# 명왕성에서 atmospheric escape 의 열역학적 모델링

1조 박준희, 신명진, 김태경, 신지훈

### Introduction

#### 2015년 8월 논문

THE ASTROPHYSICAL JOURNAL LETTERS, 808:L50 (5pp), 2015 August 1 © 2015. The American Astronomical Society. All rights reserved.

doi:10.1088/2041-8205/808/2/L50

#### ON THE PROVENANCE OF PLUTO'S NITROGEN (N2)

KELSI N. SINGER AND S. ALAN STERN
Southwest Research Institute, 1050 Walmut Street, Suite 300, Boulder, CO 80302, USA; ksinger@boulder.swri.edu
Received 2015 May 30; accepted 2015 July 15; published 2015 August 3

#### ABSTRACT

 $N_2$  is abundant in Pluto's atmosphere and on its surface, but the supply is depleted by prodigious atmospheric escape. We demonstrate that cometary impacts could not have delivered enough N mass to resupply Pluto's  $N_2$  atmospheric escape over time; thus Pluto's  $N_2$  is likely endogenous, and therefore was either acquired early in its history or created by chemistry inside/on Pluto. We find that cratering could excavate a considerable amount of  $N_2$  to resupply the atmosphere against escape if the near-surface  $N_2$  reservoir is deep. However, we find that this process likely falls short of that necessary to resupply the atmosphere against escape by at local an order of magnitude. We conclude that either the escape of  $N_2$  from Pluto's atmosphere was on average much lower than the predictions for the current epoch, or that internal activity could be necessary to bring  $N_2$  to the surface and resupply escape losses. Observations made by the New Horizons spacecraft in mid-2015 will yield further constraints on the provenance and evolution of Pluto's surface and atmospheric  $N_2$ , and could reveal evidence for past or present internal activity.

Key words: comets: general – Kuiper belt objects: individual (Pluto) – planets and satellites: atmospheres – planets and satellites: individual (Pluto) – planets and satellites: surfaces

#### 1. INTRODUCTION: PLUTO'S N<sub>2</sub> ATMOSPHERIC LOSS DILEMMA

In addition to Pluto's surface ice composition being dominated by molecular nitrogen (Cruikshank et al. 2015), Pluto's atmosphere consists of a >90% mole fraction of N<sub>2</sub> (Yelle & Elliot 1997), with surface pressures estimated on the order of  $\sim \! 10~\mu bars$  (Lellouch et al. 2009). Models predict the current  $N_2$  escape rate is  $10^{27} - 10^{28}$  molecules s<sup>-1</sup>, or  $1.5 \times 10^{12-13}$  g yr<sup>-1</sup> (e.g., Erwin et al. 2013; Zhu et al. 2014; Johnson et al. 2015; Tucker et al. 2015). For comparison, the estimated global atmospheric mass of Pluto, based on pure N2 of 10  $\mu$ bars at 35 K, is  $\sim$ 3  $\times$  10<sup>16</sup> g, demonstrating that the entire atmospheric N2 reservoir will be lost in a tiny fraction of the age of the solar system. It is unknown if Pluto's atmosphere collapses over the course of Pluto's orbit (Hansen et al. 2015; Olkin et al. 2015 and references therein). If the atmosphere exists for as little as 20% of Pluto's orbit, the escape rate would be reduced by a factor of 5. However, the higher solar ultraviolet fluxes in the past may have produced higher escape rates by a factor of several (Johnson et al. 2015).

A linear extrapolation of the above escape rates indicates a total of  $7\times 10^{21-22}\,\mathrm{g}\,N_2$  has been lost over 4 billion years. This is equivalent to a condensed global  $N_2$  surface layer on Pluto

case with comets (e.g., Jessberger et al. 1989; Crovisier et al. 2006; Mumma & Charnley 2011), and this N has been primarily converted to N<sub>2</sub> through chemical processes and given access to Pluto's surface, this could provide enough total N<sub>2</sub> mass to supply the atmosphere against escape.

Where did Pluto's  $N_2$  originate from? To address this, we proceed as follows: in Section 2 we evaluate whether  $N_2$  delivered by comets to Pluto can resupply the nitrogen; in Section 3 we explore potential excavation of endogenous  $N_2$  by these same impactors. In Section 4 we summarize our findings and briefly discuss the potential role of geologic activity for atmospheric resupply.

