Forecasting risk under temporal autocorrelation

Code for forecasting long-term extinction risk under different levels of temporal autocorrelation given an organism's thermal performance curve (TPC), compared to weak and strong persistence boundaries derived in Vasseur et al. (2025). This notebook contains all the code used to generate the simulations and create Fig. 4; for easy replication of that figure, download the required MX files from the repository beforehand (SDE10000_gamma0, 1, and 2.m; windowing_r100_l10000_white, pink, and brown.m). Subfigures from this notebook were combined into one figure using Adobe Illustrator 2024.

Background Functions

Initialize functions for spectral synthesis, which generates a series with a specified level of autocorrelation 'γ', length 'Nobs', mean 'μ', and standard deviation 'σ' using an inverse Fourier transformation.

```
In[•]:= cmplx[mod_, arg_] := ExpToTrig[mod Exp[I arg]];

SpecSynFourier[γ_, Nobs_, μ_ : 0, σ_ : 1, seed_ : 0] := Module[{
    phase, f, vec},
   If[seed == 0, SeedRandom[], SeedRandom[seed]];
   phase = RandomReal[{0, 2 Pi}, Nobs];
   f = Range[1 / Nobs, 1, 1 / Nobs];
   vec = InverseFourier[
     Join[{0 + 0 I}, Table[cmplx[1 / (f〚i〛 ^ (γ / 2)), phase〚i〛], {i, 1, (Nobs - 1)}]],
     FourierParameters → {-1, 1}];
   Standardize[Re[vec]] * σ + μ];
```

Initialize the 'Lactin2' TPC (parameterized for P. caudatum; Lactin et al, 1995) and calculate the statistical moments, which are used to generate the persistence boundaries (Vasseur et al. 2025). Edit the 'directory' line before any of the rest of the code to specify the location simulations should be exported to and/or imported from.

```
directory = "/directory/";

lactin2[T_, {a_, b_, tmax_, δT_}] := Exp[a T] - Exp[a tmax - ((tmax - T) / δT)] + b;
paramsfit = {0.044, -1.774, 35.254, 5.435};

w[T_] := lactin2[T, paramsfit];

divisions = 40;
σTrange = Range[0.01, 8.01, 8 / divisions];
μTrange = Range[10, 35, 25 / divisions];
Clear[T];
moments = Flatten[ParallelTable[Module[{mean, var, skew, kurt},
     mean = NExpectation[w[T], T ≈ NormalDistribution[μT, σT]];
     var = NExpectation[(w[T] - mean) ^2, T ≈ NormalDistribution[μT, σT]];
     skew =
      NExpectation[((w[T] - mean)) ^3, T ≈ NormalDistribution[μT, σT]] / var ^ (3 / 2);
     kurt =
      NExpectation[(w[T] - mean) ^4, T ≈ NormalDistribution[μT, σT]] / (var ^2) - 3;
     {μT, σT, mean, var, skew, kurt, If[mean > 0, Log10[var / mean], 10]}],
    {σT, σTrange}, {μT, μTrange}], 1];
```

SDE Model

Runs simulations and stores extinction outcomes. ‘EMproc2’ is Equ. 2 and ‘SpecSynFourier’ generates a temperature time series with the specified parameters. To run simulations shown in paper, use parameters in parentheses after setting γ to the desired spectral exponent (best done on a computing cluster, these take a long time!); if you have already downloaded the MX files, skip to the next block of code.

