BoF-SvF

NiCu = 'Ni'

#SvF.Resources = [ "abc\_dell" ]

#SvF.useNaN = True

#ObjToReadSols = True

#SvF.solverOptVal ["linear\_solver"] = 'ma86'

SvF.DataPath = '../../\_DATA100-158/'

CVNumOfIter = 0

InitSols = 0 #True

ShoreErr = False

#RunMode = 'S&S'

RunMode = 'L&P'

SvF.max\_workers = 10

SvF.maxJobs = 3

#SvF.OptStep = [0,0,-0.0001,0, 0.0001, 0,0]

# **Модель описывает последствия добычи и переработки Ni комбинатом Печенганикель.**

# Рассматриваются **два источника** атмосферных выбросов городах: **Никель (Nik) и Заполярный (Zap)**.

latNik = 69.413; lonNik = 30.234 # lat - широта, lon - долгота

latZap = 69.425; lonZap = 30.822

# **Временные константы:**

StartYear = 1946 # Начало работы предприятия (выбросов) в Никеле и начало моделирования

EndYear = 2030 # Полное прекращение выбросов

StartZap = 1956 # Начало работы предприятия (выбросов) в Заполярном

ProgYear = 2050 # Конец прогноза

Set: t = [ StartYear, EndYear, 1 ] # Интервал моделирования.

# Шаг модели по времени - 1 год

rmax = 130 # Радиус области моделирования

rStep = 1 # Шаг модели по пространству

Set: Fi = [-180, 180, 10]

r = [ 0, rmax, rStep ]

r1 = [ 2\*rStep, rmax, rStep ] # без точки источника

r2 = [ 50, rmax-1, rStep ] #

Set: Prognose = [ StartYear, ProgYear, 1 ]

AzimutInit ( latZap, lonZap ) # Zap in the center

# **Расстояния и Углы** **до источников**

xyNik = LatLonToAzimut ( latNik, lonNik ) # шир, долг -> декарт

def fiNik(X,Y): return degrees ( np.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 ( np.arctan2 (Y-xyZap[1], X-xyZap[0]) ) # fi

def rZap(X,Y): return sqrt ( (Y-xyZap[1])\*\*2 + (X-xyZap[0])\*\*2 ) # r

# **Модель атмосферного переноса:** выпадение - мультипликативное функция из трех частей:

# - зависимость от направления от источника до точки выпадения

# - зависимость от расстояния от источника до точки выпадения

# - зависимость от интенсивности выбросов (эмиссии) источника

############################################### Р **ROSE**

Select Fi, RosePol AS Pfi from NikRosePol37(Fi).sol

Var: Pfi ( Fi ) > 0.4; Type = Cycle # Роза ветров (одинаковая для обоих источников) -

# функция направления на точку выпадения (угла Fi).

EQ: ∫( -180, 180, Pfi(Fi)\*d(Fi)) == 360

# Pfi(-180) == Pfi(180) #Cycle

# **источник в г.НИКЕЛЬ (Nik)**

Var: EmNik ( t ) > 0 # ЭМИССИЯ в Никеле т/год #2

# Интенсивность выпадения как функция расстояния до Никеля

Var: PrNik ( r ) > 0; PrNik(0) = PrNik(rStep); PrNik( rmax ) = 0; d/dr(PrNik(r)) <=0

DepNik ( r ) > 0; DepNik( rmax ) = 1 # доля, выпадающая в круге r

EQ: d/dr1(PrNik(r1)\*r1) <= -0.003 \* PrNik(r1)\*r1

d/dr2(PrNik(r2)\*r2) >= -0.2 \* PrNik(r2)\*r2

DepNik (r) = ∫( 0, r, PrNik(r1)\*r1\*d(r1)) \* 2\*pi

# ∫( 0, rmax, d(r)\*PrNik(r)\*r) \* 2\*pi == 1

# **источник в г.ЗАПОЛЯРНЫЙ (Zap)**

Var: EmZap ( t ) > 0 # ЭМИССИЯ в Заполярный т/год #2

Var: PrZap ( r ) > 0; PrZap(0) = PrZap(rStep); PrZap( rmax ) = 0; d/dr(PrZap(r)) <=0

