Supplement of Earth Syst. Dynam., 11, 603–615, 2020
https://doi.org/10.5194/esd-11-603-2020-supplement

Earth System Dynamics
Open Access
EGU

Supplement of

Climate change in a conceptual atmosphere–phytoplankton model

György Károly et al.

Correspondence to: György Károlyi (karolyi@reak.bme.hu)

S1: Analytic results without mixing

We derive analytic results for the long-term concentration and temperature contrast behavior when mixing is negligible ($\gamma = 0$). Numbers in (brackets) refer to the equations of the main part paper, letter S and numbers refer to equations on this Supplamantary Material.

Without seasonality

Let us consider equation (2) and (9) with $\gamma = 0$ and $A = 0$ leading to

$$\dot{c} = rc\left(1 - \frac{c}{1 - \alpha(-D_0 t + \beta(c(t) - 1))}\right). \tag{S1}$$

This equation is found to possess an asymptotically linear behavior

$$c^*(t) = St + \delta. \tag{S2}$$

With this form, from (4) and (8), we find

$$F(t) - F_r = -D_0 t + \beta S t + \beta(\delta - 1) \equiv -Dt + \beta(\delta - 1), \tag{S3}$$

where

$$D = D_0 - \beta S \tag{S4}$$

is the slope of the time evolution of the temperature contrast. This shows that the temperature contrast, and in particular, the strength of the climate change, becomes *influenced* by the phytoplankton concentration.

In order to specify the unknown constants S and δ, let us rewrite (S1) as

$$S\left(\alpha D t + 1 - \alpha\beta(\delta - 1)\right) = r(St + \delta)\left[(\alpha D - S)t + 1 - \alpha\beta(\delta - 1) - \delta\right]. \tag{S5}$$

For long times, i.e. $t \gg 1/r$, the quadratic term dominates on the right hand side which cannot be compensated by anything on the left hand side. The coefficient of the quadratic term should vanish, i.e. $\alpha D = S$ from which, since $D = D_0 - \beta S$,

$$S = \frac{D_0 \alpha}{1 + \beta\alpha}, \tag{S6}$$

$$D = \frac{D_0}{1 + \beta\alpha}, \tag{S7}$$

as stated in (11,12).

Assuming that the linear form is valid not only for very large times, but for intermediate ones, too, the linear terms on both sides should compensate each other. From (S5), this yields

$$S\alpha D = rS\left[1 - \alpha\beta(\delta - 1) - \delta\right]. \tag{S8}$$

After dividing by S and using that $\alpha D = S$, we find

$$S = r(1 - \delta(1 + \alpha\beta) + \alpha\beta) \tag{S9}$$

from which

$$\delta = \frac{1 + \alpha\beta - S/r}{1 + \alpha\beta} = \frac{1 + \alpha\beta - \alpha D/r}{1 + \alpha\beta}. \tag{S10}$$

This result provides the constant δ in the long-term dynamics (S2) of phytoplankton.

With seasonality

Let us add now periodic forcing in the form of (4) with (8), yielding

$$F(t) = F_r - D_0 t + \beta(c(t) - 1) + A\sin(\omega t). \tag{S11}$$

We expect the phytoplankton concentration to be driven to oscillate with the same frequency ω but with some phase shift ϕ. For long times, $t \gg 1/r$, we assume that the snapshot attractor is of the form

$$c^*(t) = St + \delta + B\sin(\omega t - \phi) \tag{S12}$$

with an amplitude B. The corresponding long term behavior of the carrying capacity is

$$K(t) = 1 - \alpha(-Dt + \beta(\delta - 1) + A\sin(\omega t) + \beta B\sin(\omega t - \phi)). \tag{S13}$$

In order to fix the constants, we substitute these into (2) to find

$$(S + B\omega\cos(\omega t - \phi))\left[\alpha D t + 1 - \alpha(\beta(\delta - 1) + A\sin(\omega t) + \beta B\sin(\omega t - \phi))\right] =$$

$$r(St + \delta + B\sin(\omega t - \phi))\left[(\alpha D - \gamma)t + 1 - \alpha(\beta(\delta - 1) + A\sin(\omega t)) - \delta - B(1 + \beta\alpha)\sin(\omega t - \phi)\right]. \tag{S14}$$

The vanishing of the quadratic term in t provides the same equation as without periodic forcing, therefore (S6) and (S7) turn out to remain valid. Similarly, from the linear terms (S10) is recovered.

