpenTriangle, 0, s];
```

```
Bound[C, 2, t, s] = 2 Bound[WeightedOpenTriangle, 0, s];
, {t, {-2, 0, 2}}];

Bound[C, -2, 1, s] = Bound[WeightedOpenTriangle, 1, s] / (2 d gjz[s]);
Bound[C, -2, -1, s] = Bound[WeightedOpenTriangle, 1, s] / (2 d gjz[s]);
Bound[C, -1, 2, s] = Bound[WeightedOpenTriangle, 1, s] / (2 d gjz[s]);
Bound[C, -1, -2, s] = Bound[WeightedOpenTriangle, 1, s] / (2 d gjz[s]);


Bound[C, 2, 1, s] = 2 Bound[WeightedOpenTriangle, 1, s] / (2 d gjz[s]);
Bound[C, 2, -1, s] = 2 Bound[WeightedOpenTriangle, 1, s] / (2 d gjz[s]);
Bound[C, 1, 2, s] = 2 Bound[WeightedOpenTriangle, 1, s] / (2 d gjz[s]);
Bound[C, 1, -2, s] = 2 Bound[WeightedOpenTriangle, 1, s] / (2 d gjz[s]);
, {s, {i, o}}]
```

Initial Blocks

Here we define the initial and terminal part of the diagrams as described in Appendix C.3. For lattice trees this is simply

```
In[6672]:= Do[
  Do[
    Bound[P1, a, s] = Bound[A, 0, a, s];
    Bound[DeltaStart, a, s] = Bound[C, 0, a, s]
    , {a, {-2, -1, 0, 1, 2}}];
  , {s, {i, o}}];
```

Initial Iota Block without weight

Here we define the bound of the initial part of the coefficients $\Xi^{(N),\iota}$ and $\Pi^{(N),\iota,\kappa}$, as given in Appendix C.4.

```
In[6673]:= Do[
  Do[
   Bound[B, 1, a, s] = Bound[tG, {1}, 1, s] × Bound[A, 0, a, s];
   , {a, {-2, -1, 0, 1, 2}}];
  Bound[B, 2, 0, s] = (1/(2 d)) (Bound[tG, {1}, 1, s]/gjz[s])
    (Bound[Bubble, 3, s] + Bound[Triangle, 3, s]);
  Bound[B, 2, -1, s] = (Bound[tG, {1}, 1, s]/gjz[s]) (Bound[Triangle, 3, s]/(2 d));
  Bound[B, 2, -2, s] = (Bound[tG, {1}, 1, s]/gjz[s]) (Bound[Square, 4, s]/(2 d));
  Bound[B, 2, 1, s] = Bound[B, 2, -1, s];
  Bound[B, 2, 2, s] = Bound[B, 2, -2, s];
  Bound[B, 3, 0, s] =
   (Bound[Bubble, 3, s]/(2 d gjz[s])) (Bound[OpenBubble, 2, s] + Bound[OpenTriangle, 2, s]);
  Bound[B, 3, -1, s] = (Bound[Bubble, 3, s]/(2 d gjz[s])) Bound[OpenTriangle, 2, s];
  Bound[B, 3, -2, s] = (Bound[Bubble, 3, s]/(2 d gjz[s])) Bound[OpenSquare, 3, s];
  Bound[B, 3, 1, s] = Bound[B, 3, -1, s];
  Bound[B, 3, 2, s] = Bound[B, 3, -2, s];
  Bound[B, 4, 0, s] =
   (1/(2 d)) (Bound[Loop, 4, s] + 2 Bound[Bubble, 3, s] + Bound[Triangle, 3, s]);
  Bound[B, 4, 1, s] =
   (1/(2 d)) (Bound[Loop, 4, s] + 2 Bound[Bubble, 3, s] + Bound[Triangle, 4, s]);
  Bound[B, 4, 2, s] =
   (1/(2 d)) (Bound[Bubble, 3, s] + 2 Bound[Triangle, 4, s] + Bound[Square, 5, s]);
  Bound[B, 4, -1, s] =
   (1/(2 d)) (Bound[Loop, 4, s] + Bound[Bubble, 4, s] + Bound[Triangle, 3, s]);
  Bound[B, 4, -2, s] = (1/(2 d)) (2 Bound[Triangle, 4, s] + Bound[Square, 5, s]);

  Do[
   Bound[P1Iota, a, s] = Sum[Bound[B, c, a, s], {c, 1, 4}];
   , {a, {-2, -1, 0, 1, 2}}];
  , {s, {i, o}}];
```

In[6674]:=
```
Labeled[
  Grid[{Join[{"Part"}, Table[t, {t, 1, 4}]],
    Join[{"Abs."}, Table[NumberForm[Bound[B, t, 1, o], 3], {t, 1, 4}]],
    Join[{"% of Total"}, Table[NumberForm[100 * Bound[B, t, 2, o] / Bound[P1Iota, 2, o], 3],
      {t, 1, 4}]]}, Alignment → {Center}, Frame → True,
   Dividers → {{2 → True, -1 → True}, {2 → True}}, ItemStyle → {1 → Bold, 1 → Bold},
   Background → {{None}, {GrayLevel[0.9]}, {None}}],
  Style["Contribution to P1 ", Bold], Top] // Text
```

Out[6674]=

Contribution to P1

Part	**1**	**2**	**3**	**4**
Abs.	0.000574	0.000332	0.000058	0.000501
% of Total	38.3	24.	4.25	33.5

Initial weight Iota Block

Here we implement the bounds given in Appendix C.4.2. We begin the initial pieces in which $b = 0$

In[6675]:=
```
Do[Do[
   Bound[D, 1, 0, a, s] = Bound[tG, {1}, 1, s] × Bound[C, 0, a, s];
   Bound[D, 1, ei, a, s] =
    2 Bound[tG, {1}, 1, s] (Bound[C, 0, a, s] + gj[s] × Bound[A, a, 0, s]);
   Bound[D, 2, 0, a, s] = 1/(2 d) Bound[C, 0, a, s];
   Bound[D, 3, 0, a, s] = Bound[tG, {1}, 3, s] × Bound[C, -1, a, s] +
     Bound[Bubble, 4, s]/(2 d gjz[s]) Bound[C, -2, a, s];
   Bound[D, 3, ei, a, s] =
    2 Bound[D, 3, 0, a, s] +
     2 gj[s] × (Bound[tG, max, 2, s] × Bound[A, a, -1, s] +
        2 Bound[tG, max, 1, s] × Bound[A, a, -2, s]);
   Bound[D, 4, 0, a, s] = 1/(2 d) Bound[C, 0, a, s];
   , {a, {-2, -1, 0, 1, 2}}];
  Bound[D, 1, ei, 0, s] =
   Bound[tG, {1}, 1, s] (Bound[C, 0, a, s] + Bound[C, a, 0, s]);
  Bound[D, 2, 0, 0, s] = (gjz[s] + Bound[tG, {1}, 1, s]) × Bound[tG, {1}, 3, s]
    (*x=e1*) + 1/(2 d) Bound[tG, {1}, 1, s]/gjz[s]
     (2 Bound[WeightedBubble, 2, s] (*w=e1≠x or w=x≠e1*) +
       Bound[WeightedTriangle, 3, s]) (*e1≠w≠x or w=x≠e1*);
  (*For M=1 && a=0 we can use symmetry to remove the factor 2,
  created when splitting the weight ||x-eι||₂²*)
  Bound[D, 1, ei, 0, s] = Bound[tG, {1}, 1, s] × Bound[C, 0, 0, s] +
    Bound[tG, {1}, 1, s] × Bound[A, 0, 0, s];
  Bound[D, 1, ei, -1, s] = 2 Bound[tG, {1}, 1, s] × Bound[C, 0, -1, s] +
```

```
    2 Bound[tG, {1}, 1, s]
     (Bound[Loop, 4, s]/(2 d gjz[s]) + Bound[Bubble, 4, s]/(2 d gjz[s]) + Bound[Triangle, 3, s]/(2 d gjz[s]));
  Bound[D, 1, ei, -2, s] = 2 Bound[tG, {1}, 1, s] × Bound[C, 0, -2, s] +
    2 Bound[tG, {1}, 1, s] × (Bound[OpenBubble, 2, s] + Bound[OpenTriangle, 2, s]);
  Bound[D, 1, ei, 1, s] = Bound[D, 1, ei, -1, s];
  Bound[D, 1, ei, 2, s] = Bound[D, 1, ei, -2, s];
  (* TODO rework up to here.*)

  Bound[D, 2, ei, 0, s] = 1/(2 d) Bound[WeightedTriangle, 0, s];
  Bound[D, 2, ei, -1, s] = 1/(2 d) Bound[WeightedTriangle, 1, s]/(2 d gjz[s]);
  Bound[D, 2, ei, -2, s] = 1/(2 d) Bound[WeightedOpenTriangle, 0, s];
  Bound[D, 2, ei, 1, s] = Bound[D, 2, ei, -1, s];
  Bound[D, 2, ei, 2, s] = Bound[D, 2, ei, -2, s];

