Semi-empirical Modeling of Radiolytic Production in the Mars Northern Polar Region

Keh-Harng Feng

A THESIS SUBMITTED TO

THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

GRADUATE PROGRAM IN PHYSICS AND ASTRONOMY

YORK UNIVERSITY

TORONTO, ONTARIO

APRIL, 2014

Abstract

A model to simulate the radiolytic production of various compounds in the Mars northern polar region was developed. Galactic cosmic rays are used as the source of particle radiation. Particle energy distributions are modeled semi-empirically with the energy deposition inside the ice layers computed using the Stopping and Ranging of Ions in Matter (SRIM) model simulation or a customized Bethe-Bloch formula. The rate of species production is derived from laboratory measurement. Seasonal variations of the ice layers are either obtained from Global Multiscale Mars Model (GM3) results or estimation based on observed quantities.

Preliminary results show that for species such as carbon monoxide, as much as 9,000 tons can be released into the atmosphere annually. Without the use of atmospheric transport models, a direct comparison to observations is not possible. As radiolysis can potentially explain the existence of compounds such as methane on Mars, the model warrants further investigation.

Dedication

This thesis is dedicated to Dr. John C. McConnell (1945 – 2013). Dr. McConnell was my supervisor through a large part of this project. Without his knowledge and guidance I would have been lost and led astray. He was diagnosed with brain cancer in May, 2013. Even though he was plagued with debilitating pain and suffering he was tireless in supporting my research. On July 29, 2013, Dr. McConnell passed away peacefully at home surrounded by family and friends. This thesis is dedicated to him as without him, it would not be a reality.

Acknowledgments

I want to thank my committee members Dr. Michael Daly and Dr. Tom McElroy. This thesis was completed under special circumstances out of anyone's control. Without Dr. Daly and Dr. McElroy's advice my progress would have been slow and this thesis unpolished. Furthermore, without their support, the continuation of my career in science would be doomed to uncertainty and most likely failure.

Many thanks to Drs. Jacek Kaminski and Stephen Beagley for their insights into atmospheric transport models. Special thanks to everyone in my group for sharing the same ship with me, particularly Farhanaz for pointing me in the right direction and Abdulla for our friendship. It was a tough time for us all. I also want to thank Mrs. Marlene Caplan from the York University Department of Physics. Her unparalleled enthusiasm and dedication to her craft made my life much easier.

I wish to thank my family for supporting my decision to pursue higher education at York University. I also wish to thank my girlfriend Ting for sticking with me and offering valuable inputs throughout the years. A special thanks goes to my close friends for being there. Thanks for making my life more interesting.

Last and most importantly, I want to thank my supervisor, Dr. John C. McConnell. His faith in me allowed me to continue my study in science and his insights led to the conception of this study.

Thank you all for everything. I made it because you were there.

Table of Contents

List of Tables

TABLE

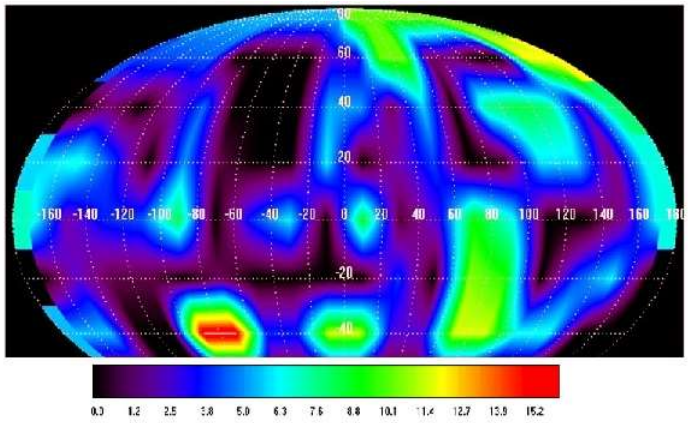
List of Figures

FIGURE

1. Introduction

The discovery of methane on Mars is currently a point of discussion. Some have suggested serpentinization of minerals such as olivine as a potential source [1]. Others postulate a biological origin [2]. Martinez et al. demonstrated a method to generate methane from water ice exposed to CO_2 gas using electrical current [3]. As Mars experiences periodic global dust storms and dust storms generate strong electric fields that point upward [4], it is possible that any free electrons (such as those generated by galactic cosmic ray (GCR) interaction with the atmosphere) can be accelerated by the electric fields and become a source of ionizing radiation for the production of methane on Mars. To illustrate, two maps for comparison are shown in Figure 1:

1 a)



1 b)

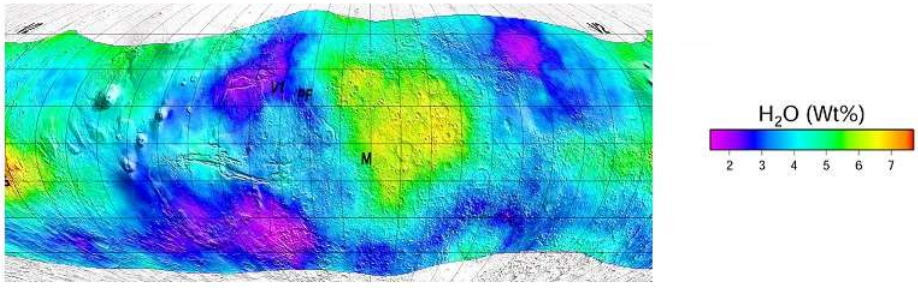


Figure 1 (a): The global methane column density on Mars during northern summer derived by Geminale et al. from PFX-MEX data [5]. **(b):** Global water concentration distribution on Mars derived from Odyssey Mars Gamma Ray Spectrometer (PDS Geosciences Node Mars Orbital Data Explorer ODY ELEMTS data product). Data is projected into a Mollweide projection to allow a better visual comparison.

The rough correlation between the water distribution and the methane distribution seems to suggest a link between the existence of near-surface water and the production of methane. While current data are insufficient to construct a complete model for the explanation of methane on Mars, the possibility of producing various chemical species at the surface by radiation is a subject not often explored for Mars. Although methane on Mars is not the focus of this research, it is a motivation to investigate the chemical effects of radiation on the Martian surface.

The goal of this work is to construct a general model for the radiolytic production of species in the Mars northern polar region. The model mechanics are covered in Chapter 2. Results and a brief discussion are included in Chapter 3. Future work on increasing model accuracy and further analysis are discussed in Chapter 4. The parameters used in the simulation and code of the model can be found in Appendices A and B respectively.

1.1 Radiolysis

Most matter exists in nature as groups of atoms. These atoms are bound together by the electrostatic force to form molecules. To rearrange the groups of atoms into other chemical compounds, it is necessary to break the original bonds. Bonds form from the interaction between orbiting electrons. Generally, energy is required to disrupt or strip the orbiting electrons and dissociate the molecules. Some of the common ways to introduce the dissociation energy in nature include lightning, absorption of photons (photolysis) and ion bombardment (radiolysis). It should be noted that in this thesis the term “radiolysis” is used solely to refer to dissociation

caused by particle radiation. Any form of dissociation caused by electromagnetic radiation shall be referred to as photolysis.

Photolysis and radiolysis are common to all celestial bodies in the solar system due to the abundance of photons from the sun and high energy ions from GCRs and ambient plasma [6]. While they may seem similar at first, there are some critical differences. As photolysis transfers energy through the absorption of photons, it is highly affected by the atmospheric conditions of the planet and is fairly limited in ground penetration. Hence, photolysis is often only considered for atmospheric chemistry. On the other hand, the rate of dissociation due to radiolysis is mainly affected by the amount of energy transferred to the target from the charged particles. Since a planet's atmosphere, such as that of Mars, is often not dense enough to stop a significant portion of GCRs (see Section 2.2), large numbers of ionizing particles can reach the surface of a planet. Lighter particles such as protons can also penetrate the surface to reach subterranean targets when their initial energy is high enough. This has been exploited to detect subsurface elements such as water-equivalent hydrogen on Mars [7] and is a potentially important factor to consider for surface and near-surface chemistry.

Radiolysis has been studied as a source of hydrogen generation using water [8]. Moreover, as many icy satellites have conditions suitable for the radiolytic production of observed chemical species it has been proposed as an alternative pathway for the production of chemical species in exposed surface ice on celestial bodies [9]. An example of this is the thin oxygen atmosphere detected on Ganymede. Radiolysis of surface water ice on Ganymede is suspected to be the cause of its existence [10]. In cases where some of the species observed in a planetary atmosphere

cannot be readily explained using atmospheric chemistry, such as the aforementioned methane on Mars, radiolysis may be a potential abiotic pathway for their production.

1.2 Mars Overview

The main constituents of the Mars atmosphere are CO_2 (95.3%), N_2 (2.7%) and Ar(2%) [11]. The abundance of CO_2 acts as a greenhouse gas in the lower atmosphere, causing a slightly higher observed mean surface temperature (240 K) than the computed equilibrium temperature from blackbody radiation (210 K) [12]. Seasonal variations around this temperature cause about one third of the atmosphere to condense onto the polar regions each year, forming the seasonal polar caps [13]. It should be noted that the temperature variation is not great enough to affect all constituents. For example, argon stays in the atmosphere even at the coldest point in the Martian winter. Combined with its inert nature, the relative stability compared to other constituents allows argon to be used as a reference in the mixing of the atmosphere [14]. On the other hand, the low atmospheric pressure generally does not allow the condensable constituents (CO_2 and H_2O specifically) to exist in liquid form on the planet's surface [12]. Seasonal exchanges at the surface, therefore, occur as direct sublimation and deposition. The main reservoir for the solid CO_2 and H_2O is the polar caps. The caps are formed as a thick layer of water ice that exchanges a small amount of water with the atmosphere each year (see Section 2.4). A smaller seasonal CO_2 ice layer lies on top (about 1 to 2 metres maximum in the northern polar region) during the winter. The northern and southern seasonal characteristics are different due to the large eccentricity of the Mars orbit. Since the southern winter occurs during the aphelion it is longer and colder compared to the north [15].

The unattenuated GCR flux for Mars should be essentially the same as that of the Earth's. However, the thin atmosphere of Mars provides relatively little protection to high energy particles from the GCRs compared to Earth (see Section 2.2). Furthermore, the planetary magnetic field of Mars is extremely weak [16]. The absence of a strong magnetic field means there is little magnetic deflection of the charged particles. The result is a much higher dose of particle radiation on the surface of Mars in comparison to Earth. On Mars, the annual surface dose rate from GCRs is about 60 to 120 mSv [17]. In contrast, the annual surface dose rate from GCRs on Earth is 0.3 to 1 mSv [18]. The amount of radiolytic products generated in the seasonal polar caps through GCR ion bombardment is, therefore, expected to be higher in the Martian radiation environment than in that of the Earth.

Radiolytic production experiments carried out by Pilling et al. on water ice, dry ice and their mixtures detected significant amount of CO, CO_3, O_3 and H_2O_2 being produced under laboratory conditions [19]. As the formants correspond to the ice found on Mars, these compounds are considered to be potential radiolytic products on Mars in the model. The details of the experiment and the way their results are incorporated into the model are described in Section 2.1.

2. Description of Model

The amount of radiolytic production, P, is dependent on the amount of energy deposited into the target from charged particles, ΔE. ΔE can be computed using the continuous slowing down approximation (CSDA) by integrating the stopping power of the material for the incoming charged particle over the distance the particle travels. For a vertically incident particle, this distance is the minimum between the thickness of the ice on the surface and the maximum CSDA range. However, for a particle coming in at an angle, the effective thickness is larger by a factor of 1/cos(θ). Therefore, one must consider the differential flux density of the incoming particles and integrate it over the range of possible incident angles in order to compute the energy deposition correctly. If the incoming radiation is isotropic, the total deposited energy, ΔE, is:

$$\Delta E = \int_{Emin}^{Emax} \int_{0}^{\frac{\pi}{2}} \int_{0}^{\min\left(R(E), \frac{h}{\cos\theta}\right)} R_{\theta}(E, \theta) \cdot S \, dx \, d\theta \, dE \qquad (1)$$

where

x = distance the charged particle traveled

θ = incident angle of the charged particle

E = initial energy of the charged particle

Emin, Emax = the range of initial energy for the charged particles

R(E) = CSDA range for a particle at energy E

h = vertical thickness of the target

R_θ = differential flux density (as a function of incident angle and initial energy) of the charged particles

S = average stopping power of the target for the current particle

Note that if there is azimuthal symmetry R_θ will have a factor of 2π once it is explicitly written out (see Section 2.2). It should also be noted that Beer's Law is not used here as a small distance would not result in many particles being absorbed. Additionally, the probability of a particle being absorbed can be considered as the probability of the particle losing all of its energy. In the case where the number of incident particles is large during the simulation period, the expected amount of energy deposited inside the medium can be derived simply from the average stopping power. The total amount produced can then be computed as long as the production rate, as a function of energy deposited, is known. The details on the components for total energy deposition and the production rate are presented below.

2.1 Production Rate

The production rate as a function of deposited energy is modeled using data obtained by Pilling et al. [19]. During the experiment, thin slabs of CO_2 and H_2O ice (0.4 μm and 0.5 μm respectively) received vertically incident 52 MeV $^{58}Ni^{13+}$ ions at a constant flux of $2*10^9$ counts cm^{-2} s^{-1}. An FTIR spectrometer was used at regular intervals to measure the column densities of various species in the parent ice. The observations were then fitted using equations derived by Pilling et al. [20] and Loeffler [21] to obtain the expression:

$$\frac{N_i}{N_{0,p}} = \frac{1}{1 + \frac{\sigma_{d,i}}{\sigma_{f,i}}} \cdot \left[1 - \exp\left(-\sigma_{d,i} \cdot F\right)\right] \quad (2)$$

N_i = the column density of the i^{th} species produced

$N_{0,p}$ = the column density of the parent species (ie: the ice)

$\sigma_{d,i}$ and $\sigma_{f,i}$ = dissociation and formation cross-sections of the produced species respectively (see Table 6 in Appendix A).

F = total fluence (the number of particles that have bombarded the target medium) of Ni ions.

Special notice should be taken for the fact that F is the total fluence of ions at the time of measurement and the values obtained from Equation 1 are only applicable to an ice sheet of the same thickness as that used by Pilling et al. In order to make use of their results the following assumptions and simplifications are made:

1. Radiolysis production is primarily dependent on the energy transferred to the target, not the type of particles.
2. Equation 1 is valid even on large time scales.
3. Ice layers of larger thickness than the ice sheets used by Pilling (referred to as a micro-layer in the model implementation) can be treated as the sum of M ice sheets, each of the same thickness as used in the Pilling experiment.

The total amount produced in the micro-layer can therefore be obtained as a summation of production occurring inside M Pilling sheets:

$$N_i = \sum_{j=1}^{M} \frac{N_{0,p} \cdot 1}{1 + \frac{\sigma_{d,i}}{\sigma_{f,i}}} \cdot \left[1 - \exp\left(-\sigma_{d,i} \cdot F_j\right)\right] \tag{3}$$

Note that F_j is the equivalent fluence of Ni ions going through the j^{th} sheet computed from deposited energy. If the micro-layer is reasonably thin, the energy deposition is approximately constant throughout all M sheets. F_j can be computed as

$$F_j = \frac{\Delta E}{M \cdot q} \tag{4}$$

where

ΔE = the amount of accumulated energy deposited in the micro-layer prior to its destruction.

q = the energy deposited by a single vertically incident $^{58}Ni^{13+}$ ion in a Pilling ice sheet.

The production inside a single micro-layer, therefore, simplifies to

$$P(\Delta E) = M \cdot N_{0,p} \cdot \left(\frac{N_{i,p}}{N_{0,p}} \left(\frac{\Delta E}{M \cdot q} \right) \right) \tag{5}$$

2.2 Cosmic Ray Surface Flux

The energy deposited in the micro-layers is assumed to be due to bombardment by galactic cosmic rays (GCRs). Since cosmic rays are primarily protons (~90%) [22] they are modeled as

galactic protons with energy ranging from 10 MeV to 50 GeV. To compute the accumulated energy deposition, knowledge about the number of particles reaching the surface and their energy levels is required. In the model all particles with the same initial energy and incident angle entering the ice layers are considered to be identical with each other. As a result, the total amount of energy deposition scales linearly with the surface flux of the particles.

Protons directly from GCRs are the primary protons. The primary proton flux at the top of the atmosphere is assumed to be isotropic and modeled using methods developed by Mizuno & Kamae [23]. Specifically the interstellar radiance is modeled using a combination of power law on rigidity and solar activity. The power law is as follows:

$$\mathrm{Unmod}(Ek) = A \cdot \left(\frac{R(Ek)}{(GV)} \right)^{a} \tag{6}$$

where A = 23.9 counts s^{-1} m^{-2} sr^{-1} MeV^{-1} and a = 2.83 as mentioned by Mizuno & Kamae. R(Ek) is the rigidity of the particle. This is computed as

$$R = \frac{P}{q} \tag{7}$$

where

$$P = \frac{\sqrt{\left(Ek + M_0 \cdot c^2\right)^2 - M_0^2 \cdot c^4}}{c} \tag{8}$$

is the momentum of a relativistic particle and q is the charge of the particle. In the case of a proton it is simply one positive electron charge.

The primary proton flux is also affected by solar activity. The solar wind decelerates galactic particles as they enter the system, hence the particle flux is inversely correlated with solar activity. The power law is thus combined with solar modulation using an energy shift [23][24]:

$$R = \text{Unmod}\left(Ek + Z \cdot e \cdot \phi\right) \cdot \frac{\left(Ek + M_0 \cdot c^2\right)^2 - \left(M_0 \cdot c^2\right)^2}{\left(Ek + M_0 \cdot c^2 + Z \cdot e \cdot \phi\right)^2 - \left(M_0 \cdot c^2\right)^2} \tag{9}$$

where Φ is the solar modulation parameter that ranges from 550 MV (minimum solar activity) to 1100 MV (maximum solar activity). It should be noted that Mizuno & Kamae have also computed the cutoff effect on galactic particles due to the Earth's geomagnetic field. However, due to the weak magnetosphere on Mars [16] this feature is not included in the model.

The flux density, I, is obtained by integrating the radiance over the hemisphere above the target surface:

$$I = \int_0^{2 \cdot \pi} \int_0^{\frac{\pi}{2}} \left[R \cdot \cos(\theta) \cdot \sin(\theta)\right] d\theta \, d\phi \tag{10}$$

R = Equation 9

θ = incident (zenith) angle

φ = azimuth angle

The cos(θ) term is a result of the cosine law, where the surface area receiving the flux is increased by a factor of 1/cos(θ) due to the inclination of the incident particles. To obtain the differential flux density, R_θ, required in Equation 1, the integrand in Equation 10 is integrated over only φ.

