Computations of the NoBLE for self-avoiding walk

Analysis of Section 3.3

Robert Fitzner
University of Technology Eindhoven

Abstract

In this file we implement the numerical part of the analysis of the non-backtracking lace expansion for nearest-neighbor self-avoiding walk. All references in this version of the notebook will be to the PhD thesis of the author.

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

We first compute bounds on the simple random walk two-point function (Section 5.2). 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 4.2. and then perform the analysis as explained in Section 3.3.

6.05.2013

Input

In which dimension should we perform the computation

```
d = 8;
```

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

```
Gamma1 = 1.00354; Gamma2 = 1.05151; Gamma3 = 1.15;
```

For the bootstrap function f_3 we use the following constants

```
c1 = 0.149; c2 = 0.4967; c3 = 0.11066; c4 = 4.735;
```

To obtain the result in dimension d=8 choose Gamma1=1.00354;Gamma2=1.05151;Gamma3=1.15; and c1=0.149;c2=0.4967;c3=0.11066;c4=4.735;

Simple Random Walk integral

Overview

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

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

see Section 5.2, using that

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

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

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

```
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}]
```

Computation

Then, we define the number of n-step SRW loop as given in Section 5.2.2, see (5.2.6)-(5.2.10)

```
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!)];
```

and compute the number $I_{n,m}(0)$ for n=0,1,2 and m=1,...,10 and save them in an two-dimensional array:

```
TableTwoPointValues = {{n m, 0, 1, 2}, {0, 1, NInt[1, d, ∞], NInt[2, d, ∞]},
   {1, 0, 0, 0}, {2, s2 / (2 d)^2, 0, 0}, {3, 0, 0, 0}, {4, s4 / (2 d)^4, 0, 0},
   {5, 0, 0, 0}, {6, s6 / (2 d)^6, 0, 0}, {7, 0, 0, 0}, {8, s8 / (2 d)^8, 0, 0}, {9, 0, 0, 0},
   {10, -1, 0, 0}};
For[i = 3, i < 13, i++,
 For[j = 3, j < 5, j++,
  TableTwoPointValues[[i, j]] =
    TableTwoPointValues[[i - 1, j]] - TableTwoPointValues[[i - 1, j - 1]];]
]
Clear[i, j]
```

For latter reference we define the constants:

```
I10 = TableTwoPointValues[[2, 3]];
I11 = TableTwoPointValues[[3, 3]];
I12 = TableTwoPointValues[[4, 3]];
I14 = TableTwoPointValues[[6, 3]];
I16 = TableTwoPointValues[[8, 3]];
I18 = TableTwoPointValues[[10, 3]];
I110 = TableTwoPointValues[[12, 3]];
I20 = TableTwoPointValues[[2, 4]];
I21 = TableTwoPointValues[[3, 4]];
I22 = TableTwoPointValues[[4, 4]];
I24 = TableTwoPointValues[[6, 4]];
I26 = TableTwoPointValues[[8, 4]];
I28 = TableTwoPointValues[[10, 4]];
I210 = TableTwoPointValues[[12, 4]];
```

Print out of all computed values of $I_{n,m}(0)$

```
Labeled[Grid[TableTwoPointValues,
   Alignment -> {{Left, Center}, Baseline, {{2, 12}, {2, 3}} -> {"."}},
   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, 4}} -> GrayLevel[0.9]}],
  Style["Value of the SRW two-point function in dimension " Text[d] , Bold],
  Top] // Text
```

Value of the SRW two−point function in dimension 8

m n	**0**	**1**	**2**
0	1	1.07865	1.289
1	0	0.078647	0.210356
2	0.0625	0.078647	0.131709
3	0	0.016147	0.0530618
4	0.0109863	0.016147	0.0369147
5	0	0.00516068	0.0207677
6	0.00301361	0.00516068	0.015607
7	0	0.00214707	0.0104464
8	0.00108259	0.00214707	0.00829929
9	0	0.00106449	0.00615221
10	−1	0.00106449	0.00508773