From the product of trigonometric and linear terms, the following equation follows:

$$B\omega D\alpha\cos(\omega t - \phi) = BS\omega\cos(\omega t - \phi)$$

$$= rS(-\alpha A\sin(\omega t) - B(1 + \alpha\beta)\sin(\omega t - \phi)).$$

After an application of trigonometric identities, from the vanishing of the coefficient of $t\cos(\omega t)$ we find

$$\tan\phi = \frac{\omega}{r(1 + \beta\alpha)}. \tag{S15}$$

From the vanishing of the coefficient of $t\sin(\omega t)$

$$B[\omega\sin\phi + r(1 + \beta\alpha)\cos\phi] = -r\alpha A,$$

from which the concentration amplitude is found to be

$$B = -\frac{\alpha A}{\sqrt{(1 + \beta\alpha)^2 + \omega^2/r^2}}. \tag{S16}$$

We have thus been able to determine analytically the snapshot attractor $c^*(t)$ in a strongly nonlinear model with a linear drift and periodic forcing. The snapshot attractor in the concentration remains point-like (no internal variability in c), but changes in an oscillatory fashion about a linear growth with an amplitude B determined by all the system parameters.

S2: Figures

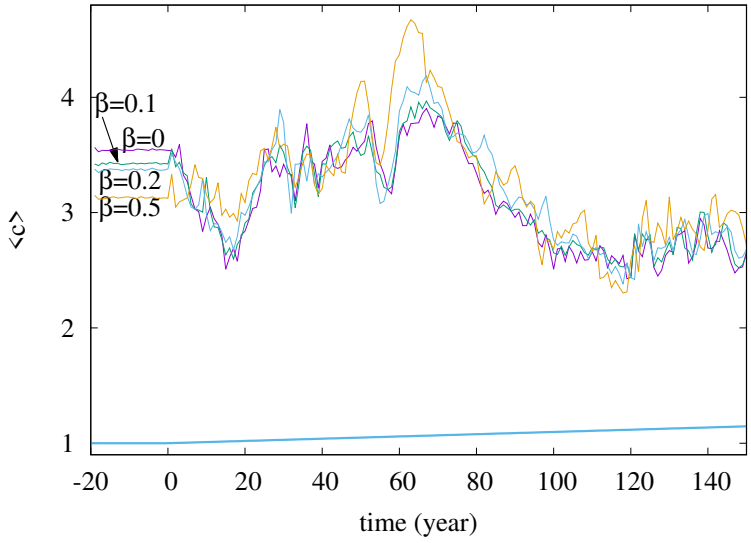

Figure S1. Average phytoplankton concentration $\langle c \rangle$ as a function of time for $\alpha = 0.05$, $\gamma = 0.1$ for various values of β. The straight blue line shows the expected phytoplankton concentration in lack of mixing ($\gamma = 0$) as predicted by Eqs. (S2), (S6), (S10).

