

Модель трансформации атмосферных загрязнений (Кольский)

Балансовая модель трансформации атмосферных выбросов N_i в почве, воде и донных отложениях для одного источника и отдельного водосбора представлена на рисунке в виде потоковой диаграммы. Рассмотрим балансовую модель трансформации атмосферных выбросов N_i в почве, воде и донных отложениях. Единицей моделирования является озеро с соответствующим водосбором (почвой). Модель описывает 2 источника выбросов и 84 (не взаимодействующих между собой) водосбора.

Трансформацию загрязнений для водосбора, представленную на рисунке, можно описать следующим образом. Источником загрязнения является промышленность (номер 1). С интенсивностью $P(t)$ (номер 2) часть загрязнений выбрасывается в атмосферу (номер 3), откуда выпадает на поверхность в зависимости от времени t , расстояния до источника r и направления f_i : интенсивность выпадения нерастворимых форм (номер 4a) $Dunsol(t, f_i, r)$, растворимых (номер 4b) $Dsol(t, f_i, r)$. Нерастворимая часть загрязнений попадает в почву (номер 5) и в донные отложения (номер 11). Загрязнение, попавшее в почву, частично задерживается в ней, частично смывается в донные отложения или переходят в растворимую форму (номер 6) и попадает в воду озера (номер 7). Загрязнения в растворимой форме попадают с интенсивностью $Dsol(t, f_i, r)$ (номер 4b) в воду озера, откуда могут выноситься с интенсивностью R (номер 8).

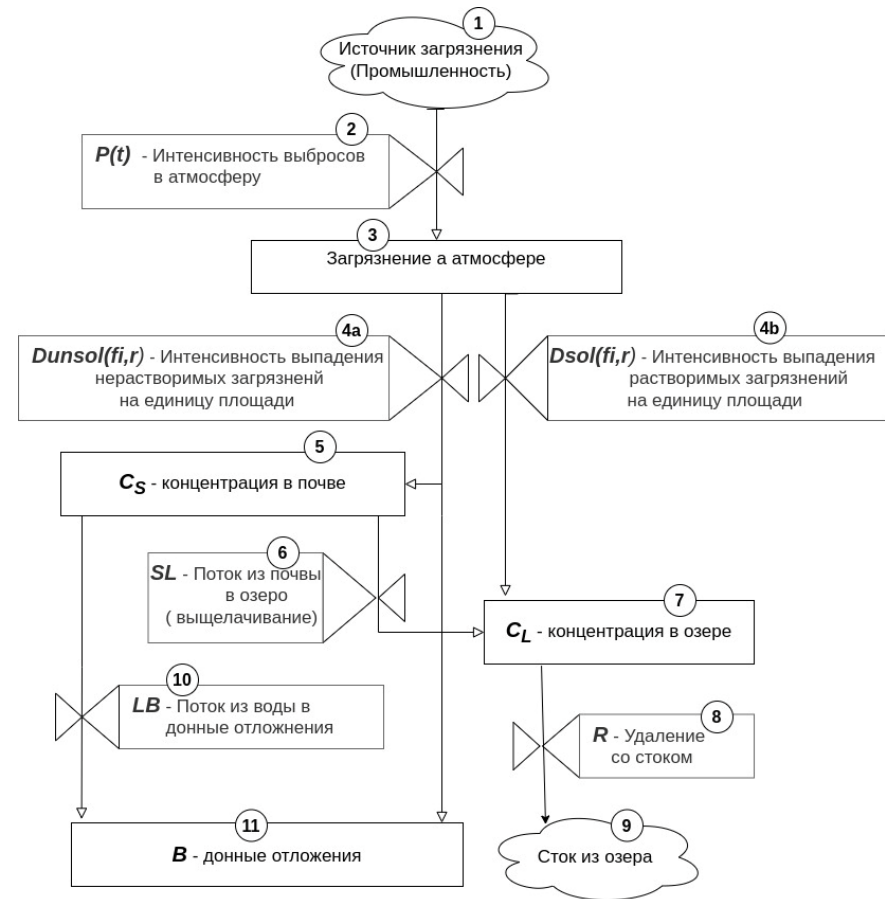


Схема на рисунке, отражает лишь общее представление о функционировании объекта. Приведем более полное формальное

математическое описание (основные уравнения) модели для водосбора. Каждый водосбор разбивается на две части: суша (площадь SQ_s) и озеро (площадь SQ_w). Начальный момент моделирования - 1946г. Используемые обозначения расшифровываются в табл. 1.