#### 2. CAN DELIVERY BY COMETS-RESUPPLY THE N2?

#### 2.1. Predicted Impactor Populations

Using observations of current KBO populations (Schlichting et al. 2012), typical KBO impact velocities onto Pluto of ∼1-2 km s<sup>-1</sup> (e.g., Zahnle et al. 2003; Bierhaus & Dones 2015), and Pluto's cross section, one of us, Stern et al. (2015a), estimated ∼14,000 comets 1 km in diameter or larger would impact Pluto over 4 billion years (comparable to an earlier estimate by Durda & Stern 2000). The rates derived in Zahnle

#### 질소 감소의 딜레마

명왕성 대기의 탈출 속도를 고려했을 때 질소는 빠른 속도로 명왕성에서 탈출한 다. 그러나 실제로는 명왕 성 대기에 질소가 너무 많 다

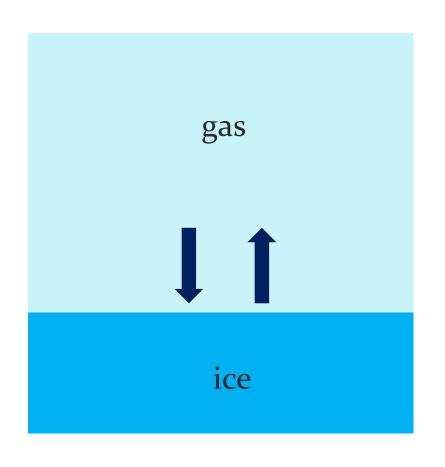
논문에서는 혜성의 충돌로 질소의 공급을 설명하려 했 지만 -> 실패

### Introduction

### 주제

명왕성 대기에서 나타나는 열역학적 현상을 모델링 하여서 atmospheric escape를 계산하여 보고, 현재의 명왕 성 질소 구성을 설명할 수 있는 가설 을 세워 본다.

### 배경지식: 명왕성의 대기



대기조성: 97% N<sub>2</sub> ~2.5% CH<sub>4</sub> ~0.5% CO

대기압: 0.65 ~ 2.4 Pa

표면온도: 약 43K

메탄, 질소, 일산화탄소의 표면 얼음과 대기와 평형을 이루고 있 다.

# Model Description

## atmospheric escape Exosphere (외기권) exobase gas phase radiative equilibrium equilibrium ice

#### **Model Algorithm**

- 1. 명왕성에서의 태양상수를 구하고, 복사 평형을 통해 표 면온도를 구한다.
- 2. 표면 온도에 해당하는  $N_2$  saturation vapor pressure를 구해 표면압력을 구한다.
- 3. hydrostatic을 가정하여 pressure profile 을 구한다.
- 4. exobase의 위치를 찾고, atmospheric escape rate을 구한다.

## 중요 가정

- 1차원 모델으로, 표면 전체의 평균적인 상태만을 나타낸다.
- 모든 현상은 큰 시간 규모에서 나타나므로 평형 상태라고 가정할 수 있다.
- 명왕성 대기 구성성분 중 대부분(97%)을 차지하는 질소 만을 고려한다. 질소는 다른 물질과 거의 반응하지 않는 다.

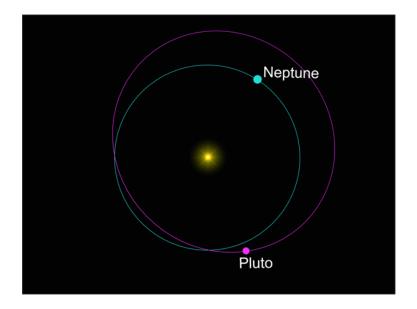


# 복사 평형 (radiative equilibrium)

$$\frac{S_0}{4}(1-A) = \epsilon \sigma T^4$$

*A* : albedo  $\sim 0.55$  (New Horizons)  $\epsilon$  : emissivity  $\sim 0.6$  (Hansen et al.)

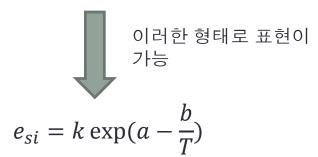
- 명왕성 대기는 기압이 매우 낮고, 또한 질소는 복사를 거의 흡수하지 않는다. 따라서 대기의 복사는 무시.
- 실제로는 메탄에 의한 온난화 효과가 있으나 표면온도에는 영향이 없음.