  Bound[D, 4, ei, 0, s] = 1/(2 d) Bound[WeightedTriangle, 1, s];
  Bound[D, 4, ei, -1, s] = 1/(2 d) Bound[WeightedTriangle, 2, s]/(2 d gjz[s]);
  Bound[D, 4, ei, -2, s] = 1/(2 d) Bound[WeightedOpenTriangle, 1, s];
  Bound[D, 4, ei, 1, s] = Bound[D, 4, ei, -1, s];
  Bound[D, 4, ei, 2, s] = Bound[D, 4, ei, -2, s];
  , {s, {i, o}}];
```

These are all contributions for lattice trees. We sum them

```
In[6676]:= Do[
  Do[
    Bound[D, 0, a, s] = Sum[Bound[D, c, 0, a, s], {c, 1, 4}];
    Bound[D, ei, a, s] = Sum[Bound[D, c, ei, a, s], {c, 1, 4}];
    , {a, {-2, -1, 0, 1, 2}}];
  , {s, {i, o}}];
```

In[6677]:=
```
Labeled[
  Grid[{Join[{"Part"}, Table[t, {t, 1, 4}]],
    Join[{"Abs."}, Table[NumberForm[Bound[D, t, 0, 2, o], 3], {t, 1, 4}]],
    Join[{"% of Total"}, Table[NumberForm[100 * Bound[D, t, 0, 2, o] / Bound[D, 0, 2, o], 3],
      {t, 1, 4}]]}, Alignment → {Center}, Frame → True,
   Dividers → {{2 → True, -1 → True}, {2 → True}}, ItemStyle → {1 → Bold, 1 → Bold},
   Background → {{None}, {GrayLevel[0.9]}, {None}}],
  Style["Contribution to D^{I,2} ", Bold], Top] // Text

Labeled[
  Grid[{Join[{"Part"}, Table[t, {t, 1, 4}]],
    Join[{"Abs."}, Table[NumberForm[Bound[D, t, ei, 2, o], 3], {t, 1, 4}]],
    Join[{"% of Total"}, Table[NumberForm[100 * Bound[D, t, ei, 2, o] / Bound[D, ei, 2, o], 3],
      {t, 1, 4}]]}, Alignment → {Center}, Frame → True,
   Dividers → {{2 → True, -1 → True}, {2 → True}}, ItemStyle → {1 → Bold, 1 → Bold},
   Background → {{None}, {GrayLevel[0.9]}, {None}}],
  Style["Contribution to D^{II,2} ", Bold], Top] // Text
```

Contribution to $D^{I,2}$

Out[6677]=

Part	**1**	**2**	**3**	**4**
Abs.	0.0174	0.0152	0.00147	0.0152
% of Total	35.3	30.8	2.98	30.8

Contribution to $D^{II,2}$

Out[6678]=

Part	**1**	**2**	**3**	**4**
Abs.	0.0363	0.0152	0.0169	0.00748
% of Total	47.9	20.	22.2	9.86

Definition of the vectors and matrices

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

index	interpretation
-2	backbone is on bottom, d (u, v) ≥ 2.
-1	backbone is on bottom, d (u, v) = 1.
0	u = v
1	backbone is on top, d (u, v) = 1.
2	backbone in on the top, d (u, v) ≥ 2.

In[6679]:=
```
Do[
 Matrix[A, s] = Table[Bound[A, a, b, s], {a, -2, 2}, {b, -2, 2}];
 Matrix[Abar, s] = Table[Bound[Abar, a, b, s], {a, -2, 2}, {b, -2, 2}];
 Matrix[C, s] = Table[Bound[C, a, b, s], {a, -2, 2}, {b, -2, 2}];

 Vector[P1, s] = Table[Bound[P1, a, s], {a, -2, 2}];
 Vector[DeltaStart, s] = Table[Bound[DeltaStart, a, s], {a, -2, 2}];
 Vector[EndOpen, s] = Table[Bound[A, b, 0, s], {b, -2, 2}];
 Vector[EndClosed, s] = Table[Bound[A, 0, b, s], {b, -2, 2}];
 Vector[DeltaEnd, s] = Table[Bound[C, b, 0, s], {b, -2, 2}];

 Vector[P1Iota, s] = Table[Bound[P1Iota, a, s], {a, -2, 2}];
 Vector[D, 0, s] = Table[Bound[D, 0, a, s], {a, -2, 2}];
 Vector[D, ei, s] = Table[Bound[D, ei, a, s], {a, -2, 2}];
 , {s, {i, o}}]
```

To compute the sum over matrices we compute a representation of the opening and closing vectors using eigenvalues of the matrices A. If the matrix is not invertible, such a representation (using real values only) does not need to exist. We bypass this problem by using a symmetric matrix that dominates A, see ASym below, and use this in our bound for $N \geq 4$.

In[6680]:=
```
Do[
 Matrix[ASym, s] = Table[Max[Bound[A, a, b, s], Bound[A, b, a, s]],
   {a, -2, 2}, {b, -2, 2}];

 EigenA[s] = Eigensystem[Transpose[Matrix[ASym, s]]];
 InverseProduct[left, s] = Inverse[Transpose[EigenA[s][[2]]]].Vector[P1, s];
 InverseProduct[iota, s] = Inverse[Transpose[EigenA[s][[2]]]].Vector[P1Iota, s];
 InverseProduct[right, s] =
  Inverse[Transpose[EigenA[s][[2]]]].Vector[EndClosed, s];
 Do[
  EigenVector[left, j, s] = EigenA[s][[2, j]] * InverseProduct[left, s][[j]];
  EigenVector[iota, j, s] = EigenA[s][[2, j]] * InverseProduct[iota, s][[j]];
  EigenVector[right, j, s] = EigenA[s][[2, j]] * InverseProduct[right, s][[j]];
  EigenValue[j, s] = EigenA[s][[1, j]];
  , {j, 1, 5}]

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

Bounds on the coefficients

Bounds for N=0

Here, we implement the bounds stated in Lemma 5.1.

In[6681]:=
```
Do[
 Bound[Xi, 0, s] = 0; Bound[Xi, 0, Delta, s] = 0;
 Bound[Xi, R, 0, s] = 0; Bound[Xi, R, 0, Delta, s] = 0;
 Bound[Psi, RII, 0, s] = 0; Bound[Psi, RII, 0, Delta, s] = 0;
 Bound[Psi, RI, 0, s] = 0; Bound[Psi, RI, 0, Delta, s] = 0;
 , {s, {i, o}}]
```

Next, we implement the bounds on $\Xi^{(0),\iota}$ stated in Lemma 5.2:

```
In[6682]:= Do[Bound[XiIota, 0, s] = muOverMub[s] × Bound[tG, {1}, 1, s];
    Bound[XiIota, RI, 0, s] = 0;
    Bound[XiIota, RII, 0, s] = 0;
    Bound[XiIota, 0, Delta, 0, s] = 0;
    Bound[XiIota, 0, Delta, ei, s] = muOverMub[s] × Bound[tG, {1}, 1, s](*x=0*);
    Bound[XiIota, alphaI, 0, Atei, s] = 0;
    Bound[XiIota, alphaII, 0, AtZero, s] = muOverMub[s] × Bound[tG, {1}, 1, s];
    Bound[XiIota, alphaI, 0, SumAroundei, s] = muOverMub[s] × Bound[tG, {1}, 1, s];
    Bound[XiIota, alphaII, 0, SumAroundZero, s] = 0;
    Bound[XiIota, RI, 0, Delta, ei, s] = 0;
    Bound[XiIota, RII, 0, Delta, 0, s] = 0;
    Bound[Pi, alpha, 0, s] = 0;
    Bound[Pi, R, 0, s] = (2 d - 1) gjz[s] × Bound[tG, {1}, 1, s](*x=0*);
    Bound[Pi, R, 0, Delta, eiek, s] = 2 d (4 + 2 × (2 d - 2)) gjz[o] × Bound[tG, {1}, 1, s]
    (*x=0*);
    , {s, {i, o}}];
```

Bounds for N=1

Implementation of the bounds stated in Lemma 5.3

In[6683]:=
```
Do[
  Bound[Xi, 1, s] = muOverMub[s] × Bound[P1, 0, s];
  Bound[Xi, R, 1, s] =
   muOverMub[s] ×
    (Bound[P1, 0, s] - Bound[A, 0, 0, s] (*b≠0*) + 2 Bound[Bubble, 4, s]
      (*b=0,w in 0,x *) + Bound[Triangle, 4, s]);

  Bound[Psi, RI, 1, s] =
   muOverMub[s] ×
    (Bound[P1, 0, s] - Bound[A, 0, 0, s] (*b≠0*) + 4 Bound[Loop, 4, s] +
      2 Bound[Bubble, 5, s] + Bound[Bubble, 3, s] + Bound[Bubble, 4, s] +
      Bound[Triangle, 5, s]);
  Bound[Psi, RII, 1, s] =
   muOverMub[s] ×
    (Bound[P1, 0, s] - Bound[A, 0, 0, s] (*b≠0*) + 2 Bound[Bubble, 4, s]
      (*b=0,w in 0,x *) + Bound[Triangle, 4, s]);

  Bound[Xi, 1, Delta, s] = Bound[DeltaStart, 0, s];
  Bound[Xi, R, 1, Delta, s] = Bound[DeltaStart, 0, s];
  Bound[Psi, RI, 1, Delta, s] = Bound[Xi, 1, Delta, s] + Bound[Psi, RI, 1, s];
  Bound[Psi, RII, 1, Delta, s] = Bound[Xi, R, 1, Delta, s];

  Bound[XiIota, 1, s] = muOverMub[s] × Bound[P1Iota, 0, s];
  Bound[XiIota, 1, Delta, 0, s] = Bound[D, 0, 0, s];
  Bound[XiIota, 1, Delta, ei, s] = Bound[D, ei, 0, s];
  , {s, {i, o}}];
```

Bounds for N = 2, 3

We consider also N=2,3 as a special case, as these are large enough to be of numerical significance. We first implement the bounds on the absolute value of the coefficient as stated in Proposition 5.4 of (II).