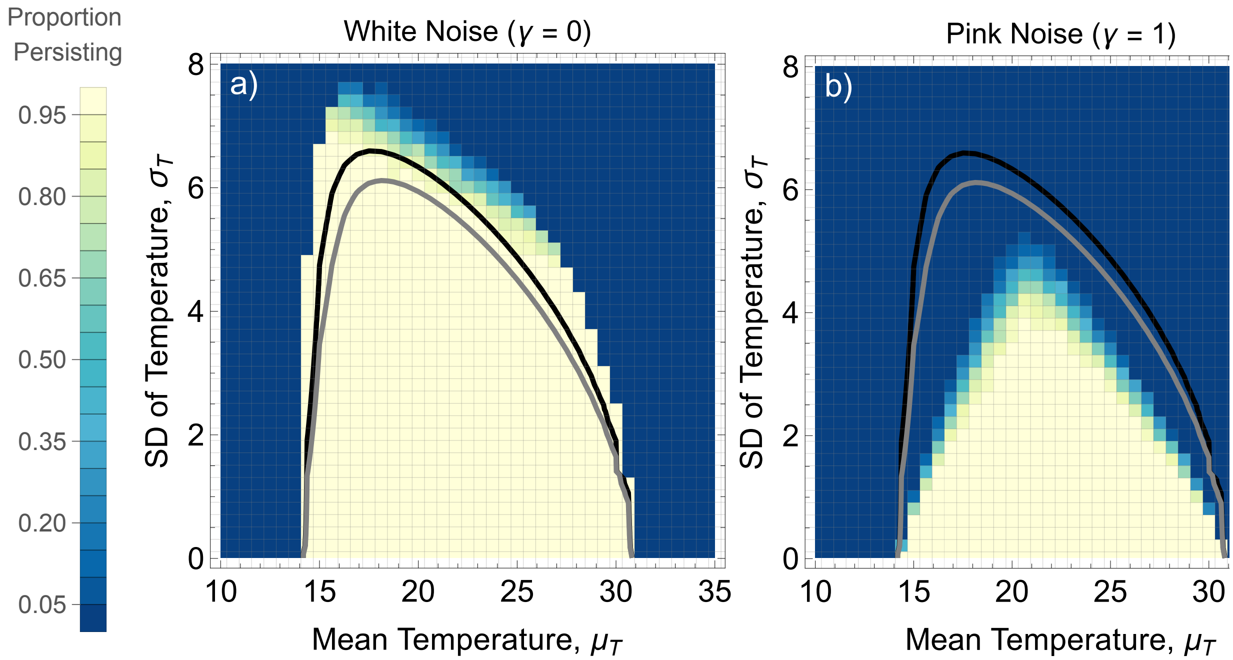
```
EMproc2[n_, T_] := n + (w[T] n - α n^2) δt;
tmax = 10 000; (*10000*)
δt = 0.01; (*.01*)
α = 0.0001;(*0.0001*)
reps = 100; (*100*)
γ = 2; (*set autocorrelation level here*)

output = Flatten[
   ParallelTable[
    Module[{
      extinct = 0},
     Do[
      p = NExpectation[w[x], x ≈ NormalDistribution[μT, σT]] / α;
      T = SpecSynFourier[γ, (tmax / δt), μT, σT];
      Do[p = EMproc2[p, T[[i]]];
       If[p < 1, extinct++; Break[]], {i, 1, tmax / δt}];, {reps}];

     {μT, σT, 1 - extinct / reps}]
    , {σT, σTrange}, {μT, μTrange}], 1];

Export[directory <> "SDE" <> ToString[tmax] <>
  "_gamma" <> ToString[γ] <> ".m", output, "MX"]
```

