In[1]:=

```
(*Set initial directory*)
directory = NotebookDirectory[];



(*Display menu for nuclear species selection*)


Print["Select Nuclear Species:"];
Print["1. U-235_Data_JENDL5_CalculationData"];
Print["2. U-233_Data_JENDL5_CalculationData"];
Print["3. U-238_Data_JENDL5_CalculationData"];
Print["4. Th-232_Data_JENDL5_CalculationData"];
Print["5. Np-237_Data_JENDL5_CalculationData"];
Print["6. Pu-239_Data_JENDL5_CalculationData"];
Print["7. Pu-240_Data_JENDL5_CalculationData"];
Print["8. Pu-242_Data_JENDL5_CalculationData"];
Print["9. Am-241_Data_JENDL5_CalculationData"];

(*Get user selection*)
choice = Input["Enter number (1-9): "];

(*Load file based on selection*)
fileName = Switch[choice, 1, "U-235_Data_JENDL5_CalculationData.m", 2,
   "U-233_Data_JENDL5_CalculationData.m", 3, "U-238_Data_JENDL5_CalculationData.m", 4,
   "Th-232_Data_JENDL5_CalculationData.m", 5, "Np-237_Data_JENDL5_CalculationData.m",
   6, "Pu-239_Data_JENDL5_CalculationData.m", 7,
   "Pu-240_Data_JENDL5_CalculationData.m", 8, "Pu-242_Data_JENDL5_CalculationData.m",
   9, "Am-241_Data_JENDL5_CalculationData.m", _, (Print["Invalid selection"];
    Abort[])];



(*Load file using full path*)
fullPath = FileNameJoin[{directory, fileName}];
Print["Attempting to load file from: ", fullPath];

If[FileExistsQ[fullPath], Get[fullPath];
  Print["Successfully loaded: ", fileName],
  Print["Error: File not found at ", fullPath]];

(*2*)


(*Initial Calculations for Nuclear Parameters*)
fragmentZ1 = atomicNumber / 2; (*First fragment atomic number*)
fragmentZ2 = atomicNumber / 2; (*Second fragment atomic number*)
reducedMass = N[Sqrt[fragmentZ1 * fragmentZ2 / (fragmentZ1 + fragmentZ2)]];


fitStartZ = 28; (*Starting atomic number for fitting*)
fitEndZ = atomicNumber - fitStartZ; (*Ending atomic number for fitting*)
```

```
fitStartIndex = fitStartZ - 22;(*Starting index for fitting data*)
fitEndIndex = fitEndZ - 22;(*Ending index for fitting data*)
(*Initialize distance parameter*)



(*Incident Neutron Kinetic Energy*)
neutronEnergy1 = 0;(*0.0253 eV*)
neutronEnergy2 = 0.5;(*500 keV*)
neutronEnergy3 = 14;(*14 MeV*)



effectiveDistance = .;

(*Initialize energy-dependent distance parameters*)
effectiveDistance0253eV = .;
effectiveDistance500keV = .;
effectiveDistance14MeV = .;

(*Initialize variable lists for each energy region*)
distanceVars0253eV = {};
distanceVars500keV = {};
distanceVars14MeV = {};

(*Initialize fermi correction lists*)
fermiVars0253eV = {};
fermiVars500keV = {};
fermiVars14MeV = {};

(*Initialize fermi correction energies*)
fermiEnergy1 = .;
fermiEnergy2 = .;
fermiEnergy3 = .;

(*Initialize various parameter lists*)
paramList0253eV = {};
paramList500keV = {};
paramList14MeV = {};
paramList5 = {};
paramList6 = {};
paramList7 = {};

fermiEnergies = {};
variablesList = {};
fissionYields = {};
neutronVars = {};
effectiveDistances = {};
distanceParams = {};

paramList50 = {};
paramList60 = {};
paramList70 = {};
```

```
Clear[theoreticalYield0253eV, theoreticalYield500keV, theoreticalYield14MeV,
 fittedYield0253eV, fittedYield500keV, fittedYield14MeV, optResult0253eV,
 optResult500keV, optResult14MeV, fermiVars0253eV, fermiVars500keV, fermiVars14MeV]

(*Mass retrieval function definition*)
getNuclearMass[Z_, A_] :=
 Module[{elementName, isotopeName, mass, numericMass}, (*Validate atomic number*)
  If[! NumberQ[Z] || Z < 1 || Z > 118, Return[Missing["NotAvailable"]]];
  (*Get element name*) elementName = ElementData[Z, "Name"];
  If[elementName === $Failed, Return[Missing["NotAvailable"]]];
  (*Format element name*) elementName =
   StringReplacePart[elementName, ToUpperCase[StringTake[elementName, 1]], {1, 1}];
  (*Create isotope name*) isotopeName = elementName <> ToString[A];
  (*Get mass data*) mass = IsotopeData[isotopeName, "AtomicMass"];
  (*Return numeric mass or missing value*) If[mass === Missing["NotAvailable"],
   Missing["NotAvailable"], numericMass = QuantityMagnitude[mass];
   numericMass]]


(*Energy pattern selection*)
{startEnergyIndex, endEnergyIndex} =
  Switch[energyPattern, 1, {1, 1}, (*0.0253 eV only*) 2, {1, 2},
   (*0.0253 eV and 500 keV*) 3, {1, 3}, (*All three energies*) 4,
   {2, 2}, (*500 keV only*) 5, {2, 3}  (*500 keV and 14 MeV*)];

(*Isotope existence check function*)
isIsotopeStable[z_, n_] :=
 Module[{element, isotope, mass}, element = ElementData[z, "Name"];
  element = StringReplacePart[element, ToUpperCase[StringTake[element, 1]], {1, 1}];
  isotope = element <> ToString[z + n];
  mass = IsotopeData[isotope, "AtomicMass"];
  mass =!= Missing["NotAvailable"] && NumberQ[QuantityMagnitude[mass]]]


(*3*) (*Main Nuclear Fission Calculation Function*)
CalculateFissionYields[energyPattern_] := Module[{dataTemp, results = {}},
  For[energyIndex = startEnergyIndex, energyIndex ≤ endEnergyIndex, energyIndex++,
   (*Set parameters for each energy region*) {promptNeutronCount, incidentEnergy,
     effectiveDistance, variableList, yieldList, neutronVariables} =
    Switch[energyIndex, 1, {promptNeutrons1, neutronEnergy1, effectiveDistance0253eV,
      distanceVars0253eV, fissionYield0253eV, fermiVars0253eV}, (*Thermal*)
     2, {promptNeutrons2, neutronEnergy2, effectiveDistance500keV,
      distanceVars500keV, fissionYield500keV, fermiVars500keV}, (*Intermediate*)
     3, {promptNeutrons3, neutronEnergy3, effectiveDistance14MeV,
      distanceVars14MeV, fissionYield14MeV, fermiVars14MeV}   (*Fast*)];
   dataTemp = Reap[For[protonNumber1 = 23, protonNumber1 ≤ atomicNumber - 23,
       protonNumber1++, For[neutronCount1 = 0, neutronCount1 ≤ neutronNumber,
        neutronCount1++, protonNumber2 = atomicNumber - protonNumber1;
        neutronCount2 = neutronNumber - neutronCount1 - Round[promptNeutronCount];
        massNumber1 = protonNumber1 + neutronCount1;
        massNumber2 = protonNumber2 + neutronCount2;



        If[ (protonNumber1 == 22 && 16 <= neutronCount1 <= 41) ||
```

```
(protonNumber1 == 23 && 17 <= neutronCount1 <= 42) ||
(protonNumber1 == 24 && 18 <= neutronCount1 <= 43) ||
(protonNumber1 == 25 && 19 <= neutronCount1 <= 44) ||
(protonNumber1 == 26 && 19 <= neutronCount1 <= 46) ||
(protonNumber1 == 27 && 20 <= neutronCount1 <= 48) ||
(protonNumber1 == 28 && 20 <= neutronCount1 <= 50) ||
(protonNumber1 == 29 && 23 <= neutronCount1 <= 51) ||
(protonNumber1 == 30 && 24 <= neutronCount1 <= 53) ||
(protonNumber1 == 31 && 25 <= neutronCount1 <= 55) ||
(protonNumber1 == 32 && 26 <= neutronCount1 <= 57) ||
(protonNumber1 == 33 && 27 <= neutronCount1 <= 59) ||
(protonNumber1 == 34 && 31 <= neutronCount1 <= 60) ||
(protonNumber1 == 35 && 32 <= neutronCount1 <= 62) ||
(protonNumber1 == 36 && 33 <= neutronCount1 <= 64) ||
(protonNumber1 == 37 && 34 <= neutronCount1 <= 65) ||
(protonNumber1 == 38 && 35 <= neutronCount1 <= 67) ||
(protonNumber1 == 39 && 37 <= neutronCount1 <= 69) ||
(protonNumber1 == 40 && 38 <= neutronCount1 <= 70) ||
(protonNumber1 == 41 && 40 <= neutronCount1 <= 72) ||
(protonNumber1 == 42 && 41 <= neutronCount1 <= 73) ||
(protonNumber1 == 43 && 42 <= neutronCount1 <= 75) ||
(protonNumber1 == 44 && 43 <= neutronCount1 <= 76) ||
(protonNumber1 == 45 && 44 <= neutronCount1 <= 77) ||
(protonNumber1 == 46 && 45 <= neutronCount1 <= 78) ||
(protonNumber1 == 47 && 46 <= neutronCount1 <= 83) ||
(protonNumber1 == 48 && 47 <= neutronCount1 <= 84) ||
(protonNumber1 == 49 && 48 <= neutronCount1 <= 86) ||
(protonNumber1 == 50 && 49 <= neutronCount1 <= 87) ||
(protonNumber1 == 51 && 52 <= neutronCount1 <= 88) ||
(protonNumber1 == 52 && 53 <= neutronCount1 <= 90) ||
(protonNumber1 == 53 && 55 <= neutronCount1 <= 91) ||
(protonNumber1 == 54 && 56 <= neutronCount1 <= 93) ||
(protonNumber1 == 55 && 57 <= neutronCount1 <= 96) ||
(protonNumber1 == 56 && 58 <= neutronCount1 <= 97) ||
(protonNumber1 == 57 && 60 <= neutronCount1 <= 98) ||
(protonNumber1 == 58 && 61 <= neutronCount1 <= 99) ||
(protonNumber1 == 59 && 62 <= neutronCount1 <= 100) ||
(protonNumber1 == 60 && 64 <= neutronCount1 <= 101) ||
(protonNumber1 == 61 && 65 <= neutronCount1 <= 102) ||
(protonNumber1 == 62 && 66 <= neutronCount1 <= 103) ||
(protonNumber1 == 63 && 67 <= neutronCount1 <= 104) ||
(protonNumber1 == 64 && 70 <= neutronCount1 <= 105) ||
(protonNumber1 == 65 && 71 <= neutronCount1 <= 106) ||
(protonNumber1 == 66 && 72 <= neutronCount1 <= 107) ||
(protonNumber1 == 67 && 73 <= neutronCount1 <= 108) ||
(protonNumber1 == 68 && 75 <= neutronCount1 <= 109) ||
(protonNumber1 == 69 && 76 <= neutronCount1 <= 110) ||
(protonNumber1 == 70 && 78 <= neutronCount1 <= 111) ||
(protonNumber1 == 71 && 79 <= neutronCount1 <= 111) ||
(protonNumber1 == 72 && 81 ≤ neutronCount1 ≤ 116) ||
```

```
  (protonNumber1 == 73 && 82 ≤ neutronCount1 ≤ 117) ||
  (protonNumber1 == 74 && 84 ≤ neutronCount1 ≤ 118) ||
  (protonNumber1 == 75 && 85 ≤ neutronCount1 ≤ 119),
 (*Check nuclear chart range for second fragment*)
 If[(protonNumber2 == 22 && 16 <= neutronCount2 <= 41) ||
   (protonNumber2 == 23 && 17 <= neutronCount2 <= 42) ||
   (protonNumber2 == 24 && 18 <= neutronCount2 <= 43) ||
   (protonNumber2 == 25 && 19 <= neutronCount2 <= 44) ||
   (protonNumber2 == 26 && 19 <= neutronCount2 <= 46) ||
   (protonNumber2 == 27 && 20 <= neutronCount2 <= 48) ||
   (protonNumber2 == 28 && 20 <= neutronCount2 <= 50) ||
   (protonNumber2 == 29 && 23 <= neutronCount2 <= 51) ||
   (protonNumber2 == 30 && 24 <= neutronCount2 <= 53) ||
   (protonNumber2 == 31 && 25 <= neutronCount2 <= 55) ||
   (protonNumber2 == 32 && 26 <= neutronCount2 <= 57) ||
   (protonNumber2 == 33 && 27 <= neutronCount2 <= 59) ||
   (protonNumber2 == 34 && 31 <= neutronCount2 <= 60) ||
   (protonNumber2 == 35 && 32 <= neutronCount2 <= 62) ||
   (protonNumber2 == 36 && 33 <= neutronCount2 <= 64) ||
   (protonNumber2 == 37 && 34 <= neutronCount2 <= 65) ||
   (protonNumber2 == 38 && 35 <= neutronCount2 <= 67) ||
   (protonNumber2 == 39 && 37 <= neutronCount2 <= 69) ||
   (protonNumber2 == 40 && 38 <= neutronCount2 <= 70) ||
   (protonNumber2 == 41 && 40 <= neutronCount2 <= 72) ||
   (protonNumber2 == 42 && 41 <= neutronCount2 <= 73) ||
   (protonNumber2 == 43 && 42 <= neutronCount2 <= 75) ||
   (protonNumber2 == 44 && 43 <= neutronCount2 <= 76) ||
   (protonNumber2 == 45 && 44 <= neutronCount2 <= 77) ||
   (protonNumber2 == 46 && 45 <= neutronCount2 <= 78) ||
   (protonNumber2 == 47 && 46 <= neutronCount2 <= 83) ||
   (protonNumber2 == 48 && 47 <= neutronCount2 <= 84) ||
   (protonNumber2 == 49 && 48 <= neutronCount2 <= 86) ||
   (protonNumber2 == 50 && 49 <= neutronCount2 <= 87) ||
   (protonNumber2 == 51 && 52 <= neutronCount2 <= 88) ||
   (protonNumber2 == 52 && 53 <= neutronCount2 <= 90) ||
   (protonNumber2 == 53 && 55 <= neutronCount2 <= 91) ||
   (protonNumber2 == 54 && 56 <= neutronCount2 <= 93) ||
   (protonNumber2 == 55 && 57 <= neutronCount2 <= 96) ||
   (protonNumber2 == 56 && 58 <= neutronCount2 <= 97) ||
   (protonNumber2 == 57 && 60 <= neutronCount2 <= 98) ||
   (protonNumber2 == 58 && 61 <= neutronCount2 <= 99) ||
   (protonNumber2 == 59 && 62 <= neutronCount2 <= 100) ||
   (protonNumber2 == 60 && 64 <= neutronCount2 <= 101) ||
   (protonNumber2 == 61 && 65 <= neutronCount2 <= 102) ||
   (protonNumber2 == 62 && 66 <= neutronCount2 <= 103) ||
   (protonNumber2 == 63 && 67 <= neutronCount2 <= 104) ||
   (protonNumber2 == 64 && 70 <= neutronCount2 <= 105) ||
   (protonNumber2 == 65 && 71 <= neutronCount2 <= 106) ||
   (protonNumber2 == 66 && 72 <= neutronCount2 <= 107) ||
   (protonNumber2 == 67 && 73 <= neutronCount2 <= 108) ||
```

```
(protonNumber2 == 68 && 75 <= neutronCount2 <= 109) ||
(protonNumber2 == 69 && 76 <= neutronCount2 <= 110) ||
(protonNumber2 == 70 && 78 <= neutronCount2 <= 111) ||
(protonNumber2 == 71 && 79 <= neutronCount2 <= 111) ||
(protonNumber2 == 72 && 81 ≤ neutronCount2 ≤ 116) ||
(protonNumber2 == 73 && 82 ≤ neutronCount2 ≤ 117) ||
(protonNumber2 == 74 && 84 ≤ neutronCount2 ≤ 118) ||
(protonNumber2 == 75 && 85 ≤ neutronCount2 ≤ 119),
(*Calculate proton number ratio*) chargeRatio =
 N[(protonNumber1 / massNumber1) / (protonNumber2 / massNumber2)];


(*Update variable lists based on energy region*)
Switch[energyIndex, 1, fermiVars0253eV = Union[AppendTo[distanceVars0253eV,
    effectiveDistance0253eV[protonNumber1, protonNumber2]]],
 2, fermiVars500keV = Union[AppendTo[distanceVars500keV,