In[6684]:=
```
Do[
  Bound[Xi, 2, s] = muOverMub[s] × Vector[P1, s].Vector[EndOpen, s];
  Bound[Xi, 3, s] =
   muOverMub[s] × Vector[P1, s].Matrix[A, s].Vector[EndOpen, s];

  Bound[XiIota, 2, s] = muOverMub[s] × Vector[P1Iota, s].Vector[EndOpen, s];
  Bound[XiIota, 3, s] =
   muOverMub[s] × Vector[P1Iota, s].Matrix[A, s].Vector[EndOpen, s];
  , {s, {i, o}}];
```

We improve the bound on the weighted diagrams stated in Proposition 5.4 of (II) slightly. The bound stated in the proposition splits the weight $|x|_2^2$ using

$$|x|_2^2 \le 2|y|_2^2 + 2|x-y|_2^2$$

when splitting the weight along the different building block, see Figure 14 of (II). In the case that the shared lines collapses to a point (u=y, so that l=0) we can use

$$|x|_2^2 = |y|_2^2 + |x-y|_2^2 + 2\sum_i x_i y_i$$

By the spatial symmetry in all direction of the building blocks, that only exist in the full extent if the shared line is collapsed, the sum of $x_i\, y_i$ cancels out. Thus, the bounds stated in Proposition 5.4 of (II) is a factor 2 too big for this specific case. This improves the bound by around 10 percent. This is analogous to the improvement explained in (6.22) of (II).

In[6685]:=
```
Do[
  Bound[Xi, 2, Delta, s] =
   2 (Vector[P1, s].Vector[DeltaEnd, s] +
       Vector[DeltaStart, s].Vector[EndClosed, s]) -
    (Vector[P1, s][[3]] × Vector[DeltaEnd, s][[3]] +
      Vector[DeltaStart, s][[3]] × Vector[EndClosed, s][[3]]);
  Bound[Xi, 3, Delta, s] =
   3 (Vector[P1, s].Matrix[A, s].Vector[DeltaEnd, s] +
       Vector[P1, s].Matrix[C, s].Vector[EndClosed, s] +
       Vector[DeltaStart, s].Matrix[A, s].Vector[EndClosed, s]) -
    Sum[Vector[P1, s][[3]] × Matrix[A, s][[3, b]] × Vector[DeltaEnd, s][[b]] +
      Vector[P1, s][[3]] × Matrix[C, s][[3, b]] × Vector[EndClosed, s][[b]] +
      Vector[DeltaStart, s][[3]] × Matrix[A, s][[3, b]] × Vector[EndClosed, s][[b]]
     , {b, 1, 5}] -
    Sum[Vector[P1, s][[a]] × Matrix[A, s][[a, 3]] × Vector[DeltaEnd, s][[3]] +
      Vector[P1, s][[a]] × Matrix[C, s][[a, 3]] × Vector[EndClosed, s][[3]] +
      Vector[DeltaStart, s][[a]] × Matrix[A, s][[a, 3]] × Vector[EndClosed, s][[3]]
     , {a, 1, 5}];

  Do[
   Bound[XiIota, 2, Delta, type, s] =
    2 (Vector[P1Iota, s].Vector[DeltaEnd, s] +
        Vector[D, type, s].Vector[EndClosed, s]) -
     (Vector[P1Iota, s][[3]] × Vector[DeltaEnd, s][[3]] +
       Vector[D, type, s][[3]] × Vector[EndClosed, s][[3]]);
   Bound[XiIota, 3, Delta, type, s] =
    Bound[XiIota, 3, Delta, type, s] =
     3 (Vector[P1Iota, s].Matrix[A, s].Vector[DeltaEnd, s] +
         Vector[P1Iota, s].Matrix[C, s].Vector[EndClosed, s] +
         Vector[D, type, s].Matrix[A, s].Vector[EndClosed, s]) -
      Sum[Vector[P1Iota, s][[a]] × Matrix[A, s][[a, 3]] × Vector[DeltaEnd, s][[3]] +
        Vector[P1Iota, s][[a]] × Matrix[C, s][[a, 3]] × Vector[EndClosed, s][[3]] +
        Vector[D, type, s][[a]] × Matrix[A, s][[a, 3]] × Vector[EndClosed, s][[3]],
       {a, 1, 5}] -
      Sum[Vector[P1Iota, s][[3]] × Matrix[A, s][[3, b]] × Vector[DeltaEnd, s][[b]] +
        Vector[P1Iota, s][[3]] × Matrix[C, s][[3, b]] × Vector[EndClosed, s][[b]] +
        Vector[D, type, s][[3]] × Matrix[A, s][[3, b]] × Vector[EndClosed, s][[b]],
       {b, 1, 5}];
   , {type, {0, ei}}];
  , {s, {i, o}}];
```

Bounds for N ≥ 4

Next, we bound the sum over all odd and even $N \geq 4$. We use the earlier computed decomposition of the vectors P and P^i in term of eigenvectors of A. We use these eigenvectors and the geometric sum to compute the sum of the bounds, see Section 5.4 of (I). We begin with the bound on the absolute value.

In[6686]:=
```
Do[
 Do[
  vl[j] = Abs[EigenVector[left, j, s]];
  vi[j] = Abs[EigenVector[iota, j, s]];

  e[j] = Abs[EigenValue[j, s]];
  evi[j] = Abs[EigenValue[j, s]];
  , {j, 1, 5}];
 Bound[Xi, EvenTail, s] =
  muOverMub[s] × Sum[e[j]^2/(1 - e[j]^2) vl[j].Vector[EndOpen, s], {j, 1, 5}];
 Bound[Xi, OddTail, s] =
  muOverMub[s] × Sum[e[j]^3/(1 - e[j]^2) vl[j].Vector[EndOpen, s], {j, 1, 5}];
 Bound[XiIota, EvenTail, s] =
  muOverMub[s] × Sum[evi[j]^2/(1 - evi[j]^2) vi[j].Vector[EndOpen, s], {j, 1, 5}];
 Bound[XiIota, OddTail, s] =
  muOverMub[s] × Sum[evi[j]^3/(1 - evi[j]^2) vi[j].Vector[EndOpen, s], {j, 1, 5}];
 , {s, {i, o}}]
```

Then, we compute the bound on the weighted coefficients:

