Abstract

This work contributes to a simulation project which is looking at the possibility of active galactic nuclei jets as sources of heavy atomic nuclei as primary particles of high-energy cosmic rays. It will be examining atomic nuclei up to iron with highly relativistic speeds interacting with sub MeV photons from the active galactic nuclei’s accretion disk, leading to disintegration, and potential nuclear decays and corresponding energy distributions during a short time period thereafter.
**Statutory Declaration**  
Leopold-Franzens-Universität Innsbruck

I hereby declare that I have authored this thesis independently, that I have not used other than the declared sources and resources, and that I have explicitly marked all material which has been quoted either literally or by content as such.

This work has not been submitted in same or similar form as a master’s thesis or dissertation before.

---

_Date_  

Author’s Signature
Contents

1 Introduction 2

2 Theoretical Basis 3
   2.1 Cosmic Rays 3
      2.1.1 Heavy Nuclei HECRs 5
   2.2 AGNs as Sources 6
      2.2.1 Active Galactic Nuclei 7
   2.3 Photodisintegration 8
      2.3.1 TALYS 11
      2.3.2 Interaction Rate 12
   2.4 Nuclear Decay 13
      2.4.1 Energy Distribution 15
   2.5 Energy Ranges and Cutoffs 18

3 Implementation 20
   3.1 TALYS Calculations 20
      3.1.1 Isotope Cascades 21
      3.1.2 Disintegration Data Filtering 22
      3.1.3 Data File Storage 22
   3.2 Isotope and Particle Data 24
      3.2.1 Data File Structures 26
   3.3 Interaction Rate Calculations 27
   3.4 Decay Chain Time Evolution 29
   3.5 Software Architecture 31

4 Results 38

5 Discussion 47
1 Introduction

Nearly every branch of modern physics has advanced tremendously fast in the past decades. Technologies for engineering a setup in a laboratory getting to higher energies and better accuracy, and instruments being able to detect ever tinier particles and energies and signals of things further and further away, have sped up advancements of old and new theories.

Another aspect has been computer equipment getting more powerful each year, which has opened possibilities for calculating models and running simulations fast and complex like never before. Although simulations can only support theories or propositions rather than confirm them, they are a valuable part of getting an idea of what measurements should expect, and, in the case of astroparticle physics, where to look with what kind of instrument.

The case at hand will in the end be able to showcase a feasible possibility for sources of ultra-high energy cosmic rays consisting of heavy atomic nuclei, up to isotopes of the element iron. Simulations for different nuclei at a number of possible energies, disintegrations and decays will run through a large amount of results at each single step considered – of which there are many – to give an idea of what could be expected if the source of a cosmic ray was an active galactic nucleus. An example nucleus will be chosen to run through these calculations in the results section. Similar work has been published before by Rodrigues et al. [1], differing, among other things, in energy ranges and distribution spectra and some limits of considered probabilities.

Although science and technology have come far, many solved mysteries lead to new questions and never before seen measurement results. Further advancement continuously opens doors to new paths, which again will leave scientists struggling to explain and engineers frustrated to detect, before new technologies, in time, will provide the missing pieces.
2 Theoretical Basis

2.1 Cosmic Rays

The name cosmic rays (CR) describes particles that come from outside of the near region around Earth and are on their way to enter our atmosphere. If they do enter the Earth’s atmosphere, they will mostly interact with particles and atoms in it again and again by ionising or colliding with them, and will cascade through multiple reactions and processes into a particle shower, a so-called air shower, which will eventually reach its shower maximum, where no more new particles will be created.

These showers are characterised by their particle composition and shower maximum depth, describing how far into the atmosphere the CR particle’s shower grows until it has reached it’s maximum. They change with the energy of the primary particle, and can therefore be used to get information on such by ground based measurements detecting mostly only shower particles.

Simulations of air showers from different CRs are depicted in Figure 1. Their resulting particles differ in their spread and composition, such as heavy particles also featuring other hadrons. Bremsstrahlung is one of the effects that adds photons to these high-energy interactions, and pair-production creates electrons and positrons. Muons are also created during CR showers, but often decay into electromagnetic subshowers.

![Simulated CR showers with the primary particles being a proton (left) and an iron 56 nucleus (right). The lines indicate resulting electrons, positrons, muons, photons and hadrons. From [3], simulated with CORSIKA [4].](image-url)
Cosmic rays can be roughly categorised into particles of galactic or extra-galactic sources, or into their energy ranges, with higher energies typically being associated with extra-galactic sources. If the energies of the primary particles exceed $10^{18}$ eV, they are categorised as ultra-high energy cosmic rays (UHECRs). These are not particularly well known, in part because there are relatively few that reach Earth, as the collected data plot in Figure 2 shows, and the flux of CR decreases with higher energies. Therefore, detectors covering an area as large as possible are the best option, which makes the Pierre Auger Observatory [5] in central Argentina a good candidate and the detectors with the most publications related to UHECRs. This thesis will focus mostly on high-energy cosmic rays (HECRs) and reach up into the low UHECR range.

Figure 2: A collection of measurements of cosmic rays from different detector technologies, showing particle flux per area and time over energy. There are various points marked with additional information for a broader understanding. From [2]
2.1 COSMIC RAYS

2.1.1 Heavy Nuclei HECRs

Not only will this simulation consider specifically cosmic rays with high energies, but also those that are atomic nuclei from heavier elements. The Pierre Auger Collaboration has taken measurements of particle showers, of which the original primary CR particle would have been in the high or ultra-high energy range, and has compared those to simulations run with specific primary particles of the same energies. As seen in Figure 3, these comparisons suggest primary CR particles to move towards heavier elements at higher energies.

![Figure 3: Mean values of the shower maximum depth over the original cosmic ray particle energy. The lines show predictions from simulations for the original cosmic ray being a proton (red) or an iron nucleus (blue), for three models each. From [6]](image)

This may also be supported by the theoretical upper limit for CR energies, the Greisen-Zatsepin-Kuz’min (GZK) limit. It describes a maximum energy for high-energy cosmic rays travelling through space at $5 \times 10^{19}$ eV, above which they will interact with photons of the cosmic microwave background (CMB) and be slowed down with that loss of energy. However, it assumes that all high-energy CRs are protons, and as mentioned above, the Pierre Auger Collaboration suggested heavier elements in UHECRs, which would not be a contradiction for CRs exceeding the GZK limit. It was also suggested that the highest energies of CR measured would originate from extra-galactic sources close enough to us to avoid a significant decrease in flux by the GZK effect ($< 100$ Mpc) [7].
2.2 AGNs as Sources

The huge detector array of the Pierre Auger Observatory did measure some UHECR events, and with their data they are trying to find out more about those high energy particles. For example, still not well known are the sources of such highly accelerated heavy nuclei. In late 2007, the Pierre Auger Collaboration already released a publication [8] looking at candidates for possible extra-galactic sources – including AGNs – of the UHECRs measured by their arrival directions, and one specifically focusing on AGNs as possible sources [7].

Newer publications from the Pierre Auger Collaboration still seek to find angular correlation of UHECRs, for example with high-energy neutrinos [9] or starburst galaxies [10] and also AGNs [11], and although no decisive correlations have been found, AGNs are thought to be one of the possible source candidates. A release from 2016 featured Figure 4 for which AGNs of the Swift-BAT catalog [12] were overlaid with UHECR events of energies above $5.5 \times 10^{19}$ eV from Auger data deprived of light elements. Some events do overlap with the AGN positions in their arrival direction, but not enough to make a statement for their connection.