Bound on the two-point function and on repulsive diagrams

Definition of Constants

We consider two setting s: we use s=i for bound on $z = z_I$ and s=o for bound on zϵ $(z_I,\ z_c)$. In Section 5.1 we argued that the bounds have a similar for. As given in (5.1.13) and (5.1.14) we define

```
z[i] = 1/(2 d - 1);
z[o] = Gamma1/(2 d - 1);
VarGamma2[i] = 2 (d - 1)/(2 d - 1);(* G_{1/(2d-1)}(x) >=B_{1/(2d-1)}(x) = (2d-2)/(2d-1) C_{1/2d}(x)*)
VarGamma2[o] = 2 (d - 1)/(2 d - 1) Gamma2;
```

Further, we define the constants given in Section 5.1.3.

```
c2ik = 2; (*c2(e1+e2)*)
c4ik = 4 + 6 (2 d - 4);(*c4(e1+e2)*)
c6ik = 16 + 84 (2 d - 4) + 36 (2 d - 4) (2 d - 6);(*c4(e1+e2)*)

c3i = (2 d - 2);(*c3(e1)*)
c5i = (3 (2 d - 2 ) + 4 (2 d - 2) (2 d - 4) );(*c5(e1)*)
c7i = (14 (2 d - 2 ) + 62 (2 d - 2) (2 d - 4) + 27 (2 d - 2) (2 d - 4) (2 d - 6) )(*c7(e1)*)

55 972
```

Bounds on two-point function

We compute the 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)

```
Do[
 Bound[G7i, s] = c7i z[s]^7 + (2 d * z[s])^9 * VarGamma2[s] * I110; (* G7,z(e1)*)
 Bound[G5i, s] = c5i z[s]^5 + Bound[G7i, s]; (* G5,z(e1)*)
 Bound[G3i, s] = c3i z[s]^3 + Bound[G5i, s]; (* G3,z(e1)*)
 Bound[G1i, s] = z[s] + Bound[G3i, s];(* G1,z(e1)*)
 , {s, {i, o}}]
```

Then, we compute $G_{n,z}(e_1 + e_2)$ and $G_{4,z}(2\,e_2)$, see (5.1.25)-(5.1.26) and (5.4.1):

```
Do[
 Bound[G8ik, s] = d/(d - 1) (2 d * z[s])^8 VarGamma2[s] I110; (*G8(e1+e2)*)
 Bound[G6ik, s] = c6ik z[s]^6 + VarGamma2[s] I18; (*G6(e1+e2)*)
 Bound[G4ik, s] = Bound[G6ik, s] + (c4ik - 2 (2 d - 3) ) z[s]^4; (*G4^1(e1+e2)*)
 Bound[G2ik, s] = Bound[G4ik, s] + (c2ik - 1) z[s]^2; (*G2^1(e1+e2)*)
 Bound[G4ii, s] = (2 d + 2) z[s]^4 + Bound[G6, s];(* Bound for sup_x G4^1(2 e1)*)
 , {s, {i, o}}]
```

We compute the supreme of the two-point function as given in (5.1.28)-(5.1.31):

```
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 G6(x)*)
 Bound[G4, s] = Max[c4ik z[s]^4, c5i z[s]^5] + Bound[G6, s]; (* Bound for sup_x G4(x)*)
 Bound[G2, s] = Max[c2ik z[s]^2, c3i z[s]^3] + Bound[G4, s]; (* Bound for sup_x G2(x)*)
 Bound[G1, s] = Max[Bound[G1i, s], Bound[G2, s]];(* Bound for sup_x G1(x)*)
 , {s, {i, o}}]
```

Bounds on repulsive bubbles

As next we bound the closed diagrams, see (5.1.37). Further, we use that $B_{\underline{1},3}(0) = 2\,d\,z\,G_{3,z}(e_1)$ and $B_{1,2}(0) = B_{\underline{1},3}(0) + B_{2,2}(0)$

```
Do[
 RepBubble[Step, G3, s] = 2 d z[s] Bound[G3i, s] ;
 RepBubble[G2, G2, s] = 2 d c3i z[s]^4 + 2 d 3 c5i z[s]^6 + 2 d 5 c7i z[s]^8 +
   (2 d z[s])^10 (4 VarGamma2[s] I110 + VarGamma2[s]^2 I210);
 RepBubble[G1, G2, s] = RepBubble[Step, G3, s] + RepBubble[G2, G2, s];
 , {s, {i, o}}]
```