Таблица 1. Сокращения и единицы измерения

Сокращения	Расшифровка	Ед. изм.
t	Время	год
r	Расстояние от источника загрязнения до точки выпадения	км
φ	Угол направления от источника загрязнения до точки выпадения	град
$P(t)$	Мощность источника	т/год
$R(r)$	Зависимость интенсивности выпадения от r	1/м ²
$Fi(\varphi)$	Зависимость интенсивности выпадения от φ	-
D	Интенсивность выпадения (на единицу площади в единицу времени)	мг/м ² /год
$Dsol$	Интенсивность выпадения растворимой формы загрязнителя	мг/м ² /год
$Dinsol$	Интенсивность выпадения нерастворимой формы загрязнителя	мг/м ² /год
sol	Доля растворимой формы	-
$insol$	Доля нерастворимой формы	-
$S(t)$	Запас загрязнителя в почве в момент времени t	мг/м ²
$Snow$	Часть года со снежным покровом	-
$leach$	Выщелачивание: переход из нерастворимой в растворимую (результатом идентификации является $leach=0.001$)	1/год
S_{fon}	Фоновые значения запасов	мг/м ²
$W(t)$	Концентрация загрязнителя в воде озера в момент времени t	мг/м ³
SQ_s	Площадь суши	м ²
SQ_w	Площадь озера	м ²
$precip$	Осадки	м/год
H	Условная глубина озера. Определяет интенсивность водообмена (результатом идентификации является $H=1.2$)	м
$B(t)$	Ежегодное поступление загрязнителя в донные отложения в момент времени t . Запас загрязнителя в слое t .	мг/м ² год
B_{fon}	Фоновое ежегодное поступление загрязнителя в донные отложения	мг/м ² год

1. Выпадение загрязняющих веществ от каждого источника. Предполагается мультипликативное представление функции.

Для источника в г. Никель (N):

$$D_N(t, \varphi_N, r_N) = P_N(t) \cdot R_N(r_N) \cdot FI_N(\varphi_N),$$

где r_N - расстояние от источника до водосбора и φ_N - соответствующее направление (полярные координаты).

Аналогично для источника в г. Заполярный:

$$D_Z(t, \varphi_Z, r_Z) = P_Z(t) \cdot R_Z(r_Z) \cdot FI_Z(\varphi_Z).$$

2. Для каждого водосбора выпадение загрязняющих веществ (в растворимой и нерастворимой форме) от двух источников суммируется:

$$D_{sol}(t) = D_N(t, \varphi_N, r_N) \cdot sol(r_N) + D_Z(t, \varphi_Z, r_Z) \cdot sol(r_Z)$$

$$D_{unsol}(t) = D_N(t, \varphi_N, r_N) \cdot unsol(r_N) + D_Z(t, \varphi_Z, r_Z) \cdot unsol(r_Z)$$

3. Динамика запасов (содержания) загрязнения в почве:

$$dS/dt = D_{unsol}(t) \cdot (1 - Snow) - leach \cdot S(t)$$

$$S(0) = S_{fon}.$$

Предполагается, что в начальный момент запасы в почве соответствуют фоновым значениям, что в почве остаются только нерастворимые формы, выпавшие в бесснежный период, что процесс выщелачивания переводит Ni в растворимую форму, которая выносится из почвы в воду.

4. Динамика содержания (концентрации) загрязнения в воде озера

$$dW/dt = (D_{sol}(t) + leach \cdot S(t) \cdot SQ_S / (SQ_W + SQ_S) - precip \cdot W) / (H + precip)$$

Предполагается, что все растворимые формы попадают в озеро вместе со всеми осадками ($precip = 0.40$ м/год), что все растворимые формы, образовавшиеся в почве в результате выщелачивания, добавляются в воду озера.

5. Донные отложения (по слоям):

$$B(t) = D_{insol}(t) \cdot (1 + SQ_S / SQ_W) \cdot Snow + B_{fon}$$

Предполагается, что слой донных отложений формируется из нерастворимых форм, попадающих непосредственно на поверхность озера, из нерастворимых форм, попадающих непосредственно на сушу в зимний период (и попавших в озеро в половодье) и из фонового значения, отражающее доиндустриальные процессы выветривания коренных пород и выщелачивания почв.

Модель содержит множество неизвестных функций, которые необходимо определить (идентифицировать) так, чтобы обеспечить баланс между близостью траектории модели к данным и сложностью модели. Для этого и используется технология сбалансированной идентификации. Ниже приводится текст программы, содержащий:

- ⑩ уравнения модели (синий цвет),
- ⑩ проекторы - операторы, связывающие переменные модели с измерениями (зеленый),
- ⑩ критерий идентификации (красный цвет).