![Figure 4: Projection of the celestial sphere in galactic coordinates. Black dots show CR events detected by the Pierre Auger Observatory above $5.5 \times 10^{19}$ eV, deprived of light elements. The red stars symbolise the positions of AGNs from the Swift-BAT catalog. The blue colours indicates the observatories field of view, darker colouring meaning larger relative exposure, each colour having equal integrated exposure. From [11]](image-url)
2.2 AGNS AS SOURCES

2.2.1 Active Galactic Nuclei

The possible sources considered by the Pierre Auger Collaboration in their overlay in Figure 4 and also in the simulations done for this work, are active galactic nuclei (AGNs). An AGN consists of a supermassive black hole (SMBH) sitting at the centre of most galaxies. Figure 5 shows an illustration of such an AGN. It is declared as active for being a very luminous object, which is usually connected to it continuously gaining more energy through collecting mass from its accretion disk. AGNs can also form jets of various scales and energy ranges. A publication about relativistic AGN jets from late 2018 [13] states, that protons can be accelerated by these jets to energies in the region of PeV ($10^{15}$ eV) to EeV ($10^{20}$ eV). It does not mention atomic nuclei, but as an accumulation of protons and neutrons so to speak, they are accelerated by the same processes of Fermi acceleration and are also able to reach high energies like this.

![Figure 5: Illustration of an AGN and its surrounding components as well as the description of what an observer would be seeing from different view points. From [14]](image-url)
2.3 PHOTODISINTEGRATION

V. Lanzinger

The idea here is to consider the photons coming from the accretion disk surrounding the AGN. Previous work by Boettcher & Reimer et al. [15] fitted the spectral energy distributions (SEDs) of a set of radio-loud AGNs (blazars) with leptonic and hadronic models. One in particular showed a particularly strong contribution by the accretion disk photons to its SED in the energy range of about $10^{-2}$ eV to $10^2$ eV, the fit is depicted in Figure 6. Almost all other blazars in the sample also had well visible contributions from the accretion disk at similar energies.

![Figure 6: Hadronic model fit to the SED of a blazar with strong accretion disk contribution (dashed line). Other lines show electron-synchrotron (dotted), synchrotron self-Compton (dot-dashed) and proton-synchrotron (dot-dot-dashed) contribution. From [15]](image)

2.3 Photodisintegration

A nucleus created and accelerated far out in the universe has a long distance to go until it can hit Earth, especially the extra-galactic ones. During their travel through space, there are a number of things that can happen, like strong gravitational pulls from cosmic objects, electromagnetic deflection in large magnetic fields, or diverting interactions with other particles. As this is a Master’s thesis and part of a larger project, the scale will be kept to only one process – the photodisintegration of the accelerated jet nuclei through photons emitted by the AGN’s accretion disk.
Figure 7: Energy dependence of attenuation length for iron 56 at redshift $z = 0$ over Lorentz factor $\Gamma$. From [16]

Figure 8: Energy dependence of total photodisintegration cross section for iron 56 showing its contribution processes. The numbered curves 1, 2, 3 and 4($\alpha$) describe the contributions to the red GDR peak of how many nucleons are emitted in the process. From [16]
The energy dependence of the energy loss length of iron 56 is shown in Figure 7. The photodisintegration can be seen to contribute strongly in all shown energies, although the adiabatic energy loss due to the expansion of the universe dominates below a nucleus’ Lorentz factor of around $10^{8.8}$. Figure 8 shows a detailed energy dependence of the photodisintegration cross section for iron 56 and its contributing processes. A similar curve is expected to be calculated here for all nuclei.

The range of the giant dipole resonance (GDR) seen in Figure 8 results in mostly one or up to four nucleons being emitted. The protons and the neutrons in the nucleus as groups can be considered as fluids and the interaction described with hydrodynamics. In the case of GDR, the absorption of a photon leads to a collective vibration of the proton fluid against the neutron fluid. The GDR is generally a dominant feature in photonuclear absorption cross sections [17].

At higher energies the quasi-deuteron (QD) process becomes dominant, and it results in mostly multi-nucleon emissions. It is called like that, because the photon is absorbed by a nucleon pair of a neutron and a proton bound in a nucleus, which would be a deuteron if the pair were free. The interacting nucleon pair can then be emitted from the nucleus [17].

At even higher energies above the pion production threshold baryon resonance (photopion production) takes place, and it produces nucleons and pions. A $\Delta$ baryon resonance is excited by the absorbed photon energy and decays mainly in the nucleon plus pion channel, and if the products’ energies are high enough they will be separated from the nucleus. There is also the possibility that the pion is reabsorbed by a nucleon pair [17].

In general, photodisintegration occurs when a photon is absorbed by an atomic nucleus and the excitation of the nucleus becomes high enough to disperse it into nuclear fragments. Formulated for an example of the nucleus ($X$) dispersing into two smaller nuclei ($Y$ and $y$) it can be written as

$$X^A_Z + \gamma \rightarrow Y^{A-a}_{Z-a} + y^a_z.$$ (1)

The end products depend on the original nucleus and the energy of both parts in the interaction. As the actual reaction only sees the energy with which both partners meet, which is the one in the centre of mass system (cms), different combinations of photon and nucleus energy can give the same output. The cms energy $c_{\text{me}} = \sqrt{s}$ of the interaction with a nucleus of mass energy $m_N$ and a photon of energy $\epsilon_\gamma$ would be calculated through

$$s = m_N^2 + 2 E_N \epsilon_\gamma (1 - \beta_N \cos \theta)$$ (2)
where \( E_N \) describes the nucleus’ total energy, \( \beta_N \) its relativistic velocity and \( \theta \) describes the angle between the particles.

A variable used more often during this thesis is the photon’s energy in the rest frame of the interacting nucleus, which would be

\[
\varepsilon' = \gamma_N \varepsilon_\gamma (1 - \beta_N \cos \theta)
\]

with \( \gamma_N \) being the nucleus’ Lorentz factor.

One interesting value for these interactions is the minimum energy needed for a given nucleus to disintegrate, the threshold energy \( \varepsilon'_\text{thr} \). The parting that needs the smallest energy would be to lose one nuclear particle, so depending on the isotope the difference in energy of either missing a proton or a neutron. This means

\[
\varepsilon'_\text{thr} = \text{the smaller between } (m_{N,p} + m_p) - m_N \quad (4)
\]
\[
\text{and } (m_{N,n} + m_n) - m_N \quad (5)
\]

made up of the mass energies of a proton (p) and a neutron (n) and those of the corresponding isotope nuclei (N). These threshold energies are typically in the range of a few MeV.

The minimum energy can have the biggest effect, when the impact is head-on, so \( \theta = 180^\circ \), which can be used to get an estimate of the ratio between nucleus energy and photon energy for the threshold of a photodisintegration together with Equation (3) as

\[
\varepsilon'_\text{thr} = \gamma_N \varepsilon_\gamma (1 + \beta_N) \approx \gamma_N \varepsilon_\gamma 2 \quad (6)
\]

with \( \beta_N \approx 1 \) used for an estimation for the highly relativistic nuclei, but it should not be used if accurate calculations are needed.

### 2.3.1 TALYS

The atomic nuclei resulting from photodisintegration will have specific energy values with corresponding probabilities. The distribution of this energy after an interaction with a photon has taken place will be handled by the reaction code TALYS [18]. This code has been created and is maintained by A. J. Koning et al. for nuclear physics and in big part also for astrophysics. It simulates the collision of two particles or atomic nuclei and has a range of options for the calculation. The basics to use TALYS are very easy, but the great amount of optional parameters for additional output and information gives a wide range of usability.