We bound the open diagram $B_{2,0}(x)$ as shown in (5.1.38) and see that we can use the same bound for $B_{1,1}(x)$ and that $B_{0,1}(x) = B_{\underline{0},1}(x) + B_{1,1}(x)$:

```
Do[
 OpenRepBubble[G2, G0, s] = Max[c2ik z[o]^2 + 3 c4ik z[o]^4, 2 c3i z[o]^3 + 4 c5i z[o]^5] +
   4 (2 d z[o])^6 VarGamma2[o] I16 + (2 d z[o])^6 VarGamma2[o]^2 I26;
 OpenRepBubble[G1, G1, s] = OpenRepBubble[G2, G0, s];
 OpenRepBubble[G0, G1, s] = Bound[G1, s] + OpenRepBubble[G1, G1, s];
 , {s, {i, o}}]
```

Bounds on weighted diagrams

We bound $\sup_x G_z(x)[1 - \cos(k\,x)]$ as how in (5.1.19) and (5.1.20):

```
Bound[DeltaG, o] = ((2 c1) * I10 + (2 c2 + c4) I20) Gamma3 ;
Bound[DeltaG, i] = (2 d - 2)/(2 d - 1) (I20 + 4 I20) ;
```

Further, we bound $\Sigma_x\, G_{2,z}(x)^2\, G_z(x)[1 - \cos(k\,x)]$ as explained in Section 5.4.1 and given (5.4.7). For $z = z_I$ $(s = i)$ we use (5.1.20) and (5.4.7) with $c_1 = 0$, $c_3 = 0$, $c_2 = 0.5$:

```
Bound[G2CubedAndOneWeight, o] :=
  (2 d z[o])^4 VarGamma2[o]^2 Gamma3
   (2 c1 I12 + (2 c2 + c3) (I12 + I22) (I24 - I12^2) + (c1 I10 + (2 c2 + c3) I20) I12^2);
Bound[G2CubedAndOneWeight, i] :=
  (2 d z[i])^4 (2 (d - 1) / (2 d - 1))^3 ((I12 + I22) (I24 - I12^2) + I20 I12^2);
```

Bound on the coefficient Ξ_z^ι

Bound for N=0

We implement the bounds given in Lemma 4.2.1.:

```
Do[
 Bound[XiIota, 0, s] = Bound[G3i, s];
 Bound[XiIota, 0, Delta, 0, s] = 2 d Bound[G3i, s] ;
 Bound[XiIota, 0, Delta, ei, s] = 0;
 , {s, {i, o}}]
```

At the end of the analysis we use a lower bound on $\sum_x \Pi^{(0),i,k}(x)$. We create this lower bound by explicite computations and $z \geq z_I = 1/(2d-1)$.

```
Bound[PsiIota, 0, l] = (c3i - 1) z[i]^3 + (c5i - 4 (2 d - 2)) z[i]^5 +
   (c7i - c6ik - c4ik c3i - c5i) z[i]^7;
```

Bound for N=1

Here we implement the bounds given in Lemma 4.2.2.:

```
Do[
 Bound[XiIota, 1, s] =
  2 Bound[G3i, s]^2 + (2 d - 2) z[s] Bound[G4ik, s] (z[s]^2 + Bound[G2ik, s]) +
   z[s] Bound[G4ii, s]^2 + Bound[G2, s] RepBubble[G2, G2, s];
 Bound[XiIota, 1, Delta, ei, s] = 2 d Bound[G3i, s]^2 + Bound[G2CubedAndOneWeight, s] / z[i];
 Bound[XiIota, 1, Delta, 0, s] =
  2 Bound[XiIota, 1, Delta, ei, s] + 2 * 2 d * Bound[XiIota, 1, s];
 , {s, {i, o}}]
```

Bound for N≥2

Now we want to implement the bound stated in Propositions 4.2.3. We immediantly compute the sum of the bounds over odd and even N

Declaration of variables

We implement the vectors and matricies as given in (4.2.12)-(4.2.16):

```
Do[
 Vector[w1, s] = {Bound[G3i, s], RepBubble[G1, G2, s] / (2 d z[s])};
 Vector[w2, s] = {Bound[G3i, s], Bound[G2, s]};
 Vector[w3, s] = {RepBubble[Step, G3, s], RepBubble[G2, G2, s]};
 MatrixB[B, s] = {{Bound[G3i, s], RepBubble[G1, G2, s] / (2 d z[s])},
   {Bound[G2, s], OpenRepBubble[G2, G0, s]}};
 MatrixB[BPrime, s] = {{Bound[G3i, s], Bound[G1, s]},
   {RepBubble[G1, G2, s] / (2 d z[s]), OpenRepBubble[G2, G0, s]}}
 , {s, {i, o}}]
```

