Model independent routines

Implementation of the model-independent part of the NoBLE by Robert Fitzner and Remco van der Hofstad

Robert Fitzner
Eindhoven University of Technology

Abstract

In the paper "Generalized approach to the non-backtracking lace expansion" by Remco van der Hofstad and the author of this Mathematica notebook, we describe a general analysis to prove mean-field behavior that several models show above their upper critical dimension. In Section 2 and 3 we work with a simplified rewrite that allows us to simplify the presentation of the analysis. In this file, we implement general functions to compute several elaborate bounds.

In the first part we translate the bounds on the NoBLE-coefficients, as stated in Assumption 4.3, to the bounds of the simplified rewrite, given in Assumption 2.7. This is derived in Appendix D.

In the second part, we implement the bounds on the bootstrap function f_3 derived in Section 3.3.

25.10.2016

Bounds on the simplified rewrite (Assumption 2.7)

Here we implement the bounds on the simplified rewrite given in Assumption 2.7 and derived in Appendix D. As input, we assume the model-dependent bounds given in Assumption 4.3.

In[1135]:=

```
(* Bounds of Step 1*)
betaMubarOverMu[MuOverMu_] := MuOverMu; (* (D.1)  *)
betaCPhiLow[d_, mu_, XiOneminusZeroAtZero_, XiIotaAlphaI_] :=
  1 - XiOneminusZeroAtZero - (2 d mu)/(1 - mu^2) XiIotaAlphaI;
betaCPhiUp[d_, mu_, XiZerominusOneAtZero_, XiIotaAlphaII_] :=
  1 + XiZerominusOneAtZero + (2 d mu^2)/(1 - mu^2) XiIotaAlphaII;
(* (D.2) *)
betaAfLow[d_, muMIN_, mu_, PsiOneminusZeroAlphaI_, PsiZerominusOneAlphaII_,
   PiSumAlpha_] :=
  (2 d muMIN)/(1 - muMIN^2) (1 - PsiOneminusZeroAlphaI - mu PsiZerominusOneAlphaII -
      1/(1 - mu^2) PiSumAlpha);
betaAfUp[d_, mu_, PsiZerominusOneAlphaI_, PsiOneminusZeroAlphaII_,
```

```
    PiSumAlphaLower_] :=
   (2 d mu)/(1 - mu^2) (1 + PsiZerominusOneAlphaI + mu PsiOneminusZeroAlphaII -
        1/(1 - mu^2) PiSumAlphaLower);
(* (D.3) *)
betaap[d_, mu_, XiOneminusZeroAlphaIei_, XiZerominusOneAlphaIei_,
    XiIotaAlphaIsumei_, XiIotaAlphaIIsumzero_] :=
   Max[2 d XiOneminusZeroAlphaIei + (2 d mu)/(1 - mu^2) XiIotaAlphaIsumei,
    2 d XiZerominusOneAlphaIei + (2 d mu^2)/(1 - mu^2) XiIotaAlphaIIsumzero];
(* (D.4)  *)
betaPiHat[d_, muPiToXiIota_, XiIotaEven_, PiLower_] :=
   2 d muPiToXiIota XiIotaEven - PiLower;
(* (D.5), be aware of the plus and minus signs of the used constant βΨ *)
betaPsiHatLower[d_, PsiToXi_, XiOdd_, PsiLower_] := PsiToXi XiOdd - PsiLower;
(* Bounds of Step 2*)
(* (D.13) *)
betaRF[d_, mu_, mubar_, PsiToXi_, muPiToXii_, XiAbs_, XigeqTwoAbs_,
    XiIotaAbs_, XiIotageqOneAbs_, PsiRI_, PsiRII_, PiR_] :=
   Module[{tmp1, firstLine, secondLine, thirdLine, fourthLine},