DepZap ( r ) > 0; DepZap( rmax ) = 1

EQ: d/dr1(PrZap(r1)\*r1) <= -0.003 \* PrZap(r1)\*r1

d/dr2(PrZap(r2)\*r2) >= -0.2 \* PrZap(r2)\*r2

DepZap (r) = ∫( 0, r, PrZap(r1)\*r1\*d(r1)) \* 2\*pi

# ∫( 0, rmax, d(r)\*PrZap(r)\*r) \* 2\*pi == 1

############################################### Оба Nik и Zap

EMS2a = Select Year as t, NiNi as EmNikZap from 2Source2030.xlsx # where t > 1976

Var: EmNikZap ( t ) > 0 ; < 700; EmNikZap ( t ) = EmNik ( t ) + EmZap ( t )

Set: t1946\_1955 = [StartYear,**StartZap-1**,1]

tMono = [1955,1976,1]

tMonoNik = [1946,1976,1]

# t1980\_2010 = [1955,2014,1]

t1955\_2014 = [1955,2014,1]

t2010\_2014 = [2010,2014,1]

t2016\_2019 = [2016,2019,1]

t2022\_2030 = [2022,EndYear,1]

t1946\_2009 = [StartYear,2009,1]

EQ:

EmZap( t1946\_1955 ) = 0

d/dtMono(EmZap(tMono)) > 1

EmNik( StartYear ) <= 10

d/dtMonoNik(EmNik(tMonoNik)) > 1

EmNikZap ( t2022\_2030 ) <= 20.

EmZap ( t1955\_2014 ) <= EmNik ( t1955\_2014 ) \* 1.2

d/dt2022\_2030 ( EmNikZap(t2022\_2030)) < 0

# **ВЫПАДЕНИЕ** от Никеля и Заполярного (mg/m2/year) в x, y, t

# Загрязнитель может быть растворимым или не растворимым.

Var: SolubleNik( r ) > 0.05; <.5; d/dr (SolubleNik) > 0 # доля растворимых загрязнителей

SolubleZap( r ) > 0.05; <.5; d/dr (SolubleZap) > 0

# Выпадение загрязняющих веществ (в растворимой и нерастворимой форме) от двух источников

def Depos\_sol (x,y,t): # Суммарное выпадение растворимых mg/m2/year #4b

dep = 0 #precip \* fonWater /1000 \* 0.4

if t <= EndYear:

if rNik(x,y) <= rmax : dep += Pfi(fiNik (x,y)) \* PrNik(rNik(x,y)) \* EmNik(t) \* SolubleNik (rNik(x,y))

if rZap(x,y) <= rmax : dep += Pfi(fiZap (x,y)) \* PrZap(rZap(x,y)) \* EmZap(t) \* SolubleZap (rZap(x,y))

return dep \* 1000

def Depos\_insol (x,y,t): # Суммарное выпадение нерастворимых mg/m2/year #4a

if t > EndYear: return 0

dep = 0

if rNik(x,y) <= rmax : dep += Pfi(fiNik (x,y)) \* PrNik(rNik(x,y)) \* EmNik(t) \* (1-SolubleNik (rNik(x,y)))

if rZap(x,y) <= rmax : dep += Pfi(fiZap (x,y)) \* PrZap(rZap(x,y)) \* EmZap(t) \* (1-SolubleZap (rZap(x,y)))

return dep \* 1000

CONS = Select nP, X,Y, Catch, S as SQw, Catch as SQs, lake1, peat1 from POINT.xlsx

CONS.SQw[:] += (CONS.Catch[:]-CONS.SQw[:]) \* (CONS.lake1[:]+CONS.peat1[:])/100

CONS.SQs[:] -= CONS.SQw[:]

# Всего в регионе моделирования рассматривается **108 точек**.

Set: nP = [ , , 1] # nP = [ 1, 108, 1]

# Это точки, в которых имеются какие-либо измерения (данные)

# Предполагается, что в каждой точке (номер nP) находится водосбор.

# Параметры водосбора:

Param: X( nP ) # координата X (km)

Y( nP ) # координата Y (km)