The methods used by TALYS have been deemed to have clear advantages in the calculation of astrophysical reaction rates over previous codes.
developed for such purposes, like MOST, for example the inclusion of the pre-equilibrium reaction mechanism, multi-particle emission and detailed width fluctuation corrections [19]. The estimation used by TALYS to calculate the cross section for a reaction $I + a \rightarrow I' + a'$ is

$$\sigma_{aa'} = D_{\text{comp}} \lambda \sum_{J=\text{mod}(I+1,1)}^{I+1} \sum_{l=1}^{l_{\text{max}}+I+s} \frac{2J+1}{(2I+1)(2s+1)} \delta_\pi(\alpha) \delta_\pi(\alpha') \langle T_{J\alpha l}^{(J)}(E_{\alpha}) \rangle \langle T_{J'\alpha'l'}^{(J')}(E_{\alpha'}) \rangle W_{J\alpha l J'\alpha'l'} \ .$$

(7)

In this equation, $s$, $l$ and $j$ are the spin, orbital and total angular momentum for the projectile, respectively. $I$ is the spin of the target at ground state, and $J$ and $\Pi$ correspond to the spin and parity of the compound system. The $\alpha$ and $\alpha'$ are identifiers for the system before and after the reaction, respectively. Furthermore, $D_{\text{comp}}$ stands for the depletion factor including a pre-equilibrium cross section, $\lambda$ is the relative motion wave length, $W$ the width fluctuation correction factor and $T(E_{\alpha})$ the transmission coefficient for the projectile energy $E_{\alpha}$. The term $\delta_\pi(\alpha) = 1$ if $(-1)^l \pi_0 \Pi_0 = \Pi$ and 0 otherwise, where $\pi_0$ is the projectile’s parity and $\Pi_0$ the parity of the target at ground state.

### 2.3.2 Interaction Rate

All energy and outcome calculations of course assume a photodisintegration has taken place. To get a value for how many favourable reactions would be happening, and how likely a given nucleus will just fly through the photon field unchanged, we find the interaction rate. It is

$$\Gamma = \frac{c}{2} \int d\varepsilon \int_{-1}^{1} d\cos \theta \ (1 - \beta N \cos \theta) \ n_\gamma(\varepsilon_\gamma) \ \sigma_{\text{int}}(\varepsilon') \ .$$

(8)

where $n_\gamma(\varepsilon_\gamma)$ describes the differential photon density, and $\sigma_{\text{int}}(\varepsilon')$ is the interaction cross section for given collision energies $\varepsilon'$. The formula is given, e.g., in Boncioli et al. [20], where they are exploring nuclear physics in UHECR scenarios. In their original writings, they worked with the limit of $\beta N \approx 1$, which here was added explicitly to the appropriate location in the formula for more accuracy.

The interaction rate describes how many interactions of interest a nucleus has in a chosen range of photon energy $\varepsilon_\gamma$ and relative impact angle $\theta$ per
unit time. Here we choose a discrete distribution of photon energies, so we use a Dirac delta distribution together with a normalisation depending on the accretion disk’s photon field

\[ n_\gamma(\varepsilon_\gamma) = n_0 \delta(\varepsilon_\gamma - \varepsilon_0) \]  

(9)

for a photon energy of \( \varepsilon_0 \) and its spectral density \( n_0 \). Through this, the integral of the photon energy can be reduced to just the constant factor \( n_0 \), while all \( \varepsilon_\gamma \) will become \( \varepsilon_0 \). Together with Equation (2) the second integral can be substituted, so that Equation (8) stands as

\[ \Gamma = -c n_0 \frac{m_\gamma}{8 E_\gamma^2 \varepsilon_0^2 \beta_N} \int_{s-1}^{s+1} ds \left( s - m_N^2 \right) \sigma_{\text{int}}(s) . \]  

(10)

with \( s_{\pm 1} \) indicating the substituted limits of \( s(\cos \theta = \pm 1) \).

As we are focusing on the collision energy, we translate the interaction rate integral with this combined form of Equations (2) and (3), so that \( s \) can be seen as

\[ s = m_N^2 + 2 m_N \varepsilon' \]  

(11)

and the new integration limits will then be calculated through simple evaluation of Equation (3) at the original limits of \( \cos \theta \). This gives us the final form of the interaction rate that will be used in our calculations

\[ \Gamma = c n_0 \frac{m_N^2}{2 E_N \varepsilon_0^2 \beta_N} \int_{s_0^{\pm}}^{s_+^{\pm}} d\varepsilon' \varepsilon' \sigma_{\text{int}}(\varepsilon') . \]  

(12)

### 2.4 Nuclear Decay

The photodisintegration of a nucleus may create unstable nuclei. If those were to decay, it would mean different nuclei are to be handled for all following steps, so this possibility cannot be neglected. Further decays following one first nuclear decay are not uncommon, they must also be considered.

Nuclear decays of unstable isotopes are described by their decay rate \( \lambda \) and, if there is more than one decay path possible for a nucleus, their respective branching ratio \( b \). The number \( N \) of remaining nuclei of an isotope \( i \) after a time \( t \), starting with \( N_{i,0} = N_i(t = 0) \), is described by

\[ N_i(t) = N_{i,0} e^{-\lambda_i t} \]  

(13)

which stems from the long known Bateman equation for the change in number of a decaying isotope over time, so
\[ \frac{dN_i(t)}{dt} = -\lambda_i N_i(t) \]  

The result of this decay will be another isotope \( i+1 \), and for that the Bateman equation is

\[ \frac{dN_{i+1}(t)}{dt} = -\lambda_{i+1} N_{i+1}(t) + b_{i,i+1} \lambda_i N_i(t) \]

with the possibility of it decaying again itself at a rate \( \lambda_{i+1} \), but receiving additional new nuclei stemming from its parent isotope \( i \) with a ratio of \( b_{i,i+1} \), which will be 1 in case there are no other decay possibilities for isotope \( i \).

These equations will only hold for small time steps, as the interval written as \( dt \) suggests.

It is not unusual, that unstable isotopes decay into other again unstable ones. If these decay again, it is called a decay chain, and depending on the decay time scales, the simulations will need to take such events into account for each nucleus that might decay into other unstable nuclei.

To deal with this in a general and dynamic way, the work of L.J. Harr [21] for the US Air Force was used in this work, which gives a liable solution to this rather complex issue. It describes the changing decay chains of multiple persisting, unstable or stable nuclei through a matrix multiplication. This method also allows for more than one parent for each resulting nucleus. An example is

\[ \frac{\dot{\vec{N}}(t)}{dt} = \Lambda \vec{N}(t) \]

which describes a decay chain process like the one depicted in Figure 9 through a combination of multiple variations of Equation 15. All unstable nuclei are considered at rest in this calculation.
Figure 9: Example of a decay chain of six unstable and stable atomic nuclei, for which the decay chain matrix is shown in equation (16). Adapted from a figure of [21]

In the scenario simulated in this work, the nuclei are of relativistic energies. This has an effect on time (time dilation) and the particles' energies.

We treat this by first transforming the parent nucleus into its own rest frame with a Lorentz transformation of the particle's energy and time steps. Then the decay calculations follow, and the resulting decayed daughter particles were transformed back through a Lorentz transformation, using the original Lorentz factor $\gamma_0$ of the parent. The total energy $E_i$ for each nucleus $i$ is calculated from their mass energy $m_i$ as

$$E_i = \gamma_0 m_i.$$  \hfill (17)

This approximation works well for beta decay, as the masses of the parent and daughter nucleus are much higher than the electron or positron mass, but for heavier particle emissions, like the alpha decay, it gets less accurate the bigger the emitted particles become compared to the daughter nucleus. However, looking at the distribution of main decay modes in Figure 10, beta decay is much more prominent in this project's ranges.

2.4.1 Energy Distribution

In general, when an unstable nucleus decays, the total energy of the original parent will be distributed onto all resulting parts according to the kinematics of the interaction. Again, each resulting daughter nucleus of a decay with multiple possible energies can then decay into the next level of daughters, all with their own energy distributions, multiplied by the ones their parent had, and so on. The growth in size and therefore processing time for these calculations is immense if approximations like the one described above can not be used.
Although, as mentioned above, this more accurate calculation of energies will not be used by the program written in this thesis, but the theory has been worked through as shown in the following.