Т е к с т п р о г р а м м ы S v F (т е х н о л о г и и с б а л а н с и р о в а н н о й и д е н т и ф и к а ц и и)

```
BoF-SvF
#ObjToReadSols = True
```

```
CVNumOfIter = 0
RunMode = 'P&S'
```

```
latNik = 69.413; lonNik = 30.234
latZap = 69.425; lonZap = 30.822
```

```
AzimuthInit ( latZap, lonZap )
```

```
# Zap in the center
```

```
rmax = 170
minR = 7
rStep = 5
```

```
# Область модел
```

```
StartYear = 1946
EndYear   = 2030
StartZap  = 1956
```

```
Snow = 0.6
notSnow = 1-Snow
```

```
SET:    R = [ 0,      rmax, rStep ]
        R1 = [ rStep,  rmax, rStep ]
        T = [ StartYear, EndYear, 1 ]
        T9 = [ StartYear, 2009, 1 ]
        Tpro= [ 2020,      EndYear, 1 ]
        TZap0 =[ StartYear, StartZap, 1 ]
        Fi = [-180, 180, 10]
```

```
Set:    t2010_2014 = [2010, 2014, 1]
        t2016_2019 = [2016, 2019, 1]
        t2022_2030 = [2022, 2030, 1]
```

```
##### # ВЫБРОСЫ Nik Zap #####
EMS2a = Select Year as T, NiNi as NikZap from ../../_DATA/2Source2030.xlsx # where T <= 2009
Var:    NikZap (T) >= 0
        Nik (T) >= 0
        Zap (T) >= 0; NikZap = Nik + Zap;
EQ:    Nik(T9) > Zap(T9)
        Nik( StartYear ) <= 10
        Zap( TZap0 ) = 0.0 # <=1956
        Zap ( t2010_2014 ) >= Nik ( t2010_2014 ) * 0.9
        Zap ( t2010_2014 ) <= Nik ( t2010_2014 ) * 1.1
        Zap ( t2016_2019 ) <= Nik ( t2016_2019 ) / 3.
        NikZap ( t2022_2030 ) <= 1.
```

```
#Draw Nik Zap;DC:red
```

```
##### Р а с т о я н и я  и  У г л ы  д о  и с т о ч н и к о в #####
No.No.No.No.No.No.No.No.No.No.No.No.No.No.No.No.No.No.No.No.No.
```

```

xyNik = LatLonToAzimut ( latNik, lonNik )           # шир, долг -> декарт
def fiNik(X,Y): return degrees ( arctan2 (Y-xyNik[1], X-xyNik[0]) )   # fi
def rNik(X,Y): return sqrt      ( (Y-xyNik[1])**2 + (X-xyNik[0])**2 )   # r

```

```

xyZap = LatLonToAzimut ( latZap, lonZap )           # шир, долг -> декарт
def fiZap(X,Y): return degrees ( arctan2 (Y-xyZap[1], X-xyZap[0]) )   # fi
def rZap(X,Y): return sqrt      ( (Y-xyZap[1])**2 + (X-xyZap[0])**2 )   # r

```

ПЕРЕНОС И ВЫПАДЕНИЕ

```

Var: Soluble( R ) > 0.1; < 0.5; d/dR (Soluble) > 0

```

```

Select Fi, RosePol AS PfiNik from ../../_DATA/NikRosePol137(Fi).sol      # НИКЕЛЬ

```

```

Var: PfiNik ( Fi ) > 0.4;                                                # Роза

```

```

EQ:      ∫ ( -180, 180, d(R)*PfiNik(R) ) == 360

```

```

PfiNik(-180) == PfiNik(180) #Cycle

```

```

Var: P_RNik( R ) >= 0; P_RNik( rmax ) = 0                                # ПЕРЕНОС

```

```

EQ:      d/dR1(P_RNik(R1)*R1) <= - P_RNik(R1)*R1*0.0032;

```

```

∫ ( 0, rmax, d(R)*P_RNik(R)*R ) * 2*pi == 1

```

```

def DEPOSITNik(X,Y,T): return fPfiNik(fiNik (X,Y)) *fP_RNik(rNik(X,Y)) * fNik(T) * 1000

```

```

def DEPOSITsolNik(X,Y,T): return DEPOSITNik(X,Y,T) * fSoluble(rNik(X,Y))

```

```

def DEPOSITinsolNik(X,Y,T): return DEPOSITNik(X,Y,T) * ( 1-fSoluble(rNik(X,Y)) )

```

```

Select Fi, RosePol AS PfiZap from ../../_DATA/ZapRosePol137(Fi).sol      # ЗАПОЛЯРНЫЙ

```

```

Var: PfiZap ( Fi ) > 0.4;                                                # Роза

```

```

EQ:      ∫ ( -180, 180, d(R)*PfiZap(R) ) == 360

```

```

PfiZap(-180) == PfiZap(180) #Cycle

```

```

Var: P_RZap( R ) >= 0; P_RZap( rmax ) = 0                                # ПЕРЕНОС