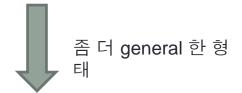
- 명왕성의 궤도는 이심률 이 큰 타원형이다. 따라서 태양 상수가 크게 변함.
- 모델에서 케플러 법칙으로 이를 계산함.

# 상평형 (phase equilibrium)

$$\ln \frac{e_{si}}{e_{s0}} = \frac{l_s}{R_v} \left( \frac{1}{T_0} - \frac{1}{T} \right) - \frac{c_i - c_{pv}}{R_v} \ln \frac{T}{T_0}$$



$$e_{si} = k \exp(a - \frac{b}{T})$$



$$e_{si} = 10^{A - \frac{B}{C + T}}$$

(Antoine equation)

#### Antoine Equation Parameters (P in mmHq, T in °C)

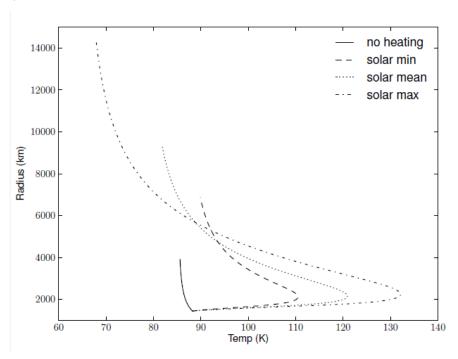
	•		•	-		
No.	A	В	С	Tmin [°C]	Tmax [°C]	$A - \frac{B}{B}$
1	5.73921	167.93	254.481	-218	-195	$P = 10^{A - \frac{S}{C + T}}$
2	6.93878	330.16	277.196	-195	-146	

실험적으로 질소의 Antoine equation parameter가 측정되 어 있다.

- 고체 질소와 기체 질소 사이의 상평형이다.
- C-C equation으로 구할 수 있지만 이렇게 낮은 온도의 경우 분자간 인력 등의 이유로 정확하지 않다.
- 실험 적인 값을 이용.

## Hydrostatic equilibrium

- 지구 대기의 경우 isothermal 이라 가정하고 압력을 구해도 된다. 그러나 명왕성의 경우 높이에 따른 온도 변화가 크므로 isothermal을 가정할 수 없다.



$$d\ln p = -\frac{g}{RT}dz$$

$$\int_{p_0}^{p} d\ln p = \int_{z_0}^{z} -\frac{g}{RT} dz$$

Model에서 Numerical 하게 적분하였다.

## Atmospheric escape

Thermal escape (**Jeans escape**)

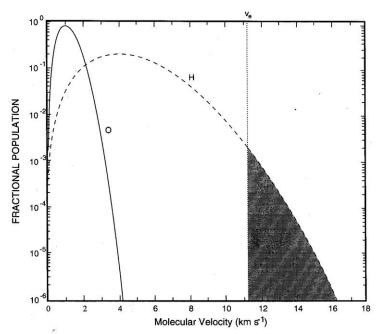
-> 분자들 중에 큰 열에너지를 가진 분자가 탈출하는 것.

Non-thermal escape

-> photochemical reaction, solar wind, ion escape .....

명왕성에서는 태양빛이 매우 약하므로 Non-thermal escape는 거의 없다고 볼 수 있다. 따라서 Jeans escape라고 가정.

## Atmospheric escape – Jeans escape



**Figure 1.6** Boltzmann distribution of velocities for a molecular ensemble of oxygen atoms and hydrogen atoms. Escape velocity  $v_e$  for earth also indicated.

- 행성의 중력을 벗어날 수 있는 속도를 탈출 속도라 한다. 탈출 속도는

$$V_e = \sqrt{\frac{2GM}{R}}$$

- 분자들의 속력 분포는 Maxwell distribution을 따르며 다양하다. 이들 중 탈출 속도를 넘어가면 행성을 탈출하게 된다.
- 그러나 탈출속도를 넘어도 다른 분
   자와 부딫히면 탈출 불가.
- 따라서 exobase 이상의 고도에서만 탈출 할 수 있다.
- **Exobase** 란 외기권의 제일 하층으로, 이 높이가 되면 분자 사이의 충돌이 거의 이루어지지 않는다.

