

^{235}U の全核分裂生成物の 放射能の時間変化

2012年2月

福井大学 工学部 物理工学科

大沼 正季

森近 厚平

(学籍番号 08380112) (学籍番号 08380554)

目次

第 1 章	序論	4
第 2 章	原子核と原子核の崩壊様式	8
2.1	原子核	8
2.1.1	原子核の発見	8
2.1.2	原子核の構成要素	8
2.1.3	原子核の構造	9
2.2	原子核崩壊	10
2.2.1	放射性崩壊の機構	10
2.2.2	核分裂	11
2.2.3	α 崩壊	14
2.2.4	γ 崩壊	14
2.2.5	β 崩壊	14
第 3 章	原子核の崩壊を表す微分方程式	16
3.1	1 つの核を扱う場合	16
3.2	複数の核を扱う場合	16
3.2.1	定数係数 1 階常微分方程式の一般解について	17
3.2.2	行列が下三角行列の場合	18
第 4 章	計算方法の詳細	20
4.1	半減期と分岐比から λ_{ij} を決める方法	20
4.2	アイソマーの扱い	21
4.3	核の個数の初期値の設定	21
4.4	β^-n 崩壊の扱い	22
4.5	C_{ij} の決定方法	22
4.6	n_i の和のチェック	22
第 5 章	結果	24
5.1	質量数 83 の系列における β 崩壊による放射能の時間推移計算結果	25
5.2	^{235}U の核分裂生成物の β 崩壊による放射能の時間推移計算結果	26
5.3	傾きについての考察	27
5.4	時間に対する放射能が高い原子核	29
5.5	原子力安全・保安院が公表した大気中に放出された放射性物質	31
第 6 章	まとめ	33

参考文献	34
謝辞	35
付録 1 原子核の β 崩壊のデータ	36
付録 2 プログラムソースコード	72

執筆担当者

第1章 大沼

第2章 森近

第3章 大沼

第4章 大沼

第5章 森近

第6章 森近

付録

核データ表 ($79 \leq A \leq 109$) 森近

核データ表 ($126 \leq A \leq 154$) 大沼

プログラムソースコードリスト 大沼、森近

第1章 序論

2011年3月11日東北地方太平洋沖地震が発生した。日本では誰も経験した事のない大地震と大津波という自然現象により日本は大きな被害を受けた。震災後最大の問題となったのが東京電力福島第一原発事故であった。

地震によって原子炉が自動停止し、その後、非常用炉心冷却装置が作動したのだが、約1時間後に押し寄せた津波により原子力発電所の全電源が失われ、圧力容器内の燃料棒や使用済み燃料貯蔵プール内の核燃料の冷却が不能になった。核分裂が止まっても、放射線を出して他の核になる放射性崩壊は止まらないので崩壊熱が出つづけた。結果的にこの崩壊熱により燃料棒が溶融し、多量の放射性物質が放出された。原子力発電所から放出された放射性物質は、風によって広域に拡散し、場合によっては雨によって狭い地域に急速に降下し土壤に付着したり河川に溶けたりした。特に ^{131}I や ^{137}Cs による外部被爆や内部被爆（農作物、畜産物）が心配されるようになった。

原子核には安定なものと不安定なものがある。安定なものは全核種の数%しかなく、残りは不安定である。不安定な原子核は半減期程度の時間スケールで核分裂や α 崩壊、 β 崩壊、 γ 崩壊（第2章）を起してより安定な核になる。それが完全に安定な核種でなければ、再び崩壊が繰り返される。その際に放出されるエネルギーが膨大な熱であったり放射線であったりする。熱は発電に利用されるのだが、放射線が人体に害を及ぼすのである。私たちの細胞やDNAは放射線のもつエネルギーによって傷ついている。しかし細胞にはDNAの傷をある程度は修復する仕組みがあるため、通常は健康を保つことができる。しかし放射線が強い環境では、傷が大量に積み重なり、細胞が死んでさまざまな急性症状をおこしたり、長期的にがんなどの原因になったりする恐れがある。

ここで放射能と放射線の違いについて簡単に説明する。放射性物質から放出されるエネルギーを（それが少数個の光子や電子等の荷電粒子に集中しているとき）放射線といい、その放射線を放出する能力が、狭義での放射能である。ただし、この放射能という話は広義では「放射線を放出する物質」という意味でも使用される。放射線を放出するものは放射性物質（詳しく言えばその中の放射性同位体の原子、さらに詳しく言えば、その中心にある不安定な原子核）であるから、「放射能」という語を「放射性物質」という語句で言い換えたほうが意味が明確になるのだが、世間一般には「放射能」という語がもっぱら使用されているので、本論文でも「放射能」という語を使うことにする。放射性物質をホタル、ホタルの光を放射線に例えると、放射能は光をだす能力ないし、ホタルそのものを意味することになる。この場合「放射線漏れ」はホタルの光が虫かごから漏れること、「放射能漏れ」はホタルが虫かごから逃げ出すことと同じである。ホタルの光は弱いので、何kmも先からは見えない。ホタルの光が見えたなら、ホタルが近くに居るのである。これと同様に原子核の放出する放射線は最も遠くまで届くガンマ線であっても空気中で吸収され何kmも先までは届かない。したがって放

射能問題は、事故をおこした原子力発電所付近を除けば放射性物質の拡散の問題である。また、ホタルは寿命がくると死に数が減っていき最後にはいなくなる。放射性物質も同様に半減期に伴って数は減っていく。ただし、放射性物質の寿命、即ち半減期はホタルのそれと比べるとはるかに時間が長い物があることが問題なのである。

この東京電力福島第一原子力発電所事故の放射能汚染問題に関連のあるテーマを卒業研究課題に選びたいと私達は考えた。そして配属された原子核理論研究室で指導を受け得るものとして選んだ課題は、公開されている核データを利用して、原子力発電の燃料である ^{235}U が核分裂した時の生成物のもつ放射能の時間推移を計算することであった。そのような計算結果があれば、原子力発電の基本的な過程であるウラン原子核の核分裂により、どのような放射性核種が生成し、それらが時間とともにどれくらいの放射線を放出しつつ別の核種に変わって行くのかを知ることができる。そして、例えば、現時点での放射能汚染の主因である ^{137}Cs と比べて、原子炉中に残されている他の放射性核種の放射線の強さがどれくらいのものなのかが分かる。それが分かれば、例えば、将来のある時点で、万一、原子炉の格納容器が破壊されて炉中の全ての元素の同位体が飛散した場合に放射能汚染はどれくらいの規模にまで達し得るかの上限を粗く見積もることができるなど、この事故について考察をする際の参考として役立つと期待される。

なお、本論文の計算は、実際の原子炉内部の状況を正確に再現しようとするものではない。本論文で示すのは、一個の ^{235}U 原子核が熱中性子の吸収で核分裂した後の放射能の時間変化という、個々の原子炉の設計や運転履歴によらない汎用性のある結果である。例えば実際の原子炉中では、崩壊系列に加えて、原子炉中に大量に存在する中性子の捕獲による核種変換が起きているが、それを扱うには複雑な計算を要するので、本論文では全く考慮していない。仮に考慮できたとしても、計算結果は炉中の中性子のスペクトル分布という個々の原子炉の特性と運転状況に依存した汎用性の低いものとなる。また、実際の原子炉の中を再現するには、長い期間に起きた核分裂反応の結果の積分値を見なければならない。その計算はさほど難しくはなく、本論文で示す結果を時間積分すれば得られるが、結果はやはり過去の運転の履歴に依存した汎用性の低いものとなる。

また、本論文では、放射能はベクレル (becquerel, [Bq]) を単位として表す。1ベクレルとは1秒間に1つの原子核が崩壊して放射線を放つ放射能の量、即ち、単位時間あたりに崩壊する不安定原子核の個数である。1個の原子核が崩壊することによる放射線の影響が核種によって異なることを考慮するには、ベクレル値に核種に固有の変換係数を乗じることで、吸収されるエネルギー量(単位はグレイ, gray, [Gy])や、生体への影響量(単位はシーベルト, sievert, [Sv])に換算すればよい。変換係数は核データとして公開されている。しかし、データのフォーマットを理解してプログラムに読み込ませて利用するにはそれなりの手間を要するため、本論文ではベクレル以外の単位での値は示さなかった。

本研究課題の数学的な部分は、時間に関する1階の連立常微分方程式を解くことであるが、本論文ではその解を解析的に構成する具体的な手順を示した。そして、その解析解の値を数値的に求めるプログラムを自作した。なお、本論文で使った解析解による解法を、Runge-Kutta法[1]に代表されるような数値的な解法と比較すると、その

利点は、

1. 数値計算の誤差を小さく抑えることが容易であること
(計算方法を微調節するためのパラメータが存在しないのでパラメータの最適化による誤差の低減はできないが、そんな面倒な手間をかけなくても誤差は小さいということ)
2. 誤差の小ささの検証が容易であること
(計算方法のパラメータの微調整による誤差への影響を調べる手間が不要だということ)
3. 計算速度が超高速であること
(理由は解析解の数式の値を評価しているだけだからである。尤も、現在のコンピュータの高速性を考えれば、そこまで計算を速くすることが是非とも必要な場合は殆ど無いと思われる)

である。逆に欠点を挙げると、異なる核種の半減期が偶然にも正確に一致する場合に適用できないことである。一致すると言っても測定値の有効数字の範囲で一致しているということに過ぎないので、本論文の計算では、片方の半減期を有効数字の末桁のさらに百分の一程度人為的に変化させてから計算を行っている。測定データの誤差を考えればこれで十分な対応だと思われるが、勿論、もしも厳密に対応できるならそれに優ることはない。解析的に厳密に対処するためには、解析解の形を半減期の偶然縮退の状況に応じて変えねばならず、複雑な場合分けを伴う煩瑣な解となる。その代替となりうるような半数値的で簡便かつ高精度に偶然縮退に対処できる方法のアイデアを研究室では持っているが、その実装・パラメータ最適化処方の開発・精度検証には来年度以降の卒業研究生による 1 年間をかけた入念な研究が必要であろう。

他方、Runge-Kutta 法のような数値的解法の利点には、半減期が偶然に縮退しても何の困難も起きないことが挙げられる。さらに、放射性物質を外部と常に交換し続けるような、より広範な状況設定にも、計算の手続きを殆ど変えることなく適用できるという利点もある。

本研究課題で、原子核の放射性崩壊の物理的な理解が不可欠となる部分は、公的研究機関により WEB 公開されている核データから、原子核の放射性崩壊系列のプログラムの入力として適切な値を選んで入力データ設定サブルーチン (C 言語の関数) にデータとして書き込むことである。私達が計算を行うためにかけた手間の半分以上はこれらの核データの読み取りと入力に費やされたものである。必要なデータは、

1. ^{235}U の熱中性子の吸収による核分裂で生成する核種の分布
(崩壊系列の計算の初期値として用いる)
2. 崩壊系列上の核種の半減期と崩壊の分岐比、

である。第 1 項のデータは米国 National Nuclear Data Center (NNDC) からダウンロードした [2]。第 2 項のデータはやはり NNDC の核図表 (nuclear chart) 頁 [3] で各核種をクリックすると表示されるデータを利用した。ただし、崩壊による娘核のアイソマー (比較的寿命の長い励起状態、異性体) と基底状態への最終的な分岐比 (即ち、母

核の崩壊後、アイソマーと基底状態の半減期より遥かに短い時間しか経過していないが、アイソマー以外の全ての励起状態からの脱励起が実質的に完了した時点でのアイソマーと基底状態の存在確率の比)は NNDC の核図表頁からは得られないので、半減期の長い重要なものに限り、国際的分担により日本原子力研究機構が整備を担当している公開データ JENDL/FPD-2000(核分裂生成物核種 1229 核種の崩壊データ) [4] から読み取った。

なお、アイソマーと基底状態への分岐比データの入手方法については、指導教員を通じて日本原子力研究開発機構の核データ評価研究グループの FAQ メールアドレス (tokai-ndc-faq@jaea.go.jp)宛てに質問したところ、原子力基礎工学研究部門の片倉純一氏から簡明な返答があり、その中で JENDL/FPD-2000 データからのデータの読み方の一例を示して頂いた。片倉氏と核データ評価研究グループに対し、この返答について感謝の意を表する。

本論文で行ったような計算は専門家により既に行われてきたことである。しかし、少数の専門家が過去において計算プログラムを作り、その後の人間はユーザーとしてそのプログラムを使っていればよい、という研究態勢で十分であると考えたらそれは危険な考え方である。そういう態勢では、今回の事故の様に状況が従来とは違ったときに、従来どおりの正確さで計算結果が出せているのかどうかを判断できない恐れがある。このような重要な計算については、それを独立して行いうる多数の研究グループが並存し、各グループが常時、自ら計算式を導出して計算の原理を理解しなおし、プログラムを自作して数値計算上の困難な箇所を把握し、入力データのどれに結果が敏感に依存するかを認識しているような知識を生きた状態で維持している状態にあるべきだと思われる。

本論文の構成は下記の通りである。第 2 章で原子核と放射線についての基礎を概説し、第 3 章と第 4 章で放射性崩壊系列の従う微分方程式とその解法を詳説する。第 5 章で計算結果を示し考察を行う。第 6 章に本論文の内容のまとめを記す。付録では、計算に用いた 崩壊系列の計算に必要な核データの全て、(半減期と崩壊の分岐比、およびデータ欠落及び計算手法の都合による仮定)を表にして示し、さらに、自作した C 言語のプログラムのソースコードを示す。

第2章 原子核と原子核の崩壊様式

この章の執筆に際しては、主として文献 [5],[6],[7] を参考にした。

2.1 原子核

2.1.1 原子核の発見

1911 年のラザフォードらによる有名な散乱実験により原子核が発見された。この実験結果により α 粒子が物質にぶつかって、時々大きく方向を変えることが見つかった。このことは、原子の中には質量と正電荷が中心に集中していると仮定して見事に説明することができ、その部分は原子核と名付けられた。原子核は原子の寸法の約 1 万分の 1 足らずしかないにもかかわらず、原子質量の大部分を持っている。

原子は原子核とその周りに分布する電子とでできている。1 つの電子の電荷の絶対値を電気素量といい、現在の推奨値は $1.602176565 \times 10^{-19}$ クーロンである。原子番号を Z とすると、原子核が持っている正電荷はちょうど電気素量の Z 倍になっている。原子番号 Z が決まれば、どの元素かが決まることになる。

2.1.2 原子核の構成要素

1920 年代に知られていた原子核模型は、原子核の質量数・電荷・大きさなどを、当時発見されていた粒子と力（電磁気力と重力）によって説明した。しかし、1920 年代後半にかけて、当時知られていた原子核模型の様々な矛盾点が見つかるようになり、それが 1932 年のチャドウィックによる中性子の発見により一挙に解決された。中性子は電氣的に中性であり、陽子とほぼ同じ質量を持っているだけでなく、他の色々な性質においてもよく似ていることがわかっている。陽子と中性子は核子と総称される。陽子の質量は $m_p \approx 1.0073u$ 、中性子の質量は $m_n \approx 1.0085u$ である。ここで u は原子質量単位であり、 $1u$ は、 ^{12}C の中性原子 1 個の質量の $\frac{1}{12}$ と定義されており、現在の推奨値は $1u = 1.660538921 \times 10^{-27} \text{kg}$ である。

この中性子の発見の直後ハイゼンベルグにより、原子核が陽子と中性子から成り立っているという模型が提唱された。陽子数は原子番号に等しく Z で表し、中性子数を N で表す。陽子数と中性子数の和である核子数は A で表される。核子数は一般的に質量数と呼ばれ、原子核の構成要素は、 Z 、 N 、 A のうち 2 つの指定によって決定される。一般に、陽子数は同じであっても中性子数が異なる核種を同位体という。化学的な性質にはほとんど違いがないが原子核としては全くの別物であり、物理的には異なった振舞いをする。また放射能を持つ同位体を放射性同位体と呼ぶ。

2.1.3 原子核の構造

原子核崩壊を理解するために原子核の持つエネルギーについて把握する必要がある。今日ではアインシュタインの相対性理論から、エネルギー E 、質量 M 、真空中の光の速さ c の間に (2.1) 式の関係があることが知られている。

$$E = Mc^2 \quad (2.1)$$

光速 c は普遍的な定数であり、エネルギーと質量は完全な比例関係にある。つまり原子核の質量が重要となる。

原子核のエネルギーはあるしきい値 (核分離エネルギー) 以下では離散的である。原子核の状態は基底状態と励起状態に分類される。基底状態とは最もエネルギーが低い状態を示しており、それ以外の高いエネルギー状態にあるものを励起状態という。半減期 (核種が崩壊し核種の個数が半分になる期間) が長い励起状態をアイソマーと呼ぶが、単なる励起状態とアイソマーを区別するための半減期の境界がはっきりと定められているわけではない。核種の基底状態の質量は、約 2000 核種に対して実験的に分かっている [9]。

結合エネルギーと液滴模型

原子核がばらばらの核子に分かれてしまわないためには、ばらばらにした核子のエネルギーの和よりも原子核のエネルギーの方が小さいことが必要である。なぜなら、核子が運動エネルギーを持って原子核から飛び出してしまうからである。そしてその差のエネルギーを結合エネルギーという。つまり原子核を個々の核子に分解するためには、外部から結合エネルギー分のエネルギーを与える必要がある。また、エネルギーと質量は同等であるため、原子核を構成している核子の質量の総和より、原子核の質量は結合エネルギーに相当する質量分小さい。この現象を質量欠損という。結合エネルギーを核子の個数 (質量数) で割った、1 核子あたりのエネルギーはその原子核の安定性を示すものであり、その値が大きいほど安定性が高いことになる。1 核子あたりの結合エネルギーが最も大きいところは質量数 60 付近となる。このことは質量数 56 の鉄が地球や宇宙において豊富に存在していることの重要な原因である。

ここで、原子核を一種の液滴としたワイツゼッカー・ベータの半経験的質量公式を示す。

$$M(Z, N) = Zm_H + Nm_n - B(Z, N)c^2 \quad (2.2)$$

$$B(Z, N) = a_v A - a_s A^{\frac{2}{3}} - a_c Z^2 A^{-\frac{1}{3}} - a_I (N - Z)^2 A^{-1} + \delta(Z, N) \quad (2.3)$$

(2.2) は質量 $M(N, Z)$ の原子核が、水素原子質量 m_H の Z 倍と中性子質量 m_n の N 倍よりも小さくなっていることを示す。この減少分は、結合エネルギー $B(Z, N)$ を質量に換算したものである。(2.3) は結合エネルギーを与えるもので、右辺の項は順番に、体積項、表面項、クーロン項、対称項、偶奇項となる。各項についての詳しい説明は文献 [6] を参照せよ。

(2.3) は実験データと比較され、係数や δ の値としていくつかの値が得られており、その一例としてワプストラが決めたものはメガ電子ボルト単位で以下ようになる。

$$a_v = 15.835, a_s = 18.33, a_c = 0.714, a_I = 23.20$$

$$\delta(Z, N) = \begin{cases} 11.2A^{-\frac{1}{2}} : Z, N \text{ とも偶数のとき} \\ 0 : A \text{ が奇数のとき} \\ -11.2A^{-\frac{1}{2}} : Z, N \text{ とも奇数のとき} \end{cases} \quad (2.4)$$

このワイツゼッカー・ベータの半経験的質量公式は結合エネルギーの実験値と数メガ電子ボルトの誤差があり、質量公式の質を上げる必要がある。そのためには陽子数 Z と中性子数 N の関数の改良だけではなく、原子核の殻構造の効果 (量子力学的効果) を取り入れる必要がある。殻構造は原子核の構造を知るために必要なものであるが、量子力学の分野となるため、ここでは参考文献 [6] を参照せよ。

魔法数

魔法数とは原子核の安定性などに関連した数で、魔法数だけの陽子数や中性子数をもつ核種の基底状態は、それと陽子数や中性子数の少し違う核種に比べて、相対的に低いエネルギー、また小さい質量を持つことになり、強く結合していることになる。そして陽子数と中性子数が共に魔法数の原子核では魔法数の性質が著しくなる。現在認められている魔法数は 2、8、20、28、50、82、126 であり、質量数 208 の鉛は陽子数と中性子数が共に魔法数である。この 2 つの魔法数を持つ鉛には全く崩壊が見られない。

魔法数は安定核種の分布などの解析から、1930 年代から 1948 年にかけて見つけれられたが、当時は不思議とされていた。1949 年に発表された M・G・メイヤー女史の論文「殻模型について」とハクセル、イェンゼン、ジュース三氏の共同論文によって、核子の受けるポテンシャルが核子の位置だけでなく核子の軌道角運動量とスピン角運動量の内積にも依存することが発見され、魔法数は解明された。魔法数を決定するには、原子質量を分析するのが最も直接的であり、原子質量と密接に関連した β 崩壊の Q 値が魔法数の決定に大きな役割を果たすことになる。ここでいう Q 値とは、はじめ静止していた原子核が崩壊したときに生じる運動エネルギーを示す。エネルギー保存則を使えば Q 値は静止エネルギーの減少量に等しい。よって Q 値は、崩壊や反応に関係する原子核や核子の質量から計算できる。

2.2 原子核崩壊

2.2.1 放射性崩壊の機構

放射性崩壊の種類

ある種の不安定な原子核は外からの力を受けなくても、安定な原子核になろうと崩壊 (壊変) する。その際に粒子や電磁波である α 線や X 線を放出する。このような原子

核の性質を放射能と呼ぶ。不安定な原子核が安定な原子核になろうとする際に放出される粒子または電磁場は7種類あり、中性子 (n)、陽子 (p)、電子 (e^-)、陽電子 (e^+)、光子 (γ)、 α 粒子 (α)、ニュートリノ ($\nu, \bar{\nu}$) である。ニュートリノは物質との相互作用が極めて弱いのでその害を考える必要は全くない。これらの他に、軌道電子を原子核が吸収する電子捕獲がある。原子核の崩壊様式は α 崩壊、 β 崩壊、 γ 崩壊、核子放出、自発核分裂などがある。

一般に電子ないし陽電子を放出して崩壊する現象は β 崩壊と呼ばれる。 β 崩壊と電子捕獲の場合には原子核の電荷が変化し原子番号のみ変化する。 α 粒子 (He 原子核) を放出して別の原子核に変わる現象は α 崩壊と呼ばれ、He 原子核は2個の陽子と2個の中性子からできているので、電荷および原子番号と質量が変化する。 β 崩壊、 α 崩壊と共に異種の元素となり化学的性質も変化する。光子 (電磁波) を放出する γ 崩壊においては、核子の再配列があるのみであり、電荷や質量数、化学的性質の変化は起こらない。 γ 崩壊は普通、 β 崩壊、 α 崩壊と共に起こる。また1つの原子核に2つの崩壊の可能性のあることを分岐と呼ぶ。 α 崩壊や β 崩壊など、どの崩壊に従うかは原子核の不安定性と核子の配列に依存している。核子放出は強い相互作用によって、 β 崩壊は弱い相互作用によって起こり、 γ 崩壊は電磁相互作用によって起こる。

2.2.2 核分裂

核分裂反応

核分裂反応は、1938年にハーンとシュトラスマンの2人によって発見された。核分裂反応では、原子核は通常2つのかなり重さの近い分裂片に分裂する。最初の発見はウランに中性子を当てることによって行われた。また、重い原子核であれば外部からの十分なエネルギーを与えられれば、大抵、核分裂を生ずることが明らかとなった。低いエネルギーの中性子でも核分裂を起こすような物質を核分裂性物質と呼び、その物質は ^{235}U , ^{239}Pu , ^{241}Pu , ^{233}U などである。それとは反対に高エネルギーの中性子でないと核分裂を生じない物質もあり、 ^{238}U , ^{232}Th がそうである。

核分裂生成物

核分裂が生ずると、原子核は通常2個の分裂片に分裂する。運動中の分裂片を核分裂片と呼び、静止状態になったものを核分裂生成物と呼ぶ。

核分裂生成物として生成される核種は、核分裂をした核種および中性子のエネルギーによって異なるが、大体の傾向は、質量数95付近と質量数140付近の2箇所をピークに持つ。重さの違う分裂片ができやすい理由は、原子核を液滴として考えると、分裂する際に原子核がまず細長い楕円形になるが、大きさの違う分裂片になる位置にくびれが生じやすいからである。図 2.1 に ^{235}U 1個が核分裂したときの核分裂生成物の核分裂収率を示す。実験データは参考文献 [2] からとった。

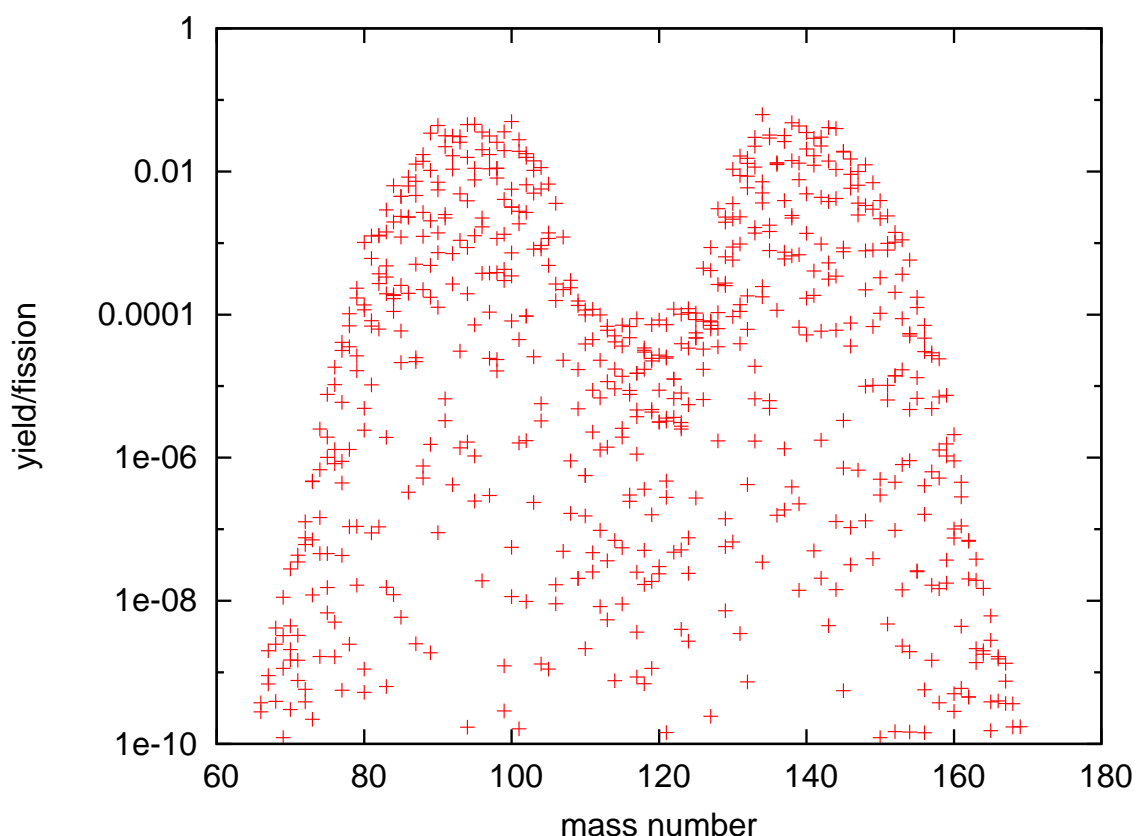


図 2.1: 熱中性子による ^{235}U , 1 個が核分裂したときの核分裂生成物

図 2.1 で核分裂収率を縦軸に、質量数を横軸にとり、核分裂収率を赤色の十字でプロットした。

^{235}U が中性子と複合核を作ると、中性子と陽子の比率は 1.57 となる。これがより質量の小さい 2 つの分裂片に分裂するのであるから、安定なためには、中性子の個数が多すぎることになる。この不安定さは中性子の放出、または、 β 線の放出によって安定化される。本研究では ^{235}U , 1 個が熱中性子 (遅い中性子) の吸収によって核分裂したときに生ずる核分裂生成物が安定核種に向かって変化する様子をプログラムを作り計算によって再現するのが目的である。同じデータを陽子数、中性子数に対してプロットしなおしたものを図 2.2 に示す。

Common logarithm of yields of fission products of ^{235}U by thermal neutron

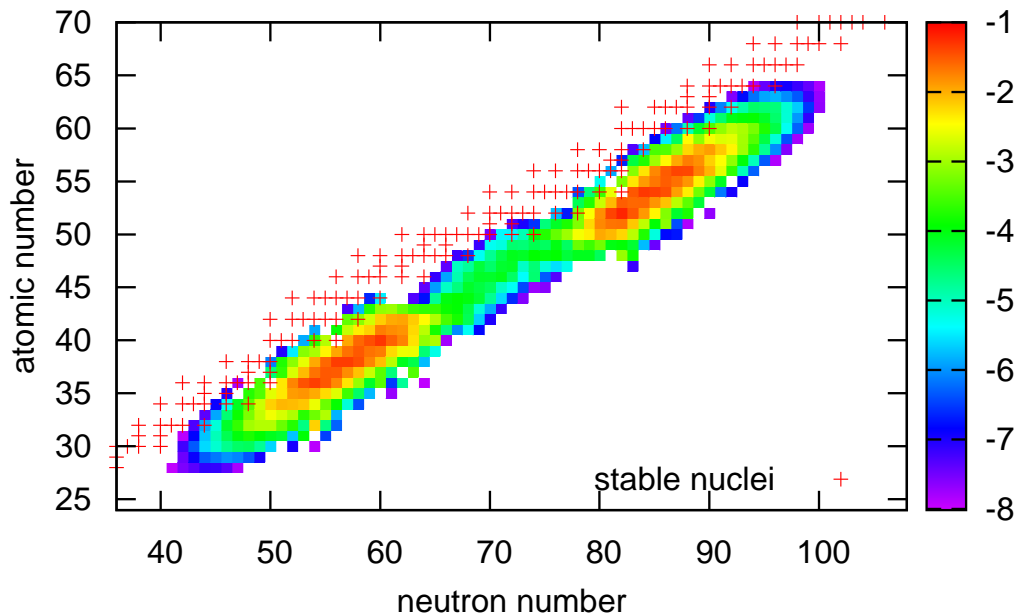


図 2.2: 熱中性子による ^{235}U 1 個が核分裂したときの核分裂生成物

図 2.2 の縦軸に陽子数を、横軸に質量数をとる。右側に出ている色グラフが核分裂収率の値の常用対数値を表し、 -1 と表記されているのは 10^{-1} を示している。図 2.2 の中に安定核種を赤色の十字でプロットしてある。崩壊は中性子が多い右下の核種から左斜め上の核種へと崩壊し、安定核種に至るまで崩壊を続ける。

臨界エネルギー

臨界エネルギーとは核分裂が生じるためのエネルギーを表す。 ^{235}U の核分裂の場合を考えると、 ^{235}U が中性子と衝突して複合核 ^{236}U を作り、この複合核が細長い楕円形になり、さらに 2 つの分裂片へと分裂する。この場合、複合核 ^{236}U は細長い楕円形になるにつれてポテンシャルエネルギーを増加し、臨界エネルギーに達したところで原子核は分裂することになる。つまり核分裂するためには、臨界エネルギー分のエネルギーを与える必要がある。臨界エネルギーの大きさは原子核によって異なるが、 $\frac{Z^2}{A}$ の増加に伴って減少する。 Z が関係するのは、静電的反発力が分裂の原因だからである。

中性子による核分裂の場合、複合核には 1 個の中性子捕獲による結合エネルギーと、その中性子の持つ運動エネルギーの合計分だけの励起エネルギーが与えられることになる。そして、この励起エネルギーが臨界エネルギーよりも大きければ核分裂が可能となる。ここで ^{235}U の臨界エネルギーは約 6.5MeV であり、結合エネルギーは約 6.8MeV である。よって、中性子の運動エネルギーが零であっても、中性子が ^{235}U に吸収され

れば核分裂は可能となる。

2.2.3 α 崩壊

α 崩壊は、重い原子核が α 粒子 (He 原子核) を放出して、より軽い原子核とヘリウム原子核になることである。He 原子核は 2 個の陽子と 2 個の中性子からできているので、元の原子核を X、新しい原子核を Y とすると、 α 崩壊は次のように示すことができる。

$${}^A_ZX \rightarrow {}^{A-4}_{Z-2}Y + {}^4_2\text{He}(\alpha) \quad (2.5)$$

α 粒子と元の原子核の間には核力とクーロン力が働いており、核力は短距離力である。 α 粒子が核力を受けない距離まで離れると、ほとんどクーロン力しか働かないが、そのポテンシャルエネルギーは、崩壊による Q 値よりもはるかに高い。つまり α 粒子には核力とクーロン力を上回るだけのエネルギーを持っているのではなく、クーロン障壁をトンネル効果により通り抜けている。

2.2.4 γ 崩壊

γ 崩壊は高いエネルギーの状態から低いエネルギーの状態へ遷移するとき光子 (γ 線) を放出する崩壊である。 γ 崩壊は電磁波相互作用によって起こる崩壊で光子を放出する。光子は非常にエネルギーの高い電磁波であり、電荷も質量もないため、光子を放出する原子核は電荷も質量も変化しない。安定な核に変化するという状態変化が起こるだけである。

γ 崩壊は、原子核の構造を研究するために最も重要な現象とされており、何万という遷移が実験的に調べられ、それをもとに多くのエネルギー準位やそのスピン・角運動量・パリティ(空間反転に関して波動関数がどう変わるかを示す量子数)が決められてきている。原子核崩壊を考察するうえで選択則というものがあるが、全角運動量に関する選択則や、パリティに関する選択則は最も基本的な対称性に因る選択則であるから、スピン角運動量やパリティの決定は大変重要である。

2.2.5 β 崩壊

β^- 崩壊、 β^+ 崩壊、(軌道) 電子捕獲について下に示す。これらは、弱い相互作用で起こる広い意味での β 崩壊である。

β 安定線

β 安定線とは β 崩壊に対して最も安定な核種を示す線である。最も安定な核種とは結合エネルギーが最大な核種と言える。ワイツゼッカー・ベータの半経験的質量公式には、陽子数 Z に依存する項が、クーロン項と対称項の 2 つある。質量数 A が等しい原子核の中で結合エネルギーが最大なものは、 A と Z で表した結合エネルギーを Z で

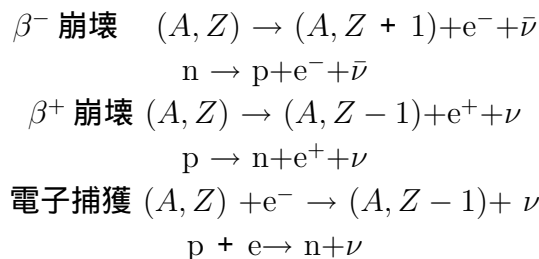
偏微分したものが零であるときとなる。 β 安定線は陽子数と中性子数が等しくなる直線 ($N = Z$) にはならず、質量数が大きくなるにつれて中性子数が多くなる方へ傾いている。

横軸に N 、縦軸に Z をとったいわゆる核図表 (nuclear chart) 上で β 安定線よりも右下にある、比較的中性子の多い核種は β^- 崩壊によって β 安定線の方へ近づき、 β 安定線より左上にある、比較的陽子の多い核種は β^+ 崩壊や軌道電子捕獲によって β 安定線に近づく。 β 安定線から遠く離れた核種ほど半減期は短く、 β 安定線に近づくにつれ半減期は長くなるという一般的な傾向がある。