    tmp1 = (2 d muPiToXii)/(1 - mu) XiIotaAbs;

    firstLine = (2 d mu)/(1 - mu) (1 + PsiToXi XiAbs)/(1 - tmp1) tmp1^2;
    secondLine = (2 d)/(1 - mu^2) ( mu PsiRI + mu^2 PsiRII + mubar (1 + mu) XigeqTwoAbs);
    thirdLine = (2 d mu)/(1 - mu^2)^2 (PiR + 2 d muPiToXii XiIotageqOneAbs);
    fourthLine = ((2 d )^2 mu muPiToXii)/(1 - mu^2)^2 (2 + mu) XiIotaAbs +
      ((2 d)^2 muPiToXii mu)/(1 - mu)^2 XiAbs XiIotaAbs;
    firstLine + secondLine + thirdLine + fourthLine

   ];
(* (D.14) *)
betaRp[d_, mu_, muPsiToXi_, muPiToXiIota_, XiAbs_, XiR_, XigeqTwoAbs_,
    XiIotaAbs_, XiIotageqOneAbs_, XiIotaRI_, XiIotaRII_] :=
   Module[{tmp1, partOfXi, sumOverPhin, lastline},
    tmp1 = (2 d muPiToXiIota)/(1 - mu) XiIotaAbs;
    partOfXi = XiR + XigeqTwoAbs;
```

```
    sumOverPhin = (2 d mu XiIotaAbs)/(1 - mu) (1 + muPsiToXi XiAbs)/(1 - tmp1) tmp1;
    lastline = (2 d mu)/(1 - mu) muPsiToXi XiAbs XiIotaAbs +
      (2 d mu)/(1 - mu^2) (XiIotaRI + mu XiIotaRII + (1 + mu) XiIotageqOneAbs);
    partOfXi + sumOverPhin + lastline
   ];
(* Bounds of Step 3*)
(* (D.21)  *)
betaRpDelta[d_, mu_, PsiToXi_, muPiToXii_, XiAbs_, XiDeltaAbs_, XiDeltaR_,
    XiDeltageqTwoAbs_, XiIotaAbs_, XiIotaDeltaEiAbs_, XiIotaDeltaZeroAbs_,
    XiIotaDeltaEigeqOneAbs_, XiIotaDeltaZerogeqOneAbs_, XiIotaDeltaRI_,
    XiIotaDeltaRII_] :=
  Module[{tmp2, firstLine, secondLine, thirdLine, fourthLine},
    (* implementation of (3.78) *)
    tmp2 = (2 d muPiToXii)/(1 - mu) XiIotaAbs;
    firstLine = XiDeltaR + XiDeltageqTwoAbs;
    secondLine = (2 d mu)/(1 - mu) tmp2/(1 - tmp2) PsiToXi XiDeltaAbs XiIotaAbs +
      (2 d)/(1 - mu^2) tmp2 (1/(1 - tmp2)^2 + 1/(1 - tmp2)) (mu + mu PsiToXi XiAbs)/(1 + mu)
       (XiIotaDeltaEiAbs + mu XiIotaDeltaZeroAbs);
    (*ps: the second term is the biggest*)
    thirdLine =
     (2 d mu PsiToXi)/(1 - mu^2)
      ((1 + mu) XiDeltaAbs XiIotaAbs +
        XiAbs (XiIotaDeltaEiAbs + mu XiIotaDeltaZeroAbs));
    fourthLine =
     (2 d mu)/(1 - mu^2) (XiIotaDeltaRI + mu XiIotaDeltaRII + XiIotaDeltaEigeqOneAbs +
        mu XiIotaDeltaZerogeqOneAbs);
    firstLine + secondLine + thirdLine + fourthLine
   ];
(* Bounds of Step 4*)
(* (D.29)  *)
betaRfDelta[d_, mu_, PsiToXi_, muPiToXii_, XiAbs_, XiDeltaAbs_, XigeqTwoAbs_,
    XiDeltageqTwoAbs_, PsiRDeltaI_, PsiRDeltaII_, XiIotaAbs_,
    XiIotaDeltaEiAbs_, XiIotaDeltaZeroAbs_, XiIotageqOneAbs_,
    XiIotaDeltaEigeqOneAbs_, betaPiRDelta_] :=
  Module[{tmp2, lines, i},
    tmp2 = (2 d muPiToXii)/(1 - mu) XiIotaAbs;
    lines = Table[0, {i, 1, 7}];
```

```
    lines[[1]] = (2 d mu)/(1 - mu) PsiToXi XiDeltaAbs tmp2^2/(1 - tmp2);
    lines[[2]] = ((2 d)^2 mu muPiToXii)/((1 - mu^2) × (1 - mu)) tmp2/(1 - tmp2)^2 (1 + PsiToXi XiAbs)
      (XiIotaDeltaEiAbs + mu XiIotaDeltaZeroAbs);