## Atmospheric escape – Jeans escape

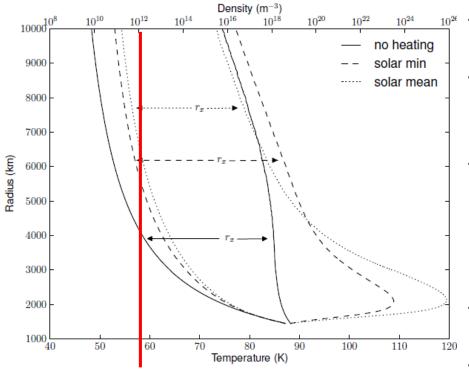
$$\phi = \frac{1}{4}n \ \bar{v} \cdot (1 + \lambda) \exp(-\lambda) \cdot 4\pi r^2$$
 
$$\lambda = U(r)/kT \qquad \bar{v} = \sqrt{\frac{8kT}{m\pi}} \text{ (Mean molecular speed)}$$

r: radius of exobase

-→ Escape rate 은 온도에 비례하고, n (number density)에 비례한다.

## Atmospheric escape – Jeans escape

#### Exobase level 구하기



- Exobase level은 온도와 number density의 함수.
- 그러나 여기서는 number density 만의 함수라고 가정한다.
- 온도는 linear하게 변하는 반면
  Number density는 exponential
  하게 변하기 때문에 number
  density의 영향이 더 크기 때문.
  - Number density < 10^12 인 곳을 exobase로 설정.
- (Erwin et al., 2013)에서 확인.

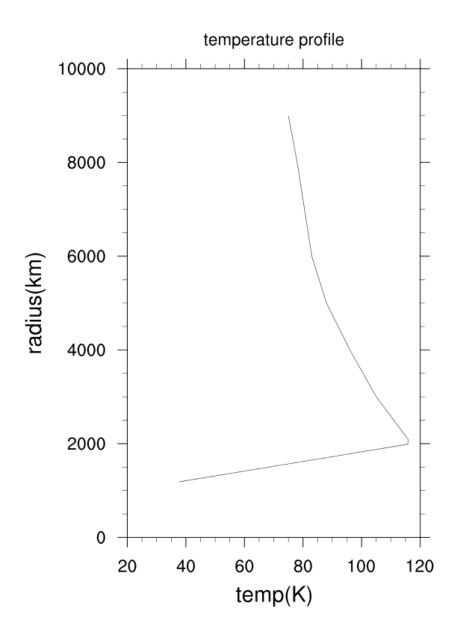
```
#include<stdio.h>
#include<math.h>
#define STEFAN 5.670367e-8
#define BOLTZMANN 1.38064e-23
#define GRAVITY 6.6738e-11
#define M_N2 4.6517e-26
#define R_N2 296.93
#define ALBEDO 0.50
#define EMISSIVITY 0.6
#define MASS 1.303e22
#define PERIOD 7804830000.0
                              //unit :second
#define DIST 39.54
#define ECC 0.24905
#define END 400.0
                               //end year
#define YEARTOSEC 31471986.0
#define S0 1361.0
#define RADIUS 1187.0e3
#define N_EXOBASE 1e12
static double n=2*M_PI/PERIOD; //mean motion
static double temp[8]=\{117,105,96,88,83,80.5,78,75\}; //temp profile
double kepler(double E, double M, int a)
          if(a<=0)
                    return M:
          return kepler(E-((E-ECC*sin(E)-M)/(1-ECC*cos(E))),M,--a);
double cal_solar(double t)
         double r=DIST*(1-ECC*cos(kepler(M,M,1000)));
          return S0/(r*r);
double surface_temp(double solar)
          return pow(solar*(1-ALBEDO)/(4*EMISSIVITY*STEFAN),0.25);
double temp_z(double z,double surf)
          double height=z/1.0e6;
          int height_int=(int)height;
          if(height_int<2){
                    double a=surf:
                    double b=temp[0];
                    return a+(b-a)*(z-RADIUS)/(2.0e6-RADIUS);
          else{
                    double a=temp[height_int-2];
                    double b=temp[height_int-1];
                    return a+(b-a)*(height-height_int);
double mol_speed(double T,double m)
          return sqrt(8*BOLTZMANN*T/m*M_PI);
double escape_rate(double n,double T,double z) //n : number density, z: exobase height
          double lambda=GRAVITY*MASS*M_N2/(BOLTZMANN*T*z);
         return 0.25*n*mol_speed(T,M_N2)*(1+lambda)*exp(-lambda)*4*M_PI*z*z;
double vapor_pressure(double T)
          return 1e3*pow(10.0,5.73921-167.93/(254.481+T-273.15))/750.06+0.35;
```

```
double p_z(double z,double T) //T:surface temp
          double p0=vapor_pressure(T);
          double z_iter, dz=1e3;
          double g;
          double tmp=0;
          for(z_iter=RADIUS;z_iter<=z;z_iter+=dz)</pre>
                     g=GRAVITY*MASS/(z_iter*z iter);
                     tmp += g*dz/(R_N2*temp_z(z_iter,T));
          return p0*exp(-tmp);
double n_z(double z,double T) //T:surface temp
          double p0=vapor_pressure(T);
          double z_iter, dz=1e3;
          double g;
          double tmp=0;
          for(z_iter=RADIUS;z_iter<=z;z_iter+=dz)</pre>
                     g=GRAVITY*MASS/(z_iter*z_iter);
                    tmp += 0.85*g*dz/(R_N2*temp_z(z_iter,T));
          return p0*exp(-tmp)/(M_N2*R_N2*temp_z(z_iter,T));
double find_exobase(double T) //T:surface temp
          double p0=vapor_pressure(T);
          double z_iter=RADIUS, dz=1e3;
          double g;
          double kkkk=log(N_EXOBASE*M_N2*R_N2/p0);
          double tmp=0;
          while(1)
                     g=GRAVITY*MASS/(z_iter*z_iter);
                     tmp += 0.85*g*dz/(R_N2*temp_z(z_iter,T));
                     if(kkkk+log(temp_z(z_iter,T))>-tmp)
                               return z_iter;
                     z_iter+=dz;
main()
          double t,year,t_step=YEARTOSEC;
          int t_iter,t_iter_last;
          FILE *fp;
          fp=fopen("result2.txt","w");
          t_iter_last=(int)(END*YEARTOSEC/t_step);
          double integral=0;
          for(t_iter=0;t_iter<t_iter_last;t_iter++)</pre>
                     t=t_iter*t_step;
                     double solar=cal_solar(t);
                     double T0=surface_temp(solar);
                     year=t/YEARTOSEC;
                     double esc=find_exobase(T0);
                     double rate=escape_rate(n_z(esc,T0),temp_z(esc,T0),esc);
                     if(year<=248.0){
                               integral+=rate*t_step;
                     fprintf(fp,"%g\t%g\n",year,rate);
          printf("%g\n",integral*M_N2/248);
          double z;
          double solar=cal_solar(PERIOD/2);
          double T0=surface_temp(solar);
          //printf("%lf\n",find_exobase(T0));
          for(z=RADIUS;z<9e6;z+=1e5)
```