```

```

EQ:      d/dR1(P_RZap(R1)*R1) <= - P_RZap(R1)*R1*0.0032;

```

```

∫ ( 0, rmax, d(R)*P_RZap(R)*R ) * 2*pi == 1

```

```

def DEPOSITZap(X,Y,T): return fPfiZap(fiZap (X,Y)) *fP_RZap(rZap(X,Y)) * fZap(T) * 1000

```

```

def DEPOSITsolZap(X,Y,T): return DEPOSITZap(X,Y,T) * fSoluble(rZap(X,Y))

```

```

def DEPOSITinsolZap(X,Y,T): return DEPOSITZap(X,Y,T) * ( 1-fSoluble(rZap(X,Y)) )

```

```

def DEPOSIT(X,Y,T):      return DEPOSITNik(X,Y,T)      + DEPOSITZap(X,Y,T)  # В Ы П А Д Е Н И Е   О б щ е е   mg/m2/year
def DEPOSITsol(X,Y,T):   return DEPOSITsolNik(X,Y,T)   + DEPOSITsolZap(X,Y,T)
def DEPOSITinsol(X,Y,T): return DEPOSITinsolNik(X,Y,T) + DEPOSITinsolZap(X,Y,T)

#_____#

CONS = Select * from CONS.txt

Set:  b = [ 0,          CONS.NoR-1, 1]
      t = [ StartYear, EndYear,    1]

Param: X( b )
        Y( b )
        Catch( b ) # П л о щ а д ь   б а с с е й н а
        S( b )#   П л о щ а д ь   о з е р а

#####                                # М О Д Е Л ь 3-ё х   с р е д
fonNiSoil = 20
Var:  sNi ( b , t ) > 7.5;  sNi(b,StartYear) <300; sNi(b,StartYear) >fonNiSoil*5.3 # 800*hSoil    # П О Ч В А      к о л - в о в   П о ч в е   mg/m2
      leaching > 0.0001; < 0.001
EQ:   d/dt ( sNi ) = DEPOSITinsol(X,Y,t) * notSnow - leaching * sNi

#-----
precip = 0.4
Var:  wNi ( b , t ) > 0;  wNi(b,StartYear) < 2                                # В О Д А      mkg/dm3 = mg/m3
      H > precip; < 3*precip
EQ:   d/dt( wNi ) = ( DEPOSITsol(X,Y,t) + leaching * sNi * (Catch(b)-S(b)) / Catch(b) - precip * wNi )/(H + precip)

#-----
fonNiSedi = 20
Var:  bNi ( b , t ) > 0;                                                    # Д О      mg/m2   д л я   к а ж д о г о   г о д а
EQ:   bNi = DEPOSITinsol(X,Y,t) * (1+(Catch(b)-S(b))/S(b)*Snow) + fonNiSedi * 0.1 # Vsedim*0.2=0.1

#####                                # П р о е к т о р ы   С В Я З ь   С
ИЗМЕРЕНИЯМИ
TDep = Select * from TDep.txt                                # В Ы П А Д Е Н И Я

```

```

##### DEPOSITION  mg/m2/г о д   1e-3   or t/km2/year or  kg /km2 /year #####
Set:   nD = [, , 1]
Param: bD      (nD)
        TD      (nD)
Var:   depNi (nD) > 0                                #   DEPOSITION  mg/m2/г о д
EQ:    depNi (nD) = DEPOSIT ( X(bD(nD)), Y(bD(nD)), TD(nD) )
#-----# ПОЧВА
Soil = Select * from Soil.txt ##### ПОЧВА  mg/kg(с у х. в е с а): soilNi*0.8(кг.с у х/литр)*10*10*10 (mg/m3) * hSoil(m) =
soilNi*800*hSoil(m) = soilNi (mg/m2)
Var:   hSoil > 0.005; < 0.01      # С л о й  в  м е т р а х                Д о л я  с у х о г о  в е щ е с т в а
С л о й  в  м е т р а х
Set:   nS = [, , 1]
Param: bS      (nS)
        tS      (nS)
Var:   soilNi (nS) > 0                #   С О Д Е Р Ж А Н И Е  mg/kg(с у х. в е с а)
        FonSoilNi (nS) > 0          #   Ф о н о в о е  з н а ч е н и е,  н а ч а л ь н о е  у с л о в и е
EQ:    soilNi * 800*hSoil = sNi ( bS , tS )
        FonSoilNi * 800*hSoil = sNi ( bS , StartYear )

TBott = Select * from TBott.txt                                #   Д О  SEDIMENTATION
##### botNi (mg/kg-dw) *0.2 (kg-dw/dm3) *10+3 (dm3/m3) *Vsedim(mm/year) 10-3 (m/mm) = botNi*Vsedim*0.2 (mg/m2/year) = bNi (mg/m2/year) - выпадение
Set:   nB = [, , 1]
Param: bB      (nB)
        tB      (nB)
        Vsedim(nB)      #   mm/year
        deIT (nB)
Var:   botNi (nB) > 0                #   С О Д Е Р Ж А Н И Е  mg/kg(с у х. в е с а)

EQ:    botNi (nB)*Vsedim(nB)*0.25 = ∫ ( tB(nB)-deIT(nB), tB(nB), d(T)*bNi (bB, T) ) / deIT(nB)

TSedAcc = Select * from TSedAcc.txt                                #   П О Т О К  Д О  SEDIMENTATION ACCAMUL  mg/m2/y
Set:   nSA = [, , 1]
Param: bSA      (nSA)
        Tsa      (nSA)
Var:   sedaccNi (nSA) > 0                                #   mg/m2/г о д
Set:   multNumSA = [ , , 1 ]