To compute the sum oven N we compute the eigensystem of B and $\overline{B}$ to the compute a decomposition of $w_1$and (1,1) as explained in Section 5.3:

```
Do[
 EigensystemsOfB[B, s] = Eigensystem[Transpose[MatrixB[B, s]]];
 EigensystemsOfB[BPrime, s] = Eigensystem[Transpose[MatrixB[BPrime, s]]];
 Vector[vOfw1, 1, s] = EigensystemsOfB[B, s][[2, 1]] *
   (Inverse[Transpose[EigensystemsOfB[B, s][[2]] ]].Vector[w1, s])[[1]];
 Vector[vOfw1, 2, s] = EigensystemsOfB[B, s][[2, 2]] *
   (Inverse[Transpose[EigensystemsOfB[B, s][[2]] ]].Vector[w1, s])[[2]];
 Vector[vOfone, 3, s] = EigensystemsOfB[BPrime, s][[2, 1]] *
   (Inverse[Transpose[EigensystemsOfB[BPrime, s][[2]] ]].{1, 1})[[1]];
 Vector[vOfone, 4, s] = EigensystemsOfB[BPrime, s][[2, 2]] *
   (Inverse[Transpose[EigensystemsOfB[BPrime, s][[2]] ]].{1, 1})[[2]];

 EigenValueB[1, s] = EigensystemsOfB[B, s][[1, 1]];
 EigenValueB[2, s] = EigensystemsOfB[B, s][[1, 2]];
 EigenValueB[3, s] = EigensystemsOfB[BPrime, s][[1, 1]];
 EigenValueB[4, s] = EigensystemsOfB[BPrime, s][[1, 2]];
 , {s, {i, o}}]
```

Bound on the absolute value (k=0)

We compute the sum of the bounds over all even and odd $N \geq 2$. These are stated in (5.4.10) and (5.4.11)

```
Do[
 Bound[XiIota, EvenTail, s] =
  Sum[(Vector[vOfw1, j, s].Vector[w2, s] * EigenValueB[j, s]) /
    (1 - EigenValueB[j, s]^2), {j, 1, 2}];
 Bound[XiIota, OddTail, s] =
  Sum[(Vector[vOfw1, j, s].Vector[w2, s] * EigenValueB[j, s]^2) /
    (1 - EigenValueB[j, s]^2), {j, 1, 2}];
 , {s, {i, o}}]
```