本研究では β 安定線よりも右下にある、比較的中性子の多い核種の β^- 崩壊の様子を計算で再現する。

β 崩壊と電子捕獲

比較的中性子の多い核種は β^- 崩壊する。 β^- 崩壊では中性子が陽子に変わり、その時に電子 e^- と反電子ニュートリノ $\bar{\nu}$ が生まれ、それらが放出される現象である。陽子数は増えるが質量数に変化はない。比較的陽子の多い核種は β^+ 崩壊する。 β^+ 崩壊では陽電子 e^+ が陽子の電荷を持ち去り、陽子が中性子になる。 β^+ 崩壊によって放出されるのは、陽電子 e^+ と電子ニュートリノ ν である。 β^+ 崩壊でも全核子数は変化せず、崩壊先の新しい原子核も元の原子核と同じ質量数を持つ。軌道電子を捕獲して、 β^+ 崩壊と似たような崩壊を起こすのが電子捕獲である。この捕獲された電子は、核内陽子と結合して中性子となる。余分なエネルギーは γ 線として放出される。陽子数は減るが全質量数に変化はない。また崩壊は原子核の中の 1 つの核子によって起こっている。まとめて下に反応式で示す。



β^+ 崩壊と電子捕獲の違いは放出されるエネルギーの大きさであり、 β^+ 崩壊の方が $2m_e c^2$ だけ放出エネルギーが大きいことである。つまり β^+ 崩壊は電子捕獲に比べて $2m_e c^2$ だけ余分なエネルギーが必要である。この $2m_e c^2$ は、 β^+ 崩壊では陽電子が 1 個生成するのに対し、電子捕獲では電子が 1 個消滅することによる。

β 崩壊によって生じる Q 値 (エネルギー) は、 β 崩壊によって放出される粒子によって持ち去られる。 β^- 崩壊の場合は、電子と反電子ニュートリノの運動エネルギーとなる。 β^+ 崩壊の場合は、陽電子と電子ニュートリノの運動エネルギーとなる。

原子核中ではなく、孤立して存在する核子については、 β^- 崩壊によって中性子は陽子に変わることができるが、陽子は安定で崩壊することはできない。なぜなら、中性子の方が陽子より少し質量が大きく、質量とエネルギーの同等性より、中性子の方が陽子よりもエネルギーの高い状態であるからである。

第3章 原子核の崩壊を表す微分方程式

この章では原子核の崩壊を表す微分方程式とその一般解の求め方について述べる。

放射性核種は、時間的に一定の確率でランダムに崩壊する。時間が経つに従い原子核の個数は減り、その結果、1秒あたりの崩壊の回数も減少する。しかし、崩壊の性質は変わることはないので半減期は同じである。

3.1 1つの核を扱う場合

放射性崩壊による原子核の個数の時間的変化を考察する。

時間幅 dt の間に崩壊する原子核の数は、そのときの原子核の数に比例する。

$$dN = -\lambda N dt \quad (3.1)$$

λ は崩壊定数 (decay constant) と呼ばれる。(3.1) は簡単に積分できて、 $t = 0$ における核の個数を N_0 とすると

$$N = N_0 e^{-\lambda t} \quad (3.2)$$

となる。半減期 $T_{\frac{1}{2}}$ に相当する時刻においては、定義より

$$\frac{N_0}{2} = N_0 e^{-\lambda T_{\frac{1}{2}}} \quad (3.3)$$

が成立する。この式の両辺の自然対数をとると

$$\ln 2 = \lambda T_{\frac{1}{2}} \quad (3.4)$$

が得られる。つまり、 λ は半減期 $T_{\frac{1}{2}}$ と次の関係がある。

$$\lambda = \frac{\ln 2}{T_{\frac{1}{2}}} \approx \frac{0.693}{T_{\frac{1}{2}}} \quad (3.5)$$

3.2 複数の核を扱う場合

複数の核を扱う場合は1つの核のときとは異なり λ が行列になり、 N がベクトルになるので、 λ の固有値を求め、次に固有ベクトルを求める作業をする必要がある。

3.2.1 定数係数 1 階常微分方程式の一般解について

行列の固有値は一般的には解析的には求まらないが、固有値が求まれば固有ベクトルは解析的に求まる。

また、固有値が全て異なる場合と縮退のある場合とでは、微分方程式の一般解の形は変わってくる。

(A) 固有値が、全て異なる実数の場合

$$\frac{d}{dt} \begin{pmatrix} n_1 \\ \vdots \\ n_N \end{pmatrix} = \begin{pmatrix} A_{ij} \end{pmatrix} \begin{pmatrix} n_1 \\ \vdots \\ n_N \end{pmatrix} \quad (3.6)$$

において係数行列 A の固有値を $(-\lambda_1, \dots, -\lambda_N)$ として、そのそれぞれに属する固有ベクトルを

$$\begin{pmatrix} p_{11} \\ \vdots \\ p_{1N} \end{pmatrix}, \dots, \begin{pmatrix} p_{N1} \\ \vdots \\ p_{NN} \end{pmatrix} \quad (3.7)$$

とすると $\lambda_i \neq \lambda_j$ ($1 \leq i < j \leq N$) のとき一般解は

$$\begin{pmatrix} n_1 \\ \vdots \\ n_N \end{pmatrix} = \sum_{i=1}^N C_i \begin{pmatrix} p_{i1} \\ \vdots \\ p_{iN} \end{pmatrix} e^{-\lambda_i t} \quad (C_i \text{は任意の定ベクトル}) \quad (3.8)$$

となる。

(B) 固有値に縮退がある場合

2 次正方行列の場合では、固有方程式が重解をもつと固有ベクトルは 1 つしか存在せず、対角化することはできない。重解の固有値を $-\lambda$ とすると行列が三角行列でない場合、一般解は

$$\begin{cases} n_1 = C_1 e^{-\lambda t} + \{bC_2 + (a + \lambda)C_1\}te^{-\lambda t} \\ n_2 = C_2 e^{-\lambda t} + \{cC_1 + (a + \lambda)C_2\}te^{-\lambda t} \end{cases} \quad (3.9)$$

となる。固有値が異なる実数のときと比べると計算が複雑になり様々な縮退パターンに対し、統一的な方法で解析的に求めることは非常に困難である。そこで縮退をなくすために λ を入力データのもつ誤差の 1% ほど人為的にずらした。

3.2.2 行列が下三角行列の場合

行列が下三角行列の場合は、行列の対角成分が行列の固有値であるから、固有値は最初から求まっている。即ち、定数係数 1 階常微分方程式

$$\frac{d}{dt} \begin{pmatrix} n_1 \\ \vdots \\ n_N \end{pmatrix} = \begin{pmatrix} \lambda_{11} & & \\ & \ddots & \\ & & \lambda_{NN} \end{pmatrix} \begin{pmatrix} n_1 \\ \vdots \\ n_N \end{pmatrix} \quad (3.10)$$

を初期条件 $t=0$ で

$$\begin{pmatrix} n_1 \\ \vdots \\ n_N \end{pmatrix} = \begin{pmatrix} n_1^0 \\ \vdots \\ n_N^0 \end{pmatrix} \quad (3.11)$$

のもとで解くとき、係数が

$$\begin{cases} i = j \text{ のとき} & \lambda_{ii} = -\lambda_i < 0 \\ i < j \text{ のとき} & \lambda_{ij} = 0 \\ i > j \text{ のとき} & \lambda_{ij} \geq 0 \\ i \neq j \text{ のとき} & \lambda_i \neq \lambda_j \text{ とする。} \end{cases} \quad (3.12)$$

を満たしているならば、その解法は、三角行列の固有値は対角成分 $(-\lambda_1, \dots, -\lambda_N)$ であるから

$$n_i = \sum_{j=1}^i C_{ij} e^{-\lambda_j t} \quad (3.13)$$

とおき、微分方程式及び初期条件に代入して、以下のようにして $C_{ij} (1 \leq i \leq N, 1 \leq j \leq i)$ を定める。

$$\begin{aligned} \frac{d}{dt} n_i &= \frac{d}{dt} \sum_{j=1}^i C_{ij} e^{-\lambda_j t} \\ &= - \sum_{j=1}^i C_{ij} \lambda_j e^{-\lambda_j t} \\ \sum_{j=1}^i \lambda_{ij} n_j &= \sum_{j=1}^i \lambda_{ij} \sum_{k=1}^j C_{jk} e^{-\lambda_k t} \\ &= \sum_{j=1}^i \sum_{k=1}^j \lambda_{ij} C_{jk} e^{-\lambda_k t} \\ &= \sum_{k=1}^i \sum_{j=k}^i \lambda_{ij} C_{jk} e^{-\lambda_k t} \\ &= \sum_{j=1}^i \sum_{k=j}^i \lambda_{ik} C_{kj} e^{-\lambda_j t} \quad (j \text{ と } k \text{ の名前を入れ替えた}) \end{aligned} \quad (3.14)$$

$$\begin{aligned}
\sum_{j=1}^i (C_{ij}\lambda_j + \sum_{k=j}^i \lambda_{ik}C_{kj})e^{-\lambda_j t} &= 0 \quad (1 \leq i \leq N, 1 \leq j \leq i) \\
C_{ij}\lambda_j + \sum_{k=j}^i \lambda_{ik}C_{kj} &= 0 \\
C_{ij}\lambda_j + \sum_{k=j}^{i-1} \lambda_{ik}C_{kj} + \lambda_{ii}C_{ij} &= 0 \quad (2 \leq i \leq N, 1 \leq j \leq i) \\
(\lambda_i - \lambda_j)C_{ij} &= \sum_{k=j}^{i-1} \lambda_{ik}C_{kj}
\end{aligned} \tag{3.15}$$

となる。よって、Cは

$$\begin{cases} C_{ii} = n_i^0 - \sum_{j=1}^{i-1} C_{ij} & (2 \leq i \leq N) \\ C_{ij} = \frac{1}{\lambda_i - \lambda_j} \sum_{k=j}^{i-1} \lambda_{ik}C_{kj} & (2 \leq i \leq N, 1 \leq j \leq i-1) \end{cases} \tag{3.16}$$

となる。したがってこのCを(3.13)に代入することで n_i は求まる。

第4章 計算方法の詳細

4.1 半減期と分岐比から λ_{ij} を決める方法

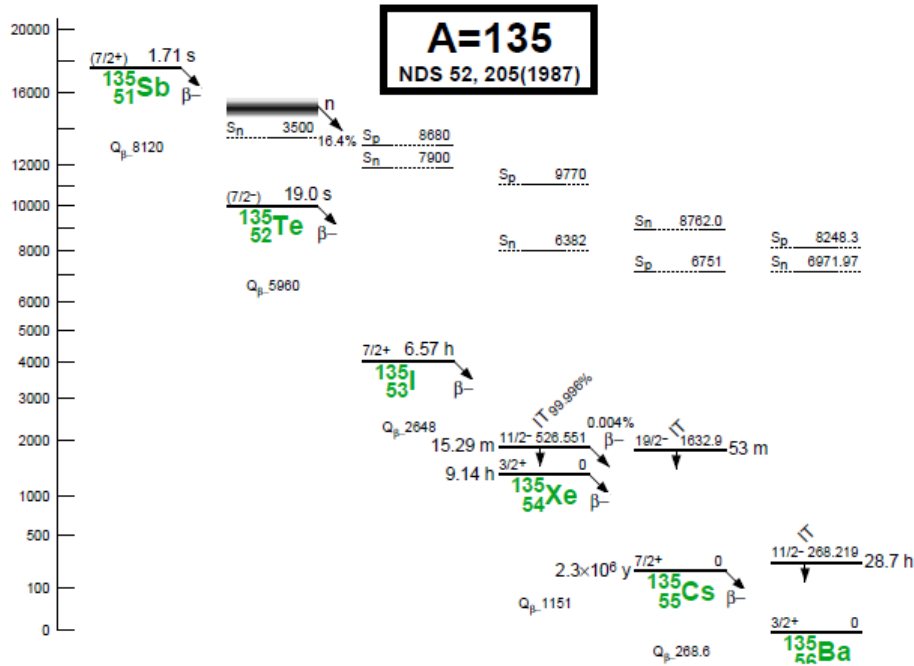


図 4.1: β 崩壊系列の例 (Table of Isotopes[11])

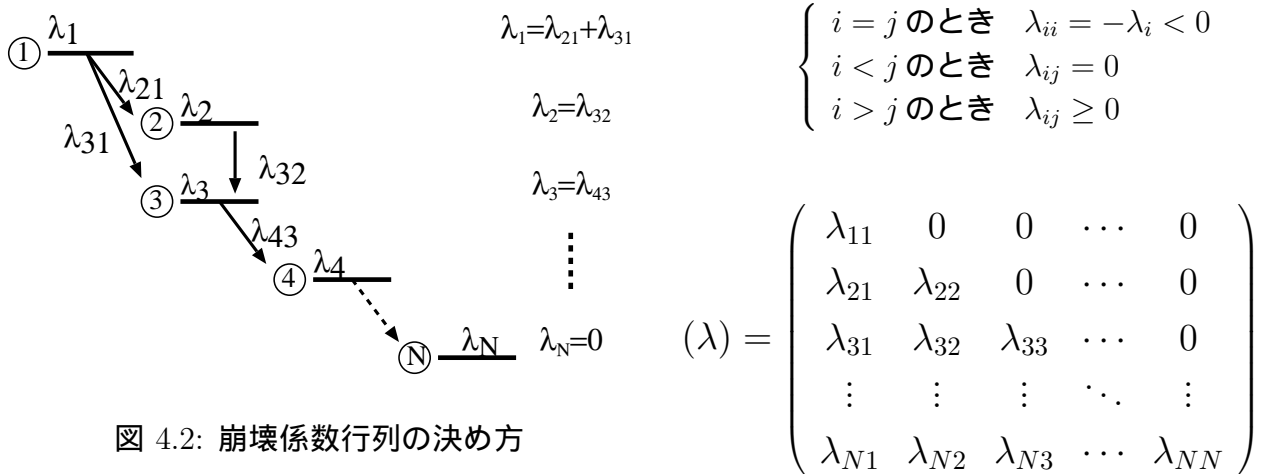


図 4.2: 崩壊係数行列の決め方

エネルギーの高い核は、不安定な核でありエネルギーの低い核 (より安定な核) へ崩壊する。その際に崩壊先の核 (娘核) の基底状態へと崩壊するとは限らず、励起状態へ

と崩壊する確率もゼロとは限らない。通常は、母核や娘核の半減期より遥かに短い時間で励起状態から基底状態へのガンマ崩壊が起きるので、非常に良い近似で崩壊途上に存在する励起状態を無視して、母核の基底状態から娘核の基底状態への直接の遷移が母核の半減期で起きるとしてよい。しかし励起状態の中に（励起状態としては）非常に半減期の長いもの（アイソマー）が存在する場合は、そのような近似は成り立たず、崩壊途上に存在するアイソマーを微分方程式にあらわに取り入れて計算しなければならない。核種の個数の時間変化を考察する上で、この分岐比について考える必要がある。分岐比は National Nuclear Data Center から WEB 公開されているデータにある。しかし、ところどころ分岐比が公開されていない核があり、その場合、その前後の核の分岐比の傾向から 100% 基底状態に崩壊することにしたたり、100% 励起状態にしたたり、50% 基底状態 50% 励起状態に崩壊するということにしたたり、私たちが推測し仮定して分岐比を設定した。特に半減期の長い重要なアイソマーに限り、国際的分担により日本原子力研究機構が整備を担当している公開データ JENDL/FPD-2000(核分裂生成物核種 1229 核種の崩壊データ) [4] から分岐比を読み取った。分岐比や半減期等として使用したデータや仮定をまとめたものを付録 1 に掲載する。原子核の β 崩壊系列を表す微分方程式を解く際に、崩壊係数行列の λ を設定する必要がある。 λ は、図 4.2 のように原子核をエネルギーの高いものから順番に並べれば下三角行列になる。対角成分 λ_{ii} は、核種 i の半減期 $T_{\frac{1}{2}}$ を使って

$$\lambda_{ii} = -\frac{\ln 2}{T_{\frac{1}{2}}} \quad (4.1)$$

と決定される。非対角成分 $\lambda_{ij} (i > j)$ は、例えば核 1 から核 2 への分岐比が b_2 、核 3 への分岐比が b_3 なら ($b_2 + b_3 = 1$)、 $\lambda_{21} = -\lambda_{11}b_2$ 、 $\lambda_{31} = -\lambda_{11}b_3$ と決定される。

4.2 アイソマーの扱い

既に述べたように、励起状態のなかでも寿命の比較的長いものをアイソマーという。短寿命の励起状態の核は多く存在するが、すぐに γ 崩壊し基底状態へ落ちる。そこで、短寿命の励起状態はプログラム上全て無視した。しかしアイソマーは、基底状態と同程度の半減期を持っていたり、アイソマーから基底状態へ崩壊せず、次の核へ直接 β 崩壊することもあり無視することはできない。したがってアイソマーは、プログラム上では独立した 1 つの核種のような扱いをすることにした。

4.3 核の個数の初期値の設定

^{235}U は中性子を吸収すると二つの核に核分裂する。その際、質量の同じ二つの核になるより、質量の異なる二つの核になることが多い。その質量数の出現確率もデータとしてあり、その分布は図 2.1、図 2.2 のようである。本研究では ^{235}U が一つ核分裂したときの放射能の時間推移を見るため核の出現確率の和は 2 に規格化した。

4.4 β^-n 崩壊の扱い

β^- 崩壊には、 β^- 崩壊後、中性子を放出し質量数が一つ少ない系列へ移動する β^-n 崩壊がある。 β^-n 崩壊は安定線から遠くはなれた核で起きるので半減期が非常に短い。また、本研究で作成したプログラムは質量数が同じ系列での放射性崩壊を計算するものである。崩壊の途中で核が別の系統に移ることは想定していない。その理由は、系統を移る崩壊を考慮するには全ての質量数の崩壊系列を一括して計算する必要があるが、そうすると半減期の偶然縮退が多発して数値計算の誤差が増大しないかどうかを詳しく調べる必要が生じるからである。その手間を省くため、質量数毎に独立して計算させることが望ましい。そこで β^-n 崩壊する核は半減期が非常に短いということ、系列端に多いことを考慮し、プログラム上では時間変化する核種にはいれず、初めに全て崩壊してしまったという近似をして次の核の初期値に繰り込むことにした。

4.5 C_{ij} の決定方法

C_{ii} 、 C_{ij} は図 4.3 のような手順で決定していく。

$$C_{11} = n_1^0, \quad C_{ij} = \frac{1}{\lambda_i - \lambda_j} \sum_{k=j}^{i-1} \lambda_{ik} C_{kj} \quad (2 \leq i \leq N, \quad 1 \leq j \leq i-1)$$

$$C_{ii} = n_i^0 - \sum_{j=1}^{i-1} C_{ij} \quad (2 \leq i \leq N, \quad 1 \leq j \leq i-1)$$

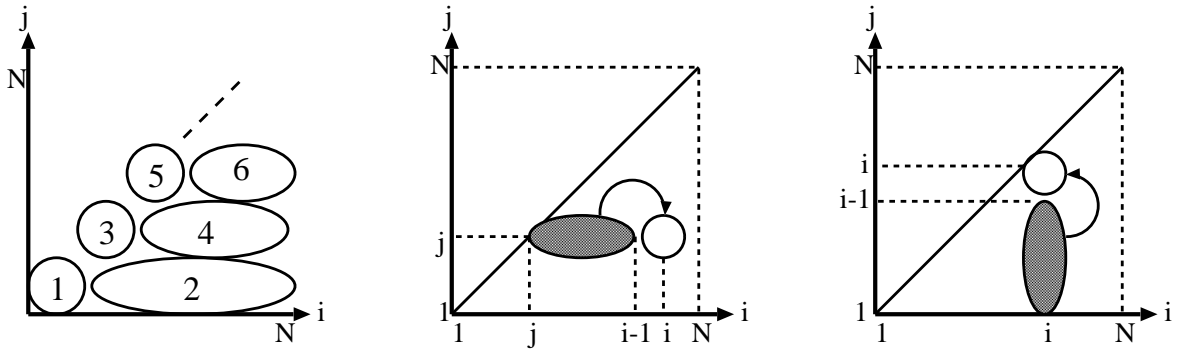


図 4.3: 左図： C の決定の流れ 中図： C_{ij} の決定の手順 右図： C_{ii} の決定の手順

1 の初期値で決まっている。2、3、6 のエリアは C_{ij} の式を用いる。 C_{ij} を求める場合は C_{jj} から $C_{i-1, j}$ の和となる。次に 3, 5 のエリアを求める場合は C_{ii} の式を用いる。 C_{i1} から C_{ii-1} までの和が C_{ii} となる。

4.6 n_i の和のチェック

放射性物質核種は系の崩壊とともにその個数は増減する。そこでプログラム中で崩壊の初めの核種の個数と最後の個数の和をチェックするようにした。もし初めと最後の

核種の個数が違っていれば、崩壊の系列のどこかで正しく崩壊していることにならない。

チェックの結果、相対誤差は計算した全ての時刻において、 10^{-16} 以下であり計算結果の信頼できることが確認できた。

第5章 結果

今回の研究では、 ^{235}U ,1 個の核分裂によって生成される核分裂生成物の β 崩壊による放射能の時間推移を自作のプログラム (付録 2 に掲載) によって再現する。 ^{235}U ,1 個の核分裂によって生成される核分裂生成物は第 2 章で説明したように、質量数 95 付近と質量数 140 付近の 2 箇所をピークに持つ。よって β 崩壊による放射能の時間推移を自作のプログラムによって再現する原子核の範囲を、質量数 79~109 と質量数 126~154 とした。

自作のプログラムに入力する半減期や分岐比の値は付録 1 に載せた、decay data の表を見ていただきたい。計算は 2 人で分担し、質量数 79~109 を森近が担当し、質量数 126~154 を大沼が担当した。

5.1 質量数83の系列における β 崩壊による放射能の時間推移計算結果

自作プログラムで計算した、質量数83の系列における β 崩壊による放射能の時間推移を図5.1に示す。

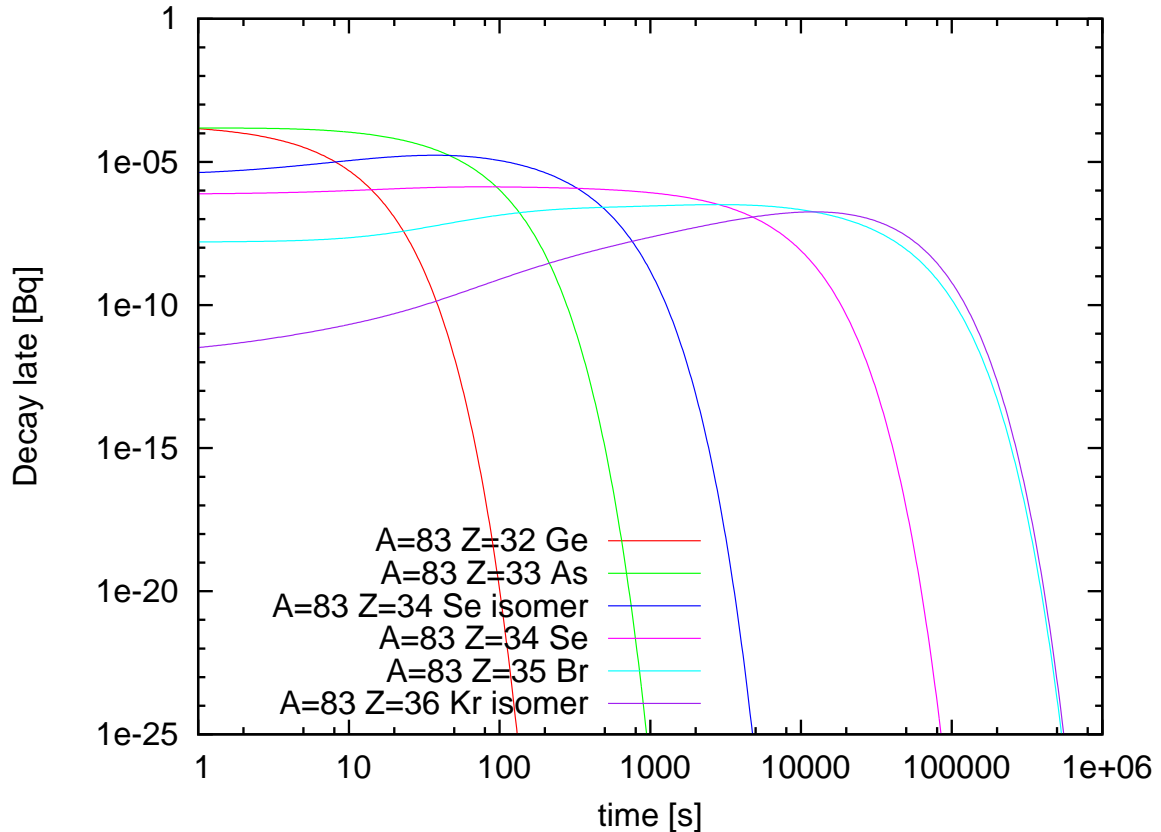


図 5.1: 質量数83系列における β 崩壊による放射能の時間推移計算結果

図5.1は両対数グラフであり、縦軸が放射能、横軸が時間である。原子核 Ge(ゲルマニウム) が核図表上で右下にある β 安定曲線から一番離れている核であり、As(ヒ素)、Se(セレン)、Br(臭素)、Kr(クリプトン)の順に β 安定曲線に近づき安定となる。

最も上流にある Ge の曲線は厳密に一成分の指数減少関数であるが、両対数グラフ中では図のような形の曲線になる。Br の曲線を見ると、100 秒までは時間経過と共に放射能の値が高くなっている。これは、Br より上流の核である、基底状態と励起状態の Se が Br に崩壊していることを示している。それから 10000 秒まではほぼ平行な直線となっている。これは、上流の核から崩壊してくる原子核の個数と、自分が崩壊する個数とが釣り合っているからで、このことを放射平衡と呼ぶ。その後 Se が全て Br に崩壊し、Br は自分の半減期に従い崩壊していく。

1×10^6 秒は約 1ヶ月であるので、この質量数83の核分裂生成物は1ヶ月の間に全て崩壊する事がわかる。

5.2 ^{235}U の核分裂生成物の β 崩壊による放射能の時間推移計算結果

^{235}U の核分裂生成物の β 崩壊による放射能の時間推移計算の全結果を図 5.2 に示す。

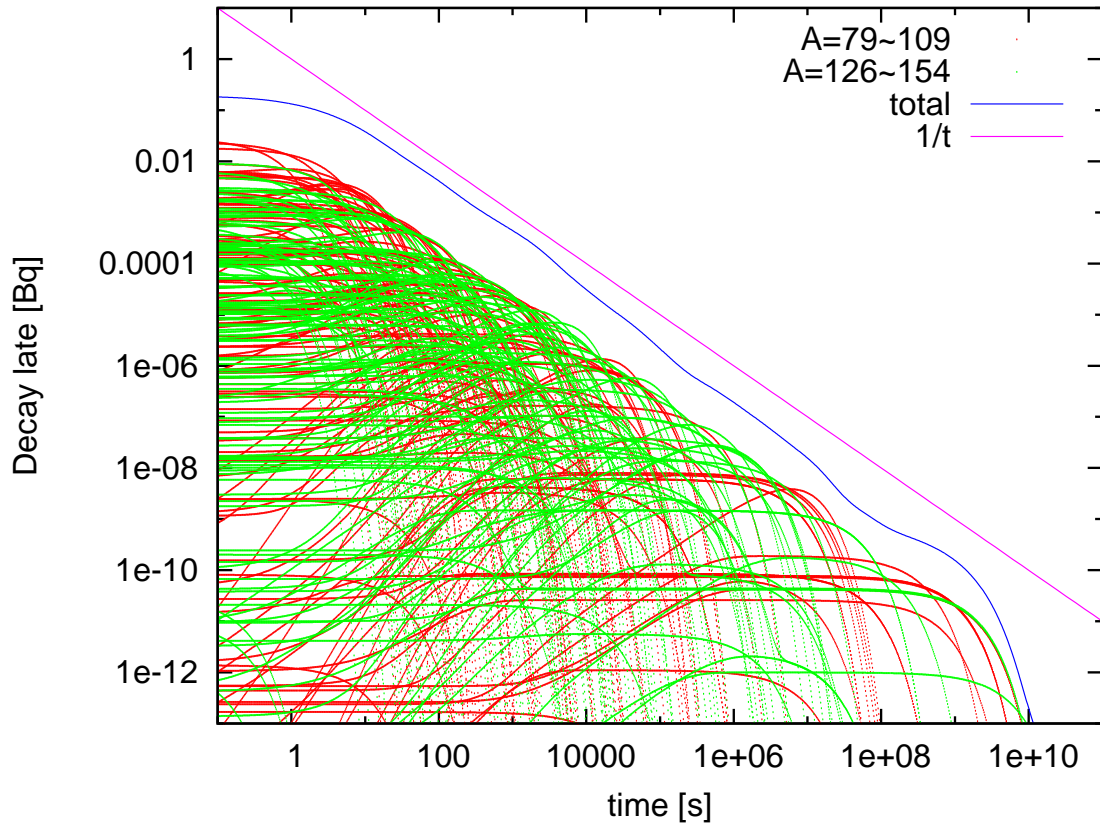


図 5.2: ^{235}U の核分裂生成物の β 崩壊による放射能の時間推移計算結果

図 5.2 も両対数グラフであり、縦軸が放射能、横軸が時間である。赤色と緑でプロットした点は、分担して計算した 2 つの質量数領域の結果を表している。赤色と緑色に偏りが無いことから、2 つの質量数領域にほぼ等しい放射能があることがわかる。無数の曲線が図の中に現れているが、これらの曲線の数だけ計算をしたことになる。

次に、total と書かれた青い線は、ある時刻における放射能の値を全て足したものを繋げた曲線である。この曲線と、ある時刻の放射能が一番大きい曲線の包絡線とは、ほぼ同じ形をしている事がわかる。これは、ある時刻の放射能が一番高い値に対して、それ以下の放射能の値は桁違いで小さく、そのため放射能を全て足しあげたとしても、一番高い値の 10 倍程度にしかならないからである。このことはどの時点でも言えることなので、包絡線と total 線が同じ形となる。

この total 線を見ると 10 秒より短い時間の時と、 1×10^9 秒 (100 年) より長い時間の時には、指数則にしたがった曲線を示しているが、10 秒 ~ 100 年の間では、傾き -1 の直線に近似出来るように見える。両対数グラフで傾き -1 の直線は、全放射能が時間に反比例することを意味している。個々の曲線は指数関数を示しているにもかかわらず、なぜ時間分の 1 という、べき関数が現れるのか私達は疑問に感じたので以下で考察した。

5.3 傾きについての考察

total 線に見られる直線についての考察をするために以下の図 5.3 を用意した。

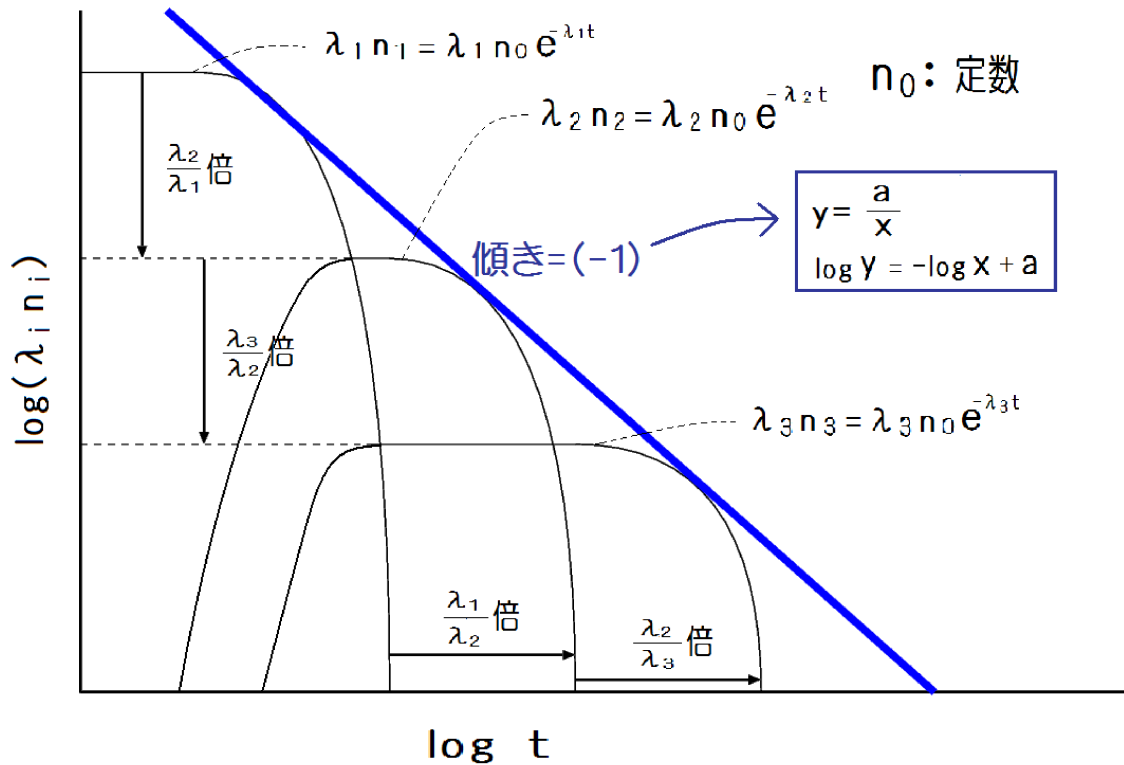


図 5.3: β 崩壊系列による放射能の時間推移の模式図

図 5.3 も両対数グラフであり、縦軸が放射能、横軸が時間となる。図の中にそれぞれの曲線の式が示してある。 λ は崩壊定数、 n は原子核の個数、 n_0 は初期の個数である。図 5.1 と同様に上流の核から下流の核へと崩壊していく。この図の中の上流の核を 1 番目の核とし、次の核を 2 番目の核、次の核を 3 番目の核とする。

1 番目の核は半減期に従い崩壊していき、2 番目の核に流れ込む。そのため、2 番目の核は放射能が増加していき、放射平衡となる。このことから、2 番目の核は最初から 1 番目の核のと同じ個数持っているに近似でき、即ち、 n_0 がどの核も等しくなる。これは β 崩壊系列の特徴である、下流の核になる (β 安定曲線に近づく) につれて、半減期が長くなることと、 β 崩壊では 1 番目の核の崩壊先が全て 2 番目の核であることが関係している。

このことから、包絡線上にある、2 番目の核の放射能の値は 1 番目の放射能の値の $\frac{\lambda_2}{\lambda_1}$ 倍となる。崩壊を表す式は $e^{-\lambda t}$ であるため、崩壊定数 λ が $\frac{\lambda_2}{\lambda_1}$ 倍小さくなると、 λ の逆比、 $\frac{\lambda_1}{\lambda_2}$ 倍、時間が大きくなる事がわかる。よって包絡線は -1 の傾きを持つ直線となる事がわかる。よって、全放射能と時間は 10 秒から 100 年の間は反比例の関係である事がわかる。

全放射能の曲線のなかに、 -1 の傾きの直線が見えるのは、 β 崩壊系列の特徴が生んだものだと結論できる。

5.4 時間に対する放射能が高い原子核

全ての計算結果を表す図 5.2 から、1 年後、10 年後、100 年後付近の時間の部分を取り出して、その時間に最も高い放射能示す核種を下の図 5.4、図 5.5 で示す。

