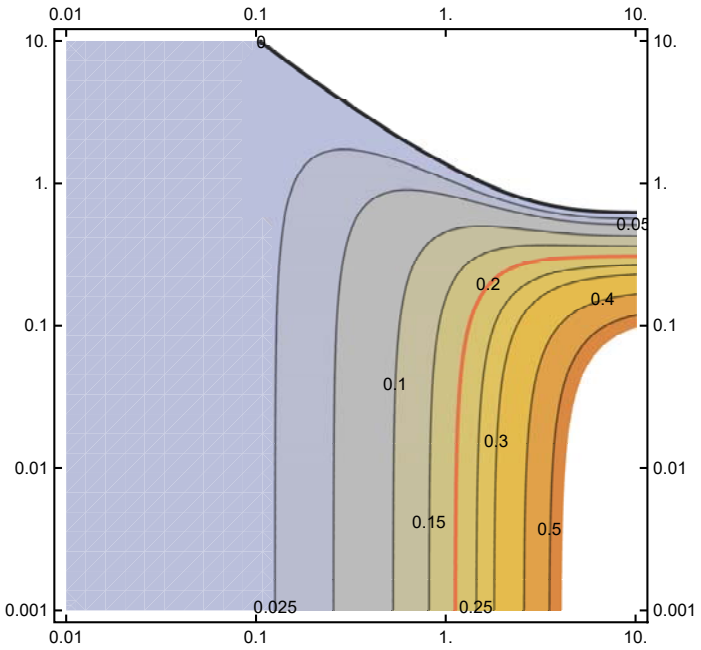
In[9]:=
```
θ2 = 2;
Q[M_] := Block[{θ1, θ2, c1, c2, M21, w21},
   θ1 = 5 * θ2;  M21 = M;
   c1 = 2 / θ1; c2 = 2 / θ2;  w21 = 4 M21 / θ1;

   {{-(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[τ_, M_] := Block[{ Esys, U, Evalues, P, θ1, θ2},
   θ1 = 5 * θ2;
   Esys = Eigensystem[Q[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[τ_, M_] := Block[{P},
  P = MatrixExp[Q[M] * τ];
  (P[[1, 1]] + P[[1, 2]] + P[[1, 3]] + P[[1, 4]]) / 3
 ]
PG1[τ_, M_] := PG1a[τ, M] + PG1b[τ, M];
```

In[14]:=
```
b = 10; Mmin = 0.001; Mmax = 10; τmin = 0.01; τmax = 10;
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^τ, b^M] - 1/3) * 3/2,
     {τ, Log[b, τmin], Log[b, τmax]}, {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[{τ, ToString[Round[b^τ, τmin]]}, {τ, Log[b, τmin], Log[b, τmax]}],
       Table[{M, ToString[Round[b^M, Mmin]]}, {M, Log[b, Mmin], Log[b, Mmax]}]}
   ] /. 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^τ, b^M], {τ, Log[b, τmin], Log[b, τmax]}, {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[{τ, ToString[Round[b^τ, τmin]]}, {τ, Log[b, τmin], Log[b, τmax]}],
      Table[{M, ToString[Round[b^M, Mmin]]}, {M, Log[b, Mmin], Log[b, Mmax]}]}
  ] /. Tooltip[x_, 0.2] :> Tooltip[newStyle[x], 0.2] /.
 Tooltip[x_, 0.7] :> Tooltip[newStyle[x], 0.7]
```

Out[17]=



Out[18]=