Bounds on Difference

Then we implement the bounds given in (5.4.12)-(5.4.15):

```
Do[
 Bound[K, s] = 2 d Bound[DeltaG, s];
 Bound[XiIota, EvenTail, Delta, ei, s] =
  Bound[K, s] Sum[{1, 1}.Vector[vOfw1, j, s] / (1 - EigenValueB[j, s]^2)^2, {j, 1, 2}]
    Sum[Vector[w3, s].Vector[vOfone, j, s] / (1 - EigenValueB[j, s]^2), {j, 3, 4}] +
   Bound[K, s] Sum[{1, 1}.Vector[vOfw1, j, s] / (1 - EigenValueB[j, s]^2), {j, 1, 2}]
    Sum[Vector[w3, s].Vector[vOfone, j, s] / (1 - EigenValueB[j, s]^2)^2 - Vector[w3, s].Vector[vOfone, j, s] / (1 - EigenValueB[j, s]^2),
     {j, 3, 4}];
 Bound[XiIota, EvenTail, Delta, 0, s] =
  2 d
    Sum[(Vector[w2, s].Vector[vOfw1, j, s] EigenValueB[j, s]) /
       (1 - EigenValueB[j, s]^2)^2 +
      (Vector[w2, s].Vector[vOfw1, j, s] EigenValueB[j, s]) / (1 - EigenValueB[j, s]^2),
     {j, 1, 2}] + Bound[K, s] Sum[{1, 1}.Vector[vOfw1, j, s] / (1 - EigenValueB[j, s]^2)^2, {j, 1, 2}]
```

```
    Sum[Vector[w3, s].Vector[vOfone, j, s] / (1 - EigenValueB[j, s]^2), {j, 3, 4}] +
   Bound[K, s] Sum[{1, 1}.Vector[vOfw1, j, s] / (1 - EigenValueB[j, s]^2), {j, 1, 2}]
    Sum[Vector[w3, s].Vector[vOfone, j, s] / (1 - EigenValueB[j, s]^2)^2, {j, 3, 4}];
 Bound[XiIota, OddTail, Delta, ei, s] =
  Bound[K, s] Sum[{1, 1}.Vector[vOfw1, j, s] / (1 - EigenValueB[j, s]^2)^2, {j, 1, 2}]
    Sum[(Vector[w3, s].Vector[vOfone, j, s] EigenValueB[j, s]) /
      (1 - EigenValueB[j, s]^2), {j, 3, 4}] +
   Bound[K, s] Sum[{1, 1}.Vector[vOfw1, j, s] / (1 - EigenValueB[j, s]^2), {j, 1, 2}]
    Sum[(Vector[w3, s].Vector[vOfone, j, s] EigenValueB[j, s]) /
      (1 - EigenValueB[j, s]^2)^2, {j, 3, 4}] +
   Bound[K, s]
    Sum[Vector[vOfw1, j, s] EigenValueB[j, s] / (1 - EigenValueB[j, s]^2)^2 + Vector[vOfw1, j, s] EigenValueB[j, s] / (1 - EigenValueB[j, s]^2),
      {j, 1, 2}].{1, 1} * OpenRepBubble[G2, G0, s];
 Bound[XiIota, OddTail, Delta, 0, s] =
  Bound[K, s] Sum[{1, 1}.Vector[vOfw1, j, s] EigenValueB[j, s] / (1 - EigenValueB[j, s]^2)^2, {j, 1, 2}]
    Sum[Vector[w3, s].Vector[vOfone, j, s] / (1 - EigenValueB[j, s]^2), {j, 3, 4}] +
   Bound[K, s] Sum[{1, 1}.Vector[vOfw1, j, s] EigenValueB[j, s] / (1 - EigenValueB[j, s]^2), {j, 1, 2}]
    Sum[Vector[w3, s].Vector[vOfone, j, s] / (1 - EigenValueB[j, s]^2)^2, {j, 3, 4}] +
   Bound[K, s]
    Sum[(Vector[w3, s].Vector[vOfone, j, s] EigenValueB[j, s]^2) /
       (1 - EigenValueB[j, s]^2)^2 +
      (Vector[w3, s].Vector[vOfone, j, s] EigenValueB[j, s]^2) /
       (1 - EigenValueB[j, s]^2), {j, 3, 4}];
 , {s, {i, o}}]
```

Sum over N

We combine the bounds computed in the preceding sections to compute β^{odd},β^{even},β^{abs}

```
Do[
 Bound[XiIota, Even, s] = Bound[XiIota, 0, s] + Bound[XiIota, EvenTail, s];
 Bound[XiIota, Odd, s] = Bound[XiIota, 1, s] + Bound[XiIota, OddTail, s];
 Bound[XiIota, Absolut, s] = Bound[XiIota, Odd, s] + Bound[XiIota, Even, s];
 , {s, {i, o}}]
```

Further, we compute

```
Do[
 Do[
  Bound[XiIota, Even, Delta, t, s] =
   Bound[XiIota, 0, Delta, t, s] + Bound[XiIota, EvenTail, Delta, t, s];
  Bound[XiIota, Odd, Delta, t, s] =
   Bound[XiIota, 1, Delta, t, s] + Bound[XiIota, OddTail, Delta, t, s];
  Bound[XiIota, Absolut, Delta, t, s] =
   Bound[XiIota, Odd, Delta, t, s] + Bound[XiIota, Even, Delta, t, s];
  , {t, {0, ei}}]
 , {s, {i, o}}]
```

Computation of constants of Proposition 3.3.1

In this section we compute the constances stated in Proposition 3.3.1, as described in Section 3.4.:

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

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

Bound on absolute value K_F and K_Φ

We compute the constances as given in Section 3.4.2:

```
Do[
 Bound[KF, s] = 2 d z[s] / (1 - z[s] - (2 d - 2) z[s] Bound[XiIota, Absolut, s]);
 Bound[KPhi, 2, s] = 1 + Bound[KF, s] Bound[XiIota, Absolut, s];
 Bound[KPhi, 1, s] = 1 - Bound[KF, s] Bound[XiIota, Absolut, s];
 Bound[KPhiabs, 2, s] = 1 + Bound[KF, s] Bound[XiIota, Absolut, s];
 Bound[KPhiabs, 1, s] = 1 - Bound[KF, s] Bound[XiIota, Absolut, s];
 Bound[KPhiWithoutZero, s] = Bound[KF, s] Bound[XiIota, Absolut, s];
 , {s, {i, o}}]
```

Bounds on differences

Here we implement the computation of Section 3.4.3.