    effectiveDistance500keV[protonNumber1, protonNumber2]]],
 3, fermiVars14MeV = Union[AppendTo[distanceVars14MeV,
    effectiveDistance14MeV[protonNumber1, protonNumber2]]]];

(*Calculate physical parameters*)
effectiveDistanceVal = effectiveDistance[protonNumber1, protonNumber2];
coulombEnergy = (1.44 * protonNumber1 * protonNumber2) / effectiveDistanceVal;

(*Calculating Q value*)
qValue = (getNuclearMass[atomicNumber, atomicNumber + neutronNumber] -
    getNuclearMass[protonNumber1, massNumber1] -
    getNuclearMass[protonNumber2, massNumber2] -
    promptNeutronCount * 1.008665) * 931.4940954;

effectiveEnergy = coulombEnergy - qValue;

(*Calculate fission probability*)
probability = 1 / (1 + Exp[2 * Pi / (neutronSeparationEnergy + incidentEnergy) *
       Sqrt[protonNumber1 * protonNumber2 / (protonNumber1 + protonNumber2)] /
        reducedMass * effectiveEnergy]);


    Sow[{protonNumber1, probability}, yieldList];]]]],
   {yieldList}, Rule][[2, All, 1]];
 (*Process results*)
 fragmentData = Part[yieldList /. dataTemp];
 processYields[data_] :=
  (Total@# / {Length@#, Total@data[[All, 2]] / 2} &) /@ GatherBy[data, First];
 AppendTo[results, {energyIndex, processYields[fragmentData]}];];
results]

(*Execute main calculation*)
fissionResults = CalculateFissionYields[energyPattern];

(*Process results based on energy pattern*)
{yieldData0253eVCalc, yieldData500keVCalc, yieldData14MeVCalc} =
```

```
Switch[energyPattern, 1, {fissionResults[[1, 2]], Null, Null},
 2, {fissionResults[[1, 2]], fissionResults[[2, 2]], Null}, 3,
 {fissionResults[[1, 2]], fissionResults[[2, 2]], fissionResults[[3, 2]]},
 4, {Null, fissionResults[[1, 2]], Null}, 5,
 {Null, fissionResults[[1, 2]], fissionResults[[2, 2]]}];
```

```
(*4*) (*Optimization and Result Display Program for Fission Parameters*)
For[energyRegion = startEnergyIndex, energyRegion ≤ endEnergyIndex,
 energyRegion++, (*Setup variables and data for each energy region*)
 {neutronVarList, calcYieldData, fitYieldData, experimentalData, energyDescription,
   theoreticalYield, optimizationResult, fittedData} = Switch[energyRegion, 1,
   {paramList0253eV = Union[fermiVars0253eV], yieldData0253eVCalc, fitYield0253eV =
     yieldData0253eVCalc[[fitStartIndex ;; fitEndIndex]], yieldData0253eV[[
     fitStartIndex ;; fitEndIndex]], "1. Incident Neutron Energy: 0.0253eV",
    theoreticalYield0253eV, optResult0253eV, fittedYield0253eV}, 2,
   {paramList500keV = Union[fermiVars500keV], yieldData500keVCalc,
    fitYield500keV = yieldData500keVCalc[[fitStartIndex ;; fitEndIndex]],
    yieldData500keV[[fitStartIndex ;; fitEndIndex]],
    "2. Incident Neutron Energy: 500keV", theoreticalYield500keV,
    optResult500keV, fittedYield500keV}, 3,
   {paramList14MeV = Union[fermiVars14MeV], yieldData14MeVCalc,
    fitYield14MeV = yieldData14MeVCalc[[fitStartIndex ;; fitEndIndex]],
    yieldData14MeV[[fitStartIndex ;; fitEndIndex]],
    "3. Incident Neutron Energy: 14MeV", theoreticalYield14MeV,
    optResult14MeV, fittedYield14MeV}];
 (*Calculate logarithmic difference between theory and experiment*)
 logDifference = (Log@fitYieldData - Log@experimentalData) [[All, 2]];
```

```
 (*Optimize parameters using least squares method*)
 Switch[energyRegion, 1,
  fittedYield0253eV = optResult0253eVRe1[[2, All, All]];
  theoreticalYield0253eV = fitYield0253eV /. optResult0253eVRe1[[2, All, All]], 2,
  fittedYield500keV = optResult500keVRe1[[2, All, All]];
  theoreticalYield500keV = fitYield500keV /. optResult500keVRe1[[2, All, All]], 3,
  fittedYield14MeV = optResult14MeVRe1[[2, All, All]];
  theoreticalYield14MeV = fitYield14MeV /. optResult14MeVRe1[[2, All, All]]];
```

```
 (*Display results*)
 Print[Style[energyDescription <> ": Effective Fission Distance Reff
     derived from experimental charge distribution", 16]];
```

```
 Print["Analysis Results"];
 Print[
  "Calculation results demonstrating that the effective fission distance Reff " <>
   "derived from optimization calculations
     accurately reproduces experimental values " <>
```

```
    "(confirming agreement between JENDL-5 experimental and theoretical values, " <>
    "and validating calculations using Mathematica ver11.2 FindMinimum)"];


  (*Create visualization plot*)
  plotOptions = {Joined → {True, True}, PlotRange → {{15, 80}, {10^(-12), 5}},
    Epilog → Inset[Style[isotopeName, Bold, 20], Scaled@{0.14, 0.9}],
    PlotMarkers → Automatic, PlotStyle →
     {Directive[PointSize[1/100], Red], Directive[PointSize[1/100], Blue]},
    Frame → True, FrameLabel → {"Atomic Number", "Fission Yield (Independent)"},
    LabelStyle → Directive[Black, 19], FrameTicks → Automatic,
    FrameStyle → {Thick, Thick, Thick, Thick},
    PlotLegends → Placed[PointLegend[Automatic, {"JENDL-5", "Theoretical"},
       Joined → {True, True, True}, Joined → {True, True},
       LabelStyle → Directive[Black, 18], LegendFunction → "Frame",
       LegendLayout → "Column", LegendMarkers → Array[{Graphics@Disk[], 10} &, 3]],
      {{0.65, 0.25}, {1, 0.9}}], AspectRatio → 0.8, ImageSize → 450,
    Epilog → Inset[Style[isotopeName, Bold], Scaled@{0.1, 0.92}]};
  Print[ListLogPlot[{experimentalData, theoreticalYield}, Evaluate[plotOptions]]];


  Print[
   "--------------------------------------------------------------------------
     ---"];];




(*5*)
(*Correlation Analysis of Fragment Charge Product and Effective Fission Distance*)
(*Process data for each energy condition*)
Do[With[{condition = Which[i == 1, {energyPattern == 1 || energyPattern == 2 ||
         energyPattern == 3, correlationData0253eV, fittedYield0253eV,
        "1. Analysis for Incident Neutron Energy: 0.0253eV"}, i == 2,
       {energyPattern == 2 || energyPattern == 3 || energyPattern == 4 || energyPattern == 5,
        correlationData500keV, fittedYield500keV,
        "2. Analysis for Incident Neutron Energy: 500keV"}, i == 3,
       {energyPattern == 3 || energyPattern == 5, correlationData14MeV, fittedYield14MeV,
        "3. Analysis for Incident Neutron Energy: 14MeV"}]}, If[condition[[1]],
    Switch[condition[[2]], correlationData0253eV, correlationData0253eV =
      Thread[{condition[[3]][[All, 1, 1]], condition[[3]][[All, 2]]}][[
       fitStartIndex ;; fitEndIndex]], correlationData500keV, correlationData500keV =
      Thread[{condition[[3]][[All, 1, 1]], condition[[3]][[All, 2]]}][[
       fitStartIndex ;; fitEndIndex]], correlationData14MeV, correlationData14MeV =
      Thread[{condition[[3]][[All, 1, 1]], condition[[3]][[All, 2]]}][[
       fitStartIndex ;; fitEndIndex]]]]], {i, 1, 3}];

Print[
  "----------------------------------------------------------------------------
    -"];


(*Result explanation and plotting*)
Module[{plotOptions = {Joined → {True, True, True}, PlotRange → {{15, 85}, {0.8, 1.3}},
     Frame → True, FrameLabel → {"Atomic Number", "Effective Distance (fm)"},
```

```
      LabelStyle → Directive[Black, 19], FrameTicks → Automatic,
      FrameStyle → {Thick, Thick, Thick, Thick}, AspectRatio → 1.1, ImageSize → 350}},
   (*Define fitting function*)analyzeFittingResults[correlationData_, energyLabel_] :=
    Module[{distanceFunction, coeffA, coeffB, coeffC, chargeNumber, normalizedFormula},
     Print[Style[energyLabel, FontSize → 16]];
     (*Fit quadratic function*)distanceFunction =
      Fit[correlationData, {1, x, x^2}, x];
     (*Extract coefficients*){coeffC, coeffB, coeffA} =
      CoefficientList[distanceFunction, x];
     (*Calculate characteristic charge number*)kValue = -coeffA;
     chargeNumber = coeffB / kValue;
     mValue = coeffC;
     (*Display formula*)Print[Column[{Style[HoldForm[Reff] ==
          N[mValue, 6] + N[kValue, 6] * (N[chargeNumber, 6] - x) * x, FontSize → 16]}]];
     distanceFunction (*Return function for later use*)];
   Print["\nAnalysis Results:"];
   Print["1. The effective fission distance
      Reff shows quadratic dependence on fragment charge"];
   Print["2. This dependence reflects fundamental laws of charge
      distribution in fission process"];
   Print["3. Similar dependence is maintained across different incident energies\n"];
   (*Create comparison plot*)plotData = Select[{correlationData0253eV,
      correlationData500keV, correlationData14MeV}, Length[#] > 0 &];

   (*Create data-dependent color and label lists*)colors = {};
labels = {};
If[Length[correlationData0253eV] > 0, AppendTo[colors, Blue];
   AppendTo[labels, "0.0253 eV"]];
If[Length[correlationData500keV] > 0, AppendTo[colors, Green];
   AppendTo[labels, "500 keV"]];
If[Length[correlationData14MeV] > 0, AppendTo[colors, Red];
   AppendTo[labels, "14 MeV"]];

   plotData = Select[{correlationData0253eV,
      correlationData500keV, correlationData14MeV}, Length[#] > 0 &];

   (*Execute plot*)
   Print[ListPlot[plotData, Evaluate[plotOptions],
     PlotStyle → (Directive[PointSize[1 / 100], #] & /@ colors),
     PlotLegends → Placed[LineLegend[colors, labels, LabelStyle → 14], {0.82, 0.85}],
     Epilog → {Inset[Style[isotopeName, Bold, 15], Scaled@{0.15, 0.85}],
       Inset[Style[databaseName, Bold, 15], Scaled@{0.15, 0.90}]}]]];
```

```
(*Define distance functions for each energy level*)
Module[{fitResult}, (*Define fitting and display functions*)
 fitAndPrint[correlationData_, energyLabel_] :=
  Module[{distanceFunction, coeffA, coeffB, coeffC, chargeNumber, normalizedFormula},
   Print[Style[energyLabel, FontSize → 16]];
   (*Fit with quadratic function*)
   distanceFunction = Fit[correlationData, {1, x, x^2}, x];
   (*Extract coefficients*){coeffC, coeffB, coeffA} =
```

```
    CoefficientList[distanceFunction, x];
   (*Calculate characteristic charge number*) kValue = -coeffA;
   chargeNumber = coeffB / kValue;
   mValue = coeffC;
   (*Display formula*) Print[Column[{Style[HoldForm[Reff] ==
        N[mValue, 6] + N[kValue, 6] * (N[chargeNumber, 6] - x) * x, FontSize → 16]}]];
   distanceFunction (*Return function*)];
 (*For 0.0253 eV case*)
 If[energyPattern == 1 || energyPattern == 2 || energyPattern == 3, distanceFunction0253eV =
   fitAndPrint[correlationData0253eV, "Incident Neutron Energy: 0.0253 eV case"]];
 (*For 500 keV case*) If[energyPattern == 2 || energyPattern == 3 ||
   energyPattern == 4 || energyPattern == 5, distanceFunction500keV =
   fitAndPrint[correlationData500keV, "Incident Neutron Energy: 500 keV case"]];
 (*For 14 MeV case*) If[energyPattern == 3 || energyPattern == 5, distanceFunction14MeV =
   fitAndPrint[correlationData14MeV, "Incident Neutron Energy: 14 MeV case"]];
 Print["\nAnalysis Results:"];
 Print["1. The effective fission distance
    Reff shows quadratic dependence on fragment charge number"];
 Print["2. This dependence reflects fundamental laws of charge
    distribution in the fission process"];
 Print["3. Similar dependence is maintained across different incident energies\n"];]
```

```
(*6*) (*Display and Analysis of Reff Values by Atomic Number*)
(*Output header for display*) Print[Style[
   "Table of Effective Fission Distance (Reff) Values by Atomic Number [Unit: fm]:",
   Bold, 16]];
Print[Style["Displaying calculated values (pre-fitting)
     and post-fitting values for each energy", 14]];
Print[Style["Fitting calculation used fragment values from atomic number " <>
    ToString[fitStartZ] <> " to " <> ToString[fitEndZ], 14]];
```

```
(*Create headers based on energy pattern*)
tableHeaders = Switch[energyPattern, 1,
   {"Z1", "Z2", "Reff [fm]\n(0.0253 eV)\nCalculated", "Fitted Value"},
   2, {"Z1", "Z2", "Reff [fm]\n(0.0253 eV)\nCalculated", "Fitted Value",
    "Reff [fm]\n(500 keV)\nCalculated", "Fitted Value"}, 3,
   {"Z1", "Z2", "Reff [fm]\n(0.0253 eV)\nCalculated", "Fitted Value",
    "Reff [fm]\n(500 keV)\nCalculated", "Fitted Value",
    "Reff [fm]\n(14 MeV)\nCalculated", "Fitted Value"}, 4,
   {"Z1", "Z2", "Reff [fm]\n(500 keV)\nCalculated", "Fitted Value"}, 5,
   {"Z1", "Z2", "Reff [fm]\n(500 keV)\nCalculated", "Fitted Value",
    "Reff [fm]\n(14 MeV)\nCalculated", "Fitted Value"}];

(*Function to get fitted value*)
getFittedDistanceValue[atomicNumber_, distanceFunction_] :=
  If[fitStartZ ≤ atomicNumber ≤ fitEndZ,
   NumberForm[N[distanceFunction /. x → atomicNumber], {6, 5}], "-"];

(*Create data table*)
tableData =
  Table[Module[{z1 = z, z2 = atomicNumber - z}, Flatten[{z1, z2, (*0.0253 eV data*)
```