Download the needed MX files and generate plots shown in Fig. 4a-c.

```
In[∘]:= SDEwhite = Import[directory <> "SDE10000_gamma0.m", "MX"];
SDEpink = Import[directory <> "SDE10000_gamma1.m", "MX"];
SDEbrown = Import[directory <> "SDE10000_gamma2.m", "MX"];

newmap[x_] := Blend[{RGBColor["#ffffd9"], RGBColor["#edf8b1"], RGBColor["#c7e9b4"],
    RGBColor["#7fcdbb"], RGBColor["#41b6c4"], RGBColor["#4eb3d3"],
    RGBColor["#2b8cbe"], RGBColor["#0868ac"], RGBColor["#084081"]}, 1 - x];

SDEwhiteplot = Show[ListContourPlot[SDEwhite, InterpolationOrder → 0, Contours → 19,
    ColorFunction → newmap, PlotLegends → Placed[BarLegend[Automatic, LegendLabel →
        "Proportion\n Persisting", LabelStyle → {Darker[Gray], 13}], Left],
    PlotLabel → Style["White Noise (γ = 0)", 15], ImageSize → 310,
    Frame → True, FrameStyle → 16,
    FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
    GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
    GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
    Epilog → {{Text[Style["a)", White, 17], {11.2, 7.7}]}}],
   ListContourPlot[moments[[1 ;;, {1, 2, 7}]], InterpolationOrder → 3,
    Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
```

```
      {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]];
SDEpinkplot = Show[ListContourPlot[SDEpink, InterpolationOrder → 0, Contours → 19,
    ColorFunction → newmap, PlotLabel → Style["Pink Noise (γ = 1)", 15],
    ImageSize → 310, Frame → True, FrameStyle → 16,
    FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
    GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
    GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
    Epilog → {{Text[Style["b)", White, 17], {11.2, 7.7}]}}],
   ListContourPlot[moments[[1 ;;, {1, 2, 7}]], InterpolationOrder → 3,
    Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
      {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]];
SDEbrownplot = Show[ListContourPlot[SDEbrown, InterpolationOrder → 0, Contours → 19,
    ColorFunction → newmap, PlotLabel → Style["Brown Noise (γ = 2)", 15],
    ImageSize → 310, Frame → True, FrameStyle → 16,
    FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
    GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
    GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
    Epilog → {{Text[Style["c)", White, 17], {11.2, 7.7}]}}],
   ListContourPlot[moments[[1 ;;, {1, 2, 7}]], InterpolationOrder → 3,
    Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
      {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]];

SDEs = GraphicsRow[
  {SDEwhiteplot, SDEpinkplot, SDEbrownplot}, Spacings → 0, ImageSize → 1000]

Export[directory <> "SDEs.pdf", SDEs, ImageResolution → 1000];
```