Catch( nP ) # Площадь бассейна (km2)

SQw( nP ) # Площадь озера (km2)

SQs( nP ) # Площадь суши (km2) SQs(nP) = Catch(nP) - SQw(nP)

# Общие для всех водосборов параметры:

precip = 0.4 # осадки m/year

fonWater = 1.0 # фоновое загрязнение воды mkg/dm3 = mg/m3

fonSoilDW = 10 # фоновое загрязнение почвы mg/kg(сух.веса)

fonSedi = 20 # фоновое загрязнение донных отложений mg/kg(сух.веса)

depISout = 0.2 # Доля загрязнений уходящих в сток из водосбора

intoIS = 0.2 # доля растворимых, связываемая (в нерастворимые) в почве

Var: Snow <0.5; >0.1 # Доля выпадений, попадающая в озеро с таянием снега

notSnow; notSnow = 1-Snow # Доля выпадений, остающаяся на суши

# Для каждой точки nP определим **МОДЕЛЬ ВОДОСБОРА с озером - МОДЕЛЬ 3-ёх сред**: суша, вода и донные отложения.

# Растворимые и нерастворимы выпадения (mg/m2/year) на водоразделе nP

Var: depIS ( nP, t ) #>=0; # ВЫПАДЕНИЕ НЕРАСТВОРИМЫЕ mg/m2/year

EQ: depIS = Depos\_insol (X(nP),Y(nP),t)

Var: depS ( nP, t ) #> 0; # ВЫПАДЕНИЕ РАСТВОРИМЫЕ mg/m2/year

EQ: depS = Depos\_sol (X,Y,t)

# изменение запасов = выпадение на (не промерзшую) почву - выщелачивание

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

#  **ПОЧВА** Динамика запасов в почве

Var: hSoil > 0.005; #< 0.04 #35 # Слой в метрах

Var: S ( nP , t ) > 7.5 # кол-во загрязнителя в почве mg/m2 #5

leaching > 0.0015; # коэффициент выщелачивания:

# доля нерастворимых переходящая в растворимые

EQ: S(nP,StartYear) = fonSoilDW \* hSoil\*800 # начальное условие

# (800\*hSoil) - пересчет из содержания (в сух.весе) в запасы

d/dt ( S ) = (depIS(nP,t)+depS(nP,t)\*intoIS) \* notSnow - leaching \* S

# изменение запасов = выпадение на (не промерзшую) почву - выщелачивание

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

**#         ДОННЫЕ ОТЛОЖЕНИЯ** Динамика запасов

Var: B ( nP , t ) > 0 # кол-во в донных отложениях mg/m2 #6

botSedLeach > 0.00; botSedLeach = leaching # коэффициент выщелачивания донных отложений

EQ: B(nP,StartYear) = 0 # начальное условие

d/dt ( B ) = depIS(nP,t) \*(1+SQs(nP)/SQw(nP)\*Snow) \*(1-depISout) - botSedLeach\*B + fonSedi\*0.1 # Vsedim\*0.2=0.1

# изменение запасов = поступление за вычетом стока - выщелачивание + фон

# Предполагается, что донные отложения формируются из нерастворимых форм, попадающих непосредственно на поверхность озера, из нерастворимых форм, попадающих непосредственно на сушу в зимний период (и переносимых в озеро в половодье) и из фонового значения, отражающее доиндустриальные процессы выветривания коренных пород и выщелачивания почв.

# **ВОДА** Динамика содержания (концентрации) загрязнения в воде озера

# кол-во = water\*S\*H + depS\*Catch + leaching·S·SQs(nP) ...

# объем = S\*H + Catch\*proc

# концентр = кол-во/объем

Var: W ( nP , t ) > 0 # концентрация загрязнителя mkg/dm3 = mg/m3 #7

H > 0; < 2 #20\*precip # глубина m

EQ: W(nP,StartYear) =fonWater # начальное условие

d/dt( W ) = ( depS(nP,t)\*(SQw(nP)+SQs(nP)\*(Snow+notSnow\*(1-intoIS))) + leaching·S·SQs(nP) + botSedLeach·B·SQw(nP) \

– Catch(nP)·precip·(W-fonWater) ) / ( SQw(nP)\*H+Catch(nP)\*precip )

# изменение концентрации = поступление с выпадением, из почвы и из донных отложение - вынос со стоком

# Предполагается, что растворимые формы (за исключением задержанных в почве) попадают в озеро вместе со всеми осадками (precip = 0.40м/год), что все растворимые формы, образовавшиеся в почве в результате выщелачивания, добавляются в воду озера и что часть выносится со стоком.