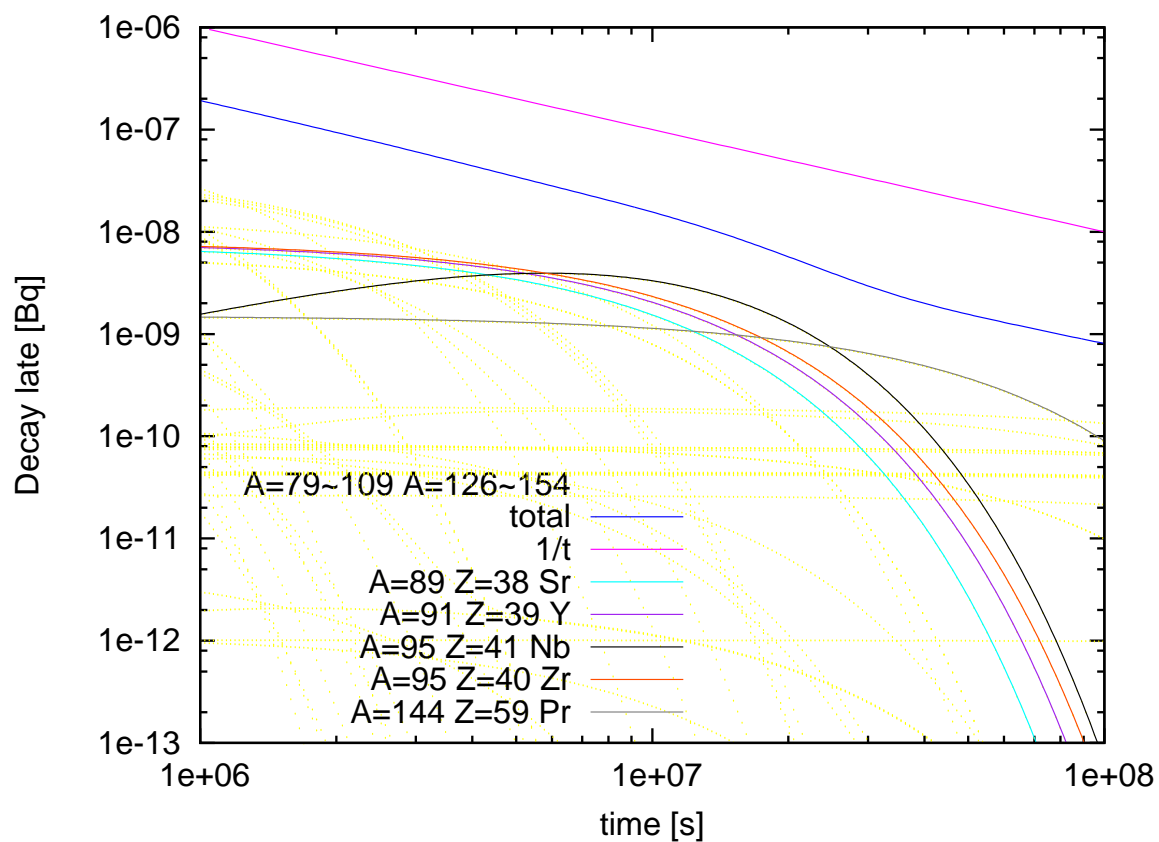


図 5.4: 1 年後に放射能の高い原子核

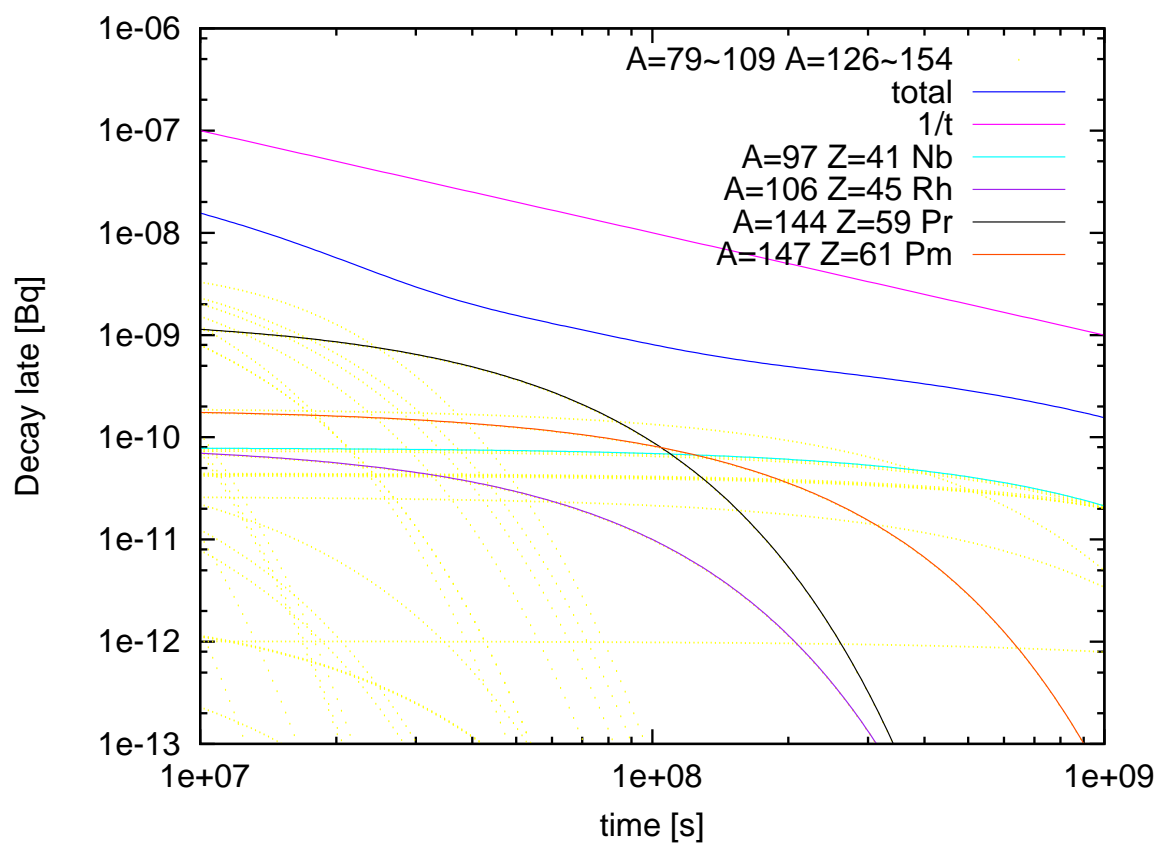


図 5.5: 10 年後に放射能の高い原子核

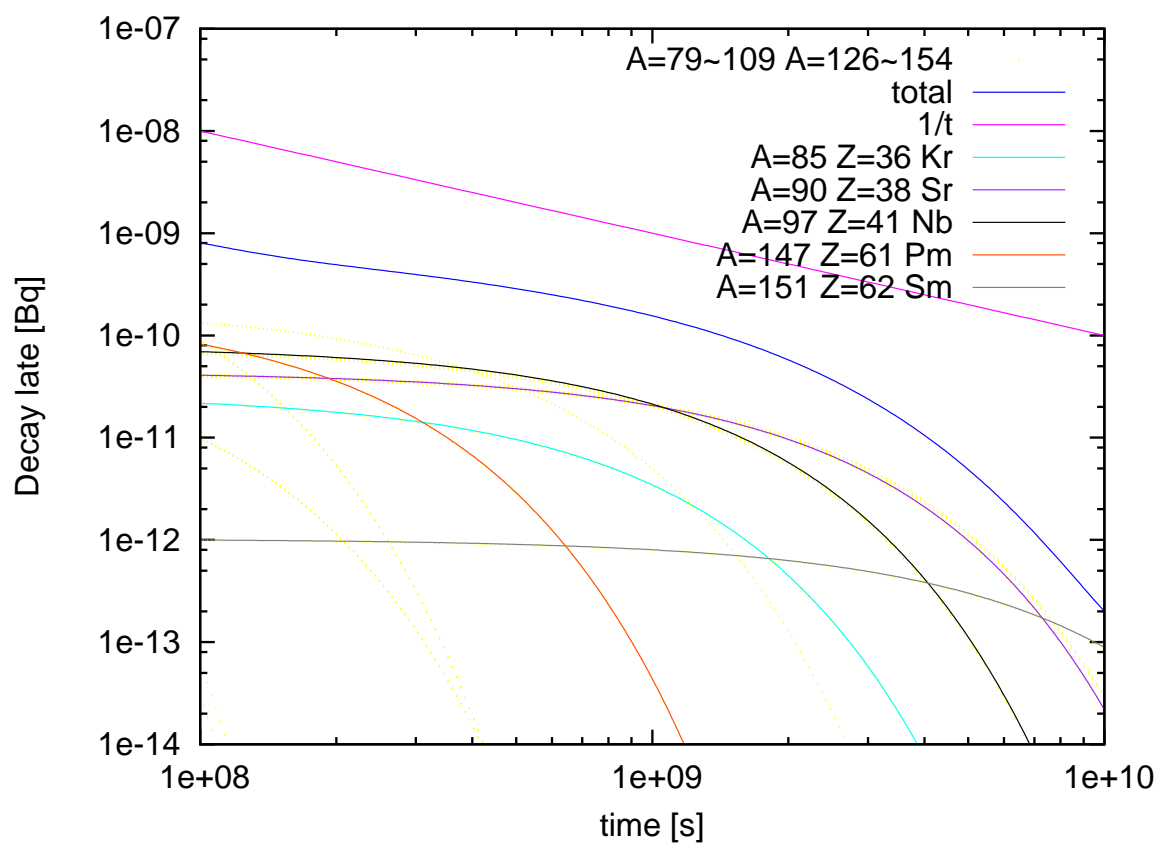


図 5.6: 100 年後に放射能の高い原子核

3枚の図に共通して言えることは高い放射能を持つ原子核は、 ^{235}U 1個が核分裂したときの核分裂生成物の核分裂収率でピークである質量数 95 付近と質量数 140 付近の原子核であることが読み取れる。つまり核分裂によって生成される生成物が多いほど、崩壊する原子核の個数も多くなり、放射能も高くなる。

福島第一原発事故後、問題視された ^{90}Sr 、 ^{131}I 、 ^{137}Cs はこの図のなかに示されていない。これらの原子核よりも放射能が高い原子核が多数存在しており、それらは、大気中に出にくかっただけである。実際大気中に放出されたとされる原子核を次の節で示す。

5.5 原子力安全・保安院が公表した大気中に放出された放射性物質

原子力安全・保安院によって平成 23 年 6 月 6 日に公表された、大気中への放射性物質の放出量を Bq で表した推測データものがある。文献 [12] を参照。その中で、原発事故発生から 1 日後までに最も多く放出されたとする原子核が、私たちが計算した図 5.2 の中でどのような位置にあるかを、図 5.7 に示す。また、それぞれの半減期や Bq 値で試算した放出量を下の表 5.7 に示す。

表 5.1: 大気中に最も放出された放射性物質

	Z	$E(\text{level})(\text{MeV})$	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$	Decay Modes Abundance	becquerel
^{131}I	53	0.0	$7/2+$	-87.4427	8.0252 d (6)	β^- : 100.00 %	1.6×10^{17}
^{132}Te	52	0.0	$0+$	-85.1803	3.204 d (13)	β^- : 100.00 %	8.8×10^{16}
^{133}I	53	0.0	$7/2+$	-85.8865	20.83 h (8)	β^- : 100.00	4.2×10^{16}
^{133}Xe	54	0.0	$3/2+$	-87.6435	5.2475 d (5)	β^- : 100.00 %	1.1×10^{19}
^{137}Cs	55	0.0	$7/2+$	-86.5459	30.08 y (9)	β^- : 100.00 %	1.5×10^{16}

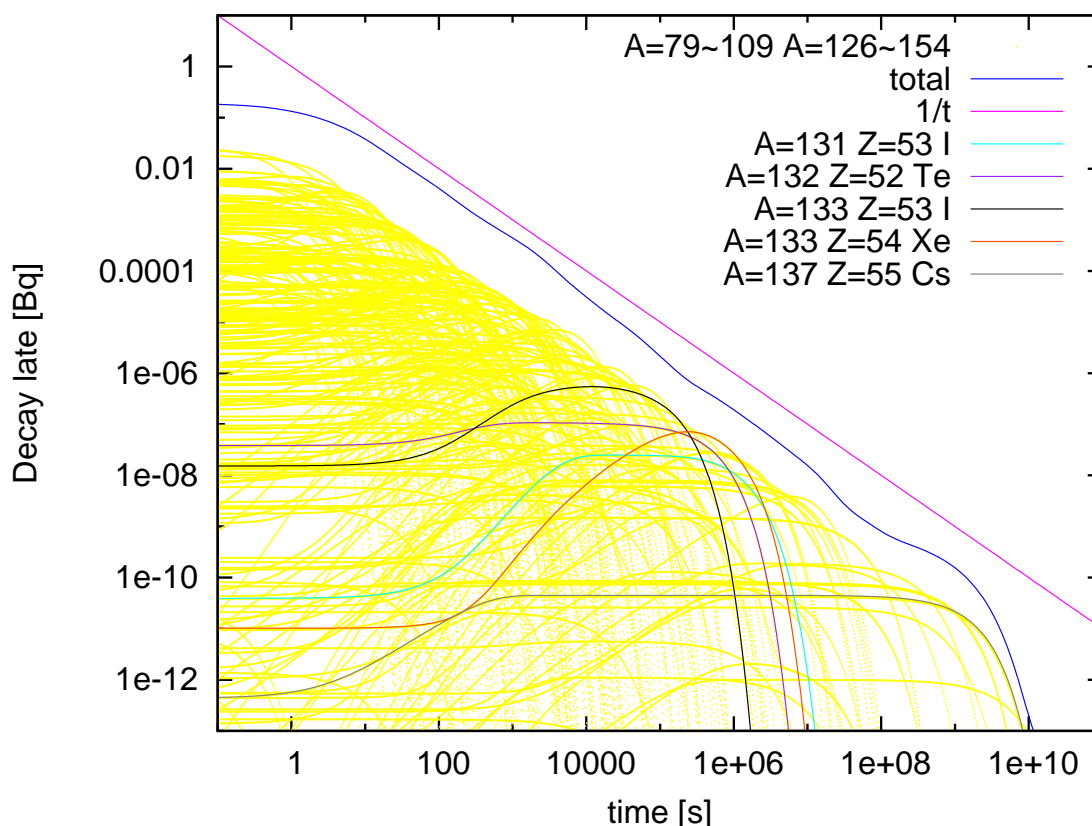


図 5.7: 大気中に最も放出された放射性物質

図 5.7 から、原発事故発生から 1 日後までに最も多く放出されたとする原子核は、私たちの再現した結果の中でも放射能は高いことがわかる。しかし、 ^{137}Cs は放射能が低い値を示しており、 ^{137}Cs よりも高い放射能を持つ原子核が多数存在していることがわかる。このことから、 ^{137}Cs は他の原子核よりも大気中に放出されやすいことがわかる。化学的性質として ^{137}Cs は、沸点が低く揮発性が高いため大気中に放出されやすい。もし、 ^{137}Cs よりも高い放射能を持つ元素が大気中に放出されていたら、現状よりも汚染は深刻になっていたことがわかる。

さて、 ^{134}Cs も注目されている原子核だが、本研究では結果に現れない。その理由は、 ^{235}U , 1 個が核分裂したときの核分裂生成物に ^{134}Cs が含まれていないこと、 ^{134}Cs は両側を安定核に挟まれているため β 崩壊により作られないこと、実際には運転中の原子炉で起きている中性子吸収による ^{134}Cs の生成を私達は考慮していないことである。

第6章 まとめ

本研究では、 β 崩壊系列を表す微分方程式の解析解を求める手順を示した。その解析解の値を自作のプログラムで計算し、 ^{235}U の核分裂生成物の β 崩壊による放射能 [Bq] の時間推移を示した。そのプログラムに入力する半減期や分岐比などのデータを、公的研究機関により WEB 公開されている核データから引用して、表にまとめて付録に掲載した。プログラムも付録に掲載した。このプログラムと核データを用いて、約 300 個の核種の放射能の計算を行った。

すべての計算結果を 1 つの図にまとめ、全放射能の時間推移を両対数グラフに描いて見ると、約 1 秒から約 100 年の時間では傾き -1 の直線でよく近似できることがわかった。その直線は全放射能と時間が反比例していることを示している。このふるまいは、 β 崩壊系列の特徴の下流の核になる (β 安定曲線に近づく) につれて半減期が長くなること、および、上流の核の崩壊先が全て下流の核であることから説明することができた。つまり、 β 崩壊系列の特徴が生み出すのだと結論できた。

さらに、福島第一原発事故によって放出された放射性物質の推量値との比較を行うことで、現実問題との接点を探った。原子力安全・保安院が公表したデータと、自作のプログラムによって計算した結果を比べてみると、 ^{137}Cs 以外の原子核では、生成されてから約 1 日経過までの放射能の値が高いことが一致している。また、高い放射能を持つ原子核は多数あり、それらがもし大気中放出されてならば、現状はもっと深刻になっていたことがわかる。

本研究で、今後改善すべき点は、 β^-n 崩壊の正しい考慮や、分岐比のうち適当に仮定したものの正確な実験データを調べることである。また研究を発展させる方向としては、放射能をベクレル単位でなく、シーベルト単位で表すことが考えられる。さらには、計算の初期値として、核分裂生成物分布を用いるのではなく、運転中の原子炉内の核種分布を用いることが意義深いと思われる。

参考文献

- [1] ニューメリカルレシピ・イン・シー 日本語版 – C 言語による数値計算のレシピ, William H. Press, William T. Vetterling, Saul A. Teukolsky, Brian P. Flannery(著), 丹慶 勝市, 佐藤 俊郎, 奥村 晴彦, 小林 誠 (訳), 技術評論社 (1993 年).
- [2] ローレンス・バークレー国立研究所 原子核科学部門 (Nuclear Science Division)
<http://ie.lbl.gov/fission/235ut.txt>
- [3] National Nuclear Data Center, <http://www.nndc.bnl.gov/chart/>
- [4] JENDL FP decay Data File 2000,
<http://wwwndc.jaea.go.jp/ftpnd/jendl/jfpdecay2000.html>
- [5] 電気学会大学講座 基礎原子力工学, 社団法人 電気学会, オーム社 (1982).
- [6] 原子核はなぜ壊れるのか ~ 放射性崩壊の鍵 ~, 山田 勝美 (著), 丸善株式会社 (1985).
- [7] 東京工業大学「原子核物理学概論」講義ノート, 武藤 一雄,
<http://www.th.phys.titech.ac.jp/muto/>
- [8] 裳華房テキストシリーズ-物理学 原子核物理学, 永江 知文, 永宮 正治 (著), 裳華房 (2000 年).
- [9] The 2003 Atomic Mass Evaluation, <http://amdc.in2p3.fr/web/masseval.html>
- [10] 新物理学シリーズ 15 ベータ崩壊と弱い相互作用, 山田 勝美, 森田 正人, 藤井 昭彦 (著), 培風館 (1974).
- [11] Table of Isotopes, R.B. Firestone, S.Y. Frank Chu, C.M. Baqlin, Wiley (1999).
- [12] 経済産業省 報道発表 平成 23 年 6 月 6 日発表東京電力株式会社福島第一原子力発電所の事故に係る 1 号機、2 号機及び 3 号機の炉心の状態に関する評価について
<http://www.meti.go.jp/press/2011/06/20110606008/20110606008-2.pdf>

謝辞

本論文を作成するにあたり、田嶋直樹先生には終始丁寧なご指導をしていただいたことに感謝し、お礼申し上げます。また林明久先生にも本研究及び日常的なことにおいても、実に丁寧な指導、お世話をしていただきました。同じ研究室の大学院生である杉本宏行氏、谷中裕太氏、大簗陽平氏にも大変為になる助言を頂きました

本研究に対してご意見をいただいた、多くの物理工学科の先生方にもお礼申し上げ、謝辞の言葉とさせていただきます。

付録1 原子核の β 崩壊のデータ

プログラムの入力データとして使用した値を質量数 A 毎に表にまとめて以下に示す。
表の見方：Zは陽子数、Nは中性子数、E(level) はエネルギー準位、 $J\pi$ はパリティ、 Δ はエネルギー、 $T_{1/2}$ は半減期、Decay Modes は崩壊様式を示す。

なお表中にアイソマーと基底状態への分岐比が与えられていない場合には、基底状態へ 100% 分岐すると仮定した場合である。崩壊の分岐先として元素名に数字 1 をつけた記号が書かれている場合は、第 1 番目のアイソマーを意味する。例えば A=79 の表中 ^{79}As の「Decay modes」の欄に「Se1 へ 97.90%分岐」とあるのは ^{79}Se の $\frac{1}{2}^-$ アイソマーへ 97.90%、基底状態 ($\frac{7}{2}^+$) へ 2.10%分岐することを意味している。

原子核の崩壊のデータ

$A = 79$							
	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{79}Cu	29	50	0.0		-41.8990 Syst	188 ms (25)	β^- : 100.00 % β^-n : 55.00 %
^{79}Zn	30	49	0.0	(9/2+)	-53.4322	0.995 s (19)	β^- : 100.00 % β^-n : 1.30 %
							以上を初期値へ繰込
^{79}Ga	31	48	0.0	3/2-	-62.5476	2.847 s (3)	β^- : 100.00 % β^-n : 0.09 % (無視)
^{79}Ge	32	47	0.0	(1/2)-	-69.5266	18.98 s (3)	β^- : 100.00 %
			0.1859	(7/2+)	-69.7125	39.0 s (10)	β^- : 96.00 % IT : 4.00 %
^{79}As	33	46	0.0	3/2-	-73.6364	9.01 m (15)	β^- : 100.00 % (Se1 へ 97.90 %分岐)
^{79}Se	34	45	0.0	7/2+	-75.9173	2.95×10^5 y (38)	β^- : 100.00 %
			0.0958	1/2-	-76.0131	3.92 m (1)	IT : 99.94 % β^- : 0.06 %
^{79}Br	35	44	0.0	3/2-	-76.0684	STABLE (50.69 % (7))	
			0.2076	9/2+	-76.2760	5.1 s (4)	IT : 100.00 %

$A = 80$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{80}Ni	28	52					β^- : 100.00 %(仮定)
^{80}Cu	29	51	0.0		-36.4310 Syst	0.17 s (+11-5)	β^- : 100.00 %(仮定)
^{80}Zn	30	50	0.0	0+	-51.6486	0.54 s (2)	β^- : 100.00 % β^-_{n} : 1.00 %
^{80}Ga	31	49	0.0	3	-59.2236	1.676 s (14)	β^- : 100.00 % β^-_{n} : 0.86 %
							以上を初期値へ繰込
^{80}Ge	32	48	0.0	0+	-69.5353	29.5 s (4)	β^- : 100.00 %
^{80}As	33	47	0.0	1+	-72.1715	15.2 s (2)	β^- : 100.00 %
^{80}Se	34	46	0.0	0+	-77.7598	STABLE (49.61 % (41))	$2\beta^-$ (無視)
^{80}Br	35	45	0.0	1+	-75.8894	17.68 m (2)	β^- : 91.70 % ϵ : 8.30 %
			0.0858	5-	-75.9751	4.4205 h (8)	IT : 100.00 %
^{80}Kr	36	44	0.0	0+	-77.8925	STABLE (2.286 % (10))	

$A = 81$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{81}Zn	30	51	0.0	(5/2+)	-46.1996	304 ms (13)	β^- : 100.00 % β^-_{n} : 7.50 %
^{81}Ga	31	50	0.0	5/2-	-57.6279	1.217 s (5)	β^- : 100.00 % β^-_{n} : 11.90 %
							以上を初期値へ繰込
^{81}Ge	32	49	0.0	(9/2+)	-66.2916	7.6 s (6)	β^- : 100.00 %
			0.6791	(1/2+)	-66.9707	7.6 s (6) (7.61 s と仮定)	β^- : 100.00 %
^{81}As	33	48	0.0	3/2-	-72.5333	33.3 s (8)	β^- : 100.00 % (Se1 \wedge 8.90 %分岐)
^{81}Se	34	47	0.0	1/2-	-76.3894	18.45 m (12)	β^- : 100.00 %
			0.1030	7/2+	-76.4924	57.28 m (2)	IT : 99.95 % β^- : 0.05 %
^{81}Br	35	46	0.0	3/2-	-77.9755	STABLE (49.31 % (7))	

$A = 82$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{82}Cu	29	53	0.0			>636 ns	β^- : 100.00 %(仮定)
^{82}Zn	30	52	0.0	0+	-42.6070 Syst	>150 ns	β^- : 100.00 %(仮定)
^{82}Ga	31	51	0.0	(1,2,3)	-52.9307	0.599 s (2)	β^- : 100.00 % β^-_{n} : 19.80 %
							以上を初期値へ繰込
^{82}Ge	32	50	0.0	0+	-65.4150	4.56 s (26)	β^- : 100.00 %
^{82}As	33	49	0.0	(2-)	-70.1030	19.1 s (5)	β^- : 100.00 %
			0.1470	(5-)	-70.2500	13.6 s (4)	β^- : 100.00 %
^{82}Se	34	48	0.0	0+	-77.5940	STABLE (8.73 % (22))	
^{82}Br	35	45	0.0	5-	-77.4972	35.282 h (7)	β^- : 100.00 %
			0.0459	2-	-77.5431	6.13 m (5)	IT : 97.60 % β^- : 2.40 %
^{82}Kr	36	46	0.0	0+	-80.5902	STABLE (11.593 % (31))	

$A = 83$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{83}Zn	30	53	0.0		-36.7380 Syst	>300 ns	β^- : 100.00 %(仮定)
^{83}Ga	31	52	0.0		-49.2571	308.1 ms (10)	β^- : 100.00 % β^-_{n} : 62.80 %
							以上を初期値へ繰込
^{83}Ge	32	51	0.0	(5/2)+	-60.9764	1.85 s (6)	β^- : 100.00 %
^{83}As	33	50	0.0	(5/2-,3/2-)	-69.6693	13.4 s (3)	β^- : 100.00 % (Se1 \wedge 63.40 %分岐)
^{83}Se	34	49	0.0	9/2+	-75.3406	22.3 m (3)	β^- : 100.00 %
			0.2285	1/2-	-75.5691	70.1 s (4)	β^- : 100.00 %
^{83}Br	35	48	0.0	3/2-	-79.0064	2.40 h (2)	β^- : 100.00 % (Kr1 \wedge 99.98 %分岐)
^{83}Kr	36	47	0.0	9/2+	-79.9900	STABLE (11.500 % (19))	
			0.0416	1/2-	-80.0316	1.85 h (3)	IT : 100.00 %

$A = 84$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{84}Zn	30	54	0.0	0+		>633 ns	β^- : 100.00 %(仮定)
^{84}Ga	31	53	0.0	(0-)	-42.2830 Syst	0.085 s (10)	β^- : 100.00 % β^-n : 74.00 %
^{84}Ge	32	52	0.0	0+	-58.1484	0.954 s (14)	β^- : 100.00 % β^-n : 10.20 %
							以上を初期値へ繰込
^{84}As	33	51	0.0	(3-)	-65.8535	4.2 s (5)	β^- : 100.00 % β^-n : 0.18 %(無視)
^{84}Se	34	50	0.0	0+	-75.9476	3.26 m (10)	β^- : 100.00 %
^{84}Br	35	49	0.0	2-	-77.7893	31.76 m (8)	β^- : 100.00 %
			0.3200	(6)-	-78.1093	6.0 m (2)	β^- : 100.00 %
^{84}Kr	36	48	0.0	0+	-82.4393	STABLE (56.987 % (15))	

$A = 85$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{85}Ga	31	54	0.0	(1/2-,3/2-)	-40.2310 Syst	<100 ms	β^- : 100.00 %(仮定) β^-n > 35.0 %(仮定)
^{85}Ge	32	53	0.0	(1/2+,5/2+)	-53.1234	0.56 s (5)	β^- : 100.00 % β^-n : 14.00 %
^{85}As	33	52	0.0	(3/2-)	-63.1891	2.021 s (10)	β^- : 100.00 % β^-n : 59.40 %
^{85}Se	34	51	0.0	(5/2+)	-72.4136	32.9 s (3)	β^- : 100.00 %
							β^- : 100.00 %(仮定) β^-n : 23.00 %(仮定)
							以上を初期値へ繰込
^{85}Br	35	50	0.0	3/2-	-78.5754	2.90 m (6)	β^- : 100.00 % (Kr1 \wedge 0.18 %分岐)
^{85}Kr	36	49	0.0	9/2+	-81.4803	10.752 y (25)	β^- : 100.00 %
			0.3050	1/2-	-81.7853	4.480 h (8)	β^- : 78.60 % IT : 21.40 % (IT : 100.00 % と仮定)
^{85}Rb	37	48	0.0	5/2-	-82.1673	STABLE (72.17 % (2))	

$A = 86$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{86}Zn	30	56					β^- : 100.00 %(仮定)
^{86}Ga	31	55	0.0		-34.4560 Syst	>150 ns	β^- : 100.00 %(仮定)
^{86}Ge	32	54	0.0	0+	-49.7600 Syst	>150 ns	β^- : 100.00 %(仮定)
^{86}As	33	53	0.0		-58.9621	0.945 s (8)	β^- : 100.00 % β^-n : 26.00 %
^{86}Se	34	52	0.0	0+	-70.5031	14.3 s (3)	β^- : 100.00 %
							β^- : 100.00 %(仮定) β^-n : 12.00 %(仮定)
							以上を初期値へ繰込
^{86}Br	35	51	0.0	(1-)	-75.6322	55.1 s (4)	β^- : 100.00 %
^{86}Kr	36	50	0.0	0+	-83.2656	STABLE (17.279 % (41))	
^{86}Rb	37	49	0.0	2-	-82.7470	18.642 d (18)	β^- : 99.99 % ϵ : 5.2×10^{-3} %(無視)
			0.5561	6-	-83.3031	1.017 m (3)	IT : 100.00 % β^- < 0.30 %(無視)
^{86}Sr	38	48	0.0	0+	-84.5232	STABLE (9.86 % (1))	

$A = 87$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{87}Ge	32	55	0.0	(5/2+)	-44.2370 Syst	≈ 0.14 s	β^- : 100.00 %(仮定)
^{87}As	33	54	0.0	(3/2-)	-55.6179	0.56 s (8)	β^- : 100.00 % β^-n : 15.40 %
^{87}Se	34	53	0.0	(5/2+)	-66.4261	5.50 s (12)	β^- : 100.00 % β^-n : 0.20 %
^{87}Br	35	52	0.0	3/2-	-73.8916	55.65 s (13)	β^- : 100.00 % β^-n : 2.60 %
							以上を初期値へ繰込
^{87}Kr	36	51	0.0	5/2+	-80.7095	76.3 m (5)	β^- : 100.00 %
^{87}Rb	37	50	0.0	3/2-	-84.5977	4.81×10^{10} y (9) (27.83 % (2))	β^- : 100.00 %
^{87}Sr	38	49	0.0	9/2+	-84.8800	STABLE (7.00 % (1))	
			0.3885	1/2-	-85.2685	2.815 h (12)	IT : 99.70 % ϵ : 0.30 %

$A = 88$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{88}Ge	32	56	0.0	0+	-40.2410 Syst	≥ 300 ns	β^- : 100.00 %
^{88}As	33	55	0.0		-50.8880 Syst	> 300 ns	β^- : 100.00 %
^{88}Se	34	54	0.0	0+	-63.8841	1.53 s (6)	β^- : 100.00 % β^-n : 0.67 %
^{88}Br	35	53	0.0	(2-)	-70.7159	16.29 s (6)	β^- : 100.00 % β^-n : 6.58 %
							以上を初期値へ繰込
^{88}Kr	36	52	0.0	0+	-79.6912	2.84 h (3)	β^- : 100.00 %
^{88}Rb	37	51	0.0	2-	-82.6089	17.773 m (11)	β^- : 100.00 %
^{88}Sr	38	50	0.0	0+	-87.9213	STABLE (82.58 % (1))	

$A = 89$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{89}As	33	56	0.0		-46.9380 Syst	> 300 ns	β^- : 100.00 % (仮定)
^{89}Se	34	55	0.0	(5/2+)	-58.9923	0.41 s (4)	β^- : 100.00 % β^-n : 7.80 %
^{89}Br	35	54	0.0	(3/2-, 5/2-)	-68.2742	4.40 s (3)	β^- : 100.00 % β^-n : 13.80 %
							以上を初期値へ繰込
^{89}Kr	36	53	0.0	3/2(+)	-76.5357	3.15 m (4)	β^- : 100.00 %
^{89}Rb	37	52	0.0	3/2-	-81.7122	15.15 m (12)	β^- : 100.00 %
^{89}Sr	38	51	0.0	5/2+	-86.2087	50.53 d (7)	β^- : 100.00 % (Y1 \sim 0.015 % 分岐)
^{89}Y	39	50	0.0	1/2-	-87.7096	STABLE (100 %)	
			0.9090	9/2+	-88.6186	15.663 s (5)	IT : 100.00 %

$A = 90$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{90}Se	34	56	0.0	0+	-55.9270 Syst	>300 ns	β^- : 100.00 %(仮定)
^{90}Br	35	55	0.0		-64.0003	1.91 s (1)	β^- : 100.00 % β^-n : 25.20 %
							以上を初期値へ繰込
^{90}Kr	36	54	0.0	0+	-74.9592	32.32 s (9)	β^- : 100.00 % (Rb1 \wedge 12.20 %分岐)
^{90}Rb	37	53	0.0	0-	-79.3649	158 s (5)	β^- : 100.00 %
			0.1069	3-	-79.4718	258 s (4)	β^- : 97.40 % IT : 2.60 %
^{90}Sr	38	52	0.0	0+	-85.9493	28.90 y (3)	β^- : 100.00 %
^{90}Y	39	51	0.0	2-	-86.4953	64.053 h (20)	β^- : 100.00 %
			0.6817	7+	-87.1770	3.19 h (6)	IT : 100.00 % β^- : 1.8×10^{-3} %(無視)
^{90}Zr	40	50	0.0	0+	-88.7742	STABLE (51.45 % (40))	
			2.3190	5-	-91.0932	809.2 ms (20)	IT : 100.00 %

$A = 91$

^{91}Se	34	57	0.0		-50.3380 Syst	0.27 s (5)	β^- : 100.00 % β^-n : 21.00 %
^{91}Br	35	56	0.0		-61.1072	0.541 s (5)	β^- : 100.00 % β^-n : 20.00 %
							以上を初期値へ繰込
^{91}Kr	36	55	0.0	5/2(+)	-70.9739	8.57 s (4)	β^- : 100.00 %
^{91}Rb	37	54	0.0	3/2(-)	-77.7464	58.4 s (4)	β^- : 100.00 %
^{91}Sr	38	53	0.0	5/2+	-83.6528	9.63 h (5)	β^- : 100.00 % (Y1 \wedge 89.00 %分岐)
^{91}Y	39	52	0.0	1/2-	-86.3529	58.51 d (6)	β^- : 100.00 %
			0.5556	9/2+	-86.9085	49.71 m (4)	IT : 100.00 % β^- < 1.50 %(無視)
^{91}Zr	40	51	0.0	5/2+	-87.8973	STABLE (11.22 % (5))	

$A = 92$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{92}Se	34	58	0.0	0+	-46.7240 Syst		β^- : 100.00 %
^{92}Br	35	57	0.0	(2-)	-56.2328	0.343 s (15)	β^- : 100.00 % β^-n : 33.10 %
							以上を初期値へ繰込
^{92}Kr	36	56	0.0	0+	-68.7693	1.840 s (8)	β^- : 100.00 % β^-n : 0.03 % (無視)
^{92}Rb	37	55	0.0	0-	-74.7726	4.492 s (20)	β^- : 100.00 % β^-n : 0.01 % (無視)
^{92}Sr	38	54	0.0	0+	-82.8674	2.66 h (4)	β^- : 100.00 %
^{92}Y	39	53	0.0	2-	-84.8179	3.54 h (1)	β^- : 100.00 %
^{92}Zr	40	52	0.0	0+	-88.4607	STABLE (17.15 % (8))	