However, as the targets are located on the planetary surface, the primary proton flux will be attenuated by the atmosphere. This must be accounted for. The amount of attenuation for a vertically downward flux can be calculated using a method similar to the Beer-Lambert law for radiation intensity:

$$N = N_0 \cdot e^{-\frac{l}{u}} \tag{11}$$

where

N_0 and N are the particle fluxes before and after attenuation respectively.

u = the nuclear interaction length of the medium, defined as the mass thickness to attenuate the flux to 1/e of its original strength.

l = the integrated mass thickness or column density of the path the particles undertake.

As the Martian atmosphere is 95.32% CO_2 [11], the nuclear interaction length of the atmosphere is approximated using the value for CO_2 (88.9 g/cm^2) [25]. The integrated mass column density of the atmosphere at the surface can then be computed as follows:

$$l = \frac{P}{g} \tag{12}$$

P = mean surface pressure of Mars = 636 Pa [11]

g = gravitational constant of Mars = 3.71 m/s^2 [11]

It is important to note that Equation 12 is only valid for a vertically downward flux. For any particles with an incident angle greater than zero, the effective path length and the integrated mass thickness it has to pass through to reach the surface will be scaled by 1/cos(θ), where theta is the incident angle. The complete expression for the differential flux density of the primary proton at the surface is thus

$$R_{\theta} = 2 \cdot \pi \cdot R \cdot \cos(\theta) \cdot \sin(\theta) \cdot e^{-\frac{p}{g \cdot u \cdot \cos(\theta)}} \tag{13}$$

Beside the attenuation of the primary proton flux, the nuclear to nuclear interaction of the particles with atmospheric molecules will in turn generate a secondary proton flux. This is again modeled using the power law fits developed by Mizuno & Kamae ([23], Section 3.2). Due to the absence of a magnetic field on Mars, the cutoff PL for the highest magnetic latitude range ($0.9 <= \theta_M <= 1$) is chosen in order to model the downward secondary particle radiance with minimum magnetic cutoff effects. Since only the surface flux is of interest there is no upward secondary proton radiance. The expression is therefore:

$$R_{secondary} = \begin{cases} F_1 \cdot \left(\frac{Ek}{(GeV)}\right)^{-a} \cdot e^{\left(-\frac{Ek}{Ecut}\right)^{-a+1}}, & 100\ MeV \leq Ek \quad (14) \\ F_0 \cdot \left(\frac{Ek}{100\ MeV}\right)^{-1}, & 10\ MeV \leq Ek \leq 100\ MeV \quad (15) \end{cases}$$

where F_0, F_1, a and E_{cut} are model parameters listed in Table 7, Appendix A. Similar to the primary proton flux the differential flux density at different incident angles can be derived as:

$$R_{secondary\ \theta} = 2 \cdot \pi \cdot R_{secondary} \cdot \cos(\theta) \cdot \sin(\theta) \tag{16}$$

The total differential flux density that should be used in Equation 1 is thus $R_{\theta} = R_{primary\ \theta} + R_{secondary\ \theta}$. An important difference here is that the secondary flux is allowed to be generated from

any direction with an arbitrary distance to the source. The attenuation due to the atmospheric mass thickness is subsequently assumed to be independent of the incident angle of the secondary proton. The actual effect of the attenuation is assumed to be accounted for by the choice of parameters, as the secondary flux model is semi-empirical. Mizuno & Kamae show the model agrees with data obtained at high altitude on Earth (~37 km) (see [4], Figure 3). While the Earth's atmospheric pressure at that altitude is roughly the same as that on Mars' surface, the vertical integrated mass thickness for Earth is smaller by a factor of about three since the gravitational constant for Mars is about three times smaller than Earth's [11]. An estimate of the deviation of the secondary proton flux is carried out in Section 4.1.2.

2.3 Stopping Power

Once the surface flux spectrum of the galactic protons is obtained, the interaction between the incident protons and the ice must be considered. The electronic stopping power is taken to be the primary means of energy transfer from the particles to the medium. This is accomplished using the Stopping and Ranging of Ions In Matter model (SRIM, version 2013) [26] for particles between 10 MeV and 10 GeV. For particles with greater energy , the transfer is modeled using the Bethe-Bloch formula [27]:

$$\frac{dE}{dx} = \frac{K \cdot z^2 \cdot A}{Z \cdot \beta^2} \cdot \left[\frac{1}{2} \cdot \ln\left(\frac{2 \cdot \beta^2 \cdot \gamma^2 \cdot m_e \cdot c^2 \cdot T_{max}}{I^2} \right) - \beta^2 - 6.5 \cdot \delta + \frac{1}{8} \cdot \frac{T_{max}^2}{(\gamma \cdot M \cdot c^2)^2} \right] \quad (17)$$

β and γ are the fraction of particle speed to the speed of light and the Lorentz factor, $\gamma = 1/\sqrt{1 + \beta^2}$, respectively. The other constants are defined in [27], Table 1.

The Bethe-Bloch formula gives the average energy loss a charged particle experiences, dE, over a distance dx as a function of particle kinetic energy and medium density. It accomplishes this by modeling the probability of a collision with valence electrons in the atoms in the medium and the subsequent transfer of kinetic energy to the electrons. As such, the energy transfer characteristic only depends on the density of electrons within the medium structure and the mean excitation energy, I. T_{max} is the maximum energy transfer from the charged particle to a single electron in one collision. This is calculated as

$$T_{max} = \frac{2 \cdot m_e \cdot c^2 \cdot \beta^2 \cdot \gamma^2}{\left(1 + \frac{2 \cdot \gamma \cdot m_e}{M} + \left(\frac{m_e}{M}\right)^2\right)} \tag{18}$$

The δ term inside the square brackets of Equation 17 is the high energy density effect. This is computed using Sternheimer's method as outlined by Groom [28] (see also [27], Section 3.4). It is worth noting that the factor 6.5 in front of δ differs from that presented by Groom. This produces a better correlation to Groom's result (Figure 2) and also provides a transition with no discontinuities from the SRIM values at the boundary point, 10 GeV (Figure 3). In addition, Sternheimer's method involves computing different density effects at different energy ranges as follows:

$$\delta = \begin{cases} 2(\ln 10)x - \overline{C} & \text{if } x \geq x_1; \\ 2(\ln 10)x - \overline{C} + a(x_1 - x)^k & \text{if } x_0 \leq x < x_1; \\ 0 & \text{if } x < x_0 \text{ (nonconductors);} \\ \delta_0 10^{2(x - x_0)} & \text{if } x < x_0 \text{ (conductors)}, \end{cases} \tag{19}$$

where $x = \log_{10}(\beta^* \gamma)$.

Most of the parameters can be found in [28] for various media. However, note that for conductors, depending on the value of δ_0 large discontinuities can be introduced for the case $x < x_0$. To avoid this problem, the value of δ_0 is solved explicitly by considering the boundary case $x = x_0$. Rearranging one obtains

$$\delta_0 = 2 \cdot \ln(10) \cdot x_0 - C + a \cdot (x_1 - x_0)^k \tag{20}$$

which allows the computation of the density effect without introducing discontinuities for conductive mediums. In any case, since SRIM is used for lower energy particles, this deviation is unlikely to have any actual effects. The Bremsstralung radiation term is neglected as its contribution is less than 3% even at 100 GeV [27]. The shell correction (Barkas correction) is only important for low energy particles. Since the implementation is only used for particles with energy greater than 10 GeV, it is also neglected.

The result of the Bethe-Bloch implementation with positive muons (energy range 0.1 MeV to 100 GeV) traversing copper shows significant overestimation of mass stopping power at the low energy range (~five times larger) and general agreement with Groom's result at higher energy (Figure 2). It is unclear why such deviations exist. However, as low energy stopping power is computed using SRIM its overall effect on the model computation is most likely minimized. The deviations for SRIM values compared to experimental results are usually within 5% [29].

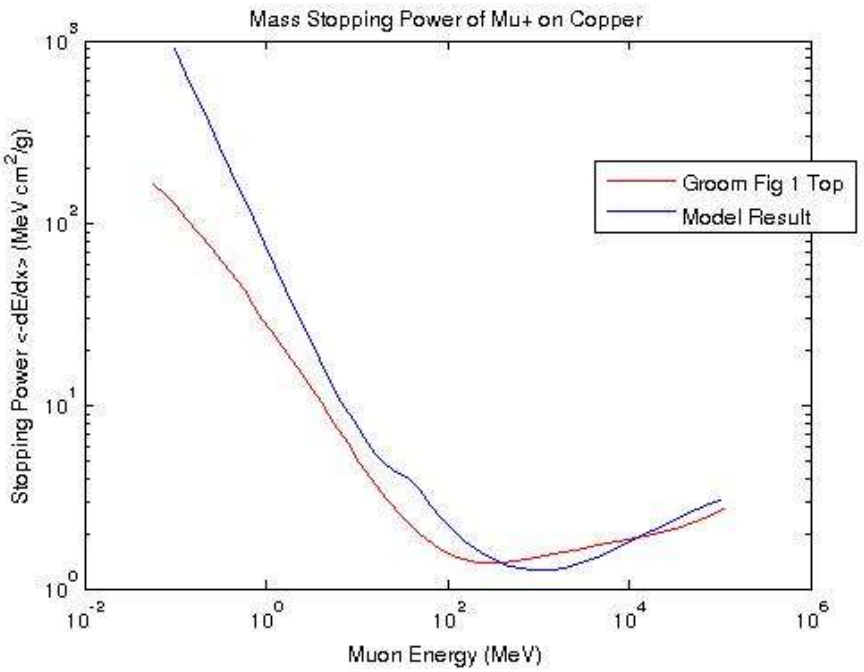


Figure 2: Model result comparison with Groom's values ([27], Figure 1 Top).

As the mass stopping power is dependent on the particle energy, the transfer of the particle energy as it travels can be approximated using forward Euler iteration. A particle is computed to have a deposition rate of dE/dx(E) at the beginning of the ice layer. This rate is approximately constant as the particle travels a small distance, dx. Its final energy at the end of the step is therefore E – dE/dx*Δx. To maintain accuracy a small step size is required. This increases required computation time significantly. A workaround is introduced by pre-computing the particle energy progression using a step-size of 0.1 mm and interpolating the points to produce a fit of particle energy as a function of distance traveled (Figure 4).

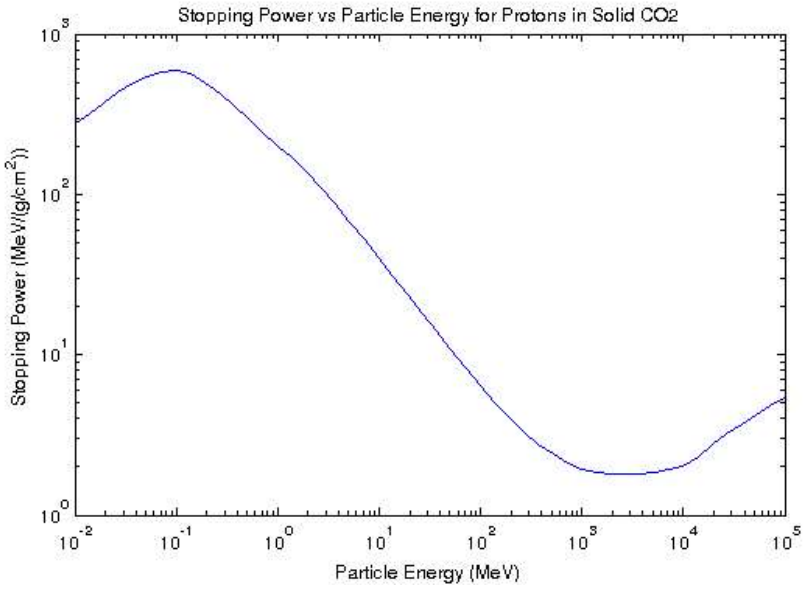


Figure 3: Mass stopping power for protons traversing CO_2. The stopping power for particles with energy between 0.1 MeV and 10 GeV is computed using SRIM version 2013. For particles with energy greater than 10 GeV it is computed using the implementation of Bethe-Bloch equation.

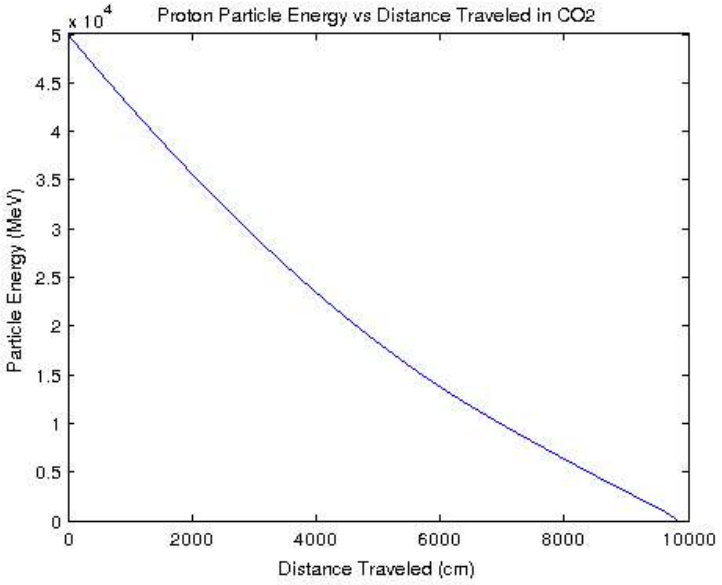


Figure 4. Proton kinetic energy as a function of distance traveled inside solid CO_2.

2.4 Ice Model

The surface proton flux and stopping power models allow the computation of the energy flux deposited into the ice at any given time. However, in order to use Equation 3, the accumulated energy deposited into the ice, ΔE, is required. This quantity represents the total amount of energy transferred to the ice prior to its sublimation and release of the products trapped within. Therefore, how the target ice layers change over time must be modeled. It is necessary to model the extent and thickness of both solid carbon dioxide and water at all times during the simulation. The carbon dioxide ice extent is determined from the seasonal neutron flux ratio (Figure 5) [30]. Neutron flux is produced as a result of GCR interacting with surface ice. Ratios greater than 1 indicate a changing amount of ice on the surface between different seasons. The north polar CO_2 ice extent is, therefore, determined to be 60º N and poleward. The CO_2 ice cover is then divided into six rings of equal width (5º latitude), each with the corresponding thickness progression as a function of time obtained from Ayodeji Akingunola's results using the Global Mars Multiscale Model (GM3) (Figure 6) [31].

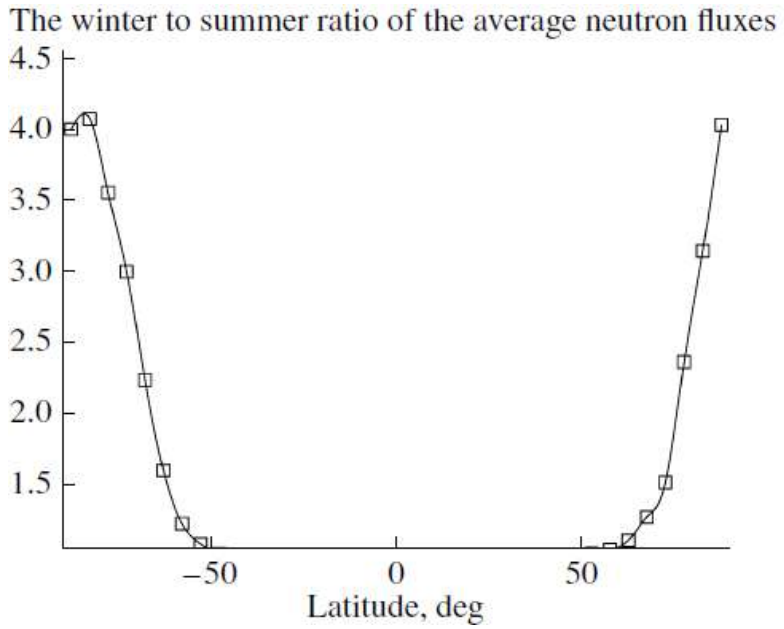


Figure 5: The ratio of the seasonal neutron flux between winter and summer. Referenced from [30].

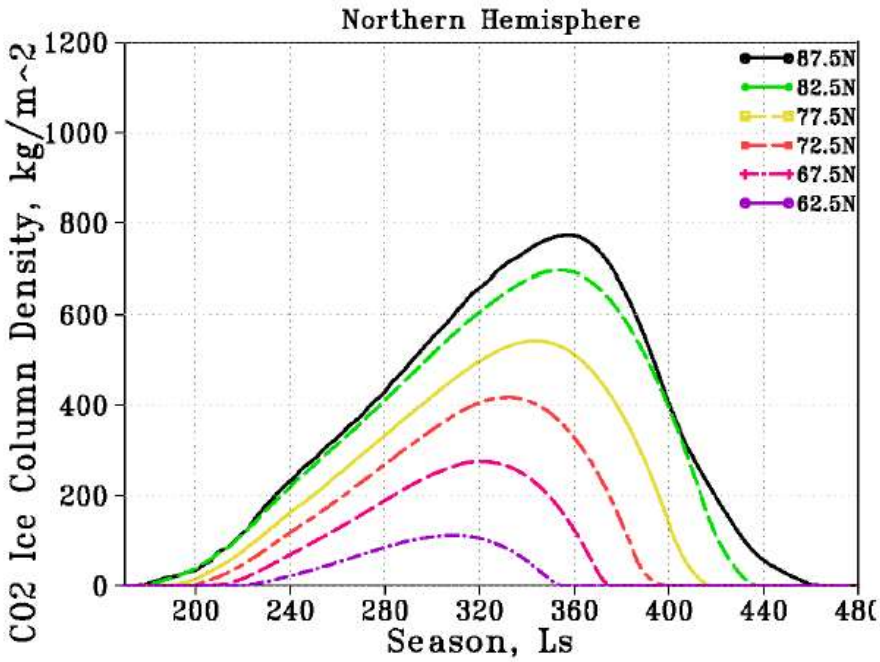


Figure 6. CO_2 ice thickness as functions of time and latitude. Figure referenced from [31].

