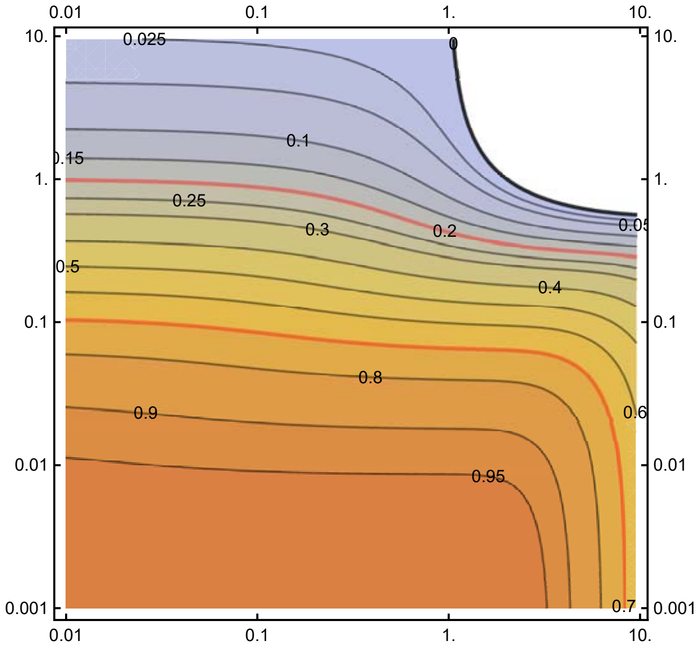
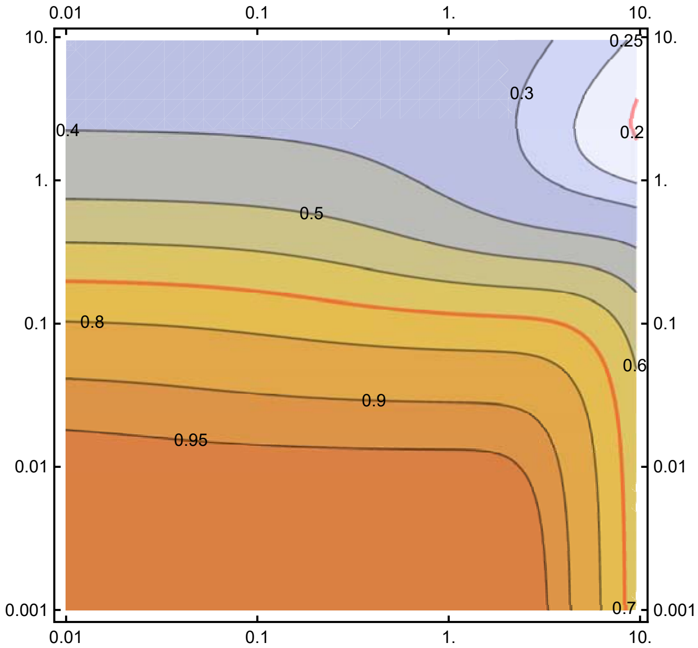
```
In[6]:= (* P(G1) and P(G1a),  under the IM model for 3 sequences (a1, a2, b) for heuristic species delimitation.
           The notation is from Kornai et al. (2024).
            PG1a works if M>0 only.  PG1aInt works if M is specified with high precision,
           such as M = 10`100.
         *)
        θ2 = 0.001; τ = 0.005;
        Q[f_, M_] := Block[{θ1, θ2, c1, c2, M21, w21},
           θ1 = f * θ2; M21 = M;
           c1 = 2 / θ1; c2 = 2 / θ2; w21 = 4 M21 / θ1;

           {{-(2*w21 + c1),         w21,        w21,   0, c1, 0,  0,  0,  0,  0,  0},
             {            0, -(w21 + c2),           0, w21,  0, c2,  0,  0,  0,  0,  0},
             {            0,           0, -(w21 + c2), w21,  0,  0, c2,  0,  0,  0,  0},
             {0, 0, 0, -3 * c2, 0, 0, 0, c2, c2, c2, 0},
             {0, 0, 0, 0, -w21, 0, 0, w21, 0, 0, 0},
             {0, 0, 0, 0, 0, -w21, 0, 0, w21, 0, 0},
             {0, 0, 0, 0, 0, 0, -w21, 0, 0, w21, 0},
             {0, 0, 0, 0, 0, 0, 0, -c2, 0, 0, c2},
             {0, 0, 0, 0, 0, 0, 0, 0, -c2, 0, c2},
             {0, 0, 0, 0, 0, 0, 0, 0, 0, -c2, c2},
             {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
           }
          ];
        PG1a[f_, M_] := Block[{ Esys, U, Evalues, P, θ1, θ2, τ},
           θ1 = f * θ2;
           Esys = Eigensystem[Q[f, M]];
           Evalues = Esys[[1]];
           If[Abs[Evalues[[11]] ] > 10^(-20), Print["eigenvalue 11 is not 0?"]];
           Evalues[[11]] = -1;  (* eigenvalues are ordered increasingly, last one is 0. *)
           Evalues = (Exp[Evalues * τ] - 1) / Evalues;
           Evalues[[11]] = τ;
           U = Transpose[Esys[[2]]];
           P = U . DiagonalMatrix[Evalues] . Inverse[U];
           P[[1, 1]] * 2 / θ1 + P[[1, 4]] * 2 / θ2
          ];
        PG1b[f_, M_] := Block[{P, τ},
           P = MatrixExp[Q[f, M] * τ];
           (P[[1, 1]] + P[[1, 2]] + P[[1, 3]] + P[[1, 4]]) / 3
          ];