$A = 93$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{93}Br	35	58	0.0	(5/2-)	-52.8530 Syst	102 ms (10)	β^- : 100.00 % β^-n : 68.00 %
^{93}Kr	36	57	0.0	1/2+	-64.1359	1.286 s (10)	β^- : 100.00 % β^-n : 1.95 %
^{93}Rb	37	56	0.0	5/2-	-72.6200	5.84 s (2)	β^- : 100.00 % β^-n : 1.39 %
							以上を初期値へ繰込
^{93}Sr	38	55	0.0	5/2+	-80.0860	7.43 m (3)	β^- : 100.00 %
^{93}Y	39	54	0.0	1/2-	-84.2285	10.18 h (8)	β^- : 100.00 %
			0.7587	(9/2)+	-84.9872	0.82 s (4)	IT : 100.00 %
^{93}Zr	40	53	0.0	5/2+	-87.1238	1.61×10^6 y (5)	β^- : 100.00 % (Nb1 \rightarrow 95.00 %分岐)
^{93}Nb	41	52	0.0	9/2+	-87.2142	STABLE (100 %)	
			0.0308	1/2-	-87.2450	16.12 y (12)	IT : 100.00 %

$A = 94$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{94}Se	34	60	0.0	0+	-36.8030 Syst	>150 ns	β^- : 100.00 %
^{94}Br	35	59	0.0		-47.5990 Syst	70 ms (20)	β^- : 100.00 % β^-n : 68.00 %
^{94}Kr	36	58	0.0	0+	-61.3477	212 ms (5)	β^- : 100.00 % β^-n : 1.11 %
^{94}Rb	37	57	0.0	3(-)	-68.5619	2.702 s (5)	β^- : 100.00 % β^-n : 10.50 %
							以上を初期値へ繰込
^{94}Sr	38	56	0.0	0+	-78.8431	75.3 s (2)	β^- : 100.00 %
^{94}Y	39	55	0.0	2-	-82.3529	18.7 m (1)	β^- : 100.00 %
^{94}Zr	40	54	0.0	0+	-87.2725	STABLE (17.38 % (28))	
^{94}Nb	41	53	0.0	6+	-86.3704	2.03×10^4 y (16)	β^- : 100.00 %
			0.0409	3+	-86.4113	6.263 m (4)	IT : 99.50 % β^- : 0.50 %
^{94}Mo	42	52	0.0	0+	-88.4141	STABLE (9.15 % (9))	

$A = 95$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{95}Kr	36	59	0.0	1/2(+)	-56.1589	0.114 s (3)	β^- : 100.00 % β^-n : 2.87 %
^{95}Rb	37	58	0.0	5/2-	-65.8944	377.7 ms (8)	β^- : 100.00 % β^-n : 8.70 %
							以上を初期値へ繰込
^{95}Sr	38	57	0.0	1/2+	-75.1238	23.90 s (14)	β^- : 100.00 %
^{95}Y	39	56	0.0	1/2-	-81.2131	10.3 m (1)	β^- : 100.00 %
^{95}Zr	40	55	0.0	5/2+	-85.6633	64.032 d (6)	β^- : 100.00 % (Nb1 \rightarrow 0.90 %分岐)
^{95}Nb	41	54	0.0	9/2+	-86.7863	34.991 d (6)	β^- : 100.00 %
			0.2357	1/2-	-87.0220	3.61 d (3)	IT : 94.40 % β^- : 5.60 %
^{95}Mo	42	53	0.0	5/2+	-87.7119	STABLE (15.84 % (11))	

$A = 96$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{96}Se	34	62					β^- : 100.00 %(假定)
^{96}Br	35	61	0.0		-38.3500 Syst	≥ 150 ns	β^- : 100.00 % β^-_{n} : 27.60 %
^{96}Kr	36	60	0.0	0+	-53.0796	80 ms (6)	β^- : 100.00 % β^-_{n} : 3.70 %
^{96}Rb	37	59	0.0	2(-)	-61.3543	203 ms (3)	β^- : 100.00 % β^-_{n} : 13.30 %
							以上を初期値へ繰込
^{96}Sr	38	58	0.0	0+	-72.9329	1.07 s (1)	β^- : 100.00 %
^{96}Y	39	57	0.0	0-	-78.3447	5.34 s (5)	β^- : 100.00 %
			1.1400	8+	-79.4847	9.6 s (2)	β^- : 100.00 %
^{96}Zr	40	56	0.0	0+	-85.4477	2.35×10^{19} y (21) (2.80 % (9))	$2\beta^-$ (無視)
^{96}Nb	41	55	0.0	6+	-85.6081	23.35 h (5)	β^- : 100.00 %
^{96}Mo	42	54	0.0	0+	-88.7949	STABLE (16.67 % (15))	

$A = 97$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{97}Kr	36	61	0.0	(3/2+)	-47.4234	63 ms (4)	β^- : 100.00 % β^-_{n} : 6.70 %
^{97}Rb	37	60	0.0	3/2+	-58.5182	169.1 ms (6)	β^- : 100.00 % β^-_{n} : 25.50 %
							以上を初期値へ繰込
^{97}Sr	38	59	0.0	1/2+	-68.5916	429 ms (5)	β^- : 100.00 % $\beta^-_{\text{n}} \leq 0.05$ %(無視)
^{97}Y	39	58	0.0	(1/2-)	-76.1303	3.75 s (3)	β^- : 100.00 % β^-_{n} : 0.06 %(無視)
			0.6675	(9/2)+	-76.7978	1.17 s (3)	$\beta^- > 99.30$ % IT < 0.70 %(無視) $\beta^-_{\text{n}} < 0.08$ %(無視)
			3.5226	(27/2-)	-79.6529	142 ms (8)	IT : 98.40 % β^- : 1.60 %
^{97}Zr	40	57	0.0	1/2+	-82.9515	16.749 h (8)	β^- : 100.00 % (Nb1 \wedge 95.00 %分岐)
^{97}Nb	41	55	0.0	9/2+	-85.6103	72.1 m (7)	β^- : 100.00 %
			0.7434	1/2-	-86.3537	58.7 s (18)	IT : 100.00 %
^{97}Mo	42	55	0.0	5/2+	-87.5448	STABLE (9.60 % (14))	

$$A = 98$$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{98}Kr	36	62	0.0	0+	-44.4600 Syst	46 ms (8)	β^- : 100.00 % β^-n : 7.00 %
^{98}Rb	37	61	0.0	(0,1)	-54.0309	102 ms (4)	β^- : 100.00 % β^-n : 13.80 % β^-2n : 0.05 %(無視)
			0.2700	(3,4)	-54.3009	96 ms (3)	β^- : 100.00 %
^{98}Sr	38	60	0.0	0+	-66.4366	0.653 s (2)	β^- : 100.00 % β^-n : 0.25 %
^{98}Y	39	59	0.0	(0)-	-72.3037	0.548 s (2)	β^- : 100.00 % β^-n : 0.33 %
			0.4100	(4,5)	-72.7137	2.0 s (2)	β^- > 80.00 % IT < 20.00 %(無視) β^-n : 3.40 %
							以上を初期値へ繰込
^{98}Zr	40	58	0.0	0+	-81.2956	30.7 s (4)	β^- : 100.00 %
^{98}Nb	41	57	0.0	1+	-83.5334	2.86 s (6)	β^- : 100.00 %
			0.0840	(5+)	-83.6174	51.3 m (4)	β^- : 99.90 % (β^- : 100.00 % と仮定) IT < 0.20 %(無視)
^{98}Mo	42	56	0.0	0+	-88.1161	STABLE (24.39 % (37))	
^{98}Tc	43	55	0.0	(6)+	-86.4318	4.2×10^6 y (3)	β^- : 100.00 %
^{98}Ru	44	54	0.0	0+	-88.2248	STABLE (1.87 % (3))	

$A = 99$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{99}Rb	37	62	0.0	(5/2+)	-51.2203	54 ms (4)	β^- : 100.00 % β^-_{n} : 15.80 %
^{99}Sr	38	61	0.0	3/2+	-62.5295	0.269 s (1)	β^- : 100.00 % β^-_{n} : 0.10 %
^{99}Y	39	60	0.0	(5/2+)	-70.6589	1.484 s (7)	β^- : 100.00 % β^-_{n} : 1.70 %
							以上を初期値へ繰込
^{99}Zr	40	59	0.0	(1/2+)	-77.6265	2.1 s (1)	β^- : 100.00 % (Nb1 \wedge 36.00 %分岐)
^{99}Nb	41	58	0.0	9/2+	-82.3301	15.0 s (2)	β^- : 100.00 %
			0.3653	1/2-	-82.6954	2.5 m (2)	β^- > 96.20 % IT < 3.80 % (β^- : 100.00 % と仮定)
^{99}Mo	42	57	0.0	1/2+	-85.9702	65.976 h (24)	β^- : 100.00 % (Tc1 \wedge 87.00% 分岐)
^{99}Tc	43	56	0.0	9/2+	-87.3271	2.111×10^5 y (12)	β^- : 100.00 %
			0.1427	1/2-	-87.4698	6.0067 y (5)	IT : 100.00 % β^- : 3.7×10^{-3} % (無視)
^{99}Ru	44	55	0.0	5/2+	-87.6202	STABLE (12.76 % (14))	

$A = 100$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{100}Kr	36	64	0.0	0+	-35.1640 Syst	7 ms (+11-3)	β^- : 100.00 % β^-_{n}
^{100}Rb	37	63	0.0	(3+,4-)	-46.5470 Syst	51 ms (8)	β^- : 100.00 % β^-_{n} : 6.00 % $\beta^-_{2\text{n}}$: 0.16 %
^{100}Sr	38	62	0.0	0+	-59.8333	202 ms (3)	β^- : 100.00 % β^-_{n} : 0.78 %
^{100}Y	39	61	0.0	1-,2-	-67.3365	735 ms (7)	β^- : 100.00 % β^-_{n} : 0.92 %
			0.1450	(3,4,5)	-67.4815	0.94 s (3)	β^- : 100.00 %
							以上を初期値へ繰込
^{100}Zr	40	60	0.0	0+	-76.3844	7.1 s (4)	β^- : 100.00 % (Nb1 \wedge 2.00 %分岐)
^{100}Nb	41	59	0.0	1+	-79.8065	1.5 s (2)	β^- : 100.00 %
			0.3140	(5+)	-80.1205	2.99 s (11)	β^- : 100.00 %
^{100}Mo	42	58	0.0	0+	-86.1878	7.3×10^{18} y (4) (9.82 % (31))	$2\beta^-$: 100.00 % (無視)
^{100}Tc	43	57	0.0	1+	-86.0202	15.46 s (19)	β^- : 100.00 % ϵ : 2.6×10^{-3} % (無視)
^{100}Ru	44	56	0.0	0+	-89.2222	STABLE (12.60 % (7))	

$A = 101$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{101}Rb	37	64	0.0	(3/2+)	−42.9750 Syst	32 ms (5)	β^- : 100.00 % β^-_{n} : 28.00 %
^{101}Sr	38	63	0.0	(5/2-)	−55.5652	118 ms (3)	β^- : 100.00 % β^-_{n} : 2.37 %
^{101}Y	39	62	0.0	(5/2+)	−65.0702	0.45 s (2)	β^- : 100.00 % β^-_{n} : 1.94 %
							以上を初期値へ繰込
^{101}Zr	40	61	0.0	(3/2+)	−73.1733	2.3 s (1)	β^- : 100.00 %
^{101}Nb	41	60	0.0	(5/2+)	−78.8863	7.1 s (3)	β^- : 100.00 %
^{101}Mo	42	59	0.0	1/2+	−83.5147	14.61 m (3)	β^- : 100.00 %
^{101}Tc	43	58	0.0	9/2+	−86.3390	14.02 m (1)	β^- : 100.00 %
^{101}Ru	44	57	0.0	5/2+	−87.9529	STABLE (17.06 % (2))	
^{101}Rh	45	56	0.0	1/2-	−87.4117	3.3 y (3)	ϵ : 100.00 %
			0.1573	9/2+	−87.5690	4.34 d (1)	ϵ : 92.80 % IT : 7.20 %

$A = 102$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{102}Rb	37	65	0.0		−37.8930 Syst	37 ms (3)	β^- : 100.00 % β^-_{n} : 18.00 %
^{102}Sr	38	64	0.0	0+	−52.4000 Syst	69 ms (6)	β^- : 100.00 % β^-_{n} : 5.50 %
^{102}Y	39	63	0.0	HighJ	−61.2150 Syst	0.36 s (4)	β^- : 100.00 % β^-_{n} : 4.90 %
			0.0	LowJ	−61.2150 Syst	0.298 s (9)	β^- : 100.00 % β^-_{n} : 4.90 %
							以上を初期値へ繰込
^{102}Zr	40	62	0.0	0+	−71.5954	2.9 s (2)	β^- : 100.00 %
^{102}Nb	41	61	0.0	(4+)	−76.3139	4.3 s (4)	β^- : 100.00 %
			0.0	1+	−76.3139	1.3 s (2)	β^- : 100.00 %
^{102}Mo	42	60	0.0	0+	−83.5729	11.3 m (2)	β^- : 100.00 %
^{102}Tc	43	59	0.0	1+	−84.5691	5.28 s (15)	β^- : 100.00 %
			0.0	(4,5)	−84.5691	4.35 m (7)	β^- : 98.00 % IT : 2.00 %
^{102}Ru	44	58	0.0	0+	−89.1012	STABLE (31.55 % (14))	
^{102}Rh	45	57	0.0	(1-,2-)	−86.7783	207.3 d (17)	ϵ : 78.00 % β^- : 22.00 %
			0.1407	6(+)	−86.9190	3.742 y (10)	ϵ : 99.77 % IT : 0.23 %
^{102}Pd	46	56	0.0	0+	−87.9286	STABLE (1.02 % (1))	

$A = 103$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{103}Y	39	64	0.0	(5/2+)	-58.5006	0.23 s (2)	β^- : 100.00 % β^-n : 8.00 %
							以上を初期値へ繰込
^{103}Zr	40	63	0.0	(5/2-)	-67.8247	1.32 s (11)	β^- : 100.00 % $\beta^-n \leq 1.00$ % (無視)
^{103}Nb	41	62	0.0	(5/2+)	-75.0234	1.5 s (2)	β^- : 100.00 %
^{103}Mo	42	61	0.0	(3/2+)	-80.9700	67.5 s (15)	β^- : 100.00 %
^{103}Tc	43	60	0.0	5/2+	-84.6004	54.2 s (8)	β^- : 100.00 %
^{103}Ru	44	59	0.0	3/2+	-87.2620	39.247 d (13)	β^- : 100.00 % (Rh1 \sim 99.75 % 分岐)
^{103}Rh	45	58	0.0	1/2-	-88.0256	STABLE (100 %)	
			0.0398	7/2+	-88.0654	56.114 m (9)	IT : 100.00 %

$A = 104$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{104}Sr	38	66	0.0	0+	-43.9200 Syst	43 ms (+9-7)	β^- : 100.00 % (仮定)
^{104}Y	39	65	0.0		-54.0640 Syst	197 ms (4)	β^- : 100.00 % β^-n
							以上を初期値へ繰込
^{104}Zr	40	64	0.0	0+	-65.7335	0.87 s (6)	β^- : 100.00 % $\beta^-n \leq 1.00$ % (無視)
^{104}Nb	41	63	0.0	(1+)	-71.8285	4.9 s (3)	β^- : 100.00 % β^-n : 0.06 % (無視)
			0.2150		-72.0435	0.94 s (4)	β^- : 100.00 % β^-n : 0.05 % (無視)
^{104}Mo	42	62	0.0	0+	-80.3591	60 s (2)	β^- : 100.00 %
^{104}Tc	43	61	0.0	(3+)	-82.5091	18.3 m (3)	β^- : 100.00 %
^{104}Ru	44	60	0.0	0+	-88.0923	STABLE (18.62 % (27))	
^{104}Rh	45	59	0.0	1+	-86.9532	42.3 s (4)	β^- : 99.55 % ϵ : 0.45 %
			0.1290	5+	-87.0822	4.34 m (3)	IT : 99.87 % β^- : 0.13 %
^{104}Pd	46	58	0.0	0+	-89.3932	STABLE (11.14 % (8))	

$A = 105$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{105}Zr	40	65	0.0		-61.4743	0.66 s (7)	β^- : 100.00 % $\beta^-n \leq 2.00$ %
^{105}Nb	41	64	0.0	(5/2+)	-69.9102	2.95 s (6)	β^- : 100.00 % β^-n : 1.70 %
							以上を初期値へ繰込
^{105}Mo	42	63	0.0	(5/2-)	-77.3465	35.6 s (16)	β^- : 100.00 %
^{105}Tc	43	62	0.0	(3/2-)	-82.2943	7.6 m (1)	β^- : 100.00 %
^{105}Ru	44	60	0.0	3/2+	-85.9311	4.44 h (2)	β^- : 100.00 %
^{105}Rh	45	59	0.0	7/2+	-87.8489	35.36 h (6)	β^- : 100.00 % (Rh1 \rightarrow 28.50 % 分岐)
			0.1298	1/2-	-87.9787	42.9 s (3)	IT : 100.00 %
^{105}Pd	46	59	0.0	5/2+	-88.4160	STABLE (22.33 % (8))	

$A = 106$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{106}Zr	40	66	0.0	0+	-59.0380 Syst	191 ms (19)	β^- : 100.00 % $\beta^-n \leq 7.00$ %
^{106}Nb	41	65	0.0		-66.1979	0.93 s (4)	β^- : 100.00 % β^-n : 4.50 %
							以上を初期値へ繰込
^{106}Mo	42	64	0.0	0+	-76.1441	8.73 s (12)	β^- : 100.00 %
^{106}Tc	43	63	0.0	(2+)	-79.7737	35.6 s (6)	β^- : 100.00 %
^{106}Ru	44	62	0.0	0+	-86.3207	371.8 d (18)	β^- : 100.00 %
^{106}Rh	45	61	0.0	1+	-86.3601	30.07 s (35)	β^- : 100.00 %
			0.1370	(6)+	-86.4971	131 m (2)	β^- : 100.00 %
^{106}Pd	46	60	0.0	0+	-89.9056	STABLE (27.33 % (3))	

$A = 107$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{107}Nb	41	66	0.0		-63.7183	300 ms (9)	β^- : 100.00 % β^-n : 8.00 %
							以上を初期値へ繰込
^{107}Mo	42	65	0.0	(5/2+)	-72.5613	3.5 s (5)	β^- : 100.00 %
^{107}Tc	43	64	0.0	(3/2-)	-78.7464	21.2 s (2)	β^- : 100.00 %
^{107}Ru	44	63	0.0	(5/2)+	-83.8590	3.75 m (5)	β^- : 100.00 %
^{107}Rh	45	62	0.0	7/2+	-86.8606	21.7 m (4)	β^- : 100.00 %
^{107}Pd	46	61	0.0	5/2+	-88.3708	6.5×10^6 y (3)	β^- : 100.00 %
			0.2146	11/2-	-88.5854	21.3 s (5)	IT : 100.00 %
^{107}Ag	47	60	0.0	1/2-	-88.4055	STABLE (51.839 % (8))	
			0.0931	7/2+	-88.4986	44.3 s (2)	IT : 100.00 %

$A = 108$

	Z	N	E(level)(MeV)	$J\pi$	$\Delta(\text{MeV})$	$T_{1/2}$ Abundance	Decay Modes
^{108}Zr	40	68	0.0	0+	-51.3530 Syst	73 ms (4)	β^- : 100.00 % β^-n
^{108}Nb	41	67	0.0	(2+)	-59.5643	220 ms (18)	β^- : 100.00 % β^-n : 8.00 %
							以上を初期値へ繰込
^{108}Mo	42	66	0.0	0+	-70.7657	1.09 s (2)	β^- : 100.00 % $\beta^-n < 0.50$ % (無視)
^{108}Tc	43	65	0.0	(2)+	-75.9193	5.17 s (7)	β^- : 100.00 %
^{108}Ru	44	64	0.0	0+	-83.6578	4.55 m (5)	β^- : 100.00 %
^{108}Rh	45	63	0.0	1+	-85.0315	16.8 s (5)	β^- : 100.00 %
			0.0	(5+)	-85.0315	6.0 m (3)	β^- : 100.00 % IT
^{108}Pd	46	62	0.0	0+	-89.5211	STABLE (26.46 % (9))	
^{108}Ag	47	61	0.0	1+	-87.6056	2.382 m (11)	β^- : 97.15 % ϵ : 2.85 %
			0.1095	6+	-87.7151	438 y (9)	ϵ : 91.30 % IT : 8.70 %
^{108}Cd	48	60	0.0	0+	-89.2524	$> 1.9 \times 10^{18}$ y (0.89 % (3))	2ϵ (無視)

$A = 109$

	Z	N	E(level)(MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$ Abundance	Decay Modes
^{109}Nb	41	68	0.0	(5/2)	-56.7840 Syst	106 ms (9)	β^- : 100.00 % $\beta^-n < 15.00$ %(無視)
^{109}Mo	42	67	0.0	(7/2-)	-66.6759	660 ms (45)	β^- : 100.00 % β^-n : 1.30 %
							以上を初期値へ繰込
^{109}Tc	43	66	0.0	(5/2+)	-74.2792	0.86 s (4)	β^- : 100.00 % β^-n : 0.08 %(無視)
^{109}Ru	44	65	0.0	(5/2+)	-80.7348	34.5 s (10)	β^- : 100.00 %
^{109}Rh	45	64	0.0	7/2+	-85.0106	80 s (2)	β^- : 100.00 %
							(IT : 100.00 % と仮定)
^{109}Pd	46	63	0.0	5/2+	-87.6034	13.7012 h (24)	β^- : 100.00 % (Ag1 \wedge 99.95 % 分岐)
			0.1890	11/2-	-87.7924	4.696 m (3)	IT : 100.00 %
^{109}Ag	47	62	0.0	1/2-	-88.7195	STABLE (48.161 % (8))	
			0.0880	7/2+	-88.8075	39.6 s (2)	IT : 100.00 %

$A = 126$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{126}Ag	47	79	0.0		-60.8550Syst	107 ms (12)	β^- : 100.00 %
^{126}Cd	48	78	0.0	0+	-72.2564	0.515 s (17)	β^- : 100.00 % $\rightarrow 100\%$ 3(+) \wedge
^{126}In	49	77	0.0	3(+)	-77.8137	1.53 s (1)	β^- : 100.00 %
			0.1020	(8-)	-77.9157	1.64 s (5)	β^- : 100.00 %
^{126}Sn	50	76	0.0	0+	-86.0207	2.30E+5 y(14)	β^- : 100.00 % $\rightarrow 100\%$ (8-) \wedge
^{126}Sb	51	75	0.0	(8-)	-86.3987	12.35 d (6)	β^- : 100.00 %
			0.0177	(5+)	-86.4164	19.15 m (8)	β^- : 86.00 % IT : 14.00 %
			0.0404	(3-)	-86.4391	≈ 11 s	IT : 100.00 % \rightarrow (無視)
^{126}Te	52	74	0.0	0+	-90.0666	Abundance 18.84 % (25)	
^{126}I	53	73	0.0	2-	-87.9122	12.93 d (5)	ε : 52.70 % β^- : 47.30 %
^{126}Xe	54	72	0.0	0+	-89.1462	Abundance 0.0890 % (2)	

$A = 127$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{127}Ag	47	80	0.0		-58.7680Syst	109 ms (25)	$\beta^- : 100.00 \%$
^{127}Cd	48	79	0.0	(3/2+)	-68.4354	0.37 s (7)	$\beta^- : 100.00 \%$ $\rightarrow 100\% (9/2+) \wedge$
^{127}In	49	78	0.0	(9/2+)	-76.8920	1.09 s (1)	$\beta^- : 100.00 \%$ $\rightarrow 50\% (11/2-) \wedge (\text{仮定})$ $\rightarrow 50\% 3/2+ \wedge (\text{仮定})$ $\beta^- n \leq 0.03 \%$
			0.4620	(1/2-)	-77.3540	3.67 s (4)	$\beta^- : 100.00 \%$ $\rightarrow 48\% (11/2-) \wedge (\text{仮定})$ $\rightarrow 52\% (3/2+) \wedge (\text{仮定})$ $\beta^- n : 0.69 \%$
			1.8630	(21/2-)	-78.7550	1.04 s (10)	$\beta^- : 100.00 \%$ $\rightarrow 100\% (11/2-) \wedge (\text{仮定})$
^{127}Sn	50	77	0.0	(11/2-)	-83.4699	2.10 h (4)	$\beta^- : 100.00 \%$
			0.0047	(3/2+)	-83.4746	4.13 m (3)	$\beta^- : 100.00 \%$
^{127}Sb	51	76	0.0	7/2+	-86.7007	3.85 d (5)	$\beta^- : 100.00 \%$ $\rightarrow 98\% 3/2+ \wedge$ $\rightarrow 2\% 11/2- \wedge$
^{127}Te	52	75	0.0	3/2+	-88.2830	9.35 h (7)	$\beta^- : 100.00 \%$
			0.0883	11/2-	-88.3713	106.1 d (7)	IT : 97.60 % $\beta^- : 2.40 \%$
^{127}I	53	74	0.0	5/2+	-88.9847	Abundance 100.00 %	

$A = 128$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{128}Pb	46	82	0.0	0+		>394 ns	β^- n:100%(假定) β^-
^{128}Ag	47	81	0.0		-54.9400 Syst	58 ms (5)	β^- : 100.00 % β^- n
^{128}Cd	48	80	0.0	0+	-67.2503	0.28 s (4)	β^- : 100.00 % \rightarrow 100% (3)+ \wedge
^{128}In	49	79	0.0	(3)+	-74.3606	0.84 s (6)	β^- : 100.00 % \rightarrow 100% 0+ \wedge β^- n < 0.05 %
			0.3400	(8-)	-74.7006	0.72 s (10)	β^- :100.00 % \rightarrow 68% 0+ \wedge (假定) \rightarrow 32% (7-) \wedge (假定) β^- n < 0.05 %
^{128}Sn	50	78	0.0	0+	-83.3362	59.07 m (14)	β^- : 100.00 % \rightarrow 100% 8- \wedge
			2.0915	(7-)	-85.4277	6.5 s (5)	IT : 100.00 %
^{128}Sb	51	77	0.0	8-	-84.6101	9.01 h (4)	β^- : 100.00 %
			0.0	5+	-84.6101	10.4 m (2)	β^- : 96.40 % IT : 3.60 %
^{128}Te	52	76	0.0	0+	-88.9937	2.41E+24 y (39) (31.74 % (8))	$2\beta^-$: 100.00 % stop
^{128}I	53	75	0.0	1+	-87.7395	24.99 m (2)	β^- : 93.10 % ε : 6.90 %
^{128}Xe	54	74	0.0	0+	-89.8602	Abundance 1.9102 % (8)	

$A = 129$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{129}Ag	47	82	0.0	(9/2+)	-50.5640Syst	46 ms +5-9	β^- : 100.00 % → 以上初期値へ繰込 β^-n
			0.0	(1/2-)	-52.5640	≈ 160 ms	β^-n β^- → 無視
^{129}Cd	48	81	0.0	(3/2+)	-63.3140 Syst	0.27 s (4)	β^- : 100%(仮定)
^{129}In	49	80	0.0	(9/2+)	-72.8138	0.61 s (1)	β^- : 100.00 % →97% (3/2+) \wedge →3% (11/2-) \wedge β^-n : 0.25 %
			0.3700	(1/2-)	-73.1837	1.23 s (3)	$\beta^- > 99.70$ % →97.5% (3/2+) \wedge β^-n : 2.50 % → 以上初期値へ繰込 IT < 0.30 %
			1.6300	(23/2-)	-74.4437	0.67 s (10)	β^- : 100.00 %
^{129}Sn	50	79	0.0	(3/2+)	-80.5937	2.23 m (4)	β^- : 100.00 % →100% 7/2+ \wedge
			0.0352	(11/2-)	-80.6289	6.9 m (1)	β^- : 100.00 % →95% 7/2+ \wedge →5% (19/2-) \wedge JENDL FP Decay Data IT < 2.0E-3 %
^{129}Sb	51	78	0.0	7/2+	-84.6293	4.40 h (1)	β^- : 100.00 % →89.4% 3/2+ \wedge →10.6% 11/2- \wedge
			1.8511	(19/2-)	-86.4804	17.7 m (1)	β^- : 85.00 % IT : 15.00 %
^{129}Te	52	77	0.0	3/2+	-87.0048	69.6 m (3)	β^- : 100.00 %
			0.1055	11/2-	-87.1103	33.6 d (1)	IT : 63.00 % β^- : 37.00 %
^{129}I	53	76	0.0	7/2+	-88.5071	1.57E+7 y (4)	β^- : 100.00 % →100% 1/2+ \wedge
^{129}Xe	54	75	0.0	1/2+	-88.6960	Abundance 26.4006 % (82)	
			0.2361	11/2-	-88.9321	8.88 d (2)	IT : 100.00 %

$A = 130$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{130}Pd	46	84					β^- -n :100%(仮定) → 初期値へ繰込
^{130}Ag	47	83	0.0		-46.2970 Syst	≈ 50 ms	β^- -n :100%(仮定) → 初期値へ繰込 β^-
^{130}Cd	48	82	0.0	0+	-61.5382	162 ms (7)	β^- : 100.00 % →96.5%1(-)へ β^- -n : 3.50 % → 初期値へ繰込
^{130}In	49	81	0.0	1(-)	-69.8882	0.29 s (2)	β^- :100.00 % →99.07%0+へ β^- -n : 0.93 % → 初期値へ繰込
			0.0500	(10-)	-69.9382	0.54 s (1)	β^- : 100.00 % →98.35%0+へ β^- -n :1.65 % → 初期値へ繰込
			0.4000	(5+)	-70.2882	0.54 s (1)	β^- : 100.00 % →98.35%0+へ β^- -n :1.65 % → 初期値へ繰込
^{130}Sn	50	80	0.0	0+	-80.1372	3.72 m (7)	β^- : 100.00 % →100% (8-) へ
			1.9469	(7-)	-82.0841	1.7 m (1)	β^- : 100.00 % →50% 8- へ (仮定) →50% (4,5)- へ (仮定)
^{130}Sb	51	79	0.0	(8-)	-82.2902	39.5 m (8)	β^- : 100.00 %
			0.0048	(4,5)+	-82.2950	6.3 m (2)	β^- : 100.00 %
^{130}Te	52	78	0.0	0+	-87.3529	$\geq 3.0\text{E}+24$ y (34.08 % (62))	$2\beta^-$: 100.00 % stop
^{130}I	53	77	0.0	5+	-86.9361	12.36 h (1)	β^- : 100.00 %
			0.0400	2+	-86.9761	8.84 m (6)	IT : 84.00 % β^- : 16.00 %
^{130}Xe	54	76	0.0	0+	-89.8804	Abundance 4.0710 % (13)	

$A = 131$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{131}Cd	48	83	0.0	(7/2-)	-55.3770 Syst	68 ms (3)	β^- : 100.00 % →96.5% (9/2+) \wedge β^-_{n} : 3.50 % → 初期値へ繰込
^{131}In	49	82	0.0	(9/2+)	-68.0497	0.28 s (3)	β^- : 100.00 % →98% (3/2+) \wedge $\beta^-_{\text{n}} \leq 2.00$ % → 初期値へ繰込
			0.3020	(1/2-)	-68.3517	0.35 s (5)	$\beta^- \geq 99.98$ % $\beta^-_{\text{n}} \leq 2.00$ % IT ≤ 0.02 % →(無視)
^{131}Sn	50	81	0.0	(3/2+)	-77.2717	56.0 s (5)	β^- : 100.00 %
			0.0	(11/2-)	-77.2717	58.4 s (5)	β^- : 100.00 % IT →(無視)
^{131}Sb	51	80	0.0	(7/2+)	-81.9760	23.03 m (4)	β^- : 100.00 % →93.5% 3/2+ \wedge →6.5% 11/2- \wedge
^{131}Te	52	79	0.0	3/2+	-85.2110	25.0 m (1)	β^- : 100.00 %
			0.1823	11/2-	-85.3933	33.25 h (25)	β^- : 74.10 % IT : 25.90 %
			1.9400	(23/2+)	-87.1510	93 ms (12)	IT : 100.00 % →(無視)
^{131}I	53	78	0.0	7/2+	-87.4427	8.0252 d (6)	β^- : 100.00 % →100% 3/2+ \wedge
^{131}Xe	54	77	0.0	3/2+	-88.4136	Abundance 21.232 % (30)	
			0.1639	11/2-	-88.5775	11.84 d (4)	IT : 100.00 % →(無視)

$A = 132$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{132}In	49	83	0.0	(7-)	-62.4135	0.207 s (6)	β^- : 93.7 % β^-n : 6.30 %
^{132}Sn	50	82	0.0	0+	-76.5485	39.7 s (8)	β^- : 100.00 % $\rightarrow 100\%$ (4)+ \wedge
^{132}Sb	51	81	0.0	(4)+	-79.6687	2.79 m (7)	β^- : 100.00 %
			0.0	(8-)	-79.6687	4.10 m (5)	β^- : 100.00 %
^{132}Te	52	80	0.0	0+	-85.1803	3.204 d (13)	β^- : 100.00 % $\rightarrow 100\%$ 4+ \wedge
^{132}I	53	79	0.0	4+	-85.6981	2.295 h (13)	β^- : 100.00 % $\rightarrow 100\%$ 0+ \wedge
			0.1200	(8-)	-85.8181	1.387 h (15)	IT : 86.00 % β^- : 14.00 %
^{132}Xe	54	78	0.0	0+	-89.2789	Abundance 26.9086 % (33)	
			2.7522	(10+)	-92.0311	8.39 ms (11)	IT : 100.00 %
^{132}Cs	55	77	0.0	2+	-87.1557	6.480 d (6)	ε : 98.13 % $\rightarrow 98.13\%$ 0+ \wedge β^- : 1.87 % $\rightarrow 1.87\%$ 0+ \wedge
^{132}Ba	56	76	0.0	0+	-88.4349	> 3.0E+21 y 0.101 % (1)	stop

$A = 133$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{133}In	49	84	0.0	(9/2+)	-57.7620 Syst	165 ms (3)	β^- : 100.00 % $\rightarrow 15\%$ 7/2- \wedge β^-n : 85.00 % \rightarrow 初期値へ繰込
^{133}Sn	50	83	0.0	7/2-	-70.8469	1.46 s (3)	β^- : 100.00 % β^-n : 0.03 %
^{133}Sb	51	82	0.0	(7/2+)	-78.9425	2.34 m (5)	β^- : 100.00 % $\rightarrow 100\%$ (3/2+) \wedge
^{133}Te	52	81	0.0	(3/2+)	-82.9445	12.5 m (3)	β^- : 100.00 % $\rightarrow 100\%$ 7/2+ \wedge
			0.3343	(11/2-)	-83.2788	55.4 m (4)	β^- : 83.50 % IT : 16.50 %
^{133}I	53	80	0.0	7/2+	-85.8865	20.83 h (8)	β^- : 100.00 % $\rightarrow 98.92\%$ 3/2+ \wedge $\rightarrow 1.08\%$ 11/2- \wedge
			1.6341	(19/2)	-87.5206	9 s (2)	IT : 100.00 % \rightarrow (無視)
^{133}Xe	54	79	0.0	3/2+	-87.6435	5.2475 d (5)	β^- : 100.00 %
			0.2332	11/2-	-87.8767	2.198 d (13)	IT : 100.00 %
^{133}Cs	55	78	0.0	7/2+	-88.0709	Abundance 100.00 %	