For water ice thickness, due to the lack of observation/simulation data, a rudimentary model was constructed. Polar surface water ice is assumed to be covering the same area beneath the CO_2 ice. Similarly, the water ice cover is divided into six rings with equal width just like the CO_2 ice cover. As the release mechanism is only dependent on the sublimation/destruction of target ice layers, only the amount of water ice that is actually deposited/sublimed each year needs to be considered. Deposition and sublimation of water ice are restricted to only occur when the water ice itself is exposed (ie: when there is no CO_2 cover). During the period of exposure, the evolution of water ice thickness is modeled as one complete cycle of the cosine function. For the i^{th} ring the amount of water deposited on the surface can then be written as:

$$G_i(T) = D_i \cdot \left[\cos\left(\left(\frac{T - T_{1i}}{T_{2i} - T_{1i}} \right) \cdot 2\pi \right) + 1 \right] \cdot A_i \qquad (21)$$

where

G_i = the total amount of water from the ring (in kg/m^2)

T = time of the year represented by LS

D_i = half of the maximum thickness for the deposited water ice to reach

T_{1i} = the time of the year for the water ice on the i^{th} ring to first become exposed

T_{2i} = the time of the year for the water ice on the i^{th} ring to first become covered by CO_2 ice

A_i = the surface area of the i^{th} ring

The +1 inside the square bracket is to ensure the minimum thickness the deposited water ice can have is 0 instead of -1. Notice that all parameters are known except D_i, half of the maximum

thickness for the deposited water ice. To solve for D_i, note that Haberle derived the total amount of water sublimated into the atmosphere in the northern polar region from Ls = 80 to Ls = 120 to be about $7*10^{11}$ kg [32]. Therefore, the following can be written:

$$\sum_{i=1}^{6}\left[G_i(80) - G_i(120)\right] = 7\cdot 10^{11} \tag{22}$$

It is important to know that Equation 22 is still unsolvable, as there are six unknowns and only one equation. However, notice that for CO_2 ice, the top most ring (the sixth ring) achieved a maximum thickness about eight times that of the first ring. Equation 22 can be solved by assuming the water ice exhibits similar behavior and linearly interpolating for all of the D_i in between. The result is shown in Figure 7.

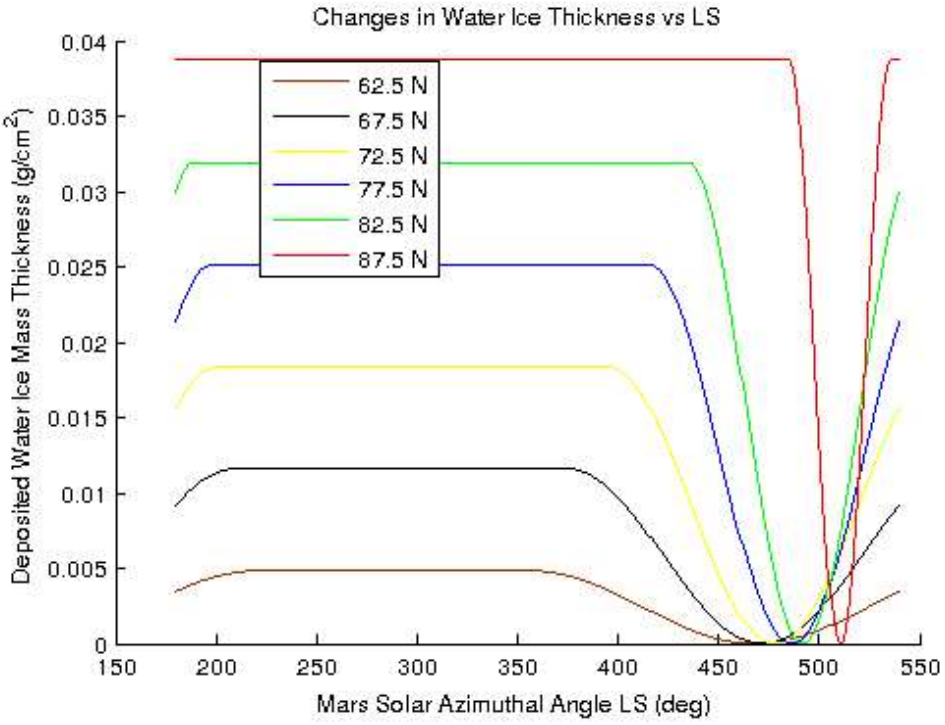


Figure 7: Deposited water ice mass thickness (g/cm^2) as a function of Mars solar azimuthal angle. Note that the water ice thickness stays constant when there is a CO_2 ice cover on top.

3. Result & Discussions

3.1 Simulation Result

A simulation was carried out using settings and parameters listed in Appendix A. Sublimation rates of the ice layers are shown in Figure 8 & 11. The surface emission fluxes of radiolytic products are shown in Figure 9 & 12. The cumulative amount of radiolytic products released into the atmosphere from the northern polar cap is shown in Figure 10 & 13.

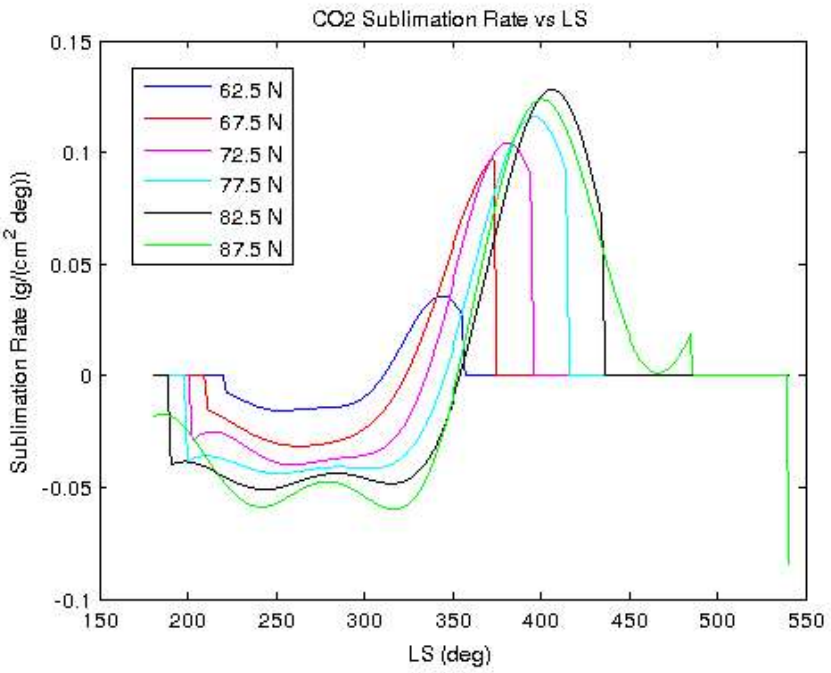
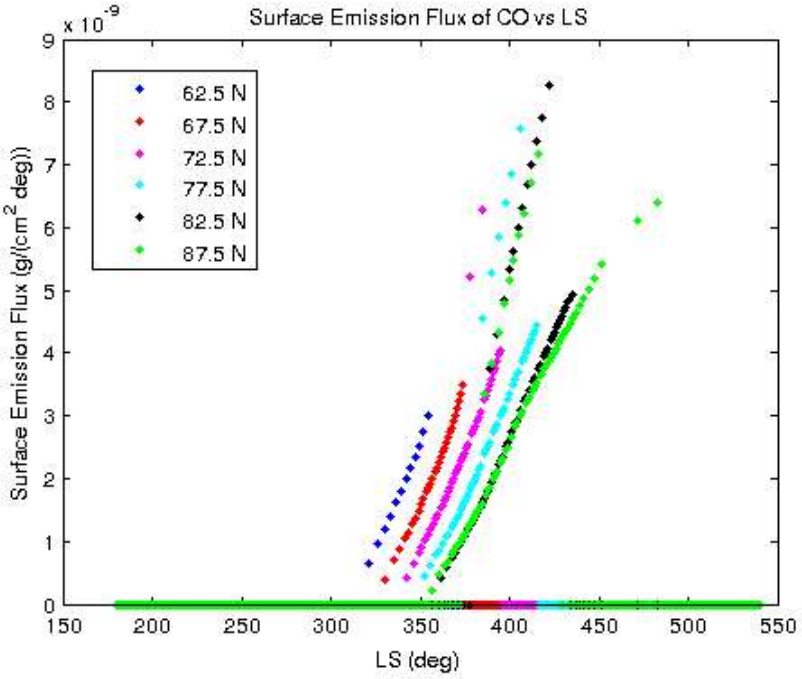
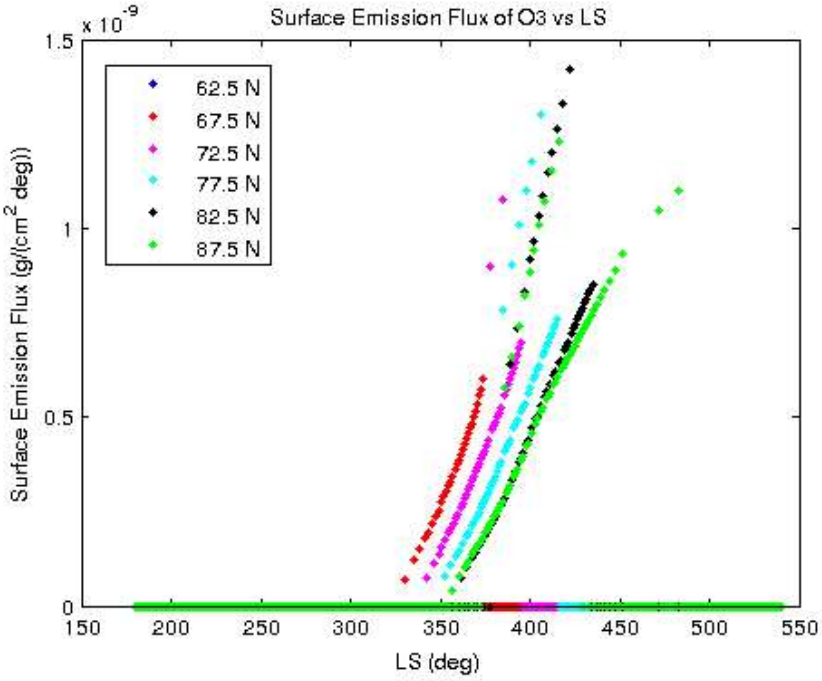


Figure 8: CO_2 sublimation rate per LS throughout the simulation year. Note that negative sublimation indicates deposition of CO_2 onto the surface.

9 a)


Surface Emission Flux of CO vs LS
x 10^-9
Surface Emission Flux (g/(cm^2 deg))
LS (deg)
62.5 N
67.5 N
72.5 N
77.5 N
82.5 N
87.5 N
0
1
2
3
4
5
6
7
8
9
150
200
250
300
350
400
450
500
550

9 b)


Surface Emission Flux of O3 vs LS
x 10^-9
Surface Emission Flux (g/(cm^2 deg))
LS (deg)
62.5 N
67.5 N
72.5 N
77.5 N
82.5 N
87.5 N
0
0.5
1
1.5
150
200
250
300
350
400
450
500
550

9 c)

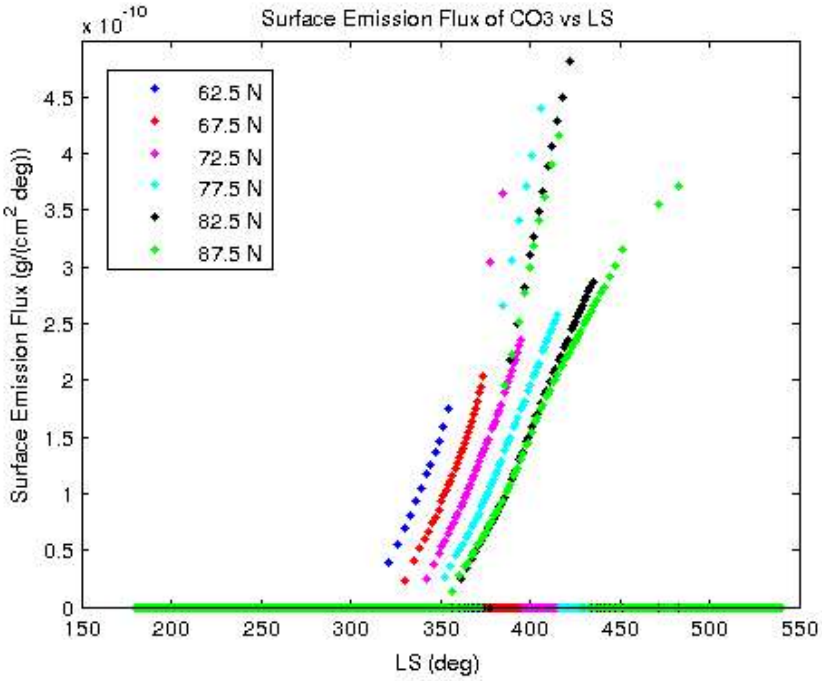
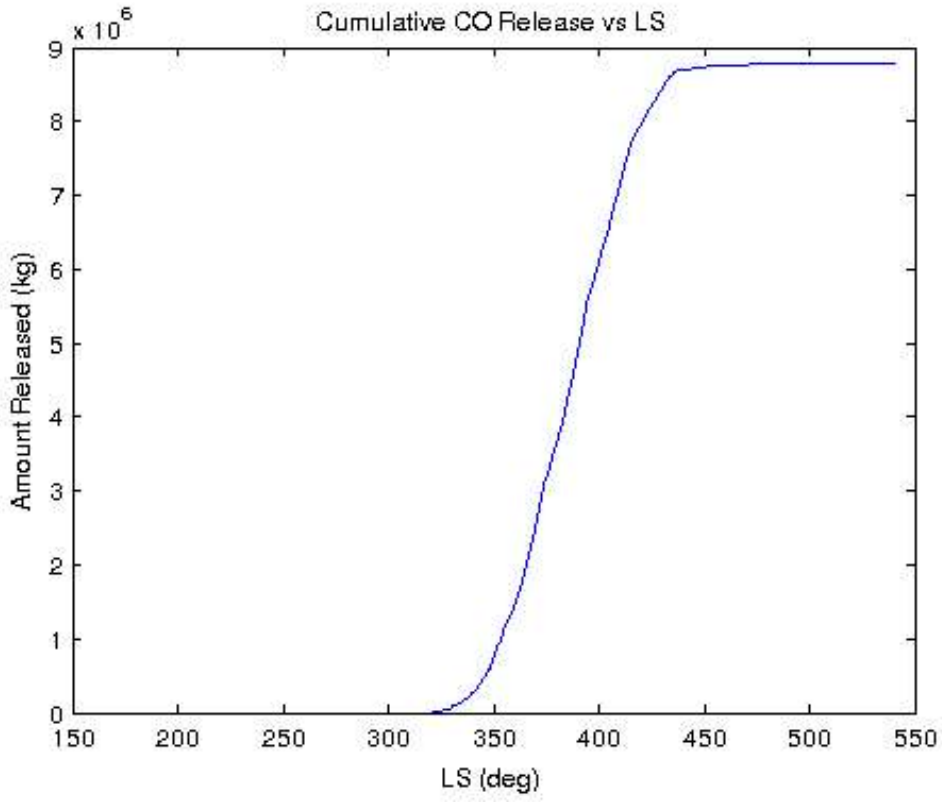
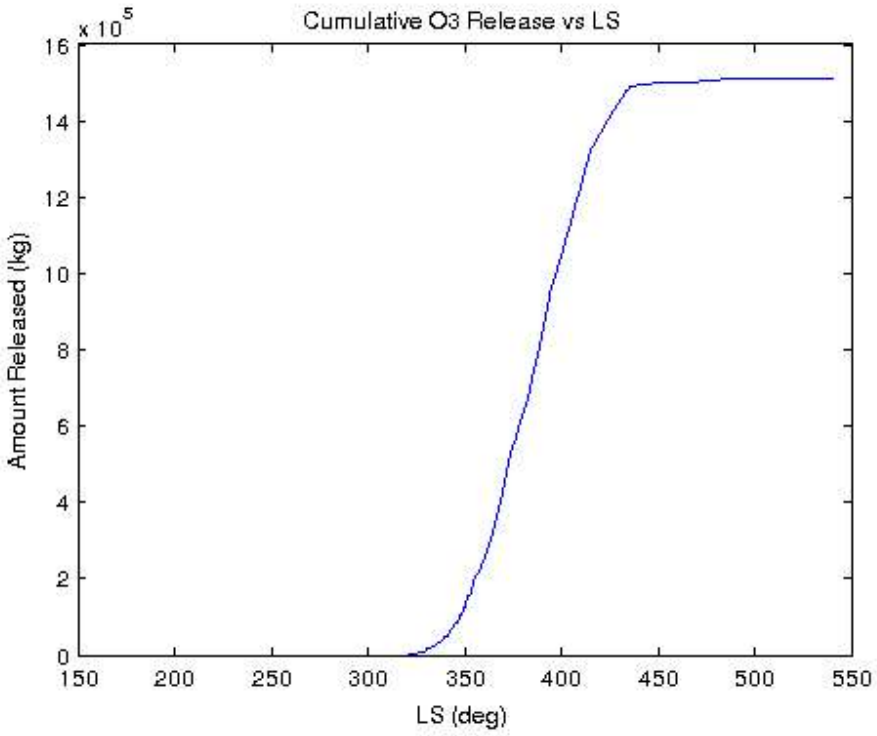


Figure 9: Surface emission fluxes of **(a)** CO, **(b)** O_3 and **(c)** CO_3 throughout the simulation year separated for each polar ice ring.

10 a)



10 b)



10 c)

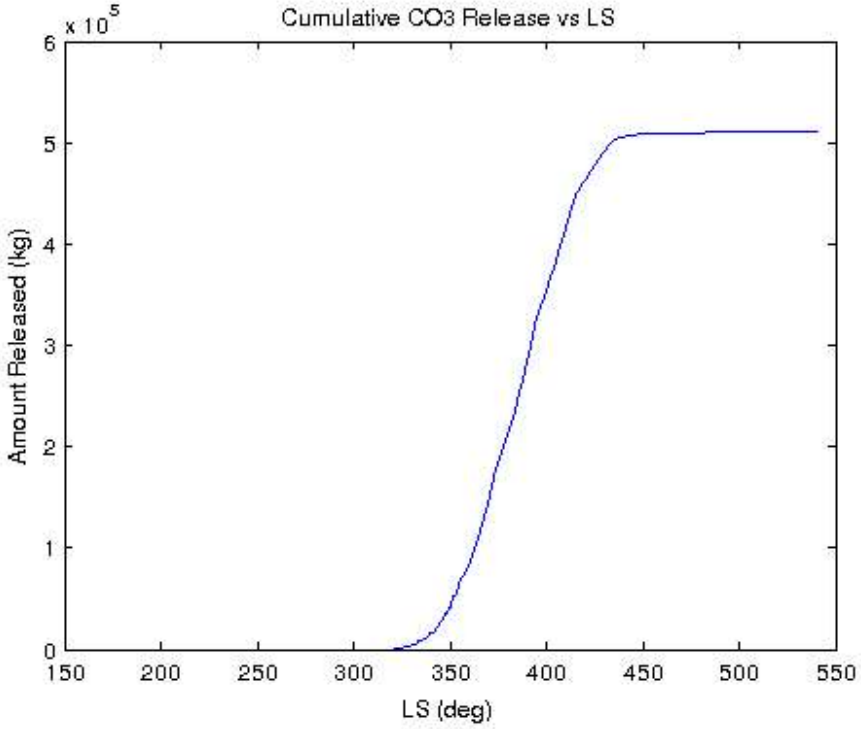


Figure 10: Cumulative amount of released CO_2 radiolytic species from the entire northern polar cap. **(a)** CO **(b)** O_3 **(c)** CO_3.