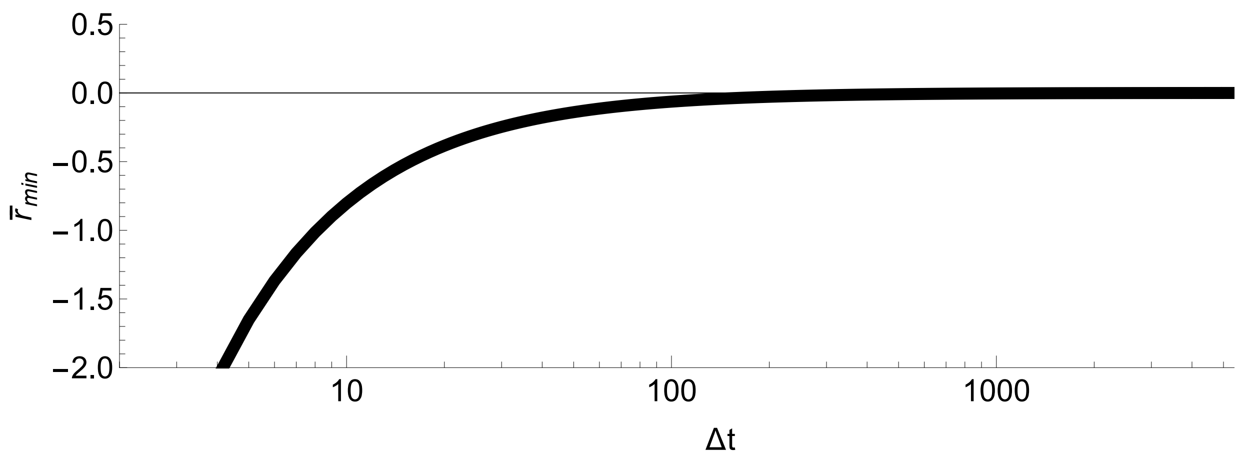
Out[ ]=



Running Mean

Use this code to calculate the running mean and generate Fig 4d. ‘minmean’ finds the lowest average value for each window length across a sequence (the running mean). ‘ExtTime’ calculates the number of time steps a population at carrying capacity can survive at any given negative growth rate.

```
In[∘]:= minmean[x_] := Module[{n = Length[x]}, ParallelTable[
        {j + 1, Min[Table[Mean[x[[i ;; i + j]]], {i, 1, n - j}]]}, {j, 1, n - 1}]];

      ExtTime[t_, α_, N0_] :=
       r /. NSolve[1 == N0 r / α Exp[r t] / (((r / α) - N0) + N0 Exp[r t]), r, Reals][[1]];
      N0 = 5000;
      α = 0.0001;
      extlimit = ListLogLinearPlot[ParallelTable[{t, ExtTime[t, α, N0]}, {t, 2, 9997}],
        ImageSize → 700, AspectRatio → 1 / 3.5, Frame → {True, True, False, False},
        FrameStyle → 16, FrameLabel → {"Δt", "r̄min"}, PlotStyle → {Thickness[0.01], Black},
        PlotRange → {{2, 10 000}, {-2, .5}}, Joined → True]
Out[∘]=
```



Calculate the running mean; skip to the next block of code if downloading existing files.

```
reps = 100; (*100*)
length = 10 000; (*10000*)
μT = 25; (*25*)
σT = 4; (*4*)

whitereps = Table[minmean[w[SpecSynFourier[0, length, μT, σT]]], {reps}];
pinkreps = Table[minmean[w[SpecSynFourier[1, length, μT, σT]]], {reps}];
brownreps = Table[minmean[w[SpecSynFourier[2, length, μT, σT]]], {reps}];

Export[directory <> "windowing_r" <> ToString[reps] <>
  "_l" <> ToString[length] <> "_white.m", whitereps, "MX"]
Export[directory <> "windowing_r" <> ToString[reps] <>
  "_l" <> ToString[length] <> "_pink.m", pinkreps, "MX"]
Export[directory <> "windowing_r" <> ToString[reps] <>
  "_l" <> ToString[length] <> "_brown.m", brownreps, "MX"]
```

Import the MX files for the running mean.

```
whitereps = Import[directory <> "windowing25,3.5_r100_l10000_white.m", "MX"];
pinkreps = Import[directory <> "windowing25,3.5_r100_l10000_pink.m", "MX"];
brownreps = Import[directory <> "windowing25,3.5_r100_l10000_brown.m", "MX"];

length = 10 000;
meanwhite = Table[Mean[whitereps[[ ;; , i]]], {i, 1, length - 1}];
meanpink = Table[Mean[pinkreps[[ ;; , i]]], {i, 1, length - 1}];
meanbrown = Table[Mean[brownreps[[ ;; , i]]], {i, 1, length - 1}];
```

Plot Fig. 4d. Note that due to the complexity of that image, it is not shown here in order to reduce file size (just remove the final ‘;’ to show it).

```
In[◦]:= opac = 0.05;
whitecol = RGBColor["#70ecff"];
pinkcol = RGBColor["#fe1910"];
browncol = RGBColor["#00555a"];

windowing = Show[
   ListLogLinearPlot[brownreps, ImageSize → 700,
    AspectRatio → 1 / 3.5, PlotStyle → {{browncol, Directive[Opacity[opac]]}},
    Frame → {True, True, False, False}, FrameStyle → 16,
    FrameLabel → {"Δt", "r̄min"}, PlotRange → {{2, 10 000}, {-2, .5}},
    Joined → True, Epilog → {Text[Style["d)", Black, 16], {.9, 0.42}]}],
   ListLogLinearPlot[pinkreps, PlotStyle → {{pinkcol, Directive[Opacity[opac]]}},
    Joined → True, PlotRange → Full], ListLogLinearPlot[whitereps, PlotStyle →
     {{whitecol, Directive[Opacity[opac]]}}, Joined → True, PlotRange → Full],
   extlimit,
   ListLogLinearPlot[{meanwhite, meanpink, meanbrown},
    PlotStyle → {{Thickness[0.005], whitecol}, {Thickness[0.005], pinkcol},
      {Thickness[0.005], browncol}}, PlotRange → Full, Joined → True,
    PerformanceGoal → Accuracy, PlotLegends → Placed[{"White Noise (γ = 0)",
       "Pink Noise (γ = 1)", "Brown Noise (γ = 2)"}, {Right, Bottom}],
    Epilog → {Text[Style["f)", Black, 18], Scaled[{.5, .5}]]}]];

Export[directory <> "windowing_r100_l1000_RGB.pdf",
  windowing, ImageResolution → 1000];
```

