Constants

Fundamental constants:

```
In[ ]:= amu = 0.9315 GeV;
In[ ]:= mn = 1.008 amu;
In[ ]:= ℏc = .197327 GeV fm;
```

In[ ]:= $\hbar = \dfrac{4.135667 \times 10^{-15}\ \text{eV s}}{2\,\pi};$

```
In[ ]:= Me = 511 keV;
```

In[ ]:= $\text{c} = 2.998 \times 10^{5}\ \text{km} / \text{s};$

In[ ]:= $\text{GeVperKg} = \text{c}^2 \left(\dfrac{1000\ \text{m}}{\text{km}}\right)^2 \dfrac{1\ \text{J}}{1\ \text{kg}\ \text{m}^2 / \text{s}^2}\ \dfrac{1\ \text{eV}}{1.602 \times 10^{-19}\ \text{J}}\ \dfrac{1\ \text{GeV}}{10^9\ \text{eV}};$

DM astrophysical parameters:

In[ ]:= $\rho = 0.3\ \text{GeV} / \text{cm}^3;$

```
In[ ]:= v0 = 220 km / s;
In[ ]:= vesc = 544 km / s;
In[ ]:= ve = 240 km / s;
```

In[ ]:= $\text{VMAX} = \dfrac{\text{vesc} + \text{ve}}{\text{c}};$

Reduced mass def:

In[ ]:= $\mu[\text{x\_}, \text{y\_}] := \dfrac{\text{x y}}{\text{x} + \text{y}}$

Set plot styles and custom plot functions

Inelastic Kinematics

Energy conservation in the CoM frame gives:

$\frac{1}{2}\,\mu_i\, v_i^{\,2} = E_1{}' + E_2{}' + E_{\text{EM}} + \delta_{\text{DM}}$

$\frac{1}{2}\,\mu_i\, v_i^{\,2} = \frac{p_F{}^2}{2\,\mu_F} + E_{\text{EM}} + \delta_{\text{DM}}$

$p_F{}^2 = \mu_i\,\mu_F\, v_i^{\,2} - 2\,\mu_F\,(E_{\text{EM}} + \delta_{\text{DM}})$

$q^2 = (p_F - p_i)^2$

Momentum transfer is same in both frames :

$$E_R \simeq \frac{p_F{}^2 + p_i{}^2 - 2\, p_F\, p_i \cos\theta_{CM}}{2\, m_T}$$

In[◦]:=
```
Er ==
  1/(2 mT) (μT μTf vi^2 - 2 μTf (δEM + δDM) + μT^2 vi^2 - 2 Sqrt[μT μTf vi^2 - 2 μTf (δEM + δDM)] μT vi cosθCM) /.
   {μT -> (mx mT)/(mx + mT), μTf -> ((mx + δDM) mT)/((mx + δDM) + mT)} // Simplify
```

Define limits of recoil energy ($\cos\theta$CM = ±1):

In[◦]:=
```
ErMin[vi_?NumericQ, δDM_?NumericQ, δEM_?NumericQ, mx_?NumericQ, mT_?NumericQ] :=
  1/(2 (mT + mx)^2 (mT + mx + δDM))
  (mT mx (mT + mx) vi^2 (mx + δDM) + mT mx^2 vi^2 (mT + mx + δDM) - 2 (mT + mx)^2 (mx + δDM) (δDM + δEM) - 2 mx
     (mT + mx) vi (mT + mx + δDM) Sqrt[(mT (mx + δDM) (mT mx vi^2 - 2 mT (δDM + δEM) - 2 mx (δDM + δEM)))/((mT + mx) (mT + mx + δDM))]);
ErMax[vi_?NumericQ, δDM_?NumericQ, δEM_?NumericQ, mx_?NumericQ, mT_?NumericQ] :=
  1/(2 (mT + mx)^2 (mT + mx + δDM))
  (mT mx (mT + mx) vi^2 (mx + δDM) + mT mx^2 vi^2 (mT + mx + δDM) - 2 (mT + mx)^2 (mx + δDM) (δDM + δEM) + 2 mx
     (mT + mx) vi (mT + mx + δDM) Sqrt[(mT (mx + δDM) (mT mx vi^2 - 2 mT (δDM + δEM) - 2 mx (δDM + δEM)))/((mT + mx) (mT + mx + δDM))]);
```

Find $v_{\min}$

Solving the simple case of $\delta_{DM} = 0$ gives the familiar result :

In[◦]:=
```
Solve[
 {Er == 1/(2 mT) (μi μf vi^2 - 2 μf (δEM + δDM) + μi^2 vi^2 - 2 Sqrt[μi μf vi^2 - 2 μf (δEM + δDM)] μi vi cosθCM) /.
    {cosθCM -> -1, μf -> μi, δDM -> 0}}, vi]
```

Out[◦]=
$$\left\{\left\{vi \to -\frac{Er\, mT + \delta EM\, \mu i}{\sqrt{2}\,\sqrt{Er}\,\sqrt{mT}\,\mu i}\right\}, \left\{vi \to \frac{Er\, mT + \delta EM\, \mu i}{\sqrt{2}\,\sqrt{Er}\,\sqrt{mT}\,\mu i}\right\}\right\}$$

In[ ]:=
$$\text{VminApprox}[\text{ER\_},\delta\text{DM\_},\delta\text{EM\_},\text{mx\_},\text{mT\_}]=\text{Abs}\left[\frac{\text{ER mT}+(\delta\text{DM}+\delta\text{EM})\,\mu[\text{mx},\text{mT}]}{\sqrt{2}\,\sqrt{\text{ER}}\,\sqrt{\text{mT}}\,\mu[\text{mx},\text{mT}]}\right];$$

In[ ]:=
$$\text{Solve}\Big[\text{ER}==\frac{1}{2\,\text{mT}}\left(\mu i\,\mu f\,vi^2-2\,\mu f\,(\delta\text{EM}+\delta\text{DM})+\mu i^2\,vi^2-2\,\sqrt{\mu i\,\mu f\,vi^2-2\,\mu f\,(\delta\text{EM}+\delta\text{DM})}\,\mu i\,vi\,\cos\theta\text{CM}\right)\text{/.}\left\{\cos\theta\text{CM}\to-1,\ \mu i\to\frac{\text{mT mx}}{\text{mT}+\text{mx}},\ \mu f\text{->}\frac{\text{mT}\,(\text{mx}+\delta\text{DM})}{\text{mT}+(\text{mx}+\delta\text{DM})}\right\},\ vi\Big][[2]]\ \text{// Simplify}$$

In[ ]:=
$$\text{VminBad}[\text{ER\_},\delta\text{DM\_},\delta\text{EM\_},\text{mx\_},\text{mT\_}]=\sqrt{2}\,\sqrt{\Bigg(\frac{1}{\text{mT}^4\,\text{mx}^2\,\delta\text{DM}^2}\Big(\text{ER mT mx}\,(\text{mT}+\text{mx})^2\,\big(2\,\text{mx}\,(\text{mx}+\delta\text{DM})^2+\text{mT}^2\,(2\,\text{mx}+\delta\text{DM})+\text{mT}\,(4\,\text{mx}^2+5\,\text{mx}\,\delta\text{DM}+\delta\text{DM}^2)\big)+\text{mT}^4\,\text{mx}\,\delta\text{DM}\,(\text{mx}+\delta\text{DM})\,(\delta\text{DM}+\delta\text{EM})+2\,\text{mT}^3\,\text{mx}^2\,\delta\text{DM}\,(\text{mx}+\delta\text{DM})\,(\delta\text{DM}+\delta\text{EM})+\text{mT}^2\,\text{mx}^3\,\delta\text{DM}\,(\text{mx}+\delta\text{DM})\,(\delta\text{DM}+\delta\text{EM})-2\,\sqrt{\big(\text{ER mT}^2\,\text{mx}^3\,(\text{mT}+\text{mx})^4\,(\text{mx}+\delta\text{DM})\,(\text{mT}+\text{mx}+\delta\text{DM})^2\,(\text{ER}\,(\text{mT}+\text{mx})\,(\text{mT}+\text{mx}+\delta\text{DM})+\text{mT}\,\delta\text{DM}\,(\delta\text{DM}+\delta\text{EM}))\big)}\Big)\Bigg)};$$

This formula has a pole at δ_{DM}->0, and exhibits stability issues at very particular values, improve stability with some redefinitions:

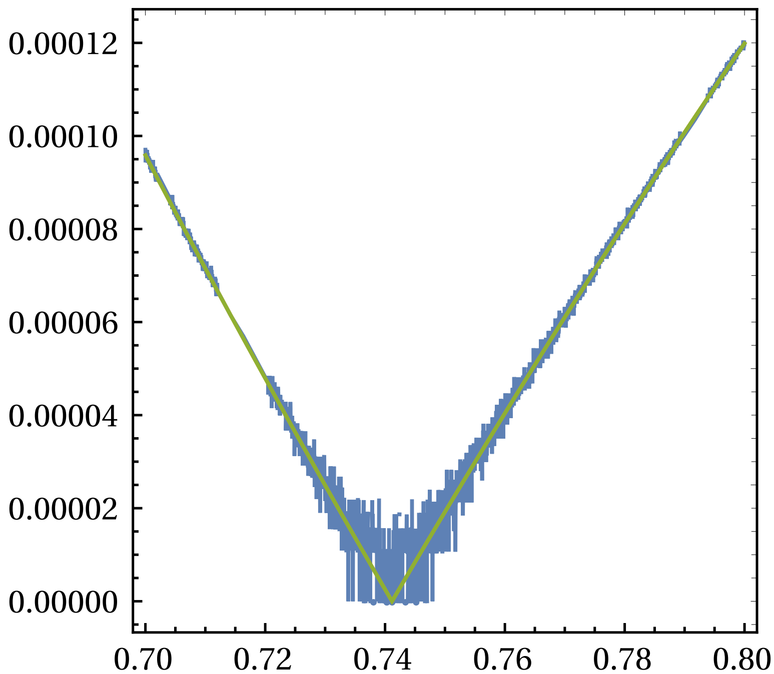
In[ ]:=
$$\left(\text{Solve}\left[0==\mu i\,(1+R)\,vi^2+2\,\sqrt{\mu i\,\mu f\,vi^2-2\,\mu f\,(\delta\text{EM}+\delta\text{DM})}\,vi+kk,\ vi\right]\text{// FullSimplify}\right)[[2]]\ \text{/.}\left\{kk\text{->}-\frac{2\,\text{mT Er}}{\mu i}-2\,(\delta\text{EM}+\delta\text{DM}),\ R\to\frac{\mu f}{\mu i}\right\}\text{/.}\left\{\mu i\to\frac{\text{mT mx}}{\text{mT}+\text{mx}},\ \mu f\text{->}\frac{\text{mT}\,(\text{mx}+\delta\text{DM})}{\text{mT}+(\text{mx}+\delta\text{DM})}\right\}\text{// FullSimplify}$$