First the differences of F_1 and Φ_1 given in lines (3.4.26), (3.4.27), (3.4.29)

```
Bound[DifferencefF, Part1, Lower, i] = (2 d z[i])/(1 - z[i]^2);
Bound[DifferencefF, Part1, Absolut, i] = (2 d z[i])/(1 - z[i]^2);
Bound[DifferencefF, Part1, Lower, o] = Min[(2 d z[o])/(1 - z[o]^2), (2 d z[i])/(1 - z[i]^2)];
Bound[DifferencefF, Part1, Absolut, o] = Max[(2 d z[o])/(1 - z[o]^2), (2 d z[i])/(1 - z[i]^2)];
Do[
 Bound[KDeltaPhi, Part1, s] =
   z[s]/(1 - z[s]^2) (Bound[XiIota, Absolut, Delta, ei, s] +
      z[s] Bound[XiIota, Absolut, Delta, 0, s]);
 , {s, {i, o}}]
```

Then the differences of F_2 and Φ_2: For the bound in (4.4.31), (4.4.32) and (4.4.35) we note that we can omit the first terms as Ξ is trivial and compute

```
Do[Bound[DifferencefF, Part2, Lower, s] =
   (-2 d z[s]^2)/((1 - z[s]^2)^2)
    ( Bound[XiIota, Even, Delta, ei, s] + 2 d Bound[XiIota, Even, s] +
      z[s]^2 Bound[XiIota, Even, Delta, 0, s] +
      z[s] (Bound[XiIota, Odd, Delta, ei, s] + 2 d Bound[XiIota, Odd, s] +
          Bound[XiIota, Odd, Delta, 0, s]));
 Bound[DifferencefF, Part2, Absolut, s] =
   (2 d z[s]^2)/(1 - z[s]^2) (Bound[XiIota, Absolut, Delta, ei, s] + 2 d Bound[XiIota, Absolut, s] +
      z[s] Bound[XiIota, Absolut, Delta, ei, s] +
      z[s] Bound[XiIota, Absolut, Delta, 0, s] + 2 d z[s] Bound[XiIota, Absolut, s] +
      z[s]^2 Bound[XiIota, Absolut, Delta, 0, s]);
 Bound[KDeltaPhi, Part2, s] =
   (2 d z[s]^2)/((1 - z[s])^2) 2 Bound[XiIota, Absolut, s] 1/(1 + z[s])
    (Bound[XiIota, Absolut, Delta, ei, s] + z[s] Bound[XiIota, 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).

```
Do[
 Bound[DifferencefF, OneOverOneMinuszPi, s] = 1 / (1 - (2 d z[s] Bound[XiIota, Absolut, s]) / (1 - z[s]));
 Bound[DifferencefF, Part3, Absolut, s] =
  Bound[XiIota, Absolut, s] z[s] / (1 - z[s]^2) (2 d z[s] / (1 - z[s]))^2
    (Bound[XiIota, Absolut, Delta, ei, s] + z[s] Bound[XiIota, Absolut, Delta, 0, s]) *
    Bound[DifferencefF, OneOverOneMinuszPi, s]^2 +
   z[s] / (1 - z[s]^2) (2 d z[s] / (1 - z[s]))^2 Bound[XiIota, Absolut, s]
    (Bound[XiIota, Absolut, Delta, ei, s] + 2 d Bound[XiIota, Absolut, s] +
      z[s] Bound[XiIota, Absolut, Delta, 0, s]) *
    Bound[DifferencefF, OneOverOneMinuszPi, s];

 Bound[DifferencefF, Part3, Lower, s] = -Bound[DifferencefF, Part3, Absolut, s];
 Bound[KDeltaPhi, Part3, s] =
  (2 d)^2 z[s]^3 / (1 - z[s])^2
   (Bound[XiIota, Absolut, Delta, ei, s] + z[s] Bound[XiIota, Absolut, Delta, 0, s])
   (Bound[DifferencefF, OneOverOneMinuszPi, s]^2 +
     Bound[DifferencefF, OneOverOneMinuszPi, 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];
 , {s, {i, o}}]
```

Check of the sufficient condition

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