    lines[[3]] = ((2 d)^2 mu muPiToXii)/((1 - mu^2) × (1 - mu)) tmp2/(1 - tmp2) (1 + PsiToXi XiAbs)
      (XiIotaDeltaEiAbs + mu XiIotaDeltaZeroAbs + XiIotaAbs);
    lines[[4]] =
     (2 d mu)/(1 - mu^2) (PsiRDeltaI + mu PsiRDeltaII + (1 + mu) PsiToXi XiDeltageqTwoAbs +
        PsiToXi XigeqTwoAbs);
    lines[[5]] =
     mu/(1 - mu^2)^2
      (betaPiRDelta + (2 d)^2 muPiToXii (XiIotaDeltaEigeqOneAbs + XiIotageqOneAbs ));
    lines[[6]] = ((2 d)^2 muPiToXii mu^2)/(1 - mu^2)^2
      (XiIotaDeltaEiAbs + (1 + mu) XiIotaDeltaZeroAbs + XiIotaAbs);
    lines[[7]] = ((2 d)^2 muPiToXii mu PsiToXi)/(1 - mu)^2 XiDeltaAbs XiIotaAbs +
      ((2 d)^2 muPiToXii mu PsiToXi)/((1 - mu^2) × (1 - mu)) XiAbs
       (XiIotaDeltaEiAbs + mu XiIotaDeltaZeroAbs + XiIotaAbs);
    Total[lines]
   ];
 (* Bounds of Step 5*)
 (* (D.32)  *)
 betaRfDeltaLower[d_, mu_, PsiToXi_, muPiToXii_, XiAbs_, XiOddAbs_,
    XiEvenAbs_, XiDeltaAbs_, XiDeltaOddAbs_, XiDeltaEvenAbs_, XigeqTwoOddAbs_,
    XiDeltageqTwoOddAbs_, XiDeltageqTwoEven_, PsiROneDeltaI_, PsiRZeroDeltaII_,
    XiIotaAbs_, XiIotaOddAbs_, XiIotaEvenAbs_, XiIotaDeltaEi_, XiIotaDeltaEiOdd_,
    XiIotaDeltaEiEven_, XiIotaDeltaZero_, XiIotaDeltaZeroOdd_,
    XiIotaDeltaZeroEven_, XiIotageqOneEven_, XiIotaDeltageqOneEven_,
    betaPiRDelta_] :=
   Module[{tmp2, lines, i},
    lines = Table[0, {i, 1, 9}];
    tmp2 = (2 d muPiToXii)/(1 - mu) XiIotaAbs;
    lines[[1]] = -(2 d mu)/(1 - mu) PsiToXi XiDeltaAbs tmp2^2/(1 - tmp2);
    lines[[2]] = -((2 d)^2 mu muPiToXii)/((1 - mu^2) × (1 - mu)) tmp2/(1 - tmp2)^2 (1 + PsiToXi XiAbs)
      (XiIotaDeltaEi + mu XiIotaDeltaZero);
```

```
lines[[3]] = - (2 d)^2 mu muPiToXii / ((1 - mu^2) × (1 - mu)) tmp2/(1 - tmp2) (1 + PsiToXi XiAbs)
   (XiIotaDeltaEi + mu XiIotaDeltaZero + XiIotaAbs);
lines[[4]] =
  - 2 d mu/(1 - mu^2) (PsiROneDeltaI + mu PsiRZeroDeltaII +
     PsiToXi (XiDeltageqTwoOddAbs + XigeqTwoOddAbs + mu XiDeltageqTwoEven));
lines[[5]] =
  - mu/(1 - mu^2)^2
   (betaPiRDelta + (2 d)^2 muPiToXii (XiIotaDeltageqOneEven + XiIotageqOneEven));
lines[[6]] = - (2 d)^2 muPiToXii mu^2/(1 - mu^2)^2
   (XiIotaDeltaEiOdd + XiIotaDeltaZeroOdd + XiIotaOddAbs + mu XiIotaDeltaZeroEven);
lines[[7]] = - (2 d)^2 muPiToXii mu/(1 - mu^2)^2
   (XiDeltaOddAbs XiIotaOddAbs (1 + mu^2) + 2 mu XiDeltaEvenAbs XiIotaEvenAbs);
lines[[8]] = - (2 d)^2 muPiToXii mu/(1 - mu^2)^2 XiOddAbs
   (XiIotaDeltaEiOdd + XiIotaOddAbs + mu XiIotaDeltaZeroEven +
     mu XiIotaEvenAbs + mu XiIotaDeltaEiEven + mu^2 XiIotaDeltaZeroOdd);
lines[[9]] = - (2 d)^2 muPiToXii mu/((1 - mu^2) × (1 - mu)) XiEvenAbs
   (XiIotaDeltaEiEven + XiIotaEvenAbs + mu XiIotaDeltaZeroOdd +
     mu XiIotaOddAbs + mu XiIotaDeltaEiOdd + mu^2 XiIotaDeltaZeroEven);

Total[lines]
];
```