$A = 134$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{134}Cd	48	86					β^-_{n} : 100.00 % (仮定) → 初期値へ繰込
^{134}In	49	85	0.0	$(4 - to 7 -)$	-52.0240 Syst	140 ms (4)	β^- : 100.00 % → 35% $0+$ \wedge β^-_{n} : 65.00 % → 初期値へ繰込
^{134}Sn	50	84	0.0	$0+$	-66.3234	1.050 s (11)	β^- : 100.00 % → 83% $(0-)$ \wedge β^-_{n} : 17.00 % → 初期値へ繰込
^{134}Sb	51	83	0.0	$(0-)$	-74.1657	0.78 s (6)	β^- : 100.00 %
			0.2790	$(7-)$	-74.4447	10.07 s (5)	β^- : 100.00 % → (無視) β^-_{n} : 0.09 %
^{134}Te	52	82	0.0	$0+$	-82.5595	41.8 m (8)	β^- : 100.00 % → 100% $(4)+$ \wedge
^{134}I	53	81	0.0	$(4)+$	-84.0725	52.5 m (2)	β^- : 100.00 % → 100% $0+$ \wedge
			0.3165	$(8)-$	-84.3890	3.52 m (4)	IT : 97.70 % β^- : 2.30 % → $7-$ \wedge
^{134}Xe	54	80	0.0	$0+$	-88.1245	$> 5.8\text{E}+22$ y (10.4357 % (21))	$2\beta^-$
			1.9655	$7-$	-90.0899	290 ms (17)	IT : 100.00 %

$A = 135$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{135}Sn	50	85	0.0	(7/2-)	-60.6120 Syst	530 ms (20)	β^- : 100.00 % →79% (7/2+) \wedge β^-_{n} : 21.00 % → 初期値へ繰込
^{135}Sb	51	84	0.0	(7/2+)	-69.7872	1.679 s (15)	β^- : 100.00 % →78% (7/2-) \wedge β^-_{n} : 22.00 % → 初期値へ繰込
^{135}Te	52	83	0.0	(7/2-)	-77.9035	19.0 s (2)	β^- : 100.00 %
^{135}I	53	82	0.0	7/2+	-83.7913	6.58 h (3)	β^- : 100.00 % →98.1% 3/2+ \wedge →1.9% 11/2- \wedge
^{135}Xe	54	81	0.0	3/2+	-86.4176	9.14 h (2)	β^- : 100.00 % →100% 7/2+ \wedge
			0.5266	11/2-	-86.9442	15.29 m (5)	IT > 99.40 % β^- < 0.60 % → 7/2+ \wedge
^{135}Cs	55	80	0.0	7/2+	-87.5818	2.3E+6 y (3)	β^- : 100.00 % →100% 3/2+ \wedge
			1.6329	19/2-	-89.2147	53 m (2)	IT : 100.00 % →(無視)
^{135}Ba	56	79	0.0	3/2+	-87.8508	Abundance 6.592 % (12)	
			0.2682	11/2-	-88.1190	28.7 h (2)	IT : 100.00 %

$A = 136$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{136}In	49	87					β^- : 50.00 % (仮定) →50% 0+ \wedge β^-n : 50.00 % (仮定) → 初期値へ繰込
^{136}Sn	50	86	0.0	0+	−56.2990 Syst	0.25 s (3)	β^- : 100.00 % →70% 1− \wedge β^-n : 30.00 % → 初期値へ繰込
^{136}Sb	51	85	0.0	1−	−64.5250 Syst	0.923 s (14)	β^- : 100.00 % →83.7% 0+ \wedge β^-n : 16.30 % → 初期値へ繰込
^{136}Te	52	84	0.0	0+	−74.4790	17.63 s (8)	β^- : 100.00 % →98.69% (1−) \wedge β^-n : 1.31 % → 初期値へ繰込
^{136}I	53	83	0.0	(1−)	−79.5721	83.4 s (10)	β^- : 100.00 %
			0.6400	(6−)	−80.2120	46.9 s (10)	β^- : 100.00 %
^{136}Xe	54	82	0.0	0+	−86.4291	$> 2.4\text{E}+21$ y (8.8573 % (44))	$2\beta^-$ stop
^{136}Cs	55	81	0.0	5+	−86.3390	13.04 d (3)	β^- : 100.00 % →100% 0+ \wedge
			0.5179	8−	−86.8569	17.5 s (2)	IT > 0.00 % β^- →(無視)
^{136}Ba	56	80	0.0	0+	−88.8872	Abundance 7.854 % (24)	
			2.0305	7−	−90.9177	0.3084 s (19)	IT : 100.00 % →(無視)

$A = 137$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{137}Sn	50	87	0.0		-50.3100 Syst	190 ms (60)	β^- : 100.00 % →42% (7/2+) \wedge β^-_{n} : 58.00 % → 初期値へ繰込
^{137}Sb	51	86	0.0	(7/2+)	-60.3980 Syst	492 ms (25)	β^- : 100.00 % →51% (7/2-) \wedge β^-_{n} : 49.00 % → 初期値へ繰込
^{137}Te	52	85	0.0	(7/2-)	-69.2901	2.49 s (5)	β^- : 100.00 % →97.01% (7/2+) \wedge β^-_{n} : 2.99 % → 初期値へ繰込
^{137}I	53	84	0.0	(7/2+)	-76.5068	24.5 s (2)	β^- : 100.00 % →92.86% 7/2- \wedge β^-_{n} : 7.14 % → 初期値へ繰込
^{137}Xe	54	83	0.0	7/2-	-82.3833	3.818 m (13)	β^- : 100.00 %
^{137}Cs	55	82	0.0	7/2+	-86.5459	30.08 y (9)	β^- : 100.00 % →5.3% 3/2+ \wedge →94.7% 11/2- \wedge
^{137}Ba	56	81	0.0	3/2+	-87.7215	Abundance 11.232 % (24)	
			0.6617	11/2-	-88.3832	2.552 m (1)	IT : 100.00 %

$A = 138$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{138}Sn	50	88	0.0	0+		>408 ns	β^- -n : 100.00 % (仮定) → 初期値へ繰込 β^-
^{138}Sb	51	87	0.0		-54.8460 Syst	350 ms (15)	β^- : 100.00 % → 28% 0+ \wedge β^- -n : 72.00 % → 初期値へ繰込
^{138}Te	52	86	0.0	0+	-65.7551	1.4 s (4)	β^- : 100.00 % → 93.7% (2-) \wedge β^- -n : 6.30 % → 初期値へ繰込
^{138}I	53	85	0.0	(2-)	-71.9050 Syst	6.23 s (3)	β^- : 100.00 % → 94.44% 0+ \wedge β^- -n : 5.56 % → 初期値へ繰込
^{138}Xe	54	84	0.0	0+	-79.9750	14.08 m (8)	β^- : 100.00 % → 100% 3- \wedge
^{138}Cs	55	83	0.0	3-	-82.8870	33.41 m (18)	β^- : 100.00 %
			0.0799	6-	-82.9669	2.91 m (8)	IT : 81.00 % β^- : 19.00 %
^{138}Ba	56	82	0.0	0+	-88.2619	Abundance 71.698 % (42)	
^{138}La	57	81	0.0	5+	-86.5215	1.02E+11 y (1)	ε : 65.60 % β^- : 34.40 % → 0+ \wedge
^{138}Ce	58	80	0.0	0+	-87.5643	$\geq 0.9\text{E}+14$ y (0.251 % (2))	2ε : 100.00 %
			2.1292	7-	-89.6935	8.65 ms (20)	IT : 100.00 % → (無視)

$A = 139$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{139}Sb	51	88	0.0		-50.3190Syst	93 ms +14-3	β^- : 100.00 % →10% (7/2-) \wedge β^-_{n} : 90.00 % → 初期値へ繰込
^{139}Te	52	87	0.0	(7/2-)	-60.3890Syst	> 150 ns	β^-_{n} : 100.00 % (仮定) → 初期値へ繰込 β^-
^{139}I	53	86	0.0	(7/2+)	-68.5272	2.280 s (11)	β^- : 100.00 % →90% 3/2- \wedge β^-_{n} : 10.00 % → 初期値へ繰込
^{139}Xe	54	85	0.0	3/2-	-75.6445	39.68 s (14)	β^- : 100.00 %
^{139}Cs	55	84	0.0	7/2+	-80.7012	9.27 m (5)	β^- : 100.00 %
^{139}Ba	56	83	0.0	7/2-	-84.9140	83.06 m (28)	β^- : 100.00 %
^{138}Te	52	86	0.0	0+	-65.7551	1.4 s (4)	β^- : 100.00 % →93.7% (2-) \wedge β^-_{n} : 6.30 % → 初期値へ繰込
^{139}La	57	82	0.0	7/2+	-87.2282	Abundance 99.9119 % (71)	
^{139}Ce	58	81	0.0	3/2+	-86.9498	137.641 d (20)	ε : 100.00 %
			0.7542	11/2-	-87.7040	54.8 s (10)	IT : 10.00 %

$A = 140$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{140}Te	52	88	0.0	0+	-56.5980Syst	> 300 ns	β^-_{n} : 100.00 % (仮定) → 初期値へ繰込 β^-
^{140}I	53	87	0.0	4(-)	-63.5958	0.86 s (4)	β^- : 100.00 % →90.7% 0(+) \wedge β^-_{n} : 9.30 % → 初期値へ繰込
^{140}Xe	54	86	0.0	0(+)	-72.9864	13.60 s (10)	β^- : 100.00 %
^{140}Cs	55	85	0.0	1(-)	-77.0503	63.7 s (3)	β^- : 100.00 %
^{140}Ba	56	84	0.0	0(+)	-83.2703	12.7527 d (23)	β^- : 100.00 %
^{140}La	57	83	0.0	3(-)	-83.3179	1.67855 d (12)	β^- : 100.00 %
^{140}Ce	58	82	0.0	0(+)	-88.0786	Abundance 88.450 % (51)	

$A = 141$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{141}I	53	88	0.0		-60.3011	0.43 s (2)	β^- : 100.00 % →78.8% 5/2(-) へ β^-_{n} : 21.20 % →初期値へ繰込
^{141}Xe	54	87	0.0	5/2(-)	-68.1973	1.73 s (1)	β^- : 100.00 % β^-_{n} : 0.04 %
^{141}Cs	55	86	0.0	7/2+	-74.4803	24.84 s (16)	β^- : 100.00 % β^-_{n} : 0.04 %
^{141}Ba	56	85	0.0	3/2-	-79.7331	18.27 m (7)	β^- : 100.00 %
^{141}La	57	84	0.0	(7/2+)	-82.9341	3.92 h (3)	β^- : 100.00 %
^{141}Ce	58	83	0.0	7/2-	-85.4354	32.508 d (13)	β^- : 100.00 %
^{141}Pr	59	82	0.0	5/2-	-86.0158	Abundance 100 %	

$A = 142$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{142}Te	52	90	0.0	0+	-46.9010Syst		β^- : 100.00 % (仮定) →初期値へ繰込
^{142}I	53	89	0.0		-55.0230Syst	2.22 ms (12)	β^- : 100.00 % β^-_{n} ?
^{142}Xe	54	88	0.0	0+	-65.2296	1.23 s (2)	β^- : 100.00 % β^-_{n} : 0.21 %
^{142}Cs	55	87	0.0	0-	-70.5253	1.684 s (14)	β^- : 100.00 % β^-_{n} : 0.09 %
^{142}Ba	56	86	0.0	0+	-77.8450	10.6 m (2)	β^- : 100.00 %
^{142}La	57	85	0.0	2-	-80.0220	91.1 m (5)	β^- : 100.00 %
^{142}Ce	58	84	0.0	0+	-84.5320	>5E+16 y (11.114 %)	$2\beta^-$ stop
^{142}Pr	59	83	0.0	2-	-83.7877	19.12 h (4)	β^- : 99.98 % ε : 0.02 %
			0.0037	5-	-83.7914	14.6 m (5)	IT : 100.00 % →(無視)
^{142}Nd	60	82	0.0	0+	-85.9493	Abundance 27.152 % (40)	

$A = 143$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{143}Xe	54	89	0.0	$5/2^-$	-60.2028	0.511 s (6)	β^- : 100.00 % →99% $3/2^+$ \wedge β^-_{n} : 1.00 % → 初期値へ繰込
^{143}Cs	55	88	0.0	$3/2^+$	-67.6749	1.791 s (7)	β^- : 100.00 % →98.36% $5/2^-$ \wedge β^-_{n} : 1.64 % → 初期値へ繰込
^{143}Ba	56	87	0.0	$5/2^-$	-73.9371	14.5 s (3)	β^- : 100.00 %
^{143}La	57	86	0.0	$(7/2)^+$	-78.1713	14.2 m (1)	β^- : 100.00 %
^{143}Ce	58	85	0.0	$3/2^-$	-81.6055	33.039 h (6)	β^- : 100.00 %
^{143}Pr	59	84	0.0	$7/2^+$	-83.0674	13.57 d (2)	β^- : 100.00 %
^{143}Nd	60	83	0.0	$7/2^+$	-84.0015	Abundance 12.174 % (26)	

$A = 144$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{144}Xe	54	90	0.0	0^+	-56.8722	0.388 s (7)	β^- : 100.00 % →97% 1^- \wedge β^-_{n} : 3.00 % → 初期値へ繰込
^{144}Cs	55	89	0.0	1^-	-63.2712	0.994 s (6)	β^- : 100.00 % →96.97% 0^+ \wedge β^-_{n} : 3.03 % → 初期値へ繰込
			0.0	(GE4)	-63.2712	< 1 s	β^- →(無視)
^{144}Ba	56	88	0.0	0^+	-71.7671	11.5 s (2)	β^- : 100.00 % →96.4% (3^-) \wedge β^-_{n} : 3.60 % → 初期値へ繰込
^{144}La	57	87	0.0	(3^-)	-74.8335	40.8 s (4)	β^- : 100.00 %
^{144}Ce	58	86	0.0	0^+	-80.4312	284.91 d (5)	β^- : 100.00 % →100% 0^- \wedge
^{144}Pr	59	85	0.0	0^-	-80.7498	17.28 m (5)	β^- : 100.00 %
			0.0590	3^-	-80.8088	7.2 m (3)	IT : 99.93 % β^- : 0.07 %
^{144}Nd	60	84	0.0	0^+	-83.7473	2.29E+15 y (16) (23.798 % (19))	α : 100.00 % stop

$A = 145$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{145}Xe	54	91	0.0		-51.4933	188 ms (4)	β^- : 100.00 % →95% 3/2+ \wedge β^-n : 5.00 % → 初期値へ繰込
^{145}Cs	55	90	0.0	3/2+	-60.0557	0.587 s (5)	β^- : 100.00 % →85.3% 5/2- \wedge β^-n : 14.70 % → 初期値へ繰込
^{145}Ba	56	89	0.0	5/2-	-67.5161	4.31 s (16)	β^- : 100.00 %
^{145}La	57	88	0.0	(5/2+)	-72.8324	24.8 s (20)	β^- : 100.00 %
^{145}Ce	58	87	0.0	(5/2-)	-77.0929	3.01 m (6)	β^- : 100.00 %
^{145}Pr	59	86	0.0	7/2+	-79.6260	5.984 h (10)	β^- : 100.00 %
^{145}Nd	60	85	0.0	7/2+	-81.4312	Abundance 8.293 % (12)	

$A = 146$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{146}Xe	54	92	0.0	0+	-47.9547	146 ms (6)	β^- : 100.00 % →93.1% 1- \wedge β^-n : 6.90 % → 初期値へ繰込
^{146}Cs	55	91	0.0	1-	-55.5691	0.321 s (2)	β^- : 100.00 % →85.8% 0+ \wedge β^-n : 14.20 % → 初期値へ繰込
^{146}Ba	56	90	0.0	0+	-64.9407	2.22 s (7)	β^- : 100.00 % →100% 2- \wedge
^{146}La	57	89	0.0	2-	-69.0490	6.27 s (10)	β^- : 100.00 %
			0.0	(6-)	-69.0490	10.0 s (1)	β^- : 100.00 %
^{146}Ce	58	88	0.0	0+	-75.6409	13.52 m (13)	β^- : 100.00 %
^{146}Pr	59	87	0.0	(2)-	-76.6852	24.15 m (18)	β^- : 100.00 %
^{146}Nd	60	86	0.0	0+	-80.9252	Abundance 17.189 % (32)	
^{146}Pm	61	85	0.0	3-	-79.4539	5.53 y (5)	ϵ : 66.00 % β^- : 34.00 %
^{146}Sm	62	84	0.0	0+	-80.9959	10.3E+7 y (5)	α : 100.00 % stop

$A = 147$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{147}Cs	55	92	0.0	(3/2+)	-52.0179	0.230 s (1)	β^- : 100.00 % →71.5% (3/2-) \wedge β^-_{n} : 28.50 % → 初期値へ繰込
^{147}Ba	56	91	0.0	(3/2-)	-60.2640	0.894 s (10)	β^- : 100.00 % β^-_{n} : 0.06 % →(無視)
^{147}La	57	90	0.0	(3/2+)	-66.6783	4.06 s (4)	β^- : 100.00 % β^-_{n} : 0.04 % →(無視)
^{147}Ce	58	89	0.0	(5/2-)	-72.0138	56.4 s (10)	β^- : 100.00 %
^{147}Pr	59	88	0.0	(5/2+)	-75.4437	13.4 m (3)	β^- : 100.00 %
^{147}Nd	60	87	0.0	5/2+	-78.1460	10.98 d (1)	β^- : 100.00 %
^{147}Pm	61	86	0.0	7/2+	-79.0416	2.6234 y (2)	β^- : 100.00 %
^{147}Sm	62	85	0.0	7/2-	-79.2657	1.060E11 y (11) (14.99 % (18))	α : 100.00 % stop

$A = 148$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{148}Xe	54	94	0.0	0+		> 408 ns	β^-_{n} : 100%(仮定) β^-
^{148}Cs	55	93	0.0		-47.2944	146 ms (6)	β^- : 100.00 % →74.9% 0+ \wedge β^-_{n} : 25.10 % → 初期値へ繰込
^{148}Ba	56	92	0.0	0+	-57.5937	0.612 s (17)	β^- : 100.00 % β^-_{n} : 0.40 % →(無視)
^{148}La	57	91	0.0	(2-)	-62.7087	1.26 s (8)	β^- : 100.00 % β^-_{n} : 0.15 % →(無視)
^{148}Ce	58	90	0.0	0+	-70.3982	56 s (1)	β^- : 100.00 % →100% 1- \wedge
^{148}Pr	59	89	0.0	1-	-72.5350	2.29 m (2)	β^- : 100.00 %
			0.0900	(4)	-72.6250	2.01 m (7)	β^- : 100.00 %
^{148}Nd	60	88	0.0	0+	-77.4068	Abundance 5.756 % (21)	

$A = 149$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{149}Cs	55	94	0.0		-43.8450 Syst	> 50 ms	β^- -n : 100%(仮定) β^-
^{149}Ba	56	93	0.0		-53.1700 Syst	0.344 s (7)	β^- : 100.00 % $\rightarrow 99.57\%$ (3/2-) \wedge β^- -n : 0.43 % \rightarrow 初期値 \wedge 繰込
^{149}La	57	92	0.0	(3/2-)	-60.2199	1.05 s (3)	β^- : 100.00 % $\rightarrow 98.57\%$ (3/2-) \wedge β^- -n : 1.43 % \rightarrow 初期値 \wedge 繰込
^{149}Ce	58	91	0.0	(3/2-)	-66.6699	5.3 s (2)	β^- : 100.00 %
^{149}Pr	59	90	0.0	(5/2+)	-71.0393	2.26 m (7)	β^- : 100.00 %
^{149}Nd	60	89	0.0	5/2-	-74.3743	1.728 h (1)	β^- : 100.00 %
^{149}Pm	61	88	0.0	7/2+	-76.0635	53.08 h (5)	β^- : 100.00 %
^{149}Sm	62	87	0.0	7/2-	-77.1350	Abundance 13.82 % (7)	

$A = 150$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{150}Cs	55	95	0.0		-38.9640 Syst	> 50 ms	β^- -n : 100%(仮定) β^-
^{150}Ba	56	94	0.0	0+	-50.3470 Syst	0.3 s	β^- β^- -n : 100%(仮定)
^{150}La	57	93	0.0	(3+)	-56.5510 Syst	0.86 s (5)	β^- : 100.00 % $\rightarrow 97.3\%$ 0+ \wedge β^- -n : 2.7 % \rightarrow 初期値に繰込
^{150}Ce	58	92	0.0	0+	-64.8493	4.0 s (6)	β^- : 100.00 %
^{150}Pr	59	91	0.0	(1)-	-68.2995	6.19 s (16)	β^- : 100.00 %
^{150}Nd	60	90	0.0	0+	-73.6832	0.79E19 y (7) (5.638 % (28))	stop
^{150}Pm	61	89	0.0	(1-)	-73.5964	2.68 h (2)	β^- : 100.00 %
^{150}Sm	62	88	0.0	0+	-77.0504	Abundance 7.38 % (1)	

$A = 151$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{151}Cs	55	96	0.0		-35.0610 Syst	> 50 ms	β^- -n : 100%(仮定) β^-
^{151}Ba	56	95	0.0		-45.5690 Syst	> 300 ns	β^- -n : 100%(仮定) β^-
^{151}La	57	94	0.0		-53.8870 Syst	> 300 ns	β^- -n : 100%(仮定) β^-
^{151}Ce	58	93	0.0	(5/2+)	-61.2250	1.76 s (6)	β^- : 100.00 %
			0.0		-61.2250	1.20 s (6)	β^- : 100%(仮定)
^{151}Pr	59	92	0.0	(3/2-)	-66.7804	18.90 s (7)	β^- : 100.00 %
^{151}Nd	60	91	0.0	3/2+	-70.9465	12.44 m (7)	β^- : 100.00 %
^{151}Pm	61	90	0.0	5/2+	-73.3887	28.40 h (4)	β^- : 100.00 %
^{151}Sm	62	89	0.0	5/2-	-74.5755	90 y (8)	β^- : 100.00 %
^{151}Eu	63	88	0.0	5/2+	-74.6517	$\geq 1.7\text{E}+18$ y (47.81 % (3))	α stop

$A = 152$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{152}Ba	56	96	0.0	0+	-42.3640 Syst	> 406 ns	β^- -n : 100%(仮定) β^-
^{152}La	57	95	0.0		-49.7320 Syst	> 150 ns	β^- : 100.00 %
^{152}Ce	58	94	0.0	0+	-59.3080 Syst	1.4 s (2)	β^- : 100.00 %
^{152}Pr	59	93	0.0	(4+)	-63.7580	3.57 s (18)	β^- : 100.00 %
^{152}Nd	60	92	0.0	0+	-70.1516	11.4 m (2)	β^- : 100.00 % $\rightarrow 100\%$ 1+ \wedge
^{152}Pm	61	91	0.0	1+	-71.2558	4.12 m (8)	β^- : 100.00 %
			0.1500	(8)	-71.4058	13.8 m (2)	β^- : 100.00 % IT ≥ 0.00 %
			0.1500	4-	-71.4058	7.52 m (8)	β^- : 100.00 % \rightarrow (無視)
^{152}Sm	62	90	0.0	0+	-74.7622	Abundance 26.75 % (16)	stop
^{152}Eu	63	89	0.0	3-	-72.8871	13.528 y (14)	ϵ : 72.10 % β^- : 27.90 %
			0.0456	0-	-72.9327	9.3116 h (13)	β^- : 72.00 % ϵ : 28.00 %
			0.1479	8-	-73.0350	96 m (1)	IT : 100.00 %
^{152}Gd	64	88	0.0	0+	-74.7065	1.08E14 y (8) (0.20 % (1))	α : 100.00 %

$A = 153$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{153}La	57	96	0.0		-46.5650 Syst	> 100 ns	β^- : 100%(仮定)
^{153}Ce	58	95	0.0		-55.2280 Syst	> 100 ns	β^- : 100%(仮定)
^{153}Pr	59	94	0.0		-61.5815	4.28 s (11)	β^- : 100.00 %
^{153}Nd	60	93	0.0	$(3/2)-$	-67.3420	31.6 s (10)	β^- : 100.00 %
^{153}Pm	61	92	0.0	$5/2-$	-70.6780	5.25 m (2)	β^- : 100.00 % $\rightarrow 100\%$ $3/2+$ \wedge
^{153}Sm	62	91	0.0	$3/2+$	-72.5593	46.284 h (4)	β^- : 100.00 %
			0.0984	$11/2-$	-72.6577	10.6 ms (3)	IT : 100.00 %
^{153}Eu	63	90	0.0	$5/2+$	-73.3661	Abundance 52.19 % (6)	

$A = 154$

	Z	N	E(level) (MeV)	$J\pi$	Δ (MeV)	$T_{1/2}$	Decay Modes
^{154}La	57	97	0.0		-42.0380 Syst		β^- : 100%(仮定) \rightarrow 初期値へ繰込
^{154}Ce	58	96	0.0	$0+$	-52.7320 Syst	> 100 ns	β^- : 100.00 % \rightarrow 初期値へ繰込
^{154}Pr	59	95	0.0	$(3+)$	-58.1947	2.3 s (1)	β^- : 100.00 %
^{154}Nd	60	94	0.0	$0+$	-65.6847	25.9 s (2)	β^- : 100.00 % $\rightarrow 100\%$ $(0-, 1-)$ \wedge
^{154}Pm	61	93	0.0	$(3, 4)$	-68.4917	2.68 m (7)	β^- : 100.00 %
			0.0	$(0-, 1-)$	-68.4917	1.73 m (10)	β^- : 100.00 %
^{154}Sm	62	92	0.0	$0+$	-72.4549	Abundance 22.75 % (29)	stop
^{154}Eu	63	91	0.0	$3-$	-71.7370	8.601 y (10)	β^- : 99.98 % ϵ : 0.02 %
			0.1453	$8-$	-71.8822	46.3 m (4)	IT : 100.00 %
^{154}Gd	64	90	0.0	$0+$	-73.7055	Abundance 2.18 % (3)	

付録2 プログラムソースコード

プログラムの概要は以下のようである。

- データソースから λ の設定
- 核の個数の初期値の設定
- C_{ij} を解析的に解く
- n_i の和のチェック

decay_chain

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define M 30
#define BIGNUM 1.0e+300
#define BIGNUM0 1.0e+299

int nn; // the number of nuclei in the chain of mass number A
int mass_number[M]; // =A
int atomic_number[M]; // =Z
int state_id[M]; // =0 for ground state, =1,2,... for isomeric states
double lambda[M][M]; // decay rate matrix
double n[M]; // population of each nuclei
double n0[M]; // initial population of nuclei
double c[M][M]; // coefficients used to express the solutions
double ca[M][M]; // analytic result for c
double halflife[M]; // halflife of each nuclei
enum{sec,min,hour,day,year};

int A_chain(int A, double tmin, double tmax, int ntime, FILE *fp_evo);
double time_in_sec(double time,int timeunit);
int solve_c();
double n_of_t(int i,double time);
int time_evolution(double tmin, double tmax, int ntime, FILE *fp_evo);
int check_sum_n(double tmin,double tmax,int ntime);
int check_derivative(double tmin,double tmax,int ntime);

int set_lambda(int A);
double halflife_to_decayrate(double halflife);
int set_halflife(int element, int A, int Z, int state, double T, int unit);
```

```

int set_initial_state(int A);
// int set_initial_state_140();

int main(){
    double tmin=1.0e-3; // 1ms      ; -1.0 for automatic determination;
    double tmax=1.0e+16; // 100years ; -1.0 for automatic determination;
    int ntime=2000; // -1 for default value = 100;
    int A;
    FILE *fp_evo=NULL; // file to write time evolution

    fp_evo=fopen("time_evolution.dat","wt");
    if(fp_evo == NULL){
        fprintf(stderr,"failed to open time evolution output file\n");
        exit(1);
    }
    for(A=126;A<=154;A++){
        A_chain(A,tmin,tmax,ntime,fp_evo);
    }
    if(fclose(fp_evo)!=0){
        fprintf(stderr,"failed to close evolution output file\n");
    }
    return 0;
}

int A_chain(int A, double tmin, double tmax, int ntime, FILE *fp_evo){
    int i,j,k,l,ng,degeneracy;

    set_lambda(A);

    for(j=0;j<nn;j++){
        printf(" [%2d] A=%3d Z=%3d state=%1d\n"
            ,j,mass_number[j],atomic_number[j],state_id[j]);
        for(i=0;i<j;i++){ // check whether there is the same nuclei
            if( mass_number[i]== mass_number[j] &&
                atomic_number[i]==atomic_number[j] &&
                state_id[i]== state_id[j] ){
                printf("Error: Entries no. %d and no. %d are the same\n",i,j);
            }
        }
    }

    printf("\n lambda[0..%d][0..%d]=\n",nn-1,nn-1);
    ng=0; // increased from zero if matrix lambda is not lower triangular
    degeneracy=0; // changed to >0 if diagonal elements are equal
    for(k=0;k<nn;k++){
        for(l=0;l<nn;l++){
            printf(" %15.7e",lambda[k][l]);
            if(l>k && lambda[k][l]!=0.0) ng++;
            if(l!=k &&
                fabs(lambda[k][k]-lambda[l][l])/

```

```

        (fabs(lambda[k][k])+fabs(lambda[l][l])+1.0e-100)
        <1.0e-13) degeneracy++;
    }
    printf("\n");
}
if(ng>0){
    printf("lambda is not lower triangular %d\n",ng);
    exit(1);
}
if(degeneracy>0){
    printf("eigen values of lambda is degenerated %d/2\n",degeneracy);
    exit(1);
}

set_initial_state(A);

printf("\n  n0[0..%d]=\n",nn-1);
for(k=0;k<nn;k++){
    printf(" %12.5e",n0[k]);
}
printf("\n");

/*printf("\n  ca[i][j]=\n");
for(k=0;k<nn;k++){
    for(l=0;l<nn;l++){
        printf(" %15.7e",ca[k][l]);
    }
    printf("\n");
}*/

solve_c();

printf("\n  c[0..%d][0..%d]=\n",nn-1,nn-1);
for(k=0;k<nn;k++){
    for(l=0;l<nn;l++){
        printf(" %15.7e",c[k][l]);
    }
    printf("\n");
}

if(tmin < 0.0){
    tmin=BIGNUM;
    for(i=0;i<nn;i++){
        if(halflife[i]>BIGNUM0) continue;
        if(tmin>halflife[i]) tmin=halflife[i];
    }
    tmin*=0.01;
}
if(tmax < 0.0){
    tmax=0.0;

```

```

        for(i=0;i<nn;i++){
            if(halflife[i]<BIGNUM0) tmax+=halflife[i];
        }
        tmax*=30.0;
    }
    if(tmin>tmax || tmin <= 0.0 || tmax > BIGNUM0){
        fprintf(stderr,"error: tmin=%e tmax=%e\n",tmin,tmax);
        exit(0);
    }
    if(ntime<0) ntime=100;

    time_evolution(tmin,tmax,ntime,fp_evo);
    check_sum_n(tmin,tmax,ntime);
    check_derivative(tmin,tmax,ntime);
    return 0;
}

double time_in_sec(double time,int timeunit){
    switch(timeunit){
        case sec :return time;break;
        case min :return time*60.0;break;
        case hour :return time*3600.0;break;
        case day :return time*86400.0;break;
        case year :return time*86400.0*365.0;break;
        default :
            fprintf(stderr,"time_in_sec:unknown time unit\n");
            exit(1);
    }
}

/*int analytic_c(){

    ca[0][0]=n0[0];
    ca[1][1]=n0[1]-((-lambda[0][0])/(-lambda[1][1]+lambda[0][0]))*ca[0][0];
    ca[2][2]=n0[2]-((-lambda[1][1])/(-lambda[2][2]+lambda[1][1]))*ca[1][1]
    -((-lambda[1][1])*(-lambda[0][0]))/((-lambda[2][2]+lambda[0][0]))
    *(-lambda[1][1]+lambda[0][0]))*ca[0][0];
    ca[1][0]=1/(-lambda[1][1]+lambda[0][0])*lambda[1][0]*ca[0][0];
    ca[2][0]=1/(-lambda[2][2]+lambda[0][0])*(lambda[2][0]*ca[0][0]+lambda[2][1]*ca[1][0]);
    ca[2][1]=1/(-lambda[2][2]+lambda[1][1])*lambda[2][1]*ca[1][1];
}
*/

int solve_c(){
    int i,j,k;
    double cjk; //c_[j][k]
    double laikckj; //lambda_[i][k]*c_[k][j]

    for(j=0;j<nn;j++){
        if(j==0){

```

```

        c[j][j]=n0[0];
    }
    else{
        cjk=0;
        for(k=0;k<=j-1;k++){
            cjk+=c[j][k];
        }
        c[j][j]=n0[j]-cjk;
    }
    if(j<nn-1){
        for(i=j+1;i<nn;i++){
            laikckj=0;
            for(k=j;k<=i-1;k++){
                laikckj+=(lambda[i][k]*c[k][j]);
            }
            c[i][j]=(laikckj)/(-lambda[i][i]+lambda[j][j]);
        }
    }
}
return 0;
}

double n_of_t(int i,double time){
    int j;
    double s=0;

    for(j=0;j<=i;j++){
        s+=c[i][j]*exp(lambda[j][j]*time);
    }
    return s;
}

int time_evolution(double tmin, double tmax, int ntime, FILE *fp_evo){
    int i,it;
    double r,rr,t,bq;
    r=pow(tmax/tmin,1.0/(ntime-1));
    printf("\n# time n[0..%d]\n",nn-1);
    rr=1;
    for(it=0; it<=ntime; it++){
        if(it==0){
            t=0;
        }
        else{
            t=tmin*rr;
            rr*=r;
        }
        printf("%11.5e",t);
        for(i=0;i<nn;i++){
            n[i]=n_of_t(i,t);
            bq=n[i]*fabs(lambda[i][i]); // becquerel= decays per second

```

```

        if(bq<1.0e-99) bq=0.0;
        if(fabs(n[i])<1.0e-99) n[i]=0.0;
        printf(" %12.5e",n[i]);
        if(fp_evo != NULL){
            fprintf(fp_evo,"%11.5e %12.5e %12.5e %3d %3d %1d %d \n"
                ,t,n[i],bq,mass_number[i],atomic_number[i],state_id[i],it);
        }
    }
    printf(" :evo\n");
}
return 0;
}

int check_sum_n(double tmin,double tmax,int ntime){
    double s=0.0,s0=0.0,ds=0.0,maxerror=0.0,time=0.0;
    double r,rr;
    int i,itime=0;

    for(i=0;i<=nn-1;i++){
        s0+=n0[i];
    }
    r=pow(tmax/tmin,1.0/(ntime-1));
    rr=1;
    for(itime=1;itime<=ntime;itime++){
        time=tmin*rr;
        rr*=r;
        s=0.0;
        for(i=0;i<nn;i++){
            n[i]=n_of_t(i,time);
            if(n[i]>=0){
                s+=n[i];
            }
            else if(n[i]<-1.0e-16){
                printf("check_sum_n:error:negative n[%d]=%15.7e at t=%d\n"
                    ,i,n[i],itime);
            }
        }
        ds=fabs(s-s0);
        if(ds>1.0e-9){
            printf("check_sum_n:warning:large error %.5e at time %.5e\n",ds,time);
        }
        if(maxerror<ds) maxerror=ds;
    }
    printf("\nSigma n's max error =%.7e, max relative error=%.7e\n"
        ,maxerror,maxerror/(s0+1.0e-99));

    return 0;
}

int check_derivative(double tmin,double tmax,int ntime){

```

```

double sumn,dn,ds,dt,time,dtopt,r,rr;
double relative_error, max_relative_error=0.0, maxerror=0.0;
double ds1,ds2,ds3,ds4,ds5,dsmin;
int i,j,k,itime;
const int printlevel = 0;

r=pow(tmax/tmin,1.0/(ntime-1));
rr=1;
for(itime=0;itime<=ntime;itime++){ //ntime_partition
    if(itime==0){
        time=0;
    }
    else{
        time=tmin*rr;
        rr*=r;
    }
    if(printlevel>0) printf("%11.5e",time);
    for(i=0;i<nn;i++){
        n[i]=n_of_t(i,time);
        sumn=0.0;
        for(j=0;j<nn;j++){
            sumn+=lambda[i][j]*n[j];
        }

        dsmin=ds1=ds2=ds3=ds4=ds5=BIGNUM;
        dt=1.0e-12;
        for(k=1;k<=200;k++){ // search dt which minimizes the error
            dn=(- n_of_t(i,time+dt*2)+8*n_of_t(i,time+dt)
                -8*n_of_t(i,time-dt) + n_of_t(i,time-dt*2))/(12*dt);
            ds5=ds4; ds4=ds3; ds3=ds2; ds2=ds1; ds1=fabs(dn-sumn);
            ds=(ds1+ds2+ds3+ds4+ds5)/5.0;
            if(dsmin > ds) {dsmin=ds; dtopt=dt;}
            dt=dt*1.77827941; // 10^(1/4)=1.778
            if(dt>tmax) break;
        }
        ds=dsmin;
        if(maxerror<ds) maxerror=ds;
        relative_error=ds/(fabs(sumn)+1.0e-50);
        if(printlevel==1) printf(" %12.5e",ds);
        if(printlevel==2) printf(" %12.5e",relative_error);
        if(printlevel==3) printf(" %12.5e %12.5e",ds,dtopt);
        if(printlevel>3) printf(" %12.5e %12.5e",relative_error,dtopt);
        if(max_relative_error<relative_error) max_relative_error=relative_error;
    }
    if(printlevel>0)printf("\n");
}
printf("\ndn/dt's max error=%.7e, max relative error=%.7e\n"
,maxerror,max_relative_error);
return 0;
}

```