```
NonBackLEf1Bound[i] = 1;
NonBackLEf1Bound[o] =
  1 + (2 d - 2) / (2 d - 1) Gamma1 (Bound[XiIota, Even, o] - Bound[PsiIota, 0, l]);

NonBackLEf2Bound[i] = (2 d - 1) / (2 d - 2) Bound[KPhiabs, 2, i] Bound[KDeltaFLower, i];
NonBackLEf2Bound[o] = (2 d - 1) / (2 d - 2) Bound[KPhiabs, 2, o] Bound[KDeltaFLower, o];
```

For f_3 we compute the four terms and then take the maximum

```
Do[
 NonBackLEf3Bound[Part1, s] =
  If[c1 ≤ 0, 1000, 1/(2 c1) Bound[KDeltaPhi, s] Bound[KDeltaFLower, s] ];
 NonBackLEf3Bound[Part2, s] =
  If[c2 ≤ 0, 1000, 1/(2 c2) Bound[KPhiabs, 2, s] Bound[KDeltaF, s]
    Bound[KDeltaFLower, s]^2];
 NonBackLEf3Bound[Part3, s] =
  If[c3 ≤ 0, 1000, (2 Bound[KDeltaFLower, s]^2)/c3
    √(Bound[KF, s] Bound[KPhiWithoutZero, s] Bound[KDeltaF, s] Bound[KDeltaPhi, s])];
 NonBackLEf3Bound[Part4, s] =
  If[c4 ≤ 0, 1000, (2 Bound[KDeltaFLower, s]^2)/c4
    (2 Bound[KDeltaF, s]^2 Bound[KPhiabs, 2, s] Bound[KDeltaFLower, s] +
      √(Bound[KF, s] Bound[KPhiWithoutZero, s] Bound[KDeltaF, s]
         Bound[KDeltaPhi, s]))];
 NonBackLEf3Bound[s] = Max[NonBackLEf3Bound[Part1, s], NonBackLEf3Bound[Part2, s],
   NonBackLEf3Bound[Part3, s], NonBackLEf3Bound[Part4, s]];
 , {s, {i, o}}]

Do[
 Succes[f1, s] = NonBackLEf1Bound[s] < Gamma1;
 Succes[f2, s] = NonBackLEf2Bound[s] < Gamma2;
 Succes[f3, s] = NonBackLEf3Bound[s] < Gamma3;
 Succes[s] = Succes[f1, s] && Succes[f2, s] && Succes[f3, s];
 , {s, {i, o}}]
Succes[overall] = Succes[i] && Succes[o];
```

Outputs

The overall result

The statement that the bootstrap was succesful is

```
Succes[overall]
```

```
True
```

If this succedes than the analysis of Section 3.3 can be used to proved mean-field behavoior for SAW. The denominator of in the intrared bound in (3.3.12) and (3.3.13) are given by:

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

```
0.981409
```

```
1.01869
```

Further, we have proven that z_c is smaller than

$$\frac{1}{2\,d-1}\,\text{Gamma1}$$