In[6687]:=
```
Do[
 Do[
  vl[j] = Abs[EigenVector[left, j, s]];
  vr[j] = Abs[EigenVector[right, j, s]];
  vi[j] = Abs[EigenVector[iota, j, s]];
  e[j] = Abs[EigenValue[j, s]];
  evi[j] = Abs[EigenValue[j, s]];
  , {j, 1, 5}];
 Bound[Xi, EvenTail, Delta, s] =
  Vector[DeltaStart, s].Sum[vr[j] * e[j]^2 (2/(1 - e[j]^2)^2 + 2/(1 - e[j]^2)), {j, 1, 5}] +
   Sum[(2/(1 - e[j]^2)^2 + 2/(1 - e[j]^2)) * vl[j] e[j]^2, {j, 1, 5}].Vector[DeltaEnd, s] +
   Sum[2/(1 - e[j]^2)^2 * vl[j], {j, 1, 5}].
    (Matrix[C, s].Matrix[A, s] + Matrix[A, s].Matrix[C, s]).
```

```
      Sum[2/(1 - e[j]^2) vr[j], {j, 1, 5}] +
     Sum[2/(1 - e[j]^2) * vl[j], {j, 1, 5}].
      (Matrix[C, s].Matrix[A, s] + Matrix[A, s].Matrix[C, s]).
      Sum[2/(1 - e[j]^2)^2 vr[j], {j, 1, 5}];
Bound[Xi, OddTail, Delta, s] =
  Vector[DeltaStart, s].Sum[vl[j] * e[j]^3 (2/(1 - e[j]^2)^2 + 3/(1 - e[j]^2)), {j, 1, 5}] +
    Sum[(2/(1 - e[j]^2)^2 + 3/(1 - e[j]^2)) * vl[j] e[j]^3, {j, 1, 5}].Vector[DeltaEnd, s] +
    Vector[DeltaStart, s].Matrix[C, s].
      Sum[vr[j] * e[j]^2 (2/(1 - e[j]^2)^2 + 3/(1 - e[j]^2)), {j, 1, 5}] +
    Sum[(2/(1 - e[j]^2)^2 + 1/(1 - e[j]^2)) * vl[j], {j, 1, 5}].Matrix[A, s].
      (Matrix[C, s].Matrix[A, s] + Matrix[A, s].Matrix[C, s]).
      Sum[2/(1 - e[j]^2) vr[j], {j, 1, 5}] +
    Sum[2/(1 - e[j]^2) * vl[j], {j, 1, 5}].Matrix[A, s].
      (Matrix[C, s].Matrix[A, s] + Matrix[A, s].Matrix[C, s]).
      Sum[2/(1 - e[j]^2)^2 vr[j], {j, 1, 5}];
Do[
    Bound[XiIota, EvenTail, Delta, type, s] =
      Vector[D, type, s].Sum[vr[j] * e[j]^2 (2/(1 - e[j]^2)^2 + 2/(1 - e[j]^2)), {j, 1, 5}] +
        Sum[(2/(1 - evi[j]^2)^2 + 2/(1 - evi[j]^2)) * vi[j] * evi[j]^2, {j, 1, 5}].
          Vector[DeltaEnd, s] + Sum[2/(1 - evi[j]^2)^2 * vi[j], {j, 1, 5}].
          (Matrix[C, s].Matrix[A, s] + Matrix[A, s].Matrix[C, s]).
          Sum[2/(1 - e[j]^2) vr[j], {j, 1, 5}] +
        Sum[2/(1 - evi[j]^2) * vi[j], {j, 1, 5}].
          (Matrix[C, s].Matrix[A, s] + Matrix[A, s].Matrix[C, s]).
```

```
      Sum[2/(1 - e[j]^2)^2 vr[j], {j, 1, 5}];
    Bound[XiIota, OddTail, Delta, type, s] =
     Vector[D, type, s].Sum[vr[j] * e[j]^3 (2/(1 - e[j]^2)^2 + 3/(1 - e[j]^2)), {j, 1, 5}] +
      Sum[evi[j]^3 (2/(1 - evi[j]^2)^2 + 3/(1 - evi[j]^2)) * vi[j], {j, 1, 5}].
       Vector[DeltaEnd, s] + Vector[P1Iota, s].Matrix[C, s].
       Sum[vr[j] * e[j]^2 (2/(1 - e[j]^2)^2 + 3/(1 - e[j]^2)), {j, 1, 5}] +
      Sum[(2/(1 - evi[j]^2)^2 + 1/(1 - evi[j]^2)) * vi[j], {j, 1, 5}].Matrix[A, s].
       (Matrix[C, s].Matrix[A, s] + Matrix[A, s].Matrix[C, s]).
       Sum[2/(1 - e[j]^2) vr[j], {j, 1, 5}] +
      Sum[2/(1 - evi[j]^2) * vi[j], {j, 1, 5}].Matrix[A, s].
       (Matrix[C, s].Matrix[A, s] + Matrix[A, s].Matrix[C, s]).
       Sum[2/(1 - e[j]^2)^2 vr[j], {j, 1, 5}];
   , {type, {0, ei}}] ×
  Clear[vl, vr, vi, e, evi]
 , {s, {i, o}}]
```

Lower bounds

The analysis of (I) requires a number of lower bounds on the coefficients. Implementing some possible lower bounds we found that while they improve our concluded bounds, using them did not allow us to lower the dimension in which we can obtain our result. So we decided to omit these lower bounds and their complication for our publication.

```
In[6688]:= Do[
  Bound[Pi, alpha, lower, 0, s] = 0;
  Bound[Psi, lower, 0, s] = 0;
  Bound[Pi, 1, Lower, s] = 0;
  , {s, {i, o}}];
```

Bounds for differences

To the bounds stated in Lemma 5.5. For these bounds we also decided to omit a number of terms, that we could have subtracted, as they did not allow us to prove the result in lower dimensions.

```
In[6689]:= Do[
  Bound[Xi, alpha, OneMinusZero, AtZero, s] = muOverMub[s] × (Bound[Loop, 4, s]);
  Bound[Xi, alpha, ZeroMinusOne, AtZero, s] = 0;

  Bound[Xi, alpha, OneMinusZero, AtEi, s] =
   1/(2 d) muOverMub[s] × (Bound[Bubble, 3, s] + 2 Bound[Loop, 4, s]);
  Bound[Xi, alpha, ZeroMinusOne, AtEi, s] = 0;
  Bound[Psi, alphaI, OneMinusZero, AroundEi, s] =
   2 gjz[s]^2 (Bound[tG, twoiNotusingi, 4, s] +
       (2 d - 2) Bound[tG, ikNotUsingi, 2, s]) +
    (2 d - 2) gjz[s]^2 (2 d gjz[s])^2 VarGamma2[s] × K[2, 2, {2}] +
    gjz[s]^2 (2 d gjz[s])^4 VarGamma2[s] × K[2, 4, {0, 1}];

  Bound[Psi, alphaI, ZeroMinusOne, AroundEi, s] =
   2 gjz[s]^2
    (Bound[tG, twoiNotusingi, 4, s] + (2 d - 2) Bound[tG, ikNotUsingi, 2, s]);

  Bound[Psi, alphaII, OneMinusZero, AroundZero, s] =
   (2 d - 1) Bound[Xi, alpha, OneMinusZero, AtEi, s];

  Bound[Psi, alphaII, ZeroMinusOne, AroundZero, s] =
   (2 d - 1) Bound[Xi, alpha, ZeroMinusOne, AtEi, s];
  , {s, {i, o}}];
```

Summing the bounds

We compute the sum over all odd/even N

In[6690]:=
```
Do[
 Bound[Xi, Even, s] = Sum[Bound[Xi, t, s], {t, {0, 2, EvenTail}}];
 Bound[Xi, Odd, s] = Sum[Bound[Xi, t, s], {t, {1, 3, OddTail}}];
 Bound[Xi, Absolut, s] = Sum[Bound[Xi, t, s], {t, {Odd , Even}}];

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


 Bound[XiIota, Even, s] = Sum[Bound[XiIota, t, s], {t, {0, 2, EvenTail}}];
 Bound[XiIota, Odd, s] = Sum[Bound[XiIota, t, s], {t, {1, 3, OddTail}}];
 Bound[XiIota, Absolut, s] = Sum[Bound[XiIota, t, s], {t, {Odd , Even}}];

 Do[
  Bound[XiIota, Even, Delta, type, s] =
   Sum[Bound[XiIota, t, Delta, type, s], {t, {0, EvenTail}}];
  Bound[XiIota, Odd, Delta, type, s] =
   Sum[Bound[XiIota, t, Delta, type, s], {t, {1, OddTail}}];
  Bound[XiIota, Absolut, Delta, type, s] =
   Sum[Bound[XiIota, t, Delta, type, s], {t, {Odd, Even}}];
  , {type, {ei, 0}}];

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

Bound on the simplified rewrite

In the preceding section we have computed all bounds required by Assumption 4.3 of (I). We use the methods provided in the General-Notebook to compute the bounds on the simplified rewrite, as derived in Appendix D of (I).

In[6691]:=
```
Do[
 mu[s] = gjz[s];
 mub[s] = gz[s];
 mumin[s] = 1/(2 d - 1);

 beta[CPhi, Lower, s] =
  betaCPhiLow[d, mu[s], Bound[Xi, alpha, OneMinusZero, AtZero, s],
   Bound[XiIota, alphaI, 0, Atei, s]];
 beta[CPhi, Upper, s] =
  betaCPhiUp[d, mu[s], Bound[Xi, alpha, ZeroMinusOne, AtZero, s],
   Bound[XiIota, alphaII, 0, AtZero, s]];

 beta[af, Lower, s] = betaAfLow[d, mumin[s], mu[s],
   Bound[Psi, alphaI, OneMinusZero, AroundEi, s],
   Bound[Psi, alphaII, ZeroMinusOne, AroundZero, s], Bound[Pi, alpha, 0, s]];
```