```
      Which[energyPattern == 1 || energyPattern == 2 || energyPattern == 3,
       {NumberForm[N[effectiveDistance0253eV[z1, z2] /. fittedYield0253eV], {6, 6}],
        getFittedDistanceValue[z1, distanceFunction0253eV]}, True, {}],
      (*500 keV data*)Which[energyPattern == 2 || energyPattern == 3 ||
        energyPattern == 4 || energyPattern == 5,
       {NumberForm[N[effectiveDistance500keV[z1, z2] /. fittedYield500keV], {6, 6}],
        getFittedDistanceValue[z1, distanceFunction500keV]}, True, {}],
      (*14 MeV data*)Which[energyPattern == 3 || energyPattern == 5,
       {NumberForm[N[effectiveDistance14MeV[z1, z2] /. fittedYield14MeV], {6, 6}],
        getFittedDistanceValue[z1, distanceFunction14MeV]},
       True, {}]}]], {z, 23, 69}];



(*Output formatted table*)
Grid[Prepend[tableData, tableHeaders], Frame → All, Alignment → Center,
 Background → {None, {LightGray, None}}, ItemStyle → {Bold, "Text"},
 Dividers → {Join[{2}, Table[2 i + 2, {i, 1, Length[tableHeaders] / 2 - 1}]] → True,
    {2 → True}}, Spacings → {1.5, 1.2}]

(*7*)(*Generate Effective Fission Distance Functions*)
(*Generate functions for each energy region*)
If[energyPattern == 1 || energyPattern == 2 || energyPattern == 3,
  distanceData0253eV = Table[{x, atomicNumber - x} → distanceFunction0253eV,
    {x, fitStartZ - 6, fitEndZ + 6}]];

If[energyPattern == 2 || energyPattern == 3 || energyPattern == 4 || energyPattern == 5,
  distanceData500keV = Table[{x, atomicNumber - x} → distanceFunction500keV,
    {x, fitStartZ - 6, fitEndZ + 6}]];

If[energyPattern == 3 || energyPattern == 5, distanceData14MeV =
   Table[{x, atomicNumber - x} → distanceFunction14MeV, {x, fitStartZ - 6, fitEndZ + 6}]];

(*Define effective distance functions*)
If[energyPattern == 1 || energyPattern == 2 || energyPattern == 3, effDistanceFunc0253eV =
   Thread[Thread[effectiveDistance0253eV[distanceData0253eV[[All, 1, 1]],
        distanceData0253eV[[All, 1, 2]]]] → distanceData0253eV[[All, 2]]]];

If[energyPattern == 2 || energyPattern == 3 || energyPattern == 4 || energyPattern == 5,
  effDistanceFunc500keV =
   Thread[Thread[effectiveDistance500keV[distanceData500keV[[All, 1, 1]],
        distanceData500keV[[All, 1, 2]]]] → distanceData500keV[[All, 2]]]];

If[energyPattern == 3 || energyPattern == 5, effDistanceFunc14MeV =
   Thread[Thread[effectiveDistance14MeV[distanceData14MeV[[All, 1, 1]],
        distanceData14MeV[[All, 1, 2]]]] → distanceData14MeV[[All, 2]]]];

(*Calculate final yield data*)
If[energyPattern == 1 || energyPattern == 2 || energyPattern == 3,
  finalYield0253eV = fitYield0253eV /. effDistanceFunc0253eV];

If[energyPattern == 2 || energyPattern == 3 || energyPattern == 4 || energyPattern == 5,
  finalYield500keV = fitYield500keV /. effDistanceFunc500keV];

If[energyPattern == 3 || energyPattern == 5,
```

```
  finalYield14MeV = fitYield14MeV /. effDistanceFunc14MeV];

(*Display separator*)
Print[
  "--------------------------------------------------------------------------
    -"];


(*8*)(*Comparison of Experimental and Theoretical Fission Yields*)
(*Display analysis summary*)
Print["Analysis of Fission Yields: Comparison
    between Experimental Data and Theoretical Calculations (Ex=0)"];
Print["\nTheoretical Analysis Results using Effective Fission
    Distance Reff proportional to fragment charge product,"];
Print["with zero Fermi Energy (Ex=0)"];
Print["- Quantitative reproduction of
    experimentally observed asymmetric fission yield distributions"];



(*Create common plot settings*)
commonPlotSettings = {Joined → {True, True},
   PlotRange → {{15, 80}, {10^(-12), 100}}, PlotMarkers → Automatic,
   Frame → True, FrameLabel → {"Atomic Number", "Fission Yield (Independent)
       "}, LabelStyle → Directive[Black, 19], FrameTicks → Automatic,
   FrameStyle → {Thick, Thick, Thick, Thick}, AspectRatio → 0.8, ImageSize → 450};

(*Legend settings*)
legendSettings = Placed[PointLegend[Automatic,
    {"JENDL-5 (Experimental)", "Theoretical curve (Fermi Energy=0)"},
    Joined → {True, True}, LabelStyle → Directive[Black, 15],
    LegendFunction → "Frame", LegendLayout → "Column",
    LegendMarkers → Array[{Graphics@Disk[], 10} &, 3]], {{0.26, 0.23}, {0.2, 0.9}}];


(*Plot style settings*)
plotStyles = {Directive[PointSize[1/100], Red],
   Directive[PointSize[1/100], Blue], Directive[PointSize[1/100], Green]};




(*Generate plots for each energy region*)
If[energyPattern == 1 || energyPattern == 2 || energyPattern == 3,
  Print["1. Analysis for Incident Neutron Energy: 0.0253eV"];
  Print[ListLogPlot[{yieldData0253eV[[fitStartIndex ;; fitEndIndex]],
     finalYield0253eV}, Evaluate[commonPlotSettings],
    PlotStyle → plotStyles, PlotLegends → legendSettings,
    Epilog → {Inset[Style[isotopeName, 20, Bold], Scaled@{0.16, 0.9}]}]];];

If[energyPattern == 2 || energyPattern == 3 || energyPattern == 4 || energyPattern == 5,
  Print["2. Analysis for Incident Neutron Energy: 500keV"];
  Print[ListLogPlot[{yieldData500keV[[fitStartIndex ;; fitEndIndex]],
     finalYield500keV}, Evaluate[commonPlotSettings],
```

```
    PlotStyle → plotStyles, PlotLegends → legendSettings,
    Epilog → {Inset[Style[isotopeName, 20, Bold], Scaled@{0.16, 0.9}]}]];];

If[energyPattern == 3 || energyPattern == 5,
  Print["3. Analysis for Incident Neutron Energy: 14MeV"];
  Print[ListLogPlot[{yieldData14MeV[[fitStartIndex ;; fitEndIndex]], finalYield14MeV},
    Evaluate[commonPlotSettings], PlotStyle → plotStyles, PlotLegends → legendSettings,
    Epilog → {Inset[Style[isotopeName, 20, Bold], Scaled@{0.16, 0.9}]}]];];




(*9*)(*Analysis of Effective Fission Distance and Fission Probability*)
(*Display section separator*)Print[
  "--------------------------------------------------------------------------
    -"];
Print["Analysis of Effective Fission Distance and Fission Probability"];
Print[""];
Print["Analysis Contents:"];
Print["1. Calculation of Effective
    Fission Distance (Reff) for each incident neutron energy"];
Print["2. Evaluation of fission probability using η function"];
Print["3. Derivation of normalization factor κ (= Ex/Reff)"];
Print[""];
Print["The vertical axis κ represents the ratio of Fermi Energy(Ex)"];
Print["to effective fission distance (Reff)."];
Print[
  "κ ≈ 1 suggests the fission Fermi Energy is proportional to effective distance."];
Print[
  "--------------------------------------------------------------------------
    -"];

(*Analysis and visualization module*)
Module[{}, (*Process for each incident neutron energy*)
  (*1. Calculate effective fission distance function values*)
  (*2. Calculate fission probability using η function*)
  (*3. Calculate normalization factor κ*)
  If[energyPattern == 1 || energyPattern == 2 || energyPattern == 3, distanceData0253eV =
    Table[{x, atomicNumber - x} → distanceFunction0253eV, {x, fitStartZ, fitEndZ}];
   distanceFunc0253eV = Thread[Thread[effectiveDistance0253eV[distanceData0253eV[[All,
         1, 1]], distanceData0253eV[[All, 1, 2]]]] → distanceData0253eV[[All, 2]]];
   normFactor0253eV = Thread[{distanceData0253eV[[All, 1, 1]],
      fittedYield0253eV[[All, 2]][[fitStartIndex ;; fitEndIndex]] /
       distanceFunc0253eV[[All, 2]]}];];
  If[energyPattern == 2 || energyPattern == 3 || energyPattern == 4 || energyPattern == 5,
   distanceData500keV =
    Table[{x, atomicNumber - x} → distanceFunction500keV, {x, fitStartZ, fitEndZ}];
   distanceFunc500keV = Thread[Thread[effectiveDistance500keV[distanceData500keV[[All,
         1, 1]], distanceData500keV[[All, 1, 2]]]] → distanceData500keV[[All, 2]]];
   normFactor500keV = Thread[{distanceData500keV[[All, 1, 1]],
      fittedYield500keV[[All, 2]][[fitStartIndex ;; fitEndIndex]] /
       distanceFunc500keV[[All, 2]]}];];
  If[energyPattern == 3 || energyPattern == 5, distanceData14MeV =
    Table[{x, atomicNumber - x} → distanceFunction14MeV, {x, fitStartZ, fitEndZ}];
```

```
   distanceFunc14MeV = Thread[Thread[effectiveDistance14MeV[distanceData14MeV[[All,
         1, 1]], distanceData14MeV[[All, 1, 2]]]] → distanceData14MeV[[All, 2]]];
   normFactor14MeV = Thread[{distanceData14MeV[[All, 1, 1]],
      fittedYield14MeV[[All, 2]][[fitStartIndex ;; fitEndIndex]] /
       distanceFunc14MeV[[All, 2]]}];];
  (*Common plot options*)
  plotOptions =
   {Joined → {True, True}, PlotRange → {{15, 80}, {0.9, 1.1}}, PlotMarkers → Automatic,
    Frame → True, FrameLabel → {"Atomic Number", "Normalization Factor ϰ = Ex/Reff"},
    LabelStyle → Directive[Black, 19], FrameTicks → Automatic,
    FrameStyle → {Thick, Thick, Thick, Thick}, AspectRatio → 1, ImageSize → 400};
  (*Create visualization based on energy pattern*)
  Switch[energyPattern, 1, Print[ListPlot[{normFactor0253eV},
     Evaluate[plotOptions], PlotStyle → {Directive[PointSize[1/100], Blue]},
     PlotLegends → Placed[PointLegend[{"0.0253eV"}, LabelStyle → 14,
        LegendFunction → "Frame"], {{0.975, 0.925}, {1, 0.9}}],
     Epilog → {Inset[Style[isotopeName, Bold, 14], Scaled@{0.15, 0.95}],
       Inset[Style[databaseName, Bold, 14], Scaled@{0.15, 0.90}]}]], 2, Print[
    ListPlot[{normFactor0253eV, normFactor500keV}, Evaluate[plotOptions], PlotStyle →
      {Directive[PointSize[1/100], Blue], Directive[PointSize[1/100], Green]},
     PlotLegends → Placed[PointLegend[{"0.0253eV", "500keV"}, LabelStyle → 14,
        LegendFunction → "Frame"], {{0.975, 0.925}, {1, 0.9}}],
     Epilog → {Inset[Style[isotopeName, Bold, 14], Scaled@{0.15, 0.95}],
       Inset[Style[databaseName, Bold, 14], Scaled@{0.15, 0.90}]}]], 3,
   Print[ListPlot[{normFactor0253eV, normFactor500keV, normFactor14MeV},
     Evaluate[plotOptions], PlotStyle → {Directive[PointSize[1/100], Blue],
       Directive[PointSize[1/100], Green], Directive[PointSize[1/100], Red]},
     PlotLegends → Placed[PointLegend[{"0.0253eV", "500keV", "14MeV"},
        LabelStyle → 14, LegendFunction → "Frame"], {{0.975, 0.925}, {1, 0.9}}],
     Epilog → {Inset[Style[isotopeName, Bold, 14], Scaled@{0.15, 0.95}],
       Inset[Style[databaseName, Bold, 14], Scaled@{0.15, 0.90}]}]],
   4, Print[ListPlot[{normFactor500keV}, Evaluate[plotOptions],
     PlotStyle → {Directive[PointSize[1/100], Green]},
     PlotLegends → Placed[PointLegend[{"500keV"}, LabelStyle → 14,
        LegendFunction → "Frame"], {{0.975, 0.925}, {1, 0.9}}],
     Epilog → {Inset[Style[isotopeName, Bold, 14], Scaled@{0.15, 0.95}],
       Inset[Style[databaseName, Bold, 14], Scaled@{0.15, 0.90}]}]], 5, Print[
    ListPlot[{normFactor500keV, normFactor14MeV}, Evaluate[plotOptions], PlotStyle →
      {Directive[PointSize[1/100], Green], Directive[PointSize[1/100], Red]},
     PlotLegends → Placed[PointLegend[{"500keV", "14MeV"}, LabelStyle → 14,
        LegendFunction → "Frame"], {{0.975, 0.925}, {1, 0.9}}],
     Epilog → {Inset[Style[isotopeName, Bold, 14], Scaled@{0.15, 0.95}],
       Inset[Style[databaseName, Bold, 14], Scaled@{0.15, 0.90}]}]]];];

Print[
  "----------------------------------------------------------------------------
    -"];


(*10*)(*Initialize Fermi Energy Variables and Main Calculation Function*)
(*Initialize Fermi Energy*)
```

```
fermiEnergy0253eV = .;
fermiEnergy500keV = .;
fermiEnergy14MeV = .;

(*Initialize parameter lists for each energy region*)
paramList0253eV = {};
paramList500keV = {};
paramList14MeV = {};

paramList5 = {};
paramList6 = {};
paramList7 = {};

(*Nuclear Fission Calculation Main Function*)
CalculateFissionYieldsWithfermi[energyPattern_] := Module[{tempData, results = {}},
  For[energyIndex = startEnergyIndex, energyIndex ≤ endEnergyIndex,
   energyIndex++, (*Set parameters for each energy region*)
   {promptNeutronCount, incidentEnergy, fermiEnergy, parameterList, yieldList,
     neutronVarList, effectiveDist, distanceFunc} = Switch[energyIndex, 1,
     {promptNeutrons1, neutronEnergy1, fermiEnergy0253eV, paramList0253eV,
      fissionYield0253eV, paramList5, effectiveDist0253eV, distanceFunction0253eV},
     (*Thermal*)2, {promptNeutrons2, neutronEnergy2, fermiEnergy500keV,
      paramList500keV, fissionYield500keV, paramList6, effectiveDist500keV,
      distanceFunction500keV}, (*Intermediate*)3, {promptNeutrons3,
      neutronEnergy3, fermiEnergy14MeV, paramList14MeV, fissionYield1400keV,
      paramList7, effectiveDist14MeV, distanceFunction14MeV} (*Fast*)];