**##############################################################** # **Проекторы** СВЯЗЬ С ИЗМЕРЕНИЯМИ

UseNaN = True

TDep = Select nP AS nPD, Year as tD, ROWNUM as nD, depNi as deposD, X as Xd,Y as Yd from EXP\_DEPOS.xlsx # **ВЫПАДЕНИЯ**

###### DEPOSITION mg/m2/год 1e-3 or t/km2/year or kg /km2 /year #####

Set: nD = [,,1]

Param: nPD (nD)

tD (nD)

Var: **deposD** (nD) > 0 # DEPOSITION mg/m2/год

EQ: deposD (nD) = depIS(nPD(nD),tD(nD)) + depS(nPD(nD),tD(nD))

#---------------------# mg/kg(сух.веса): soilD\***0.8**(кг.сух/литр)\*10\*10\*10 (mg/m3) \* hSoil(m) = soilD\*800\*hSoil(m) = S (mg/m2) **ПОЧВА**

TSoil = Select nP AS nPS, Year as tS, ROWNUM as nS, soilNi as **soilD**, X as Xs,Y as Ys from EXP\_SOIL.xlsx

**for i in TSoil.sR : TSoil.soilD[i] = max (TSoil.soilD[i],fonSoilDW)**

Set: nS = [,,1]

Param: nPS (nS)

tS (nS)

Var: soilD (nS) > 0 # СОДЕРЖАНИЕ mg/kg(сух.веса)

# FonsoilD (nS) > 0 # Фоновое значение, начальное условие

EQ: soilD \* 800\*hSoil = S ( nPS , tS )

#--------------------------------------------------------------------------------------------------------------- # **ВОДА**

TWater = Select X as Xw,Y as Yw, Year AS tW, Avg-TNi AS waterD, ROWNUM as nW, nP AS nPW, nP, CONn from EXP\_LAKE.xlsx

#MakeSets\_byParam nPW 7

CV: CV\_NumSets=7; CV\_Unit ='nPW'

Set: nW = [,,1]

Param: nPW (nW)

tW (nW)

Var: waterD (nW) > 0 # mkg/dm3 ВОДА

# wFon > 0

#EQ: waterD = **Depos\_sol** ( X(**nPW**(nW)), Y(**nPW**(nW)), tW(nW) ) / precip + wFon

EQ: waterD = W ( nPW , tW )

#**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**# **ДО SEDIMENTION**

# B(mg/kg-dw) \*0.25(kg-dw/dm3) \*10+3(dm3/m3) \*Vsedim(mm/year) 10-3(m/mm) = botSedD\*Vsedim\*0.25(mg/m2/year) = B (mg/m2/year) - поступление

TBott = Select nP AS nPB, Year as tB, ROWNUM as nB, Vsedim, delT, Ni\_0\_1 as botSedD, X as Xb,Y as Yb from EXP\_SEDIMNT.xlsx

#for ii in TBott.sR :

**# TBott.delT[ii] = min (TBott.delT[ii],2)**

Set: nB = [,,1]

Param: nPB (nB)

tB (nB)

Vsedim(nB) # mm/year

delT (nB)

Var: **botSedD** (nB) > 0 # СОДЕРЖАНИЕ mg/kg(сух.веса)

sedimMult > 0

EQ: botSedD(nB)\*Vsedim(nB)\* **sedimMult\*0.25** = ( B(nPB,tB(nB)) - B(nPB,tB(nB)-delT(nB)) ) / delT(nB)

#########################################################################################################

def TimeStep(Xi,Yi,ti,Catchi,Si, s,w,b):

SQsi = Catchi-Si

depSo = Depos\_sol ( Xi,Yi,ti )

depISo = Depos\_insol ( Xi,Yi,ti )