```
beta[af, Upper, s] =
 betaAfUp[d, mu[s], Bound[Psi, alphaI, ZeroMinusOne, AroundEi, s],
  Bound[Psi, alphaII, OneMinusZero, AroundZero, s],
  Bound[Pi, alpha, lower, 0, s]];

beta[ap, s] = betaap[d, mu[s], Bound[Xi, alpha, OneMinusZero, AtEi, s],
  Bound[Xi, alpha, ZeroMinusOne, AtEi, s],
  Bound[XiIota, alphaI, 0, SumAroundei, s],
  Bound[XiIota, alphaII, 0, SumAroundZero, s]];

beta[PiHat, s] = betaPiHat[d, mub[s], Bound[XiIota, Even, s],
  Bound[Pi, 1, Lower, s]];
beta[PsiHat, s] = betaPsiHatLower[d, mubOverMu[s], Bound[Xi, Odd, s],
  Bound[Psi, lower, 0, s]];


beta[Rf, s] = betaRF[d, mu[s], mub[s], mubOverMu[s], mub[s],
  Bound[Xi, Absolut, s], Bound[Xi, EvenTail, s] + Bound[Xi, OddTail, s],
  Bound[XiIota, Absolut, s], Sum[Bound[XiIota, t, s], {t, {Odd , EvenTail}}],
  Bound[Psi, RI, 0, s] + Bound[Psi, RI, 1, s],
  Bound[Psi, RII, 0, s] + Bound[Psi, RII, 1, s], Bound[Pi, R, 0, s]];


beta[Rp, s] = betaRp[d, mu[s], mubOverMu[s], mub[s], Bound[Xi, Absolut, s],
  Bound[Xi, R, 0, s] + Bound[Xi, R, 1, s],
  Bound[Xi, EvenTail, s] + Bound[Xi, OddTail, s], Bound[XiIota, Absolut, s],
  Sum[Bound[XiIota, t, s], {t, {Odd , EvenTail}}], Bound[XiIota, RI, 0, s],
  Bound[XiIota, RII, 0, s]];

beta[Rp, Delta, s] = betaRpDelta[d, mu[s], mubOverMu[s], mub[s],
  Bound[Xi, Absolut, s], Bound[Xi, Absolut, Delta, s],
  Bound[Xi, R, 0, s] + Bound[Xi, R, 1, s],
  Sum[Bound[Xi, t, Delta, s], {t, {OddTail , EvenTail}}],
  Bound[XiIota, Absolut, s], Bound[XiIota, Absolut, Delta, ei, s],
  Bound[XiIota, Absolut, Delta, 0, s],
  Sum[Bound[XiIota, t, Delta, ei, s], {t, {Odd, EvenTail}}],
  Sum[Bound[XiIota, t, Delta, 0, s], {t, {Odd, EvenTail}}],
  Bound[XiIota, RI, 0, Delta, ei, s], Bound[XiIota, RII, 0, Delta, 0, s]];


beta[Rf, abs, Delta, s] = betaRfDelta[d, mu[s], mubOverMu[s], mub[s],
  Bound[Xi, Absolut, s], Bound[Xi, Absolut, Delta, s],
  Bound[Xi, EvenTail, s] + Bound[Xi, OddTail, s],
  Sum[Bound[Xi, t, Delta, s], {t, {OddTail , EvenTail}}],
  Bound[Psi, RI, 0, Delta, s] + Bound[Psi, RI, 1, Delta, s],
  Bound[Psi, RII, 0, Delta, s] + Bound[Psi, RII, 1, Delta, s],
```

```
    Bound[XiIota, Absolut, s], Bound[XiIota, Absolut, Delta, ei, s],
    Bound[XiIota, Absolut, Delta, 0, s],
    Sum[Bound[XiIota, t, s], {t, {Odd, EvenTail}}],
    Sum[Bound[XiIota, t, Delta, ei, s], {t, {Odd, EvenTail}}],
    Bound[Pi, R, 0, Delta, eiek, s]];

  beta[Rf, Lower, Delta, s] =
   betaRfDeltaLower[d, mu[s], mubOverMu[s], mub[s], Bound[Xi, Absolut, s],
    Bound[Xi, Odd, s], Bound[Xi, Even, s], Bound[Xi, Absolut, Delta, s],
    Bound[Xi, Odd, Delta, s], Bound[Xi, Even, Delta, s], Bound[Xi, OddTail, s],
    Bound[Xi, OddTail, Delta, s], Bound[Xi, EvenTail, Delta, s],
    Bound[Psi, RI, 1, Delta, s], Bound[Psi, RII, 0, Delta, s],
    Bound[XiIota, Absolut, s], Bound[XiIota, Odd, s], Bound[XiIota, Even, s],
    Bound[XiIota, Absolut, Delta, ei, s], Bound[XiIota, Odd, Delta, ei, s],
    Bound[XiIota, Even, Delta, ei, s], Bound[XiIota, Absolut, Delta, 0, s],
    Bound[XiIota, Odd, Delta, 0, s], Bound[XiIota, Odd, Delta, 0, s],
    Bound[XiIota, EvenTail, s], Bound[XiIota, EvenTail, Delta, ei, s],
    Bound[Pi, R, 0, Delta, eiek, s]];
  , {s, {i, o}}]
```

Improvement of Bounds

In this section we implement the computations of Section 3 of (I) to verify whether we can conclude from $f_i(z) \leq \Gamma_i$ that $f_i(z) < \Gamma_i - \epsilon$. The sufficient condition for this to succeed is stated in Definition 2.9 of (I). We check the conditions one line at a time.

Technical conditions

All these conditions are neccessary conditions. However, numerically they are next to trivial, in the sense that other conditions (most likely f_2 or f_3) will fail first

```
In[6692]:= Do[
  TechCondition[I, s] = (beta[CPhi, Lower, s] - beta[ap, s] - beta[Rp, s]) > 0;
  (* Part of Assumption 2.7. of (I), stating that the nominator of Ĝz(k),
  being Φ̂(k), is positive *)
  TechCondition[II, s] = (beta[af, Lower, s] - beta[Rf, Lower, Delta, s]) > 0;
  (*Part of Assumption 2.7. of (I),
  neccesary to ensure that f2 is well defined *)
  TechCondition[III, s] = ((2 d - 1) mub[s] / (1 - mu[s]) Bound[XiIota, Absolut, s]) < 1;
  (* Condition (4.34) of (I),
  which is necessary to make the geometric series converge *)

  TechCondition[s] = TechCondition[I, s] && TechCondition[II, s] &&
    TechCondition[III, s]
  , {s, {i, o}}]
```

Improvement of f_1

Following the bounds derived in Section 3.1 of (I):

In[6693]:=
```
Do[
  boundF1[part1, s] = mubOverMu[s] (1 + beta[PiHat, s]) / (1 - (2 d)/(2 d-1) beta[PsiHat, s]);
  boundF1[part2, s] = cmu (1 + beta[PiHat, s]) / (1 - (2 d)/(2 d-1) beta[PsiHat, s]);
  boundF1[s] = Max[boundF1[part1, s], boundF1[part2, s]];

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

Improvement of f_2

Next we implement the bound derived in Section 3.2 of (I)

In[6694]:=
```
Do[
  boundF2[s] = (2 d - 1)/(2 d - 2) (beta[CPhi, Upper, s] + beta[ap, s] + beta[Rp, s]) / (beta[af, Lower, s] + beta[Rf, Lower, Delta, s]);
  , {s, {i, o}}]
```

Improvement of f_3

Here we start with some preparation and compute the value at z_I

In[6695]:=

```
(* The values to compare to *)
const[1] = GammaThreeClosed[1, 6];
const[2] = GammaThreeClosed[2, 6];
const[3] = GammaThree[1, 0];
const[4] = GammaThree[1, 1];
const[5] = GammaThree[1, 2];
const[6] = GammaThree[1, 3];
const[7] = GammaThree[2, 0];
const[8] = GammaThree[2, 1];
const[9] = GammaThree[2, 2];
const[10] = GammaThree[2, 3];

(* Inital bounds as in Section 3.3.3, obtained by pure SRW computations,
Methods provided in General.nb *)
boundF3[1, i] = BoundFThreeInital[d, 1, 6, 1, {{0}} ];
boundF3[2, i] = BoundFThreeInital[d, 2, 6, 1, {{0}} ];

boundF3[3, i] = BoundFThreeInital[d, 1, 0, 1, {{1}, {2}, {0, 1}} ];
boundF3[4, i] = BoundFThreeInital[d, 1, 1, 1, {{1}, {2}, {0, 1}} ];
boundF3[5, i] = BoundFThreeInital[d, 1, 2, 1, {{1}, {2}, {0, 1}} ];
boundF3[6, i] = BoundFThreeInital[d, 1, 3, 1, {{1}, {2}, {0, 1}} ];

boundF3[7, i] = BoundFThreeInital[d, 2, 0, 1, {{1}, {2}, {0, 1}} ];
boundF3[8, i] = BoundFThreeInital[d, 2, 1, 1, {{1}, {2}, {0, 1}} ];
boundF3[9, i] = BoundFThreeInital[d, 2, 2, 1, {{1}, {2}, {0, 1}} ];
boundF3[10, i] = BoundFThreeInital[d, 2, 3, 1, {{1}, {2}, {0, 1}} ];
```