   tempData = Reap[For[protonNumber1 = 23, protonNumber1 ≤ atomicNumber - 23,
       protonNumber1++, For[neutronCount1 = 0, neutronCount1 ≤ neutronNumber,
        neutronCount1++, protonNumber2 = atomicNumber - protonNumber1;
        neutronCount2 = neutronNumber - neutronCount1 - Round[promptNeutronCount];
        massNumber1 = protonNumber1 + neutronCount1;
        massNumber2 = protonNumber2 + neutronCount2;


        (*Check fragment existence*)

        If[ (protonNumber1 == 22 && 16 <= neutronCount1 <= 41) ||
          (protonNumber1 == 23 && 17 <= neutronCount1 <= 42) ||
          (protonNumber1 == 24 && 18 <= neutronCount1 <= 43) ||
          (protonNumber1 == 25 && 19 <= neutronCount1 <= 44) ||
          (protonNumber1 == 26 && 19 <= neutronCount1 <= 46) ||
          (protonNumber1 == 27 && 20 <= neutronCount1 <= 48) ||
          (protonNumber1 == 28 && 20 <= neutronCount1 <= 50) ||
          (protonNumber1 == 29 && 23 <= neutronCount1 <= 51) ||
          (protonNumber1 == 30 && 24 <= neutronCount1 <= 53) ||
          (protonNumber1 == 31 && 25 <= neutronCount1 <= 55) ||
          (protonNumber1 == 32 && 26 <= neutronCount1 <= 57) ||
          (protonNumber1 == 33 && 27 <= neutronCount1 <= 59) ||
          (protonNumber1 == 34 && 31 <= neutronCount1 <= 60) ||
          (protonNumber1 == 35 && 32 <= neutronCount1 <= 62) ||
          (protonNumber1 == 36 && 33 <= neutronCount1 <= 64) ||
          (protonNumber1 == 37 && 34 <= neutronCount1 <= 65) ||
          (protonNumber1 == 38 && 35 <= neutronCount1 <= 67) ||
```

```
(protonNumber1 == 39 && 37 <= neutronCount1 <= 69) ||
(protonNumber1 == 40 && 38 <= neutronCount1 <= 70) ||
(protonNumber1 == 41 && 40 <= neutronCount1 <= 72) ||
(protonNumber1 == 42 && 41 <= neutronCount1 <= 73) ||
(protonNumber1 == 43 && 42 <= neutronCount1 <= 75) ||
(protonNumber1 == 44 && 43 <= neutronCount1 <= 76) ||
(protonNumber1 == 45 && 44 <= neutronCount1 <= 77) ||
(protonNumber1 == 46 && 45 <= neutronCount1 <= 78) ||
(protonNumber1 == 47 && 46 <= neutronCount1 <= 83) ||
(protonNumber1 == 48 && 47 <= neutronCount1 <= 84) ||
(protonNumber1 == 49 && 48 <= neutronCount1 <= 86) ||
(protonNumber1 == 50 && 49 <= neutronCount1 <= 87) ||
(protonNumber1 == 51 && 52 <= neutronCount1 <= 88) ||
(protonNumber1 == 52 && 53 <= neutronCount1 <= 90) ||
(protonNumber1 == 53 && 55 <= neutronCount1 <= 91) ||
(protonNumber1 == 54 && 56 <= neutronCount1 <= 93) ||
(protonNumber1 == 55 && 57 <= neutronCount1 <= 96) ||
(protonNumber1 == 56 && 58 <= neutronCount1 <= 97) ||
(protonNumber1 == 57 && 60 <= neutronCount1 <= 98) ||
(protonNumber1 == 58 && 61 <= neutronCount1 <= 99) ||
(protonNumber1 == 59 && 62 <= neutronCount1 <= 100) ||
(protonNumber1 == 60 && 64 <= neutronCount1 <= 101) ||
(protonNumber1 == 61 && 65 <= neutronCount1 <= 102) ||
(protonNumber1 == 62 && 66 <= neutronCount1 <= 103) ||
(protonNumber1 == 63 && 67 <= neutronCount1 <= 104) ||
(protonNumber1 == 64 && 70 <= neutronCount1 <= 105) ||
(protonNumber1 == 65 && 71 <= neutronCount1 <= 106) ||
(protonNumber1 == 66 && 72 <= neutronCount1 <= 107) ||
(protonNumber1 == 67 && 73 <= neutronCount1 <= 108) ||
(protonNumber1 == 68 && 75 <= neutronCount1 <= 109) ||
(protonNumber1 == 69 && 76 <= neutronCount1 <= 110) ||
(protonNumber1 == 70 && 78 <= neutronCount1 <= 111) ||
(protonNumber1 == 71 && 79 <= neutronCount1 <= 111) ||
(protonNumber1 == 72 && 81 ≤ neutronCount1 ≤ 116) ||
(protonNumber1 == 73 && 82 ≤ neutronCount1 ≤ 117) ||
(protonNumber1 == 74 && 84 ≤ neutronCount1 ≤ 118) ||
(protonNumber1 == 75 && 85 ≤ neutronCount1 ≤ 119),
(*Check nuclear chart range for second fragment*)
If[(protonNumber2 == 22 && 16 <= neutronCount2 <= 41) ||
  (protonNumber2 == 23 && 17 <= neutronCount2 <= 42) ||
  (protonNumber2 == 24 && 18 <= neutronCount2 <= 43) ||
  (protonNumber2 == 25 && 19 <= neutronCount2 <= 44) ||
  (protonNumber2 == 26 && 19 <= neutronCount2 <= 46) ||
  (protonNumber2 == 27 && 20 <= neutronCount2 <= 48) ||
  (protonNumber2 == 28 && 20 <= neutronCount2 <= 50) ||
  (protonNumber2 == 29 && 23 <= neutronCount2 <= 51) ||
  (protonNumber2 == 30 && 24 <= neutronCount2 <= 53) ||
  (protonNumber2 == 31 && 25 <= neutronCount2 <= 55) ||
  (protonNumber2 == 32 && 26 <= neutronCount2 <= 57) ||
  (protonNumber2 == 33 && 27 <= neutronCount2 <= 59) ||
```

```
(protonNumber2 == 34 && 31 <= neutronCount2 <= 60) ||
(protonNumber2 == 35 && 32 <= neutronCount2 <= 62) ||
(protonNumber2 == 36 && 33 <= neutronCount2 <= 64) ||
(protonNumber2 == 37 && 34 <= neutronCount2 <= 65) ||
(protonNumber2 == 38 && 35 <= neutronCount2 <= 67) ||
(protonNumber2 == 39 && 37 <= neutronCount2 <= 69) ||
(protonNumber2 == 40 && 38 <= neutronCount2 <= 70) ||
(protonNumber2 == 41 && 40 <= neutronCount2 <= 72) ||
(protonNumber2 == 42 && 41 <= neutronCount2 <= 73) ||
(protonNumber2 == 43 && 42 <= neutronCount2 <= 75) ||
(protonNumber2 == 44 && 43 <= neutronCount2 <= 76) ||
(protonNumber2 == 45 && 44 <= neutronCount2 <= 77) ||
(protonNumber2 == 46 && 45 <= neutronCount2 <= 78) ||
(protonNumber2 == 47 && 46 <= neutronCount2 <= 83) ||
(protonNumber2 == 48 && 47 <= neutronCount2 <= 84) ||
(protonNumber2 == 49 && 48 <= neutronCount2 <= 86) ||
(protonNumber2 == 50 && 49 <= neutronCount2 <= 87) ||
(protonNumber2 == 51 && 52 <= neutronCount2 <= 88) ||
(protonNumber2 == 52 && 53 <= neutronCount2 <= 90) ||
(protonNumber2 == 53 && 55 <= neutronCount2 <= 91) ||
(protonNumber2 == 54 && 56 <= neutronCount2 <= 93) ||
(protonNumber2 == 55 && 57 <= neutronCount2 <= 96) ||
(protonNumber2 == 56 && 58 <= neutronCount2 <= 97) ||
(protonNumber2 == 57 && 60 <= neutronCount2 <= 98) ||
(protonNumber2 == 58 && 61 <= neutronCount2 <= 99) ||
(protonNumber2 == 59 && 62 <= neutronCount2 <= 100) ||
(protonNumber2 == 60 && 64 <= neutronCount2 <= 101) ||
(protonNumber2 == 61 && 65 <= neutronCount2 <= 102) ||
(protonNumber2 == 62 && 66 <= neutronCount2 <= 103) ||
(protonNumber2 == 63 && 67 <= neutronCount2 <= 104) ||
(protonNumber2 == 64 && 70 <= neutronCount2 <= 105) ||
(protonNumber2 == 65 && 71 <= neutronCount2 <= 106) ||
(protonNumber2 == 66 && 72 <= neutronCount2 <= 107) ||
(protonNumber2 == 67 && 73 <= neutronCount2 <= 108) ||
(protonNumber2 == 68 && 75 <= neutronCount2 <= 109) ||
(protonNumber2 == 69 && 76 <= neutronCount2 <= 110) ||
(protonNumber2 == 70 && 78 <= neutronCount2 <= 111) ||
(protonNumber2 == 71 && 79 <= neutronCount2 <= 111) ||
(protonNumber2 == 72 && 81 ≤ neutronCount2 ≤ 116) ||
(protonNumber2 == 73 && 82 ≤ neutronCount2 ≤ 117) ||
(protonNumber2 == 74 && 84 ≤ neutronCount2 ≤ 118) ||
(protonNumber2 == 75 && 85 ≤ neutronCount2 ≤ 119),


Switch[energyIndex, 1, paramList5 = Union[AppendTo[paramList0253eV,
    fermiEnergy0253eV[protonNumber1, protonNumber2]]], 2,
 paramList6 = Union[AppendTo[paramList500keV, fermiEnergy500keV[
     protonNumber1, protonNumber2]]], 3, paramList7 = Union[AppendTo[
    paramList14MeV, fermiEnergy14MeV[protonNumber1, protonNumber2]]]];
(*Calculate fission parameters*)effectiveDistVal =
```

```
              distanceFunc /. x → protonNumber1;
            coulombEnergy = (1.44 * protonNumber1 * protonNumber2) / effectiveDistVal;
            qValue = (getNuclearMass[atomicNumber, atomicNumber + neutronNumber] -
                getNuclearMass[protonNumber1, massNumber1] -
                getNuclearMass[protonNumber2, massNumber2] -
                promptNeutronCount * 1.008665) * 931.4940954;
            effectiveEnergy = coulombEnergy - qValue;
            (*Calculate fission probability with fermi correction*)
            probability = 1 / (1 + Exp[2 * Pi / (neutronSeparationEnergy + incidentEnergy) *
                    Sqrt[protonNumber1 * protonNumber2 / (protonNumber1 + protonNumber2)] /
                     reducedMass * (effectiveEnergy -
                       fermiEnergy[protonNumber1, protonNumber2])]);
            Sow[{protonNumber1, probability}, yieldList];]]]], {yieldList}, Rule][[
       2, All, 1]];
     (*Process results*)fragmentData = Part[yieldList /. tempData];
     processYields[data_] :=
      (Total@# / {Length@#, Total@data[[All, 2]] / 2} &) /@ GatherBy[data, First];
     AppendTo[results, {energyIndex, processYields[fragmentData]}];];
   results]

(*Execute calculation with Pattern selection*)
{startEnergyIndex, endEnergyIndex} =
  Switch[energyPattern, 1, {1, 1}, 2, {1, 2}, 3, {1, 3}, 4, {2, 2}, 5, {2, 3}];

(*Main calculation*)
results = CalculateFissionYieldsWithfermi[energyPattern];
{yieldData0253eVCalc, yieldData500keVCalc, yieldData14MeVCalc} =
  Switch[energyPattern, 1, {results[[1, 2]], Null, Null}, 2, {results[[1, 2]],
     results[[2, 2]], Null}, 3, {results[[1, 2]], results[[2, 2]], results[[3, 2]]}, 4,
    {Null, results[[1, 2]], Null}, 5, {Null, results[[1, 2]], results[[2, 2]]}];

(*11*)

(*Common Plot Generation Functions for Fission Yields*)
(*Define general yield plot function*)
CreateFissionYieldPlot[experimentalData_, calculatedData_, energyLabel_, plotColor_] :=
  ListLogPlot[{experimentalData, calculatedData}, (*Basic plot settings*)
    Joined → {True, True}, PlotRange → {{15, 80}, {10^(-12), 100}}, PlotMarkers →
     Automatic, (*Style settings*)PlotStyle → {Directive[PointSize[1 / 100], Red],
      Directive[PointSize[1 / 100], plotColor]}, (*Frame settings*)
    Frame → True, FrameLabel → {"Atomic Number", "Fission Yield (Independent)"},
    LabelStyle → Directive[Black, 19], FrameTicks → Automatic,
    FrameStyle → {Thick, Thick, Thick, Thick}, (*Legend settings*)
    PlotLegends → Placed[PointLegend[Automatic, {"JENDL-5", "Theoretical Curve"},
        Joined → {True, True, True}, Joined → {True, True},
        LabelStyle → 16, LegendFunction → "Frame", LegendLayout → "Column",
        LegendMarkers → Array[{Graphics@Disk[], 10} &, 3]], {{0.72, 0.25}, {1, 0.9}}],
    (*Layout settings*)AspectRatio → 0.8, ImageSize → 400, (*Title and isotope
     label*)Epilog → Inset[Style[isotopeName, Bold, 18], Scaled@{0.14, 0.94}]];

(*Define data analysis function*)
AnalyzeIsotopeYield[yieldData_, label_] :=
  Module[{maxZ1, maxZ2, maxYield1, maxYield2, peakAvgYield},
    (*Display analysis header*)Print[label];
```

```
(*Find primary peak*)
maxZ1 = Position[yieldData, Max[yieldData[[All, 2]]]][[1, 1]] + fitStartZ - 1;
maxZ2 = atomicNumber - maxZ1;
maxYield1 = Max[yieldData[[All, 2]]];
maxYield2 = yieldData[[maxZ2 - fitStartZ + 1]][[2]];
(*Display isotope information*)
For[atomicNum = 23, atomicNum ≤ 71, atomicNum++,
 If[maxZ1 == atomicNum, Print["Primary Fragment: ", ElementData[atomicNum, "Name"],
   "(Z=", atomicNum, ")", "; Yield: ", maxYield1, " MeV"]];
 If[maxZ2 == atomicNum, Print["Secondary Fragment: ", ElementData[atomicNum, "Name"],
   "(Z=", atomicNum, ")", "; Yield: ", maxYield2, " MeV"]];];
(*Calculate and display average peak yield*)
peakAvgYield = (maxYield1 + maxYield2) / 2;
Print["Average Peak Yield: ", peakAvgYield, " MeV"];
Print[
 "----------------------------------------------------------------------------
   ---"];];


(*12*)(*Optimization Program for Fermi Energy*)(*Display program description*)
Print[
  "Theoretical Analysis and Experimental Comparison of Fission Yield Distributions"];
Print[""];
Print["Calculation Process:"];
Print["1. Evaluate logarithmic differences between
    JENDL-5 experimental data and theoretical calculations"];
Print["2. Optimize Fermi Energy (Ex) using least squares method"];
Print["3. Generate theoretical curves using optimized Ex"];
Print["4. Compare and verify experimental vs theoretical values"];
Print[""];
Print["Optimization Goals:"];
Print[
  "・Theoretical reproduction of experimentally observed asymmetric fission yields"];
Print["・Understanding fission mechanisms at each incident neutron energy"];
Print["・Systematic determination of Fermi Energy (Ex)"];
Print[""];
Print["Evaluation Methods:"];
Print[
  "・Minimize sum of squared logarithmic differences between theory and experiment"];
Print["・Parameter optimization using FindMinimum function"];
Print["・Validation of theoretical curves with optimized parameters"];
Print[
  "-----------------------------------------------------------------------------
    -"];



(*Main optimization loop for each energy region*)
For[energyRegion = startEnergyIndex, energyRegion ≤ endEnergyIndex, energyRegion++,


  (*Initialize variables for each energy region*)
  If[energyRegion == 1,