Out[ ]=
$$\left\{vi\to2\,\sqrt{\left(-\left((\text{Er}\,(\text{mT}+\text{mx})+\text{mx}\,(\delta\text{DM}+\delta\text{EM}))^2\Big/\left(\text{mx}^2\left(-4\,\sqrt{\frac{\text{Er mT}^2\,(\text{mx}+\delta\text{DM})\,(\text{Er}\,(\text{mT}+\text{mx})\,(\text{mT}+\text{mx}+\delta\text{DM})-\text{mT}\,\delta\text{DM}\,(\delta\text{DM}+\delta\text{EM}))}{\text{mx}\,(\text{mT}+\text{mx}+\delta\text{DM})^2}}-\frac{2\,\text{mT}\,(\text{Er}\,(\text{mT}+\text{mx})\,(2\,\text{mx}\,(\text{mx}+\delta\text{DM})+\text{mT}\,(2\,\text{mx}+\delta\text{DM}))-\text{mT mx}\,\delta\text{DM}\,(\delta\text{DM}+\delta\text{EM}))}{\text{mx}\,(\text{mT}+\text{mx})\,(\text{mT}+\text{mx}+\delta\text{DM})}\right)\right)\right)\right)}\right\}$$

In[ ]:=
$$\text{Vmin}[\text{Er\_},\delta\text{DM\_},\delta\text{EM\_},\text{mx\_},\text{mT\_}]=\frac{\sqrt{2}\,\text{mT}\,(\text{Er}\,(\text{mT}+\text{mx})+\text{mx}\,(\delta\text{DM}+\delta\text{EM}))\ \text{// Abs}}{(\text{mT}+\text{mx})\,\sqrt{\dfrac{\text{mT mx}\left(\text{Er mT}\left(\frac{\text{mT mx}}{\text{mT}+\text{mx}}+\frac{\text{mT}\,(\text{mx}+\delta\text{DM})}{\text{mT}+\text{mx}+\delta\text{DM}}\right)-\frac{\text{mT}^3\,\text{mx}\,\delta\text{DM}\,(\delta\text{DM}+\delta\text{EM})}{(\text{mT}+\text{mx})^2\,(\text{mT}+\text{mx}+\delta\text{DM})}+2\,\sqrt{\frac{\text{Er mT}^4\,\text{mx}\,(\text{mx}+\delta\text{DM})\,(\text{Er}\,(\text{mT}+\text{mx})\,(\text{mT}+\text{mx}+\delta\text{DM})-\text{mT}\,\delta\text{DM}\,(\delta\text{DM}+\delta\text{EM}))}{(\text{mT}+\text{mx})^2\,(\text{mT}+\text{mx}+\delta\text{DM})^2}}\right)}{\text{mT}+\text{mx}}}};$$

In[ ]:=
```
Plot[
 {VminBad[0.023918853046650618` × 10^-6, -4 × 10^-6, 3.803008654588058`*^-8 , mx, 131 amu/GeV],
  Vmin[0.023918853046650618` × 10^-6, -4 × 10^-6, 3.803008654588058`*^-8 , mx, 131 amu/GeV],
  VminApprox[0.023918853046650618` × 10^-6, -4 × 10^-6,
   3.803008654588058`*^-8 , mx, 131 amu/GeV]}, {mx, .7, .8}]
```

Out[ ]=



Global max inelastic energy:

In[ ]:=
$$\delta\text{MaxGeV[mX\_, mT\_]} = \frac{\mu[\text{mT, mX}]\,\text{VMAX}^2}{2};$$

Maximum inelastic energy for a given nuclear recoil:

In[ ]:=
$$\text{EemMaxERGeV[ER\_, Vmax\_, mX\_, mT\_]} = \sqrt{2\,\text{mT ER Vmax}^2} - \text{ER (mT + mX) / mX};$$

Define integration limits:

In[ ]:=
```
ErMaxGeV[δDM_?NumericQ, δEM_?NumericQ, mX_?NumericQ, mT_?NumericQ] :=
 If[δMaxGeV[mX, mT] < δDM + δEM, 0, ErMax[VMAX, δDM, δEM, mX, mT]]
```

In[ ]:=
```
ErMinGeV[δDM_?NumericQ, δEM_?NumericQ, mX_?NumericQ, mT_?NumericQ] :=
 If[δMaxGeV[mX, mT] < δDM + δEM, ∞, ErMin[VMAX, δDM, δEM, mX, mT]]
```

The fractional difference between approximate formula and full formula are small in our region of interest, but start to become ~0.5% when we consider $\delta\text{DM}/m_x \sim 0.01$:

In[ ]:=

$$\texttt{Plot}\Big[\Big\{\frac{\texttt{Vmin}[\texttt{ER}\ 10^{-9}, -10^{-5}, 10^{-6}, .001, 131] - \texttt{VminApprox}[\texttt{ER}\ 10^{-9}, -10^{-5}, 10^{-6}, .001, 131]}{\texttt{Vmin}[\texttt{ER}\ 10^{-9}, -10^{-5}, 10^{-6}, .001, 131]},$$

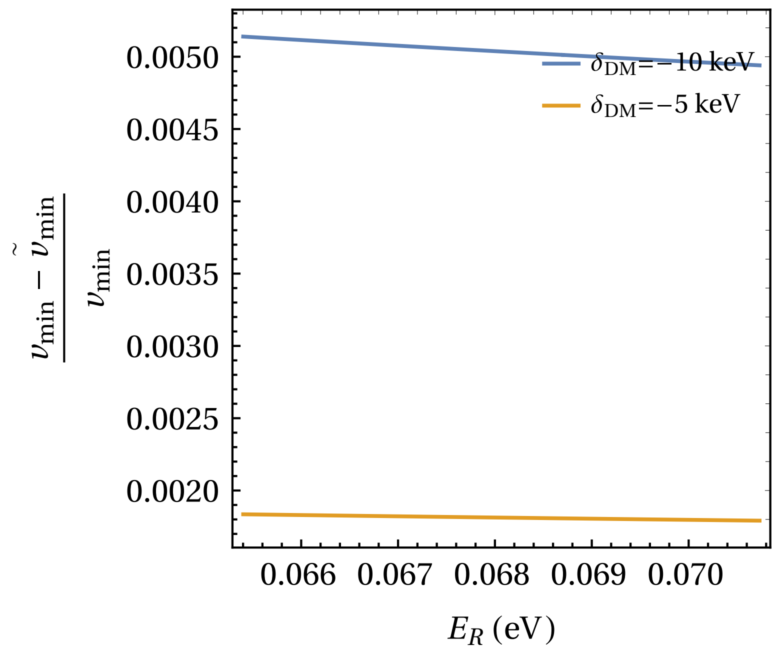
$$\frac{\texttt{Vmin}[\texttt{ER}\ 10^{-9}, -5\times 10^{-6}, 10^{-6}, .001, 131] - \texttt{VminApprox}[\texttt{ER}\ 10^{-9}, -5\times 10^{-6}, 10^{-6}, .001, 131]}{\texttt{Vmin}[\texttt{ER}\ 10^{-9}, -5\times 10^{-6}, 10^{-6}, .001, 131]}\Big\},$$

$$\{\texttt{ER}, \texttt{ErMinGeV}[-10^{-5}, 10^{-6}, .001, 131]\ 10^{9}, \texttt{ErMaxGeV}[-10^{-5}, 10^{-6}, .001, 131]\ 10^{9}\},$$

$$\texttt{FrameLabel} \to \Big\{\texttt{"}E_R\ \texttt{(eV)"}, \texttt{"}\frac{v_{min} - \tilde{v}_{min}}{v_{min}}\texttt{"}\Big\},$$

$$\texttt{PlotLegends} \to \texttt{Legend}[\{\texttt{"}\delta_{DM}\texttt{=-10 keV"}, \texttt{"}\delta_{DM}\texttt{=-5 keV"}\}]\Big]$$

Out[ ]=



Min recoil energy

mean collision time:

In[ ]:=

$$\left(\frac{.001\ \texttt{km}}{\pi\,(2\times 10^{-10})^2\left(\frac{2\,860\,000}{131}\times 6.02\times 10^{23}\right)}\right) \Big/ \left(\sqrt{\frac{3\times 170\ \texttt{Kelvin}\ 8.6\times 10^{-5}\ \texttt{eV/Kelvin}}{131\ \texttt{mn}\ 10^{9}\ \texttt{eV/GeV}}}\ \texttt{c}\right)$$

Out[ ]= 3.38213×10^{-12} s

Interatomic spacing:

In[ ]:=

$$\texttt{d} = \left(\frac{2860\ \texttt{kg}}{\texttt{meter}^3}\ \frac{6.02\times 10^{23}}{\texttt{kg}}\right)^{-1/3}\ \texttt{// Refine[\#, Assumptions} \to \texttt{meter > 0] \&}$$

Out[ ]= 8.34345×10^{-10} meter

Time to cross d using sound speed at 170K:

In[ ]:= t1 = d / vs /. vs → 626 meter / s

Out[ ]= 1.33282×10^{-12} s

This time-scale is on the conservative side of mean collision time.