```
0.0669027
```

The improvement of bounds

```
bubbles = {Graphics[{Green, Disk[{0, 0}, 0.8]}, ImageSize -> 40],
   Graphics[{Red, Disk[{0, 0}, 0.8]}, ImageSize -> 40]};
MethodeFourTablePart3 =
 {{Quantity, F1, F2, F3Part1, F3Part2, F3Part3, F3Part4, F2 - init, F3 - init},
  {"To beat", Gamma1, Gamma2, Gamma3, Gamma3, Gamma3, Gamma3, Gamma2, Gamma3},
  {Computed, NonBackLEf1Bound[o], NonBackLEf2Bound[o], NonBackLEf3Bound[Part1, o],
   NonBackLEf3Bound[Part2, o], NonBackLEf3Bound[Part3, o],
   NonBackLEf3Bound[Part4, o], NonBackLEf2Bound[i], NonBackLEf3Bound[i]},
  {check, If[Succes[f1, o], bubbles[[1]], bubbles[[2]]],
   If[Succes[f2, o], bubbles[[1]], bubbles[[2]]],
   If[NonBackLEf3Bound[Part1, o] < Gamma3, bubbles[[1]], bubbles[[2]]],
   If[NonBackLEf3Bound[Part2, o] < Gamma3, bubbles[[1]], bubbles[[2]]],
   If[NonBackLEf3Bound[Part3, o] < Gamma3, bubbles[[1]], bubbles[[2]]],
   If[NonBackLEf3Bound[Part4, o] < Gamma3, bubbles[[1]], bubbles[[2]]],
   If[Succes[f2, i], bubbles[[1]], bubbles[[2]]],
   If[NonBackLEf3Bound[i] < Gamma3, bubbles[[1]], bubbles[[2]]]},
  {Required c_i, NonBackLEf1Bound[o], NonBackLEf2Bound[o],
   NonBackLEf3Bound[Part1, o] * c1 / Gamma3, NonBackLEf3Bound[Part2, o] * c2 / Gamma3,
   NonBackLEf3Bound[Part3, o] * c3 / Gamma3, NonBackLEf3Bound[Part4, o] * c4 / Gamma3},
  {given c_i , , , c1, c2, c3, c4}};
Labeled[Grid[MethodeFourTablePart3, 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
```

Bound on coefficients in Dimension 8

Quantity	**F1**	**F2**	**F3Part1**	**F3Part2**	**F3Part3**	**F3Part4**	**F2 − init**	**F3 − init**
To beat	1.00354	1.05151	1.15	1.15	1.15	1.15	1.05151	1.15
Computed	1.00229	1.04836	1.09186	1.1377	1.11162	1.12879	1.03023	1.03141
check								
Required c_i	1.00229	1.04836	0.141468	0.491389	0.106967	4.64767		
given c_i			0.149	0.4967	0.11066	4.735		

The table above shows whether the improvement of bound for the function f_i was succesfull. As the bound on f_3 is the hardest to improve we show all four parts of the maximum. This helps to choose proper value for c_i correctly me mark the required values in the table.

Additional output

```
MethodeFourTable = {{Quantity, Ξ^Zero, Ξ^One, Ξ^EvenTail, Ξ^OddTail},
   {Bound for , Bound[XiIota, 0, o], Bound[XiIota, 1, o], Bound[XiIota, EvenTail, o],
    Bound[XiIota, OddTail, o]},
   {Delta Zero , Bound[XiIota, 0, Delta, 0, o], Bound[XiIota, 1, Delta, 0, o],
    Bound[XiIota, EvenTail, Delta, 0, o], Bound[XiIota, OddTail, Delta, 0, o]},
   {Delta Iota , Bound[XiIota, 0, Delta, ei, o], Bound[XiIota, 1, Delta, ei, o],
    Bound[XiIota, EvenTail, Delta, ei, o], Bound[XiIota, OddTail, Delta, ei, o]}};
MethodeFourTablePart1 = {{Quantity, KF, KPhi, DELTAFLower, DELTAFAbsolut, DELTAPhi},
   {Bound for , Bound[KF, o], Bound[KPhi, 2, 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, 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 constants 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["contributions of the Delta-constants in Dimension " Text[d], Bold], Top] //
 Text
```

Bound on coefficients in Dimension 8

Quantity	Ξ^{Zero}	Ξ^{One}	$\Xi^{EvenTail}$	$\Xi^{OddTail}$
Bound for	0.00741295	0.000551476	0.0000397166	2.93268×10^{-6}
Delta Zero	0.118607	1.78071	0.413957	0.138824
Delta Iota	0	0.881532	0.208005	0.105241

Bound on constants in Dimension 8

Quantity	**KF**	**KPhi**	**DELTAFLower**	**DELTAFAbsolut**	**DELTAPhi**
Bound for	1.15649	1.00926	0.969487	1.19142	0.335616

contributions of the Delta−constants in Dimension 8

Quantity	**DELTA⋱ FLow⋱ er**	**2**	**3**	**DELTA⋱ FAbs⋱ olut**	**2**	**3**	**DELTA⋱ Phi**	**2**	**3**
Bound for	$\frac{15}{14}$	−0.0379⋱ 127	−0.0020⋱ 429⋱ 8	1.07526	0.114123	0.00204⋱ 298	0.09131⋱ 81	0.00167⋱ 763	0.242621