```

//-----
#include "set_lambda_140.c"

int set_lambda(int A){
    nn=-1;
    if(A==126) set_lambda_126(A);
    if(A==127) set_lambda_127(A);
    if(A==128) set_lambda_128(A);
    if(A==129) set_lambda_129(A);
    if(A==130) set_lambda_130(A);
    if(A==131) set_lambda_131(A);
    if(A==132) set_lambda_132(A);
    if(A==133) set_lambda_133(A);
    if(A==134) set_lambda_134(A);
    if(A==135) set_lambda_135(A);
    if(A==136) set_lambda_136(A);
    if(A==137) set_lambda_137(A);
    if(A==138) set_lambda_138(A);
    if(A==139) set_lambda_139(A);
    if(A==140) set_lambda_140(A);
    if(A==141) set_lambda_141(A);
    if(A==142) set_lambda_142(A);
    if(A==143) set_lambda_143(A);
    if(A==144) set_lambda_144(A);
    if(A==145) set_lambda_145(A);
    if(A==146) set_lambda_146(A);
    if(A==147) set_lambda_147(A);
    if(A==148) set_lambda_148(A);
    if(A==149) set_lambda_149(A);
    if(A==150) set_lambda_150(A);
    if(A==151) set_lambda_151(A);
    if(A==152) set_lambda_152(A);
    if(A==153) set_lambda_153(A);
    if(A==154) set_lambda_154(A);
    if(nn==-1){
        fprintf(stderr,"set_lambda: A=%d is not implemented.\n",A);
        exit(1);
    }
    return 0;
}

double halflife_to_decayrate(double halflife){ // T_{1/2} -> lambda converter
    const double ln2=log(2.0);
    static int zero_counter=0;
    if(halflife>BIGNUM0){
        return (++zero_counter +10)*1.0e-31; // to avoid degeneracy of zero
    }
    return (ln2/(halflife));
}

```



```

int set_half-life(int element, int A, int Z, int state, double T, int unit){
// element =0,1,...,nn-1 is the serial number to specify nucleus and state
// A is mass number
// Z is atomic number
// state is =0 for ground state, =1,2,.. for isomers
// T is half life
// unit is the unit of T to be chosen among enum{sec,min,hour,day,year}
    mass_number[element]=A;
    atomic_number[element]=Z;
    state_id[element]=state;
    half-life[element]=time_in_sec(T,unit);
    lambda[element][element]=-half-life_to_decayrate(half-life[element]);
    return 0;
}

//-----

#define ZMAX1 100
#define NMAX1 150
#define ISMR1 3

double fission_yield[ZMAX1][NMAX1][ISMR1]; // [Z][N][iex]
// iex=0 for ground state, iex>=1 for isomers

int set_initial_state(int A){
    FILE *fp;
    char buff[1024];
    int i1,i2,i3,i4,i,j,k,z,n; // NB) name collision of "n" with a global array
    double r1,sum;
    const int debug=1;

    for(i=0;i<ZMAX1;i++){
        for(j=0;j<NMAX1;j++){
            for(k=0;k<ISMR1;k++){
                fission_yield[i][j][k]=0.0;
            }
        }
    }
    i=0;
    sum=0.0;
    fp=fopen("235ut.dat","rt");
    if(fp==NULL){
        fprintf(stderr,"fission yield data file cannot be opened.\n");
        exit(1);
    }
    while(fgets(buff,1024,fp) != NULL){
        if(buff[0] == '#') {continue;} // regarded as a comment line
        sscanf(buff,"%d %d %d %d %le",&i1,&i2,&i3,&i4,&r1);
        if(i1<0 || i2<0 || i3<0 || i1!=i2+i3 || i1>400 || i4<0 || i4>10 || r1<0.0){

```

```

        fprintf(stderr,"error in reading a line:\n%s",buff);
        exit(1);
    }
    if(i2>=ZMAX1 || i3>=NMAX1 || i4>=ISMR1){
        fprintf(stderr,"to small array sizes for a data line:\n%s",buff);
        exit(1);
    }
    r1*=0.01; // original data are for 100 fissions. Now, for 1 fission.
    fission_yield[i2][i3][i4]+=r1;
    sum+=r1;
    i++;
}
fclose(fp);
if(debug>0) printf("#fission yield data: number=%d sum=%.6f \n",i,sum);

// modifications for neutron emission
#include "set_initial_state_140.c"

for(j=0;j<nn;j++){
    z=atomic_number[j];
    n=mass_number[j]-z;
    i=state_id[j];
    if(z>=ZMAX1 || n>=NMAX1) exit(1);
    n0[j]=fission_yield[z][n][i];
}

// check whether there are any decay products which are not taken into account
// in the decay chain of mass number A

for(z=0;z<ZMAX1;z++){
    n=A-z;
    if(n<0) continue;
    for(i=0;i<ISMR1;i++){
        if(fission_yield[z][n][i]<=0.0) continue;
        for(j=0;j<nn;j++){
            if(mass_number[j]==A && atomic_number[j]==z && state_id[j]==i)goto L1;
        }
        printf("Warning: A=%d Z=%d state=%d yield=%e is not considered in decay chain\n",
            z+n,z,i,fission_yield[z][n][i]);
        L1;;
    }
}

return 0;
}

set_intial_95
//----- modifications for neutron emission : begin -----

fission_yield[42][67][0]+=fission_yield[41][68][0];

```

```

fission_yield[41][68][0]=0.0;

fission_yield[43][66][0]+=fission_yield[42][67][0]*0.987;
fission_yield[43][65][0]+=fission_yield[42][67][0]*0.013;
fission_yield[42][67][0]=0.0;

fission_yield[45][64][0]+=fission_yield[45][64][1];
fission_yield[45][64][1]=0.0;

fission_yield[41][67][0]+=fission_yield[40][68][0];
fission_yield[40][68][0]=0.0;

fission_yield[42][66][0]+=fission_yield[41][67][0]*0.92;
fission_yield[42][65][0]+=fission_yield[41][67][0]*0.08;
fission_yield[41][67][0]=0.0;

fission_yield[42][65][0]+=fission_yield[41][66][0]*0.92;
fission_yield[42][64][0]+=fission_yield[41][66][0]*0.08;
fission_yield[41][66][0]=0.0;

fission_yield[41][65][0]+=fission_yield[40][66][0]*0.93;
fission_yield[41][64][0]+=fission_yield[40][66][0]*0.07;
fission_yield[40][66][0]=0.0;

fission_yield[42][64][0]+=fission_yield[41][65][0]*0.955;
fission_yield[42][63][0]+=fission_yield[41][65][0]*0.045;
fission_yield[41][65][0]=0.0;

fission_yield[41][64][0]+=fission_yield[40][65][0]*0.98;
fission_yield[41][63][0]+=fission_yield[40][65][0]*0.02;
fission_yield[40][65][0]=0.0;

fission_yield[42][63][0]+=fission_yield[41][64][0]*0.983;
fission_yield[42][62][0]+=fission_yield[41][64][0]*0.017;
fission_yield[41][64][0]=0.0;

fission_yield[39][65][0]+=fission_yield[38][66][0];
fission_yield[38][66][0]=0.0;

fission_yield[40][64][0]+=fission_yield[39][65][0];
fission_yield[39][65][0]=0.0;

fission_yield[40][63][0]+=fission_yield[39][64][0]*0.92;
fission_yield[40][62][0]+=fission_yield[39][64][0]*0.08;
fission_yield[39][64][0]=0.0;

fission_yield[38][64][0]+=fission_yield[37][65][0]*0.82;
fission_yield[38][63][0]+=fission_yield[37][65][0]*0.18;
fission_yield[37][65][0]=0.0;

```

```

fission_yield[39][63][0] += fission_yield[38][64][0] * 0.945;
fission_yield[39][62][0] += fission_yield[38][64][0] * 0.055;
fission_yield[38][64][0] = 0.0;

fission_yield[40][62][0] += fission_yield[39][63][1] * 0.951;
fission_yield[40][61][0] += fission_yield[39][63][1] * 0.049;
fission_yield[39][63][1] = 0.0;

fission_yield[40][62][0] += fission_yield[39][63][0] * 0.951;
fission_yield[40][61][0] += fission_yield[39][63][0] * 0.049;
fission_yield[39][63][0] = 0.0;

fission_yield[38][63][0] += fission_yield[37][64][0] * 0.72;
fission_yield[38][62][0] += fission_yield[37][64][0] * 0.28;
fission_yield[37][64][0] = 0.0;

fission_yield[39][62][0] += fission_yield[38][63][0] * 0.9763;
fission_yield[39][61][0] += fission_yield[38][63][0] * 0.0237;
fission_yield[38][63][0] = 0.0;

fission_yield[40][61][0] += fission_yield[39][62][0] * 0.9806;
fission_yield[40][60][0] += fission_yield[39][62][0] * 0.0194;
fission_yield[39][62][0] = 0.0;

fission_yield[37][63][0] += fission_yield[36][64][0];
fission_yield[36][64][0] = 0.0;

fission_yield[38][62][0] += fission_yield[37][63][0] * 0.9384;
fission_yield[38][61][0] += fission_yield[37][63][0] * 0.06;
fission_yield[38][60][0] += fission_yield[37][63][0] * 0.0016;
fission_yield[37][63][0] = 0.0;

fission_yield[39][61][0] += fission_yield[38][62][0] * 0.9922;
fission_yield[39][60][0] += fission_yield[38][62][0] * 0.0078;
fission_yield[38][62][0] = 0.0;

fission_yield[40][60][0] += fission_yield[39][61][1];
fission_yield[39][61][1] = 0.0;

fission_yield[40][60][0] += fission_yield[39][61][0] * 0.9908;
fission_yield[40][59][0] += fission_yield[39][61][0] * 0.0092;
fission_yield[39][61][0] = 0.0;

fission_yield[38][61][0] += fission_yield[37][62][0] * 0.842;
fission_yield[38][60][0] += fission_yield[37][62][0] * 0.158;
fission_yield[37][62][0] = 0.0;

fission_yield[39][60][0] += fission_yield[38][61][0] * 0.999;
fission_yield[39][59][0] += fission_yield[38][61][0] * 0.001;

```

```

fission_yield[38][61][0]=0.0;

fission_yield[40][59][0]+=fission_yield[39][60][0]*0.983;
fission_yield[40][58][0]+=fission_yield[39][60][0]*0.017;
fission_yield[39][60][0]=0.0;

fission_yield[37][61][0]+=fission_yield[36][62][0]*0.93;
fission_yield[37][60][0]+=fission_yield[36][62][0]*0.07;
fission_yield[36][62][0]=0.0;

fission_yield[38][60][0]+=fission_yield[37][61][1];
fission_yield[37][61][1]=0.0;

fission_yield[38][60][0]+=fission_yield[37][61][0]*0.862;
fission_yield[38][59][0]+=fission_yield[37][61][0]*0.138;
fission_yield[37][61][0]=0.0;

fission_yield[39][59][0]+=fission_yield[38][60][0]*0.9975;
fission_yield[39][58][0]+=fission_yield[38][60][0]*0.0025;
fission_yield[38][60][0]=0.0;

fission_yield[40][58][0]+=fission_yield[39][59][1]*0.966;
fission_yield[40][57][0]+=fission_yield[39][59][1]*0.034;
fission_yield[39][59][1]=0.0;

fission_yield[40][58][0]+=fission_yield[39][59][0]*0.9967;
fission_yield[40][57][0]+=fission_yield[39][59][0]*0.0033;
fission_yield[39][59][0]=0.0;

fission_yield[37][60][0]+=fission_yield[36][61][0]*0.933;
fission_yield[37][59][0]+=fission_yield[36][61][0]*0.067;
fission_yield[36][61][0]=0.0;

fission_yield[38][59][0]+=fission_yield[37][60][0]*0.745;
fission_yield[38][58][0]+=fission_yield[37][60][0]*0.255;
fission_yield[37][60][0]=0.0;

fission_yield[35][61][0]+=fission_yield[34][62][0];
fission_yield[34][62][0]=0.0;

fission_yield[36][60][0]+=fission_yield[35][61][0]*0.724;
fission_yield[36][59][0]+=fission_yield[35][61][0]*0.276;
fission_yield[35][61][0]=0.0;

fission_yield[37][59][0]+=fission_yield[36][60][0]*0.963;
fission_yield[37][58][0]+=fission_yield[36][60][0]*0.037;
fission_yield[36][60][0]=0.0;

fission_yield[38][58][0]+=fission_yield[37][59][0]*0.867;
fission_yield[38][57][0]+=fission_yield[37][59][0]*0.133;

```

```

fission_yield[37][59][0]=0.0;

fission_yield[37][58][0]+=fission_yield[36][59][0]*0.9713;
fission_yield[37][57][0]+=fission_yield[36][59][0]*0.0287;
fission_yield[36][59][0]=0.0;

fission_yield[38][57][0]+=fission_yield[37][58][0]*0.913;
fission_yield[38][56][0]+=fission_yield[37][58][0]*0.087;
fission_yield[37][58][0]=0.0;

fission_yield[35][59][0]+=fission_yield[34][60][0];
fission_yield[34][60][0]=0.0;

fission_yield[36][58][0]+=fission_yield[35][59][0]*0.32;
fission_yield[36][57][0]+=fission_yield[35][59][0]*0.68;
fission_yield[35][59][0]=0.0;

fission_yield[37][57][0]+=fission_yield[36][58][0]*0.9889;
fission_yield[37][56][0]+=fission_yield[36][58][0]*0.0111;
fission_yield[36][58][0]=0.0;

fission_yield[38][56][0]+=fission_yield[37][57][0]*0.895;
fission_yield[38][55][0]+=fission_yield[37][57][0]*0.105;
fission_yield[37][57][0]=0.0;

fission_yield[36][57][0]+=fission_yield[35][58][0]*0.32;
fission_yield[36][56][0]+=fission_yield[35][58][0]*0.68;
fission_yield[35][58][0]=0.0;

fission_yield[37][56][0]+=fission_yield[36][57][0]*0.9805;
fission_yield[37][55][0]+=fission_yield[36][57][0]*0.0195;
fission_yield[36][57][0]=0.0;

fission_yield[38][55][0]+=fission_yield[37][56][0]*0.9861;
fission_yield[38][54][0]+=fission_yield[37][56][0]*0.0139;
fission_yield[37][56][0]=0.0;

fission_yield[35][57][0]+=fission_yield[34][58][0];
fission_yield[34][58][0]=0.0;

fission_yield[36][56][0]+=fission_yield[35][57][0]*0.669;
fission_yield[36][55][0]+=fission_yield[35][57][0]*0.331;
fission_yield[35][57][0]=0.0;

fission_yield[35][56][0]+=fission_yield[34][57][0]*0.79;
fission_yield[35][55][0]+=fission_yield[34][57][0]*0.21;
fission_yield[34][57][0]=0.0;

fission_yield[36][55][0]+=fission_yield[35][56][0]*0.80;
fission_yield[36][54][0]+=fission_yield[35][56][0]*0.20;

```

```

fission_yield[35][56][0]=0.0;

fission_yield[35][55][0]+=fission_yield[34][56][0];
fission_yield[34][56][0]=0.0;

fission_yield[36][54][0]+=fission_yield[35][55][0]*0.748;
fission_yield[36][53][0]+=fission_yield[35][55][0]*0.252;
fission_yield[35][55][0]=0.0;

fission_yield[34][55][0]+=fission_yield[33][56][0];
fission_yield[33][56][0]=0.0;

fission_yield[35][54][0]+=fission_yield[34][55][0]*0.922;
fission_yield[35][53][0]+=fission_yield[34][55][0]*0.078;
fission_yield[34][55][0]=0.0;

fission_yield[36][53][0]+=fission_yield[35][54][0]*0.862;
fission_yield[36][52][0]+=fission_yield[35][54][0]*0.138;
fission_yield[35][54][0]=0.0;

fission_yield[33][55][0]+=fission_yield[32][56][0];
fission_yield[32][56][0]=0.0;

fission_yield[34][54][0]+=fission_yield[33][55][0];
fission_yield[33][55][0]=0.0;

fission_yield[35][53][0]+=fission_yield[34][54][0]*0.9933;
fission_yield[35][52][0]+=fission_yield[34][54][0]*0.0067;
fission_yield[34][54][0]=0.0;

fission_yield[36][52][0]+=fission_yield[35][53][0]*0.9342;
fission_yield[36][51][0]+=fission_yield[35][53][0]*0.0658;
fission_yield[35][53][0]=0.0;

fission_yield[33][54][0]+=fission_yield[32][55][0];
fission_yield[32][55][0]=0.0;

fission_yield[34][53][0]+=fission_yield[33][54][0]*0.846;
fission_yield[34][52][0]+=fission_yield[33][54][0]*0.154;
fission_yield[33][54][0]=0.0;

fission_yield[35][52][0]+=fission_yield[34][53][0]*0.998;
fission_yield[35][51][0]+=fission_yield[34][53][0]*0.002;
fission_yield[34][53][0]=0.0;

fission_yield[36][51][0]+=fission_yield[35][52][0]*0.974;
fission_yield[36][50][0]+=fission_yield[35][52][0]*0.026;
fission_yield[35][52][0]=0.0;

fission_yield[31][55][0]+=fission_yield[30][56][0];

```

```

fission_yield[30][56][0]=0.0;

fission_yield[32][54][0]+=fission_yield[31][55][0];
fission_yield[31][55][0]=0.0;

fission_yield[33][53][0]+=fission_yield[32][54][0];
fission_yield[32][54][0]=0.0;

fission_yield[34][52][0]+=fission_yield[33][53][0]*0.74;
fission_yield[34][51][0]+=fission_yield[33][53][0]*0.26;
fission_yield[33][53][0]=0.0;

fission_yield[35][51][0]+=fission_yield[34][52][0];
fission_yield[34][52][0]=0.0;

fission_yield[36][50][0]+=fission_yield[35][51][1]*0.88;
fission_yield[36][49][0]+=fission_yield[35][51][1]*0.12;
fission_yield[35][51][1]=0.0;

fission_yield[32][53][0]+=fission_yield[31][54][0]*0.65;
fission_yield[32][52][0]+=fission_yield[31][54][0]*0.35;
fission_yield[31][54][0]=0.0;

fission_yield[33][52][0]+=fission_yield[32][53][0]*0.86;
fission_yield[33][51][0]+=fission_yield[32][53][0]*0.14;
fission_yield[32][53][0]=0.0;

fission_yield[34][51][0]+=fission_yield[33][52][0]*0.406;
fission_yield[34][50][0]+=fission_yield[33][52][0]*0.594;
fission_yield[33][52][0]=0.0;

fission_yield[35][50][0]+=fission_yield[34][51][0];
fission_yield[34][51][0]=0.0;

fission_yield[35][50][0]+=fission_yield[34][51][1]*0.77;
fission_yield[35][49][0]+=fission_yield[34][51][1]*0.23;
fission_yield[34][51][1]=0.0;

fission_yield[31][53][0]+=fission_yield[30][54][0];
fission_yield[30][54][0]=0.0;

fission_yield[32][52][0]+=fission_yield[31][53][0]*0.26;
fission_yield[32][51][0]+=fission_yield[31][53][0]*0.74;
fission_yield[31][53][0]=0.0;

fission_yield[33][51][0]+=fission_yield[32][52][0]*0.898;
fission_yield[33][50][0]+=fission_yield[32][52][0]*0.102;
fission_yield[32][52][0]=0.0;

fission_yield[31][52][0]+=fission_yield[30][53][0];

```



```

fission_yield[30][53][0]=0.0;

fission_yield[32][51][0]+=fission_yield[31][52][0]*0.372;
fission_yield[32][50][0]+=fission_yield[31][52][0]*0.628;
fission_yield[31][52][0]=0.0;

fission_yield[30][52][0]+=fission_yield[29][53][0];
fission_yield[29][53][0]=0.0;

fission_yield[31][51][0]+=fission_yield[30][52][0];
fission_yield[30][52][0]=0.0;

fission_yield[32][50][0]+=fission_yield[31][51][0]*0.802;
fission_yield[32][49][0]+=fission_yield[31][51][0]*0.198;
fission_yield[31][51][0]=0.0;

fission_yield[31][50][0]+=fission_yield[30][51][0]*0.925;
fission_yield[31][49][0]+=fission_yield[30][51][0]*0.075;
fission_yield[30][51][0]=0.0;

fission_yield[32][49][0]+=fission_yield[31][50][0]*0.881;
fission_yield[32][48][0]+=fission_yield[31][50][0]*0.119;
fission_yield[31][50][0]=0.0;

fission_yield[29][51][0]+=fission_yield[28][52][0];
fission_yield[28][52][0]=0.0;

fission_yield[30][50][0]+=fission_yield[29][51][0];
fission_yield[29][51][0]=0.0;

fission_yield[31][49][0]+=fission_yield[30][50][0]*0.99;
fission_yield[31][48][0]+=fission_yield[30][50][0]*0.01;
fission_yield[30][50][0]=0.0;

fission_yield[32][48][0]+=fission_yield[31][49][0]*0.9914;
fission_yield[32][47][0]+=fission_yield[31][49][0]*0.0086;
fission_yield[31][49][0]=0.0;

fission_yield[30][49][0]+=fission_yield[29][50][0]*0.45;
fission_yield[30][48][0]+=fission_yield[29][50][0]*0.55;
fission_yield[29][50][0]=0.0;

fission_yield[31][48][0]+=fission_yield[30][49][0]*0.987;
fission_yield[31][47][0]+=fission_yield[30][49][0]*0.013;
fission_yield[30][49][0]=0.0;

//----- modifications for neutron emission : end -----
set_intial_140
//----- modifications for neutron emission : begin -----

```

```

// A Z N el      A Z N el raito
// 128 46 82 Pb -> 127 47 80 Ag 1
fission_yield[47][80][0]+=fission_yield[46][82][0];
fission_yield[46][82][0]=0.0;
// 130 46 84 Pd -> 129 47 82 Ag 1
fission_yield[47][82][0]+=fission_yield[46][84][0];
fission_yield[46][84][0]=0.0;
// 129 47 82 Ag -> 129 48 81 Cd 1
fission_yield[48][81][0]+=fission_yield[47][82][0];
fission_yield[47][82][0]=0.0;
// 130 47 83 Ag -> 129 48 81 Cd 1
fission_yield[48][81][0]+=fission_yield[47][83][0];
fission_yield[47][83][0]=0.0;
// A Z N el      A Z N el ratio
// 129 48 81 Cd -> 129 49 80 In 1
fission_yield[49][80][0]+=fission_yield[48][81][0];
fission_yield[48][81][0]=0.0;
// 129 49 80 In1 -> 128 50 78 Sn 0.025
// 129 49 80 In1 -> 129 50 79 Sn 0.975
fission_yield[50][78][0]+=fission_yield[49][80][1]*0.025;
fission_yield[50][79][0]+=fission_yield[49][80][1]*0.975;
fission_yield[49][80][1]=0.0;

// A Z N el      A Z N el ratio
// 131 48 83 Cd -> 130 49 81 In 0.035
// 131 48 83 Cd -> 131 49 82 In 0.965
fission_yield[49][81][0]+=fission_yield[48][83][0]*0.035;
fission_yield[49][82][0]+=fission_yield[48][83][0]*0.965;
fission_yield[48][83][0]=0.0;
// 130 48 82 Cd -> 130 49 81 In 0.965
// 130 48 82 Cd -> 129 49 80 In 0.035
fission_yield[49][81][0]+=fission_yield[48][82][0]*0.965;
fission_yield[49][80][0]+=fission_yield[48][82][0]*0.035;
fission_yield[48][82][0]=0.0;
// 130 49 81 In -> 130 50 80 Sn 0.9907
// 130 49 81 In -> 129 50 79 Sn 0.0093
fission_yield[50][80][0]+=fission_yield[49][81][0]*0.9907;
fission_yield[50][79][0]+=fission_yield[49][81][0]*0.0093;
fission_yield[49][81][0]=0.0;
// 130 49 81 In1 -> 130 50 80 Sn 0.9835
// 130 49 81 In1 -> 129 50 79 Sn 0.0165
fission_yield[50][80][0]+=fission_yield[49][81][1]*0.9835;
fission_yield[50][79][0]+=fission_yield[49][81][1]*0.0165;
fission_yield[49][81][1]=0.0;
// 130 49 81 In2 -> 130 50 80 Sn 0.9835
// 130 49 81 In2 -> 129 50 79 Sn 0.0165
fission_yield[50][80][0]+=fission_yield[49][81][2]*0.9835;
fission_yield[50][79][0]+=fission_yield[49][81][2]*0.0165;
fission_yield[49][81][2]=0.0;

```

```

// A   Z   N el      A   Z   N el   ratio
// 131 49 82 In -> 131 50 81 Sn 0.98
// 131 49 82 In -> 130 50 80 Sn 0.02
fission_yield[50][81][0]+=fission_yield[49][82][0]*0.98;
fission_yield[50][80][0]+=fission_yield[49][82][0]*0.02;
fission_yield[49][82][0]=0.0;

// A   Z   N el      A   Z   N el   ratio
// 132 49 83 In -> 132 50 82 Sn 0.937
// 132 49 83 In -> 131 50 81 Sn 0.063
fission_yield[50][82][0]+=fission_yield[49][83][0]*0.937;
fission_yield[50][81][0]+=fission_yield[49][83][0]*0.063;
fission_yield[49][83][0]=0.0;

// A   Z   N el      A   Z   N el   ratio
// 134 48 86 Cd -> 133 49 84 In 1.0
fission_yield[49][84][0]+=fission_yield[48][86][0];
fission_yield[48][86][0]=0.0;
// 133 49 84 In -> 133 50 83 Sn 0.15
// 133 49 84 In -> 132 50 82 Sn 0.85
fission_yield[50][83][0]+=fission_yield[49][84][0]*0.15;
fission_yield[50][82][0]+=fission_yield[49][84][0]*0.85;
fission_yield[49][84][0]=0.0;

// A   Z   N el      A   Z   N el   ratio
// 134 49 85 In -> 134 50 84 Sn 0.35
// 134 49 85 In -> 133 50 83 Sn 0.65
fission_yield[50][84][0]+=fission_yield[49][85][0]*0.35;
fission_yield[50][83][0]+=fission_yield[49][85][0]*0.65;
fission_yield[49][85][0]=0.0;
// 134 50 84 Sn -> 134 51 83 Sb 0.83
// 134 50 84 Sn -> 133 51 82 Sb 0.17
fission_yield[51][83][0]+=fission_yield[50][84][0]*0.83;
fission_yield[51][82][0]+=fission_yield[50][84][0]*0.17;
fission_yield[50][84][0]=0.0;

// A   Z   N el      A   Z   N el   ratio
// 136 49 87 In -> 136 50 86 Sn 0.5
// 136 49 87 In -> 135 50 85 Sn 0.5
fission_yield[50][86][0]+=fission_yield[49][87][0]*0.5;
fission_yield[50][85][0]+=fission_yield[49][87][0]*0.5;
fission_yield[49][87][0]=0.0;
// 136 50 86 Sn -> 135 51 84 Sb 0.3
// 136 50 86 Sn -> 136 51 85 Sb 0.7
fission_yield[51][84][0]+=fission_yield[50][86][0]*0.3;
fission_yield[51][85][0]+=fission_yield[50][86][0]*0.7;
fission_yield[50][86][0]=0.0;
// 135 50 85 Sn -> 135 51 84 Sb 0.79
// 135 50 85 Sn -> 134 51 83 Sb 0.21
fission_yield[51][84][0]+=fission_yield[50][85][0]*0.79;

```

```

fission_yield[51][83][0]+=fission_yield[50][85][0]*0.21;
fission_yield[50][85][0]=0.0;
// 135 51 84 Sb -> 135 52 83 Te 0.78
// 135 51 84 Sb -> 134 52 82 Te 0.22
fission_yield[52][83][0]+=fission_yield[51][84][0]*0.78;
fission_yield[52][82][0]+=fission_yield[51][84][0]*0.22;
fission_yield[51][84][0]=0.0;

// A Z N el      A Z N el ratio
// 138 50 88 Sn -> 137 51 86 Sb 1.0
fission_yield[51][86][0]+=fission_yield[50][88][0];
fission_yield[50][88][0]=0.0;
// 137 50 87 Sn -> 136 51 85 Sb 0.58
// 137 50 87 Sn -> 137 51 86 Sb 0.42
fission_yield[51][85][0]+=fission_yield[50][87][0]*0.58;
fission_yield[51][86][0]+=fission_yield[50][87][0]*0.42;
fission_yield[50][87][0]=0.0;
// 137 51 86 Sb -> 136 52 84 Te 0.49
// 137 51 86 Sb -> 137 52 85 Te 0.51
fission_yield[52][84][0]+=fission_yield[51][86][0]*0.49;
fission_yield[52][85][0]+=fission_yield[51][86][0]*0.51;
fission_yield[51][86][0]=0.0;

// 136 51 85 Sb -> 136 52 84 Te 0.837
// 136 51 85 Sb -> 135 52 83 Te 0.163
fission_yield[52][84][0]+=fission_yield[51][85][0]*0.837;
fission_yield[52][83][0]+=fission_yield[51][85][0]*0.163;
fission_yield[51][85][0]=0.0;
// 136 52 84 Te -> 136 53 83 I 0.9869
// 136 52 84 Te -> 135 53 82 I 0.0131
fission_yield[53][83][0]+=fission_yield[52][84][0]*0.9869;
fission_yield[53][82][0]+=fission_yield[52][84][0]*0.0131;
fission_yield[52][84][0]=0.0;

// A Z N el      A Z N el ratio
// 138 51 87 Sb -> 138 52 86 Te 0.28
// 138 51 87 Sb -> 137 52 85 Te 0.72
fission_yield[52][86][0]+=fission_yield[51][87][0]*0.28;
fission_yield[52][85][0]+=fission_yield[51][87][0]*0.72;
fission_yield[51][87][0]=0.0;
// 139 51 88 Sb -> 139 52 87 Te 0.1
// 139 51 88 Sb -> 138 52 86 Te 0.9
fission_yield[52][87][0]+=fission_yield[51][88][0]*0.9;
fission_yield[52][86][0]+=fission_yield[51][88][0]*0.1;
fission_yield[51][88][0]=0.0;
// 138 52 86 Te -> 138 53 84 I 0.063
// 138 52 86 Te -> 138 53 85 I 0.937
fission_yield[53][84][0]+=fission_yield[52][86][0]*0.063;
fission_yield[53][85][0]+=fission_yield[52][86][0]*0.937;
fission_yield[52][86][0]=0.0;

```

```

// 137 52 85 Te -> 137 53 84 I 0.9701
// 137 52 85 Te -> 136 53 83 I 0.0299
fission_yield[53][84][0]+=fission_yield[52][85][0]*0.9701;
fission_yield[53][83][0]+=fission_yield[52][85][0]*0.0299;
fission_yield[52][85][0]=0.0;
// 137 53 84 I -> 137 54 83 Xe 0.9286
// 137 53 84 I -> 136 54 82 Xe 0.0714
fission_yield[54][83][0]+=fission_yield[53][84][0]*0.9286;
fission_yield[54][82][0]+=fission_yield[53][84][0]*0.0714;
fission_yield[53][84][0]=0.0;

// A Z N el      A Z N el ratio
// 139 52 87 Te -> 138 53 85 I 1.0
fission_yield[53][85][0]+=fission_yield[52][87][0];
fission_yield[52][87][0]=0.0;
// 138 53 85 I -> 138 54 84 Xe 0.9444
// 138 53 85 I -> 137 54 83 Xe 0.0556
fission_yield[54][84][0]+=fission_yield[53][85][0]*0.9444;
fission_yield[54][83][0]+=fission_yield[53][85][0]*0.0556;
fission_yield[53][85][0]=0.0;

// A Z N el      A Z N el ratio
// 140 52 88 Te -> 139 53 86 I 1.0
fission_yield[53][86][0]+=fission_yield[52][88][0];
fission_yield[52][88][0]=0.0;
// 139 53 86 I -> 139 54 85 Xe 0.9
// 139 53 86 I -> 138 54 84 Xe 0.1
fission_yield[54][85][0]+=fission_yield[53][86][0]*0.9;
fission_yield[54][84][0]+=fission_yield[53][86][0]*0.1;
fission_yield[53][86][0]=0.0;

// A Z N el      A Z N el ratio
// 140 53 87 I -> 140 54 86 Xe 0.907
// 140 53 87 I -> 139 54 85 Xe 0.093
fission_yield[54][86][0]+=fission_yield[53][87][0]*0.907;
fission_yield[54][85][0]+=fission_yield[53][87][0]*0.093;
fission_yield[53][87][0]=0.0;

// A Z N el      A Z N el ratio
// 141 53 88 I -> 141 54 87 Xe 0.788
// 141 53 88 I -> 140 54 86 Xe 0.212
fission_yield[54][87][0]+=fission_yield[53][88][0]*0.788;
fission_yield[54][86][0]+=fission_yield[53][88][0]*0.212;
fission_yield[53][88][0]=0.0;

// A Z N el      A Z N el ratio
// 142 52 90 Te -> 142 53 89 I 1.0
fission_yield[53][89][0]+=fission_yield[52][90][0];
fission_yield[52][90][0]=0.0;

```

```

// A   Z   N e l       A   Z   N e l   r a t i o
// 144 54 90 Xe -> 143 55 88 Cs 0.03
// 144 54 90 Xe -> 144 55 89 Cs 0.97
fission_yield[55][88][0]+=fission_yield[54][90][0]*0.03;
fission_yield[55][89][0]+=fission_yield[54][90][0]*0.97;
fission_yield[54][90][0]=0.0;
// 143 54 89 Xe -> 143 55 88 Cs 0.99
// 143 54 89 Xe -> 142 55 87 Cs 0.01
fission_yield[55][88][0]+=fission_yield[54][89][0]*0.99;
fission_yield[55][87][0]+=fission_yield[54][89][0]*0.01;
fission_yield[54][89][0]=0.0;
// 143 55 88 Cs -> 143 56 87 Ba 0.9836
// 143 55 88 Cs -> 142 56 86 Ba 0.0164
fission_yield[56][87][0]+=fission_yield[55][88][0]*0.9836;
fission_yield[56][86][0]+=fission_yield[55][88][0]*0.0164;
fission_yield[55][88][0]=0.0;