Debye frequency (at 170K) - not defined for liquid - but for comparison:

In[ ]:= $\omega\text{Dinv} = 1 \Big/ \left(\left(6\,\pi^2\,\text{nXe}\,\text{soundSpeed}^3\right)^{1/3} \text{/. } \left\{\text{nXe} \to \left(\frac{2.86 \times 100^3}{\text{meter}^3}\,\frac{6.02 \times 10^{23}}{131}\right), \text{soundSpeed} \to 626\,\text{meter / s}\right\} \text{ // Refine[\#, Assumptions} \to s > 0] \,\& \right)$

Out[ ]= 1.73665×10^{-13} s

Which has energy:

In[ ]:= $\omega\text{D}\,\hbar$

Out[ ]= 6.58212×10^{-16} eV s ωD

So the impulse approximation requires time scale be faster than $\sim 10^{-12}$

In[ ]:= $\text{MinERcutoffGeV} = \frac{100\,\hbar}{\text{t1}}\,\frac{10^{-9}(\text{*GeV*})}{\text{eV}}$

Out[ ]= 4.93849×10^{-11}

Find conservative lower limit on mass:

In[ ]:= $\text{NSolve}\left[\text{ErMaxGeV}\left[(0)\,10^{-6}, 0, \text{mx}, 131\,\frac{\text{amu}}{\text{GeV}}\right] == \text{MinERcutoffGeV}, \text{mx}\right]$

NSolve : Inverse functions are being used by NSolve , so some solutions may not be found ; use Reduce for complete solution information .

Out[ ]= {{mx → 0.0209942 }}

In[ ]:= $\text{NSolve}\left[\text{ErMaxGeV}\left[(10)\,10^{-6}, 0, \text{mx}, 131\,\frac{\text{amu}}{\text{GeV}}\right] == \text{MinERcutoffGeV}, \text{mx}\right]$

NSolve : Inverse functions are being used by NSolve , so some solutions may not be found ; use Reduce for complete solution information .

Out[ ]= {{mx → 0.365626 }}

In[ ]:= $\text{NSolve}\left[\text{ErMaxGeV}\left[(-10)\,10^{-6}, 0, \text{mx}, 131\,\frac{\text{amu}}{\text{GeV}}\right] == \text{MinERcutoffGeV}, \text{mx}\right]$

NSolve : Inverse functions are being used by NSolve , so some solutions may not be found ; use Reduce for complete solution information .

Out[ ]= {{mx → 0.000595534 }}

Nuclear recoil rate

Form factor

```
In[ ]:= F[A_, ErkeV_] := 3 (Sin[q r] - (q r) Cos[q r])/(q r)^3 Exp[-(q s)^2/2] /.
   {q -> 6.92*10^-3 Sqrt[A] Sqrt[ErkeV], r -> ((1.23 A^(1/3) - .6)^2 + 7/3 π^2 (0.52)^2 - 5 (0.9)^2)^(1/2), s -> 0.9};
```

Maxwell Boltzmann with cutoff

```
In[ ]:= fMBc[v_, cosθ_, norm_] := 1/norm (Exp[-(v^2 - 2 ve/c v cosθ + (ve/c)^2)/(v0/c)^2] - Exp[-vesc^2/v0^2])
   HeavisideTheta[(vesc/c)^2 - (v^2 - 2 ve/c v cosθ + (ve/c)^2)];
```

```
In[ ]:= normMB = NIntegrate[fMBc[v , cosθ, 1] 2 π v^2, {v, 0, (ve + vesc)/c}, {cosθ, -1, 1}];
```

··· NIntegrate : Numerical integration converging too slowly ; suspect one of the following : singularity , value of the integration is 0, highly oscillatory integrand , or WorkingPrecision too small .

Find speed distribution by integrating over the angles, with work-around for the step function (which mathematica does not integrate properly):

```
In[ ]:= Fc // Clear;
   Fc[V_?NumericQ] = Piecewise[{{Integrate[fMBc[V, cosθ, normMB] 2 π V^2, {cosθ, -1, 1},
        Assumptions → V ∈ Reals], V < (vesc - ve)/c}, {0, V ≥ (vesc + ve)/c}}, Integrate[
      fMBc[V, cosθ, normMB] 2 π V^2, {cosθ, 1 - ((vesc/c)^2 - (V - ve/c)^2)/(2 ve/c V), 1}, Assumptions → V ∈ Reals]];
```

Integrate to get as function of vmin

```
In[ ]:= (*G[vmin_]=Integrate[Fc[V]/V,{V,vmin,(ve+vesc)/c},Assumptions→{vmin∈Reals,vmin>0,vmin<(ve+vesc)/c ]];
```

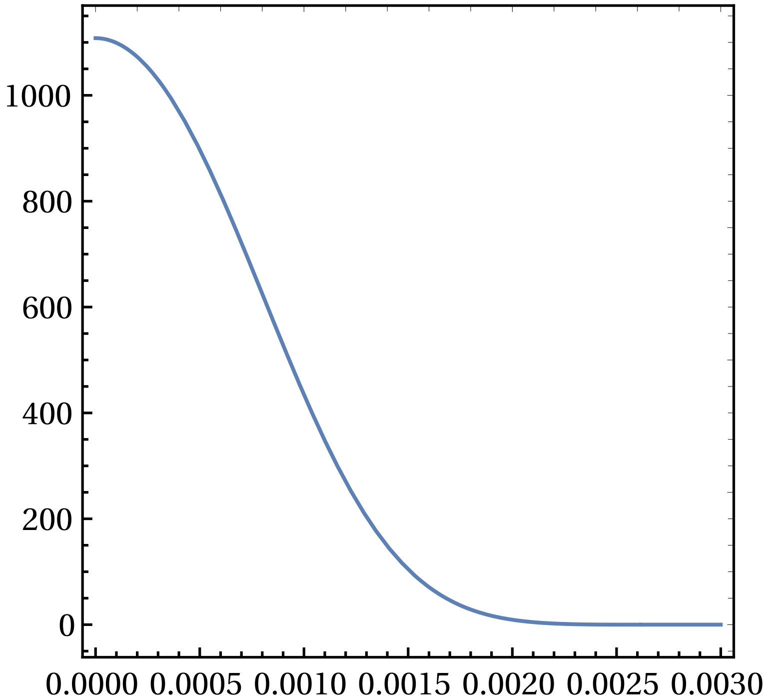
Numerical evaluation:

```
In[ ]:= integrand[v_?NumericQ] := Fc[v]/v;
```

```
In[ ]:= G = ParallelTable[{vmin, NIntegrate[integrand[V], {V, vmin, (ve + vesc)/c}]},
      {vmin, 0, (ve + vesc)/c, (ve + vesc)/(200 c)}] // Interpolation[#, InterpolationOrder → 1] &;
```

```
In[ ]:= Gv[vmin_] := If[vmin < VMAX, G[vmin], 0];
```

In[ ]:=
```
Plot[Gv[v], {v, 0, .003}]
```

Out[ ]=


Differential rates

Define rate in /tonne/yr/keV

In[ ]:=
```
dRdEr[ErkeV_?NumericQ, δDMkeV_?NumericQ,
  δEMkeV_?NumericQ, σncm2_?NumericQ, mxGeV_?NumericQ, A_?NumericQ] :=
```

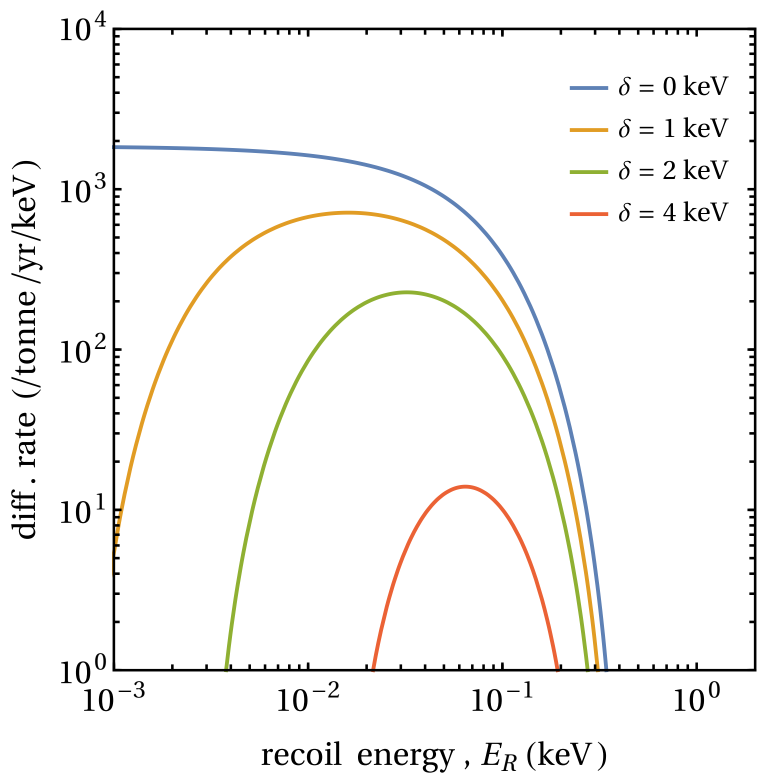
$$\left(365.25\times3600*24\,\frac{s}{1(*year*)}\right)\left(c\,\frac{100\,000\ cm}{km}\right)\left(\frac{1\ GeV}{10^{6}(*keV*)}\right)\left(GeVperKg\,\frac{1000\ kg}{1(*tonne*)}\right)*$$
$$\left(\frac{\rho}{2\ mxGeV\ GeV}\right)\left(\frac{1}{\mu[mn,\ mxGeV\ GeV]^{2}}\right)\left(A^{2}\ F[A,\ ErkeV]^{2}\right)$$
$$\left(\sigma ncm2\ cm^{2}\right)Gv\left[Vmin\left[ErkeV\ 10^{-6},\ \delta DMkeV\ 10^{-6},\ \delta EMkeV\ 10^{-6},\ mxGeV,\ A\,\frac{amu}{GeV}\right]\right]$$

Examples

Endothermic:

In[ ]:=
```
LogLogPlot[{dRdEr[ERkeV, 0, 0, 10^-45, 2, 131], dRdEr[ERkeV, 1, 0, 10^-45, 2, 131],
  dRdEr[ERkeV, 2, 0, 10^-45, 2, 131], dRdEr[ERkeV, 4, 0, 10^-45, 2, 131]}, {ERkeV, 10^-4, 40},
 PlotLegends → Legend[{"δ = 0 keV", "δ = 1 keV", "δ = 2 keV", "δ = 4 keV"}],
 FrameLabel → {"recoil energy, E_R (keV)", "diff. rate (/tonne/yr/keV)"},
 PlotRange → {{10^-3, 2}, {1, 10^4}}, FrameTicks → {LogTicks[1, 10^6], LogTicks[10^-6, 1]}]
```