In[6715]:=

In[6716]:=
```
(* The bounds summarized in (3.80) *)
BoundFThreeBound[n_, l_, vs_] :=
  BoundFThree[d, n, l, vs, Gamma2, beta[CPhi, Upper, o], beta[af, Lower, o],
   beta[af, Upper, o], beta[ap, o], beta[Rp, o], beta[Rf, o],
   beta[Rf, abs, Delta, o], beta[Rp, Delta, o],
   1/(beta[af, Lower, o] + beta[Rf, Lower, Delta, o])];

boundF3[1, o] := BoundFThreeBound[1, 6, {{0}}];
boundF3[2, o] = BoundFThreeBound[2, 6, {{0}}];

BoundFThreeOpen[n_, l_] := BoundFThreeBound[n, l, {{1}, {2}, {0, 1}}];

onstepContribution = BoundFThreeBound[1, 2, {{1}}] +
   1/(2 d) (2 × (2 d - 2) Bound[tG, {2}, 2, o] + 4 Bound[tG, {0, 1}, 2, o]);
boundF3[3, o] = Max[2 d gjz[o] onstepContribution + Bound[tG, {1}, 1, o],
   2 d gjz[o] × BoundFThreeBound[1, 1, {{2}}] + Bound[tG, {2}, 2, o],
   2 d gjz[o] × BoundFThreeBound[1, 1, {{0, 1}}] + Bound[tG, {0, 1}, 2, o],
   BoundFThreeBound[1, 0, {{3}, {1, 1}, {0, 0, 1}}]];
boundF3[4, o] = Max[onstepContribution, BoundFThreeBound[1, 1, {{2}, {0, 1}}]];
boundF3[5, o] = BoundFThreeOpen[1, 2];
boundF3[6, o] = BoundFThreeOpen[1, 3];

boundF3[7, o] =
  Max[2 d gjz[o] × (BoundFThreeBound[1, 1, {{1}}] + BoundFThreeBound[2, 1, {{1}}]) +
     Bound[tG, {1}, 2, o],
    2 d gjz[o] × (BoundFThreeBound[1, 1, {{2}}] + BoundFThreeBound[2, 1, {{2}}]) +
     Bound[tG, {2}, 2, o],
    2 d gjz[o] × (BoundFThreeBound[1, 1, {{0, 1}}] +
        BoundFThreeBound[2, 1, {{0, 1}}]) + + Bound[tG, {0, 1}, 2, o],
    BoundFThreeBound[2, 0, {{3}, {1, 1}, {0, 0, 1}}]];
boundF3[8, o] = BoundFThreeOpen[2, 1];
boundF3[9, o] = BoundFThreeOpen[2, 2];
boundF3[10, o] = BoundFThreeOpen[2, 3];


BoundsF3Table = Table[boundF3[j, s] / const[j], {s, {i, o}}, {j, 1, 10}];
boundF3[i] = Ceiling[Max[BoundsF3Table[[1]]], 10^-9];
boundF3[o] = Max[BoundsF3Table[[2]]];
```

Results

Preparation of output

In[6731]:=
```
Do[
  SuccesF[1, s] = boundF1[s] < Gamma1;
  SuccesF[2, s] = boundF2[s] < Gamma2;
  SuccesF[3, s] = boundF3[s] < Gamma3;
  Succes[s] = SuccesF[1, s] && SuccesF[2, s] && SuccesF[3, s] && TechCondition[s];
  , {s, {i, o}}];
Succes[overall] = Succes[i] && Succes[o];
```

In[6733]:=
```
overAllStatement = "The statement that the bootstrap was successful is "
   If[Succes[overall], Style[TextString[Succes[overall]], Bold, Green],
    Style[TextString[Succes[overall]], Bold, Red]];
```