Bo = b + depISo \* (1 + SQsi/Si\*Snow )\*(1-depISout) **-botSedLeach**\*b +fonSedi \* 0.1

So = s \* (1 - leaching ) + (depISo+depSo\*intoIS) \* notSnow

# Wo = w + ( depSo\*(Si+SQsi\*(Snow+notSnow\*(1-intoIS)))+leaching·s·SQsi + botSedLeach·b·Si -Catchi\*precip\*w )/(Si\*H+Catchi\*precip)

Wo = w + ( depSo\*(Si+SQsi\*(Snow+notSnow\*(1-intoIS)))+leaching·s·SQsi + botSedLeach·b·Si -Catchi\*precip\*(w-fonWater) )/(Si\*H+Catchi\*precip)

return depSo, depISo, Bo, So, Wo

def RunModel ( start, end ) :

if start == StartYear :

S.grd[:,0] = fonSoilDW \* hSoil\*800 #fonSoil

W.grd[:,0] = fonWater

B.grd[:,0] = 0

for bi in nP.NodS:

bv = nP.Val[bi]

for tv in range (start, end+1, 1) :

ti = tv - StartYear

# print (ti)

**depS.grd[bi,ti],depIS.grd[bi,ti],Bo, So, Wo = TimeStep(X(bv),Y(bv),tv, Catch(bv),SQw(bv), \**

**S(bv,tv),W(bv,tv),B(bv,tv))**

if tv < end :

S.grd[bi,ti+1] = So

W.grd[bi,ti+1] = Wo

**B.grd[bi,ti+1] = Bo**

if (InitSols):

Task.ReadSols ()

# DrLake (71)

RunModel ( StartYear, EndYear )

#Draw **depIS**

#Draw **depS**

#Draw B

#Draw soil

#Draw water

for n in nD.NodS:

nv = nD.Val[n]

deposD.grd[n] = depIS(nPD(nv),tD(nv)) + depS(nPD(nv),tD(nv))

for n in nS.NodS:

nv = nS.Val[n]

soilD.grd[n] = S ( nPS(nv) , tS(nv) ) / (800\*hSoil)

for n in nB.NodS:

nv = nB.Val[n]

# botNi.grd[n] = ∫( tB(nv)-delT(nv), tB(nv), d(T)\*bNi(bB(nv),T) ) / delT(nv) / (Vsedim(nv)\*0.25)

# botSedD.grd[n] = ∫( tB(nv)-delT(nv), tB(nv), d(t)\*botSed(nPB(nv),t) ) / delT(nv) / (Vsedim(nv)\***sedimMult\***0.25)

botSedD.grd[n] = ( B(nPB(nv),tB(nv)) - B(nPB(nv),tB(nv)-delT(nv)) ) / delT(nv) / (Vsedim(nv)\***sedimMult\*0.25)**

**# ( botSed(nPB,tB(nB)) - botSed(nPB,tB(nB)-delT(nB)) ) / delT(nB)**

# botSedD(nB)\*Vsedim(nB)\* **sedimMult\*0.25** = ∫( tB(nB)-delT(nB), tB(nB), d(t)\*botSed(nPB,t) ) / delT(nB) /()

# Draw botSedD

Task.SaveSols ()

**Obj:** waterD.MSDrel(2.0) + 0.0002\*deposD.MSD\_no\_mu() + Penal[4]\*EmNikZap.MSD\_no\_mu() + Penal[5]\*soilD.MSDrel\_no\_mu(30.0) \

+ Penal[7] \* Pfi.MSD\_no\_mu() +Penal[6]\*botSedD.MSDrel\_no\_mu(40) \

+ Pfi.ComplCyc0E (Penal[0]) + DepNik.Compl (Penal[1]) + EmNik.ComplMean2 (Penal[2]) \

+ DepZap.Compl (Penal[1]) + EmZap.ComplMean2 (Penal[2]) \

+ SolubleNik.ComplMean2 (Penal[3]) + SolubleZap.ComplMean2 (Penal[3])

INCLUDE: ../Draw-09.mng

**EoF**