        PG1[f_, M_] := PG1a[f, M] + PG1b[f, M];

In[11]:= b = 10; Mmin = 0.001; Mmax = 9.5; fmin = 0.01; fmax = 9.5;
        newStyle[x_] := x /. l_Line :> Sequence[Opacity[.4], Thick, Red, l]
        newStyle2[x_] := x /. l_Line :> Sequence[Opacity[.8], Thick, Black, l]
        ContourPlot[(PG1[b^f, b^M] - 1 / 3) * 3 / 2,
            {f, Log[b, fmin], Log[b, fmax]}, {M, Log[b, Mmin], Log[b, Mmax]},
            Contours → {0, 0.025, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95},
            (* ContourStyle→{{Red, Thin}}, *)
            BaseStyle → {FontFamily → "Arial", FontSize → 9}, PlotPoints → 30, ContourLabels → All,
            ColorFunction → (ColorData[{"BeachColors", "Reverse"}]), ContourStyle → Thin, AspectRatio → 1,
            (* Frame→False, *)
            FrameTicks → {Table[{f, ToString[Round[b^f, fmin]]}, {f, Log[b, fmin], Log[b, 100]}],
               Table[{M, ToString[Round[b^M, Mmin]]}, {M, Log[b, Mmin], Log[b, 50]}]}
          ] /. Tooltip[x_, 0.2] :> Tooltip[newStyle[x], 0.2] /.
         Tooltip[x_, 0.7] :> Tooltip[newStyle[x], 0.7] /. Tooltip[x_, 0] :> Tooltip[newStyle2[x], 0]

        ContourPlot[PG1a[b^f, b^M], {f, Log[b, fmin], Log[b, fmax]}, {M, Log[b, Mmin], Log[b, Mmax]},
           Contours → {0, 0.025, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95},
           BaseStyle → {FontFamily → "Arial", FontSize → 9}, PlotPoints → 30, ContourLabels → All,
           ColorFunction → (ColorData[{"BeachColors", "Reverse"}]), ContourStyle → Thin, AspectRatio → 1,
           (* Frame→False, *)
           FrameTicks → {Table[{f, ToString[Round[b^f, fmin]]}, {f, Log[b, fmin], Log[b, 50]}],
              Table[{M, ToString[Round[b^M, Mmin]]}, {M, Log[b, Mmin], Log[b, 10]}]}
         ] /. Tooltip[x_, 0.2] :> Tooltip[newStyle[x], 0.2] /.
         Tooltip[x_, 0.7] :> Tooltip[newStyle[x], 0.7]
```

Out[14]=


0.01
0.1
1.
10.
0.025
0
0.1
0.15
0.25
0.3
0.2
0.05
0.5
0.4
0.8
0.9
0.6
0.95
0.7
0.001
0.01
0.1
1.
10.

Out[15]=


0.01
0.1
1.
10.
0.25
0.3
0.4
0.2
0.5
0.8
0.6
0.9
0.95
0.7
0.001
0.01
0.1
1.
10.