fprintf(fp,"%g\t%g\n",z/1000.0,n\_z(z,T0));

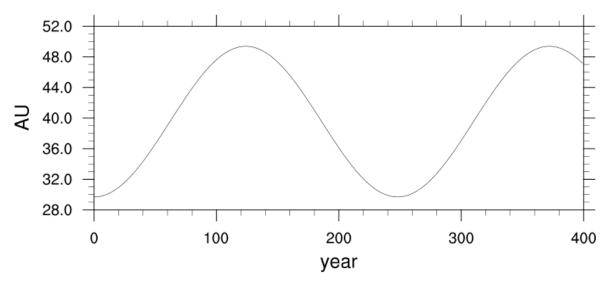
### code

# Model Result



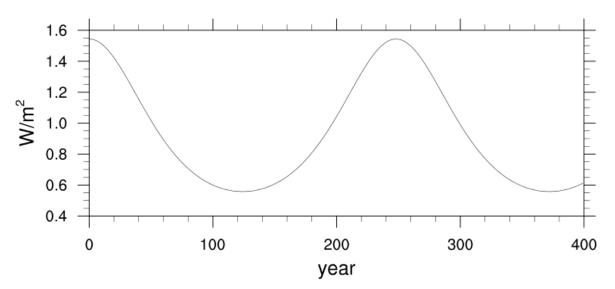
- Temperature profile을 구하기 위해서는 대기에서의 radiation과 convection등을 모두 고려해줘야 하므로 기존 연구 결과를 사용. (Erwin et al., 2013)
- temperature profile은
   hydrostatic equation으로 압력
   profile을 구하기 위해서 사용됨.