In[6734]:=
```
bubbles = {Graphics[{Green, Disk[{0, 0}, 0.8]}, ImageSize → 30],
   Graphics[{Red, Disk[{0, 0}, 0.8]}, ImageSize → 40]};

Do[
 CoefficientboundsTable[s] =
  {{Quantity, "ΞZero", "ΞOne", "ΞTwo", "ΞThree", "ΞEven,>3", "ΞOdd,>3"},
   {Text[Bound for Ξ̂], Bound[Xi, 0, s], Bound[Xi, 1, s] , Bound[Xi, 2, s],
    Bound[Xi, 3, s] , Bound[Xi, EvenTail, s], Bound[Xi, OddTail, s]},
   {Text[Bound for Ξ̂ι], Bound[XiIota, 0, s], Bound[XiIota, 1, s] ,
    Bound[XiIota, 2, s], Bound[XiIota, 3, s] , Bound[XiIota, EvenTail, s],
    Bound[XiIota, OddTail, s]},
   {Text[Ξ̂ "|x|22" ], Bound[Xi, 0, Delta, s], Bound[Xi, 1, Delta, s],
    Bound[Xi, 2, Delta, s], Bound[Xi, 3, Delta, s], Bound[Xi, EvenTail, Delta, s],
    Bound[Xi, OddTail, Delta, s]},
   {Text[Ξι "|x-eι|22"] , Bound[XiIota, 0, Delta, ei, s],
    Bound[XiIota, 1, Delta, ei, s] , Bound[XiIota, 2, Delta, ei, s],
    Bound[XiIota, 3, Delta, ei, s], Bound[XiIota, EvenTail, Delta, ei, s],
    Bound[XiIota, OddTail, Delta, ei, s]},
   {Text[Ξι "|x|22"] , Bound[XiIota, 0, Delta, 0, s], Bound[XiIota, 1, Delta, 0, s] ,
    Bound[XiIota, 2, Delta, 0, s], Bound[XiIota, 3, Delta, 0, s],
    Bound[XiIota, EvenTail, Delta, 0, s], Bound[XiIota, OddTail, Delta, 0, s]}};
 TableCoefficients[s] =
  Labeled[Grid[CoefficientboundsTable[s], Alignment → {Center},
     Frame → True, Dividers → {{2 → True, -1 → True}, {2 → True}},
     ItemStyle → {1 → Bold, 1 → Bold},
     Background → {{None}, {GrayLevel[0.9]}, {None}}],
    Style["Bounds on the coefficients in dimension " <> TextString[d], Bold],
    Top] // Text;


 CoefficientSplitsBoundsI[s] =
```

```
{{Quantity, "Ξ_R^Zero", "Ψ_RI^Zero", "Ψ_RII^Zero", "Ξ_R^One", "Ψ_RI^One", "Ψ_RII^One"},
 {Text[Abs Bound], Bound[Xi, R, 0, s], Bound[Psi, RI, 0, s],
  Bound[Psi, RII, 0, s], Bound[Xi, R, 1, s] , Bound[Psi, RI, 1, s],
  Bound[Psi, RII, 1, s]}, {Text[Ξ̂ "|x|_2^2" ], Bound[Xi, R, 0, Delta, s],
  Bound[Psi, RI, 0, Delta, s], Bound[Psi, RII, 0, Delta, s],
  Bound[Xi, R, 1, Delta, s] , Bound[Psi, RI, 1, Delta, s],
  Bound[Psi, RII, 1, Delta, s]}};
TableCoefficientSplitI[s] =
 Labeled[Grid[CoefficientSplitsBoundsI[s], Alignment → {Center},
    Frame → True, Dividers → {{2 → True, -1 → True}, {2 → True}},
    ItemStyle → {1 → Bold, 1 → Bold},
    Background → {{None}, {GrayLevel[0.9]}, {None}}],
   Style[
    "Bounds on the remainder terms of the split of Ξ and Ψ in dimension " <>
     TextString[d], Bold], Top] // Text;

CoefficientSplitsBoundsII[s] =
 {{Quantity, " Ξ̂_alpha^zero(0)-Ξ̂_alpha^One(0)", " Ξ̂_alpha^zero(e_1)-Ξ̂_alpha^One(e_1)", "∑Ψ_alphaI^Zero-Ψ_alphaI^One",
   "∑Ψ_alphaII^Zero-Ψ_alphaII^One"},
  {Text[Lower Bound], Bound[Xi, alpha, OneMinusZero, AtZero, s],
   Bound[Xi, alpha, OneMinusZero, AtEi, s],
   Bound[Psi, alphaI, OneMinusZero, AroundEi, s],
   Bound[Psi, alphaII, OneMinusZero, AroundZero, s]},
  {Text[Upper Bound ], Bound[Xi, alpha, ZeroMinusOne, AtZero, s],
   Bound[Xi, alpha, ZeroMinusOne, AtEi, s],
   Bound[Psi, alphaI, ZeroMinusOne, AroundEi, s],
   Bound[Psi, alphaII, ZeroMinusOne, AroundZero, s]}};
TableCoefficientSplitII[s] =
 Labeled[Grid[CoefficientSplitsBoundsII[s], Alignment → {Center},
    Frame → True,
    Dividers → {{2 → True, 3 → True, 4 → True, 5 → True, -1 → True}, {2 → True}},
    ItemStyle → {1 → Bold, 1 → Bold},
    Background → {{None}, {GrayLevel[0.9]}, {None}}],
   Style["Bounds on the difference of explicit terms in dimension " <>
     TextString[d], Bold], Top] // Text;

CoefficientSplitsBoundsIII[s] =
 {{Quantity, " Ξ̂^ι", "|x-e_ι|_2^2 Ξ̂^ι", " Ξ̂_alphaI^ι(e_ι)", "∑ Ξ̂_alphaI^ι", " Ξ̂_RI^ι",
   "|x-e_ι|_2^2 Ξ̂_RI^ι", "|x|_2^2 Ξ̂^ι", " Ξ̂_alphaII^ι(0)", "∑ Ξ̂_alphaII^ι", "Ξ̂_RII^ι", "|x|_2^2 Ξ̂_RII^ι"},
  {, Bound[XiIota, 0, s], Bound[XiIota, 0, Delta, ei, s],
   Bound[XiIota, alphaI, 0, Atei, s], Bound[XiIota, alphaI, 0, SumAroundei, s],
   Bound[XiIota, RI, 0, s], Bound[XiIota, RI, 0, Delta, ei, s],
   Bound[XiIota, 0, Delta, 0, s], Bound[XiIota, alphaII, 0, AtZero, s],
   Bound[XiIota, alphaII, 0, SumAroundZero, s], Bound[XiIota, RII, 0, s],
   Bound[XiIota, RII, 0, Delta, 0, s]}};
```

```
TableCoefficientSplitIII[s] =
  Labeled[Grid[CoefficientSplitsBoundsIII[s], Alignment → {Center},
     Frame → True, Dividers → {{2 → True, 5 → True, 8 → True, -1 → True},
       {2 → True}},
     ItemStyle → {1 → Bold, 1 → Bold},
     Background → {{None}, {GrayLevel[0.9]}, {None}}],
    Style["Bounds on split of the coefficient Ξ^(0),ι in dimension " <>
      TextString[d], Bold], Top] // Text;
 , {s, {i, o}}]

SimpleNotationBoundsPart1 =
  {{Quantity, "aF-Lower", "aF-Upper", "|aΦ|", "cΦ-Lower", "cΦ-Upper", Π, Ψ},
   {"Bound i" , beta[af, Lower, i], beta[af, Upper, i], beta[ap, i],
    beta[CPhi, Lower, i], beta[CPhi, Upper, i], beta[PiHat, i], beta[PsiHat, i]},
   {"Bound o" , beta[af, Lower, o], beta[af, Upper, o], beta[ap, o],
    beta[CPhi, Lower, o], beta[CPhi, Upper, o], beta[PiHat, o], beta[PsiHat, o]}};

TableSimpleNotation1 =
  Labeled[Grid[SimpleNotationBoundsPart1, Alignment → {Center},
     Frame → True, Dividers → {{2 → True, -1 → True}, {2 → True}},
     ItemStyle → {1 → Bold, 1 → Bold},
     Background → {{None}, {GrayLevel[0.9]}, {None}}],
    Style["Bounds simplified rewrite in dimension " <> TextString[d], Bold],
    Top] // Text;

SimpleNotationBoundsPart2 =
  {{Quantity, "|RF|", "|RΦ|", "x2^2RF-lower", "|x2^2RF|", "|x2^2RΦ|"},
   {"Bound i", beta[Rf, i], beta[Rp, i], beta[Rf, Lower, Delta, i],
    beta[Rf, abs, Delta, i], beta[Rp, Delta, i]},
   {"Bounds on", beta[Rf, o], beta[Rp, o], beta[Rf, Lower, Delta, o],
    beta[Rf, abs, Delta, o], beta[Rp, Delta, o]}};
TableSimpleNotation2 =
  Labeled[Grid[SimpleNotationBoundsPart2, Alignment → {Center},
     Frame → True, Dividers → {{2 → True, 4 → True, -1 → True}, {2 → True}},
     ItemStyle → {1 → Bold, 1 → Bold},
     Background → {{None}, {GrayLevel[0.9]}, {None}}],
    Style["Bounds on remainder of rewrite in dimension " <> TextString[d], Bold],
    Top] // Text;

ContentCheckf1f2 =
  {{Bounds, "f1(zI)", "f2(zI)", "f3(zI)", "f1(z)", "f2(z)", "f3(z)"},
   {"Γi", NumberForm[N[Gamma1], 10], NumberForm[N[Gamma2], 10],
    NumberForm[N[Gamma3], 10], NumberForm[N[Gamma1], 10],
    NumberForm[N[Gamma2], 10], NumberForm[N[Gamma3], 10] },
   {Bounds, NumberForm[boundF1[i], 10], NumberForm[boundF2[i], 10],
    NumberForm[N[boundF3[i]], 10], NumberForm[boundF1[o], 10],
    NumberForm[boundF2[o], 10], NumberForm[boundF3[o], 10]}, { "check",
```

```
    If[SuccesF[1, i], bubbles⟦1⟧, bubbles⟦2⟧],
    If[SuccesF[2, i], bubbles⟦1⟧, bubbles⟦2⟧],
    If[SuccesF[3, i], bubbles⟦1⟧, bubbles⟦2⟧],
    If[SuccesF[1, o], bubbles⟦1⟧, bubbles⟦2⟧],
    If[SuccesF[2, o], bubbles⟦1⟧, bubbles⟦2⟧],
    If[SuccesF[3, o], bubbles⟦1⟧, bubbles⟦2⟧]}};

TableCheckf1f2 =
  Labeled[Grid[ContentCheckf1f2, Alignment → {Center}, Frame → True,
     Dividers → {{2 → True, -1 → True}, {2 → True}},
     ItemStyle → {1 → Bold, 1 → Bold},
     Background → {{None}, {GrayLevel[0.9]}, {None}}],
    Style["Bounds on the LT-bootstrap functions in dimension " <> TextString[d],
     Bold], Top] // Text;

Do[
 tableCheckf3Bubble[s] =
  {{Bounds, "F3-1,6,{0}", "F3-1,0,x≠0", "F3-1,1,x≠0", "F3-1,2,x≠0", "F3-1,3,x≠0"},
   {"Assumed bounds", NumberForm[N[const[1] * Gamma3, 8], 8],
    NumberForm[N[const[3] * Gamma3, 8], 8], NumberForm[N[const[4] * Gamma3, 8], 8],
    NumberForm[N[const[5] * Gamma3, 8], 8], NumberForm[N[const[6] * Gamma3, 8], 8] },
   {"Concluded bounds", NumberForm[N[boundF3[1, s], 8], 8],
    NumberForm[N[boundF3[3, s], 8], 8], NumberForm[N[boundF3[4, s], 8], 8],
    NumberForm[N[boundF3[5, s], 8], 8], NumberForm[N[boundF3[6, s], 8], 8]},
   { "check",
    If[boundF3[1, s] < const[1] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧],
    If[boundF3[3, s] < const[3] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧],
    If[boundF3[4, s] < const[4] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧],
    If[boundF3[5, s] < const[5] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧],
    If[boundF3[6, s] < const[6] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧]}};

 tableCheckf3Triangle[s] =
  {{Bounds, "F3-2,6,{0}", "F3-2,0,x≠0", "F3-2,1,x≠0", "F3-2,2,x≠0", "F3-2,3,x≠0"},
   {"Assumed bounds", NumberForm[N[const[2] * Gamma3, 8], 8] ,
    NumberForm[N[const[7] * Gamma3, 8], 8], NumberForm[N[const[8] * Gamma3, 8], 8],
    NumberForm[N[const[9] * Gamma3, 8], 8],
    NumberForm[N[const[10] * Gamma3, 8], 8] },
   {"Concluded bounds", NumberForm[N[boundF3[2, s], 8], 8],
    NumberForm[N[boundF3[7, s], 8], 8], NumberForm[N[boundF3[8, s], 8], 8],
    NumberForm[N[boundF3[9, s], 8], 8], NumberForm[N[boundF3[10, s], 8], 8]},
   { "check",
    If[boundF3[2, s] < const[2] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧],
    If[boundF3[7, s] < const[7] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧],
    If[boundF3[8, s] < const[8] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧],
    If[boundF3[9, s] < const[9] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧],
    If[boundF3[10, s] < const[10] * Gamma3, bubbles⟦1⟧, bubbles⟦2⟧]}};

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

```
TableCheckf3Bubble =
  Labeled[Grid[tableCheckf3Bubble[o], Alignment → {Center}, Frame → True,
     Dividers → {{2 → True, -1 → True}, {2 → True}},
     ItemStyle → {1 → Bold, 1 → Bold},
     Background → {{None}, {GrayLevel[0.9]}, {None}}],
    Style["Bounds on the LT-weighted bubble in dimension " <> TextString[d],
     Bold], Top] // Text;