Depending on the mode of decay, the method of calculating the distribution of energy changes. The two blanketing main decay modes that are the only ones of high enough probability in the framework of these simulations are the beta decay and the particle emission, better known as alpha decay in the case of the emitted particle being an alpha particle. Figure 10 shows the range of nuclei considered in this project and their main decay modes, to give a feeling of the distribution of the modes worked with.

The particle emission is a two body problem, meaning the distribution of the energy can directly be calculated through momentum and energy conservation and both parts will get energy appropriate to their relative mass. These calculations are again considered to be relativistic, with two resulting decay parts. Each part’s energy will first be calculated in the parent’s rest frame. It uses with the relativistic energy-momentum relation

\[ E^2 = p^2 + m^2 \]  

for total energy \( E \), momentum energy \( p \) and mass energy \( m \), and the fact that the relation of the decay parts’ momentum energies \( p_1 \) and \( p_2 \) is
\[ p_1^2 = p_2^2 \]  
\[ (19) \]
due to momentum conservation. This equation together with the above stands as
\[ E_1^2 - m_1^2 = E_2^2 - m_2^2 \]  
\[ (20) \]
for respective decay part total energies \( E_1 \) and \( E_2 \) and their mass energies \( m_1 \) and \( m_2 \). Energy conservation states that
\[ E_0 = E_1 + E_2 \]  
\[ (21) \]
for the parent’s total energy \( E_0 \). By replacing one decay part energies at a time, the two equations
\[ E_1 = \frac{E_0^2 + m_1^2 - m_2^2}{2E_0} \]  
\[ (22) \]
\[ E_2 = \frac{E_0^2 + m_2^2 - m_1^2}{2E_0} \]  
\[ (23) \]
are formed. As in this case of the parent’s rest frame the total parent energy equals the parent’s mass energy, the resulting rest frame energies for each decay part are
\[ E_{1,\text{rest}} = \frac{m_0^2 + m_1^2 - m_2^2}{2m_0} \]  
\[ (24) \]
\[ E_{2,\text{rest}} = \frac{m_0^2 + m_2^2 - m_1^2}{2m_0} . \]  
\[ (25) \]

The system needs to be transformed back into the observer frame to take into account the kinetic energy via Lorentz transformation. The parent’s Lorentz factor \( \gamma_0 \) is calculated from the parent’s total energy \( E_0 \) and its mass energy \( m_0 \) as
\[ \gamma_0 = \frac{E_0}{m_0} \]  
\[ (26) \]
and the total energies of both particle emission parts therefore result in
\[ E_1 = \gamma_0 \ E_{1,\text{rest}} = E_0 \frac{m_0^2 + m_1^2 - m_2^2}{2m_0^2} \]  
\[ (27) \]
\[ E_2 = \gamma_0 \ E_{2,\text{rest}} = E_0 \frac{m_0^2 + m_2^2 - m_1^2}{2m_0^2} . \]  
\[ (28) \]
For a beta decay, there are three resulting parts: the daughter nucleus, an electron or positron, and an anti-neutrino or neutrino. The new atomic nucleus can be considered to have the same Lorentz factor as the parent nucleus according to the nuclear physics book by Krane [23], due to its large mass compared to the other two parts. The resulting $e/e^+$ and $\nu/\bar{\nu}$ can then take on complementing energies across a spectrum for each such decay. It depends on the remaining energy $E_{\text{rem}}$ in the system after the daughter nucleus' energy has been determined with its mass $m_d$, so that

$$E_{\text{rem}} = E_{\text{parent}} + Q - \gamma_{\text{parent}} m_d - m_{e/e^+}$$

with the decay energy value $Q$, describes the maximum energy that the electron or positron together with its neutrino can reach. The kinematic energy $T_e$ distribution of the electrons or positrons, for now both referred to as just $e$ for simplicity, can be determined via

$$N(T_e) = \sqrt{T_e^2 + 2T_e m_e (T_e + m_e)(Q - T_e)(Q - T_e)^2 - m_d^2} F(T_e)$$

from [24], where $F(T_e)$ represents the Fermi function. The $\nu$ will then take the corresponding remaining kinetic energy of the system.

### 2.5 Energy Ranges and Cutoffs

For all the advancements science has made over the decades, it is not feasibly possible to simulate a problem perfectly under consideration of absolutely everything that is happening in the universe. We can, however, determine which processes we know of are insignificant on a level that they are virtually irrelevant to the results we are looking for. And even if they are not disregarded, most models can only do so much to approximate real problems.

In reality, many things would for example often be able to take on an infinite number of energies on a continuous spectrum. Any viable process cannot be expected to distinguish between infinite options, it has to use chosen discrete step sizes to come to a result in the end, and oftentimes set upper or lower limits.

The large number of possibilities here lead to very long processing times. To reduce this as much as possible without skewing the results of the overall simulations a lot, probability cutoffs of similar works like [1] were compared, leading us to the same or farther reaching branching ratio cutoff for photodisintegration results of $< 10^{-4}$. To keep scaling throughout the project comparable, the same cutoff of $< 10^{-4}$ was also chosen for the individual
2.5 ENERGY RANGES AND CUTOFFS V. Lanzinger

decay channel probabilities. A limit of $6.5 \times 10^{11}$ s was chosen for half-life times for the nuclei to be seen here as stable. Half-lives were chosen too short to be considered below $4 \times 10^{-4}$ s. Figure 11 shows an area in which the photodisintegration results lie and their respective half-life ranges.

![Figure 11: Table of nuclides in the range of $N \leq 26$ and $A \leq 56$ with colours indicating respective half-life ranges. From [22]](image)

One unavoidable cutoff for simulations is the precision of a computer and the programming language that is utilised, but usually they are not a direct problem and chosen cutoffs are set for less precision than modern systems will give by default. Other approximations are rooted in the various tools used, which have to be worked with. These will be discussed further for the tools used here in the next section.

When there are infinite possibilities of energies, boundaries and discrete values have to be defined. The step sizes were chosen equidistant on a logarithmic scale, and to have one system where all resulting energies could also be sorted in, energy bins of similar sizes were set around each discrete value. The boundaries of the considered energy range were chosen to cover typical multi-wavelength spectral energy distributions of AGNs.

The considered energy ranges for the accelerated nuclei are values between $10^{-3}$ GeV and $10^{12}$ GeV with logarithmic step sizes of 0.05, giving 300 bins. For the accretion disk photons the range lies between $10^{-19}$ GeV and $10^{-3}$ GeV with logarithmic steps of 0.1, making 160 bins.
3 Implementation

Gathering information is only half of the work, the gained knowledge then needs to be processed to gain physics results. All pieces need to fit together in the end, so limitations and scales must be known and fixed in correspondence with the foregoing and following tasks where they are necessary. With the information of what data is needed and where it can be found, the next step is to gather everything systematically, and in the case of a large amount of info needed like in this project everything should be as automated as possible. The data that is gathered would typically be used by other parts or steps of the simulation, so translating, storing and then again reading in selected values is a big part of communication between processes.

3.1 TALYS Calculations

A large part of this project in terms of duration was spent on getting all data on the photodisintegrations. A comprehensive list of input nuclei for a collection of energies is needed, with all possible resulting secondary nuclei together with their respective production probabilities, each including their own energy distributions, and the overall production cross section for the photodisintegration of the original nucleus. These are all gathered from TALYS calculations.