#### distance to PLUTO



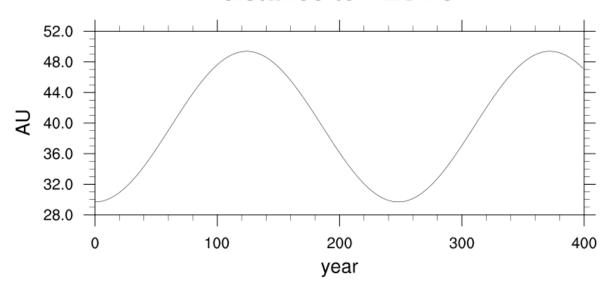
- 케플러 법칙을 통해 시간에 따른 거리를 계산

#### Solar constant of PLUTO

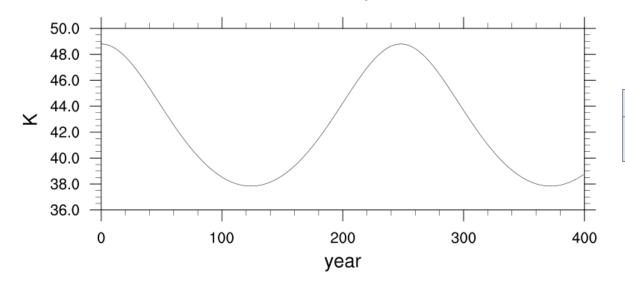


- 참고로 명왕성의 공전 주기는 248년이다.
- 거리에 따라서 태양상 수가 급격히 차이나는 것을 볼 수 있다.

### distance to PLUTO



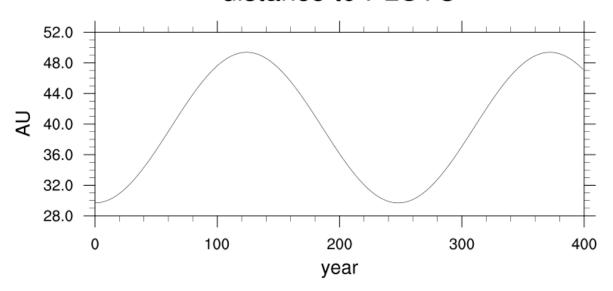
### surface temperature



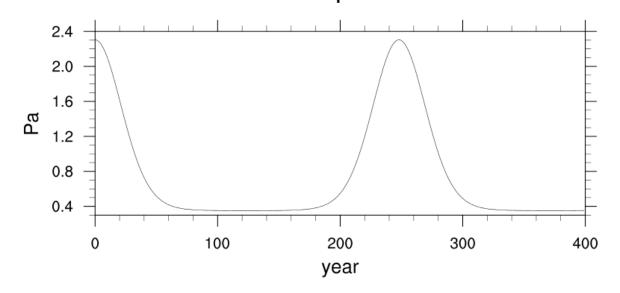
### 실제:

Surface temp.	min	mean	max
<u>Kelvin</u>	33 K	44 K (- 229 °C)	55 K

#### distance to PLUTO



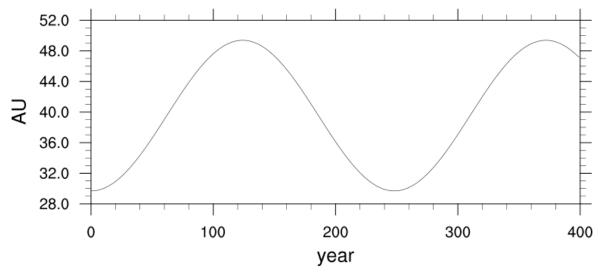
### surface pressure

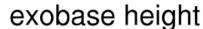


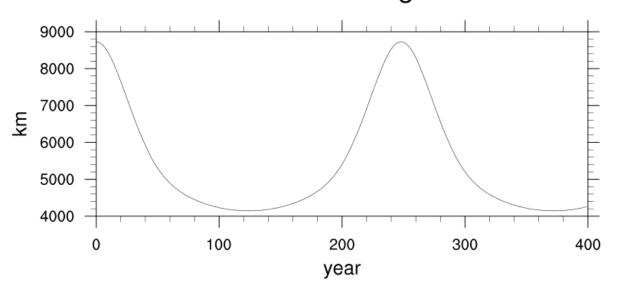
태양과의 거리가 가까울
 때 급격하게 압력이 증가
 한다. Vapor pressure는
 exponential 하게 증가하
 므로.