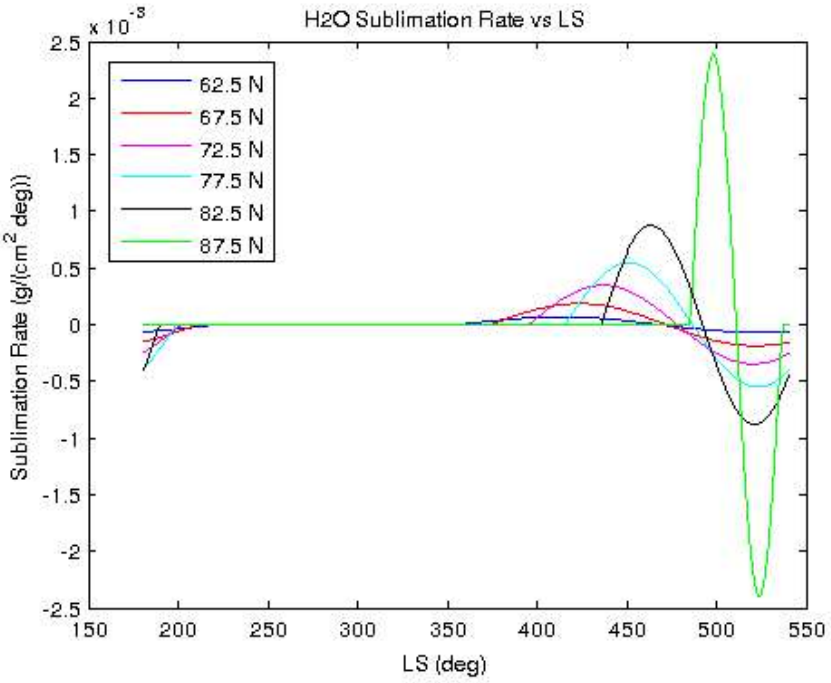


Figure 11: H_2O sublimation rate per LS throughout the simulation year.

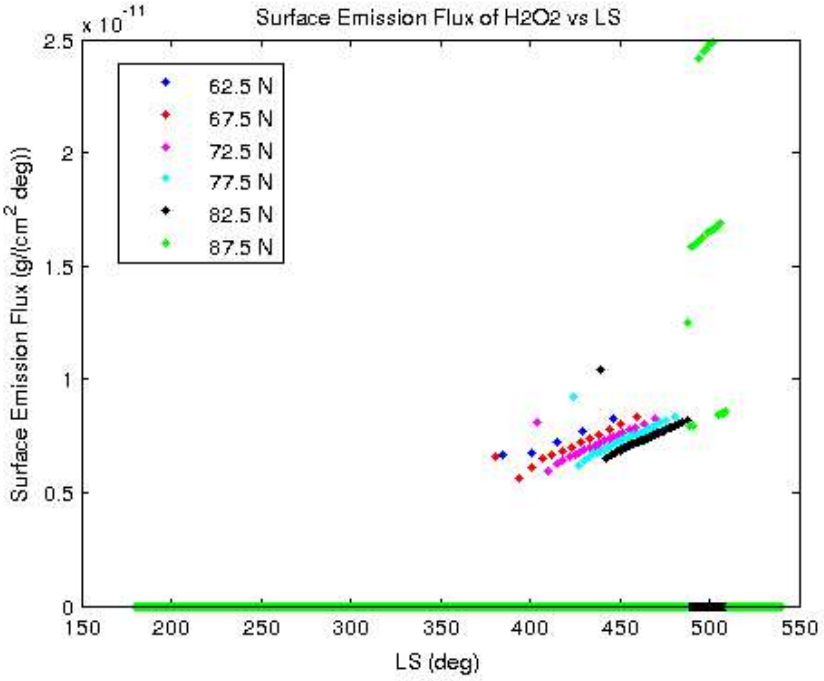


Figure 12: Surface emission flux of H_2O_2 throughout the simulation year.

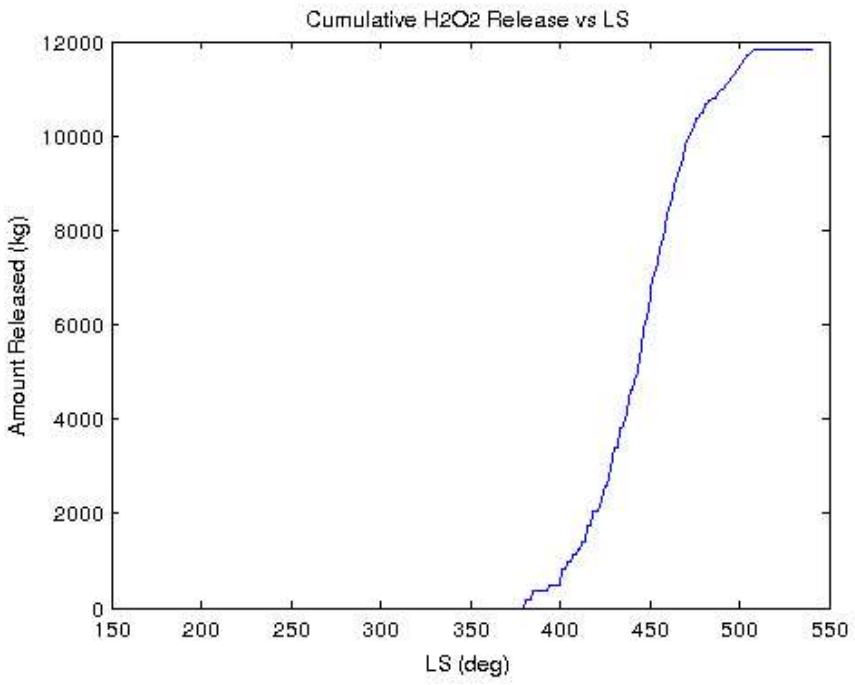


Figure 13: Cumulative amount of released H_2O radiolytic species (H_2O_2) from the entire northern polar cap as a function of LS.

The presence of surface emission fluxes (Figure 9 & 12) corresponds to the positive rate of sublimation in the respective regions (Figure 8 & 11). This is expected, as the sublimation of the parent species triggers the release of trapped radiolytic products. However, when the sublimation rate slows the emission flux continues to climb. This may be due to the fact that the slowing sublimation also corresponds to the destruction of the older ice layers that contain higher concentrations of radiolytic products. The amount released for the radiolytic species differs greatly. For example, about 9,000 tons of radiolytic CO is released into the atmosphere while only about 1.2 tons of H_2O_2 is released. This is due to a combination of low radiolytic yield for the species and low amount of deposition/sublimation of the parent target. Nevertheless, the

cumulative total amount (Figure 10 & 13) does not take photochemical reactions in the atmosphere into account. Therefore, it should not be treated as a direct modification of the actual amount present in the atmosphere at the end of the simulation.

3.2 Conclusions

The original motivation of this research is to study the viability of radiolysis as a production pathway for methane on Mars. Although recent findings by the Curiosity rover show a lower level of methane than previously thought on Mars, methane spikes detected at Gale crater do support the notion of a source other than photolytic production for methane [33]. On the other hand, laboratory measurement done by Pilling et al. [19] shows no detectable amount of methane from radiolysis of CO_2 and H_2O ice mixtures. This means that either methane cannot be produced radiolytically from CO_2 and H_2O ice mixtures, or the amount produced was too small to be detected during the Pilling experiment (the measurement obtained by the Curiosity rover corresponds to an annual methane production rate of 10 to 20 tons on Mars [34]). Nevertheless, by combining the results of controlled, small-scale production in the laboratory, the identification of the radiation source, modeling of the source interaction with the target material and the incorporation of a seasonal variation using model/observation data, radiolytic production over a large area can be simulated for an arbitrary time period. Although currently only the north polar region is simulated, the model can be applied for any area as surface data for the corresponding region becomes available. The source of particle radiation can also be expanded as more flux data is acquired.

While the model can simulate production efficiently, the accuracy and implications of the result are unclear. The lack of analysis using a proper atmospheric transport model means that a direct comparison with atmospheric mixing ratios from other models or observations is not possible. Furthermore, while the model computes the annual release from changes in the amount of formate deposits, it does not account for any possible accumulation in the permanent formate deposit. For example, the possibility of radiolytic formation of methane clathrate in the permafrost may warrant examinations on Earth as well as on Mars, especially when the climate is changing. As radiolysis provides a constant pathway to introduce chemical species into the environment, its long-term effect is worth investigating. Several potentially important factors that can affect the accuracy of the model are identified in Section 4.1. A strategy to validate the data produced by the model is presented in Section 4.2. A more in-depth study and application of these concepts should be a priority for future investigations.

4. Assessment of Uncertainties & Future Work

The model simulation results can be improved by identifying and accounting for factors that can introduce bias. The consequences and implications of radiolytic production can be further analyzed using atmospheric transport models. These will be explored in this section.

4.1 Uncertainties

Each component in the model contains uncertainties. The following sections contain a qualitative analysis of significant contributors.

4.1.1 Production Model

The assumption that the production rate of radiolysis is only dependent on the amount of energy transferred to the target is not completely correct. To demonstrate, define the G value as the numbers of a chemical species radiolytically produced from a charged particle losing 100 ev of its energy in water. Studies show that the G values for a heavy charged particle are consistently lower than a less massive particle for some radiolytic species (see Table 1 next page). This is due to the spatial distribution of the deposited energy (or rather, the spatial distribution of molecules produced from radiolysis) [35]. When the stopping power is high the large amount of energy transfer results in dissociated molecules being distributed in a dense pattern close to each other. This increases the chance of the molecules recombining back into the parent species. In contrast,

when the charged particle is light, the large distance between the dissociated molecules allows more molecules to combine into products different from the parent, therefore increasing the observed G value. This means that G value for a particle depends inversely on the stopping power for the particle in water hence the energy of the particle. This is particularly true when the particle has low energy due to the large increase in stopping power as the particle energy decreases (see Figure 3 in Section 2.3).

Table 1: G values of protons and alpha particles at various energy levels. Table referenced from [35].

Species Type	**Protons (MeV)**				**Alpha Particles (MeV)**			
	1	2	5	10	4	8	20	40
OH	1.05	1.44	2.00	2.49	0.35	0.66	1.15	1.54
H_3O^+	3.53	3.70	3.90	4.11	3.29	3.41	3.55	3.70
e^-_{aq}	0.19	0.40	0.83	1.19	0.02	0.08	0.25	0.46
H	1.37	1.53	1.66	1.81	0.79	1.03	1.33	1.57
H_2	1.22	1.13	1.02	0.93	1.41	1.32	1.19	1.10
H_2O_2	1.48	1.37	1.27	1.18	1.64	1.54	1.41	1.33
Fe^{3+}	8.69	9.97	12.01	13.86	6.07	7.06	8.72	10.31

The production model does not account for the variation in G value. The experiment carried out by Pilling et al. uses Ni ions. Ni experiences significantly higher stopping power at 52 MeV (the energy of Ni ions used by Pilling et al.) compared to protons at all energy levels (Figure 14). As such, the G value for Ni should be lower than the G value for protons. Therefore, for the same amount of energy transferred to a target, the amount of products from Ni irradiation should be lower than that from protons. It follows that the production mechanism is most likely underestimating the amount of species produced.

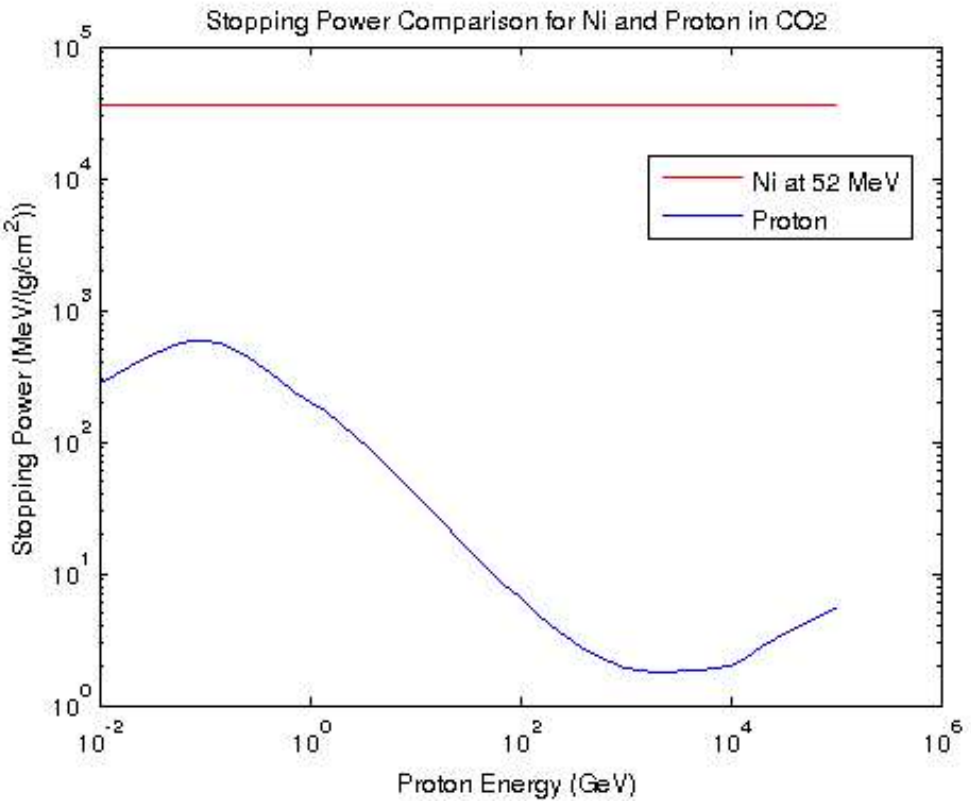


Figure 14: Comparison of stopping powers between 52 MeV Ni ions and protons at all energy levels used in simulation.

It is worth pointing out that Pilling's result shows asymptotic progressions that tends towards steady states for all radiolytic products [19]. It is possible that the aforementioned saturation of dissociated molecules causes the production to slow down. However, since the measurement is made over a span of over three hours, it is unlikely that many dissociated ions in the ice would survive for that long without combining into something first. Another possible explanation is the increasing concentration of radiolytic products raises the likelihood for them to become dissociated by radiolysis. In any case, the production rate decreases and eventually reaches zero

as total fluence increases. Without a clear understanding of the physical process behind this phenomenon, the production has to be modeled based on total deposited energy instead of per amount of unit energy deposited.

4.1.2 Cosmic Ray Flux

The primary cosmic ray flux is attenuated by the atmosphere of Mars. As shown in Section 2.2, the amount of attenuation is dependent on the vertically integrated mass and therefore pressure of the atmosphere. The model uses the average surface pressure for the computation. However, CO_2 will condense onto the surface during the winter. The total amount condensed each Martian year is about one third of the atmosphere mass. However, the northern hemisphere winter is shorter and warmer, resulting in only about half of the amount condensed in the southern winter [31]. Thus about one ninth of the CO_2 in the Martian atmosphere will condense during the northern winter. At the same time, the southern polar cap is rapidly shrinking due to rising temperature. The overall effect is a net increase in atmosphere pressure [12]. Therefore, the actual GCR flux that reaches the surface should be somewhat lower than what is currently used.

The computation of secondary proton flux involves a large amount of uncertainty because it is modeled after Earth observations, not data from Mars (see Section 2.2). Although the lack of magnetic deflection on Mars is taken into consideration by the primary GCR flux, it does not account for the higher vertically integrated mass from the atmosphere. On one hand, the higher integrated mass would cause more attenuation to the primary and secondary protons. On the

other hand, more attenuation means more secondary protons being produced. The combined effect on secondary proton flux is not immediately apparent. The production rate of secondary protons is, therefore, analyzed using Webber's formulation [36]:

$$Q(\Delta E') = 4\,\pi \cdot n_T \cdot \int j(E) \cdot \sigma(E, E')\, dE'\, dE \tag{23}$$

where

n_T = number density of the target gas species

j = differential cosmic ray flux

σ = differential production cross-section for the target gas species

Equation 23 has the advantage that the production is scaled by the number density of the gas. The only other terms that depend on the target characteristics are the cosmic ray flux and production cross-section. For the cosmic ray flux, Mizuno et al. calculated the probability of vertically incident cosmic rays reaching the GLAST location at an altitude of 38 km on Earth to be 95.8% [23]. Using the Beer-Lambert law and the nuclear interaction length of CO_2 (see Section 2.2), the probability for vertically incident cosmic rays to reach the surface of Mars is estimated to be around 82.5%. The vertically incident cosmic ray flux on Mars is, therefore, roughly 86% of the flux at the GLAST location on Earth.

Next, a comparison between production cross-sections of the two atmospheres must be made. This is done by assuming the production cross-section for secondary protons to be directly

proportional to the total cross-section in proton-nucleus impact for each gas. The total cross-section can be approximated using the method developed by Bradt and Peters [37][38]:

$$\sigma_r = \pi \cdot r_0^2 \cdot \left(A_p^{\frac{1}{3}} + A_t^{\frac{1}{3}} - b_0 \right)^2 \tag{24}$$

r_0 = proportionality constant for geometrical nucleus radius = $1.45*10^{-13}$ cm

A_p = number of nucleons in the incident particle = 1

A_t = number of nucleons in the target atom

b_0 = overlap transparency parameter computed using $\Delta R = 0.85*10^{-13}$ cm

= 1.172413793

For compound molecules like CO_2 the cross-section is considered to be additively contributed by each atom in the molecule. The average cross-section for each of the atmospheres is then computed as a weighted average using the cross-sections of the primary constituents in the corresponding atmosphere. The primary constituents and average total cross sections are listed in Table 2 next page.

Table 2: Simplified atmosphere compositions for Earth and Mars and the computed total cross-sections using Bradt-Peters formula.

Earth Primary Constituents	**Percentage**	**Total Cross Section (cm^2)**
N_2	78%	$6.62*10^{-25}$
O_2	21%	$7.28*10^{-25}$
Ar	1%	$6.97*10^{-25}$
Weighted Mean	N/A	$6.76*10^{-25}$
Mars Primary Constituents		
CO_2	95.3%	$1.75*10^{-24}$
N_2	2.7%	$6.62*10^{-25}$
Ar	2%	$6.97*10^{-25}$
Weighted Mean	N/A	$1.70*10^{-24}$

As the results indicate, the weighted mean total cross-section for the Mars atmosphere is almost three times as large as that of the Earth. The number densities of the atmospheres at the altitudes of interest must now be computed in order to compare secondary proton production rate. The average number density can be calculated from the ideal gas law:

$$N = \frac{P \cdot V}{k \cdot T} \tag{25}$$

where

k = Boltzmann constant = $1.38*10^{-23}$ J/K

P = atmosphere pressure in Pa

V = unit volume (set to 1 m^3)

T = temperature in kelvin.

The values used and results computed can be found in Table 3 next page.