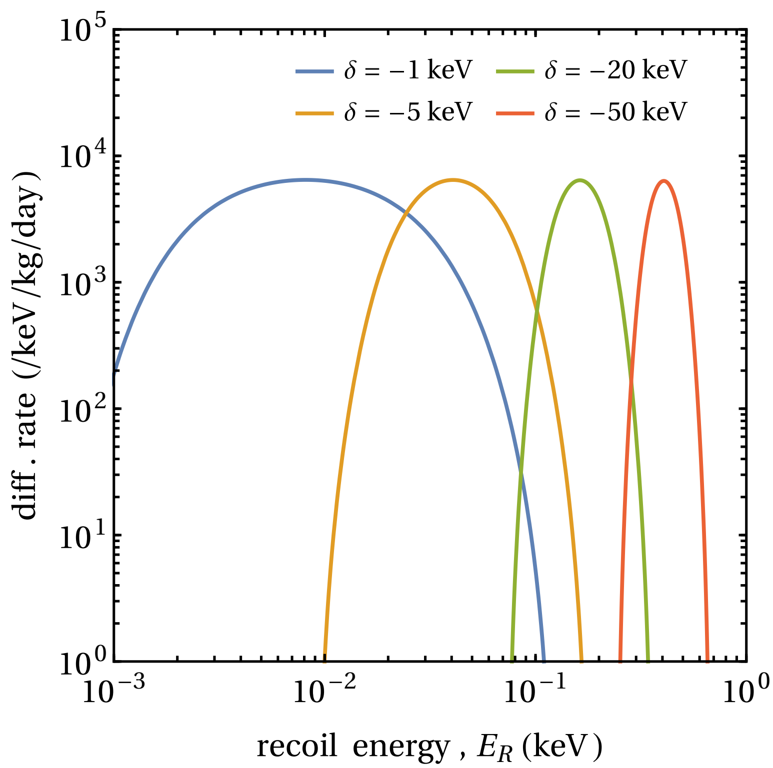
Out[ ]=



Exothermic:

In[ ]:=
```
LogLogPlot[{dRdEr[ER, -1, 0, 10^-45, 1, 131], dRdEr[ER, -5, 0, 10^-45, 1, 131],
   dRdEr[ER, -20, 0, 10^-45, 1, 131], dRdEr[ER, -50, 0, 10^-45, 1, 131]}, {ER, 10^-4, 40},
  PlotLegends → Legend[{"δ = -1 keV", "δ = -5 keV", "δ = -20 keV", "δ = -50 keV"},
    LegendLayout → {"Column", 2}, Position → {.6, .9}],
  FrameLabel → {"recoil energy, E_R (keV)", "diff. rate (/keV/kg/day)"},
  PlotRange → {{10^-3, 1}, {10^0, 10^5}},
  FrameTicks → {LogTicks[10^-6, 10^6], LogTicks[10^-6, 20]}]
```

Out[ ]=



Migdal rate

Import

In[ ]:=
```
NotebookDirectory[] // SetDirectory;
```

In[ ]:=
```
datXe = Import["data/Xe_new.dat"];
```

In[ ]:=
```
datSepXe = Table[datXe[[4 + 254 j ;; 254 + 254 j]], {j, 0, Floor[Length[datXe] / 256]}];
```

Reproducing Fig.3

Xenon

In[ ]:=
```
datN1 = Table[datXe[[4 + 254 j ;; 254 + 254 j]], {j, 0, 0}];
datN2 = Table[datXe[[4 + 254 j ;; 254 + 254 j]], {j, 1, 2}];
datN3 = Table[datXe[[4 + 254 j ;; 254 + 254 j]], {j, 3, 5}];
datN4 = Table[datXe[[4 + 254 j ;; 254 + 254 j]], {j, 6, 8}];
datN5 = Table[datXe[[4 + 254 j ;; 254 + 254 j]], {j, 9, 10}];
```

In[ ]:=
```
Me = 511 keV;
```

In[ ]:=

$$\text{totN1} = \left\{\frac{\text{datN1[[1, All, 1]]}}{1000}, \frac{1000}{2\,\pi}\left(\frac{10^{-3}\,\text{Me}}{.001\,\text{keV}}\right)^2 \text{datN1[[All, All, 2]][[1]]}\right\} \text{ // Thread};$$

$$\text{totN2} = \left\{\frac{\text{datN2[[1, All, 1]]}}{1000}, \frac{1000}{2\,\pi}\left(\frac{10^{-3}\,\text{Me}}{.001\,\text{keV}}\right)^2 \text{Apply[Plus, datN2[[All, All, 2]]]}\right\} \text{ // Thread};$$

$$\text{totN3} = \left\{\frac{\text{datN3[[1, All, 1]]}}{1000}, \frac{1000}{2\,\pi}\left(\frac{10^{-3}\,\text{Me}}{.001\,\text{keV}}\right)^2 \text{Apply[Plus, datN3[[All, All, 2]]]}\right\} \text{ // Thread};$$

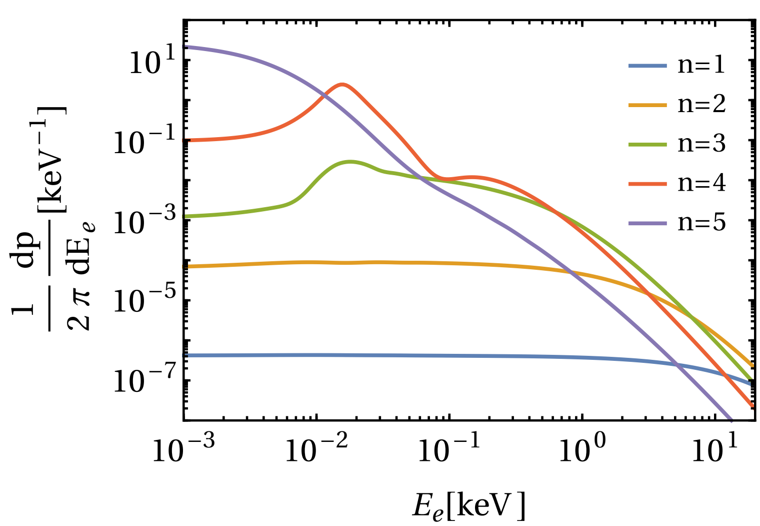
$$\text{totN4} = \left\{\frac{\text{datN4[[1, All, 1]]}}{1000}, \frac{1000}{2\,\pi}\left(\frac{10^{-3}\,\text{Me}}{.001\,\text{keV}}\right)^2 \text{Apply[Plus, datN4[[All, All, 2]]]}\right\} \text{ // Thread};$$

$$\text{totN5} = \left\{\frac{\text{datN5[[1, All, 1]]}}{1000}, \frac{1000}{2\,\pi}\left(\frac{10^{-3}\,\text{Me}}{.001\,\text{keV}}\right)^2 \text{Apply[Plus, datN5[[All, All, 2]]]}\right\} \text{ // Thread};$$

In[ ]:=

$$\begin{aligned}&\text{plot = ListLogLogPlot}\Big[\text{\{totN1, totN2, totN3, totN4, totN5\},}\\&\quad \text{Joined} \to \text{True, AspectRatio} \to .7\text{, PlotRange} \to \{\{.001, 20\}, \{10^{-8}, 100\}\},\\&\quad \text{PlotLegends} \to \text{Legend[\{"n=1", "n=2", "n=3", "n=4", "n=5"\}],}\\&\quad \text{FrameLabel} \to \left\{\text{"E}_\text{e}\text{[keV]", "}\frac{1}{2\,\pi}\frac{\text{dp}}{\text{dE}_\text{e}}\text{[keV}^{-1}\text{]"}\right\},\\&\quad \text{FrameTicks} \to \{\text{LogTicks}[10^{-8}, 100, 2], \text{LogTicks}[.001, 10]\}\Big]\end{aligned}$$

Out[ ]=



Migdal Definitions

Nuclear recoil queching factor:

```
In[ ]:= LeffXe = 0.15;
```

Momentum transfer to electron:

$$\text{In[ ]:= qeKeV[ErGeV\_, MtGeV\_]} = \frac{\text{Me}}{\text{keV}}\sqrt{\frac{2\ \text{ErGeV}}{\text{MtGeV}}};$$

List of states contained in the file

```
In[ ]:= statesXe = Table[datXe[[2 + 254 j]], {j, 0, Floor[Length[datXe] / 256]}];
```

List of binding energies of the states in keV

$$\text{In[ ]:= EstatesXe} = \frac{1}{1000}\{3.5\times10^4, 5.4\times10^3, 4.9\times10^3, 1.1\times10^3, 9.3\times10^2, 6.6\times10^2, 2.0\times10^2, 1.4\times10^2, 6.1\times10, 2.1\times10, 9.8\};$$

Array of binding energies, can be accessed as Enl[[n,l]]

$$\text{In[ ]:= EnlXe} = \frac{1}{1000}\{\{3.5\times10^4, 0, 0, 0\}, \{5.4\times10^3, 4.9\times10^3, 0, 0\}, \{1.1\times10^3, 9.3\times10^3, 6.6\times10^2, 0\}, \{2.0\times10^2, 1.4\times10^2, 6.1\times10, 0.85\}, \{2.1\times10, 9.8, 1.6, 0\}, \{3.3, 2.2, 0, 0\}\};$$

Ionization probability for the (n,l) state into a free state with kinetic energy E_e from momentum kick q_e

```
In[ ]:= z[x_] = 0;
```

```
In[ ]:= znlxeTable = Table[If[EnlXe[[ni, li + 1]] > 0, (Switch @@ Join[{{ni, li}},
          ({statesXe, Table[datSepXe[[i]] // Interpolation, {i, 1, datSepXe // Length}]} //
            Thread // Flatten[#, 1] &)]), z], {ni, 1, Length[EnlXe]}, {li, 0, 3}];
```

Setting probability to zero below threshold doesn't seem to have an affect on limits, but doesn't match rates in Ibe et al.:

$$\text{In[ ]:= ZnlXe[n\_, l\_, qEkeV\_, Ee\_] := If}\left[(1000\ \text{Ee}) > 1, \left(\frac{\text{qEkeV}}{.001}\right)^2\frac{1000}{2\pi}\text{znlxeTable}[\![n, l+1]\!][1000\ \text{Ee}], 0\right]$$

$$\text{In[ ]:= ZnlXe[n\_, l\_, qEkeV\_, Ee\_] := If}\left[(1000\ \text{Ee}) > 1, \left(\frac{\text{qEkeV}}{.001}\right)^2\frac{1000}{2\pi}\text{znlxeTable}[\![n, l+1]\!][1000\ \text{Ee}], \left(\frac{\text{qEkeV}}{.001}\right)^2\frac{1000}{2\pi}\text{znlxeTable}[\![n, l+1]\!][.001]\right];$$