```



```

Var:      multSA(multNumSA) >0.5; <2
Param:    multNumSA (nSA)
          #multSA > 0
EQ:      sedaccNi (nSA) = DEPOSIT (X(bSA(nSA)), Y(bSA(nSA)), Tsa(nSA)) *(1+(Catch(bSA)-S(bSA))/S(bSA)*Snow) *multSA(multNumSA(nSA))

```

```
Water = Select * from Water.txt
```

```

Set:      nW = [, , 1]
Param:    bW      (nW)
          tW      (nW)
Var:      waterNi (nW) > 0          #          mkg/dm3      В О Д А
EQ:      waterNi = wNi ( bW , tW )

```

```

#MakeSets_byParts 7
MakeSets_byParam bW 7

```

```

Obj:      0.3 * PfiNik.MSD() + PfiNik.Compl ( 0.04 ) + P_RNik.Complexity ( [Penal[0]] ) + Nik.Complexity ( [Penal[1]] )  ¥
          + 0.3 * PfiZap.MSD() + PfiZap.Compl ( 0.04 ) + P_RZap.Complexity ( [Penal[0]] ) + Zap.Complexity ( [Penal[1]] ) + 3*NikZap.MSD()  ¥
          + 20*waterNi.MSDnan() + Soluble.Compl(0.04) + depNi.MSD() + 0.1*soilNi.MSD() +0.1*botNi.MSD() +0.1*sedaccNi.MSD() + 0.01*FonSoilNi.MSD()
          # +0.3*sedNiacc.MSD() +0.3*SediNi93.MSD() +0.3*SoilNi.MSD()

```

```
##### Draw ##### О т р и с о в к а  р е з у л ь т а т о в
```

```

Set:      Year = [ StartYear, EndYear, 1]
TBott = Select botNi as Ni, bB, tB as Year from TBott.txt      where bB == 9
Param:    Ni (Year)
for tt in t.NodS :
#      Ni.grd[tt] = bNi.grd[9,tt]      /(1.7*0.25)
      Ni.grd[tt] = bNi.grd[9,tt]      /(1.7*0.25) *0.55      # П о д г о н к а !

```

```
Draw Ni;Transp
```

```

Draw botNi
Draw Soluble

```

```

SvF.Xsize = 8
SvF.Ylabel_x=0.13;
Substitute=False
NikZap.V. oname = 'Ni - м о д е л ь'; NikZap.V. data_name = 'Ni - д а н н ы е'; NikZap.V. draw_name = 'В ы б р о с ы Ni, т / г о д';
NikZap.A[0]. oname = 'Г о д ы';
Substitute=True
Draw NikZap;LW:1.5;MS:0

```

```

SQUARE = sum ( Catch.grd[int(ib)] for ib in b.NodS )
S_Lakes = sum ( S.grd[int(ib)] for ib in b.NodS )
print (len(b.NodS), 'n      sum SQUARE', SQUARE , '      sum S_Lakes', S_Lakes )

```

```

#Draw

```

```

Param: S_DEPOSITION ( T )
      A_DEPOSITION ( T )
      S_SOIL ( T )
      A_SOIL ( T )
      M_SOIL ( T )
      S_SEDIMENTION ( T )
      A_SEDIMENTION ( T )
      S_WATER ( T )
      A_WATER ( T )
      M_WATER ( T )
      S_OUT ( T )
      A_OUT ( T )