Bound on the integral H_i

In this section we implement the bounds derived in Section 3.3.5. We bound (3.58) :

$$\int_{(-\pi,\pi)^d} \hat{H}_i(k)\, \hat{D}^l(k)\, \hat{G}_z^n(k)\, \hat{D}^{(x)}(k)\, \mathrm{dk},$$

with $\hat{H}_i(k)$ defined in (3.52)-(3.56).

In[1147]:=

```
BoundH[1, d_, n_, l_, v_, Gamma2dash_, cp_, afmin_, afmax_, ap_, bRf_, bRp_,
   bRfDelta_, bRpDelta_, Kunderline_] :=
  If[n == 0, (* (3.61)  *) cp IM[0, l, v] + ap IM[0, l + 1, v] + ap/afmin IM[-1, l, v]
   (*IM[-1,l,v] is defined accordingly *), If[n == 1, (* (3.64) *)
    cp^2/afmin IM[1, l, v] + cp ap/afmin IM[0, l, v] + 2 ap cp/afmin IM[1, l + 1, v] +
     ap^2/afmin IM[0, l + 1, v] + ap^2/afmin IM[1, l + 2, v] +
```

```
        (bRp + bRfDelta Gamma2dash)
        --------------------------- (cp T[3, l, v] + ap T[3, l + 1, v] + ap T[2, l, v]),
                  afmin^2
      If[n == 2, (* (3.68)+(3.70) *)
        cp^3                     cp^2 ap                      ap cp^2
        -------- IM[2, l, v] +  -------- IM[1, l, v] + 3  -------- IM[2, l + 1, v] +
        afmin^2                  afmin^2                      afmin^2
         2 cp^2 ap                         ap^2 cp                          ap^3
         --------- IM[1, l + 1, v] + 3  -------- IM[2, l + 2, v] +  -------- IM[1, l + 2, v] +
         afmin^2                           afmin^2                          afmin^2
         ap^3                     (bRp + bRfDelta Gamma2dash)   (  cp              )
         -------- IM[2, l + 3, v] + --------------------------- ( ----- + Gamma2dash )
         afmin^2                            afmin^2              ( afmin             )
          (cp T[4, l, v] + ap T[4, l + 1, v] + ap T[3, l, v]) +
            (bRp + bRfDelta Gamma2dash)
         ap ---------------------------
                    afmin^3
          ( cp T[4, l + 1, v] + ap T[4, l + 2, v] + ap T[3, l + 1, v]), -1]]];
(* (3.74) *)
BoundH[2, d_, n_, l_, v_, Gamma2dash_, cp_, afmin_, afmax_, ap_, bRf_, bRp_,
    bRfDelta_, bRpDelta_, Kunderline_] :=
  bRfDelta Gamma2dash^n Kunderline
      (                                              (   1               )