```

```
 fermiParams0253eV = Union[paramList5] (*Thermal neutron parameters*)];
If[energyRegion == 2, fermiParams500keV = Union[paramList6]
  (*Intermediate energy parameters*)];
If[energyRegion == 3, fermiParams14MeV = Union[paramList7]
  (*Fast neutron parameters*)];


(*Prepare experimental data*)
If[energyRegion == 1,
 expData = yieldData0253eV[[fitStartIndex ;; fitEndIndex]] (*JENDL-5 0.0253eV*)];
If[energyRegion == 2, expData = yieldData500keV[[fitStartIndex ;; fitEndIndex]]
  (*JENDL-5 500keV*)];
If[energyRegion == 3, expData = yieldData14MeV[[fitStartIndex ;; fitEndIndex]]
  (*JENDL-5 14MeV*)];



(*Prepare theoretical calculation data*)
If[energyRegion == 1,
 yieldData0253eVTheory = yieldData0253eVCalc[[fitStartIndex ;; fitEndIndex]]];
If[energyRegion == 2, yieldData500keVTheory =
   yieldData500keVCalc[[fitStartIndex ;; fitEndIndex]]];
If[energyRegion == 3, yieldData14MeVTheory =
   yieldData14MeVCalc[[fitStartIndex ;; fitEndIndex]]];




(*Store optimized parameters*)
If[energyRegion == 1, fittedParams0253eV = optResult0253eVRe2[[2, All, All]]];
If[energyRegion == 2, fittedParams500keV = optResult500keVRe2[[2, All, All]]];
If[energyRegion == 3, fittedParams14MeV = optResult14MeVRe2[[2, All, All]]];
(*Display optimized parameters*)
If[energyRegion == 1, Print[fittedParams0253eV[[fitStartIndex ;; fitEndIndex]]]];
If[energyRegion == 2, Print[fittedParams500keV[[fitStartIndex ;; fitEndIndex]]]];
If[energyRegion == 3, Print[fittedParams14MeV[[fitStartIndex ;; fitEndIndex]]]];
(*Calculate theoretical yields with optimized parameters*)
If[energyRegion == 1, theoreticalYield0253eV =
   yieldData0253eVTheory /. optResult0253eVRe2[[2, All, All]]];
If[energyRegion == 2, theoreticalYield500keV =
   yieldData500keVTheory /. optResult500keVRe2[[2, All, All]]];
If[energyRegion == 3, theoreticalYield14MeV =
   yieldData14MeVTheory /. optResult14MeVRe2[[2, All, All]]];
(*Generate and display plots for each energy region*)
If[energyRegion == 1, Print["1. Analysis for Incident Neutron Energy: 0.0253eV"];
 Print[CreateFissionYieldPlot[expData,
    theoreticalYield0253eV, "0.0253eV", Blue]];];
If[energyRegion == 2, Print["2. Analysis for Incident Neutron Energy: 500keV"];
 Print[CreateFissionYieldPlot[expData, theoreticalYield500keV, "500keV", Blue]];];
If[energyRegion == 3, Print["3. Analysis for Incident Neutron Energy: 14MeV"];
 Print[CreateFissionYieldPlot[expData, theoreticalYield14MeV, "14MeV", Blue]];];
(*Display separator*)Print[
 "----------------------------------------------------------------------
    ---"];];
```

```
(*13*)(*Analysis and Visualization of Optimization Results*)
(*Display analysis title*)
Print["Quantitative Analysis Results of Fermi Energy Ex"];
Print["Based on Optimization Calculations using Experimental Fission Yield Data"];

(*Process and display results for each energy region*)
If[energyPattern == 1 || energyPattern == 2 || energyPattern == 3,
  Print["1. Analysis for Incident Neutron Energy: 0.0253eV"];
  Print[fermiData0253eV = Thread[{fittedParams0253eV[[All, 1, 1]],
       fittedParams0253eV[[All, 2]]}][[fitStartIndex ;; fitEndIndex]]];];

If[energyPattern == 2 || energyPattern == 3 || energyPattern == 4 || energyPattern == 5,
  Print["2. Analysis for Incident Neutron Energy: 500keV"];
  Print[fermiData500keV = Thread[{fittedParams500keV[[All, 1, 1]],
       fittedParams500keV[[All, 2]]}][[fitStartIndex ;; fitEndIndex]]];];

If[energyPattern == 3 || energyPattern == 5,
  Print["3. Analysis for Incident Neutron Energy: 14MeV"];
  Print[fermiData14MeV = Thread[{fittedParams14MeV[[All, 1, 1]],
       fittedParams14MeV[[All, 2]]}][[fitStartIndex ;; fitEndIndex]]];];

Print[
  "--------------------------------------------------------------------------------
    -"];


Print[" Systematic Analysis Results of Fermi Energy Ex"];
Print[" Using Optimized Effective Fission Distance Reff"];


(*Define common plot settings*)
plotBaseSettings = {PlotRange → {{15, 75}, {-8, 12}}, PlotMarkers → Automatic,
   Frame → True, FrameLabel → {"Atomic Number", "Energy (MeV)"},
   LabelStyle → Directive[Black, 19], FrameTicks → Automatic,
   FrameStyle → {Thick, Thick, Thick, Thick}, AspectRatio → 1.1, ImageSize → 400};


(*Create visualization based on energy pattern*)
Switch[energyPattern, 1, (*0.0253eV only*)plotData = {fermiData0253eV, {}, {}};
  Print[ListPlot[Select[plotData, Length[#] > 0 &],
    Joined → {True, True}, Evaluate@plotBaseSettings, PlotStyle →
     {Directive[PointSize[1/100], Blue], Directive[PointSize[1/100], Green],
      Directive[PointSize[1/100], Red]}, PlotLegends →
     Placed[PointLegend[Automatic, {"0.0253eV"}, Joined → {True, True, True},
       LabelStyle → 14, LegendFunction → "Frame", LegendLayout → "Column",
       LegendMarkers → Array[{Graphics@Disk[], 10} &, 3]], {{0.955, 0.95}, {1, 0.9}}],
    Epilog → {Inset[Style[isotopeName, 20, Bold], Scaled@{0.15, 0.95}],
      Inset[Style[databaseName, 20, Bold], Scaled@{0.15, 0.90}]}]], 2,
 (*0.0253eV and 500keV*)plotData = {fermiData0253eV, fermiData500keV, {}};
  Print[ListPlot[Select[plotData, Length[#] > 0 &],
    Joined → {True, True}, Evaluate@plotBaseSettings, PlotStyle →
     {Directive[PointSize[1/100], Blue], Directive[PointSize[1/100], Green],
      Directive[PointSize[1/100], Red]}, PlotLegends →
     Placed[PointLegend[Automatic, {"0.0253eV", "500keV"}, Joined → {True, True, True},
```

```
        LabelStyle → 14, LegendFunction → "Frame", LegendLayout → "Column",
        LegendMarkers → Array[{Graphics@Disk[], 10} &, 3]], {{0.955, 0.95}, {1, 0.9}}],
     Epilog → {Inset[Style[isotopeName, 20, Bold], Scaled@{0.15, 0.95}],
       Inset[Style[databaseName, 20, Bold], Scaled@{0.15, 0.90}]}]],
   3, (*All three energies*)plotData = {fermiData0253eV,
     fermiData500keV, fermiData14MeV};
   Print[ListPlot[Select[plotData, Length[#] > 0 &], Joined → {True, True},
     Evaluate@plotBaseSettings, PlotStyle → {Directive[PointSize[1/100], Blue],
       Directive[PointSize[1/100], Green], Directive[PointSize[1/100], Red]},
     PlotLegends → Placed[PointLegend[Automatic, {"0.0253eV", "500keV", "14MeV"},
        Joined → {True, True, True}, LabelStyle → 14,
        LegendFunction → "Frame", LegendLayout → "Column",
        LegendMarkers → Array[{Graphics@Disk[], 10} &, 3]], {{0.955, 0.95}, {1, 0.9}}],
     Epilog → {Inset[Style[isotopeName, 20, Bold], Scaled@{0.15, 0.95}],
       Inset[Style[databaseName, 20, Bold], Scaled@{0.15, 0.90}]}]],
   4, (*500keV only*)plotData = {fermiData500keV};
   Print[ListPlot[plotData, Joined → True, PlotRange → {{20, 75}, {-8, 12}},
     PlotMarkers → Automatic, PlotStyle → Directive[PointSize[1/100], Blue],
     Frame → True, LabelStyle → Directive[Black, 19],
     FrameTicks → Automatic, FrameStyle → {Thick, Thick, Thick, Thick},
     PlotLegends → Placed[PointLegend[{Blue}, {"500keV"}, Joined → True,
        LabelStyle → 14, LegendFunction → "Frame", LegendLayout → "Column",
        LegendMarkers → {Graphics@Disk[]}], {{0.955, 0.95}, {1, 0.9}}], AspectRatio →
      1.1, Epilog → {Inset[Style[isotopeName, 20, Bold], Scaled@{0.15, 0.95}],
       Inset[Style[databaseName, 20, Bold], Scaled@{0.15, 0.90}]}]], 5,
   (*500keV and 14MeV*)plotData = {{}, fermiData500keV, fermiData14MeV};
   Print[ListPlot[Select[plotData, Length[#] > 0 &], Joined → {True, True},
     PlotRange → {{20, 75}, {-8, 12}}, PlotMarkers → Automatic, PlotStyle →
      {Directive[PointSize[1/100], Green], Directive[PointSize[1/100], Red]},
     Frame → True, LabelStyle → Directive[Black, 19], FrameTicks → Automatic,
     FrameStyle → {Thick, Thick, Thick, Thick},
     PlotLegends → Placed[PointLegend[Automatic, {"500keV", "14MeV"},
        Joined → {True, True}, LabelStyle → 14, LegendFunction → "Frame",
        LegendLayout → "Column", LegendMarkers → Array[{Graphics@Disk[], 10} &, 3]],
       {{0.955, 0.95}, {1, 0.9}}], AspectRatio → 1.1,
     Epilog → {Inset[Style[isotopeName, 20, Bold], Scaled@{0.15, 0.95}],
       Inset[Style[databaseName, 20, Bold], Scaled@{0.15, 0.90}]}]]]];
 (*14*)(*Final Analysis and Visualization of Nuclear Species*)
 (*Fragment Analysis Function*)
 AnalyzeIsotopeYield[yieldData_, label_] := Module[{centerZ = Floor[atomicNumber/2],
     (*Calculate center atomic number*)rangeStart = Floor[atomicNumber/2] - 10,
     (*Lower bound*)rangeEnd = Floor[atomicNumber/2] + 10, (*Upper bound*)
     maxZ1, maxZ2, maxYield1, maxYield2, peakAvgYield, filteredData},
    (*Display analysis header*)Print[label];
    (*Filter data to only include atomic numbers within our range*)
    filteredData = Select[yieldData, rangeStart ≤ First[#] ≤ rangeEnd &];
    (*Find maximum yield within our range*)maxYield1 = Max[filteredData[[All, 2]]];
    maxZ1 = First[First[Select[filteredData, #[[2]] == maxYield1 &]]];
    maxZ2 = atomicNumber - maxZ1;
    maxYield2 = yieldData[[maxZ2 - First[yieldData][[1]] + 1, 2]];
    (*Display fragment information*)Print["Secondary Fragment: ",
     ElementData[maxZ1, "Name"], "(Z=", maxZ1, "); Yield: ", maxYield1, " MeV"];
    Print["Primary Fragment: ", ElementData[maxZ2, "Name"],
     "(Z=", maxZ2, "); Yield: ", maxYield2, " MeV"];
```

```
  (*Calculate and display average peak yield*)
  peakAvgYield = (maxYield1 + maxYield2) / 2;
  Print["Average Peak Yield: ", peakAvgYield, " MeV"];
  Print[
   "-------------------------------------------------------------------------
     ---"];];

(*Perform isotope analysis for each energy region*)
If[energyPattern == 1 || energyPattern == 2 || energyPattern == 3, AnalyzeIsotopeYield[
   fermiData0253eV, "1. Analysis for Incident Neutron Energy: 0.0253eV"]];

If[energyPattern == 2 || energyPattern == 3 || energyPattern == 4 || energyPattern == 5,
  AnalyzeIsotopeYield[fermiData500keV,
   "2. Analysis for Incident Neutron Energy: 500keV"]];

If[energyPattern == 3 || energyPattern == 5, AnalyzeIsotopeYield[
   fermiData14MeV, "3. Analysis for Incident Neutron Energy: 14MeV"];
  Print[
   "-------------------------------------------------------------------------
     ---"]];

(*Define logarithmic scale transformation function*)
LogScaleTransform[value_] := Rescale[Log10[value], {-8, 1}, {-8, 12}];

(*Common plot settings for combined visualization*)
combinedPlotSettings =
  {Joined → {True}, PlotRange → {{15, 75}, {-10, 12}}, PlotMarkers → Automatic,
   Frame → True, FrameLabel → {{Style["Energy (MeV)", 19, Black],
      Style["Fission Yield (Independent)", 19, Black]},
     {Style["Atomic Number", 19, Black], None}},
   LabelStyle → Directive[Black, 19], FrameTicks →
    {{Automatic, ({LogScaleTransform[10^#], If[# == 0, 1, Superscript[10, #]]} &) /@
       {-8, -6, -4, -2, 0, 2}}, {Automatic, None}},
   FrameStyle → {Thick, Thick, Thick, Thick}, AspectRatio → 1.1,
   ImageSize → 400, Axes → {True, False}};

(*Plot generation function*)
CreateEnergyPlot[fermiData_, yieldData_, energy_, color_] := Module[{scaledYieldData},
   scaledYieldData = ({#[[1]], LogScaleTransform@#[[2]]} &) /@ yieldData;
   ListPlot[{fermiData, scaledYieldData[[fitStartIndex ;; fitEndIndex]]},
    Evaluate[combinedPlotSettings],
    PlotStyle → {Directive[PointSize[1 / 100], color], Directive[PointSize[1 / 100],
       GrayLevel[0.6 - 0.2 * Position[{Blue, Green, Red}, color][[1, 1]]]]},
    PlotLegends → Placed[PointLegend[Automatic, {"Fermi Energy",
        "Charge Distribution (" <> databaseName <> ")"}, Joined → True,
       LabelStyle → 14, LegendFunction → "Frame", LegendLayout → "Column",
       LegendMarkers → Array[{Graphics@Disk[], 10} &, 2]], {{0.94, 0.19}, {1, 0.9}}],
    Epilog → {Inset[Style[ToString[isotopeName] <> " " <> energy, 18, Bold],
       Scaled@{0.4, 0.94}]}]];

(*Generate plots for each energy region*)
Module[{}, If[energyPattern == 1 || energyPattern == 2 || energyPattern == 3,
   Print[CreateEnergyPlot[fermiData0253eV, yieldData0253eV, "0.0253eV", Blue]]];
  If[energyPattern == 2 || energyPattern == 3 || energyPattern == 4 || energyPattern == 5,
```

```
   Print[CreateEnergyPlot[fermiData500keV, yieldData500keV, "500keV", Green]]];
  If[energyPattern == 3 || energyPattern == 5,
   Print[CreateEnergyPlot[fermiData14MeV, yieldData14MeV, "14MeV", Red]]];];

(*Final separator*)
Print[
  "----------------------------------------------------------------------------
    -"];
```