Supplemental Figures

Generate supplemental SDE figures (S2-3: propagating TPC uncertainty; and S4: SDE outputs at every different autocorrelation level).

```
(*run this code to propagate TPC uncertainty throughout SDE sims. this is
  quite slow - if you have already downloaded the SDEbootstrap MX files,
skip to the next block of code*)
(*import TPC data and fit lactin2 model*)
data = Import[directory <> "TPC Data.xlsx"][[1, 2 ;;]][[1 ;; 72, {3, 8}]];
lactin2[T_, {a_, b_, tmax_, δT_}] := Exp[a T] - Exp[a tmax - ((tmax - T) / δT)] + b;
nlm = NonlinearModelFit[data, {Exp[a T] - Exp[a tmax - ((tmax - T) / δT)] + b,
    a > 0, tmax > 20, δT > 1}, {a, b, tmax, δT}, T, Method → "NMinimize"];
paramsfit = {a, b, tmax, δT} /. nlm["BestFitParameters"];

(*measurement temperatures*)
pt18 = Select[data, #[[1]] == 18 &];
pt22 = Select[data, #[[1]] == 22 &];
pt24 = Select[data, #[[1]] == 24 &];
pt26 = Select[data, #[[1]] == 26 &];
pt28 = Select[data, #[[1]] == 28 &];
pt30 = Select[data, #[[1]] == 30 &];

(*bootstrap parameter choices x1000*)
reps = 1000;
bootparams = ParallelTable[
   Module[{resampled = Join[RandomChoice[pt18, 12], RandomChoice[pt22, 12],
        RandomChoice[pt24, 12], RandomChoice[pt26, 12],
        RandomChoice[pt28, 12], RandomChoice[pt30, 12]], nlm},
    nlm = NonlinearModelFit[resampled, {Exp[a T] - Exp[a tmax - ((tmax - T) / δT)] + b,
        a > 0, tmax > 20, δT > 1}, {a, b, tmax, δT}, T, Method → "NMinimize"];
    {a, b, tmax, δT} /. nlm["BestFitParameters"]], {reps}];

divisions = 40;
σTrange = Range[0.01, 8.01, 8 / divisions];
μTrange = Range[10, 35, 25 / divisions];

(*rerun simulations using bootstrapped params*)
EMproc3[n_, T_, r_] := n + (lactin2[T, bootparams[[r]]] n - α n^2) δt;
tmax = 10 000;
δt = 0.01;
α = 0.0001;
reps = 100;
γ = 2; (*SET AUTOCORRELATION LEVEL HERE*)

output = Flatten[ParallelTable[Module[{extinct = 0}, Do[p =
        NExpectation[lactin2[x, bootparams[[r]]], x ≈ NormalDistribution[μT, σT]] / α;
```

```
   T = SpecSynFourier[γ, (tmax / δt), μT, σT];
   Do[p = EMproc3[p, T[[i]], r];
    If[p < 1, extinct++;
     Break[]], {i, 1, tmax / δt}];, {r, 1, reps}];
  {μT, σT, 1 - extinct / reps}], {σT, σTrange}, {μT, μTrange}], 1];

Export[directory <> "SDEbootstrap_gamma2.m", output, "MX"]
```