The first steps included getting to know TALYS and understanding its output format, which is raw text only. Even after reading and deciphering most output lines, not all available information can be found through simply trying. The large manual the TALYS team provides needed to be studied to understand the possibilities and parameters, some dependant on one another, and to find what could be calculated and what would bring the desired quantities. After some testing the suitable TALYS parameters to get our results were set. The first lists of residual isotope nuclei were mostly around or above one hundred secondary parts, depending on the collision energy.

As mentioned in Section 2.5, the nucleus and photon energies were chosen on a preset grid of energies. TALYS itself expects energy values in the rest frame of the nucleus. The input collision energies are calculated using Equation (3) under the assumption of isotropic particle and photon distribution. TALYS treats photodisintegration from the interaction threshold up to a collision energy of \( \varepsilon' = 1 \text{ GeV} \). This upper limit greatly impacts the amount of data that can be worked with, also in later calculations, and will lead to some higher energy bins being hardly filled with particles.
3.1.1 Isotope Cascades

To get a comprehensive list of possible outcomes, the idea was to start with one isotope and repeatedly add all those that are disintegration results, and that have not yet been the input nucleus themselves before, and start TALYS again with each new nucleus as input. Figure 12 shows a visualisation of the general workflow practised when repeatedly calculating rounds of photodisintegration data and gathering additional information for those results, namely physical properties of all seen isotope cores which are needed for further calculations.

The isotope that was chosen to start the TALYS runs with was iron 56, thereby the heaviest isotope considered in these simulations. It was chosen as it is the heaviest stable nucleus produced in stars’ nuclear fusion and will disintegrate via photon interaction in an endothermic reaction, as will all nuclei smaller than iron. When there were no new resulting residual isotopes,
that had no TALYS run of their own yet, there were no further runs started. Otherwise a new input file and folder structure was created for each missing nucleus and started in bulk. The data filtering and storage file creation will be discussed below.

TALYS does not have mass data on some very rare isotopes, nor do the isotope property sources have information on them. Fortunately, these results had branching ratios below the cutoff of $10^{-4}$ (see Section 2.5) and did not have to be considered specifically.

### 3.1.2 Disintegration Data Filtering

TALYS output files are one long list of text. Parameters can be set for more output files, but these are just in addition to the values also being in the total output file. The method that was chosen was to walk each file line by line and look for certain unique triggers to enter a specific sequence in reading the lines after the trigger, until the next trigger was found. The filtered values were stored into variables through a unique key combination, kept similar to the chosen data file structure for better overview.

For each TALYS output file, a separate file was appended with all possible isotope results, not including duplicates. After the whole filtering process for a new set was complete, an additional file was created containing all isotope names that had their TALYS data filtered successfully. Together with the file for invalid isotopes that were started but failed to run, all missing residuals that needed to be set up for a future TALYS run could be determined. The residual file contained a complete list of all isotope names that were already seen, including the ones that failed for TALYS inputs, as that list would also be used to gather the physical properties that would be needed for further calculations, and that includes all outcomes for each valid run.

### 3.1.3 Data File Storage

The data format used to store calculated data and needed values in this project is HDF5 [25]. It can go into multiple dimensions of data structures and features attributes for groups and data sets while still keeping low load times and manageable file sizes, which makes it a great choice for this work. It is also supported by all widely used higher programming languages, especially those often found in data processing.

One whole TALYS output file would be read entirely and the filtered data stored into variables before all data for one isotope nucleus was written to a HDF5 file. Each nucleus has its own data file named after its charge and mass number. A visual representation of the structure of the nuclei’s HDF5 files
is shown in Figure 13, generalising the names and identifier for the multiple possibilities. This generalisation includes the symbol for secondary particles, like protons, neutrons, photons, electrons, etc., which was defined for them to be of a similar structure to the isotope identifiers but still distinguishable. The transport code which will use this data uses code numbers for each particle, so the same codes were used here. To keep the naming scheme similar, a 0 was added instead of the isotope nuclei’s charge numbers and also connected via a minus (−) to the code number. For example, the code number for a neutron is 14, so the corresponding symbol used here is 0-14. An example file with values can be seen in Section 4.

Figure 13: Visualisation of the structure of each nucleus’ TALYS output data HDF5 file, including units. \( Z \) and \( A \) represent the corresponding nucleus’ charge and mass number, respectively, each having an own file with its name in the form of \( Z-A \). All attributes are string values in HDF5. The first set of subgroups is named after the value of \( \varepsilon' \) of the TALYS calculation, containing one subgroup for every residual isotope, again identified by \( Z \) and \( A \), and one for every secondary particle, called by its defined symbol (see main text). Each of these subgroups has a set of attributes and secondary isotope nuclei in addition have data sets for their energy distribution with corresponding cross sections.
This naming of isotope nuclei and their files was chosen to be distinct, clear and iterable – meaning there is no question if the name is in uppercase letters, if there is a space between letters and numbers, if the mass number is first or second. For this, TALYS entries had to be translated, and vice versa. The formatting of TALYS’ isotope names follows the form of ‘56Fe’, starting with the mass number, followed directly by the name found on the periodic table, with the first letter uppercase and the second lowercase. In the case of a one letter name, for example ‘52V’, the space was added after, so was stripped automatically by the filtering anyway. If the mass number was below ten, like ‘9Be’, a space was added in front, as opposed to a zero, also making no problems for filtering.

Isotope names were translated through a name to number mapping, where all elements below iron were entered and paired with their corresponding charge number. Deuteron, triton, helium-3 cores and alpha particles do not fall under the symbol naming scheme and were added correctly to the list manually.

The procedure of creating the data files runs through all collected data and automatically used the same keys as the variables had used as identifiers for each defined subgroup. Attributes, data sets and further subgroups to a group were stored into separate variables. Using the Python package h5py the HDF5 files were written, and also read back out when using the data again in other processes.

In total 355 HDF5 files created from valid TALYS runs, each with about 40 to 60 different collision energies from the aforementioned bins, depending on the lowest possible threshold energy for the specific nucleus. The resulting possibilities varied strongly between one and above two hundred, changing with input nucleus and energy.

3.2 Isotope and Particle Data

When creating the TALYS nucleus collection featuring every possible residual, they were all added to a comprehensive list of observed isotopes. The ones that TALYS could not run anymore because it had no mass information for them were also added, because they could be needed for other calculations nonetheless.

The data on physical and chemical properties for each residual isotope and secondary particle was collected using Wolfram Mathematica [26] and its own programming language. Their database for isotope properties is gathered from multiple official sources [27] like NIST. A similar one exists for particles, namely leptons, mesons, baryons and bosons, for which they also use multiple databases and publications [28] as sources. The data for
the considered nuclei and particles is collected, uniformly structured and stored in HDF5 files.

Starting from the text list of isotopes, it was translated to a Mathematica readable input to loop over using a Python script, as direct communication between the two was not possible and not needed, as it was only text to be copied once an updated list of isotope data was needed. The charge numbers were translated back to element names using the same map as for TALYS outputs, as those are used as input for the object of the data collection.

The full collection of element names and their included mass numbers was run through in Mathematica, where it was prepared to give a custom formatted output. Every property needed was collected or calculated from others that were available, like for instance the threshold energy for photodisintegration. Units can be changed very well in Mathematica, as all physical constants are available for use from official sources, so going from masses in atomic units to GeV was done there.

Formatting was done so that everything was output as a text that could be interpreted as a dictionary of custom nucleus objects in Python. Number precision, ordering, what would be a string or a float, what should be connected in a list and separating commas all needed to be adjusted. Some values were unavailable, some where combined in special ways, and some were positive or negative infinite. All special case entries that would not have been valid for the generalised Python transcript were caught and handled without missing any. This was a necessary step to ensure fully automated runs for the over 300 different nuclei, in case any needed parameter should change or more conditions were added.