// A   Z   N e l       A   Z   N e l   r a t i o
// 146 54 92 Xe -> 145 55 90 Cs 0.069
// 146 54 92 Xe -> 146 55 91 Cs 0.931
fission_yield[55][90][0]+=fission_yield[54][92][0]*0.069;
fission_yield[55][91][0]+=fission_yield[54][92][0]*0.931;
fission_yield[54][92][0]=0.0;
// 145 54 91 Xe -> 144 55 89 Cs 0.05
// 145 54 91 Xe -> 145 55 90 Cs 0.95
fission_yield[55][89][0]+=fission_yield[54][91][0]*0.05;
fission_yield[55][90][0]+=fission_yield[54][91][0]*0.95;
fission_yield[54][91][0]=0.0;
// 145 55 90 Cs -> 144 56 88 Ba 0.147
// 145 55 90 Cs -> 145 56 89 Ba 0.853
fission_yield[56][88][0]+=fission_yield[55][90][0]*0.147;
fission_yield[56][89][0]+=fission_yield[55][90][0]*0.853;
fission_yield[55][90][0]=0.0;

// 144 55 89 Cs -> 144 56 88 Ba 0.9697
// 144 55 89 Cs -> 143 56 87 Ba 0.0303
fission_yield[56][88][0]+=fission_yield[55][89][0]*0.9697;
fission_yield[56][87][0]+=fission_yield[55][89][0]*0.0303;
fission_yield[55][89][0]=0.0;
// 144 56 88 Ba -> 144 57 87 La 0.964
// 144 56 88 Ba -> 143 57 86 La 0.036
fission_yield[57][87][0]+=fission_yield[56][88][0]*0.964;
fission_yield[57][86][0]+=fission_yield[56][88][0]*0.036;
fission_yield[56][88][0]=0.0;

// A   Z   N e l       A   Z   N e l   r a t i o
// 146 55 91 Cs -> 146 56 90 Ba 0.858
// 146 55 91 Cs -> 145 56 89 Ba 0.142
fission_yield[56][90][0]+=fission_yield[55][91][0]*0.858;

```

```

fission_yield[56][89][0]+=fission_yield[55][91][0]*0.142;
fission_yield[55][91][0]=0.0;

// A   Z   N el      A   Z   N el   ratio
// 148 54 94 Xe ->  147 55 92 Cs   1
fission_yield[55][92][0]+=fission_yield[54][94][0];
fission_yield[54][94][0]=0.0;
// 147 55 92 Cs ->  147 56 91 Ba   0.715
// 147 55 92 Cs ->  146 56 90 Ba   0.285
fission_yield[56][91][0]+=fission_yield[55][92][0]*0.715;
fission_yield[56][90][0]+=fission_yield[55][92][0]*0.285;
fission_yield[55][92][0]=0.0;

// A   Z   N el      A   Z   N el   ratio

// 148 55 93 Cs ->  148 56 92 Ba   0.749
// 148 55 93 Cs ->  147 56 91 Ba   0.251
fission_yield[56][92][0]+=fission_yield[55][93][0]*0.749;
fission_yield[56][91][0]+=fission_yield[55][93][0]*0.251;
fission_yield[55][93][0]=0.0;

// A   Z   N el      A   Z   N el   ratio
// 150 55 95 Cs ->  149 56 93 Ba   1.0
fission_yield[56][93][0]+=fission_yield[55][95][0]*1.0;
fission_yield[55][95][0]=0.0;
// 149 55 94 Cs ->  148 56 92 Ba   1.0
fission_yield[56][92][0]+=fission_yield[55][94][0]*1.0;
fission_yield[55][94][0]=0.0;
// 149 56 93 Ba ->  149 57 92 La   0.9957
// 149 56 93 Ba ->  148 57 91 La   0.0043
fission_yield[57][92][0]+=fission_yield[56][93][0]*0.9957;
fission_yield[57][91][0]+=fission_yield[56][93][0]*0.0043;
fission_yield[56][93][0]=0.0;
// 149 57 92 La ->  149 58 91 Ce   0.9857
// 149 57 92 La ->  148 58 90 Ce   0.0143
fission_yield[58][91][0]+=fission_yield[57][92][0]*0.9857;
fission_yield[58][90][0]+=fission_yield[57][92][0]*0.0143;
fission_yield[57][92][0]=0.0;

// A   Z   N el      A   Z   N el   ratio

// 151 55 96 Cs ->  150 56 94 Ba   1.0
fission_yield[56][94][0]+=fission_yield[55][96][0]*1.0;
fission_yield[55][96][0]=0.0;
// 151 56 95 Ba ->  150 57 93 La   1.0
fission_yield[57][93][0]+=fission_yield[56][95][0]*1.0;
fission_yield[56][95][0]=0.0;
// 150 56 94 Ba ->  149 57 93 La   1.0
fission_yield[57][93][0]+=fission_yield[56][94][0]*1.0;
fission_yield[56][94][0]=0.0;

```

```

// 150 57 93 La -> 150 58 92 Ce 0.973
// 150 57 93 La -> 149 58 91 Ce 0.027
fission_yield[58][92][0]+=fission_yield[57][93][0]*0.973;
fission_yield[58][91][0]+=fission_yield[57][93][0]*0.027;
fission_yield[57][93][0]=0.0;

// A Z N el      A Z N el ratio
// 152 56 96 Ba -> 151 57 94 Ba 1.0
fission_yield[57][94][0]+=fission_yield[56][96][0]*1.0;
fission_yield[56][96][0]=0.0;

// 151 57 94 La -> 150 58 92 Ce 1.0
fission_yield[58][92][0]+=fission_yield[57][94][0]*1.0;
fission_yield[57][94][0]=0.0;

// A Z N el      A Z N el ratio
// 154 57 97 La -> 154 58 96 Ce 1.0
fission_yield[58][96][0]+=fission_yield[57][97][0]*1.0;
fission_yield[57][97][0]=0.0;
// 154 58 96 La -> 154 59 95 Pr 1.0
fission_yield[59][95][0]+=fission_yield[58][96][0]*1.0;
fission_yield[58][96][0]=0.0;

//----- modifications for neutron emission : end -----

#include "set_lambda_95.c"

int set_lambda_79(int A){
    const int A0=79;
    enum{Ga,Ge1,Ge,As,Se1,Se,Br1,Br,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ga, A, 31, 0, 2.847,sec);
    set_halflife(Ge1, A, 32, 1, 39.0,sec);
    set_halflife(Ge, A, 32, 0, 18.98,sec);
    set_halflife(As, A, 33, 0, 9.01,min);
    set_halflife(Se1, A, 34, 1, 3.92,min);
    set_halflife(Se, A, 34, 0, 2.95e+5,year);
    set_halflife(Br1, A, 35, 1, 5.1,sec);
    set_halflife(Br, A, 35, 0, BIGNUM,sec);

    lambda[Ge][Ga]=-lambda[Ga][Ga];
    lambda[Ge][Ge1]=-lambda[Ge1][Ge1]*0.04;
    lambda[As][Ge1]=-lambda[Ge1][Ge1]*0.96;
    lambda[As][Ge]=-lambda[Ge][Ge];

```



```

    lambda[Se][As] = -lambda[As][As]*0.021;
    lambda[Se1][As] = -lambda[As][As]*0.979;
    lambda[Se][Se1] = -lambda[Se1][Se1]*0.9994;
    lambda[Br][Se1] = -lambda[Se1][Se1]*0.0006;
    lambda[Br][Se] = -lambda[Se][Se];
    lambda[Br][Br1] = -lambda[Br1][Br1];

    return 0;
}

int set_lambda_80(int A){
    const int A0=80;
    enum{Ge,As,Br1,Br,Se,Kr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ge, A, 32, 0, 29.5,sec);
    set_halflife(As, A, 33, 0, 15.2,sec);
    set_halflife(Se, A, 34, 0, BIGNUM,sec);
    set_halflife(Br1, A, 35, 1, 4.4205,hour);
    set_halflife(Br, A, 35, 0, 17.68,min);
    set_halflife(Kr, A, 36, 0, BIGNUM,sec);

    lambda[As][Ge] = -lambda[Ge][Ge];
    lambda[Se][As] = -lambda[As][As];
    lambda[Br][Br1] = -lambda[Br1][Br1];
    lambda[Se][Br] = -lambda[Br][Br]*0.083;
    lambda[Kr][Br] = -lambda[Br][Br]*0.917;

    return 0;
}

int set_lambda_81(int A){
    const int A0=81;
    enum{Ge1,Ge,As,Se1,Se,Br,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ge1, A, 32, 1, 7.61,sec);
    set_halflife(Ge, A, 32, 0, 7.6,sec);
    set_halflife(As, A, 33, 0, 33.3,sec);
    set_halflife(Se1, A, 34, 1, 57.28,min);

```

```

set_halfllife(Se, A, 34, 0, 18.45,min);
set_halfllife(Br, A, 35, 0, BIGNUM,sec);

lambda[As][Ge1]=-lambda[Ge1][Ge1];
lambda[As][Ge]=-lambda[Ge][Ge];
lambda[Se][As]=-lambda[As][As]*0.911;
lambda[Se1][As]=-lambda[As][As]*0.089;
lambda[Se][Se1]=-lambda[Se1][Se1]*0.9995;
lambda[Br][Se1]=-lambda[Se1][Se1]*0.0005;
lambda[Br][Se]=-lambda[Se][Se];

return 0;
}

int set_lambda_82(int A){
    const int A0=82;
    enum{Ge,As1,As,Se,Br1,Br,Kr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halfllife(Ge, A, 32, 0, 4.56,sec);
    set_halfllife(As1, A, 33, 1, 13.6,sec);
    set_halfllife(As, A, 33, 0, 19.1,sec);
    set_halfllife(Se, A, 34, 0, BIGNUM,sec);
    set_halfllife(Br1, A, 35, 1, 6.13,min);
    set_halfllife(Br, A, 35, 0, 35.282,hour);
    set_halfllife(Kr, A, 36, 0, BIGNUM,sec);

    lambda[As][Ge]=-lambda[Ge][Ge];
    lambda[As][As1]=-lambda[As1][As1];
    lambda[Se][As]=-lambda[As][As];
    lambda[Br][Br1]=-lambda[Br1][Br1]*0.976;
    lambda[Kr][Br1]=-lambda[Br1][Br1]*0.024;
    lambda[Kr][Br]=-lambda[Br][Br];

    return 0;
}

int set_lambda_83(int A){
    const int A0=83;
    enum{Ge,As,Se1,Se,Br,Kr1,Kr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);

```

```

for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_halfllife(Ge, A, 32, 0, 1.85,sec);
set_halfllife(As, A, 33, 0, 13.4,sec);
set_halfllife(Se1, A, 34, 1, 70.1,sec);
set_halfllife(Se, A, 34, 0, 22.3,min);
set_halfllife(Br, A, 35, 0, 2.40,hour);
set_halfllife(Kr1, A, 36, 1, 1.85,hour);
set_halfllife(Kr, A, 36, 0, BIGNUM,sec);

lambda[As][Ge]=-lambda[Ge][Ge];
lambda[Se][As]=-lambda[As][As]*0.366;
lambda[Se1][As]=-lambda[As][As]*0.634;
lambda[Br][Se1]=-lambda[Se1][Se1];
lambda[Br][Se]=-lambda[Se][Se];
lambda[Kr][Br]=-lambda[Br][Br]*0.0002;
lambda[Kr1][Br]=-lambda[Br][Br]*0.9998;
lambda[Kr][Kr1]=-lambda[Kr1][Kr1];

return 0;
}

int set_lambda_84(int A){
    const int A0=84;
    enum{As,Se,Br1,Br,Kr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halfllife(As, A, 33, 0, 4.2,sec);
    set_halfllife(Se, A, 34, 0, 3.26,min);
    set_halfllife(Br1, A, 35, 1, 6.0,min);
    set_halfllife(Br, A, 35, 0, 31.76,min);
    set_halfllife(Kr, A, 36, 0, BIGNUM,sec);

    lambda[Se][As]=-lambda[As][As];
    lambda[Br][Se]=-lambda[Se][Se];
    lambda[Kr][Br1]=-lambda[Br1][Br1];
    lambda[Kr][Br]=-lambda[Br][Br];

    return 0;
}

int set_lambda_85(int A){
    const int A0=85;
    enum{Br,Kr1,Kr,Rb,END};

```

```

    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Br, A, 35, 0, 2.90,min);
    set_halflife(Kr1, A, 36, 1, 4.480,hour);
    set_halflife(Kr, A, 36, 0, 10.752,year);
    set_halflife(Rb, A, 37, 0, BIGNUM,sec);

    lambda[Kr][Br]=-lambda[Br][Br]*0.9982;
    lambda[Kr1][Br]=-lambda[Br][Br]*0.0018;
    lambda[Kr][Kr1]=-lambda[Kr1][Kr1];
    lambda[Rb][Kr]=-lambda[Kr][Kr];

    return 0;
}

int set_lambda_86(int A){
    const int A0=86;
    enum{Br,Rb1,Rb,Kr,Sr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Br, A, 35, 0, 55.1,sec);
    set_halflife(Kr, A, 36, 0, BIGNUM,sec);
    set_halflife(Rb1, A, 37, 1, 1.017,min);
    set_halflife(Rb, A, 37, 0, 18.642,day);
    set_halflife(Sr, A, 38, 0, BIGNUM,sec);

    lambda[Kr][Br]=-lambda[Br][Br];
    lambda[Rb][Rb1]=-lambda[Rb1][Rb1];
    lambda[Sr][Rb]=-lambda[Rb][Rb];

    return 0;
}

int set_lambda_87(int A){
    const int A0=87;
    enum{Kr,Sr1,Rb,Sr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);

```

```

    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_half-life(Kr, A, 36, 0, 76.3,min);
    set_half-life(Rb, A, 37, 0, 4.81e+10,year);
    set_half-life(Sr1, A, 38, 1, 2.815,hour);
    set_half-life(Sr, A, 38, 0, BIGNUM,sec);

    lambda[Rb][Kr]=-lambda[Kr][Kr];
    lambda[Sr][Rb]=-lambda[Rb][Rb];
    lambda[Rb][Sr1]=-lambda[Sr1][Sr1]*0.003;
    lambda[Sr][Sr1]=-lambda[Sr1][Sr1]*0.997;

    return 0;
}

int set_lambda_88(int A){
    const int A0=88;
    enum{Kr,Rb,Sr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_half-life(Kr, A, 36, 0, 2.84,hour);
    set_half-life(Rb, A, 37, 0, 17.773,min);
    set_half-life(Sr, A, 38, 0, BIGNUM,sec);

    lambda[Rb][Kr]=-lambda[Kr][Kr];
    lambda[Sr][Rb]=-lambda[Rb][Rb];

    return 0;
}

int set_lambda_89(int A){
    const int A0=89;
    enum{Kr,Rb,Sr,Y1,Y,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_half-life(Kr, A, 36, 0, 3.15,min);
    set_half-life(Rb, A, 37, 0, 15.15,min);
    set_half-life(Sr, A, 38, 0, 50.53,day);
    set_half-life(Y1, A, 39, 1, 15.663,sec);
    set_half-life(Y, A, 39, 0, BIGNUM,sec);

```

```

    lambda[Rb][Kr] = -lambda[Kr][Kr];
    lambda[Sr][Rb] = -lambda[Rb][Rb];
    lambda[Y][Sr] = -lambda[Sr][Sr]*0.99985;
    lambda[Y1][Sr] = -lambda[Sr][Sr]*0.00015;
    lambda[Y][Y1] = -lambda[Y1][Y1];

    return 0;
}

int set_lambda_90(int A){
    const int A0=90;
    enum{Kr,Rb1,Rb,Sr,Y1,Y,Zr1,Zr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Kr, A, 36, 0, 32.32,sec);
    set_halflife(Rb1, A, 37, 1, 258,sec);
    set_halflife(Rb, A, 37, 0, 158,sec);
    set_halflife(Sr, A, 38, 0, 28.90,year);
    set_halflife(Y1, A, 39, 1, 3.19,hour);
    set_halflife(Y, A, 39, 0, 64.053,hour);
    set_halflife(Zr1, A, 40, 1, 809.2e-3,sec);
    set_halflife(Zr, A, 40, 0, BIGNUM,sec);

    lambda[Rb][Kr] = -lambda[Kr][Kr]*0.878;
    lambda[Rb1][Kr] = -lambda[Kr][Kr]*0.122;
    lambda[Rb][Rb1] = -lambda[Rb1][Rb1]*0.026;
    lambda[Sr][Rb1] = -lambda[Rb1][Rb1]*0.974;
    lambda[Sr][Rb] = -lambda[Rb][Rb];
    lambda[Y][Sr] = -lambda[Sr][Sr];
    lambda[Y][Y1] = -lambda[Y1][Y1];
    lambda[Zr][Y] = -lambda[Y][Y];
    lambda[Zr][Zr1] = -lambda[Zr1][Zr1];

    return 0;
}

int set_lambda_91(int A){
    const int A0=91;
    enum{Kr,Rb,Sr,Y1,Y,Zr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);

```

```

for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_halflife(Kr, A, 36, 0, 8.57,sec);
set_halflife(Rb, A, 37, 0, 58.4,sec);
set_halflife(Sr, A, 38, 0, 9.63,hour);
set_halflife(Y1, A, 39, 1, 49.71,min);
set_halflife(Y, A, 39, 0, 58.51,day);
set_halflife(Zr, A, 40, 0, BIGNUM,sec);

lambda[Rb][Kr]=-lambda[Kr][Kr];
lambda[Sr][Rb]=-lambda[Rb][Rb];
lambda[Y][Sr]=-lambda[Sr][Sr]*0.11;
lambda[Y1][Sr]=-lambda[Sr][Sr]*0.89;
lambda[Y][Y1]=-lambda[Y1][Y1];
lambda[Zr][Y]=-lambda[Y][Y];

return 0;
}

int set_lambda_92(int A){
    const int A0=92;
    enum{Kr,Rb,Sr,Y,Zr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Kr, A, 36, 0, 1.840,sec);
    set_halflife(Rb, A, 37, 0, 4.492,sec);
    set_halflife(Sr, A, 38, 0, 2.66,hour);
    set_halflife(Y, A, 39, 0, 3.54,hour);
    set_halflife(Zr, A, 40, 0, BIGNUM,sec);

    lambda[Rb][Kr]=-lambda[Kr][Kr];
    lambda[Sr][Rb]=-lambda[Rb][Rb];
    lambda[Y][Sr]=-lambda[Sr][Sr];
    lambda[Zr][Y]=-lambda[Y][Y];

    return 0;
}

int set_lambda_93(int A){
    const int A0=93;
    enum{Sr,Y1,Y,Zr,Nb1,Nb,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);

```

```

    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_half-life(Sr, A, 38, 0, 7.43,min);
    set_half-life(Y1, A, 39, 1, 0.82,sec);
    set_half-life(Y, A, 39, 0, 10.18,hour);
    set_half-life(Zr, A, 40, 0, 1.61e+6,year);
    set_half-life(Nb1, A, 41, 1, 16.12,hour);
    set_half-life(Nb, A, 41, 0, BIGNUM,sec);

    lambda[Y][Sr]=-lambda[Sr][Sr];
    lambda[Y][Y1]=-lambda[Y1][Y1];
    lambda[Zr][Y]=-lambda[Y][Y];
    lambda[Nb][Zr]=-lambda[Zr][Zr]*0.05;
    lambda[Nb1][Zr]=-lambda[Zr][Zr]*0.95;
    lambda[Nb][Nb1]=-lambda[Nb1][Nb1];

    return 0;
}

int set_lambda_94(int A){
    const int A0=94;
    enum{Sr,Y,Zr,Nb1,Nb,Mo,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_half-life(Sr, A, 38, 0, 75.3,sec);
    set_half-life(Y, A, 39, 0, 18.7,min);
    set_half-life(Zr, A, 40, 0, BIGNUM,sec);
    set_half-life(Nb1, A, 41, 1, 6.263,min);
    set_half-life(Nb, A, 41, 0, 2.03e+4,year);
    set_half-life(Mo, A, 42, 0, BIGNUM,sec);

    lambda[Y][Sr]=-lambda[Sr][Sr];
    lambda[Zr][Y]=-lambda[Y][Y];
    lambda[Nb][Nb1]=-lambda[Nb1][Nb1]*0.995;
    lambda[Mo][Nb1]=-lambda[Nb1][Nb1]*0.005;
    lambda[Mo][Nb]=-lambda[Nb][Nb];

    return 0;
}

int set_lambda_95(int A){
    const int A0=95;
    enum{Sr,Y,Zr,Nb1,Nb,Mo,END};
    int i,j;
    nn=END;

```



```

printf("set_lambda: A=%d nn=%d\n",A,nn);
if(A!=A0 || M<nn) exit(1);
for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_halflife(Sr, A, 38, 0, 23.90,sec);
set_halflife(Y, A, 39, 0, 10.3,min);
set_halflife(Zr, A, 40, 0, 64.032,day);
set_halflife(Nb1, A, 41, 1, 3.61,day);
set_halflife(Nb, A, 41, 0, 34.991,day);
set_halflife(Mo, A, 42, 0, BIGNUM,sec);

lambda[Y][Sr]=-lambda[Sr][Sr];
lambda[Zr][Y]=-lambda[Y][Y];
lambda[Nb][Zr]=-lambda[Zr][Zr]*0.991;
lambda[Nb1][Zr]=-lambda[Zr][Zr]*0.009;
lambda[Nb][Nb1]=-lambda[Nb1][Nb1]*0.944;
lambda[Mo][Nb1]=-lambda[Nb1][Nb1]*0.056;
lambda[Mo][Nb]=-lambda[Nb][Nb];

return 0;
}

int set_lambda_96(int A){
    const int A0=96;
    enum{Sr,Y1,Y,Zr,Nb,Mo,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Sr, A, 38, 0, 1.07,sec);
    set_halflife(Y1, A, 39, 1, 9.6,sec);
    set_halflife(Y, A, 39, 0, 5.34,sec);
    set_halflife(Zr, A, 40, 0, 2.35e+19,year);
    set_halflife(Nb, A, 41, 0, 23.35,hour);
    set_halflife(Mo, A, 42, 0, BIGNUM,sec);

    lambda[Y][Sr]=-lambda[Sr][Sr];
    lambda[Zr][Y1]=-lambda[Y1][Y1];
    lambda[Zr][Y]=-lambda[Y][Y];
    lambda[Mo][Nb]=-lambda[Nb][Nb];

    return 0;
}

int set_lambda_97(int A){
    const int A0=97;

```

```

enum{Sr,Y2,Y1,Y,Zr,Nb1,Nb,Mo,END};
int i,j;
nn=END;
printf("set_lambda: A=%d nn=%d\n",A,nn);
if(A!=A0 || M<nn) exit(1);
for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_halflife(Sr, A, 38, 0, 429e-3,sec);
set_halflife(Y2, A, 39, 2, 142e-3,sec);
set_halflife(Y1, A, 39, 1, 1.17,sec);
set_halflife(Y, A, 39, 0, 3.75,sec);
set_halflife(Zr, A, 40, 0, 16.749,year);
set_halflife(Nb1, A, 41, 1, 58.7,sec);
set_halflife(Nb, A, 41, 0, 72.1,min);
set_halflife(Mo, A, 42, 0, BIGNUM,sec);

lambda[Y][Sr]=-lambda[Sr][Sr];
lambda[Y][Y2]=-lambda[Y2][Y2]*0.984;
lambda[Zr][Y2]=-lambda[Y2][Y2]*0.016;
lambda[Zr][Y1]=-lambda[Y1][Y1];
lambda[Zr][Y]=-lambda[Y][Y];
lambda[Nb][Zr]=-lambda[Zr][Zr]*0.05;
lambda[Nb1][Zr]=-lambda[Zr][Zr]*0.95;
lambda[Nb][Nb1]=-lambda[Nb1][Nb1];
lambda[Mo][Nb]=-lambda[Nb][Nb];

return 0;
}

int set_lambda_98(int A){
const int A0=98;
enum{Zr,Nb1,Nb,Mo,Tc,Ru,END};
int i,j;
nn=END;
printf("set_lambda: A=%d nn=%d\n",A,nn);
if(A!=A0 || M<nn) exit(1);
for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_halflife(Zr, A, 40, 0, 30.7,sec);
set_halflife(Nb1, A, 41, 1, 51.3,min);
set_halflife(Nb, A, 41, 0, 2.86,sec);
set_halflife(Mo, A, 42, 0, BIGNUM,sec);
set_halflife(Tc, A, 43, 0, 4.2e+6,year);
set_halflife(Ru, A, 44, 0, BIGNUM,sec);

lambda[Nb][Zr]=-lambda[Zr][Zr];
lambda[Mo][Nb1]=-lambda[Nb1][Nb1];
lambda[Mo][Nb]=-lambda[Nb][Nb];

```

```

    lambda[Ru][Tc] = -lambda[Tc][Tc];

    return 0;
}

int set_lambda_99(int A){
    const int A0=99;
    enum{Zr,Nb1,Nb,Mo,Tc1,Tc,Ru,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_half-life(Zr, A, 40, 0, 2.1,sec);
    set_half-life(Nb1, A, 41, 1, 2.5,min);
    set_half-life(Nb, A, 41, 0, 15.0,sec);
    set_half-life(Mo, A, 42, 0, 69.976,hour);
    set_half-life(Tc1, A, 43, 1, 6.0067,year);
    set_half-life(Tc, A, 43, 0, 2.111e+5,year);
    set_half-life(Ru, A, 44, 0, BIGNUM,sec);

    lambda[Nb][Zr] = -lambda[Zr][Zr]*0.64;
    lambda[Nb1][Zr] = -lambda[Zr][Zr]*0.36;
    lambda[Nb][Nb1] = -lambda[Nb1][Nb1]*0.038;
    lambda[Mo][Nb1] = -lambda[Nb1][Nb1]*0.962;
    lambda[Mo][Nb] = -lambda[Nb][Nb];
    lambda[Tc][Mo] = -lambda[Mo][Mo]*0.13;
    lambda[Tc1][Mo] = -lambda[Mo][Mo]*0.87;
    lambda[Tc][Tc1] = -lambda[Tc1][Tc1];
    lambda[Ru][Tc] = -lambda[Tc][Tc];

    return 0;
}

int set_lambda_100(int A){
    const int A0=100;
    enum{Zr,Nb1,Nb,Tc,Mo,Ru,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_half-life(Zr, A, 40, 0, 7.1,sec);
    set_half-life(Nb1, A, 41, 1, 2.99,sec);
    set_half-life(Nb, A, 41, 0, 1.5,sec);
    set_half-life(Mo, A, 42, 0, 7.3e+18,hour);
    set_half-life(Tc, A, 43, 0, 15.46,sec);

```

```

set_halflife(Ru, A, 44, 0, BIGNUM,sec);

lambda[Nb][Zr]=-lambda[Zr][Zr]*0.98;
lambda[Nb1][Zr]=-lambda[Zr][Zr]*0.02;
lambda[Mo][Nb1]=-lambda[Nb1][Nb1];
lambda[Mo][Nb]=-lambda[Nb][Nb];
lambda[Ru][Mo]=-lambda[Mo][Mo];
lambda[Ru][Tc]=-lambda[Tc][Tc];

return 0;
}

int set_lambda_101(int A){
    const int A0=101;
    enum{Zr,Nb,Mo,Tc,Rh1,Rh,Ru,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Zr, A, 40, 0, 2.3,sec);
    set_halflife(Nb, A, 41, 0, 7.1,sec);
    set_halflife(Mo, A, 42, 0, 14.61,min);
    set_halflife(Tc, A, 43, 0, 14.02,min);
    set_halflife(Ru, A, 44, 0, BIGNUM,sec);
    set_halflife(Rh1, A, 45, 1, 4.34,day);
    set_halflife(Rh, A, 45, 0, 3.3,year);

    lambda[Nb][Zr]=-lambda[Zr][Zr];
    lambda[Mo][Nb]=-lambda[Nb][Nb];
    lambda[Tc][Mo]=-lambda[Mo][Mo];
    lambda[Ru][Tc]=-lambda[Tc][Tc];
    lambda[Ru][Rh1]=-lambda[Rh1][Rh1]*0.928;
    lambda[Rh][Rh1]=-lambda[Rh1][Rh1]*0.072;
    lambda[Ru][Rh]=-lambda[Rh][Rh];

    return 0;
}

int set_lambda_102(int A){
    const int A0=102;
    enum{Zr,Nb1,Nb,Mo,Tc1,Tc,Rh1,Rh,Pd,Ru,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

```

```

set_halflife(Zr, A, 40, 0, 2.9,sec);
set_halflife(Nb1, A, 41, 1, 1.3,sec);
set_halflife(Nb, A, 41, 0, 4.3,sec);
set_halflife(Mo, A, 42, 0, 11.3,min);
set_halflife(Tc1, A, 43, 1, 4.35,min);
set_halflife(Tc, A, 43, 0, 5.28,sec);
set_halflife(Ru, A, 44, 0, BIGNUM,sec);
set_halflife(Rh1, A, 45, 1, 3.742,year);
set_halflife(Rh, A, 45, 0, 207.3,day);
set_halflife(Pd, A, 46, 0, BIGNUM,sec);

lambda[Nb][Zr]=-lambda[Zr][Zr];
lambda[Mo][Nb1]=-lambda[Nb1][Nb1];
lambda[Mo][Nb]=-lambda[Nb][Nb];
lambda[Tc][Mo]=-lambda[Mo][Mo];
lambda[Tc][Tc1]=-lambda[Tc1][Tc1]*0.02;
lambda[Ru][Tc1]=-lambda[Tc1][Tc1]*0.98;
lambda[Ru][Tc]=-lambda[Tc][Tc];
lambda[Ru][Rh1]=-lambda[Rh1][Rh1]*0.9977;
lambda[Rh][Rh1]=-lambda[Rh1][Rh1]*0.0023;
lambda[Ru][Rh]=-lambda[Rh][Rh]*0.78;
lambda[Pd][Rh]=-lambda[Rh][Rh]*0.22;

return 0;
}

int set_lambda_103(int A){
    const int A0=103;
    enum{Zr,Nb,Mo,Tc,Ru,Rh1,Rh,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Zr, A, 40, 0, 1.32,sec);
    set_halflife(Nb, A, 41, 0, 1.5,sec);
    set_halflife(Mo, A, 42, 0, 67.5,sec);
    set_halflife(Tc, A, 43, 0, 54.2,sec);
    set_halflife(Ru, A, 44, 0, 39.247,day);
    set_halflife(Rh1, A, 45, 1, 56.114,min);
    set_halflife(Rh, A, 45, 0, BIGNUM,sec);

    lambda[Nb][Zr]=-lambda[Zr][Zr];
    lambda[Mo][Nb]=-lambda[Nb][Nb];
    lambda[Tc][Mo]=-lambda[Mo][Mo];
    lambda[Ru][Tc]=-lambda[Tc][Tc];

```

```

    lambda[Rh][Ru] = -lambda[Ru][Ru]*0.0025;
    lambda[Rh1][Ru] = -lambda[Ru][Ru]*0.9975;
    lambda[Rh][Rh1] = -lambda[Rh1][Rh1];

    return 0;
}

int set_lambda_104(int A){
    const int A0=104;
    enum{Zr,Nb1,Nb,Mo,Tc,Rh1,Rh,Ru,Pd,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Zr, A, 40, 0, 0.87,sec);
    set_halflife(Nb1, A, 41, 1, 0.94,sec);
    set_halflife(Nb, A, 41, 0, 4.9,sec);
    set_halflife(Mo, A, 42, 0, 60,sec);
    set_halflife(Tc, A, 43, 0, 18.3,min);
    set_halflife(Ru, A, 44, 0, BIGNUM,sec);
    set_halflife(Rh1, A, 45, 1, 4.34,min);
    set_halflife(Rh, A, 45, 0, 42.3,sec);
    set_halflife(Pd, A, 46, 0, BIGNUM,sec);

    lambda[Nb][Zr] = -lambda[Zr][Zr];
    lambda[Mo][Nb1] = -lambda[Nb1][Nb1];
    lambda[Mo][Nb] = -lambda[Nb][Nb];
    lambda[Tc][Mo] = -lambda[Mo][Mo];
    lambda[Ru][Tc] = -lambda[Tc][Tc];
    lambda[Rh][Rh1] = -lambda[Rh1][Rh1]*0.9987;
    lambda[Pd][Rh1] = -lambda[Rh1][Rh1]*0.0013;
    lambda[Ru][Rh] = -lambda[Rh][Rh]*0.0045;
    lambda[Pd][Rh] = -lambda[Rh][Rh]*0.9955;

    return 0;
}

int set_lambda_105(int A){
    const int A0=105;
    enum{Mo,Tc,Ru,Rh1,Rh,Pd,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Mo, A, 42, 0, 35.6,sec);

```

```

set_halflife(Tc, A, 43, 0, 7.6,min);
set_halflife(Ru, A, 44, 0, 4.44,hour);
set_halflife(Rh1, A, 45, 1, 42.9,sec);
set_halflife(Rh, A, 45, 0, 35.36,hour);
set_halflife(Pd, A, 46, 0, BIGNUM,sec);

lambda[Tc][Mo]=-lambda[Mo][Mo];
lambda[Ru][Tc]=-lambda[Tc][Tc];
lambda[Rh][Ru]=-lambda[Ru][Ru]*0.715;
lambda[Rh1][Ru]=-lambda[Ru][Ru]*0.285;
lambda[Rh][Rh1]=-lambda[Rh1][Rh1];
lambda[Pd][Rh]=-lambda[Rh][Rh];

return 0;
}

int set_lambda_106(int A){
    const int A0=106;
    enum{Mo,Tc,Ru,Rh1,Rh,Pd,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Mo, A, 42, 0, 8.73,sec);
    set_halflife(Tc, A, 43, 0, 35.6,sec);
    set_halflife(Ru, A, 44, 0, 371.8,day);
    set_halflife(Rh1, A, 45, 1, 131,min);
    set_halflife(Rh, A, 45, 0, 30.07,sec);
    set_halflife(Pd, A, 46, 0, BIGNUM,sec);

    lambda[Tc][Mo]=-lambda[Mo][Mo];
    lambda[Ru][Tc]=-lambda[Tc][Tc];
    lambda[Rh][Ru]=-lambda[Ru][Ru];
    lambda[Pd][Rh1]=-lambda[Rh1][Rh1];
    lambda[Pd][Rh]=-lambda[Rh][Rh];

    return 0;
}

int set_lambda_107(int A){
    const int A0=107;
    enum{Mo,Tc,Ru,Rh,Pd1,Pd,Ag1,Ag,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);

```