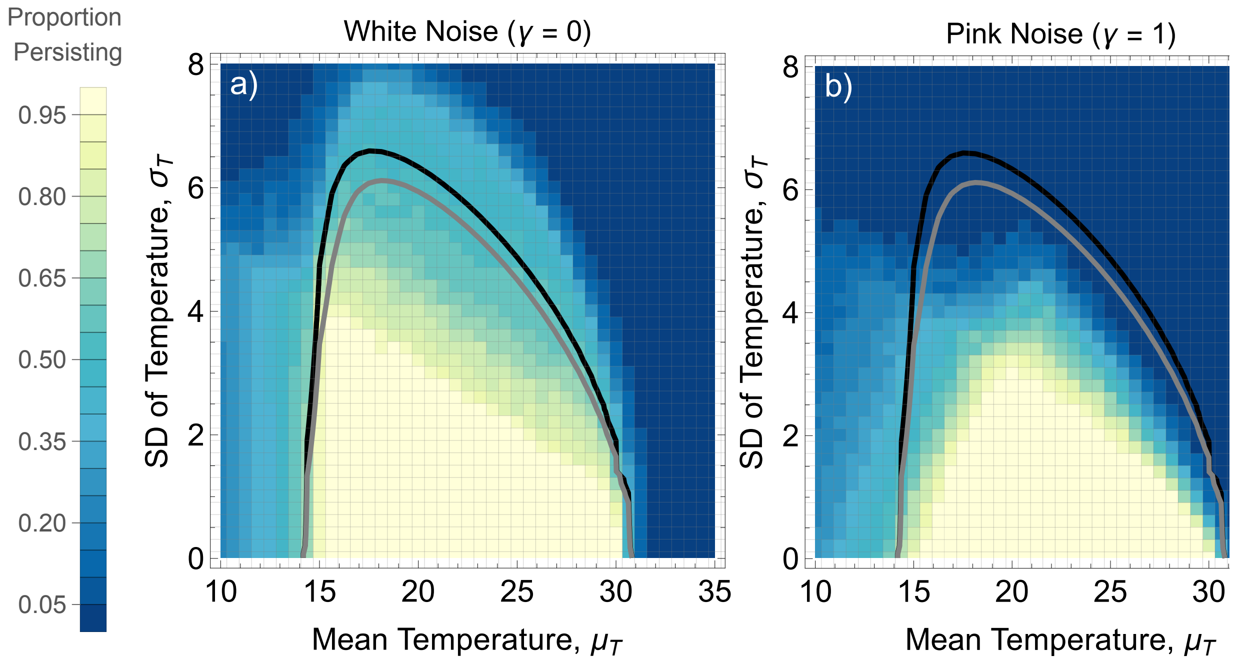
In[3462]:=

```
(*fig S2: plot the results of propagating TPC uncertainty throughout SDE sims*)
SDEboot0 = Import[directory <> "SDEbootstrap_gamma0.m", "MX"];
SDEboot1 = Import[directory <> "SDEbootstrap_gamma1.m", "MX"];
SDEboot2 = Import[directory <> "SDEbootstrap_gamma2.m", "MX"];

SDEboot0plot = Show[ListContourPlot[SDEboot0, InterpolationOrder → 0, Contours → 19,
    ColorFunction → newmap, PlotLegends → Placed[BarLegend[Automatic, LegendLabel →
        "Proportion\n Persisting", LabelStyle → {Darker[Gray], 13}], Left],
    PlotLabel → Style["White Noise (γ = 0)", 15], ImageSize → 310,
    Frame → True, FrameStyle → 16,
    FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
    GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
    GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
    Epilog →
     {{Text[Style["a)", White, 17], {11.2, 7.7}]}}],
   ListContourPlot[moments[[1 ;;, {1, 2, 7}]], InterpolationOrder → 3,
    Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
      {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]];
SDEboot1plot = Show[ListContourPlot[SDEboot1, InterpolationOrder → 0, Contours → 19,
    ColorFunction → newmap, PlotLabel → Style["Pink Noise (γ = 1)", 15],
    ImageSize → 310, Frame → True, FrameStyle → 16,
    FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
    GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
    GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
    Epilog →
     {{Text[Style["b)", White, 17], {11.2, 7.7}]}}],
   ListContourPlot[moments[[1 ;;, {1, 2, 7}]], InterpolationOrder → 3,
    Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
      {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]];
SDEboot2plot = Show[ListContourPlot[SDEboot2, InterpolationOrder → 0, Contours → 19,
    ColorFunction → newmap, PlotLabel → Style["Brown Noise (γ = 2)", 15],
    ImageSize → 310, Frame → True, FrameStyle → 16,
    FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
    GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
    GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
    Epilog →
```