Similar to the isotopes, properties for the secondary particles were also needed. The hydrogen and helium isotope cores were added to the list of nuclei, while single particles like protons, neutrons, positrons and muons were listed on their own. As they do not have such a universal naming scheme as the periodic table – or rather many universally recognisable symbols use superscript, subscript, or are represented by Greek letters – their identifiers had to be found to list them all. Once a list was complete, those names were not changing, as the secondary particles TALYS considers stay the same.

These few names could manually be listed in Mathematica, and given to their own loop of property selection. Again, units were switched to the desired ones and missing values were calculated if possible, afterwards everything was transformed into a text similar to the isotope nuclei for read-in into another custom Python object.
3.2 ISOTOPE AND PARTICLE DATA

3.2.1 Data File Structures

To store these isotope and particle properties into files, the same approach as with the TALYS data files was chosen, which was to translate everything into Python variables, which was done in this case in the form of Python interpreter readable text representing objects and their attributes, and systematically write the data to HDF5 files.

Because of the difference in data structure it was chosen to write two different files, one for the isotope properties and one for those of the other particles. Figure 14 represents the form and names of stored data inside both files. All attributes are again strings, and all data sets are lists of zero or more values, either strings or floats.

The isotopes’ data sets are ordered lists in case there is more than one decay mode, and each value corresponds to the paired data sets’ values of the same index. Lifetimes of infinity are represented by $-1$, unknown decay mode probabilities were also represented by $-1$, while unknown decay energies were set as 0 because the values would be used in further calculations. This way all diverse and unclear options were caught by general condition checks when filtering for data of interest or simply follow the equations through without falsifying them.

For the particles’ decays, it was not so easy. Each decay possibility lists more than one decay daughter, because it lists all particles, not just atomic nuclei like the isotope data does. This was handled by creating single subgroups for each decay, which were named simply by numbers in sequence, with an attribute for the probability of the specific decay mode and a data set that is a list with all resulting particles of that one decay. Stable particles would have no subgroups.

All mentions of other isotopes or particles in the data of another will be the same names as the ones used for their own entry or file. Every identifier can be used to call another property or result without transforming the key further. Because of this all transforms and code mappings were done for all sources and entries beforehand, so that there can be no confusion and any additional workload is reduced for continuous calls from the stored data.
3.3 Interaction Rate Calculations

The calculation of the interaction rate presented in Section 2.3.2 was not simple to implement. The first apparent problem was that only discrete values for the interaction cross section were available for the calculation of
Equation (12), which includes an integral. This problem was solvable by applying a quadrature for numerical integration. The trapezoidal rule was chosen, as it gives enough accuracy without demanding too much run time. For each given point included in the integral, a halfway point between it and its left neighbour was determined and the area for a rectangle with the point’s horizontal distance as width. All single areas are summed up to give the numerically approximated value of the integral.

As Equation (12) shows, the variables needed in the integral are the interaction cross sections corresponding to the photon energy and the total nucleus energy, as well as other values tied to the latter. The two energies combined make up one collision energy, as described in Equation (3).

Because the interaction rate should be available for specified nucleus energies, the calculations will be run for each of those energies inside the chosen range, combined with a fixed photon energy, one at a time. This poses the next problem of starting from two separate values, and having the ones dependent of both with only a discrete list of values available.

This was especially noticeable because of the integration limits. For each combination of nucleus and photon energies, the physical limits were calculated according to Equation (12). If the threshold energy for the given nucleus, that was discussed in Section 2.3, was higher than the lower integration limit, then it would be the new lower limit. If the upper limit was above 1000 MeV, that combination of energies would unfortunately have to be dismissed, as there are no values above that collision energy because TALYS cannot provide them. This means that for many higher energies no interaction rates can be found for this simulation at the moment.

With both limits determined, the next step was to find only those values inside the list collected from TALYS that would fall between those integration limits. The trapezoidal rule described above would then lead to area segments $A_i$ calculated for each index $i$ determined valid as

$$A_i = (\varepsilon'_i - \varepsilon'_{i-1}) \left( \frac{\varepsilon'_i \sigma_{\text{int},i} + \varepsilon'_{i-1} \sigma_{\text{int},i-1}}{2} \right)$$

which would all be summed up to give the interaction rate for one combination of nucleus energy and photon energy. The first included point does thereby not have an area segment, calculations start with the second point.

In this manner, interaction rates are calculated for all considered nuclei in any available energy combination, and all data is again saved to HDF5 files. The data structure inside the files is simpler for these than for the TALYS data files, as there are just two data sets for each photon energy, again put into a separate file for each isotope nucleus. A graphical representation of the files’ structure can be seen in Figure 15.
Figure 15: Visualisation of the HDF5 data files for the interaction rates. Every considered isotope has a separate file, again named after the identifier with \( Z \) and \( A \) used above, with an added “ir” to not confuse these files with the TALYS data files. Data is stored in the format of a list of strings that are valid floats.

Trying to create a collection of interaction rate values for each nucleus that has a TALYS data file, showed that some few isotopes had been available for calculations in TALYS, but there was insufficient data from Wolfram, so they had no entry in the isotope data file. These cases – ten in number, out of 355 – do not have an interaction rate file.

It was a tricky problem to find a good approach for setting variable limits from combinations of chosen values, while having value lists dependant of one another, but also of the chosen values, and still having all play together as limits and factors to an integral. Also, before realising that some values simply were out of bounds and were not possible to be calculated correctly with the given data, the results had made no sense. This made finding the actual errors and their causes much more difficult, and led to some wrong turns and trying to fix approaches where there was nothing to fix. Only when faulty values were left out altogether things started to become traceable and the interaction rate values made progress to being calculated correctly.

### 3.4 Decay Chain Time Evolution

The problem of variable possibilities of nuclear decay needs a great amount of automation in the most general manner. Every value needs to be gathered via unique identifiers, and all decay channels, daughters, numbers, time steps, repetitions, numbers of parts, energy distributions, relativistic adjustments, and all secondary decay particles have to be dynamic, because most of them
would change if only one other was set differently.

The first step was creating a program structure, that could create something similar to the time evolution matrix described in Section 2.4, but would be dynamic in size and yet strict to keep rows and columns clear each repetition. The chosen approach collects the nuclei’s decay daughters in order, starting from one original parent nucleus and walking through all possible decay chain paths.

During this collecting, each occurring part gets an incrementing ID number, which then functioned as their respective row and column number, the first parent starting from zero. The highest ID thereby automatically gives the size of the square evolution matrix to be created. Each given part in the decay chain gets an entry other than zero for decaying themselves or gaining numbers from other parts’ decays on their respective ID’s column in the evolution matrix.

The required information that was gathered together with the decay daughters for each part was the respective parent, the corresponding branching ratio, and the own decay rate, which was collected from the earlier described isotope data stored in an HDF5 file. At this stage, there was no kinetic energy and no relativistic factor added, to make it easier to check for correct results first and add more complex themes later, when the core elements were already in place and working.

After the matrix structure for time evolution had been created, the next essential part was calculating the resulting number of particles for each given time, like the $N_i(t)$ vector from Equation (16). This process takes each entry of the matrix and multiplies it by the time step to get the change in number for the given point. Each point needs to be attributed to the right decay part, and the change in number added or subtracted from the number it had been the time step before. The approach of keeping a matrix form for the numbers $\dot{N}_i(t)$ of the same structure as the evolution matrix was chosen, to keep all numbers clear. For the overall number of particles at a given time, each row $i$ summed up would then give the $N_i(t)$ of the decay chain part with the ID $i$.