Differential rate definition

The differential Migdal rate for a given atomic shell (constant quenching factor):

In[ ]:=
```
dRmigdEdet[EdetkeV_?NumericQ, δDMkeV_?NumericQ,
  σ_?NumericQ, Mx_?NumericQ, Ni_?IntegerQ] :=
 If[EdetkeV < Min[EstatesXe[[Position[statesXe[[All, 1]], Ni] // Flatten]]], 0,
  NIntegrate[
   dRdEr[ErkeV, δDMkeV, (EdetkeV - ErkeV * LeffXe), σ, Mx, 131] *
    Sum[
     UnitStep[(EdetkeV - ErkeV * LeffXe) - EstatesXe[[nl]]] *
      ZnlXe[statesXe[[nl, 1]], statesXe[[nl, 2]],
       qeKeV[ErkeV 10^-6, 131 amu/GeV], EdetkeV - ErkeV * LeffXe - EstatesXe[[nl]]],
     {nl, Position[statesXe[[All, 1]], Ni] // Flatten}],
   {ErkeV,
    ErMinGeV[(δDMkeV) 10^-6, 0, Mx, 131 amu/GeV] 10^6,
    Min[(EdetkeV - Min[EstatesXe[[Position[statesXe[[All, 1]], Ni] // Flatten]]])/.15,
     ErMaxGeV[(δDMkeV) 10^-6, 0, Mx, 131 amu/GeV] 10^6]}, PrecisionGoal → 3]]
```

Elastic Migdal rate

In[ ]:=
```
dataNR = ParallelTable[{Edet, 1/LeffXe dRdEr[Edet / LeffXe, 0, 0, 10^-40, 2, 131]},
   {Edet, logSpace[.01, ErMaxGeV[0, 0, 2, 131. amu/GeV] 10^6, 100]}];
```

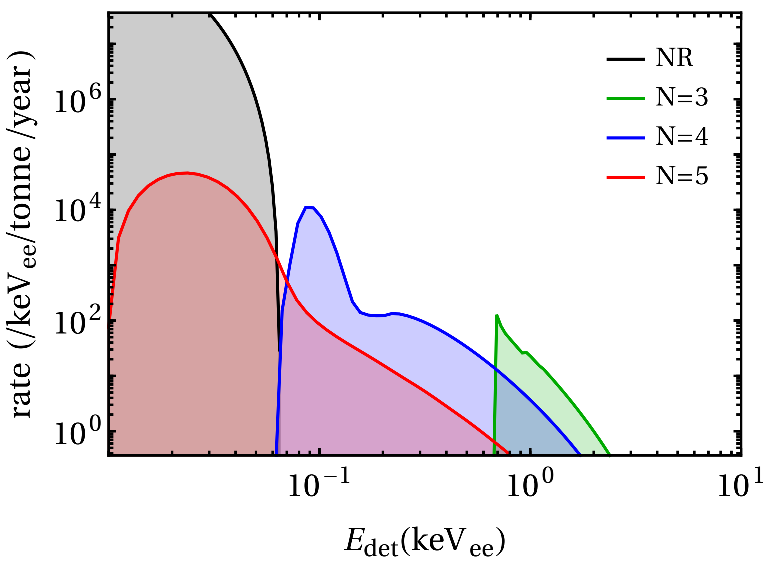
In[ ]:=
```
dataEM3 =
  ParallelTable[{Edet, dRmigdEdet[Edet, 0, 10^-40, 2, 3]}, {Edet, logSpace[.661, 6.5, 50]}];
```

In[ ]:=
```
dataEM4 =
  ParallelTable[{Edet, dRmigdEdet[Edet, 0, 10^-40, 2, 4]}, {Edet, logSpace[.0611, 4, 50]}];
```

In[ ]:=
```
dataEM5 =
  ParallelTable[{Edet, dRmigdEdet[Edet, 0, 10^-40, 2, 5]}, {Edet, logSpace[.01, 2, 50]}];
```

In[ ]:=
```
ListLogLogPlot[{dataNR, dataEM3, dataEM4, dataEM5},
  Joined → True, PlotRange → {{.01, 10}, {365 250 × 10^-6, 365 250 × 10^2}},
  FrameTicks → {LogTicks[10^-7, 10^10, 2], LogTicks[10^-2, 10^1, 1]},
  PlotStyle → {Black, Green // Darker, Blue, Red},
  FrameLabel → {"E_det(keV_ee)", "rate (/keV_ee/tonne/year)"},
  PlotLegends → Legend[{"NR", "N=3", "N=4", "N=5"}], AspectRatio → .7, Filling → Bottom]
```

Out[ ]=



This matches Fig.5 of Ibe et al.

Compare Lindhard to constant *L*

With Lindhard factor (From 1412.4417):

In[ ]:=
```
k = 0.1394;
g[ϵ_] := 3 ϵ^0.15 + 0.7 ϵ^0.6 + ϵ;
LindXe[ErkeV_] = (k g[11.5 ErkeV 54^(-7/5)]) / (1 + k g[11.5 ErkeV 54^(-7/5)])
```

Out[ ]=

$$\frac{0.1394\ (1.87251\ \text{ErkeV}^{0.15} + 0.106244\ \text{ErkeV}^{0.6} + 0.0431864\ \text{ErkeV})}{1 + 0.1394\ (1.87251\ \text{ErkeV}^{0.15} + 0.106244\ \text{ErkeV}^{0.6} + 0.0431864\ \text{ErkeV})}$$

In[ ]:=
```
LindXeF =
   Table[{er, LindXe[er]}, {er, 0, 50, .05}] // Interpolation[#, InterpolationOrder → 1] &;
```

In[ ]:=
```
dRmigLdEdet[EdetkeV_?NumericQ, δDMkeV_?NumericQ,
  σ_?NumericQ, Mx_?NumericQ, Ni_] := NIntegrate[
  dRdEr[ErkeV, δDMkeV, (EdetkeV - ErkeV * LindXeF[ErkeV]), σ, Mx, 131] *
   Sum[
    UnitStep[(EdetkeV - ErkeV * LindXeF[ErkeV]) - EstatesXe[[nl]]] *
     ZnlXe[statesXe[[nl, 1]], statesXe[[nl, 2]],
      qeKeV[ErkeV 10^-6, 131 amu/GeV], EdetkeV - ErkeV * LindXeF[ErkeV] - EstatesXe[[nl]]],
    {nl, Position[statesXe[[All, 1]], Ni] // Flatten}],
  {ErkeV, ErMinGeV[(δDMkeV) 10^-6, 0, Mx, 131. amu/GeV] 10^6,
   ErMaxGeV[(δDMkeV) 10^-6, 0, Mx, 131. amu/GeV] 10^6}, PrecisionGoal → 3]
```

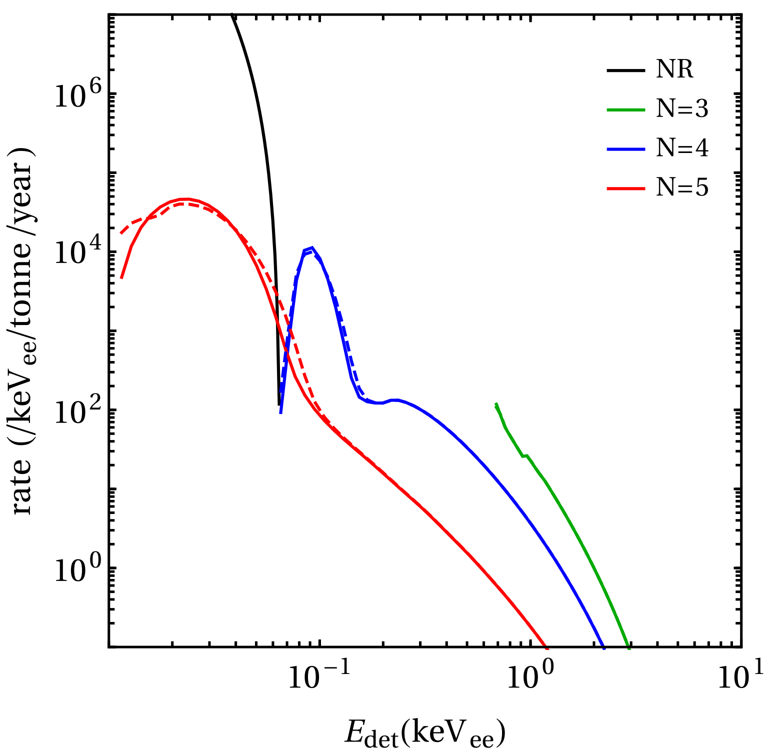
In[ ]:=
```
dataEML3 = ParallelTable[
   {Edet, dRmigLdEdet[Edet, 0, 10^-40, 2, 3]}, {Edet, logSpace[.626, 6.5, 50]}];
```

In[ ]:=
```
dataEML4 =
  ParallelTable[{Edet, dRmigLdEdet[Edet, 0, 10^-40, 2, 4]}, {Edet, logSpace[.06, 4, 50]}];
```

In[ ]:=
```
dataEML5 =
  ParallelTable[{Edet, dRmigLdEdet[Edet, 0, 10^-40, 2, 5]}, {Edet, logSpace[.0115, 2, 50]}];
```

In[ ]:=
```
migPlotN = Show[ListLogLogPlot [{dataNR , dataEM3 , dataEM4 , dataEM5},
    Joined → True , PlotRange → {{.01 , 10}, {10^-1 , 10^7}},
    FrameTicks → {LogTicks[10^-7 , 10^10 , 2], LogTicks[10^-2 , 10^1 , 1]},
    PlotStyle → {Black , Green // Darker , Blue , Red},
    FrameLabel → {"E_det(keV_ee)", "rate (/keV_ee/tonne/year)"},
    PlotLegends → Legend[{"NR", "N=3", "N=4", "N=5"}]],
  ListLogLogPlot [{dataEML3 , dataEML4 , dataEML5}, Joined → True ,
    PlotStyle → {{Dashed , Green // Darker}, {Dashed , Blue}, {Dashed , Red}}]]
```