      ((cp T[n + 2, l, v] + ap T[n + 2, l + 1, v]) ( ----- + Kunderline ) +
      (                                              ( afmin             )
        ap
       ----- T[n + 1, l, v]) + afmax bRpDelta Kunderline^2 T[n + 2, l, v];
       afmin
(* (3.77) *)
BoundH[3, d_, n_, l_, v_, Gamma2dash_, cp_, afmin_, afmax_, ap_, bRf_, bRp_,
    bRfDelta_, bRpDelta_, Kunderline_] :=
  2 Kunderline^2 Gamma2dash^n ap ( bRfDelta
                                 ( -------- + Max[Abs[afmin - 1], Abs[afmax - 1]] )
                                 (  afmin                                         )
    U[n + 2, l, v] + 2 Kunderline^2 Gamma2dash^(n+1)
    (afmax Max[Abs[afmin - 1], Abs[afmax - 1]] + bRfDelta) U[n + 3, l, v];
(* (3.78) *)
BoundH[4, d_, n_, l_, v_, Gamma2dash_, cp_, afmin_, af_, ap_, bRf_, bRp_,
    bRfDelta_, bRpDelta_, Kunderline_] :=
  Kunderline (bRpDelta K[n, l, v] + bRfDelta Gamma2dash K[n + 1, l, v]);
(* (3.86) *)
BoundH[5, d_, n_, l_, v_, Gamma2dash_, cp_, afmin_, af_, ap_, bRf_, bRp_,
    bRfDelta_, bRpDelta_, Kunderline_] :=
  2 Kunderline^2 Gamma2dash^(n+1) (2 af bRfDelta + bRfDelta^2) U[n + 3, l, v] +
   2 Kunderline^2 Gamma2dash^n (af bRpDelta + (ap + bRpDelta) bRfDelta) U[n + 2, l, v];
BoundH[d_, n_, l_, v_, Gamma2dash_, cp_, afmin_, af_, ap_, bRf_, bRp_,
    bRfDelta_, bRpDelta_, Kunderline_] :=
  Sum[BoundH[i, d, n, l, v, Gamma2dash, cp, afmin, af, ap, bRf, bRp, bRfDelta,
     bRpDelta, Kunderline], {i, 1, 5}];
```

Improvements

We compute the bound on f_3 for $z \in (z_I, z_c)$ as explained in Section 3.3.3. As input we require the bounds of Assumption 2.7.

In[1153]:=
```
BoundFThreeInital[d_, n_, l_, rho_, vecs_] := Module[{v2, i},
   v2 = Table[rho ((2 d - 2)/(2 d - 1))^(n+1) IM[n, l, vecs[[i]] ], {i, 1, Length[vecs]}];
   Max[v2]
  ];
BoundFThree[d_, n_, l_, vecs_, Gamma2dash_, cp_, afmin_, afmax_, ap_, bRf_,
   bRp_, bRfDelta_, bRpDelta_, Kunderline_] := Module[{v2, i},
   v2 = Table[BoundH[d, n, l, vecs[[i]], Gamma2dash, cp, afmin, afmax, ap,
      bRf, bRp, bRfDelta, bRpDelta, Kunderline], {i, 1, Length[vecs]}];
   Max[v2]
  ];
```