Table 3: Values used in the computation of average number densities at the altitudes of interest for both planets. See [39] and [11] respectively.

	Earth	**Mars**
Altitude of Interest	38 (km) (GLAST BFEM altitude)	0 (km) (surface)
Atmosphere Pressure (P)	365.486 (Pa)	636 (Pa)
Temperature (T)	245.448 (K)	210 (K)
Average Number Density (N)	1.07E23 (molecules/m^3)	2.19E23 (molecules/m^3)

As shown, the number density for Mars is almost double that of the density for Earth at the measurement altitude. Combined with a slightly lower cosmic ray flux and a weighted mean total cross-section of almost three times the magnitude of Earth's, the secondary proton flux on Mars can be as high as five times the amount on Earth. The secondary proton flux model does not account for this, therefore, the energy deposition from secondary protons and the radiolytic production from them are most likely underestimated.

Although the model uses GCR primary protons as the only particle radiation source, there are other sources such as heavier GCR particles (He, Li, Fe, etc.) and solar protons [40]. Besides the obvious increase in overall production, due to the increased total energy deposition, this may have an effect on any potential leakage to the atmosphere from the ice surface. Physically, solar protons behave more or less similarly to GCR protons. However, other types of GCR particles have different stopping power curves due to their increased mass and sizes. Generally, the heavier the particle, the faster it loses energy because higher incident particle mass leads to

higher maximum energy transfer per collision (see Equation 17, Section 2.3). Consequently, heavier ions will contribute more to the production of radiolytic species near the surface of the ice. If a pathway exists for a product to leak into the atmosphere, then a higher surface product concentration will magnify the effect of this pathway. A potential mechanism for surface leakage is proposed in Section 4.1.3.

4.1.3 Stopping Power

The stopping power for particle energies less than 10 GeV is computed using SRIM. The expected uncertainty for SRIM is roughly 5% around laboratory measurements [29]. It should be noted that only electronic stopping power is modeled. Energy loss due to nuclear stopping (physical collision of the nuclei) is neglected as it contributes only around 1% of the total stopping power even when the incident particles have low energy (see Figure 15). At high energy the nuclear stopping power is smaller relative to electronic stopping. It is, therefore, neglected entirely in the energy deposition model, both in SRIM simulation and the Bethe-Bloch implementation. Nevertheless, the model's lack of nuclear stopping results in all of the energy from the charged particle being lost through electronic stopping. This will overestimate radiolytic production slightly.

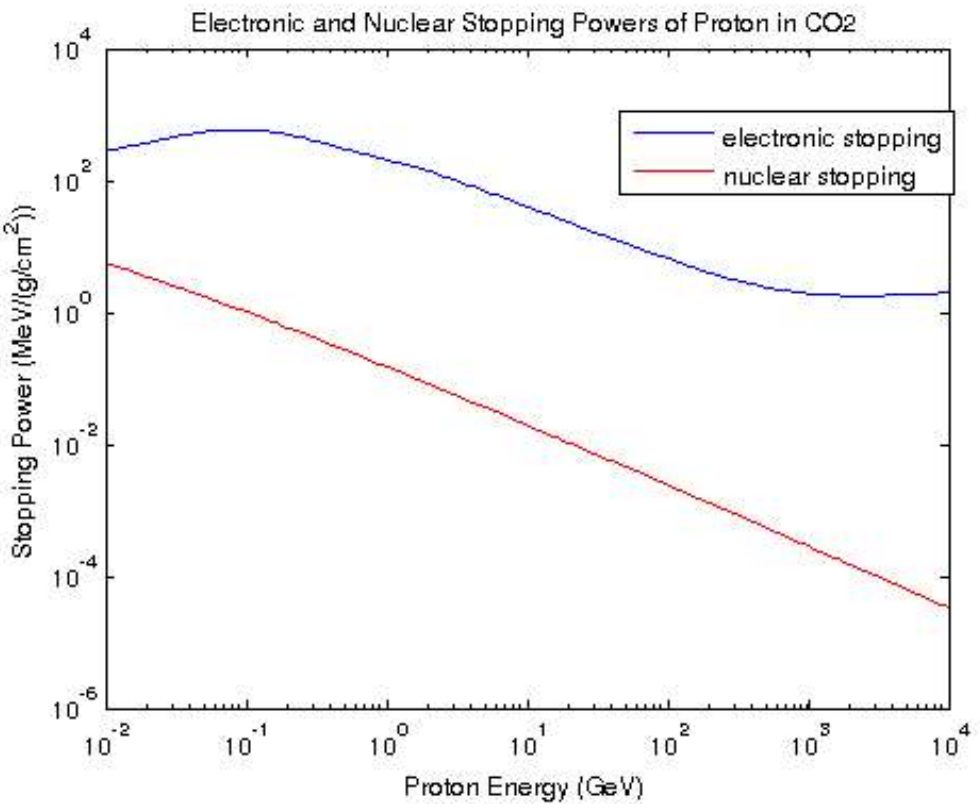


Figure 15: Log-Log plot of electronic and nuclear stopping powers of protons in CO_2 ice. The values shown were obtained from a SRIM-2013 simulation.

The energy deposited due to nuclear stopping could potentially cause molecules of the target material to be knocked loose from the structure of the target. This is called sputtering. Sputtering can release the target material (and anything trapped inside it) into the atmosphere even when there is no sublimation or other leakages [41]. As the model currently does not consider sputtering or any other forms of leakage, the amount of cumulative radiolytic products stored in the ice layers at a given time (represented by cumulative energy deposition in the model) is most likely overestimated. If the production rate is constant versus fluence this will not

cause any deviation. However, as Pilling's production curves show asymptotic trends that tends towards equilibria, the production rate of a particular radiolytic product with high amount of said product already inside the ice should be lower. Thus, by overestimating the amount of cumulative radiolytic products stored in the ice at a given time the production rate is being underestimated and, therefore, the total amount of product that is eventually released into the atmosphere is underestimated as well.

4.1.4 Ice Thickness

GM3 computes the CO_2 ice thickness using the heat balance of Mars atmosphere. While the total deposited mass agrees with total observed CO_2 mass deposit on the surface from High Energy Neutron Detector (HEND) result [31], it does not account for asymmetries and local features. For example, in reality, the southern polar cap is asymmetrical and contains a patch of permanent CO_2 ice [42][43]. GM3 does not account for this. If one wishes to extend the model to the entire planet a more detailed surface ice model or observation data should be used. Similarly, the water ice model is built on assumed thickness progression constrained to observed total sublimated water mass in the region. Latitudinal variations are approximated from CO_2 latitudinal changes based on GM3 results. This is even less accurate as heat balance is not taken into account. However, the water ice model is only meant to serve as a guideline since current data on the seasonal variation of water ice cap is minimal and not accurate enough to show surface water ice thickness variation. Without extensive seasonal maps for Mars surface ice variation, it is difficult to estimate the overall effects these potential deviations may have on radiolytic production results.

4.2 Method for Data Validation

Suppose a chemical species X has a lifetime of L years and a total amount of K tons in the Mars atmosphere. Since the Mars atmosphere has persisted in the current condition for a long period of time, the time of production for species X currently in the atmosphere can be roughly assumed to spread out uniformly over L years into the past. After some t years elapsed, the amount of species destroyed therefore must be $D_{tot} = t/L*K$ tons. Similarly, the amount produced must be $P_{tot} = D_{tot} = t/L*K$ tons as well. Otherwise the mixing ratio of species X will not be stable over the long term. If the production of X is mainly caused by photolysis with an amount of P_p tons during the period t, then $P_o = P_{tot} - P_p$ represents the production from mechanisms other than photolysis. While P_o is not limited exclusively to radiolysis, the amount produced by radiolysis must be at the very least capped by P_o. In the case that there is no other significant mechanisms for the production of X besides photolysis and radiolysis, P_o must be equal to the production due to radiolysis.

Some difficulties exist for this method. The first one being that X should be carefully chosen to maximize photolytic and radiolytic productions but minimize all other potential mechanisms. Also seasonal variation in the mixing ratio, production and extinction rates for species X may prevent one from making the assumption that the production time of all existing amount of species X is uniformly spread out. The deviation caused by this can be minimized by choosing a period that is an integer multiple of a Martian year and thus all seasonal effects are included in the examination period. In general though an atmospheric transport model should be utilized to predict the effect of ground emission on the overall mixing ratio and compare it to measured data. The methodology to construct a simple 1D transport model is outlined in section 4.2.1. Naturally

a photochemistry model for the Mars atmosphere is also required in order to compute the amount of *X* produced in the period t due to photolysis. This is covered in Section 4.2.2.

4.2.1 Atmospheric Transport Model

An atmospheric transport model is necessary for the proper analysis of the effect of surface emission from radiolytic production. A 1-D transport model will allow one to study the vertical profiles of species distribution with a good balance of simplicity, performance and accuracy. A model following Wofsy & McConnell's scheme [44] is outlined below.

Governing Equations

Consider a fixed volume or a box in the atmosphere, the conservation of mass for a particular species i is specified by the species continuity equation:

$$\mathbf{N}(\mathbf{s}) \cdot \frac{\partial f_i(\mathbf{s})}{\partial t} + \nabla \cdot \mathbf{U}(\mathbf{s}) = E_i + P_i - L_i - D_i \qquad (26)$$

where

$N(\mathbf{s})$ = total number density of the atmosphere inside the box at location $\mathbf{s}$ (m^{-3})

$f_i(\mathbf{s})$ = mixing ratio of species i at location $\mathbf{s}$

$\mathbf{U}(\mathbf{s})$ = flow field of the fluid/air (s^{-1})

E_i = emission of the species (ie: radiolysis) (m^{-3} s^{-1})

P_i = volumetric production of the species due to photochemistry ($m^{-3}s^{-1}$)

L_i = volumetric loss of the species due to photochemistry ($m^{-3}s^{-1}$)

D_i = deposition of the species ($m^{-3}s^{-1}$)

Distinctions are made between emission and volumetric production (also volumetric loss and deposition) because photochemical production and loss are coupled with other species while emission and deposition are not. In 1-D, Equation 26 becomes

$$N(z) \cdot \frac{\delta f_i}{\delta t} + \frac{\delta \varphi(z)}{\delta z} = E_i(z) + P_i(z) - L_i(z) - D_i(z) \quad (27)$$

Note that everything is the 1-D equivalent of Equation 26. φ is the vertical flux of the species in molecules/(cm^2 s). It can be defined by writing out the diffusion equation:

$$N(z) \cdot \frac{\delta f_i}{\delta z} = \frac{D}{D+K} \cdot N(z) \cdot f_i \cdot \left(m_{avg} - m_i\right) \cdot \frac{g}{k \cdot T} - \frac{\varphi}{D+K} \quad (28)$$

where

N(z) = Total number density of the atmosphere at altitude z

D = molecular diffusion coefficient

K = eddy diffusion coefficient

m_{avg} = mean molecular mass

m_i = the molecular mass of the diffusing constituent from the i^{th} species

g = gravitational acceleration constant

k = Boltzmann constant

T = temperature (in kelvin)

By combining Equation 27 and Equation 28 and solving for Φ, the amount of molecules transported out of the box can be calculated at each time step. A vertical profile of the species can then be constructed from N(z).

4.2.2 Photochemistry Model

In the atmosphere the constituents can react with other constituents. Photolysis also occurs as a result of irradiation from the sun. These reactions result in the loss of reactants and production of the products. This is then scaled by the rate coefficients (see Table 4 & 5 next page):

Table 4: Chemical reaction rates in the Mars atmosphere. Table referenced from [45]

No.	Reaction	Rate Coefficient*	Reference†
(R1)	$CO + O + CO_2 \rightarrow CO_2 + CO_2$	$k_1 = 2 \times 10^{-37}$	*Slanger et al.* [1972]
(R2)	$CO + OH \rightarrow CO_2 + H$	$k_2 = 6.0 \times 10^{-13} (0.25 + A[M]/(1 + A[M]))$ $A = 1.82 \times 10^{-20}$	
(R3)	$O(^3P) + O_2 + CO_2 \rightarrow O_3 + CO_2$	$k_3 = 6.0 \times 10^{-34} (T/300)^{-2.3}$	
(R4)	$O(^3P) + O_3 \rightarrow O_2 + O_2$	$k_4 = 8.0 \times 10^{-12} \exp(-2060/T)$	
(R5)	$O(^3P) + O(^3P) + CO_2 \rightarrow O_2 + CO_2$	$k_5 = 9.4 \times 10^{-34} \exp(484/T)$	*Hampson* [1980]
(R6)	$O(^1D) + CO_2 \rightarrow O(^3P) + CO_2$	$k_6 = 7.4 \times 10^{-11} \exp(117/T)$	
(R7)	$O(^1D) + H_2 \rightarrow OH + H$	$k_7 = 1.0 \times 10^{-10}$	
(R8)	$O(^1D) + H_2O \rightarrow OH + OH$	$k_8 = 2.2 \times 10^{-10}$	
(R9)	$H + O_2 + CO_2 \rightarrow HO_2 + CO_2$	$k_9 = 1.3 \times 10^{-31} (T/300)^{-1.6}$	*Lindner* [1988]
(R10)	$H + O_3 \rightarrow OH + O_2$	$k_{10} = 1.4 \times 10^{-10} \exp(-470/T)$	
(R11)	$O(^3P) + OH \rightarrow O_2 + H$	$k_{11} = 2.2 \times 10^{-11} \exp(117/T)$	
(R12)	$O(^3P) + HO_2 \rightarrow OH + O_2$	$k_{12} = 3.0 \times 10^{-11} \exp(200/T)$	
(R13)	$OH + HO_2 \rightarrow H_2O + O_2$	$k_{13} = 1.7 \times 10^{-11} \exp(416/T)$	
(R14)	$H + HO_2 \rightarrow H_2 + O_2$	$k_{14} = 6.66 \times 10^{-12}$	
(R15)	$H + HO_2 \rightarrow H_2O + O$	$k_{15} = 2.96 \times 10^{-12}$	
(R16)	$H + HO_2 \rightarrow OH + OH$	$k_{16} = 6.44 \times 10^{-11}$	
(R17)	$HO_2 + HO_2 \rightarrow H_2O_2 + O_2$	$k_{17} = 2.3 \times 10^{-13} \exp(590/T)$	
(R18)	$OH + OH + CO_2 \rightarrow H_2O_2 + CO_2$	$k_{18} = 1.6 \times 10^{-30} (T/300)^{-0.8}$	*Lindner* [1988]
(R19)	$O(^3P) + H_2O_2 \rightarrow H_2O + O_2$	$k_{19} = 1.4 \times 10^{-12} \exp(-2000/T)$	
(R20)	$H + H + CO_2 \rightarrow H_2 + CO_2$	$k_{20} = 2.0 \times 10^{-32}$	*Hampson* [1980]
(R21)	$HO_2 + O_3 \rightarrow OH + 2O_2$	$k_{21} = 1.4 \times 10^{-14} \exp(-580/T)$	
(R22)	$H_2 + OH \rightarrow H + H_2O$	$k_{22} = 6.1 \times 10^{-12} \exp(-2030/T)$	
(R23)	$H_2O_2 + OH \rightarrow HO_2 + H_2O$	$k_{23} = 3.1 \times 10^{-12} \exp(-187/T)$	
(R24)	$CO_2^+ + H_2 \rightarrow CO_2H^+ + H$	$k_{24} = 1.4 \times 10^{-9}$	*Krasnopolsky* [1986]
(R25)	$CO_2H^+ + e \rightarrow CO_2 + H$	$k_{25} = 3.0 \times 10^{-7}$	*Krasnopolsky* [1986]

*All values are quoted in the molecule cm s system.
†Except as noted, reaction rate constants are those recommended by *DeMore et al.* [1985].

Table 5: Photodissociation processes in the Mars atmosphere used by Rodrigo et al. [45]. Note that rate coefficients are not included in the table.

No.	Reaction	Wavelength, nm
1	$CO_2 + h\nu \rightarrow CO + O(^1D)$	$\lambda < 167$
2	$CO_2 + h\nu \rightarrow CO + O$	$\lambda \geq 167$
3	$O_3 + h\nu \rightarrow O_2 + O(^1D)$	Lyman-α $ef_{O(^1D)} = 0.7$ $\lambda < 410$ $ef_{O(^1D)} = 0.7$
4	$O_3 + h\nu \rightarrow O_2 + O$	$410 \leq \lambda < 730$
5	$O_2 + h\nu \rightarrow O + O(^1D)$	Lyman-α $\lambda < 175.4$
6	$O_2 + h\nu \rightarrow O + O$	$175.4 < \lambda \leq 243.9$
7	$H_2O_2 + h\nu \rightarrow OH + OH$	$\lambda < 350$
8	$HO_2 + h\nu \rightarrow OH + O$	$175 \leq \lambda < 275$
9	$H_2O + h\nu \rightarrow H + OH$	Lyman-α ef = 0.75 $\lambda < 198$
10	$H_2O + h\nu \rightarrow H_2 + O$	Lyman-α ef = 0.25

Since photolysis depends on both the solar flux and the spectral absorption cross-sections of the species, the rate coefficients can be computed accordingly [46]:

$$J_{jz} = \sum_{i,\lambda} \sigma_{i\lambda} \cdot \phi_{z\lambda} \tag{29}$$

where

J_{jz} = rate coefficient for the jth photodissociation reaction at an altitude of z.

$\sigma_{i\lambda}$ = absorption cross-section of the ith constituent in the reaction for photons with wavelength λ.

$\varphi_{z\lambda}$ = spectral solar flux of wavelength λ at altitude z.

Chemical reactions and photolysis account for the L and P terms in Equation 27. By incorporating it into the transport model the concentrations of the constituents become coupled with each other. This is crucial in determining the effects of surface emission from radiolytic productions. Future efforts should prioritize on the construction of a chemical transport model to allow comparison with observation data.

Bibliography

[1] A. Neubeck, N. T. Duc, D. Bastviken, P. Crill, and N. G. Holm, "Formation of H_2 and CH_4 by weathering of olivine at temperatures between 30 and 70°C.," *Geochem. Trans.*, vol. 12, no. 1, p. 6, Jan. 2011.

[2] M. E. Summers, B. J. Lieb, E. Chapman, and Y. L. Yung, "Atmospheric biomarkers of subsurface life on Mars," *Geophys. Res. Lett.*, vol. 29, no. 24, p. 2171, 2002.

[3] A. Robledo-Martinez, H. Sobral, and A. Ruiz-Meza, "Electrical discharges as a possible source of methane on Mars: Lab simulation," *Geophys. Res. Lett.*, vol. 39, no. 17, Sept. 2012.

[4] N. O. Renno and J. F. Kok, "Electrical Activity and Dust Lifting on Earth, Mars, and Beyond," *Space Sci. Rev.*, vol. 137, no. 1–4, pp. 419–434, Jun. 2008.