Out[ ]=



Best fit rates

Define integrated rate

In[ ]:=
```
XeltMigRate[eMin_?NumericQ, eMax_?NumericQ, σ_?NumericQ,
   Mx_?NumericQ, δDMkeV_?NumericQ, ERthGeV_: MinERcutoffGeV] := NIntegrate[
   dRdEr[ErkeV, δDMkeV, (EdetkeV - ErkeV * LeffXe), σ, Mx, 131] *
    Sum[
     UnitStep[(EdetkeV - ErkeV * LeffXe) - EstatesXe[[nl]]] *
      ZnlXe[statesXe[[nl, 1]], statesXe[[nl, 2]],
       qeKeV[ErkeV 10^-6, 131 amu/GeV], EdetkeV - ErkeV * LeffXe - EstatesXe[[nl]]],
     {nl, Range[4, 11]}],
   {EdetkeV, eMin, eMax},
   {ErkeV,
    Max[ErMinGeV[(δDMkeV) 10^-6, 0, Mx, 131. amu/GeV] 10^6, ERthGeV 10^6],
    Max[ErMaxGeV[(δDMkeV) 10^-6, 0, Mx, 131. amu/GeV] 10^6, ERthGeV 10^6]}, PrecisionGoal → 3]
```

Calculate elastic rates

In[ ]:=
```
nBins = 5;
```

In[ ]:=
```
binB2 = logSpace[.186, 2, nBins + 1];
```

In[ ]:=
```
elasMigRateM2 = Table[XeltMigRate[binB2[[ii]], binB2[[ii + 1]], 10^-40, 2, 0], {ii, 1, nBins}];
```

In[ ]:=
```
binBp5 = logSpace[.186, 1.5, nBins + 1];
```

In[ ]:=
```
elasMigRateMp5 =
  Table[XeltMigRate[binBp5[[ii]], binBp5[[ii + 1]], 10^-40, 0.5, 0], {ii, 1, nBins}];
```

Calculate integrated rate for plots:

In[ ]:=
```
migTotM2δ0 =
  ParallelTable[{Edet, dRmigdEdet[Edet, 0, 10^-40, 2, 5] + dRmigdEdet[Edet, 0, 10^-40, 2, 4] +
      dRmigdEdet[Edet, 0, 10^-40, 2, 3]}, {Edet, logSpace[.01, 3, 100]}];
```

In[ ]:=
```
migTotMp5δ0 = ParallelTable[
   {Edet, dRmigdEdet[Edet, 0, 10^-40, 0.5, 5] + dRmigdEdet[Edet, 0, 10^-40, 0.5, 4] +
     dRmigdEdet[Edet, 0, 10^-40, 0.5, 3]}, {Edet, logSpace[.01, 3, 100]}];
```

Combined plots

m_χ = 2 GeV

```
BestFitM2 =
  {{NMinimize[{dataEMin =
        Table[Xe1tMigRate[binB2[[ii]], binB2[[ii + 1]], 10^-40, mass, -4], {ii, 1, nBins}];
       Plus @@ (elasMigRateM2 - σ dataEMin)^2 / elasMigRateM2, σ > 0, 0.4 < mass < 2},
      {σ, mass}, PrecisionGoal → 2], δDM → -4} // Flatten
   , {NMinimize[{dataEMin = Table[Xe1tMigRate[binB2[[ii]],
          binB2[[ii + 1]], 10^-40, mass, 4], {ii, 1, nBins}];
       Plus @@ (elasMigRateM2 - σ dataEMin)^2 / elasMigRateM2, σ > 0, 0.4 < mass < 2},
      {σ, mass}, PrecisionGoal → 2], δDM -> 4} // Flatten};
```

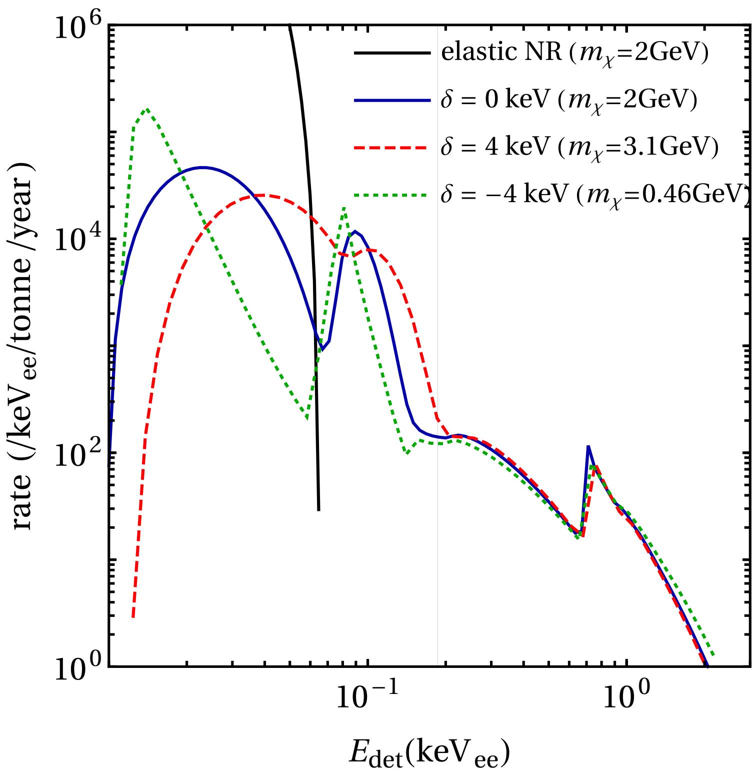
Un-comment to use pre-calculated values:

```
(*BestFitM2 =
  {{0.8718096784364927` ,σ→0.6803322377489064` ,mass→0.4582381004697444` ,δDM→-4},
   {0.3174743804851736` ,σ→3.4978366680522615` ,mass→3.057260120368556` ,δDM→4}};*)
```

```
In[ ]:= migTotBFexoM2δ4 = ParallelTable[{Edet, dRmigdEdet[Edet, δDM, σ 10^-40, mass, 5] +
         dRmigdEdet[Edet, δDM, σ 10^-40, mass, 4] + dRmigdEdet[Edet, δDM, σ 10^-40, mass, 3]} /.
       BestFitM2[[1, 2 ;; All]], {Edet, logSpace[.01, 2.2, 50]}];
migTotBFendoM2δ4 = ParallelTable[{Edet, dRmigdEdet[Edet, δDM, σ 10^-40, mass, 5] +
         dRmigdEdet[Edet, δDM, σ 10^-40, mass, 4] + dRmigdEdet[Edet, δDM, σ 10^-40, mass, 3]} /.
       BestFitM2[[2, 2 ;; All]], {Edet, logSpace[.01, 2, 50]}];
```

In[ ]:=
```
migPlotBFm2 = ListLogLogPlot[{dataNR, migTotM2δ0, migTotBFendoM2δ4, migTotBFexoM2δ4},
  Joined → True, PlotRange → {{.01, 3}, {10^0, 10^6}},
  FrameTicks → {LogTicks[10^-7, 10^10, 2], LogTicks[10^-2, 10^1, 1]},
  PlotStyle → {Black, Blue // Darker, {Red, Dashed}, {Green // Darker, Dotted}},
  FrameLabel → {"E_det(keV_ee)", "rate (/keV_ee/tonne/year)"},
  GridLines → {{.186}, {}},
  PlotLegends → Legend[{"elastic NR (m_χ=2GeV)", "δ = 0 keV (m_χ=2GeV)",
    "δ = 4 keV (m_χ=3.1GeV)", "δ = -4 keV (m_χ=0.46GeV) "}, Position → {.7, .84}]]
```

Out[ ]=



m_χ = 0.5 GeV

```
BestFitMp5 =
  {{NMinimize[{dataEMin = ParallelTable[
        Xe1tMigRate[binBp5[[ii]], binBp5[[ii + 1]], 10^-40, mass, -4], {ii, 1, nBins}];
      Plus @@ (elasMigRateMp5 - σ dataEMin)^2 / elasMigRateMp5, σ > 0, 0.005 < mass < .5},
     {σ, mass}, PrecisionGoal → 2], δ → -4} // Flatten,
   {NMinimize[{dataEMin = ParallelTable[Xe1tMigRate[binBp5[[ii]],
         binBp5[[ii + 1]], 10^-40, mass, 4], {ii, 1, nBins}];
      Plus @@ (elasMigRateMp5 - σ dataEMin)^2 / elasMigRateMp5, σ > 0, 0.5 < mass < 2},
     {σ, mass}, PrecisionGoal → 2], δ → 4} // Flatten};
```

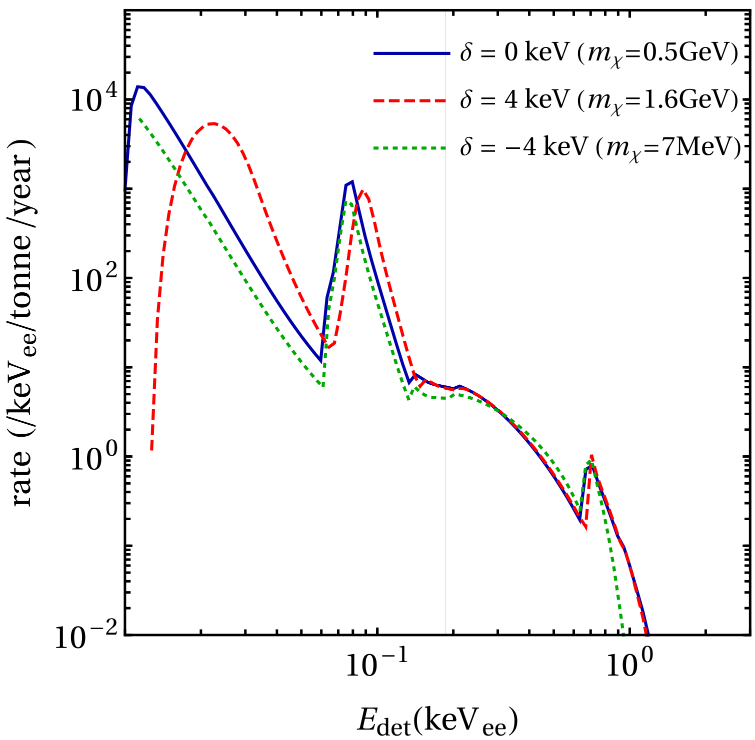
Un-comment to use pre-calculated values:

```
(*BestFitMp5 ={{0.039254515608165286` ,
    σ→0.008360619758029353` ,mass→0.007329923913981587` ,δ→-4},
   {0.0007578776792402522` ,σ→27.36059009062346` ,mass→1.6363565810444212` ,δ→4}};*)
```

```
In[ ]:= migTotBFexoMp5δ4 = ParallelTable[{Edet, dRmigdEdet[Edet, δ, σ 10^-40, mass, 5] +
        dRmigdEdet[Edet, δ, σ 10^-40, mass, 4] + dRmigdEdet[Edet, δ, σ 10^-40, mass, 3]} /.
      BestFitMp5[[1, 2 ;; All]], {Edet, logSpace[.0115, 2, 100]}];
migTotBFendoMp5δ4 = ParallelTable[{Edet, dRmigdEdet[Edet, δ, σ 10^-40, mass, 5] +
        dRmigdEdet[Edet, δ, σ 10^-40, mass, 4] + dRmigdEdet[Edet, δ, σ 10^-40, mass, 3]} /.
      BestFitMp5[[2, 2 ;; All]], {Edet, logSpace[.0115, 2, 100]}];
```