TableCheckf3Triangle =
  Labeled[Grid[tableCheckf3Triangle[o], Alignment → {Center}, Frame → True,
     Dividers → {{2 → True, -1 → True}, {2 → True}},
     ItemStyle → {1 → Bold, 1 → Bold},
     Background → {{None}, {GrayLevel[0.9]}, {None}}],
    Style["Bounds on the LT-weighted triangle in dimension " <> TextString[d],
     Bold], Top] // Text;
```

Simple output

In[6745]:=
```
overAllStatement
```

Out[6745]= The statement that the bootstrap was successful is **True**

If this succeeds, then the infrared bound Theorem 2.10 of (I) or Theorem 1.1 of (II) holds with

In[6746]:=
```
"Ĝz(k)[1-D̂(k)] ≤ " <> TextString[Ceiling[(2 d - 2)/(2 d - 1) Gamma2, 0.001]]
"Ĝ̄z(k)[1-D̂(k)] ≤ " <> TextString[Ceiling[g[o] (2 d - 2)/(2 d - 1) Gamma2, 0.001]] <>
 " = A2(d)"
"A1(d) = " <>
 TextString[
  Ceiling[g[o] × (beta[CPhi, Upper, o] + beta[ap, o] + beta[Rp, o])
    Max[1/(beta[af, Lower, o] + beta[Rf, Lower, Delta, o]),
     1/(beta[CPhi, Lower, o] - beta[ap, o] - beta[Rp, o])], 0.001]]
```

Out[6746]= $\hat{G}_z(k)\,[1-\hat{D}(k)] \leq 1.293$

Out[6747]= $\hat{\bar{G}}_z(k)\,[1-\hat{D}(k)] \leq 3.872 = A_2(d)$

Out[6748]= $A_1(d) = 3.872$

Further, we have proven that g_{z_c} and $g_{z_c}\,z_c$, respectively are smaller than:

In[6749]:=
```
Print["g_{z_c} ≤ ", Ceiling[Exp[1] Gamma1, 0.0001]]
Print["(2d-1) z_c g_{z_c} ≤ ", Ceiling[Gamma1, 0.0001]]
```

$g_{z_c} \le 2.9963$

$(2d-1) z_c g_{z_c} \le 1.1023$

Bounds on the bootstrap functions

In[6751]:=
```
TableCheckf1f2
TableCheckf3Bubble
TableCheckf3Triangle
```

Bounds on the LT–bootstrap functions in dimension 16

Out[6751]=

Bounds	$f_1(z_I)$	$f_2(z_I)$	$f_3(z_I)$	$f_1(z)$	$f_2(z)$	$f_3(z)$
Γ_i	1.10225007	1.335307	1.	1.10225007	1.335307	1.
Bounds	1.087512433	1.175876674	0.160056646	1.102250067	1.335306858	0.9999998442
check	●	●	●	●	●	●

Bounds on the LT–weighted bubble in dimension 16

Out[6752]=

Bounds	F3–1,6,{0}	F3–1,0,x≠0	F3–1,1,x≠0	F3–1,2,x≠0	F3–1,3,x≠0
Assumed bounds	0.0036348100	0.2913684	0.077867200	0.035275500	0.015211960
Concluded bounds	0.0036348017	0.29136833	0.077867164	0.035275459	0.015211952
check	●	●	●	●	●

Bounds on the LT–weighted triangle in dimension 16

Out[6753]=

Bounds	F3–2,6,{0}	F3–2,0,x≠0	F3–2,1,x≠0	F3–2,2,x≠0	F3–2,3,x≠0
Assumed bounds	0.01167464	0.4864781	0.21793600	0.09178173	0.0422438
Concluded bounds	0.011674638	0.48647795	0.21793564	0.091781714	0.042243741
check	●	●	●	●	●

Bounds on the coefficients

In[6754]:=
```
TableCoefficients[o]
TableSimpleNotation1
TableSimpleNotation2
TableCoefficientSplitI[o]
TableCoefficientSplitII[o]
TableCoefficientSplitIII[o]
```

Bounds on the coefficients in dimension 16

Out[6754]=

Quantity	Ξ^{Zero}	Ξ^{One}	Ξ^{Two}	Ξ^{Three}	$\Xi^{Even,>3}$	$\Xi^{Odd,>3}$
Bound for $\hat{\Xi}$	0	0.0201978	0.00203196	0.000209369	0.0000256693	2.85872×10^{-6}
Bound for $\hat{\Xi}'$	0.0357884	0.00190295	0.000192147	0.0000198012	2.42808×10^{-6}	2.70412×10^{-7}
$\lvert x\rvert_2^2 \hat{\Xi}$	0	0.0714926	0.154269	0.0366432	0.0147036	0.00638245
$\lvert x-e_l\rvert_2^2 \Xi'$	0.0357884	0.021593	0.0191486	0.00412877	0.00149134	0.000230642
$\lvert x\rvert_2^2 \Xi'$	0	0.00809565	0.0150407	0.00352682	0.00138184	0.000215419

Out[6755]=

Bounds simplified rewrite in dimension 16

Quantity	a_F–Lower	a_F–Upper	$\lvert a_\Phi\rvert$	c_Φ–Lower	c_Φ–Upper	Π	Ψ
Bound i	1.03311	1.03359	0.03988577-0176091-2432228-612778	0.99879203-3714501-1146114-7811343	1.00111427-6061840-2025826-0650312-49	0.03527080-7477086-5365540-595006	0.01353107-3253739-1684012-02176
Bound o	1.03298	1.10017	0.0467684	0.998396	1.00135	0.0409417	0.0211404

Out[6756]=

Bounds on remainder of rewrite in dimension 16

Quantity	$\lvert R_F\rvert$	$\lvert R_\Phi\rvert$	$x_2^2 R_F$–lower	$\lvert x_2^2 R_F\rvert$	$\lvert x_2^2 R_\Phi\rvert$
Bound i	0.1272508175809093151-0581804	0.0097733891195887351-5668074	–0.109714	0.122654	0.0178458
Bounds on	0.165521	0.0165093	–0.209113	0.277981	0.0792803

Out[6757]=

Bounds on the remainder terms of the split of Ξ and Ψ in dimension 16

Quantity	Ξ_R^{Zero}	Ψ_{RI}^{Zero}	Ψ_{RII}^{Zero}	Ξ_R^{One}	Ψ_{RI}^{One}	Ψ_{RII}^{One}
Abs Bound	0	0	0	0.0112242	0.0186725	0.0112242
$\lvert x\rvert_2^2 \hat{\Xi}$	0	0	0	0.0714926	0.0901651	0.0714926

Out[6758]=

Bounds on the difference of explicit terms in dimension 16

Quantity	$\hat{\Xi}_{alpha}^{zero}(0)-\hat{\Xi}_{alpha}^{One}(0)$	$\hat{\Xi}_{alpha}^{zero}(e_1)-\hat{\Xi}_{alpha}^{One}(e_1)$	$\Sigma\Psi_{alphaI}^{Zero}-\Psi_{alphaI}^{One}$	$\Sigma\Psi_{alphaII}^{Zero}-\Psi_{alphaII}^{One}$
Bound Lower	0.00160437	0.000231522	0.000340669	0.0071772
Bound Upper	0	0	0.000102376	0

Out[6759]=

Bounds on split of the coefficient $\Xi^{(0),\prime}$ in dimension 16

Quanti-ty	$\hat{\Xi}'$	$\lvert x-e_I\rvert_2^2 \hat{\Xi}'$	$\hat{\Xi}'_{alphaI}(e_I)$	$\Sigma \hat{\Xi}'_{alphaI}$	$\hat{\Xi}'_{RI}$	$\lvert x-e_I\rvert_2^2 \hat{\Xi}'_{RI}$	$\lvert x\rvert_2^2 \hat{\Xi}'$	$\hat{\Xi}'_{alphaII}(0)$	$\Sigma \hat{\Xi}'_{alphaII}$	$\hat{\Xi}'_{RII}$	$\lvert x\rvert_2^2 \hat{\Xi}'_{RII}$
	0.0357-884	0.0357-884	0	0.0357-884	0	0	0	0.0357-884	0	0	0

In[6760]:=

In[6761]:=

Algorithm to find good values for the constants

The follow is a semi-automated procedure to find appropriate values for the constants $\Gamma_i$and c_i.

How to use it: Initially, we guess a good value for the constant and make a first computation. Then, we deactivate the choice of the constants at the very beginning of this document. Then we recompile the entire document multiple times (Menu Evaluate>>Evaluate notebook) and hope that the algorithm below converges to a fix-point for the parameters.

The idea to use the previously concluded bounds(+ ε) as new initial values for the Γs and constants. Then, recompile and how that we can conclude the same bounds starting from these values. As initial value we recommend to either use the values of the bounds at z_I, denote by s=i, which are independent of the values. Another reasonable choice would be to start with values that work in a slightly higher dimension.

In[6762]:=

```
Gamma1 = Ceiling[Max[boundF1[i], boundF1[o]], 10^-8] + 10^-8;
Gamma2 = Ceiling[Max[boundF2[i], boundF2[o]], 10^-8] + 10^-8;


GammaThreeClosed[1, 6] = Ceiling[Max[boundF3[1, i], boundF3[1, o]], 10^-8] + 10^-8;
GammaThreeClosed[2, 6] = Ceiling[Max[boundF3[2, i], boundF3[2, o]], 10^-8] + 10^-8;
GammaThree[1, 0] = N[Ceiling[Max[boundF3[3, i], boundF3[3, o]], 10^-8] + 10^-8];
GammaThree[1, 1] = Ceiling[Max[boundF3[4, i], boundF3[4, o]], 10^-8] + 10^-8;
GammaThree[1, 2] = Ceiling[Max[boundF3[5, i], boundF3[5, o]], 10^-8] + 10^-8;
GammaThree[1, 3] = Ceiling[Max[boundF3[6, i], boundF3[6, o]], 10^-8] + 10^-8;
GammaThree[2, 0] = Ceiling[Max[boundF3[7, i], boundF3[7, o]], 10^-8] + 10^-8;
GammaThree[2, 1] = Ceiling[Max[boundF3[8, i], boundF3[8, o]], 10^-8] + 10^-8;
GammaThree[2, 2] = Ceiling[Max[boundF3[9, i], boundF3[9, o]], 10^-8] + 10^-8;
GammaThree[2, 3] = Ceiling[Max[boundF3[10, i], boundF3[10, o]], 10^-8] + 10^-8;

cmu = mubOverMu[o] + 0.000001;
```