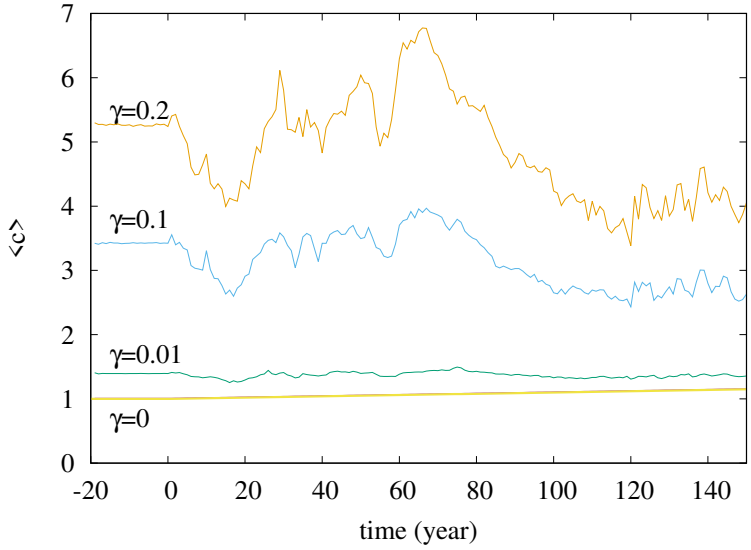

Figure S2. Average phytoplankton concentration $\langle c \rangle$ as a function of time for $\alpha = 0.05$, $\beta = 0.1$ for various values of γ ($\gamma = 0, 0.01, 0.1$ and 0.2). The straight yellow line shows the expected phytoplankton concentration in the lack mixing ($\gamma = 0$) as predicted by Eqs. (S2), (S6), (S10).

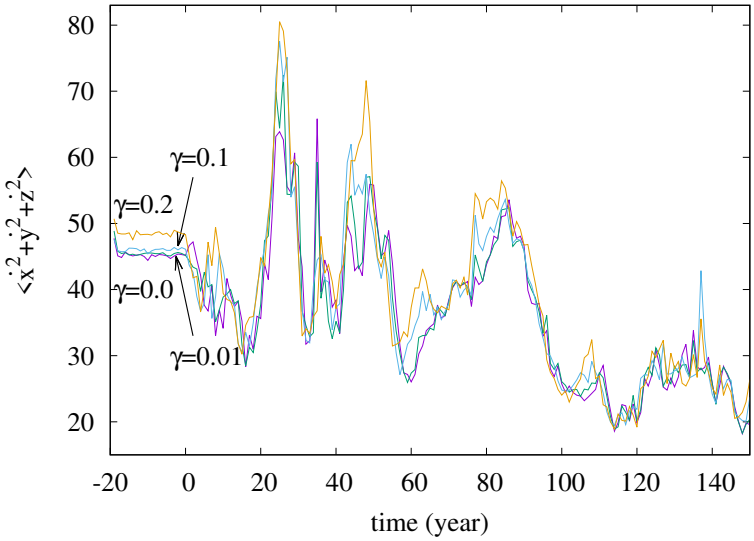

Figure S3. Time-dependence of the ensemble average (over 50000 realization) of the total atmospheric kinetic energy $\dot{x}^2 + \dot{y}^2 + \dot{z}^2$ for $\beta = 0.1$, $\alpha = 0.05$ and $\gamma = 0, 0.01, 0.1, 0.2$. The curves for different values of γ are quite similar.

S3: List of variables and main parameters

$\alpha \in [-0.2, 0.2]$	Enrichment parameter
$\beta \in [0, 0.5]$	Extraction parameter: Strength of CO_2 extraction by phytoplankton
$\gamma \in [0, 0.2]$	Mixing parameter
x, y, z	Variables of Lorenz's atmospheric model, see Eq. (1)
$a = 1/4$, $b = 4$, $G = 1$	Parameters of Lorenz's atmospheric model
$t \in [-10y, 150y]$	Time (in years)
$F(t)$	Time-dependent temperature contrast between the Pole and the Equator, see Eq. (4)
D	Temporal slope of temperature contrast $F(t)$, see Eq. (10)
$F_0(t)$	Time-dependent external forcing of temperature contrast, see Eq. (8)
$D_0 = 2/7300$	Temporal slope of temperature contrast in $F_0(t)$ due to anthropogenic origins
$c(t)$	Time-dependent phytoplankton concentration, see Eq. (2)
S	Temporal slope of phytoplankton concentration $c(t)$, see Eq. (10)
$K(t)$	Time-dependent carrying capacity, see Eq. (9)

Index $()_r$ refers to values in the reference state without anthropogenic effects. Bracket $<>$ indicates ensemble average.