[5] A. Geminale, V. Formisano, and G. Sindoni, "Mapping methane in Martian atmosphere with PFS-MEX data," *Planet. Space Sci.*, vol. 59, no. 2–3, pp. 137–148, Feb. 2011.

[6] R. Johnson and T. Quickenden, "Photolysis and radiolysis of water ice on outer solar system bodies," *J. Geophys. Res.*, vol. 102, pp. 10985–10996, 1997.

[7] W. C. Feldman, T. Prettyman, S. Maurice, J. J. Plaut, D. L. Bish, D. T. Vaniman, M. T. Mellon, A. E. Metzger, S. W. Squyres, S. Karunatillake, W. V. Boynton, R. C. Elphic, H. O. Funsten, D. J. Lawrence, and R. L. Tokar, "Global distribution of near-surface hydrogen on Mars," *J. Geophys. Res.*, vol. 109, no. E09006, 2004.

[8] N. Bibler, J. Pareizs, T. Fellinger, and C. Bannochie, "Measurement and Prediction of Radiolytic Hydrogen Production in Defense Waste Processing Slurries at Savannah River Site," *Waste Manag. Conf.*, vol. 2, no. 1, 2007.

[9] M. Moore, R. Khanna, and B. Donn, "Studies of Proton Irradiated H2O+CO2 and H2O+CO Ices and Analysis of Synthesized Molecules," *J. Geophys. Res.*, vol. 96, no. 91, pp. 17541–17545, 1991.

[10] D. Bahr, M. Fama, R. Vidal, and R. Baragiola, "Radiolysis of water ice in the outer solar system: Sputtering and trapping of radiation products," *J. Geophys. Res.*, vol. 106, no. 2000, pp. 285–290, 2001.

[11] NASA, "Mars Fact Sheet." [Online]. Available: http://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html.

[12] N. Barlow, *Mars: An introduction to its interior, surface and atmosphere*, 1st ed. Cambridge University Press, 2008.

[13] N. Thomas, C. J. Hansen, A. Pommerol, G. Portyankina, and K. Aye, "Observations of the surface effects of the CO_2 and H_2O cycles on Mars," in *X Scientific Meeting of the Spanish Astronomical Society*, 2012.

[14] A. L. Sprague, W. V. Boynton, K. E. Kerry, D. M. Janes, N. J. Kelly, M. K. Crombie, S. M. Nelli, J. R. Murphy, R. C. Reedy, and A. E. Metzger, "Mars' atmospheric argon: Tracer for understanding Martian atmospheric circulation and dynamics," *J. Geophys. Res.*, vol. 112, no. E03S02, Jan. 2007.

[15] F. Bauregger, "Seasons on Mars." [Online]. Available: http://pweb.jps.net/~gangale3/bauregger/seasons.html.

[16] J. G. Luhmann and C. T. Russell, "Mars: Magnetic Field and Magnetosphere," *Encyclopedia of Planetary Sciences*. Chapman and Hall, pp. 454–456, 1997.

[17] L. Simonsen and J. Nealy, "Mars surface radiation exposure for solar maximum conditions and 1989 solar proton events," NASA, 1993.

[18] M. Balonov, D. Chambers, K. Faulkner, G. Howe, G. Ibbott, G. Kirchner, D. Melo, R. Ricks, E. Rochedo, M. Stabin, G. Webb, and D. Woodhead, "Sources and Effects of Ionizing Radiation," 2008.

[19] S. Pilling and E. S. Duarte, "Radiolysis of H2O: CO2 ices by heavy energetic cosmic ray analogs," *Astron. Astrophys.*, 2010.

[20] S. Pilling, E. Seperuelo Duarte, E. F. da Silveira, E. Balanzat, H. Rothard, A. Domaracka, and P. Boduch, "Radiolysis of ammonia-containing ices by energetic, heavy, and highly charged ions inside dense astrophysical environments," *Astron. Astrophys.*, vol. 509, no. A87, Jan. 2010.

[21] M. J. Loeffler, G. A. Baratta, M. E. Palumbo, G. Strazzulla, and R. A. Baragiola, "CO_2 synthesis in solid CO by Lyman- α photons and 200 keV protons," *Astron. Astrophys.*, vol. 594, pp. 587–594, 2005.

[22] "NASA Imagine the Universe." [Online]. Available: http://imagine.gsfc.nasa.gov/docs/science/know_l1/cosmic_rays.html.

[23] T. Mizuno, T. Kamae, and G. Godfrey, "Cosmic-ray background flux model based on a gamma-ray large area space telescope balloon flight engineering model," *Astrophys. J.*, vol. 614, pp. 1113–1123, 2004.

[24] L. Gleeson and W. Axford, "Solar modulation of galactic cosmic rays," *Astrophys. J.*, vol. 154, 1968.

[25] Particle Data Group, "Atomic and Nuclear Properties of Materials." [Online]. Available: http://pdg.lbl.gov/2012/AtomicNuclearProperties/index.html.

[26] J. Ziegler, M. Ziegler, and J. Biersack, *SRIM–The stopping and range of ions in matter*. .

[27] D. E. Groom, N. V Mokhov, and S. I. Striganov, "Muon Stopping Power and Range," *At. Data Nucl. Data Tables*, vol. 76, no. 2, 2001.

[28] R. Sternheimer, M. Berger, and S. Seltzer, "Density effect for the ionization loss of charged particles in various substances," *Phys. Rev. B*, vol. 26, no. 11, pp. 6067–6076, 1982.

[29] J. F. Ziegler, "SRIM Accuracy Plots." [Online]. Available: http://www.srim.org/SRIM/SRIMPICS/STOPPLOTS.htm.

[30] M. L. Litvak, I. G. Mitrofanov, a. S. Kozyrev, a. B. Sanin, V. I. Tret'yakov, W. V. Boynton, C. Shinohara, D. Hamara, S. Saunders, and D. M. Drake, "Seasonal Carbon Dioxide Depositions on the Martian Surface as Revealed from Neutron Measurements by the HEND Instrument Onboard the 2001 Mars Odyssey Spacecraft," *Sol. Syst. Res.*, vol. 38, no. 3, pp. 167–177, May 2004.

[31] A. Akingunola, "Martian water cycle modeling with the second generation of the Global Mars Multiscale Model," York University, 2009.

[32] R. Haberle and B. Jakosky, "Sublimation and transport of water from the north residual polar cap on Mars," *J. Geophys. Res.*, vol. 95, no. 89, pp. 1423–1437, 1990.

[33] C. R. Webster, P. R. Mahaffy, S. K. Atreya, G. J. Flesch, M. A. Mischna, P.-Y. Meslin, K. A. Farley, P. G. Conrad, L. E. Christensen, A. A. Pavlov, J. Martin-Torres, M.-P. Zorzano, T. H. McConnochie, T. Owen, J. L. Eigenbrode, D. P. Glavin, A. Steele, C. A. Malespin, P. D. Archer, B. Sutter, P. Coll, C. Freissinet, C. P. McKay, J. E. Moores, S. P. Schwenzer, J. C. Bridges, R. Navarro-Gonzalez, R. Gellert, and M. T. Lemmon, "Mars methane detection and variability at Gale crater," *Science (80-.).*, vol. 347, no. 6220, pp. 415–417, 2014.

[34] G. Webster and D. Brown, "NASA Curiosity Rover Detects No Methane On Mars." [Online]. Available: http://mars.nasa.gov/msl/news/whatsnew/index.cfm?FuseAction=ShowNews&NewsID=1519.

[35] J. Coderre, "Principles of Radiation Interactions." Massachusetts Institute of Technology: MIT OpenCourseWare, 2004.

[36] J. Feit and S. Milford, "Secondary protons from cosmic ‐ ray collisions in space," *J. Geophys. Res.*, vol. 70, no. 23, pp. 5957–5960, 1965.

[37] L. Sihver and M. Lantz, "A study of total reaction cross section models used in particle and heavy ion transport codes," *2011 Aerosp. Conf.*, vol. 2, no. 1, pp. 1–10, Mar. 2011.

[38] H. Bradt and B. Peters, “The heavy nuclei of the primary cosmic radiation,” *Phys. Rev.*, vol. 77, no. 1, 1950.

[39] I. Kroo, “Standard Atmosphere Calculator,” 1997. [Online]. Available: http://aero.stanford.edu/stdatm.html.

[40] “Tables of Physical & Chemical Constants (16th edition 1995). 2.1.4 Hygrometry. Kaye & Laby Online. Version 1.0 (2005).” [Online]. Available: http://www.kayelaby.npl.co.uk/general_physics/2_7/2_7_7.html.

[41] R. Johnson, “Sputtering and desorption from icy surfaces,” *Sol. Syst. Ices*, 1998.

[42] M. Giuranna, D. Grassi, V. Formisano, L. Montabone, F. Forget, and L. Zasova, “PFS/MEX observations of the condensing CO2 south polar cap of Mars,” *Icarus*, vol. 197, no. 2, pp. 386–402, Oct. 2008.

[43] M. Richardson and R. Wilson, “Investigation of the nature and stability of the Martian seasonal water cycle with a general circulation model,” *J. Geophys. Res.*, vol. 107, 2002.

[44] S. C. Wofsy, J. C. McConnell, and M. B. McElroy, “Atmospheric CH_4, CO, and CO2,” *J. Geophys. Res.*, vol. 77, no. 24, 1972.

[45] R. Rodrigo, E. Garcia-Alvarez, M. J. Lopez-Gonzalez, and J. J. Lopez-Moreno, “A nonsteady one-dimensional theoretical model of Mars’ neutral atmospheric composition between 30 and 200 km,” *J. Geophys. Res.*, vol. 95, no. B9, pp. 14795–14810, 1990.

[46] R. Rodrigo, J. Lopez-Moreno, M. Lopez-Puertas, F. Moreno, and A. Molina, “Neutral atmospheric composition between 60 and 220 km: A theoretical model for mid-latitudes,” *Planet. Space Sci.*, vol. 34, no. 8, pp. 723–743, 1986.

Appendix A

A.1 Parameter Values

Table 6: Formation cross-sections and dissociation cross-sections used for the production model (see Section 2.1). Values obtained from [19]. Mediums are either pure CO_2 or H_2O.

Species	σ_f (10^{-13} cm^2)	σ_d (10^{-13} cm^2)
CO	1.8	11
O_3	0.27	16
CO_3	0.09	97
H_2O_2	0.19	10

Table 7: Parameters for secondary proton power law fits (see Section 2.2). Values taken from [23].

Parameter	Value
F_0	0.44
F_1	0.037
A	1.98
Ecut	0.21

Table 8: Settings used for the simulation.

Parameter	Value
Mars Planetary Radius	3390 (km)
Minimum Incident Angle	0 (deg)
Maximum Incident Angle	90 (deg)
Mars Surface Pressure	6.36 (mb)
Mars Gravitational Constant	3.711 (m/s^2)
Nuclear Interaction Length of Mars Atmosphere	88.9 (g/cm^2)
Start Time	180 (LS)
End Time	540 (LS)
Minimum Energy (for GCR particles)	10 (MeV)
Maximum Energy (for GCR particles)	50000 (MeV)

A.2 Step Size Settings

When integration is required during simulation, the model approximates the result using numerical methods. Special input parameters are constructed to test the effects of step size settings. Ice layer deposition and evaporation are artificially limited to very low amount in the span of one day.

1. Computation of the particle energy as a function of distance traveled in the medium using forward Euler iteration (see Section 2.3). Ideally the step size (SS) used here should be set close to the ice thickness Pilling used in the experiment (ie: 0.4 μm and 0.5 μm for CO_2 and H_2O respectively). However, computation using 0.1 mm shows nearly identical results at all energy ranges with much improved performance (see Figure 16 on p55). This is the setting used for the computation of energy v. distance tables during the simulation.
2. Energy depositions from individual particles are integrated over all possible initial energy of the particles (from 10 MeV to 50 GeV). This is computed using a simple Gaussian quadrature. Various test runs show convergence as the energy step size decreases. However, a large energy step size leads to an underestimation of radiolytic production (Table 9). Further complications arise from the memory requirement involved in vectorized computation. The amount of memory allocated for the computation grows in the order of M*N, where M is the number of energy steps and N is the number of micro-layers the particles must traverse. Setting the energy step too small will result in operations that are potentially memory unsafe and should be avoided.

3. For speed reasons, the ice layers are constructed as vertically stacked micro-layers of identical thickness (see Section 2.1). As Table 9 indicates, convergence can be achieved by decreasing micro-layer thickness. The primary factor that affects the accuracy here is whether the chosen micro-layer thickness can accurately reflect the retrieved total ice thickness. Also, as mentioned in the Section 2.1, to simplify Equation 5 it requires a small micro-layer thickness such that the rate of energy loss for the particle is close to constant as it travels the distance.
4. The deposited energy is scaled according to the particle's flux at a particular incident angle. This has to be integrated over all possible incident angles from 0 to 90 deg. Simple trapezoidal rule is utilized to increase accuracy without sacrificing too much speed. Comparison to results obtained from adaptive quadrature (error tolerance: 10^{-6}) can be seen in Table 9. Even with large angle step size (ie: 5 deg) the difference is small compared to the effects of layer or energy step sizes.
5. The time step size primarily controls the time resolution of changing ice thickness. The potential error is limited to the amount of radiolytic product produced within the ice that is deposited or sublimed within a single time step. Thus its effect on overall uncertainty can be limited by choosing a value that captures ice thickness changes within a reasonable time frame.

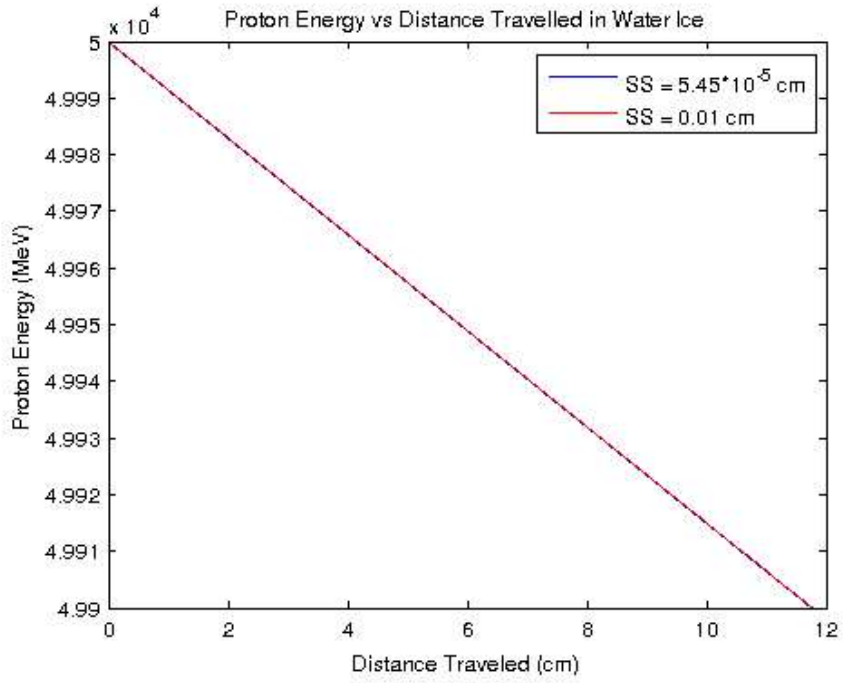


Figure 16: Proton energy versus distance traveled in water ice using the Bethe-Bloch implementation.

Table 9: Effects of various step sizes on the amount of radiolytic CO computed by model simulation.

Energy Step Size	100 MeV	10 MeV	1 MeV
Amount of CO Produced in the First Ring During Energy Test	$7.348324*10^5$ g	$8.437743*10^5$ g	$8.481224*10^5$ g
Micro-layer Size (CO2)	0.1 cm	0.01 cm	0.001 cm
Amount of CO Produced in the First Ring During Layer Test	$7.348324*10^5$ g	$8.082885*10^5$ g	$8.082885*10^5$ g
Angle Step Size	5 deg	1 deg	Adaptive Quadrature
Amount of CO Produced in the First Ring During Angle Test	$7.263266*10^5$ g	$7.355722*10^5$ g	$7.361347*10^5$ g

Table 10: Step size settings used for the actual simulation run. The values are chosen to provide reasonable accuracy and speed based on test results shown in Table 9.

Type	Step Size Value
Time	1 (LS)
Energy	10 (MeV)
Incident Angle	5 (deg)
CO2 Micro-Layer Thickness	0.1 (cm)
H2O Micro-Layer Thickness `	0.001 (cm)

A.3 Input Container Format & Values

Upon the execution of the model code, an input file is required. The input is a .mat file containing the following:

Table 11: Format and values of the various parameters inside the input variable. **a)** top level container structure. **b)** Ni definition. **c)** Medium definitions. **d)** Particle definition.

11 a)

Variable Name	Type	Usage
Ni	structure	Definition of Ni particles in the Pilling experiment.
mediums	cell array	Definition of target mediums in the order of placement on the surface from the top.
particles	cell array	Definition of incoming charged particles.

11 b) Variable **Ni** contains:

Variable Name	Type	Usage	Value
z	Int	Charge of Ni particles.	13+ (electron charge)
m_0	Double	Mass of Ni particles.	$9.63*10^{-26}$ (kg)
energy_transfer	cell array	Energy transfer fits for Ni particles in each target mediums.	{CO2_transfer_fit, H2O_transfer_fit}

11 c) Variable **mediums** contains medium definition structures in the order of placement from the top (CO_2 first, H_2O second). Each contains the following:

Variable Name	Type	Usage	Value for CO_2	Value for H_2O
A	double	Atomic mass of the medium	44.01	18.02
Z	double	Atomic number of the medium	22.0002	10.0027
rho	double	Density of the medium (in g/cm^3)	0.91	0.918
I	double	Mean excitation energy (in eV)	85	79.7
hwp	double	Plasma energy (in eV)	25.47	20.57
conductor	int	Boolean flag for conductors	0	0
k	double	Density effect parameters (see [29])	3	3
C	double	As above.	3.4103	3.5020
x1	double	As above.	2	2.5
x0	double	As above.	0.2000	0.24
a	double	As above.	0.4268	0.2065
thickness	1x2 cell	Seasonal thickness evolution fits for all rings & the LS dates for the start of deposition and the end of sublimation	{{CO2_ring1_fit, CO2_ring2_fit.... }, {[ring1_begin,ring1_end], [ring2_begin, ring2_end]....}}	{{H2O_ring1_fit, H2O_ring2_fit.... },{[ring1_begin,ring1_end],[ring2_begin,ring2_end]....}}
thickness_p	double	Pilling's ice sheet thickness in the experiment (in cm)	$4.0*10^{-5}$	$5.0*10^{-5}$
rho_p	double	Density of the medium used by Pilling (in g/cm^3)	1.8	1
species	cell	Dissociation & formation cross sections for the radiolytic species	{[CO $\sigma_{d,}$ CO σ_f], [O3 $\sigma_{d,}$ O3 σ_f],[CO3 $\sigma_{d,}$ CO3 σ_f]}	{[H2O2 $\sigma_{d,}$ H2O2 σ_f]}
species_names	cell	Names of the radiolytic species in medium	{'CO', 'O3', 'CO3'}	{'H2O2'}
step_size	double	default step size setting for micro-layer (obsolete → overwritten by configuration file)	$7.9121*10^{-5}$	$5.4466*10^{-5}$
name	string	Name of the medium	'CO2'	'H2O'

11 d) Variable **particles** contains charged particle definitions. Currently it only defines protons:

Variable Name	Type	Usage	Value
z	Int	Charge.	1
m_0	Double	Mass of the particle in kg.	$1.67*10^{-27}$
transfer	Cell	Stopping power cache (stopping power of the particle in the mediums for the simulated energy range). Each medium requires a stopping power, ordered the same way as in the **mediums** variable.	{cfit1, cfit2...}
name	String	Name of the particle.	'Proton'

Appendix B

The source code of the model (written for MATLAB R2010b) is included here. All scripts should be placed under the same directory.

front_end.m

This is the main method of the model. It also controls the format of the output files.