Select Nuclear Species:

1. U-235_Data_JENDL5_CalculationData

2. U-233_Data_JENDL5_CalculationData

3. U-238_Data_JENDL5_CalculationData

4. Th-232_Data_JENDL5_CalculationData

5. Np-237_Data_JENDL5_CalculationData

6. Pu-239_Data_JENDL5_CalculationData

7. Pu-240_Data_JENDL5_CalculationData

8. Pu-242_Data_JENDL5_CalculationData

9. Am-241_Data_JENDL5_CalculationData

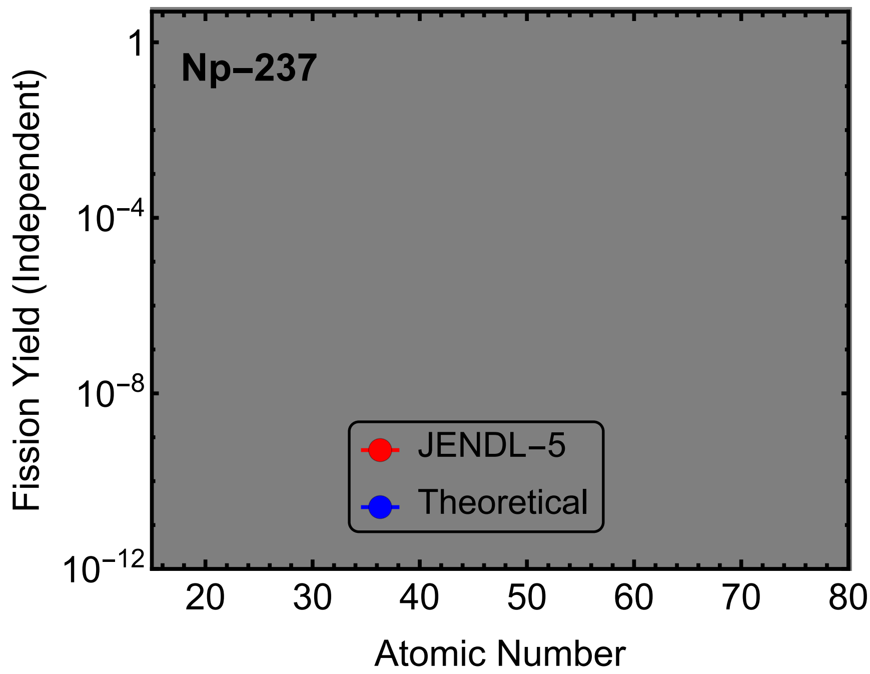
Attempting to load file from: C:\Users\HM202\OneDrive\デスクトップ\14用Fermi-mathematica3\Np-237_Data_JENDL5_CalculationData.m

Successfully loaded: Np-237_Data_JENDL5_CalculationData.m

1. Incident Neutron Energy: 0.0253eV: Effective Fission Distance Reff derived from experimental charge distribution

Analysis Results

Calculation results demonstrating that the effective fission distance Reff derived from optimization calculations accurately reproduces experimental values （confirming agreement between JENDL-5 experimental and theoretical values, and validating calculations using Mathematica ver11.2 FindMinimum）

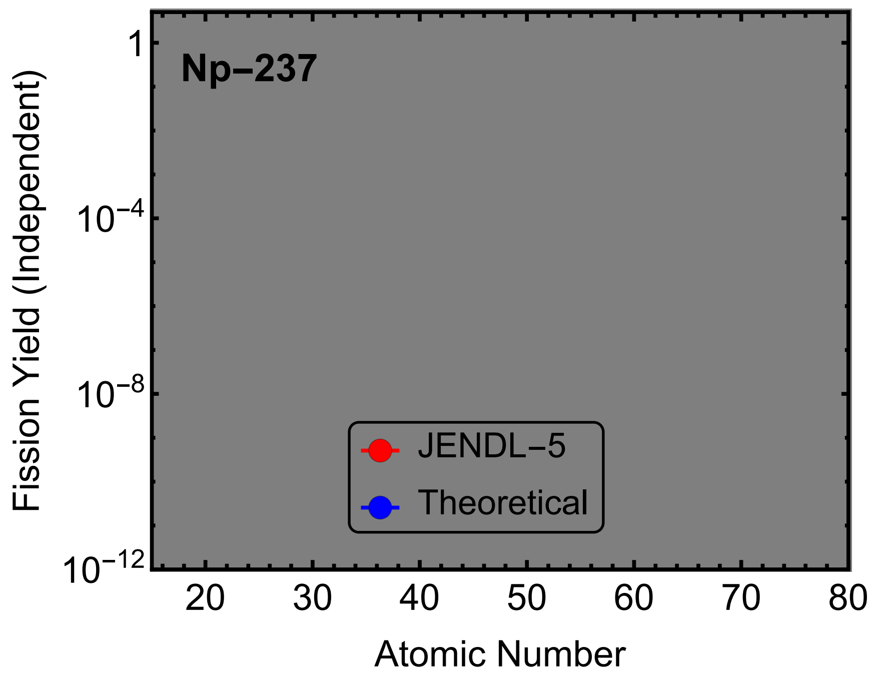


---

2. Incident Neutron Energy: 500keV: Effective Fission Distance Reff derived from experimental charge distribution

Analysis Results

Calculation results demonstrating that the effective fission distance Reff derived from optimization calculations accurately reproduces experimental values (confirming agreement between JENDL-5 experimental and theoretical values, and validating calculations using Mathematica ver11.2 FindMinimum)

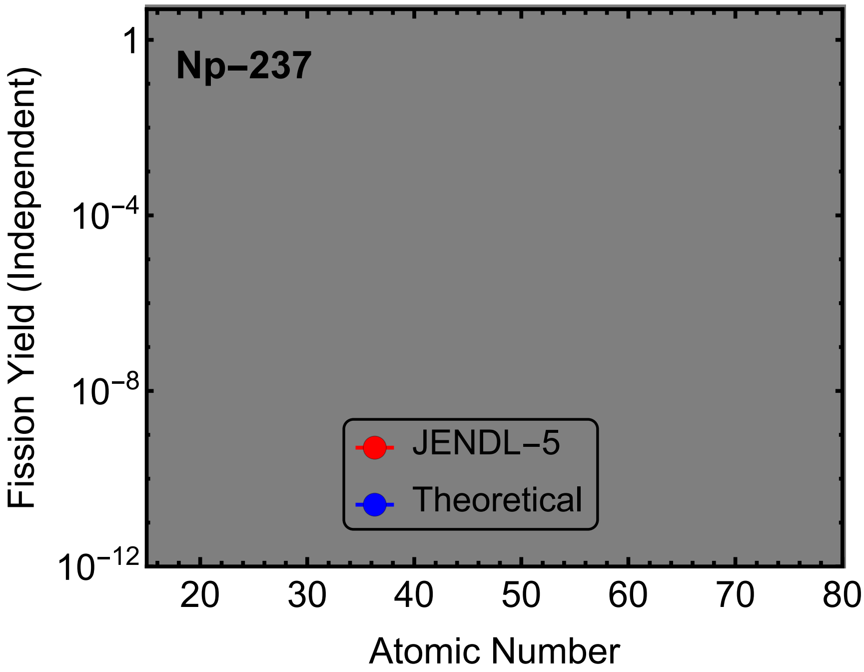


---

3. Incident Neutron Energy: 14MeV: Effective Fission Distance Reff derived from experimental charge distribution

Analysis Results

Calculation results demonstrating that the effective fission distance Reff derived from optimization calculations accurately reproduces experimental values (confirming agreement between JENDL-5 experimental and theoretical values, and validating calculations using Mathematica ver11.2 FindMinimum)

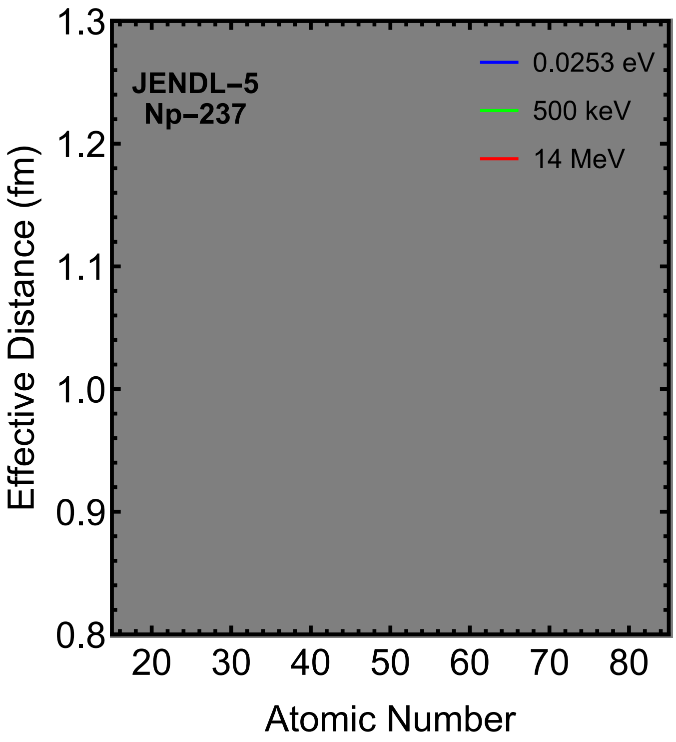


Analysis Results:

1. The effective fission distance Reff shows quadratic dependence on fragment charge

2. This dependence reflects fundamental laws of charge distribution in fission process

3. Similar dependence is maintained across different incident energies



Incident Neutron Energy: 0.0253 eV case

$$\text{Reff} == -0.364831 + 0.000729022\,(92.995 - x)\,x$$

Incident Neutron Energy: 500 keV case

$$\text{Reff} == -0.38408 + 0.00073711\,(92.9986 - x)\,x$$

Incident Neutron Energy: 14 MeV case

$$\text{Reff} == -0.532936 + 0.000817476\,(92.9977 - x)\,x$$

Analysis Results:

1. The effective fission distance Reff shows quadratic dependence on fragment charge number
2. This dependence reflects fundamental laws of charge distribution in the fission process
3. Similar dependence is maintained across different incident energies

Table of Effective Fission Distance (Reff) Values by Atomic Number [Unit: fm]:

Displaying calculated values (pre-fitting) and post-fitting values for each energy

Fitting calculation used fragment values from atomic number 28 to 65

Z1	Z2	Reff [fm] (0.0253 eV) Calculated	Fitted Value	Reff [fm] (500 keV) Calculated	Fitted Value	Reff [fm] (14 MeV) Calculated	Fitted Value
23	70	0.188359	-	0.211074	-	0.133311	-

Out[73]=

24	69	0.560658	-	0.572471	-	0.413075	-
25	68	0.761650	-	0.765860	-	0.720066	-
26	67	0.825969	-	0.826889	-	0.867300	-
27	66	0.886240	-	0.884482	-	0.914719	-
28	65	0.961088	0.96189	0.956787	0.95743	0.955635	0.95482
29	64	0.987766	0.98813	0.983650	0.98397	0.984555	0.98425
30	63	1.012590	1.01291	1.008760	1.00903	1.011670	1.01204
31	62	1.036270	1.03624	1.032630	1.03261	1.037680	1.03819
32	61	1.058010	1.05810	1.054680	1.05472	1.061880	1.06272
33	60	1.078810	1.07851	1.075600	1.07536	1.085090	1.08560
34	59	1.098140	1.09746	1.095060	1.09453	1.106610	1.10686
35	58	1.116150	1.11496	1.113180	1.11222	1.126710	1.12647
36	57	1.131990	1.13099	1.129300	1.12843	1.144870	1.14446
37	56	1.146600	1.14557	1.144100	1.14317	1.161610	1.16081
38	55	1.159150	1.15869	1.156840	1.15644	1.176040	1.17552
39	54	1.170480	1.17035	1.168310	1.16823	1.188870	1.18860
40	53	1.179960	1.18055	1.177990	1.17855	1.199860	1.20004
41	52	1.188720	1.18929	1.186830	1.18739	1.209650	1.20985
42	51	1.195780	1.19658	1.193980	1.19476	1.217550	1.21802
43	50	1.201900	1.20241	1.200060	1.20066	1.223960	1.22456
44	49	1.206460	1.20678	1.204730	1.20508	1.229080	1.22946
45	48	1.209380	1.20969	1.207800	1.20803	1.232850	1.23273
46	47	1.211170	1.21115	1.209780	1.20950	1.234940	1.23436
47	46	1.211110	1.21114	1.209760	1.20950	1.235030	1.23436
48	45	1.209120	1.20968	1.207820	1.20803	1.232910	1.23272
49	44	1.206260	1.20676	1.204710	1.20508	1.229180	1.22945
50	43	1.201660	1.20238	1.200080	1.20065	1.224040	1.22454
51	42	1.195780	1.19655	1.193960	1.19476	1.217530	1.21800
52	41	1.188770	1.18925	1.186840	1.18738	1.209730	1.20983
53	40	1.180010	1.18050	1.177970	1.17854	1.199910	1.20001
54	39	1.170480	1.17029	1.168340	1.16822	1.188850	1.18857
55	38	1.159190	1.15862	1.156830	1.15642	1.175990	1.17549
56	37	1.146600	1.14550	1.144110	1.14315	1.161590	1.16077
57	36	1.131960	1.13091	1.129280	1.12841	1.144860	1.14442
58	35	1.116050	1.11487	1.113200	1.11219	1.126680	1.12643
59	34	1.098050	1.09737	1.095020	1.09450	1.106570	1.10681
60	33	1.078710	1.07841	1.075580	1.07533	1.085050	1.08555
61	32	1.057970	1.05800	1.054650	1.05469	1.061840	1.06266

62	31	1.036260	1.03612	1.032710	1.03258	1.037560	1.03814
63	30	1.012360	1.01279	1.008690	1.00899	1.011620	1.01198
64	29	0.987534	0.98800	0.983619	0.98393	0.984437	0.98418
65	28	0.960894	0.96175	0.956610	0.95739	0.955512	0.95475
66	27	0.886240	-	0.884482	-	0.914719	-
67	26	0.825969	-	0.826889	-	0.867300	-
68	25	0.761650	-	0.765860	-	0.720066	-
69	24	0.560658	-	0.572471	-	0.413075	-