```

```

for it in T.NodS:
    t = T.Val[it]
    S_DEPOSITION.grd[it] = sum ( DEPOSIT (X.grd[ib],Y.grd[ib],t) * Catch.grd[ib] for ib in b.NodS ) /SQUARE
    S_SOIL.grd[it] = sum ( DEPOSITinsol(X.grd[ib],Y.grd[ib],t) * (Catch.grd[ib]-S.grd[ib]) * notSnow for ib in b.NodS ) /SQUARE
    S_SEDIMENTION.grd[it] = sum ( DEPOSITinsol(X.grd[ib],Y.grd[ib],t) * (S.grd[ib]+ (Catch.grd[ib]-S.grd[ib])*Snow) for ib in b.NodS ) /SQUARE
    S_WATER.grd[it] = sum ( DEPOSITsol (X.grd[ib],Y.grd[ib],t) * ( Catch.grd[ib] ) for ib in b.NodS ) /SQUARE
    S_OUT.grd[it] = sum ( wNi.grd[ib, it] * precip * ( Catch.grd[ib] ) for ib in b.NodS ) /SQUARE
    if it == 0:
        A_DEPOSITION.grd[it] = S_DEPOSITION.grd[it]
        A_SEDIMENTION.grd[it] = S_SEDIMENTION.grd[it]
        A_OUT.grd[it] = S_OUT.grd[it]

```

```

#         A_SEDIMENTION.grd[it] = sum ( bNi.grd[ib, it] * ( S.grd[ib] ) for ib in b.NodS ) /SQUARE
    else:
        A_DEPOSITION.grd[it] = S_DEPOSITION.grd[it] + A_DEPOSITION.grd[it-1]
        A_SEDIMENTION.grd[it] = S_SEDIMENTION.grd[it] + A_SEDIMENTION.grd[it-1]
        A_OUT.grd[it] = S_OUT.grd[it] + A_OUT.grd[it-1]
#         A_SEDIMENTION.grd[it] = sum ( bNi.grd[ib, it] * ( S.grd[ib] ) for ib in b.NodS )/SQUARE + A_SEDIMENTION.grd[it-1]
    A_SOIL.grd[it] = sum ( sNi.grd[ib, it] * (Catch.grd[ib]-S.grd[ib]) for ib in b.NodS ) /SQUARE
    M_SOIL.grd[it] = sum ( sNi.grd[ib, it] * (Catch.grd[ib]-S.grd[ib]) for ib in b.NodS ) /SQUARE / (800*hSoil)
    A_WATER.grd[it] = sum ( wNi.grd[ib, it] * S.grd[ib] for ib in b.NodS ) / S_Lakes * (H+precip)
    M_WATER.grd[it] = sum ( wNi.grd[ib, it] * S.grd[ib] for ib in b.NodS ) / S_Lakes
A_SOIL.grd[:] -= A_SOIL.grd[0]

```

```

SvF.Ylabel_x=0.35;
S_WATER.V.oname = ' вода'; S_DEPOSITION.V.oname = ' всего'; S_SOIL.V.oname = ' почва'; S_SEDIMENTION.V.oname = ' донные
отложения'; # ВЫПАДЕНИЕ
Substitute=False
S_SOIL.V.draw_name = ' Распределение выпадения Ni, мг/м2/год'; S_SOIL.A[0].oname = ' Годы';
Substitute=True
Draw S_DEPOSITION;MS:0;LW:1.5 S_SEDIMENTION;LC:black;LSt:dotted S_WATER;LC:b;LSt:dashdot S_SOIL;LC:brown;LSt:dashed

```

```

SvF.Ylabel_x=0.2; #
Накопление
A_WATER.V.oname = ' вода'; A_DEPOSITION.V.oname = ' всего'; A_SOIL.V.oname = ' почва'; A_SEDIMENTION.V.oname = ' донные
отложения'; A_OUT.V.oname = ' потери со стоком';
Substitute=False
A_WATER.V.draw_name = ' Баланс накопления Ni, мг/м2'; A_WATER.A[0].oname = ' Годы';
Substitute=True
#Draw A_DEPOSITION;MS:0;LW:2 A_SEDIMENTION;LW:1;LC:black;LSt:dotted A_OUT;LC:green;LSt:dashdot A_SOIL;LC:brown;LSt:solid A_WATER;LC:c;LSt:dashed
Draw A_DEPOSITION;MS:0;LW:1.5 A_SEDIMENTION;LC:black;LSt:dotted A_OUT;LC:green;LSt:solid;MS:3;LW:1 A_SOIL;LC:brown;LSt:dashed;MS:0;LW:1.5
A_WATER;LC:b;LSt:dashdot

```

```

SvF.Ylabel_x=0.25;
Substitute=False
M_WATER.V.draw_name = ' Концентрация Ni, мг/м2 в воде'; M_WATER.A[0].oname = ' Годы'; M_WATER.V.oname = ' вода';
Substitute=True
Draw M_WATER;LC:b;LSt:dashdot;LW:1.5;MS:0 #M_WATER;LC:c