Code:

```
function [product_result, lat_centers] = front_end()

global input prod_track

input = configuration();

% TODO: lat_range not generalized

lat_range = input.mediums{1}.thickness{2};

lat_centers = zeros(1, numel(lat_range));

for i = 1:numel(lat_range)

lat_centers(i) = (lat_range{i}(1) + lat_range{i}(2))/2;

end

product_result = cell(1,numel(lat_centers));

[headername, path] = uiputfile('*.hdr', 'Save Header File for Output As...');

fid = fopen(strcat(path, headername), 'w');

fprintf(fid, sprintf('Simulating CR Irradiation from LS %d to LS %d¥n', input.start_time,
input.end_time));

fprintf(fid, sprintf('Time Step Size: %d (LS)¥n', input.time_step_size));

fprintf(fid, sprintf('Medium1: %s, layer step size: %d (cm)¥n', input.mediums{1}.name,
input.mediums{1}.step_size));
```

```
fprintf(fid, sprintf('Medium2: %s, layer step size: %d (cm)¥n', input.mediums{2}.name,
input.mediums{2}.step_size));

fprintf(fid, sprintf('CR Particles: %s¥n', input.particles{1}.name));

fprintf(fid, sprintf('Energy Range: %d to %d (MeV)¥n', input.start_energy, input.end_energy));

fprintf(fid, sprintf('Energy Step Size: %d (MeV)¥n', input.energy_step_size));

fprintf(fid, sprintf('Incident Angle Range: %d to %d (deg)¥n', input.angle_low,
input.angle_high));

fprintf(fid, sprintf('Incident Angle Step Size: %d (deg)¥n', input.angle_step_size));

fclose(fid);

for i = 1:numel(lat_centers)

lat = lat_centers(i);

fprintf('Now processing ring centered at lat %d deg¥n',lat_centers(i));

% setting up file to track production per time step

prod_track1 = fopen(strcat(path, sprintf('RING%d_prod_CO2.txt', i)), 'w');

prod_track2 = fopen(strcat(path, sprintf('RING%d_prod_H2O.txt', i)), 'w');

prod_track = [prod_track1, prod_track2];

% compute product from a unit area on the ring

ring_product = ring_progression(lat);

for index = 1:numel(prod_track)

fclose(prod_track(i));

end

% compute area of the ring

radius = input.radius;

% convert radius to cm

radius = radius*100000;
```

```
range = lat_range{i};

A = area(radius, range(1), range(2), 0, 360);

filename = sprintf('RING%d.txt', i);

fid = fopen(strcat(path, filename), 'w');

fprintf(fid, sprintf('Latitude Center: %d¥n', lat));

fprintf(fid, sprintf('Latitude Range: %d to %d¥n', range(1), range(2)));

fprintf(fid, sprintf('Area: %d (km^2)¥n¥n', A/1E10));

fprintf(fid, sprintf('Radiolysis Products (g):¥n¥n'));

for j = 1:numel(ring_product)

% scale product up by the area of the ring

ring_product{j} = ring_product{j}*A;

end

product_result{i} = ring_product;

for index = 1:2

fprintf(fid, sprintf('Medium %d (%s) Species:¥n', index,
input.mediums{index}.name));

names = input.mediums{index}.species_names;

[rows,cols]=size(names);

for indicator=1:rows

if cols > 1

fprintf(fid,'%s,',names{indicator,1:end-1});

fprintf(fid,'%s¥n',names{indicator,end});

else

fprintf(fid,'%s¥n',names{indicator,end});
```

```
end

end

vals = ring_product{index};

[rows,cols] = size(ring_product{index});

for indicator=1:rows

if cols > 1

fprintf(fid,'%d,',vals(indicator,1:end-1));

fprintf(fid,'%d¥n¥n',vals(indicator,end));

else

fprintf(fid,'%d¥n¥n',vals(indicator,end));

end

end

end

fclose(fid);

end

end
```

configuration.m

This loads the input container (configurations of target mediums & cosmic ray particles). Settings for parameters such as step sizes are also contained here. See Appendix A.X for an explanation on the values used here.

Code:

```
function [input] = configuration()

uiopen('load');

input = struct();

% set mediums and particles

input.mediums = mediums;

input.particles = particles;

% set up energy transfer cache for the particles

for i = 1:numel(input.particles)

particle = input.particles{i};

particle.cache = cell(1,numel(input.mediums));

particle.cache_exists = zeros(1,numel(input.mediums));

for j = 1:numel(input.mediums)

try

medium = input.mediums{j};

msg = sprintf('Select energy transfer cache file for particle %s on medium %s',
particle.name, medium.name);

[filename, pathname] = uigetfile('*.mat', msg);

% cache file contains one single cell array of the following

% format:

% cell_array = {E(x)_fit, x(E)_fit}, E = energy of particle,
```

```
% x = distance particle traveled (for a 50 GeV particle)

S = struct2cell(load(strcat(pathname, filename)));

S = S{1};

particle.cache{j} = S;

particle.cache_exists(j) = 1;

catch error

display(error.message);

display('Error setting up energy transfer cache for particle! Computation might
be slow.');

end

end

input.particles{i} = particle;

end

% simulation layer step size overide (default gives equivalent mass

% thickness to one Pilling layer)

% CO2 override

input.mediums{1}.step_size = 0.1; % in cm

% H2O override

input.mediums{2}.step_size = 0.001;

step_size_array = zeros(1,numel(input.mediums));

for i = 1:numel(input.mediums)

step_size_array(i) = input.mediums{i}.step_size;

end

% start time and end time (in Ls)
```

```
input.start_time = 180;

input.end_time = 540;

% time step size (in Ls)

input.time_step_size = 1;

% low and high energy bounds (in MeV)

input.start_energy = 10;

input.end_energy = 50000;

% setting energy step size to the minimum safe value

% input.energy_step_size = 1/min(step_size_array); % (in MeV)

input.energy_step_size = 10;

if 200/min(step_size_array)*abs(input.start_energy - input.end_energy)/input.energy_step_size
> 10^7

display('Potentially unsafe for fit evaluation. Increase layer/energy step
size!');

end

% low and high incident angle bounds (in deg)

input.angle_low = 0;

input.angle_high = 90;

input.angle_step_size = 5; % (in deg)

% radius of the planet (in km)

input.radius = 3390;

% Ni definition used by Pilling

input.Ni = Ni;

end
```

area.m

This helper function is used to compute the surface area of each simulated ring covering the polar region.

Code:

```
% compute surface area encapsulated by the specified lat and lon range on a

% sphere of radius R

% note: latitudes must follow 90N to 90S format.

% longitudes go from 0 to 360.

function [A] = area(R, lat_start, lat_end, lon_start, lon_end)

% convert angles to radians

theta_start = (90 - max(lat_start, lat_end))*pi()/180;

theta_end = (90 - min(lat_start, lat_end))*pi()/180;

lon_start = lon_start*pi()/180;

lon_end = lon_end*pi()/180;

lon_range = abs(lon_end-lon_start);

handle = @(theta) R^2*sin(theta);

A = lon_range*abs(quad(handle, theta_start, theta_end));

end
```

ring_progression.m

This is the main container for the various components of model computation.

Code:

```
function [ring_product] = ring_progression(lat)

global input t prod_track

start_time = input.start_time;

end_time = input.end_time;

energy_low = input.start_energy;

energy_high = input.end_energy;

energy_step_size = input.energy_step_size;

% time step size in Ls

time_step_size = input.time_step_size;

% convert to seconds

step_seconds = 686.98*24*60*60/360*time_step_size;

% CO2 medium definition

CO2 = input.mediums{1};

% H2O medium definition

H2O = input.mediums{2};

% incident proton

H = input.particles{1};

depositions = {[], []};

empty_prod = {production(CO2, 1, 0), production(H2O, 2, 0)};

ring_product = empty_prod;

% format header for file tracking production per time step
```

```
for i = 1:numel(prod_track)
fprintf(prod_track(i), sprintf('%s Species:¥n', input.mediums{i}.name));
fprintf(prod_track(i), 'LS,');
names = input.mediums{i}.species_names;
[rows,cols]=size(names);
for indicator=1:rows
if cols > 1
fprintf(prod_track(i),'%s,',names{indicator,1:end-1});
fprintf(prod_track(i),'%s¥n',names{indicator,end});
else
fprintf(prod_track(i),'%s¥n',names{indicator,end});
end
end
end
% set initial layer to 0 -> initial time must be such that all
% rings have no ice on it! might want to change this so user can
% supply initial conditions instead.
prev_num_layers = [0, 0];
% toggling flag for vectorized computation
vectorized = prod(H.cache_exists);
% toggling flag for guaranteed angle integration accuracy using adaptive
% quadrature numerical integration from matlab
angle_int_debug = 0;
% start simulation
```

```
for t = start_time:time_step_size:end_time

% vertical effective mass thickness TODO: generalize

CO2_thickness = thickness_retrieve(CO2, t, lat);

H2O_thickness = thickness_retrieve(H2O, t, lat);

fprintf('Time (Ls): %d¥n', t);

fprintf('CO2 thickness (g/cm^2): %d¥nH2O thickness (g/cm^2): %d¥n',
CO2_thickness, H2O_thickness);

% num of layers

num_layers = [round(CO2_thickness/(CO2.step_size*CO2.rho)),...

round(H2O_thickness/(H2O.step_size*H2O.rho))];

fprintf('Total number of layers from all mediums: %d¥n', sum(num_layers));

% first check if there's any release

for i = 1:numel(input.mediums)

if num_layers(i) < prev_num_layers(i)

% compute product release using current accumulated energy

% deposition in the layers

[left_over, product] = product_release(i, depositions{i}, ...

prev_num_layers(i)-num_layers(i));

% set current energy deposition state in the layers to

% the amount left over after release.

depositions{i} = left_over;

% accumulated species production

ring_product{i} = ring_product{i} + product;

else
```

```
product = empty_prod{i};
end
% write released product to tracker file
cols = numel(product);
fprintf(prod_track(i), '%d,', t);
if cols > 1
fprintf(prod_track(i),'%d,',product(1:end-1));
fprintf(prod_track(i),'%d¥n',product(end));
else
fprintf(prod_track(i),'%d¥n',product(end));
end
end
wait_message = sprintf(...
'Computing deposition over energy spectrum for date %d out of %d¥n', ...
t-start_time, end_time-start_time);
k = waitbar(0, wait_message);
% find total deposited energy in the ice layer by integrating over
% energy range
deposition_step = zeros(sum(num_layers), 1);
% compute deposition over all incident angles and starting particle
% energies
if vectorized && angle_int_debug
display('Computing angle integration using adaptive quadrature! Might be
slow!');
```

```
% compute integral using adaptive quad rule by matlab

% might be very slow!

handle = @(angle) vectorized_energy_deposit_per_angle(H, ...

input.mediums, num_layers, energy_list, angle)*energy_step_size;

deposition_step = quadv(handle,input.angle_low,input.angle_high);

else

% compute normally using simple quadrature (trapezoidal) with user

% defined angle step size.

energy_list = ((energy_low+energy_step_size/2):energy_step_size:(energy_high-
energy_step_size/2))';

% angle_list =
((input.angle_low+input.angle_step_size/2):input.angle_step_size:(input.angle_high-
input.angle_step_size/2))';

angle_list = (input.angle_low:input.angle_step_size:input.angle_high)';

for i = 1:numel(angle_list)

waitbar(i/numel(angle_list));

% check if particle has complete cached energy transfer fit

if vectorized

% use vectorized computation

angle_a = angle_list(i) - input.angle_step_size;

angle_b = angle_list(i);

% computing using cache!

f_a = vectorized_energy_deposit_per_angle(H,input.mediums,...

num_layers,energy_list,angle_a)*energy_step_size;

f_b = vectorized_energy_deposit_per_angle(H,input.mediums,...
```

```
num_layers,energy_list,angle_b)*energy_step_size;
deposition_step = deposition_step + (f_b + f_a)/2*input.angle_step_size;
% deposition_step = deposition_step + vectorized_energy_deposit_per_angle(H, ...
% input.mediums,num_layers,energy_list,angle)*energy_step_size...
% *input.angle_step_size;
else
% TODO: need to be fixed to use trapezoidal rule
handle = @(E) energy_deposit_per_angle(H, input.mediums, ...
num_layers, E, angle)*energy_step_size;
result = arrayfun(handle, energy_list, 'UniformOutput',false);
% compute sum from all depositions from all energies
deposition_step = deposition_step + sum(cell2mat(result), 2)*input.angle_step_size;
end
%
% for j = 1:numel(energy_list)
%
% E = energy_list(j);
%
% deposition_step = deposition_step + energy_deposit_per_angle(H,...
% input.mediums, num_layers, E, angle)*energy_step_size;
% end
% deposition_step = deposition_step*input.angle_step_size;
end
end
```

```
close(k);

% set CO2 deposition step over this time step

CO2_dep_step = deposition_step(1:num_layers(1))*step_seconds;

% add to current deposition

depositions{1} = add_deposition_energy(depositions{1}, CO2_dep_step);

if num_layers(2) > 0

% set H2O deposition step over this time step

H2O_dep_step = deposition_step(num_layers(1)+1:end)*step_seconds;

% add to current H2O deposition

depositions{2} = add_deposition_energy(depositions{2}, H2O_dep_step);

end

% set number of layers that were seen in the previous time step

% for CO2

prev_num_layers(1) = num_layers(1);

% for H2O

prev_num_layers(2) = num_layers(2);

end

end
```

thickness_retrieve.m

This retrieves the ice thickness from the input container.

Code:

```
% retrieve the thickness of the medium by latitude and time
%
% Arguments:
% time: the time of interest (in Ls)
% lat: latitude for the center line of the ice ring
% return: ice thickness approximated to 1 cm (or the equivalent mass
% effective thickness)
function [result] = thickness_retrieve(medium, time, lat)
index = find((time > 360) | (time < 0));
time(index) = mod(time(index), 360);
% Non-zero thickness data fits
fit = medium.thickness{1};
% latitude range of the ring {[begin, end];[begin2,end2];.....}
lat_range = medium.thickness{2};
% % time range for non-zero thickness {[T_start, T_end]; [T_start2,
% % T_end2];....}
% time_range = medium.thickness{3};
index = 0;
% locate ring using lat
for i = 1:numel(lat_range)
% set index to the current ring
```

```
index = i;
if (lat_range{i}(1) <= lat) && (lat <= lat_range{i}(2))
% found the right ring, break
break;
end
if i == numel(lat_range)
display('Location outside of defined thickness data range!');
index = 0;
end
end
if index ~= 0
% retrieve thickness
% if (time > time_range{index}(1)) && (time < time_range{index}(2))
% inside of range for nonzero thickness
result = fit{index}(time); % kg/m^2
% deji's fit is in kg/m^3!
% convert to g/cm^2
result = 1/10*result;
% end
else
result = -1;
end
end
```

vectorized_energy_deposit_per_angle.m

This computes the total amount of energy deposited into all the current micro-layers inside the ice from charged particles of a specific incident angle. This code is optimized for speed by using energy transfer cache and vectorized computation.

Code:

```
function [deposition] = vectorized_energy_deposit_per_angle(particle, mediums, num_layers, E, incident_angle)

% only works if particle contains complete copies of energy transfer cache!

% col vector representing the current energy

curr_energy = E;

tot = sum(num_layers);

% column vector representing the deposition of particles with energies

% specified by vector E in each layer of the mediums.

deposition = zeros(tot,1);

if tot > 0

current_layer = 1;

% compute flux as a function of incident angle

flux = flux_per_angle(E, incident_angle);

for medium_index = 1:numel(mediums)

medium = mediums{medium_index};

% % total number of layers from previous mediums (not including the

% % current one) traversed by the particle

% past_layers = sum(num_layers(1:medium_index-1));

% number of layers in the current medium

medium_num_layers = num_layers(medium_index);
```

```
cache = particle.cache{medium_index};
% compute deposition from cache.
delx = medium.step_size;
layer_path_length = delx/cos(incident_angle*pi()/180);
% speed hack for big incident angle
% if layer_path_length is bigger 10000 cm (ie: more than 100 meters)
% then the particle deposits ALL of its energy in the first
% layer
if layer_path_length > 10000
deposition(current_layer) = sum(curr_energy.*flux);
break;
else
if medium_num_layers > 0
% energy as a function of distance
energy_distance = cache{1};
% distance as a function of energy
distance_energy = cache{2};
% find distance offset for the current particle at the starting
% energy
% col vector containing offset for the current energy
dist_offset = distance_energy(curr_energy);
% index of all negative distance offset (caused by
% inaccuracies in the distance_energy fit)
negative = dist_offset < 0;
```

```
% reset them to 0
dist_offset(negative) = 0;
% basic position template for all particles
position = 0:layer_path_length:layer_path_length*medium_num_layers;
% each row representing the positions of a particle of
% different energy as they move through the medium
position = repmat(position, numel(curr_energy),1);
% add distance offset to the positions
position = bsxfun(@plus, position, dist_offset);
pos_size = size(position);
% TODO: deal with very big position matrix (ie: when
% energy_step_size and layer step_size is small)
% if prod(pos_size) > 10^6
%
% divide = round((prod(pos_size)/10^6));
%
%
% end
% column vector of the format:
% [val(1,1);val(2,1);val(1,2);val(2,2)....]
energy = energy_distance(position);
% reset all energies less than 0 back to 0 (particle lost
% all of its energy prior to reaching that layer).
index = energy < 0;
```

```
energy(index) = 0;
% convert back to the format matching position
energy = reshape(energy, pos_size);
% starting energy at each layer
energy_start = energy(1:pos_size(1),1:pos_size(2)-1);
% resulting energy after each layer
energy_end = energy(1:pos_size(1), 2:pos_size(2));
% multiply with flux
deposition_per_E = bsxfun(@times, energy_start - energy_end, ...
flux);
% summing up depositions from particles of all specified
% energies for each layer
result = sum(deposition_per_E, 1)';
deposition(current_layer:current_layer+medium_num_layers-1)...
= result;
dep_size = size(energy_end);
curr_energy = energy_end(:, dep_size(2));
current_layer = current_layer + medium_num_layers;
end
end
end
end
end
```

energy_deposit_per_angle.m

This is the non-vectorized version of energy deposition code. It can work without the energy transfer cache by going through SRIM and Bethe-Bloch computation directly. However it will be very slow! Recommended approach is to utilize this to compute and save energy transfer cache then use vectorized version for actual model simulation.