S4: Applied numerical code

We used various versions of the following C code during our numerical simulations.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define NMAX 4                          /* Number of unknowns           */
#define PONTOSSAG 1e-15                 /* Small number                 */
#define PI 3.14159265358979324
#define DT 0.01                         /* Timestep                     */
#define ZR 0.01                         /* Radius around Z=0            */
#define STARTYEAR -20.0                 /* Starting year                */
#define ITERYEAR 150.0                  /* Last year of simulation      */
#define ITERMAX 73*ITERYEAR             /* Max number of timesteps      */
#define FELTMAX 1000000.0               /* Number of initial conditions */
#define STARTYEAR -20.0                 /* Starting year                */
#define STARTTIME 73*STARTYEAR          /* Start year in time units,
                                           1 unit  = 5 days
                                          73 units = 1 year */

#define a 0.25                          /* Parameters of the Lorenz model*/
#define b 4.0
#define G 1.0

#define alpha 0.0                       /* Parameters of the model      */
#define beta 0.05
#define D 0.00027397260274
#define FA 2.0
#define omega PI*0.027397260274
#define FF 9.5
#define epsilon 0.0
#define R 1.0

int deriv(double x[NMAX], double t, double xder[NMAX])
{
  double ft;
  if (t<0)
  {
    ft=alpha*(x[3]-1.0)+FA*sin(omega*t);
  }
  else
  {
    ft=alpha*(x[3]-1.0)-D*t+FA*sin(omega*t);
  }
```

```
  xder[0]=-pow(x[1],2)-pow(x[2],2)-a*x[0]+a*(ft+FF);
  xder[1]=x[0]*x[1]-b*x[0]*x[2]-x[1]+G;
  xder[2]=x[0]*x[2]+b*x[0]*x[1]-x[2];
  xder[3]=R*x[3]*(1-x[3]/(1-beta*ft+epsilon*(pow(xder[0],2)+ pow(xder[1],2)+pow(xder[2],2)))
  return 0;
}

int rk4(double x[NMAX], double t, double dt)
{
  double k1[NMAX],k2[NMAX],k3[NMAX],k4[NMAX],y[NMAX];
  int i;

  deriv(x,t,k1);
  for (i=0; i<NMAX; y[i]=x[i]+dt*k1[i]/2, i++);
  deriv(y,t+dt/2,k2);
  for (i=0; i<NMAX; y[i]=x[i]+dt*k2[i]/2, i++);
  deriv(y,t+dt/2,k3);
  for (i=0; i<NMAX; y[i]=x[i]+dt*k3[i], i++);
  deriv(y,t+dt,k4);
  for (i=0; i<NMAX; x[i]=x[i]+dt*(k1[i]+2*k2[i]+2*k3[i]+k4[i])/6, i++);
  return 0;
}

int main()
{
  double t, x[NMAX], E, xder[NMAX];
  int i, j, k;
  FILE *fp;
  srand(time(NULL));
  fp = fopen( "s00.dat","w" );

  /* Writes parameters*/
  fprintf(fp, "#FA=%.2f, FF=%.2f, alpha=%.2f, beta=%.2f, epsilon=%.2f, D=%.6f, omega=%.6f, r
  fprintf(fp, "#Datasets: time  x  dx/dt  y  dy/dt  z  dz/dt  c  dc/dt\n");

  /* Initial data */
  for(k=0; k<FELTMAX; k++) {
    t=STARTTIME;
    x[0]=((double)rand()/(double)(RAND_MAX))*3.5-0.5;
    x[1]=((double)rand()/(double)(RAND_MAX))*5-2.5;
    x[2]=((double)rand()/(double)(RAND_MAX))*5-2.5;
    x[3]=((double)rand()/(double)(RAND_MAX))*0.2 +0.9;
    fprintf(fp, "# IC: time = %f x,y,z,c = ", t);
    for (j=0; j<NMAX; j++) {
        fprintf(fp, "%f ", x[j]);
    }
    fprintf(fp,"\n");
    for (i=(int)(STARTTIME/DT); i<(int)(ITERMAX/DT); i++) {
        t = (float)i*DT;
```

```
        rk4(x,t,DT);
        t = t+DT;
        deriv(x,t,xder);
        if (fabs(t/73 - (float)(int)(t/73))<0.00000001 && x[2]>-ZR && x[2]<ZR){
            fprintf(fp, "%f ", t);
            for (j=0; j<NMAX; j++) {
                fprintf(fp, "%f %f ", x[j],xder[j]);
            }
            fprintf(fp, "\n");
        }
    }
  }

  fclose(fp);
  return 0;
}
```