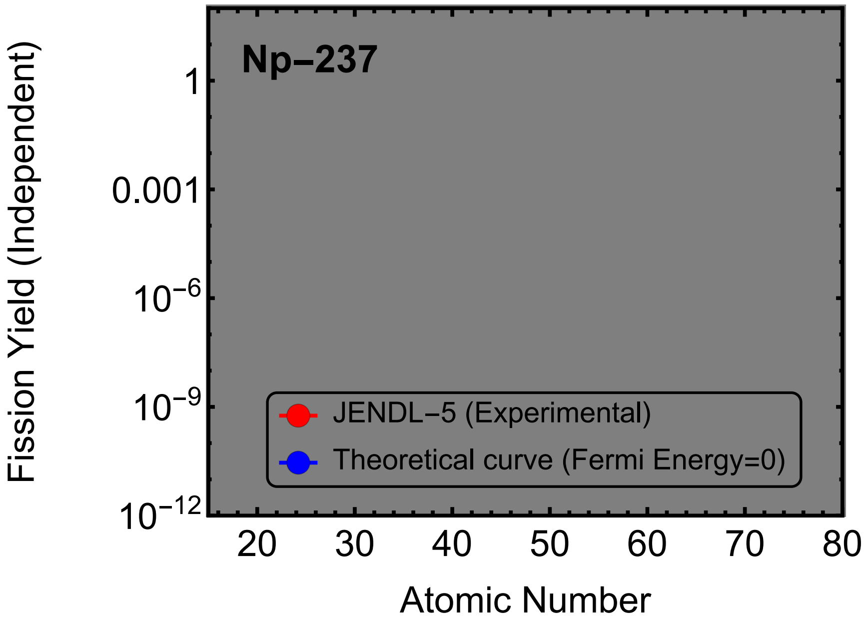
Analysis of Fission Yields: Comparison
between Experimental Data and Theoretical Calculations (Ex=0)

Theoretical Analysis Results using Effective
Fission Distance Reff proportional to fragment charge product,

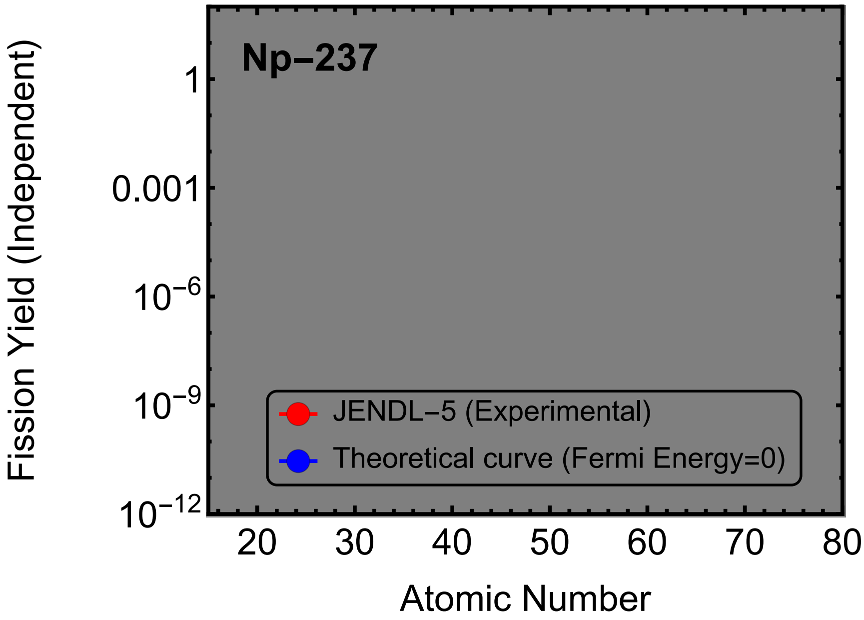
with zero Fermi Energy (Ex=0)

- Quantitative reproduction of experimentally observed asymmetric fission yield distributions

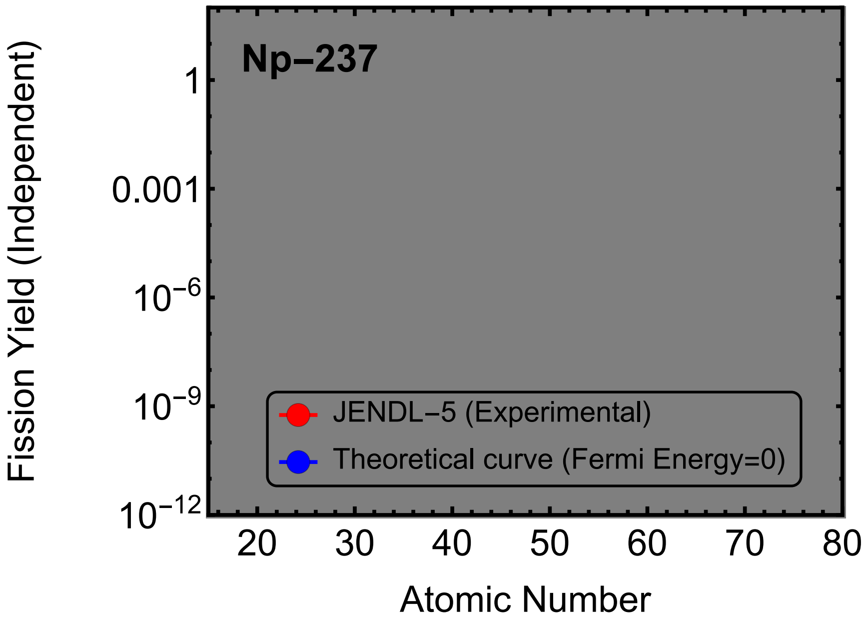
1. Analysis for Incident Neutron Energy: 0.0253eV



2. Analysis for Incident Neutron Energy: 500keV



3. Analysis for Incident Neutron Energy: 14MeV



Analysis of Effective Fission Distance and Fission Probability

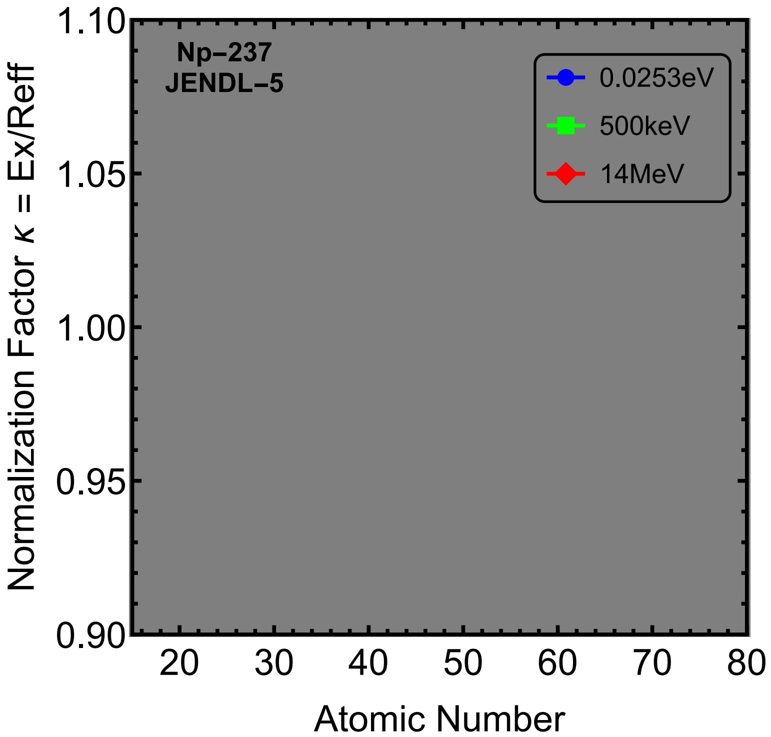
Analysis Contents:

1. Calculation of Effective Fission Distance (Reff) for each incident neutron energy
2. Evaluation of fission probability using η function
3. Derivation of normalization factor κ (= Ex/Reff)

The vertical axis κ represents the ratio of Fermi Energy(Ex)

to effective fission distance (Reff).

$\kappa \approx 1$ suggests the fission Fermi Energy is proportional to effective distance.



Theoretical Analysis and Experimental Comparison of Fission Yield Distributions

Calculation Process:

1. Evaluate logarithmic differences
   between JENDL–5 experimental data and theoretical calculations

2. Optimize Fermi Energy (Ex) using least squares method

3. Generate theoretical curves using optimized Ex

4. Compare and verify experimental vs theoretical values

Optimization Goals:

· Theoretical reproduction of experimentally observed asymmetric fission yields

· Understanding fission mechanisms at each incident neutron energy

· Systematic determination of Fermi Energy (Ex)

Evaluation Methods:

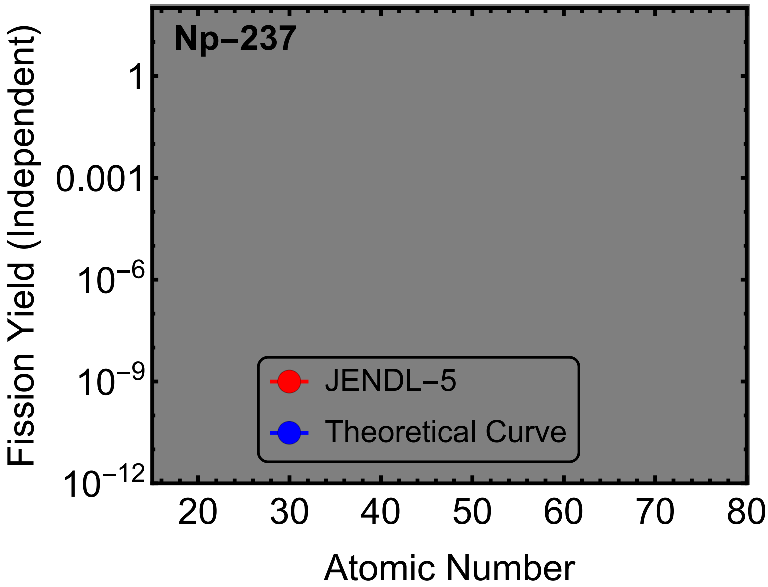
· Minimize sum of squared logarithmic differences between theory and experiment

· Parameter optimization using FindMinimum function

· Validation of theoretical curves with optimized parameters

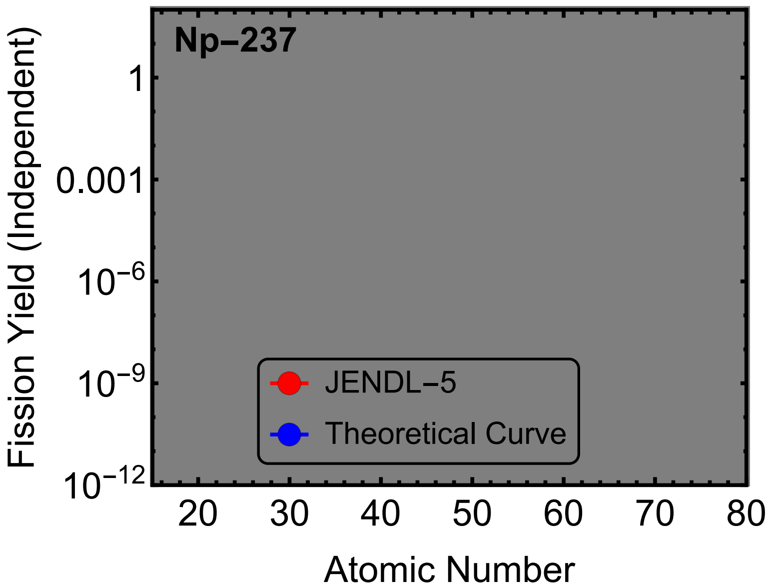
```
{fermiEnergy0253eV[28, 65] → -1.19553, fermiEnergy0253eV[29, 64] → 0.0669928,
 fermiEnergy0253eV[30, 63] → 0.197393, fermiEnergy0253eV[31, 62] → 1.11947,
 fermiEnergy0253eV[32, 61] → 0.796036, fermiEnergy0253eV[33, 60] → 1.75392,
 fermiEnergy0253eV[34, 59] → 2.63899, fermiEnergy0253eV[35, 58] → 3.80619,
 fermiEnergy0253eV[36, 57] → 3.3049, fermiEnergy0253eV[37, 56] → 3.34545,
 fermiEnergy0253eV[38, 55] → 2.0458, fermiEnergy0253eV[39, 54] → 1.28256,
 fermiEnergy0253eV[40, 53] → -0.304359, fermiEnergy0253eV[41, 52] → -0.261818,
 fermiEnergy0253eV[42, 51] → -0.744209, fermiEnergy0253eV[43, 50] → -0.100097,
 fermiEnergy0253eV[44, 49] → 0.308816, fermiEnergy0253eV[45, 48] → 0.31689,
 fermiEnergy0253eV[46, 47] → 1.02588, fermiEnergy0253eV[47, 46] → 0.908761,
 fermiEnergy0253eV[48, 45] → -0.21088, fermiEnergy0253eV[49, 44] → -0.0797438,
 fermiEnergy0253eV[50, 43] → -0.576353, fermiEnergy0253eV[51, 42] → -0.665365,
 fermiEnergy0253eV[52, 41] → -0.0537801, fermiEnergy0253eV[53, 40] → -0.0782913,
 fermiEnergy0253eV[54, 39] → 1.41121, fermiEnergy0253eV[55, 38] → 2.26499,
 fermiEnergy0253eV[56, 37] → 3.5096, fermiEnergy0253eV[57, 36] → 3.41674,
 fermiEnergy0253eV[58, 35] → 3.771, fermiEnergy0253eV[59, 34] → 2.63548,
 fermiEnergy0253eV[60, 33] → 1.7513, fermiEnergy0253eV[61, 32] → 0.976056,
 fermiEnergy0253eV[62, 31] → 1.39883, fermiEnergy0253eV[63, 30] → -0.0953588,
 fermiEnergy0253eV[64, 29] → -0.215831, fermiEnergy0253eV[65, 28] → -1.36152}
```

1. Analysis for Incident Neutron Energy: 0.0253eV



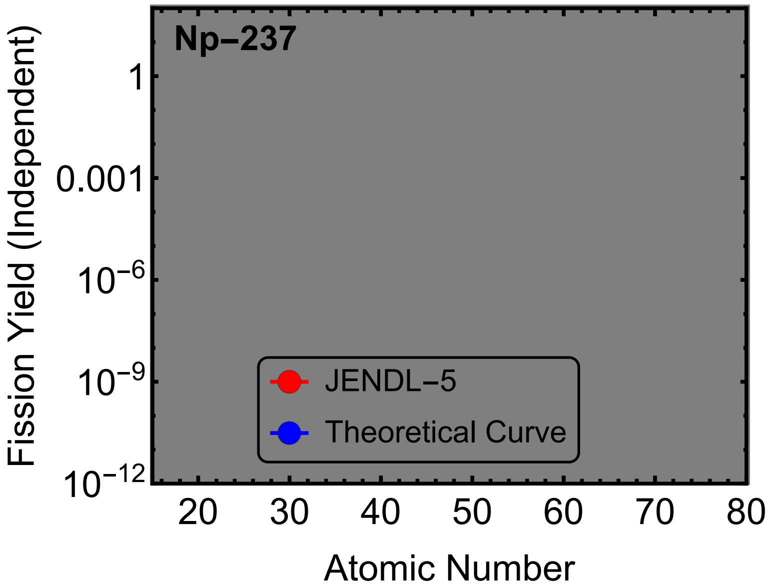
```
{fermiEnergy500keV[28, 65] → -0.771394, fermiEnergy500keV[29, 64] → 0.187696,
 fermiEnergy500keV[30, 63] → 0.35045, fermiEnergy500keV[31, 62] → 1.09222,
 fermiEnergy500keV[32, 61] → 0.915501, fermiEnergy500keV[33, 60] → 1.61952,
 fermiEnergy500keV[34, 59] → 2.29894, fermiEnergy500keV[35, 58] → 3.27993,
 fermiEnergy500keV[36, 57] → 3.01711, fermiEnergy500keV[37, 56] → 3.11982,
 fermiEnergy500keV[38, 55] → 1.90644, fermiEnergy500keV[39, 54] → 1.16747,
 fermiEnergy500keV[40, 53] → -0.247582, fermiEnergy500keV[41, 52] → -0.236192,
 fermiEnergy500keV[42, 51] → -0.715081, fermiEnergy500keV[43, 50] → -0.304627,
 fermiEnergy500keV[44, 49] → 0.239014, fermiEnergy500keV[45, 48] → 0.484038,
 fermiEnergy500keV[46, 47] → 1.56189, fermiEnergy500keV[47, 46] → 1.52418,
 fermiEnergy500keV[48, 45] → 0.543481, fermiEnergy500keV[49, 44] → 0.188687,
 fermiEnergy500keV[50, 43] → -0.241116, fermiEnergy500keV[51, 42] → -0.737429,
 fermiEnergy500keV[52, 41] → -0.19058, fermiEnergy500keV[53, 40] → -0.247819,
 fermiEnergy500keV[54, 39] → 1.25911, fermiEnergy500keV[55, 38] → 1.91687,
 fermiEnergy500keV[56, 37] → 3.19166, fermiEnergy500keV[57, 36] → 3.02764,
 fermiEnergy500keV[58, 35] → 3.39547, fermiEnergy500keV[59, 34] → 2.26974,
 fermiEnergy500keV[60, 33] → 1.6402, fermiEnergy500keV[61, 32] → 0.925688,
 fermiEnergy500keV[62, 31] → 1.37456, fermiEnergy500keV[63, 30] → 0.243142,
 fermiEnergy500keV[64, 29] → 0.204629, fermiEnergy500keV[65, 28] → -1.16709}
```