In[ ]:=
```
migPlotBFmp5 = ListLogLogPlot[{migTotMp5δ0, migTotBFendoMp5δ4, migTotBFexoMp5δ4},
  Joined → True, PlotRange → {{.01, 3}, {10^-2, 10^5}},
  FrameTicks → {LogTicks[10^-7, 10^10, 2], LogTicks[10^-2, 10^1, 1]},
  PlotStyle → {Blue // Darker, {Red, Dashed}, {Green // Darker, Dotted}},
  FrameLabel → {"E_det(keV_ee)", "rate (/keV_ee/tonne/year)"},
  GridLines → {{.186}, {}},
  PlotLegends → Legend[{"δ = 0 keV (m_χ=0.5GeV)",
     "δ = 4 keV (m_χ=1.6GeV)", "δ = -4 keV (m_χ=7MeV) "}, Position → {0.7, .85}]]
```

Out[ ]=



In[ ]:=
```
Export["./fig_migRate_M2.pdf", migPlotBFm2];
Export["./fig_migRate_Mp5.pdf", migPlotBFmp5];
```

Experimental constraints

Nuclear recoil limits

S2-only analysis

In[ ]:=
```
NotebookDirectory[] // SetDirectory;
```

In[ ]:=
```
s2datRaw = Import["data/s2_only_data.csv"] // Drop[#, 1] &;
```

In[ ]:=
```
s2bgExp = Import["data/s2_only_expected.csv"] // Drop[#, 1] &;
```

```
In[ ]:= s2datBins = Import["data/xe1t_s2only_bins.csv"];
```

```
In[ ]:= s2erCalib = Import["data/xe1t_s2only_s2er_calib.csv"] //
          Interpolation[{#[[All, 2]], #[[All, 1]]} // Thread, InterpolationOrder → 1] &;
```

```
In[ ]:= s2nrXe1tLim = Import["data/xe1t_SInr_S2only_2019.csv"];
```

Effective exposure in tonne-years

```
In[ ]:= xe1tEffectiveExpNRS2 =
         Import["data/xe1t_effExp_s2only_NR.csv"] // {#[[All, 1]], #[[All, 2]] / 365.25} & // Thread //
            Prepend[#, {.7, 0}] & // Prepend[#, {0, 0}] & //
          Interpolation[#, InterpolationOrder → 1] &;
```

Convert s2 data to absolute number of events:

```
bins = logSpace[150, 3000, Length[s2datRaw] + 1];
```

```
binsWidthsER = Table[s2erCalib[bins[[i + 1]]] - s2erCalib[bins[[i]]], {i, 1, Length[bins] - 1}];
```

Data reported per tonne.day. keV_{ee}, convert this to total number of events in the 22 tonne.day exposure:

```
In[ ]:= s2dat = 22 s2datRaw[[All, 2]] binsWidthsER ;
        s2exp = 22 s2bgExp[[All, 2]] binsWidthsER ;
```

Total expected events:

```
In[ ]:= Nexp = Plus @@ s2exp
Out[ ]= 23.3908
```

Total observed:

```
In[ ]:= Nobs = Plus @@ s2dat
Out[ ]= 60.852
```

Find CL90%

```
In[ ]:= logpERs2[x_, bgexp_, obs_, bgN_] = -(bgexp bgN + x) +
          obs Log[bgexp bgN + x] - Log[obs !] + Log[PDF[NormalDistribution[1, .15], bgN]];
```

```
In[ ]:= sol =
         FindMinimum[{(√(-2 (logpERs2[x, Nexp, Nobs, 1] - logpERs2[Nobs - Nexp, Nexp, Nobs, 1])) -
              InverseCDF[NormalDistribution[0, 1], 0.90])^2, x > 1}, {x, 40}]
Out[ ]= {7.02364 × 10^-17, {x → 48.0131}}
```

Or including uncertainty in background:

bgNorm which maximizes the likelihood for a given x:

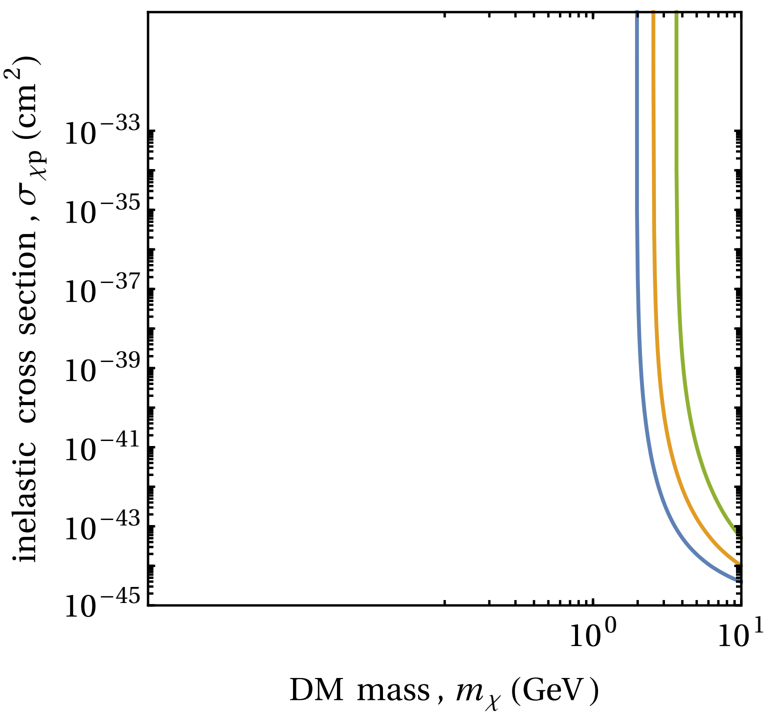
```
In[ ]:= bgNcondMax[x_, bgexp_, obs_] =
   bgN /. (Solve[D[-(bgexp bgN + x) + obs Log[bgexp bgN + x] - Log[obs!] +
         Log[PDF[NormalDistribution[1, .15], bgN]], bgN] == 0, bgN][[2, 1]]);
```

Solve : Solve was unable to solve the system with inexact coefficients . The answer was obtained by solving a corresponding exact system and numericizing the result .

```
In[ ]:= logpERs2[x_?NumericQ, bgexp_, obs_] =
   -(bgexp bgNcondMax[x, bgexp, obs] + x) + obs Log[bgexp bgNcondMax[x, bgexp, obs] + x] -
    Log[obs!] + Log[PDF[NormalDistribution[1, .15], bgNcondMax[x, bgexp, obs]]];

In[ ]:= sol = FindMinimum[
   {(√(-2 (logpERs2[x, Nexp, Nobs, bgNcondMax[x, Nexp, Nobs]] - logpERs2[Nobs - Nexp, Nexp,
            Nobs, 1])) - InverseCDF[NormalDistribution[0, 1], 0.90])^2, x > 1}, {x, 40}]

Out[ ]= {7.77418 × 10^-17, {x → 48.8913}}

In[ ]:= NupperLimit = x /. sol[[2]]

Out[ ]= 48.8913

In[ ]:= NReventsXe1tS2only[δDMkeV_?NumericQ, σncm2_?NumericQ, mxGeV_?NumericQ] :=
  If[δMaxGeV[mxGeV, 131 amu/GeV] < δDMkeV 10^-6, 0,
   (365.25 × 3600 * 24 s/1(*year*)) (c 100000 cm/km) (1 GeV/10^6(*keV*)) (GeVperKg 1000 kg/1(*tonne*)) *
    (ρ/(2 mxGeV GeV)) (1/μ[mn, mxGeV GeV]^2) (σncm2 cm^2) NIntegrate[xe1tEffectiveExpNRS2[ErkeV]
      (131^2 F[131, ErkeV]^2) Gv[Vmin[ErkeV 10^-6, δDMkeV 10^-6, 0, mxGeV, 131 amu/GeV]],
     {ErkeV, Max[.7, 10^6 ErMinGeV[δDMkeV 10^-6, 0, mxGeV, 131 amu/GeV]],
      Min[40, 10^6 ErMaxGeV[δDMkeV 10^-6, 0, mxGeV, 131 amu/GeV]]},
     PrecisionGoal → 2, Method → {Automatic, "SymbolicProcessing" → False}]]

In[ ]:= xe1tS2datNRmxCSendo = ParallelTable[
    {mx, NupperLimit / (NReventsXe1tS2only[δ, 1, mx] + 10^-100)},
    {δ, {0, 10, 50, 100, 200}},
    {mx, logSpace[1, 10, 200]}];

In[ ]:= xe1tS2datNRmxCSexo = ParallelTable[
    {mx, NupperLimit / (NReventsXe1tS2only[δ, 1, mx] + 10^-100)},
    {δ, {0, -5, -10, -50}},
    {mx, logSpace[1, 10, 200]}];
```