```

for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_halflife(Mo, A, 42, 0, 3.5,sec);
set_halflife(Tc, A, 43, 0, 21.2,sec);
set_halflife(Ru, A, 44, 0, 3.75,min);
set_halflife(Rh, A, 45, 0, 21.7,min);
set_halflife(Pd1, A, 46, 1, 21.3,sec);
set_halflife(Pd, A, 46, 0, 6.5e+6,year);
set_halflife(Ag1, A, 47, 1, 44.3,sec);
set_halflife(Ag, A, 47, 0, BIGNUM,sec);

lambda[Tc][Mo]==-lambda[Mo][Mo];
lambda[Ru][Tc]==-lambda[Tc][Tc];
lambda[Rh][Ru]==-lambda[Ru][Ru];
lambda[Pd][Rh]==-lambda[Rh][Rh];
lambda[Pd][Pd1]==-lambda[Pd1][Pd1];
lambda[Ag][Pd]==-lambda[Pd][Pd];
lambda[Ag][Ag1]==-lambda[Ag1][Ag1];

return 0;
}

int set_lambda_108(int A){
    const int A0=108;
    enum{Mo,Tc,Ru,Rh1,Rh,Ag1,Ag,Cd,Pd,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Mo, A, 42, 0, 1.09,sec);
    set_halflife(Tc, A, 43, 0, 5.17,sec);
    set_halflife(Ru, A, 44, 0, 4.55,min);
    set_halflife(Rh1, A, 45, 1, 6.0,min);
    set_halflife(Rh, A, 45, 0, 16.8,sec);
    set_halflife(Pd, A, 46, 0, BIGNUM,sec);
    set_halflife(Ag1, A, 47, 1, 438,year);
    set_halflife(Ag, A, 47, 0, 2.382,min);
    set_halflife(Cd, A, 48, 0, 1.9e+18,year);

    lambda[Tc][Mo]==-lambda[Mo][Mo];
    lambda[Ru][Tc]==-lambda[Tc][Tc];
    lambda[Rh][Ru]==-lambda[Ru][Ru];
    lambda[Pd][Rh1]==-lambda[Rh1][Rh1];
    lambda[Pd][Rh]==-lambda[Rh][Rh];
    lambda[Pd][Ag1]==-lambda[Ag1][Ag1]*0.913;

```



```

    lambda[Ag][Ag1] = -lambda[Ag1][Ag1]*0.087;
    lambda[Pd][Ag] = -lambda[Ag][Ag]*0.0285;
    lambda[Cd][Ag] = -lambda[Ag][Ag]*0.9715;

    return 0;
}

int set_lambda_109(int A){
    const int A0=109;
    enum{Tc,Ru,Rh,Pd1,Pd,Ag1,Ag,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_half-life(Tc, A, 43, 0, 0.86,sec);
    set_half-life(Ru, A, 44, 0, 34.5,sec);
    set_half-life(Rh, A, 45, 0, 80,sec);
    set_half-life(Pd1, A, 46, 1, 4.696,min);
    set_half-life(Pd, A, 46, 0, 13.7012,hour);
    set_half-life(Ag1, A, 47, 1, 39.6,sec);
    set_half-life(Ag, A, 47, 0, BIGNUM,sec);

    lambda[Ru][Tc] = -lambda[Tc][Tc];
    lambda[Rh][Ru] = -lambda[Ru][Ru];
    lambda[Pd][Rh] = -lambda[Rh][Rh];
    lambda[Pd][Pd1] = -lambda[Pd1][Pd1];
    lambda[Ag][Pd] = -lambda[Pd][Pd]*0.0005;
    lambda[Ag1][Pd] = -lambda[Pd][Pd]*0.9995;
    lambda[Ag][Ag1] = -lambda[Ag1][Ag1];

    return 0;
}

set_lambda_140

int set_lambda_126(int A){
    const int A0=126;
    enum{Ag,Cd,In1,In,Sn,Sb1,Sb,I,Te,Xe,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_half-life(Ag, A, 47, 0, 0.107,sec);
    set_half-life(Cd, A, 48, 0, 0.515,sec);
    set_half-life(In1,A, 49, 1, 1.64,sec);

```

```

set_halflife(In, A, 49, 0, 1.53,sec);
set_halflife(Sn, A, 50, 0, 2.30e+5,year);
set_halflife(Sb, A, 51, 0, 12.35,day);
set_halflife(Sb1,A, 51, 1, 19.15,min);
set_halflife(Te, A, 52, 0, BIGNUM,sec);
set_halflife( I, A, 53, 0, 12.93,day);
set_halflife(Xe, A, 54, 0, BIGNUM,sec);

lambda[Cd][Ag]=--lambda[Ag][Ag];
lambda[In][Cd]=--lambda[Cd][Cd];
lambda[Sn][In]=--lambda[In][In];
lambda[Sn][In1]=--lambda[In1][In1];
lambda[Sb][Sn]=--lambda[Sn][Sn];
lambda[Te][Sb]=--lambda[Sb][Sb];
lambda[Te][Sb1]=--lambda[Sb1][Sb1]*0.86;
lambda[Sb][Sb1]=--lambda[Sb1][Sb1]*0.14;
lambda[Te][I]=--lambda[I][I]*0.527;
lambda[Xe][I]=--lambda[I][I]*0.473;

return 0;
}

int set_lambda_127(int A){
    const int A0=127;
    enum{Ag,Cd,In2,In1,In,Sn1,Sn,Sb,Te1,Te,I,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ag, A, 47, 0, 0.109,sec);
    set_halflife(Cd, A, 48, 0, 0.37,sec);
    set_halflife(In, A, 49, 0, 1.09,sec);
    set_halflife(In1,A, 49, 1, 3.67,sec);
    set_halflife(In2,A, 49, 2, 1.04,sec);
    set_halflife(Sn, A, 50, 0, 2.10,hour);
    set_halflife(Sn1,A, 50, 1, 4.13,min);
    set_halflife(Sb, A, 51, 0, 3.85,day);
    set_halflife(Te, A, 52, 0, 9.35,hour);
    set_halflife(Te1,A, 52, 1, 106.1,day);
    set_halflife(I , A, 53, 0, BIGNUM,sec);

    lambda[Cd][Ag]=--lambda[Ag][Ag];
    lambda[In][Cd]=--lambda[Cd][Cd];
    lambda[Sn][In]=--lambda[In][In]*0.5;
    lambda[Sn1][In]=--lambda[In][In]*0.5;
    lambda[Sn][In1]=--lambda[In1][In1]*0.48;
    lambda[Sn1][In1]=--lambda[In1][In1]*0.52;
    lambda[Sn][In2]=--lambda[In2][In2];

```

```

    lambda[Sb][Sn] = -lambda[Sn][Sn];
    lambda[Sb][Sn1] = -lambda[Sn1][Sn1];
    lambda[Te][Sb] = -lambda[Sb][Sb]*0.98;
    lambda[Te1][Sb] = -lambda[Sb][Sb]*0.02;
    lambda[Te][Te1] = -lambda[Te1][Te1]*0.976;
    lambda[I][Te1] = -lambda[Te1][Te1]*0.024;
    lambda[I][Te] = -lambda[Te][Te];

    return 0;
}

int set_lambda_128(int A){
    const int A0=128;
    enum{Ag,Cd,In1,In,Sn1,Sn,Sb1,Sb,I,Te,Xe,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ag, A, 47, 0, 0.058,sec);
    set_halflife(Cd, A, 48, 0, 0.28,sec);
    set_halflife(In, A, 49, 0, 0.84,sec);
    set_halflife(In1,A, 49, 1, 0.72,sec);
    set_halflife(Sn, A, 50, 0, 59.07,min);
    set_halflife(Sn1,A, 50, 1, 6.5,sec);
    set_halflife(Sb, A, 51, 0, 9.01,hour);
    set_halflife(Sb1,A, 51, 1, 10.4,min);
    set_halflife(Te, A, 52, 0, BIGNUM,sec); //MUG
    set_halflife(I, A, 53, 0, 24.99,min);
    set_halflife(Xe, A, 54, 0, BIGNUM,sec);

    lambda[Cd][Ag] = -lambda[Ag][Ag];
    lambda[In][Cd] = -lambda[Cd][Cd];
    lambda[Sn][In] = -lambda[In][In];
    lambda[Sn][In1] = -lambda[In1][In1]*0.68;
    lambda[Sn1][In1] = -lambda[In1][In1]*0.32;
    lambda[Sn][Sn1] = -lambda[Sn1][Sn1];
    lambda[Sb][Sn] = -lambda[Sn][Sn];
    lambda[Sb][Sb1] = -lambda[Sb1][Sb1]*0.036;
    lambda[Te][Sb1] = -lambda[Sb1][Sb1]*0.964;
    lambda[Te][Sb] = -lambda[Sb][Sb];
    lambda[Te][I] = -lambda[I][I]*0.069;
    lambda[Xe][I] = -lambda[I][I]*0.931;

    return 0;
}

int set_lambda_129(int A){
    const int A0=129;

```

```

enum{In2,In,Sn1,Sn,Sb1,Sb,Te1,Te,I,Xe1,Xe,END};
int i,j;
nn=END;
printf("set_lambda: A=%d nn=%d\n",A,nn);
if(A!=A0 || M<nn) exit(1);
for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_halflife(In, A, 49, 0, 0.61,sec);
set_halflife(In2,A, 49, 2, 0.67,sec);
set_halflife(Sn, A, 50, 0, 2.23,min);
set_halflife(Sn1,A, 50, 1, 6.9,min);
set_halflife(Sb, A, 51, 0, 4.40,hour);
set_halflife(Sb1,A, 51, 1, 17.7,min);
set_halflife(Te, A, 52, 0, 69.6,min);
set_halflife(Te1,A, 52, 1, 33.6,day);
set_halflife( I, A, 53, 0, BIGNUM,sec);//MUG1.57e+7 だと warning ができる
set_halflife(Xe, A, 54, 0, BIGNUM,sec);
set_halflife(Xe1,A, 54, 1, 8.88,day);

lambda[Sn][In2]=-lambda[In2][In2];
lambda[Sn][In]=-lambda[In][In]*0.97;
lambda[Sn1][In]=-lambda[In][In]*0.03;
lambda[Sb][Sn1]=-lambda[Sn1][Sn1]*0.95;
lambda[Sb1][Sn1]=-lambda[Sn1][Sn1]*0.05;
lambda[Sb][Sn]=-lambda[Sn][Sn];
lambda[Sb][Sb1]=-lambda[Sb1][Sb1]*0.15;
lambda[Te][Sb1]=-lambda[Sb1][Sb1]*0.85;
lambda[Te][Sb]=-lambda[Sb][Sb]*0.894;
lambda[Te1][Sb]=-lambda[Sb][Sb]*0.106;
lambda[I][Te]=-lambda[Te][Te];
lambda[Te][Te1]=-lambda[Te1][Te1]*0.63;
lambda[I][Te1]=-lambda[Te1][Te1]*0.37;
/*lambda[Xe][I]=-lambda[I][I];*/
lambda[Xe][Xe1]=-lambda[Xe1][Xe1];

return 0;
}

int set_lambda_130(int A){
const int A0=130;
enum{Sn1,Sn,Sb1,Sb,I1,I,Te,Xe,END};
int i,j;
nn=END;
printf("set_lambda: A=%d nn=%d\n",A,nn);
if(A!=A0 || M<nn) exit(1);
for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_halflife(Sn, A, 50, 0, 3.72,min);
set_halflife(Sn1,A, 50, 1, 1.7,min);
set_halflife(Sb, A, 51, 0, 39.5,min);

```

```

    set_halflife(Sb1,A, 51, 1, 6.3,min);
    set_halflife(Te, A, 52, 0, BIGNUM,sec);
    set_halflife( I, A, 53, 0, 12.36,hour);
    set_halflife(I1, A, 53, 1, 8.84,min);
    set_halflife(Xe, A, 54, 0, BIGNUM,sec);

    lambda[Sb][Sn1]=-lambda[Sn1][Sn1]*0.5;
    lambda[Sb1][Sn1]=-lambda[Sn1][Sn1]*0.5;
    lambda[Sb][Sn]=-lambda[Sn][Sn];
    lambda[Te][Sb1]=-lambda[Sb1][Sb1];
    lambda[Te][Sb]=-lambda[Sb][Sb];
    lambda[Xe][I]=-lambda[I][I];
    lambda[I][I1]=-lambda[I1][I1]*0.84;
    lambda[Xe][I1]=-lambda[I1][I1]*0.16;

    return 0;
}

int set_lambda_131(int A){
    const int A0=131;
    enum{Sn,Sb,Te1,Te,I,Xe1,Xe,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}}

    set_halflife(Sn, A, 50, 0, 56.0,sec);
    set_halflife(Sb, A, 51, 0, 23.03,min);
    set_halflife(Te, A, 52, 0, 25.0,min);
    set_halflife(Te1,A, 52, 1, 33.25,hour);
    set_halflife( I, A, 53, 0, 8.0252,day);
    set_halflife(Xe, A, 54, 0, BIGNUM,sec);
    set_halflife(Xe1,A, 54, 1, 11.84,day);

    lambda[Sb][Sn]=-lambda[Sn][Sn];
    lambda[Te][Sb]=-lambda[Sb][Sb]*0.935;
    lambda[Te1][Sb]=-lambda[Sb][Sb]*0.065;
    lambda[I][Te]=-lambda[Te][Te];
    lambda[Te][Te1]=-lambda[Te1][Te1]*0.259;
    lambda[I][Te1]=-lambda[Te1][Te1]*0.741;
    lambda[Xe][I]=-lambda[I][I];
    lambda[Xe][Xe1]=-lambda[Xe1][Xe1];

    return 0;
}

int set_lambda_132(int A){
    const int A0=132;
    enum{Sn,Sb1,Sb,Te,I1,I,Cs,Ba,Xe1,Xe,END};

```

```

int i,j;
nn=END;
printf("set_lambda: A=%d nn=%d\n",A,nn);
if(A!=A0 || M<nn) exit(1);
for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_half-life(Sn, A, 50, 0, 39.7,sec);
set_half-life(Sb, A, 51, 0, 2.79,min);
set_half-life(Sb1,A, 51, 1, 4.10,min);
set_half-life(Te, A, 52, 0, 3.204,day);
set_half-life( I, A, 53, 0, 2.295,hour);
set_half-life(I1, A, 53, 1, 1.387,hour);
set_half-life(Xe, A, 54, 0, BIGNUM,sec);
set_half-life(Xe1,A, 54, 1, 0.00839,sec);
set_half-life(Cs, A, 55, 0, 6.480,day);
set_half-life(Ba, A, 56, 0, BIGNUM,sec);

lambda[Sb][Sn]==-lambda[Sn][Sn];
lambda[Te][Sb]==-lambda[Sb][Sb];
lambda[Te][Sb1]==-lambda[Sb1][Sb1];
lambda[I][Te]==-lambda[Te][Te];
lambda[Xe][I]==-lambda[I][I];
lambda[Xe][I1]==-lambda[I1][I1]*0.14;
lambda[I][I1]==-lambda[I1][I1]*0.86;
lambda[Xe][Cs]==-lambda[Cs][Cs]*0.9813;
lambda[Ba][Cs]==-lambda[Cs][Cs]*0.0187;

return 0;
}

int set_lambda_133(int A){
const int A0=133;
enum{Sn,Sb,Te1,Te,I,Xe1,Xe,Cs,END};
int i,j;
nn=END;
printf("set_lambda: A=%d nn=%d\n",A,nn);
if(A!=A0 || M<nn) exit(1);
for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_half-life(Sn, A, 50, 0, 1.46,sec);
set_half-life(Sb, A, 51, 0, 2.34,min);
set_half-life(Te, A, 52, 0, 12.5,min);
set_half-life(Te1,A, 52, 1, 55.4,min);
set_half-life( I, A, 53, 0, 20.83,hour);
set_half-life(Xe, A, 54, 0, 5.2475,day);
set_half-life(Xe1,A, 54, 1, 2.198,day);
set_half-life(Cs, A, 55, 0, BIGNUM,sec);

lambda[Sb][Sn]==-lambda[Sn][Sn];
lambda[Te][Sb]==-lambda[Sb][Sb];

```

```

    lambda[Te][Te1] = -lambda[Te1][Te1]*0.165;
    lambda[I][Te1] = -lambda[Te1][Te1]*0.835;
    lambda[I][Te] = -lambda[Te][Te];
    lambda[Xe][I] = -lambda[I][I]*0.9892;
    lambda[Xe1][I] = -lambda[I][I]*0.0108;
    lambda[Xe][Xe1] = -lambda[Xe1][Xe1];
    lambda[Cs][Xe] = -lambda[Xe][Xe];

    return 0;
}

int set_lambda_134(int A){
    const int A0=134;
    enum{Sb,Te,I1,I,Xe1,Xe,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Sb, A, 51, 0, 0.78,sec);
    set_halflife(Te, A, 52, 0, 41.8,min);
    set_halflife( I, A, 53, 0, 52.5,min);
    set_halflife(I1, A, 53, 1, 3.52,min);
    set_halflife(Xe, A, 54, 0, BIGNUM,sec);
    set_halflife(Xe1,A, 54, 1, 0.290,sec);

    lambda[Te][Sb] = -lambda[Sb][Sb];
    lambda[I][Te] = -lambda[Te][Te];
    lambda[Xe][I] = -lambda[I][I];
    lambda[I][I1] = -lambda[I1][I1]*0.977;
    lambda[Xe1][I1] = -lambda[I1][I1]*0.023;
    lambda[Xe][Xe1] = -lambda[Xe1][Xe1];

    return 0;
}

int set_lambda_135(int A){
    const int A0=135;
    enum{Te,I,Xe1,Xe,Cs,Ba1,Ba,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Te, A, 52, 0, 19.0,sec);
    set_halflife( I, A, 53, 0, 6.58,hour);
    set_halflife(Xe, A, 54, 0, 9.14,hour);
    set_halflife(Xe1,A, 54, 1, 15.29,min);

```

```

set_halflife(Cs, A, 55, 0, BIGNUM,sec); //2.3e+6 year だと error
set_halflife(Ba, A, 56, 0, BIGNUM,sec);
set_halflife(Ba1,A, 56, 1, 28.7,hour);

lambda[I][Te]=-lambda[Te][Te];
lambda[Xe][I]=-lambda[I][I]*0.981;
lambda[Xe1][I]=-lambda[I][I]*0.019;
lambda[Cx][Xe]=-lambda[Xe][Xe];
lambda[Xe][Xe1]=-lambda[Xe1][Xe1]*0.994;
lambda[Cx][Xe1]=-lambda[Xe1][Xe1]*0.006;
/*lambda[Ba][Cs]=-lambda[Cx][Cx];*/
lambda[Ba][Ba1]=-lambda[Ba1][Ba1];

return 0;
}

int set_lambda_136(int A){
    const int A0=136;
    enum{I1,I,Cs,Xe,Ba,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife( I, A, 53, 0, 83.4,sec);
    set_halflife(I1, A, 53, 1, 46.9,sec);
    set_halflife(Xe, A, 54, 0, BIGNUM,sec);
    set_halflife(Cs, A, 55, 0, 13.04,day);
    set_halflife(Ba, A, 56, 0, BIGNUM,sec);

    lambda[Xe][I]=-lambda[I][I];
    lambda[Xe][I1]=-lambda[I1][I1];
    lambda[Ba][Cs]=-lambda[Cx][Cx];

    return 0;
}

int set_lambda_137(int A){
    const int A0=137;
    enum{Xe,Cs,Ba1,Ba,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Xe, A, 54, 0, 3.818,min);
    set_halflife(Cs, A, 55, 0, 30.08,year);

```



```

    set_halflife(Ba, A, 56, 0, BIGNUM,sec);
    set_halflife(Ba1,A, 56, 1, 2.552,min);

    lambda[Cx][Xe]=-lambda[Xe][Xe];
    lambda[Ba][Cs]=-lambda[Cx][Cx]*0.053;
    lambda[Ba1][Cx]=-lambda[Cx][Cx]*0.947;
    lambda[Ba][Ba1]=-lambda[Ba1][Ba1];

    return 0;
}

int set_lambda_138(int A){
    const int A0=138;
    enum{Xe,Cs1,Cs,La,Ce,Ba,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Xe, A, 54, 0, 14.08,min);
    set_halflife(Cs, A, 55, 0, 33.41,min);
    set_halflife(Cs1,A, 55, 1, 2.91,min);
    set_halflife(Ba, A, 56, 0, BIGNUM,sec);
    set_halflife(La, A, 57, 0, 1.02e+11,year);
    set_halflife(Ce, A, 58, 0, BIGNUM,sec);

    lambda[Cx][Xe]=-lambda[Xe][Xe];
    lambda[Cx][Cs1]=-lambda[Cx1][Cx1]*0.81;
    lambda[Ba][Cs1]=-lambda[Cx1][Cx1]*0.19;
    lambda[Ba][Cx]=-lambda[Cx][Cx];
    lambda[Ba][La]=-lambda[La][La]*0.656;
    lambda[Ce][La]=-lambda[La][La]*0.344;

    return 0;
}

int set_lambda_139(int A){
    const int A0=139;
    enum{Xe,Cs,Ba,Ce1,Ce,La,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Xe, A, 54, 0, 39.68,sec);
    set_halflife(Cs, A, 55, 0, 9.27,min);
    set_halflife(Ba, A, 56, 0, 83.06,min);
    set_halflife(La, A, 57, 0, BIGNUM,sec);

```

```

set_halflife(Ce1,A, 58, 1, 54.8,sec);
set_halflife(Ce, A, 58, 0, 137.641,day);

lambda[Cs][Xe]=-lambda[Xe][Xe];
lambda[Ba][Cs]=-lambda[Cs][Cs];
lambda[La][Ba]=-lambda[Ba][Ba];
lambda[Ce][Ce1]=-lambda[Ce1][Ce1];
lambda[La][Ce]=-lambda[Ce][Ce];

return 0;
}

int set_lambda_140(int A){
    const int A0=140;
    enum{Xe,Cs,Ba,La,Ce,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Xe, A, 54, 0, 13.60,sec);
    set_halflife(Cs, A, 55, 0, 63.7,sec);
    set_halflife(Ba, A, 56, 0, 12.7527,day);
    set_halflife(La, A, 57, 0, 1.67855,day);
    set_halflife(Ce, A, 58, 0, BIGNUM,sec);

    lambda[Cs][Xe]=-lambda[Xe][Xe];
    lambda[Ba][Cs]=-lambda[Cs][Cs];
    lambda[La][Ba]=-lambda[Ba][Ba];
    lambda[Ce][La]=-lambda[La][La];

    return 0;
}

int set_lambda_141(int A){
    const int A0=141;
    enum{Xe,Cs,Ba,La,Ce,Pr,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Xe, A, 54, 0, 1.73,sec);
    set_halflife(Cs, A, 55, 0, 24.84,sec);
    set_halflife(Ba, A, 56, 0, 18.27,min);
    set_halflife(La, A, 57, 0, 3.92,hour);
    set_halflife(Ce, A, 58, 0, 32.508,day);
    set_halflife(Pr, A, 59, 0, BIGNUM,sec);

```

```

    lambda[Ce][La]=-lambda[La][La];
    lambda[Pr][Ce]=-lambda[Ce][Ce];

    return 0;
}

int set_lambda_142(int A){
    const int A0=142;
    enum{I,Xe,Cs,Ba,La,Pr,Ce,Nd,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife( I, A, 53, 0, 0.00222,sec);
    set_halflife(Xe, A, 54, 0, 1.23,sec);
    set_halflife(Cs, A, 55, 0, 1.684,sec);
    set_halflife(Ba, A, 56, 0, 10.6,min);
    set_halflife(La, A, 57, 0, 91.1,min);
    set_halflife(Ce, A, 58, 0, BIGNUM,sec);
    set_halflife(Pr, A, 59, 0, 19.12,hour);
    set_halflife(Nd, A, 60, 0, BIGNUM,sec);

    lambda[Xe][I]=-lambda[I][I];
    lambda[Ce][Xe]=-lambda[Xe][Xe];
    lambda[Ba][Cs]=-lambda[Cs][Cs];
    lambda[La][Ba]=-lambda[Ba][Ba];
    lambda[Ce][La]=-lambda[La][La];
    lambda[Ce][Pr]=-lambda[Pr][Pr]*0.0002;
    lambda[Nd][Pr]=-lambda[Pr][Pr]*0.9998;

    return 0;
}

int set_lambda_143(int A){
    const int A0=143;
    enum{Ba,La,Ce,Pr,Nd,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ba, A, 56, 0, 14.5,sec);
    set_halflife(La, A, 57, 0, 14.2,min);

```

```

    set_halflife(Ce, A, 58, 0, 33.039, hour);
    set_halflife(Pr, A, 59, 0, 13.57, day);
    set_halflife(Nd, A, 60, 0, BIGNUM, sec);

    lambda[La][Ba] = -lambda[Ba][Ba];
    lambda[Ce][La] = -lambda[La][La];
    lambda[Pr][Ce] = -lambda[Ce][Ce];
    lambda[Nd][Pr] = -lambda[Pr][Pr];

    return 0;
}

int set_lambda_144(int A){
    const int A0=144;
    enum{La,Ce,Pr1,Pr,Nd,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(La, A, 57, 0, 40.8, sec);
    set_halflife(Ce, A, 58, 0, 284.91, day);
    set_halflife(Pr, A, 59, 0, 17.28, min);
    set_halflife(Pr1,A, 59, 1, 7.2, min);
    set_halflife(Nd, A, 60, 0, BIGNUM, sec);

    lambda[Ce][La] = -lambda[La][La];
    lambda[Pr][Ce] = -lambda[Ce][Ce];
    lambda[Pr][Pr1] = -lambda[Pr1][Pr1]*0.9993;
    lambda[Nd][Pr1] = -lambda[Pr1][Pr1]*0.0007;
    lambda[Nd][Pr] = -lambda[Pr][Pr];

    return 0;
}

int set_lambda_145(int A){
    const int A0=145;
    enum{Ba,La,Ce,Pr,Nd,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ba, A, 56, 0, 4.31, sec);
    set_halflife(La, A, 57, 0, 24.8, sec);
    set_halflife(Ce, A, 58, 0, 3.01, min);
    set_halflife(Pr, A, 59, 0, 5.984, hour);
    set_halflife(Nd, A, 60, 0, BIGNUM, sec);

```

```

    lambda[La][Ba] = -lambda[Ba][Ba];
    lambda[Ce][La] = -lambda[La][La];
    lambda[Pr][Ce] = -lambda[Ce][Ce];
    lambda[Nd][Pr] = -lambda[Pr][Pr];

    return 0;
}

int set_lambda_146(int A){
    const int A0=146;
    enum{Ba,La1,La,Ce,Pr,Pm,Nd,Sm,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ba, A, 56, 0, 2.22,sec);
    set_halflife(La, A, 57, 0, 6.27,sec);
    set_halflife(La1,A, 57, 1, 10.0,sec);
    set_halflife(Ce, A, 58, 0, 13.52,min);
    set_halflife(Pr, A, 59, 0, 24.15,min);
    set_halflife(Nd, A, 60, 0, BIGNUM,sec);
    set_halflife(Pm, A, 61, 0, 5.53,year);
    set_halflife(Sm, A, 62, 0, BIGNUM,sec);

    lambda[La][Ba] = -lambda[Ba][Ba];
    lambda[Ce][La] = -lambda[La][La];
    lambda[Ce][La1] = -lambda[La1][La1];
    lambda[Pr][Ce] = -lambda[Ce][Ce];
    lambda[Nd][Pr] = -lambda[Pr][Pr];
    lambda[Nd][Pm] = -lambda[Pm][Pm]*0.66;
    lambda[Sm][Pm] = -lambda[Pm][Pm]*0.34;

    return 0;
}

int set_lambda_147(int A){
    const int A0=147;
    enum{Ba,La,Ce,Pr,Nd,Pm,Sm,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ba, A, 56, 0, 0.894,sec);
    set_halflife(La, A, 57, 0, 4.06,sec);
    set_halflife(Ce, A, 58, 0, 56.4,sec);

```

```

set_halfllife(Pr, A, 59, 0, 13.4,min);
set_halfllife(Nd, A, 60, 0, 10.98,day);
set_halfllife(Pm, A, 61, 0, 2.6234,year);
set_halfllife(Sm, A, 62, 0, BIGNUM,sec);

lambda[La][Ba]=-lambda[Ba][Ba];
lambda[Ce][La]=-lambda[La][La];
lambda[Pr][Ce]=-lambda[Ce][Ce];
lambda[Nd][Pr]=-lambda[Pr][Pr];
lambda[Pm][Nd]=-lambda[Nd][Nd];
lambda[Sm][Pm]=-lambda[Pm][Pm];

return 0;
}

int set_lambda_148(int A){
    const int A0=148;
    enum{Ba,La,Ce,Pr1,Pr,Nd,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halfllife(Ba, A, 56, 0, 0.612,sec);
    set_halfllife(La, A, 57, 0, 1.26,sec);
    set_halfllife(Ce, A, 58, 0, 56.0,sec);
    set_halfllife(Pr, A, 59, 0, 2.29,min);
    set_halfllife(Pr1,A, 59, 1, 2.01,min);
    set_halfllife(Nd, A, 60, 0, BIGNUM,sec);

    lambda[La][Ba]=-lambda[Ba][Ba];
    lambda[Ce][La]=-lambda[La][La];
    lambda[Pr][Ce]=-lambda[Ce][Ce];
    lambda[Nd][Pr]=-lambda[Pr][Pr];
    lambda[Nd][Pr1]=-lambda[Pr1][Pr1];

    return 0;
}

int set_lambda_149(int A){
    const int A0=149;
    enum{Ce,Pr,Nd,Pm,Sm,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halfllife(Ce, A, 58, 0, 5.3,sec);

```

```

    set_halflife(Pr, A, 59, 0, 2.26,min);
    set_halflife(Nd, A, 60, 0, 1.728,hour);
    set_halflife(Pm, A, 61, 0, 53.08,hour);
    set_halflife(Sm, A, 62, 0, BIGNUM,sec);

    lambda[Pr][Ce]=-lambda[Ce][Ce];
    lambda[Nd][Pr]=-lambda[Pr][Pr];
    lambda[Pm][Nd]=-lambda[Nd][Nd];
    lambda[Sm][Pm]=-lambda[Pm][Pm];

    return 0;
}

int set_lambda_150(int A){
    const int A0=150;
    enum{Ce,Pr,Nd,Pm,Sm,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ce, A, 58, 0, 4.0,sec);
    set_halflife(Pr, A, 59, 0, 6.19,sec);
    set_halflife(Nd, A, 60, 0, BIGNUM,sec);
    set_halflife(Pm, A, 61, 0, 2.68,hour);
    set_halflife(Sm, A, 62, 0, BIGNUM,sec);

    lambda[Pr][Ce]=-lambda[Ce][Ce];
    lambda[Nd][Pr]=-lambda[Pr][Pr];
    lambda[Pm][Nd]=-lambda[Nd][Nd];
    lambda[Sm][Pm]=-lambda[Pm][Pm];

    return 0;
}

int set_lambda_151(int A){
    const int A0=151;
    enum{Ce1,Ce,Pr,Nd,Pm,Sm,Eu,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(Ce, A, 58, 0, 1.76,sec);
    set_halflife(Ce1,A, 58, 1, 1.20,sec);
    set_halflife(Pr, A, 59, 0, 18.90,sec);
    set_halflife(Nd, A, 60, 0, 12.44,min);
    set_halflife(Pm, A, 61, 0, 28.40,hour);

```

```

set_halflife(Sm, A, 62, 0, 90.0,year);
set_halflife(Eu, A, 63, 0, BIGNUM,sec);

lambda[Pr][Ce]=-lambda[Ce][Ce];
lambda[Pr][Ce1]=-lambda[Ce1][Ce1];
lambda[Nd][Pr]=-lambda[Pr][Pr];
lambda[Pm][Nd]=-lambda[Nd][Nd];
lambda[Sm][Pm]=-lambda[Pm][Pm];
lambda[Eu][Sm]=-lambda[Sm][Sm];

return 0;
}

int set_lambda_152(int A){
    const int A0=152;
    enum{La,Ce,Pr,Nd,Pm1,Pm,Eu2,Eu1,Eu,Gd,Sm,END};
    int i,j;
    nn=END;
    printf("set_lambda: A=%d nn=%d\n",A,nn);
    if(A!=A0 || M<nn) exit(1);
    for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

    set_halflife(La, A, 57, 0, 0.000000150,sec);
    set_halflife(Ce, A, 58, 0, 1.4,sec);
    set_halflife(Pr, A, 59, 0, 3.57,sec);
    set_halflife(Nd, A, 60, 0, 11.4,min);
    set_halflife(Pm, A, 61, 0, 4.12,min);
    set_halflife(Pm1,A, 61, 1, 13.8,min);
    set_halflife(Sm, A, 62, 0, BIGNUM,sec);
    set_halflife(Eu, A, 63, 0, 13.528,year);
    set_halflife(Eu1,A, 63, 1, 9.3116,hour);
    set_halflife(Eu2,A, 63, 2, 96.0,min);
    set_halflife(Gd, A, 64, 0, BIGNUM,sec);

    lambda[Ce][La]=-lambda[La][La];
    lambda[Pr][Ce]=-lambda[Ce][Ce];
    lambda[Nd][Pr]=-lambda[Pr][Pr];
    lambda[Pm][Nd]=-lambda[Nd][Nd];
    lambda[Sm][Pm]=-lambda[Pm][Pm];
    lambda[Sm][Pm1]=-lambda[Pm1][Pm1];
    lambda[Eu][Eu2]=-lambda[Eu2][Eu2];
    lambda[Sm][Eu1]=-lambda[Eu1][Eu1]*0.28;
    lambda[Gd][Eu1]=-lambda[Eu1][Eu1]*0.72;
    lambda[Sm][Eu]=-lambda[Eu][Eu]*0.721;
    lambda[Gd][Eu]=-lambda[Eu][Eu]*0.279;

    return 0;
}

int set_lambda_153(int A){

```



```

const int A0=153;
enum{La,Ce,Pr,Nd,Pm,Sm1,Sm,Eu,END};
int i,j;
nn=END;
printf("set_lambda: A=%d nn=%d\n",A,nn);
if(A!=A0 || M<nn) exit(1);
for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_half-life(La, A, 57, 0, 0.00000011,sec);//MUG ずらした
set_half-life(Ce, A, 58, 0, 0.0000001,sec);
set_half-life(Pr, A, 59, 0, 4.28,sec);
set_half-life(Nd, A, 60, 0, 31.6,sec);
set_half-life(Pm, A, 61, 0, 5.25,min);
set_half-life(Sm, A, 62, 0, 46.284,hour);
set_half-life(Sm1,A, 62, 1, 0.0106,sec);
set_half-life(Eu, A, 63, 0, BIGNUM,sec);

lambda[Ce][La]=-lambda[La][La];
lambda[Pr][Ce]=-lambda[Ce][Ce];
lambda[Nd][Pr]=-lambda[Pr][Pr];
lambda[Pm][Nd]=-lambda[Nd][Nd];
lambda[Sm][Pm]=-lambda[Pm][Pm];
lambda[Sm][Sm1]=-lambda[Sm1][Sm1];
lambda[Eu][Sm]=-lambda[Sm][Sm];

return 0;
}

int set_lambda_154(int A){
const int A0=154;
enum{Pr,Nd,Pm1,Pm,Eu1,Eu,Sm,Gd,END};
int i,j;
nn=END;
printf("set_lambda: A=%d nn=%d\n",A,nn);
if(A!=A0 || M<nn) exit(1);
for(i=0;i<nn;i++){for(j=0;j<nn;j++)lambda[i][j]=0;}

set_half-life(Pr, A, 59, 0, 2.3,sec);
set_half-life(Nd, A, 60, 0, 25.9,sec);
set_half-life(Pm, A, 61, 0, 2.68,min);
set_half-life(Pm1,A, 61, 1, 1.73,min);
set_half-life(Sm, A, 62, 0, BIGNUM,sec);
set_half-life(Eu, A, 63, 0, 8.601,year);
set_half-life(Eu1,A, 63, 1, 46.3,min);
set_half-life(Gd, A, 64, 0, BIGNUM,sec);

lambda[Nd][Pr]=-lambda[Pr][Pr];
lambda[Pm][Nd]=-lambda[Nd][Nd];
lambda[Sm][Pm]=-lambda[Pm][Pm];
lambda[Sm][Pm1]=-lambda[Pm1][Pm1];

```

```
lambda[Sm][Eu]=-lambda[Eu][Eu]*0.0002;  
lambda[Gd][Eu]=-lambda[Eu][Eu]*0.9998;  
lambda[Eu][Eu1]=-lambda[Eu1][Eu1];  
  
return 0;  
}
```