실제 대기압: 0.65 ~ 2.4 Pa



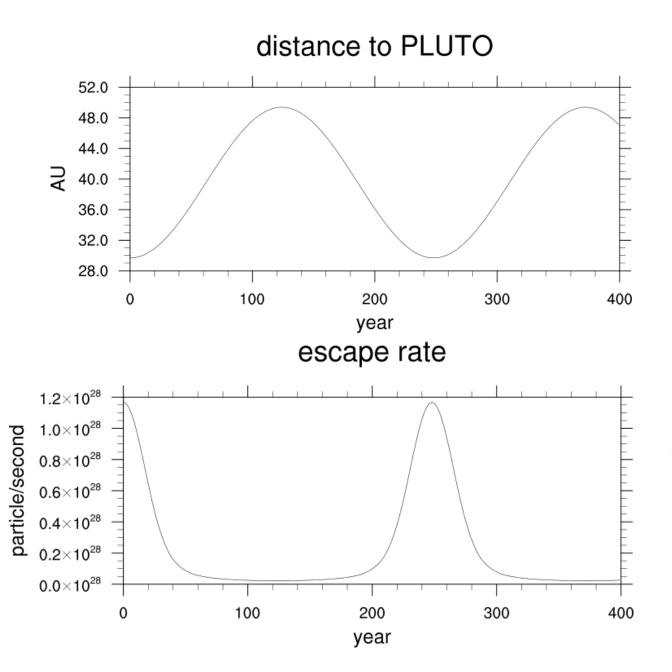




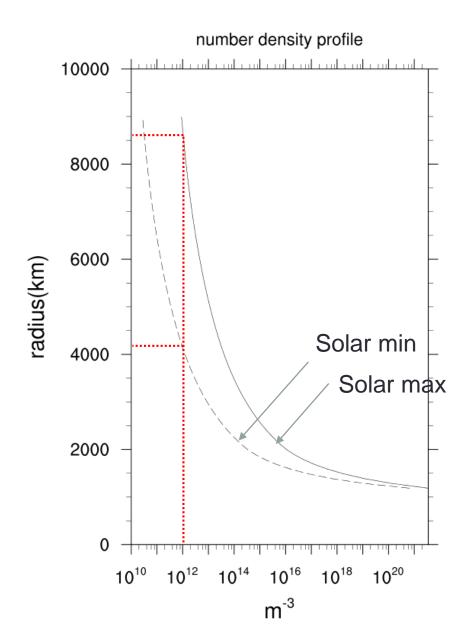


- 온도가 높으면 압력이 높 아지고, 이로인해 number density가 증가하므로 exobase height가 높아진다.

다른 GCM에서 예측 된 값: 4000~12000 km



탈출하는 분자 대부 분이 태양과 가까울 때 탈출함을 알 수 있 다.



Exobase height is much higher when solar insolation is max

Escaping rate 을 248년에 대해 적분해서 평균을 하면 = $3.65443 \times 10^9 \ kg \ yr^{-1}$ 

다른 모델들(PGCM 등)에서 예측된 값은 = $1.5 \times 10^{9-10} \ kg \ yr^{-1}$ 

로 거의 일치한다!

40억년 동안 명왕성에서 탈출한 질소는 약  $1.5 \times 10^{19}$  kg

현재 명왕성 대기의 질량은  $3 \times 10^{13} \text{ kg}$ 

-→ 명왕성에서 질소의 reservoir 가 빠르게 감소하고 있다.

### reference

- Singer, K. N., & Stern, S. A. (2015). On the provenance of Pluto's nitrogen (N2). *The Astrophysical Journal Letters*, *808*(2), L50.
- Erwin, J., Tucker, O. J., & Johnson, R. E. (2013). Hybrid fluid/kinetic modeling of Pluto's escaping atmosphere. *Icarus*, *226*(1), 375-384.
- Hansen, C. J., Paige, D. A., & Young, L. A. (2015). Pluto's climate modeled with new observational constraints. *Icarus*, *246*, 183-191.
- Hansen, C. J., & Paige, D. A. (1996). Seasonal nitrogen cycles on Pluto. *Icarus*, *120*(2), 247-265.