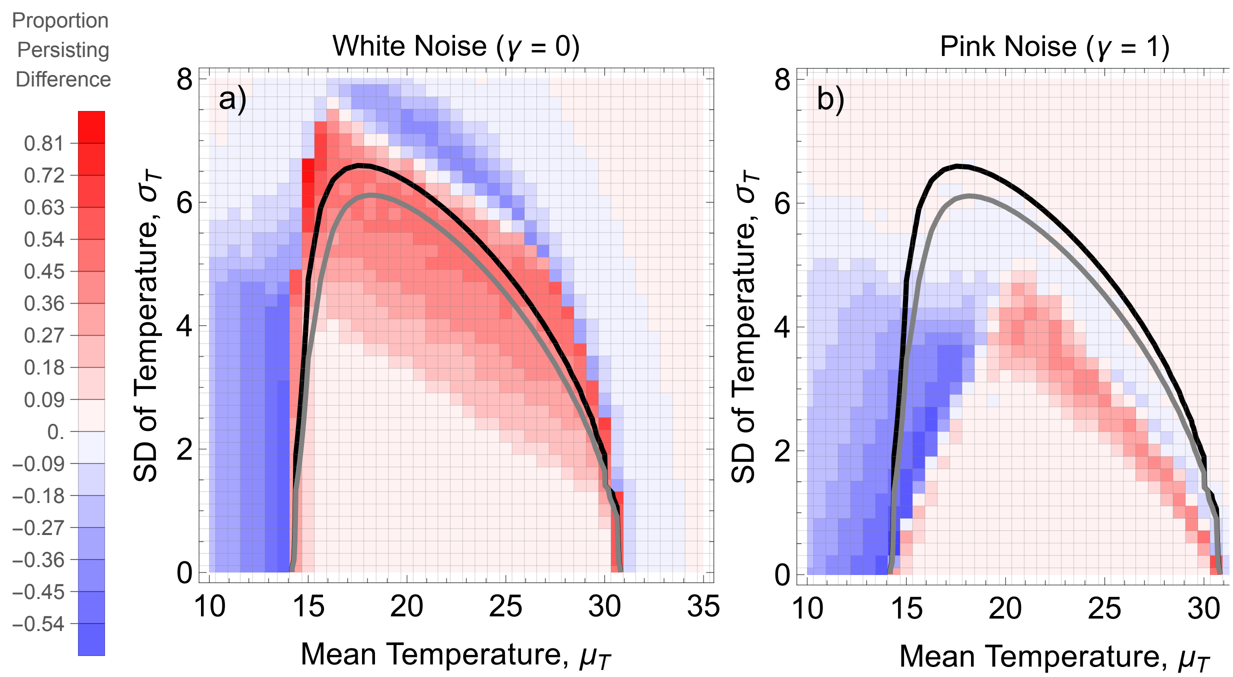
```
      {{Text[Style["c)", White, 17], {11.2, 7.7}]}}],
    ListContourPlot[moments[[1 ;;, {1, 2, 7}]], InterpolationOrder → 3,
     Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
       {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]];

SDEbootstraps = GraphicsRow[
  {SDEboot0plot, SDEboot1plot, SDEboot2plot}, Spacings → 0, ImageSize → 1000]
(*Export[directory<>"SDEbootstraps.pdf",
 SDEbootstraps,"PDF",ImageResolution→1000]*)

(*fig S3: difference plots of propagating TPC uncertainty throughout SDE sims*)
SDE0 = Import[directory <> "SDE10000_gamma0.m", "MX"];
SDE1 = Import[directory <> "SDE10000_gamma1.m", "MX"];
SDE2 = Import[directory <> "SDE10000_gamma2.m", "MX"];

diff0 = SDE0[[All, 3]] - SDEboot0[[All, 3]];
comp0 = Transpose[{SDE0[[All, 1]], SDE0[[All, 2]], diff0}];
comp0plot = Show[ListContourPlot[comp0, InterpolationOrder → 0,
     Contours → 19, PlotRange → {minVal, maxVal}, ColorFunction → newmap2,
     ColorFunctionScaling → False, PlotLegends → Placed[BarLegend[Automatic,
        All, LegendLabel → "Proportion\n Persisting\n Difference",
        LabelStyle → {Darker[Gray], 11}], Left],
     (*Placed[Automatic,Left]*)PlotLabel → Style["White Noise (γ = 0)", 15],
     ImageSize → 310, Frame → True, FrameStyle → 16,
     FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
     GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
     GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
     Epilog →
      {{Text[Style["a)", Black, 17], {11.2, 7.7}]}}],
    ListContourPlot[moments[[1 ;;, {1, 2, 7}]], InterpolationOrder → 3,
     Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
       {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]];

diff1 = SDE1[[All, 3]] - SDEboot1[[All, 3]];
comp1 = Transpose[{SDE1[[All, 1]], SDE1[[All, 2]], diff1}];
comp1plot =
  Show[ListContourPlot[comp1, InterpolationOrder → 0, Contours → 19, PlotRange →
       {minVal, maxVal}, ColorFunction → newmap2, ColorFunctionScaling → False,
     (*PlotLegends→Placed[BarLegend[Automatic,All],Left],*)
     (*Placed[Automatic,Left]*)PlotLabel → Style["Pink Noise (γ = 1)", 15],
     ImageSize → 310, Frame → True, FrameStyle → 16,
     FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
     GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
     GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
```

```
    Epilog →
     {{Text[Style["b)", Black, 17], {11.2, 7.7}]}}],
   ListContourPlot[moments⟦1 ;;, {1, 2, 7}⟧, InterpolationOrder → 3,
    Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
      {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]];

diff2 = SDE2⟦All, 3⟧ - SDEboot2⟦All, 3⟧;
comp2 = Transpose[{SDE2⟦All, 1⟧, SDE2⟦All, 2⟧, diff2}];
comp2plot =
  Show[ListContourPlot[comp2, InterpolationOrder → 0, Contours → 19, PlotRange →
     {minVal, maxVal}, ColorFunction → newmap2, ColorFunctionScaling → False,
    (*PlotLegends→Placed[BarLegend[Automatic,All],Left],*)
    (*Placed[Automatic,Left]*)PlotLabel → Style["Brown Noise (γ = 2)", 15],
    ImageSize → 310, Frame → True, FrameStyle → 16,
    FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
    GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
    GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
    Epilog →
     {{Text[Style["c)", Black, 17], {11.2, 7.7}]}}],
   ListContourPlot[moments⟦1 ;;, {1, 2, 7}⟧, InterpolationOrder → 3,
    Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
      {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]];

bootdiffs =
 GraphicsRow[{comp0plot, comp1plot, comp2plot}, Spacings → 0, ImageSize → 1000]
(*Export[directory<>"bootdiffs.pdf",bootdiffs,"PDF",ImageResolution→1000]*)
```