Code:

```
% This function computes the total amount of energy deposited into the ice

% layers according to the incident angle of the cosmic rays.

% RETURN:

% deposition: array containing deposited energy from all of the particles

% with incident_angle in each layer

function [deposition] = energy_deposit_per_angle(particle, mediums, num_layers, E,
incident_angle)

% waitbar(incident_angle/90);

% set current energy of particle to starting energy E

curr_energy = E;

% energy deposition into all layers for the incident angle

% total number of layers from all mediums

tot = 0;

for i=1:numel(num_layers)

tot = tot + num_layers(i);

end

deposition = zeros(tot, 1);

% compute flux as a function of incident angle

flux = flux_per_angle(E, incident_angle);

if tot > 0
```

```
current_layer = 1;
for medium_index = 1:numel(mediums)
medium = mediums{medium_index};
% total number of layers from previous mediums (not including the
% current one) traversed by the particle
past_layers = sum(num_layers(1:medium_index-1));
% number of layers in the current medium
medium_num_layers = num_layers(medium_index);
% check if cache exists for the current medium
if particle.cache_exists(medium_index)
cache = particle.cache{medium_index};
% compute deposition from cache.
delx = medium.step_size;
layer_path_length = delx/cos(incident_angle*pi()/180);
% speed hack for big incident angle
% if layer_path_length is bigger 10000 cm (ie: more than 100 meters)
% then the particle deposits ALL of its energy in the layer
if layer_path_length > 10000
deposition(current_layer) = curr_energy;
curr_energy = 0;
else
if medium_num_layers > 0
% energy as a function of distance
energy_distance = cache{1};
```

```
% distance as a function of energy
distance_energy = cache{2};
% find distance offset for the current particle at the starting
% energy
% col vector containing offset for the current energy
dist_offset = distance_energy(curr_energy);
% reset them to 0
if dist_offset < 0
dist_offset = 0;
end
% distance traveled by the particle as it traverses through the medium
% (with offset)
position = (0:layer_path_length:layer_path_length*medium_num_layers)...
+ dist_offset;
% compute energy at each position
energy = energy_distance(position);
% double check if there's any negative values in energy
% (ie: traveled too far)
index = find(energy < 0);
if ~isempty(index)
%negative values exist! particle will lose all of its
%energy before traversing the entire medium.
% index of where the first negative value occurs
index = index(1);
```

```
% only take the energy up to that point
energy = energy(1:index);
end
% shifted energy vector (the energy values at the end of
% the layer
energy_end = energy(2:end);
% starting energy at each layer
energy_start = energy(1:end-1);
diff = energy_start - energy_end;
if ~isempty(diff)
% multiple layers received deposition
% set the deposition to the the difference between starting
% and resulting energy at each layer
deposition(current_layer:(current_layer+numel(energy_end)-1)) = ...
diff;
%
% try
% set current energy of the particle
curr_energy = energy_end(end);
else
% only the first layer recieved deposition
deposition(current_layer) = curr_energy;
curr_energy = 0;
end
```

```
% push current_layer to the next medium
current_layer = current_layer + medium_num_layers;
% particle has lost all of its energy
if numel(index) > 0
curr_energy = 0;
end
end
end
else
% cache doesn't exist. Do it the hard way.
while (curr_energy > 0) && ((current_layer - past_layers) < ...
medium_num_layers)
% compute & store energy deposited in each layer
[deposit curr_energy] = layer_energy_deposit_per_angle(curr_energy,...
particle, medium, medium_index, incident_angle);
if curr_energy > E
display('Problem with energy transfer! Double check!¥n');
return;
end
deposition(current_layer) = deposit;
current_layer = current_layer + 1;
end
end
end
```

```
end

% scale deposited energy up by flux

deposition = deposition*flux;

end
```

layer_energy_deposit_per_angle.m

This code is used to compute the amount of energy deposited into a single micro-layer of ice. This is only used by the non-vectorized energy deposition code.

Code:

```
function [energy_deposit, curr_energy] = layer_energy_deposit_per_angle(starting_energy, particle, medium, medium_index, incident_angle)

global input

step_size = medium.step_size;

% convert angle to radians

incident_angle = incident_angle*pi()/180;

% step_size = 20*10^(-4); % 20 um for Ni simulation

% set delx to step_size*medium density (0.91 g/cm^2 for CO2 ice) so about 1 step size cm on avg

delx = step_size*medium.rho; % (g/cm^2) <- vertical layer thickness

% single mass effective thickness unit the particle must travel to reach

% one delx verticle effective thickness

% problem: incident_angle ~ 90 -> explode!

layer_path_length = delx/cos(incident_angle);

traveled_length = 0;

curr_energy = starting_energy;
```

```
energy_deposit = 0;
% speed hack for big incident angle
% if layer_path_length is bigger than 10000 times the density of the
% medium (ie: more than 100 meters) then the particle deposits ALL of
% its energy in the layer
if layer_path_length > medium.rho*10000
energy_deposit = energy_deposit + curr_energy;
curr_energy = 0;
else
while traveled_length <= layer_path_length && curr_energy > 0
dedx = 0;
try
% see if particle contains user supplied optional energy transfer
% fit
energy_transfer = particle.energy_transfer{medium_index};
dedx = energy_transfer(curr_energy);
catch error
%fprintf('No user supplied energy-transfer fit! Using customized bethe-bloch (slow)!¥n');
% no optional energy transfer fit. compute energy deposition (MeV/(g/cm^2))
% using bethe_bloch
dedx = abs(bethe_bloch(curr_energy, particle, medium));
end
% energy transfered in the step
step_transfer = dedx*delx;
```

```
% fprintf('dedx over distance: %d', dedx*delx);

if curr_energy > step_transfer

curr_energy = curr_energy - step_transfer;

energy_deposit = energy_deposit + step_transfer;

traveled_length = traveled_length + delx;

else

% not enough energy left to travel delx

% dump all energy from the particle

energy_deposit = energy_deposit + curr_energy;

curr_energy = 0;

break;

end

end

end

end
```

bethe_bloch.m

This is the customized implementation of Bethe-Bloch formula.

Code:

```
% this function computes dE/dx usig bethe-bloch formula

%

% Arguments:

% Ek: kinetic energy of incident particle (in MeV)

% returns:
```

```
% beta_gamma: beta*gamma
% dedx: energy deposited per distance travelled (MeV/(g/cm^2)) ie: mass
% stopping power
function [dedx] = bethe_bloch(E, particle, medium)
% speed of light
c = 3.0*10^8; % m/s
% K (MeV*cm^2/g)
K = 0.307075;
% mass of electron (kg)
m_e = 9.11*10^(-31);
% me*c^2 (MeV)
mec2 = 0.510998918;
dedx = zeros(1, numel(E));
for i = 1:numel(E)
% E is kinetic energy. compute lorentz factor
gamma = E(i)*1.6*10^(-13)/(particle.m_0*c^2) + 1;
% ratio to speed of light
beta = sqrt(1-1./gamma.^2);
% compute maximum kinetic energy transfer
Tmax = 2*mec2*beta^2*gamma^2/...
(1 + 2*gamma*m_e/particle.m_0 + (m_e/particle.m_0)^2);
% % compute low energy shell correction (ignored)
% eta = beta.*gamma;
%
```

```
% C = (0.422377*eta.^(-2) + 0.0304043*eta.^(-4) - 0.00038106*eta.^(-6))...

% *10^(-6)*medium.I^2 + (3.858019*eta.^(-2) - 0.1667989*eta.^(-4) + ...

% 0.00157955*eta.^(-6))*10^(-9)*medium.I^3;

% compute density effect correction del/2

delta = del(beta, gamma, medium);

% M: particle rest mass (MeV/c^2)

M = particle.m_0*c^2/(1.6*10^-19)/10^6;

% bremsstrahlung radiation term (is it required for protons?)

delE = K/(4*pi)*medium.Z/medium.A*(1/137)*(log(2*E(i)/...

M) - 1/3*log(2*Tmax/mec2)).*(log(2*Tmax/mec2)).^2;

% compute energy transfer

% dedx(i) = K*(medium.Z/medium.A)./beta.^2.*...

% (1/2*log(2*0.511*beta.^2.*gamma.^2.*Tmax/medium.I^2) -beta.^2 - delta/2 - C/medium.Z+
1/8*Tmax.^2/(gamma.*M).^2) + delE;

phi = (1/2*log(2*beta^2*gamma^2*mec2*Tmax/(medium.I)^2) - beta^2 - 6.5*delta -
1/8*Tmax^2/(gamma*M)^2);

D = K*particle.z^2*medium.Z/medium.A*1/beta^2;

dedx(i) = D*phi;

if i == numel(E)

display('High Energy');

fprintf('E: %d ¥n', E(i))

fprintf('beta: %d ¥n', beta);

fprintf('gamma: %d ¥n', gamma);

fprintf('Tmax: %d ¥n', Tmax);
```

```
fprintf('Shell correction: %d ¥n', C/medium.Z);

fprintf('density effect: %d ¥n', delta/2);

fprintf('Bremsstrahlung: %d ¥n', delE);

end

end

end
```

del.m

This computes the density effect for the Bethe-Bloch implementation.

Code:

```
function [del] = del(beta, gamma, medium)

% compute x

% x = log10(beta.*gamma);

x = log(beta*gamma)/log(10);

try

x0 = medium.x0;

x1 = medium.x1;

a = medium.a;

C = medium.C;

catch error

try

C = medium.C;

catch error
```

```
display('Missing C!');
del = 0;
return;
end
if x1 == 0 && x2 == 0
if medium.I < 100
x1 = 2;
if C < 3.681
x0 = 0.2;
else
x0 = 0.326*C - 1;
end
else
x1 = 3;
if C < 5.215
x0 = 0.2;
else
x0 = 0.326*C - 1.5;
end
end
end
end
end
k = 3;
del = 0;
```

```
if x >= x1

del = 2*log(10)*x - C;

elseif x0 <= x && x < x1

del = 2*log(10)*x - C + a*(x1 - x)^k;

elseif x < x0

if medium.conductor

del0 = 2*log(10)*x0 - C + a*(x1-x0)^k;

% try

% del0 = medium.del0;

% catch error

% del0 = 0;

% end

del = del0*10^(2*(x-x0));

end

end

end
```

add_deposition_energy.m

Helper function that adds the computed energy deposition into the existing micro-layers.

Code:

```
function [result] = add_deposition_energy(current_deposition, new_deposition)

if numel(current_deposition) == 0

% no current layers to begin with

result = new_deposition;
```

```
return;
elseif numel(current_deposition) < numel(new_deposition)
% new layers have formed. new_deposition specifies the top.
% old layers start from i = numel(deposition) -
% numel(current_deposition) + 1
num_layers = numel(new_deposition);
big = new_deposition;
old_start = num_layers - numel(current_deposition) + 1;
temp = zeros(num_layers, 1);
temp(old_start:end, 1) = current_deposition;
else
% no new layers formed (may have lost some). simply add the two
% together.
num_layers = numel(current_deposition);
big = current_deposition;
temp = zeros(num_layers, 1);
temp(1:numel(new_deposition), 1) = new_deposition;
end
result = big + temp;
end
```

flux_per_angle.m

This function computes the attenuated differential flux density per particle incident angle. It also contains the settings for Mars atmospheric parameters.

Code:

```
function [flux] = flux_per_angle(Ek, theta)

% convert to radians

theta = theta*pi()/180;

% atmospheric pressure at the location (mb) TODO: variable pressure

P = 6.36;

% gravitational constant;

g = 3.711;

% (vertical) mass thickness above the point

M = P*100/g/10; % g/cm^2

% effective mass thickness after with theta incident angle

M_eff = M./cos(theta);

% nuclear interaction length of Mars atmosphere

interaction_length = 88.9; % g/cm^2

% vertically downward primary flux per solid angle (radiance)

prim_flux = primary_flux(Ek);

% convert to flux per theta & phi

prim_flux = prim_flux.*cos(theta).*sin(theta);

% flux after attenuation

prim_flux = prim_flux.*exp(-M_eff/interaction_length);

% compute secondary flux
```

```
sec_flux = secondary_flux(Ek);

% convert to flux per theta & phi

sec_flux = sec_flux.*cos(theta).*sin(theta);

% Note: assume NO further attenutation on sec_flux as Mizuno's model is

% based on observation (although mass thickness at that altitude is about 3

% times smaller than that on Mars -> underestimation on secondary flux?)

% total flux (converted to counts/(sec*cm^2*MeV))

flux = 2*pi().*(prim_flux + sec_flux)/(1E4);

end
```

primary_flux.m

Computes the unattenuated primary flux as discussed in Section 2.2.

Code:

```
% computes the CR TOA primary proton flux (using Mizuno's model)

function [flux] = primary_flux(Ek)

% Convert Ek to Joules

Ek_J = Ek*10^6*1.6*10^(-19);

% constants for the particle

% protons:

M = 1.67*10^(-27); % mass (kg)

Z = 1; % atomic number

% compute primary energy flux

% setting constants:

% c: speed of light (m/s)
```

```
% e: electron charge (C)

% phi: solar activity (V) (550 MV - 1100 MV)

c = 3.0*10^8;

e = 1.602*10^(-19);

phi = 650*10^6;

% Ek_J: kinetic energy of particle

% Z: atomic number of particle

% M: mass of particle

% Compute rigidity with shifted energy (due to solar modulation)

R_shift = rigidity(Ek_J + Z*e*phi, M);

mod_Ek_J = unmod(R_shift);

mod_Ek_J = mod_Ek_J.*((Ek_J+M*c^2).^2-(M*c^2)^2)./((Ek_J+M*c^2+Z*e*phi).^2-(M*c^2)^2);

%((Ek_J.^2+2*Ek_J*M*c^2)/(Ek_J.^2+2*Ek_J*M*c^2+2*Ek_J*Z*e*phi+...

%2*M*c^2*Z*e*phi+Z^2*e^2*phi^2));

% proportionality constant to match Mizuno's scale

rho = 1/10^24; %/10^53;

flux = rho*mod_Ek_J;

end
```

Unmod.m

Unmodulated power law (Eqn 6) for top of the atmosphere primary proton flux.

Code:

```
function [result] = unmod(R)

% convert R to GV

R = R/10^9;

% setting constants for primary spectrum

A = 23.9; % (counts/(s*sr*m^2*MeV))

a = 2.83;

result = A*(R).^(-a);

end
```

secondary_flux.m

Computes secondary proton flux (see Section 2.2).

Code:

```
% This function models the secondary proton flux near Martian surface. We

% use high geomagnetic latitude (0.9 < theta_m < 1) and all associated

% cutoff PL model parameters. Upward flux is ignored because this is flux

% near the surface.

% Ek in MeV

function [sec_flux] = secondary_flux(Ek)

% Set parameters (see Mizuno Kamae, Table 1. 0.9 < theta_m < 1

F_0 = 0.44;

F_1 = 0.037;
```

```
a = 1.98;

Ecut = 0.21; % GeV

% % Convert Ek to GeV

% Ek_GeV = Ek/1000;

% vectorized version

low = Ek(10 <= Ek & Ek <= 100);

high = Ek(Ek > 100);

rest = Ek(Ek < 10);

sec_flux_low = F_0*(low/100).^(-1);

sec_flux_high = F_1*(high/1000).^(-a).*exp(-((high/1000)/Ecut).^(-a+1));

sec_flux_rest = rest*0;

[rows, cols] = size(Ek);

if (rows == 1) && (cols == 1)

sec_flux = [sec_flux_rest,sec_flux_low,sec_flux_high];

elseif (rows == 1) && (cols > 1)

% Ek is a row vector. Concatenate horizontally.

sec_flux = horzcat(sec_flux_rest, sec_flux_low,sec_flux_high);

elseif (rows > 1) && (cols == 1)

% Ek is a column vector. Concatenate vertically.

sec_flux = vertcat(sec_flux_rest, sec_flux_low, sec_flux_high);

else

% Ek is a matrix! Problem

display('Check Ek format!');

return;
```

```
end

end
```

product_release.m

This function computes the total amount of product released from destroyed/sublimed micro-layers.

Code:

```
% num_layers has to be smaller than the total numer of layers in product.

%

% ARGUMENTS:

% deposition: current accumulated energy deposited in each layer

% num_layers: number of micro layers to be destroyed

%

% RETURNS:

% release: total released chemical species stored in the destroyed layers

% remain: remaining energy deposition layers that have yet to be destroyed

function [remain, release] = product_release(medium_index, deposition, num_layers)

global input

medium = input.mediums{medium_index};

% zero vec

release = production(medium, medium_index, 0);

for i = 1:num_layers

layer_production = production(medium, medium_index, deposition(i));

release = release + layer_production;
```

```
end

remain = deposition(num_layers+1:end, 1);

end
```

production.m

Computes production of radiolytic species from equivalent fluence of Ni particles inside a micro-layer (Equation 5).

Code:

```
% Arguements:

% layer_energy: input energy flux into each ice layer (in MeV)

% time_length: the time elapsed for the layer to be under the energy flux

% given (in seconds).

function [result] = production(medium, medium_index, layer_energy)

global input

step_size = medium.step_size;

% how many times bigger each micro layer is compared to Pilling's layer

% size.

scaling_factor = step_size*medium.rho/(medium.rho_p*medium.thickness_p);

% col density of the medium inside the Pilling layer

N_1 = medium.rho_p*medium.thickness_p;

% amount of energy deposited into each Pilling layer.

layer_energy = layer_energy/scaling_factor;

% species (cell array each representing one of the species produced. each

% cell contains [dis_cross form_cross] for the species
```

```
species = medium.species;

result = zeros(1, numel(species));

Ni = input.Ni;

% energy deposited by a single Ni particle in the Pilling layer at 52 MeV.

energy_transfer = Ni.energy_transfer{medium_index};

dedx = energy_transfer(52);

single_Ni_deposit = dedx*medium.rho_p*medium.thickness_p;

if layer_energy > 0

% compute the equivalent Ni 13+ fluence from layer_energy

F = layer_energy/single_Ni_deposit;

for i = 1:numel(species)

% compute production for each species over the entire layer (in

% g/cm^2)

N_prod = N_1*pilling_ratio(species{i}(1), species{i}(2), F)*scaling_factor;

result(i) = N_prod;

end

end

end
```