The first number matrix for the starting time was created as all zeros, except the original nucleus, for which the number could be chosen. All following time steps were then stored together in order, to keep an overview of each step. After the decay chain process, all nuclei’s total energies were calculated via simple Lorentz transformation from the nuclei’s rest frame to the observer frame as mentioned in Section 2.4.
3.5 Software Architecture

In this section the program structures of the processes discussed above will be described in more detail. All work and saved data was regularly pushed onto a private GitLab repository, in which all scripts and modules were collected, but they would be called separately. Figure 16 shows an overview for everything contained in this combined working directory repository.

![Figure 16: Schematic overview of the working repository structure. Cuboids represent directories, with the two top ones being Python modules, while the rectangle stands for a file. Yellow colour indicates import function files, white runnable parts, blue input and processing data and green stands for processed output data. The text blocks on the side each describe the contents of the part to their left in a simplistic manner.](image-url)

Processes for calculating nuclei decay chains and their energies
Input: starting nucleus and its total energy, desired time frame and step sizes
Output: list of possible outcomes with probabilities and energies

Processes for calculating photon interaction rate values for nuclei with TALYS data
Input: specific isotope nucleus, possible constant value for photon spectral density
Output: IR values listed or plotted, or full collection of HDF5 files

Collection of scripts for processing TALYS data informations and preparing follow-up, also contains resources for isotope and secondary particle data as well as automated TALYS input creations and additional information files

Collection of all created and stored data files in HDF5 format

Collection of TALYS’ direct output files for each isotope run through TALYS, and utility scripts for organising and processing the data

Process for filtering all needed data from all available TALYS output files and creating HDF5 files for each, also creates full lists of residuals and faulty runs
The largest individual process is the decay chain calculations. The file structure of this module is shown in Figure 17 together with some description of the contents of each file. Everything specifically needed to create the data for the decay chain time evolution that was discussed in Sections 2.4 and 3.4 is contained in this module folder. Resources used from outside this structure were only the HDF5 files of the isotope and particle parameters.

The second individual program block is the one for calculating the interaction rates as discussed in Section 3.3. The folder structure for it is shown in Figure 18. It is smaller than the decay chain module in terms of functions but features different output possibilities. For overview purposes callable files for printing the used input values from TALYS data files and the calculated interaction rate values were added. Visualisation plots can also be created for all considered energies of one chosen nucleus at a time.
Figure 18: Structure of the interaction-rates Python module from the working repository. Yellow colour indicates import function files, white runnable files, and green stands for processed output data. The text blocks each describe the contents of the part to their left in a simplistic manner.

The most important final implementation for this module was to create a complete set of interaction rate HDF5 files for all available TALYS data files. This only needed to be called once after all requirements for the data were complete to create the data files for storage. It would only have to be run again if something changed in the source data or for the output structure. One whole new run of all 345 nuclei available for this took two full days, while calling HDF5 files would be a lot faster, which shows how useful a pre-calculation and storage of this data is worth in run time.
The HDF5 files created for data storage after such processing runs and saved to be called separately whenever needed were placed in their own collection directory. The structure shown in Figure 19 was chosen such that all processed data needed by other programs can be found in one place and the contents looked for can be discerned by the path to each file. The collective size of these data files is 1.6 GB, and they are structured as the diagrams in Figures 13, 14 and 15 show.

Figure 19: Structure of the hdf5-files directory from the working repository. Green colour stands for processed output data. The text blocks each describe the contents of the part to their left in a simplistic manner.
Another structure depicted is the collection of scripts and information shown in Figure 20. Most of its Python scripts’ purpose is clear from the names and descriptions. Similar to the files creating the TALYS data files and the interaction rate ones, the two remaining categories of HDF5 files are
created by scripts and raw data stored inside this folder. Not all files inside the resources directory were listed separately, but rather grouped thematically, as their number would make the overview too big.

TALYS input files are text files for the input energy list and a TALYS input file template and bash scripts for starting them sequentially and in parallel on the remote servers. The TALYS output information files are a collection of listings of done runs, failed runs, missing runs and all occurring nuclei. These files are used by most of the runnable scripts in this collection directory.

The Wolfram Mathematica (WM) data collection files are text formatted for Mathematica and also output from Mathematica formatted for Python. There is no direct communication between those two programming languages here, so their few interchanges are handled like this. The WM notebook files themselves are also stored here for integrity, even though they cannot be run outside of Mathematica.

The last part of the repository is shown in Figure 21 and deals with the raw TALYS output. Finished TALYS runs are collected from the remote server and organised into a clean format by the appropriately named script.

![Figure 21: Structure of the talys-files directory and its associated file from the working repository. White colour indicates runnable files and blue stand for input and processing data. The text blocks each describe the contents of the part to their left in a simplistic manner.](image-url)
Updated complete name lists are created and the script to process new TALYS output data into their respective HDF5 data files also creates process information files, like the complete list of all occurring residual nuclei. Each script found in the whole program structure that works with these lists can then be run also to update all other resources produces by them.

These resources were mainly needed during the process of collecting and handling everything from the different sources and over the duration of the calculations and other developments. After all calculations were complete and all parts had updated all information, ideally only the HDF5 data files will be needed, and the routine for calculating the decay chains.

During working on this project, everything was collected into one repository to handle everything together in one version control program, namely Git. It helped with making clean and structured saving points, working dynamically from different locations and on the go, and would also always mean safe storage of all work, even past versions, without the possibility of overwriting the wrong files, which happens more easily with simple manual-upload-based saving methods.
4 Results

The process from starting input to results from this project will be played through in the following. Isotopes of nuclei that are observed regularly in cosmic rays were chosen here as examples, namely nitrogen 18 (18N), silicon 24 (24Si) and iron 56 (56Fe).

The first process is calculating interaction rates, which starts with running TALYS for each nucleus as a target with the whole list of collision energies. The results are then filtered for the interaction cross sections of each energy, which can be seen in Figures 22, 23 and 24. The graphs look similar to the one in Figure 8, as expected, only the higher energy values in the photopion production range are cut off due to TALYS’ upper energy limit. These cross sections are saved into HDF5 files, among others.

The interaction rates are calculated from these cross sections, as explained in Section 2.3.2. For one fixed photon energy, in this case $10^{-8}$ GeV, the nucleus’ energy is varied to reach different collision energies and calculate the corresponding interaction rates, as shown in Figures 25, 26 and 27. The interaction rate values are calculated for all considered energy combinations and stored into HDF5 files of the structure described in Section 3.3, a part of the file for nitrogen 18 is shown in Figure 28.

The spectral density $n_0$ from Equation (12) still needs to be multiplied in for the photon field the nucleus must cross. With typical values for an AGN’s luminosity $L = 10^{44}$ erg/s and a chosen example photon energy of $10^{-8}$ GeV, at a distance of 0.1 pc from the accretion disk the spectral density is about $1.7 \times 10^8$ cm$^{-3}$. The optical depth $\tau_{\text{photodis}}$ for photodisintegration of a specific region can also be calculated from $n_0$ and the interaction rate $\Gamma$. Some example values for a spherical volume with a radius of $10^{16}$ cm are listed in Table 1. An optical depths of 1 means it is very likely for interaction to take place. All the listed values for $\tau_{\text{photodis}}$ are quite low, which means there is a low chance of photodisintegration happening at these values. The lower the photon energy gets, the higher $n_0$ becomes, but in general photodisintegration then also needs higher nucleus energies. One of the lowest photon energies with which interaction rates for iron 56 were calculated is $10^{-12.1}$ GeV. In this example that energy gives $n_0 = 2.2 \times 10^{12}$ cm$^{-3}$, and an iron 56 nucleus with a total energy of $10^{12}$ GeV then sees an optical depth of $\tau_{\text{photodis}} = 731$. It is very likely that photodisintegration occurs here and the nucleus does not survive.
Figure 22: Points gathered from TALYS calculations for nitrogen 18 of interaction cross sections for different collision energies.