2. Analysis for Incident Neutron Energy: 500keV



{fermiEnergy14MeV[28, 65] → 2.88311, fermiEnergy14MeV[29, 64] → 1.38421,
fermiEnergy14MeV[30, 63] → -0.449056, fermiEnergy14MeV[31, 62] → -0.798546,
fermiEnergy14MeV[32, 61] → -1.57541, fermiEnergy14MeV[33, 60] → -0.72529,
fermiEnergy14MeV[34, 59] → -0.0646372, fermiEnergy14MeV[35, 58] → 1.05095,
fermiEnergy14MeV[36, 57] → 1.43065, fermiEnergy14MeV[37, 56] → 2.28115,
fermiEnergy14MeV[38, 55] → 1.63331, fermiEnergy14MeV[39, 54] → 1.08773,
fermiEnergy14MeV[40, 53] → 0.115082, fermiEnergy14MeV[41, 52] → 0.0875894,
fermiEnergy14MeV[42, 51] → -0.473493, fermiEnergy14MeV[43, 50] → -0.731202,
fermiEnergy14MeV[44, 49] → -0.295889, fermiEnergy14MeV[45, 48] → 0.735599,
fermiEnergy14MeV[46, 47] → 1.66612, fermiEnergy14MeV[47, 46] → 1.85491,
fermiEnergy14MeV[48, 45] → 0.873659, fermiEnergy14MeV[49, 44] → -0.064459,
fermiEnergy14MeV[50, 43] → -0.540943, fermiEnergy14MeV[51, 42] → -0.49514,
fermiEnergy14MeV[52, 41] → 0.285351, fermiEnergy14MeV[53, 40] → 0.266209,
fermiEnergy14MeV[54, 39] → 1.10894, fermiEnergy14MeV[55, 38] → 1.59611,
fermiEnergy14MeV[56, 37] → 2.30749, fermiEnergy14MeV[57, 36] → 1.51076,
fermiEnergy14MeV[58, 35] → 1.07628, fermiEnergy14MeV[59, 34] → -0.0536667,
fermiEnergy14MeV[60, 33] → -0.707708, fermiEnergy14MeV[61, 32] → -1.53488,
fermiEnergy14MeV[62, 31] → -0.952747, fermiEnergy14MeV[63, 30] → -0.414195,
fermiEnergy14MeV[64, 29] → 1.24096, fermiEnergy14MeV[65, 28] → 2.72846}

3. Analysis for Incident Neutron Energy: 14MeV



Quantitative Analysis Results of Fermi Energy Ex

Based on Optimization Calculations using Experimental Fission Yield Data

1. Analysis for Incident Neutron Energy: 0.0253eV

{{28, -1.19553}, {29, 0.0669928}, {30, 0.197393}, {31, 1.11947}, {32, 0.796036}, {33, 1.75392},
{34, 2.63899}, {35, 3.80619}, {36, 3.3049}, {37, 3.34545}, {38, 2.0458}, {39, 1.28256},
{40, -0.304359}, {41, -0.261818}, {42, -0.744209}, {43, -0.100097}, {44, 0.308816},
{45, 0.31689}, {46, 1.02588}, {47, 0.908761}, {48, -0.21088}, {49, -0.0797438},
{50, -0.576353}, {51, -0.665365}, {52, -0.0537801}, {53, -0.0782913}, {54, 1.41121},
{55, 2.26499}, {56, 3.5096}, {57, 3.41674}, {58, 3.771}, {59, 2.63548}, {60, 1.7513},
{61, 0.976056}, {62, 1.39883}, {63, -0.0953588}, {64, -0.215831}, {65, -1.36152}}

2. Analysis for Incident Neutron Energy: 500keV

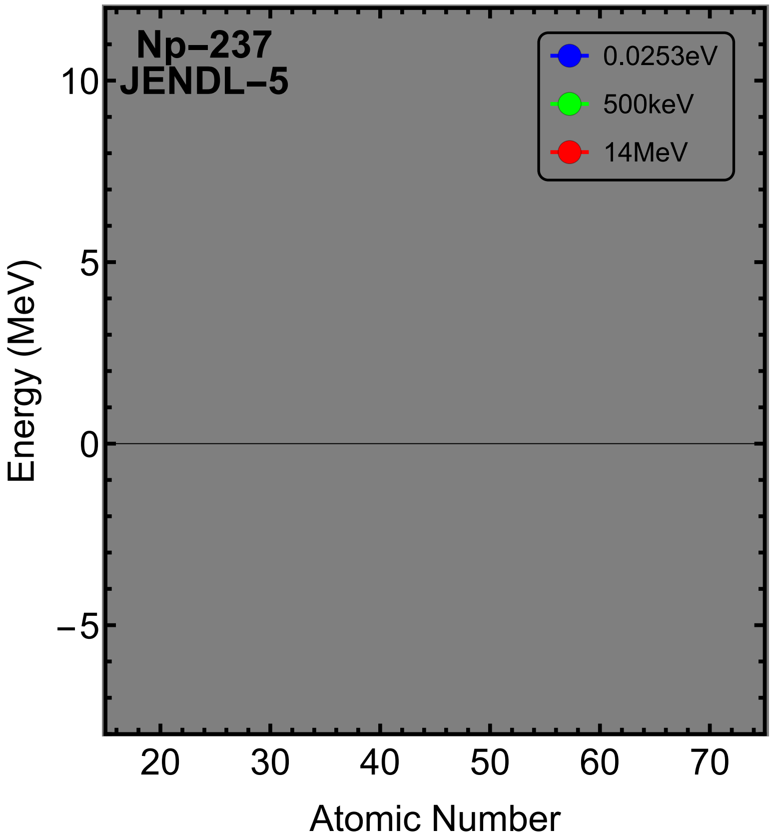
{{28, -0.771394}, {29, 0.187696}, {30, 0.35045}, {31, 1.09222}, {32, 0.915501}, {33, 1.61952},
{34, 2.29894}, {35, 3.27993}, {36, 3.01711}, {37, 3.11982}, {38, 1.90644}, {39, 1.16747},
{40, -0.247582}, {41, -0.236192}, {42, -0.715081}, {43, -0.304627}, {44, 0.239014},
{45, 0.484038}, {46, 1.56189}, {47, 1.52418}, {48, 0.543481}, {49, 0.188687},
{50, -0.241116}, {51, -0.737429}, {52, -0.19058}, {53, -0.247819}, {54, 1.25911},
{55, 1.91687}, {56, 3.19166}, {57, 3.02764}, {58, 3.39547}, {59, 2.26974}, {60, 1.6402},
{61, 0.925688}, {62, 1.37456}, {63, 0.243142}, {64, 0.204629}, {65, -1.16709}}

3. Analysis for Incident Neutron Energy: 14MeV

{{28, 2.88311}, {29, 1.38421}, {30, -0.449056}, {31, -0.798546}, {32, -1.57541}, {33, -0.72529},
{34, -0.0646372}, {35, 1.05095}, {36, 1.43065}, {37, 2.28115}, {38, 1.63331}, {39, 1.08773},
{40, 0.115082}, {41, 0.0875894}, {42, -0.473493}, {43, -0.731202}, {44, -0.295889},
{45, 0.735599}, {46, 1.66612}, {47, 1.85491}, {48, 0.873659}, {49, -0.064459},
{50, -0.540943}, {51, -0.49514}, {52, 0.285351}, {53, 0.266209}, {54, 1.10894}, {55, 1.59611},
{56, 2.30749}, {57, 1.51076}, {58, 1.07628}, {59, -0.0536667}, {60, -0.707708},
{61, -1.53488}, {62, -0.952747}, {63, -0.414195}, {64, 1.24096}, {65, 2.72846}}

Systematic Analysis Results of Fermi Energy Ex

Using Optimized Effective Fission Distance Reff



1. Analysis for Incident Neutron Energy: 0.0253eV

Secondary Fragment: barium(Z=56); Yield: 3.5096 MeV

Primary Fragment: rubidium(Z=37); Yield: 3.34545 MeV

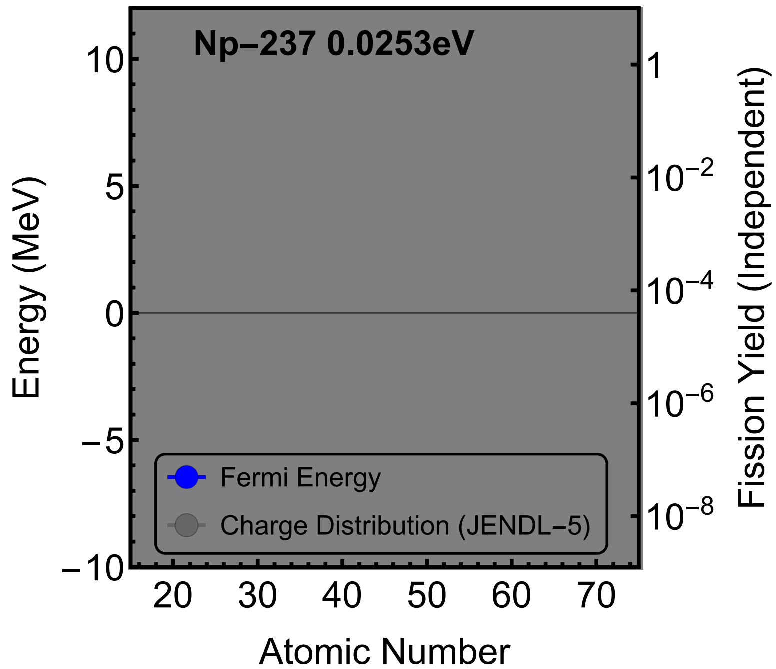
Average Peak Yield: 3.42752 MeV

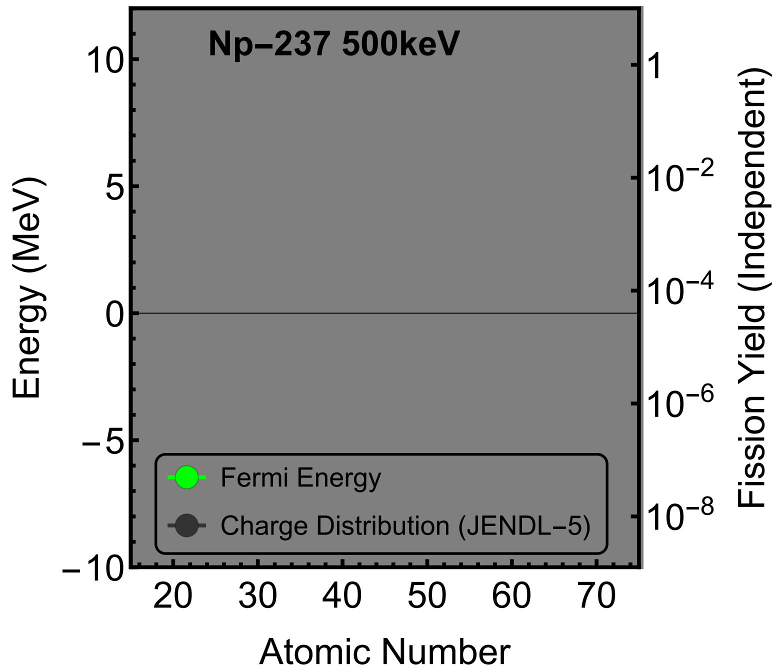
----

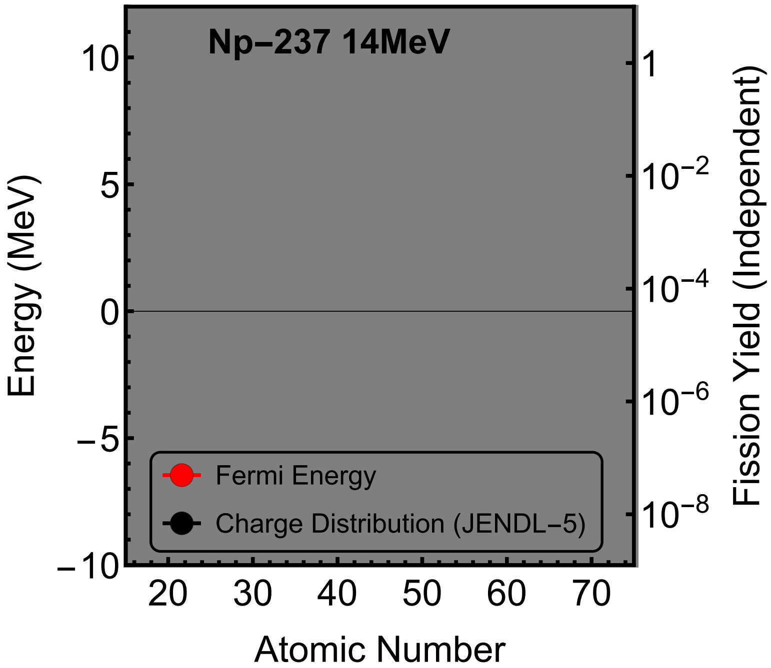
2. Analysis for Incident Neutron Energy: 500keV

Secondary Fragment: barium(Z=56); Yield: 3.19166 MeV

```
Primary Fragment: rubidium(Z=37); Yield: 3.11982 MeV

Average Peak Yield: 3.15574 MeV

--------------------------------------------------------------------------------

3. Analysis for Incident Neutron Energy: 14MeV

Secondary Fragment: barium(Z=56); Yield: 2.30749 MeV

Primary Fragment: rubidium(Z=37); Yield: 2.28115 MeV

Average Peak Yield: 2.29432 MeV

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
```






Np−237 14MeV
Energy (MeV)
10
5
0
−5
−10
Fission Yield (Independent)
1
10^{-2}
10^{-4}
10^{-6}
10^{-8}
Fermi Energy
Charge Distribution (JENDL−5)
20
30
40
50
60
70
Atomic Number