In[ ]:=
```
s2nrLim = ListLogLogPlot[{xe1tS2datNRmxCSexo[[3]],
   xe1tS2datNRmxCSendo[[1]], xe1tS2datNRmxCSendo[[2]]}, Joined → True,
  PlotRange → {{.001, 10}, {10^-45, 10^-30}},
  FrameLabel → {"DM mass, mχ (GeV)", "inelastic cross section, σχp (cm^2)"},
  FrameTicks → {LogTicks[10^-48, 10^-33, 2], LogTicks[10^0, 10^4]}]
```

Out[ ]=



Migdal limits

S2-only analysis

In[ ]:=
```
xe1tEffectiveExpERS2 =
  Import["data/xe1t_effExp_s2only_ER.csv"] // {#[[All, 1]], #[[All, 2]] / 365.25} & // Thread //
     Prepend[#, {.183, 0}] & // Prepend[#, {0, 0}] & //
   Interpolation[#, InterpolationOrder → 1] &;
```

```
In[ ]:= Xe1tMigRateS2[σ_?NumericQ, Mx_?NumericQ, δDMkeV_?NumericQ,
  ERthGeV_ : MinERcutoffGeV] := If[δMaxGeV[Mx, 131 amu/GeV] 10^6 < δDMkeV, 0,
  NIntegrate[xe1tEffectiveExpERS2[EdetkeV] ×
    dRdEr[ErkeV, δDMkeV, (EdetkeV - ErkeV * LeffXe), σ, Mx, 131] *
     Sum[
      UnitStep[(EdetkeV - ErkeV * LeffXe) - EstatesXe[[nl]]] *
       ZnlXe[statesXe[[nl, 1]], statesXe[[nl, 2]],
        qeKeV[ErkeV 10^-6, 131. amu/GeV], EdetkeV - ErkeV * LeffXe - EstatesXe[[nl]]],
      {nl, {Position[statesXe[[All, 1]], 4], Position[statesXe[[All, 1]], 3],
          Position[statesXe[[All, 1]], 5]} // Flatten}],
   {EdetkeV, 0.189, 3.827},
   {ErkeV, Max[ErMinGeV[(δDMkeV) 10^-6, 0, Mx, 131. amu/GeV] 10^6, ERthGeV 10^6],
    Max[ErMaxGeV[(δDMkeV) 10^-6, 0, Mx, 131 amu/GeV] 10^6, ERthGeV 10^6]}, PrecisionGoal → 2]]

In[ ]:= xe1tMigdalLimS2 = {ParallelTable[
     {mx, NupperLimit / (Xe1tMigRateS2[1, mx, -10] + 10^-100)},
     {mx, logSpace[.00065, 3, 60]}],
    ParallelTable[
     {mx, NupperLimit / (Xe1tMigRateS2[1, mx, 0] + 10^-100)},
     {mx, logSpace[.04, 4, 40]}],
    ParallelTable[
     {mx, NupperLimit / (Xe1tMigRateS2[1, mx, 10] + 10^-100)},
     {mx, logSpace[1, 6, 20]}]};

In[ ]:= MigLimitPlot = Show[
    ListLogLogPlot[{xe1tS2datNRmxCSexo[[3, 1 ;; 172]], xe1tS2datNRmxCSendo[[1, 1 ;; 181]],
      xe1tS2datNRmxCSendo[[2, 1 ;; 183]]}, Joined → True,
     PlotRange → {{.00061, 8}, {10^-43, 10^-31}},
     PlotStyle → {Green // Darker, Blue, Red // Darker},
     FrameLabel → {"DM mass, mχ (GeV)", "inelastic cross section, σχp (cm²)"},
     FrameTicks → {LogTicks[10^-47, 10^-30, 2], LogTicks[10^-3, 10^4]}],
    ListLogLogPlot[xe1tMigdalLimS2, Joined → True, PlotStyle →
      {{DotDashed, Green // Darker}, {DotDashed, Blue}, {DotDashed, Red // Darker}}]];
```

LZ projections

Define number of events for LZ assuming a threshold of 0.5 keV_{ee} and using Xe1t ER efficiency:

```
In[ ]:= xe1tEff =
   Import["data/xenon1t_eff_excess.csv"] // Interpolation[#, InterpolationOrder → 1] &;
```

```
In[ ]:= LZMigEvents[σ_?NumericQ, Mx_?NumericQ, δDMkeV_?NumericQ,
   ERthGeV_ : MinERcutoffGeV] := If[δMaxGeV[Mx, 131 amu/GeV] 10^6 < δDMkeV, 0,
   5.6 × 1000/365.25 NIntegrate[xe1tEff[EdetkeV] ×
     dRdEr[ErkeV, δDMkeV, (EdetkeV - ErkeV * LeffXe), σ, Mx, 131] *
      Sum[
       UnitStep[(EdetkeV - ErkeV * LeffXe) - EstatesXe[[nl]]] *
        ZnlXe[statesXe[[nl, 1]], statesXe[[nl, 2]],
         qeKeV[ErkeV 10^-6, 131. amu/GeV], EdetkeV - ErkeV * LeffXe - EstatesXe[[nl]]],
       {nl, {Position[statesXe[[All, 1]], 4], Position[statesXe[[All, 1]], 3],
          Position[statesXe[[All, 1]], 5]} // Flatten}],
     {EdetkeV, .5, 4},
     {ErkeV, Max[ErMinGeV[(δDMkeV) 10^-6, 0, Mx, 131. amu/GeV] 10^6, ERthGeV 10^6],
      Max[ErMaxGeV[(δDMkeV) 10^-6, 0, Mx, 131 amu/GeV] 10^6, ERthGeV 10^6]}, PrecisionGoal → 2]]
```

Dominant background in low-energy region is Rn222, at 2x 10^{-5} dru

```
In[ ]:= Nexp = Nobs = 2 × 10^-5 × 5600 × 1000 × 3.5
Out[ ]= 392.
```

```
In[ ]:= sol = FindMinimum[
   {((√(-2 (logpERs2[x, Nexp, Nobs, bgNcondMax[x, Nexp, Nobs]] - logpERs2[Nobs - Nexp, Nexp,
            Nobs, 1])) - InverseCDF[NormalDistribution[0, 1], 0.90])^2, x > 1}, {x, 40}]
Out[ ]= {1.23993 × 10^-15, {x → 79.5689}}
```

```
In[ ]:= NupperLimitLZ = x /. sol[[2]];
```

```
In[ ]:= LZMigdalLim = {ParallelTable[
     {mx, NupperLimitLZ / (LZMigEvents[1, mx, -10] + 10^-100)},
     {mx, logSpace[.00065, 3, 50]}],
    ParallelTable[
     {mx, NupperLimitLZ / (LZMigEvents[1, mx, 0] + 10^-100)},
     {mx, logSpace[.1, 4, 40]}],
    ParallelTable[
     {mx, NupperLimitLZ / (LZMigEvents[1, mx, 10] + 10^-100)},
     {mx, logSpace[3, 6, 20]}]};
```

In[ ]:=
```
LZprojPlot = ListLogLogPlot [LZMigdalLim ,
    Joined → True , PlotStyle → {{Thick , Dashing[Tiny], Green // Darker},
      {Dashing[Tiny], Blue}, {Dashing[Tiny], Red // Darker}}];
```

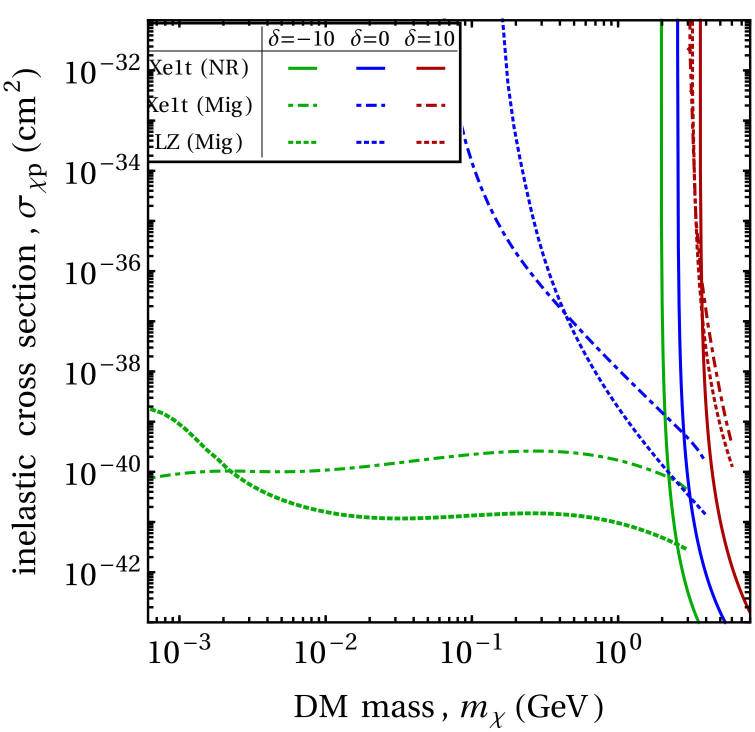
Assemble final plot

In[ ]:=
```
table[pairs_] := TableForm[{pairs[[1 ;; 3, 1]], pairs[[4 ;; 6, 1]], pairs[[7 ;; 9, 1]]},
   TableHeadings → Map[Style[#, 12, FontFamily → "Times"] &,
     {{"Xe1t (NR)", "Xe1t (Mig)", "LZ (Mig)"}, {"δ=-10", "δ=0", "δ=10"}}, {2}],
   TableAlignments → Center , TableSpacing → {1, 1}]
```

In[ ]:=
```
legend = ListLogLogPlot [{{0}, {0}, {0}, {0}, {0}, {0}, {0}, {0}, {0}},
    Joined → True , PlotStyle → {{Green // Darker}, {Blue}, {Red // Darker},
      {DotDashed , Green // Darker}, {DotDashed , Blue}, {DotDashed , Red // Darker},
      {Dashing[Tiny], Green // Darker}, {Dashing[Tiny], Blue}, {Dashing[Tiny], Red // Darker}},
    PlotLegends → Legend[{"", "", "", "", "", "", "", "", ""},
      LegendMargins → {{0, 0}, {0, 0}}, LegendMarkerSize → 15,
      LegendFunction → (Framed[#, FrameMargins → 0, Background → White] &),
      LegendLayout → table , Position → {.255 , .88}]];
```

In[ ]:=
```
fullBoundPlot = Show[MigLimitPlot , LZprojPlot , legend]
```

Out[ ]=




```
Export["./fig_bounds.pdf", fullBoundPlot ];
```