Out[3468]=



Out[3481]=



```
(*fig S4: plot SDE outputs for any autocorrelation level. pick
   your desired plot and import the corresponding gamma MX file;
note that label does not update automatically. Options are gamma0,
0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2*)
SDEoutput = Import[directory <> "SDE10000_gamma0.75.m", "MX"];
Show[ListContourPlot[SDEoutput, InterpolationOrder → 0,
  Contours → 19, ColorFunction → newmap, PlotLegends →
   Placed[BarLegend[Automatic, LegendLabel → "Proportion\n Persisting",
     LabelStyle → {Darker[Gray], 13}], Left], ImageSize → 310, Frame → True,
  FrameStyle → 16, FrameLabel → {"Mean Temperature, μT", "SD of Temperature, σT"},
  GridLines → {μTrange + .5 × 25 / 40, σTrange + .5 × 8 / 40},
  GridLinesStyle → Directive[Opacity[0.4], Thickness[0.0001]],
  Epilog → {{Text[Style["γ = 0.75", White, Bold, 20], {30.5, 7.5}]}}],
 ListContourPlot[moments[[1 ;; , {1, 2, 7}]], InterpolationOrder → 3,
  Contours → {0, Log10[2]}, ContourStyle → {{Thickness[0.01], Opacity[1], Gray},
    {Thickness[0.01], Opacity[1], Black}}, ContourShading → None]]
```

Out[3503]=