```

```

SvF.Ylabel_x=0.35;
Substitute=False
M_SOIL.V.draw_name = ' Концентрация Ni, мг/кг (сух. веса) в почве'; M_SOIL.A[0].oname = ' Годы'; M_SOIL.V.oname
=' почва';
Substitute=True
Draw M_SOIL;LC:brown;LSt:dashed;LW:1.5;MS:0 #M_SOIL;LC:brown

```

```

Draw P_RZap P_RNik

```

```

#Set: T = [ 1988, EndYear, 1 ]

```

```

#For: Lppn in [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16] :
# TLake = Select DisNik, DisZap, X, Y, LakeNi as Ni, T, Lpp, Catch, S, SdCatch, ROWNUM as Npp, N1995 from Dat7 as TLake where Lpp == Lppn
# Param: Ni ( T )
# Ni.grd[:] = LakeNi.grd[Lppn,1988-StartYear:]
# # for i in range (1980, EndYear+1) : Ni.grd[i] = LakeNi.grd[Lppn,i]
# Ni.V.name = 'Ni-'+str (TLake.N1995[0])
# print ('N1995', TLake.N1995[0])
# Draw Ni

```

```

TShore = Select * from ../_DATA/SHORE.txt as TShore # SHORE
xx, yy = LatLonToAzimut ( TShore.Y[:], TShore.X[:] )
#xx, yy = LatLonToAzimut ( TShore.Y, TShore.X )
Shore = Polyline (xx, yy, None, 'Shore' )

```

```

TTown = Select * from ../_DATA/TOWN.txt as TTown # TOWN
xx, yy = LatLonToAzimut ( TTown.Y[:], TTown.X[:] )
Town = Polyline (xx, yy, None, 'Town' )

```

```

TBrus = Select * from ../_DATA/BORDER_R.txt as TBrus # BORDER
xx, yy = LatLonToAzimut ( TBrus.Y[:], TBrus.X[:] )
Brus = Polyline (xx, yy, None, 'Brus' )
TBnor_fin = Select * from ../_DATA/BORDER_F_N.txt as TBnor_fin
xx, yy = LatLonToAzimut ( TBnor_fin.Y[:], TBnor_fin.X[:] )
Bnor_fin = Polyline (xx, yy, None, 'Bnor_fin' )

```

```

For: paral in [68 69 70]:
    TParal = Select * from ../_DATA/Pparal.txt as TParal          # paral
    xx, yy = LatLonToAzimut ( TParal.Y[:], TParal.X[:] )
    Pparal = Polyline (xx, yy, None, 'Pparal' )

TMERID = Select * from ../_DATA/MERID.txt as TMERID              # MERID
xx,yy = LatLonToAzimut ( TMERID.Y[:], TMERID.X[:] )
MERID = Polyline (xx, yy, None, 'MERID' )

#Lakes = Polyline (Water.dat('X'), Water.dat('Y'), None, 'Lakes' )
#Depos = Polyline (TDep.dat('X'), TDep.dat('Y'), None, 'Depos' )
Lakes = Polyline (Water.X[:], Water.Y[:], None, 'Lakes' )
Depos = Polyline (TDep.X[:], TDep.Y[:], None, 'Depos' )
TBott = Select * from TBott.txt
Bott = Polyline (TBott.X[:], TBott.Y[:], None, 'Bott' )
SedAcc = Polyline (TSedAcc.X[:], TSedAcc.Y[:], None, 'SedAcc' )
PSoil  = Polyline (Soil.X[:], Soil.Y[:], None, 'PSoil' )


SvF.Legend = False
SvF.X_lim   = [-110,110]
SvF.Y_lim   = [-110,80]


Draw P68;LC:gray;LW:1;MS:0;LSt:dotted P69;MS:0 P70;MS:0 MERID ¥
    Shore;LC:blue;LW:1;MS:0;LSt:solid Brus;LC:red;MS:0 Bnor_fin;LC:red;MS:0 Town;MS:6;LW:0;MC:red;MEW:2;MEC:black ¥
    Lakes;MC:w;LW:0;MS:7;MEW:1;MEC:blue Depos;MC:w;LC:w;MS:6;M:s;MEW:1;MEC:black Bott;M:~;MS:6;MEC:r SedAcc PSoil;MC:g;M:X;MS:9;MEW:1;MEC:w


SET:  X = [ -100, 100, 2 ]
      Y = [ -100, 100, 2 ]


SvF.curentTabl = None
Param: pPOL ( X, Y )
Param: SoilXY ( X, Y )
Param: WaterXY ( X, Y )

```

```

SoilXY.grd[:, :] = fonNiSoil # 800*hSoil
WaterXY.grd[:, :] = 1

```

```

DataMarkerSize = 2
SvF.Ylabel_x = 0.05
SvF.Xlabel_x = 0.96
SvF.Ylabel_y = 0.90
SvF.Xlabel_y = 0.07
SvF.X_lim = [-100, 100]
SvF.Y_lim = [-100, 100]
SvF.xaxis_step = 50
SvF.yaxis_step = 50
SvF.Xsize = 5.
SvF.Ysize = 4.1