Figure 23: Points gathered from TALYS calculations for silicon 24 of interaction cross sections for different collision energies.
Figure 24: Points gathered from TALYS calculations for iron 56 of interaction cross sections for different collision energies.

Figure 25: Interaction rates for a nitrogen 18 nucleus of different energies with a photon of energy $\varepsilon_0$. 

$Z = 26, A = 56$

$\varepsilon_0 = 10^{-8.00}\,\text{GeV}$
Figure 26: Interaction rates for a silicon 24 nucleus of different energies with a photon of energy $\varepsilon_0$.

Figure 27: Interaction rates for an iron 56 nucleus of different energies with a photon of energy $\varepsilon_0$. 
Table 1: Example optical depths for $n_0 = 1.7 \times 10^8$ cm$^{-3}$.

<table>
<thead>
<tr>
<th>nucleus</th>
<th>total energy (GeV)</th>
<th>$\Gamma$ (cm$^3$s$^{-1}$)</th>
<th>$\tau_{\text{photodis}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$^{56}\text{Fe}$</td>
<td>$10^{7.8}$</td>
<td>1.155e-15</td>
<td>6.684e-2</td>
</tr>
<tr>
<td>$^{56}\text{Fe}$</td>
<td>$10^{8.8}$</td>
<td>9.159e-17</td>
<td>5.300e-3</td>
</tr>
<tr>
<td>$^{24}\text{Si}$</td>
<td>$10^{7.8}$</td>
<td>2.529e-16</td>
<td>1.464e-2</td>
</tr>
<tr>
<td>$^{24}\text{Si}$</td>
<td>$10^{8.8}$</td>
<td>1.396e-17</td>
<td>8.079e-4</td>
</tr>
<tr>
<td>$^{18}\text{N}$</td>
<td>$10^{7.8}$</td>
<td>1.408e-16</td>
<td>8.148e-3</td>
</tr>
<tr>
<td>$^{18}\text{N}$</td>
<td>$10^{8.8}$</td>
<td>5.821e-18</td>
<td>3.369e-4</td>
</tr>
</tbody>
</table>

Figure 28: Data file for nitrogen 18 interaction rates (the file is named “7-18_ir.hdf5”). The highlighted entry is for $\varepsilon_0 = 10^{-8}$ GeV, and the lists of values on the right are the corresponding nucleus energies and interaction rates as shown in Figure 25.
More values than just the interaction cross sections are filtered from the TALYS output, most of all are the possible disintegration outcome nuclei and their energy distributions. All information for one target nucleus is stored into one HDF5 file as described in Section 3.1.3. An example part of such a file is shown in Figure 29. The data collected for the properties of each isotope is also stored in an HDF5 file, it is shown in Figure 30. Most of these properties are needed for the nuclear decay calculations.

Two of the three example nuclei chosen here are unstable. Figure 31 depicts the decay chains starting with nitrogen 18 and silicon 24. The time evolution of these decay chains via the matrix method described in Section 2.4 is run through, visualisations of the changes of the respective numbers over time are shown in Figures 32 and 33. Starting from an example of one nucleus with a total energy of 100 GeV, the distribution of nuclei and their approximated energies after 5.5 seconds are listed in Tables 2 and 3.

Figure 29: TALYS output data file for 18N (named “7-18.hdf5”). Highlighted is a value for the collision energy, which shows its attributes seen below. The open value lists on the right are energy distributions for one of the possible outcome nuclei for that collision.
Figure 30: Isotope properties file, highlighted is nitrogen 18 and the open value lists on the right show its possible decay daughters and their respective branching ratios.

Figure 31: A schematic diagram for the decay chains of silicon 24 and nitrogen 18. Decay possibilities for nuclei are depicted from left to right, with the respective branching ratios next to each arrow.
Figure 32: Visualisation of the decay chain time evolution of nitrogen 18. The diagram in Figure 31 shows information on the branching ratios.

Table 2: Results for the example 18N decay chain time evolution.

<table>
<thead>
<tr>
<th>nucleus</th>
<th>number / probability</th>
<th>total energy (GeV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>18N</td>
<td>1.86e-03</td>
<td>100</td>
</tr>
<tr>
<td>18O</td>
<td>7.69e-01</td>
<td>99.92</td>
</tr>
<tr>
<td>14C</td>
<td>1.22e-01</td>
<td>77.73</td>
</tr>
<tr>
<td>17O</td>
<td>1.09e-01</td>
<td>94.37</td>
</tr>
<tr>
<td>14N</td>
<td>2.16e-12</td>
<td>77.73</td>
</tr>
</tbody>
</table>
Figure 33: Visualisation of the decay chain time evolution of silicon 24. The diagram in Figure 31 shows information on the branching ratios.

Table 3: Results for the example 24Si decay chain time evolution.

<table>
<thead>
<tr>
<th>nucleus</th>
<th>number / probability</th>
<th>total energy (GeV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24Si</td>
<td>2.58e-14</td>
<td>100</td>
</tr>
<tr>
<td>24Al</td>
<td>1.02e-01</td>
<td>99.95</td>
</tr>
<tr>
<td>23Mg</td>
<td>2.72e-01</td>
<td>95.76</td>
</tr>
<tr>
<td>24Mg</td>
<td>5.18e-01</td>
<td>99.89</td>
</tr>
<tr>
<td>20Ne</td>
<td>1.81e-04</td>
<td>83.26</td>
</tr>
<tr>
<td>23Na</td>
<td>1.04e-01</td>
<td>95.74</td>
</tr>
</tbody>
</table>
5 Discussion

The results shown above were combined with typical values to put them into context. These calculations can be interpreted to see if it is even likely to have any heavy nuclei left after an AGN jet acceleration, or how they change into different ones, or if the effort to look for them becomes futile. The examples in the section above show that the combination of AGN properties and photon and nucleus energies can give very different survival probabilities for the same kind of nucleus.

This work is per se, as mentioned above, not a self-contained process. Much of the work was gathering data and figuring out ways to get all needed information, how to handle the steps in sequence for every new development, and how to translate physical formulas into generalised, dynamically scaled and fully automated processes, that would also handle special cases appropriately.

The routine for calculating decay chains is at the moment written in Python. This is not ideal, as most other parts are likely to be written in Fortran or C/C++, which cannot communicate easily with Python. The main structure and overall logic behind the calculations now exists, and should be adaptable for another programmer versed in one of the mentioned languages without much problem.

As the project stands now, only discrete energy values are used for photon and nucleus energies, and thereby for the collision energies. This leads to all other values inevitably also being finite lists. In the given structure this can only be made better by using finer steps between each value, especially if steep slopes and rapidly changing values can be observed. Using full functions instead of single points is not possible for the worked out structures. The numerical approximation of the integral is also unavoidable, but at that point it does not heavily worsen the results.

One hold back here is the used reaction code TALYS. Not only does it take discrete values for each round of calculations, and also only gives back a list of a set amount of discrete values, it has a hard upper limit of 1000 MeV. This limits mainly the interaction rates and calculations for heavier nuclei, but at the energy ranges chosen, every nucleus is affected by this. This limit is not something that can be changed by us or any other group working with TALYS, until they themselves raise it in a future version.

Of course the chosen cutoffs and ranges of energy that are necessary to be able to run all processes in a practical time frame are also a limiting factor for ideal results. Similar works have, as mentioned before, worked with similar or smaller ranges, which makes this work not meaningless, relatively speaking.
At some point, keeping every tiniest possibility will not change the intended result of the simulation immensely.

What is indeed missing from these calculations and data are error estimations. This is also something TALYS does not provide completely for all data, so there was no use in adding some errors for parts, when there were none for others. Future versions could add these features, which would make it possible for further simulations and calculations to add error estimation to the data.
References


REFERENCES


50
my sincerest thanks to Manuel for staying by my side
and keeping me sane throughout the years