```

```

#SvF.X_lim = []
#SvF.Y_lim = []

```

```

SvF.Legend = True
from matplotlib import ticker
SvF.locator=ticker.LogLocator(base=2.0)

```

```

for t in range (1946, 2031, 1):
    for x in X.NodS :
        for y in Y.NodS :
            xv = X.Val[x]
            yv = Y.Val[y]
            dNik = rNik(xv, yv)
            dZap = rZap(xv, yv)
            Dep = 0
            DepInsol = 0
            DepSol = 0
            if dNik < rmax :
                Dep += DEPOSITNik(xv, yv, t)
                DepInsol += DEPOSITinsolNik(xv, yv, t) #fNik(YY) * fP_RNik( dNik ) *fPfiNik(fiNik(xv, yv)) * 100
                DepSol += DEPOSITsolNik(xv, yv, t)
            if dZap < rmax :

```

```

        Dep      += DEPOSITZap(xv, yv, t)
        DepInsol += DEPOSITinsolZap(xv, yv, t)      #fZap(YY) * fP_RZap( dZap ) *fPfiNik(fiZap(xv, yv)) * 1000
        DepSol    += DEPOSITsolZap(xv, yv, t)
    pPOL.grd[x, y] = Dep
    SoilXY.grd[x, y] += ( DepInsol * notSnow - leaching.grd * SoilXY.grd[x, y]*800*hSoil ) / (800*hSoil)
    WaterXY.grd[x, y] += (DepSol+leaching· SoilXY.grd[x, y]· 15-precip· WaterXY.grd[x, y]) / (H+precip)#15 = (Catch(b)-S(b))/Catch(b)

# if t % 10 == 0 :
    if t in [1980, 2005, 2018, 2030] :
        pPOL.V.draw_name = ''      #В ы п а д е н и е Ni, м г / м 2 / г о д ' +str(t)
        SoilXY.V.draw_name = ' '   #Ni в почве, м г / к г (с у х . в е с а) ' +str(t)
        WaterXY.V.draw_name = 'Ni в воде, м г / м 2 ' +str(t)
        print (t)
#     pPOL.grd[:, :] = 0
    Draw pPOL:L:1, 2, 4, 8, 16, 32, 64, 128, 9999;MS:0 ¥
        P68:LC:gray;LW:1;MS:0;LSt:dotted P69:MS:0 P70:MS:0 MERID ¥
        Shore:LC:blue;LW:1;MS:0;LSt:solid Brus:LC:red;MS:0 Bnor_fin:LC:red;MS:0 Town:MS:6;LW:0;MC:red;MEW:2;MEC:black ¥
#     Lakes:MC:w;LW:0;MS:6;MEW:1;MEC:blue Depos:MC:w;LC:w;MS:6;M:s;MEW:1;MEC:black Bott:M:^;MS:6;MEC:r SedAcc
    PSoil:MC:g;M:X;MS:9;MEW:1;MEC:w

    Draw SoilXY:L:25, 50, 100, 200, 1000, 9999;MS:0 ¥
        P68:LC:gray;LW:1;MS:0;LSt:dotted P69:MS:0 P70:MS:0 MERID ¥
        Shore:LC:blue;LW:1;MS:0;LSt:solid Brus:LC:red;MS:0 Bnor_fin:LC:red;MS:0 Town:MS:6;LW:0;MC:red;MEW:2;MEC:black ¥
#     Lakes:MC:w;LW:0;MS:6;MEW:1;MEC:blue Depos:MC:w;LC:w;MS:6;M:s;MEW:1;MEC:black Bott:M:^;MS:6;MEC:r SedAcc PSoil:MC:g;M:X;MS:9;MEW:1;MEC:w

    Draw WaterXY:L:1, 3, 10, 30, 100, 300, 1000, 3000;MS:0 ¥
        P68:LC:gray;LW:1;MS:0;LSt:dotted P69:MS:0 P70:MS:0 MERID ¥
        Shore:LC:blue;LW:1;MS:0;LSt:solid Brus:LC:red;MS:0 Bnor_fin:LC:red;MS:0 Town:MS:6;LW:0;MC:red;MEW:2;MEC:black ¥
        Lakes:MC:w;LW:0;MS:6;MEW:1;MEC:blue #Depos:MC:w;LC:w;MS:6;M:s;MEW:1;MEC:black Bott:M:^;MS:6;MEC:r SedAcc PSoil:MC:g;M:X;MS:9;MEW:1;MEC:w

```

EOF