5/4/2021 Milind Diwan

This is a parameterized calculation using
arXiv: 0012075v3.pdf, Murayama and Pierce, PRD 65 013012, 2002

This is of course based on various sources, in particular the phenomelogical parameterization by P. Vogel and J. Engel. Phys. Rev. D39, 3378 (1989).

This has been superseded by more recent and precise calculations by Mueller, Huber, and ab-initio calculations. For a review see: Huber, Berryman, 2005.01756

However, this parametrization is still useful as a pedagogical tool and for quick order of magnitude (or better) estimates.

Towards the end of this notebook we also provide a tabulated spectrum.

Parameters

```
In[•]:= a0U235 = 0.870; a1U235 = -0.160; a2U235 = -0.0910;
        a0P239 = 0.896; a1P239 = -0.239; a2P239 = -0.0981;
        a0U238 = 0.976; a1U238 = -0.162; a2U238 = -0.0790;
        a0P241 = 0.793; a1P241 = -0.080; a2P241 = -0.1085;
        JoulesToMeV = 6.24150974 * 10^12;
        EperFisU235 = 201.7;
        EperFisP239 = 205.0;
        EperFisU238 = 210.;
        EperFisP241 = 212.4;
        GWtoMeVperDay = 10^9 * 24 * 3600 * JoulesToMeV;
        TargetMass = 10^6; (* gm*)
        Avagadro = 6.022141 * 10^23;
        TargetNucleons = TargetMass * Avagadro;
        HydrogenFraction = 30 / (30 + 12 * 18); (* for C18H30 *)
```

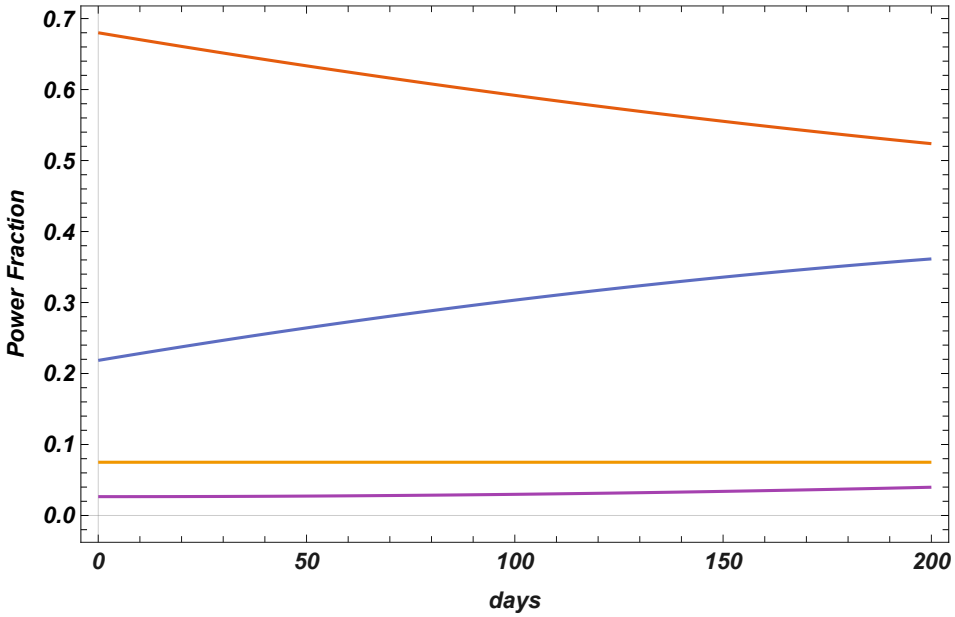
Fuel burn up model. This is generally provided by the reactor operator

In[•]:=
```
(*U235VsTime[day_] :=0.69- day*(.69-.52)/220;*)
U235VsTime[Day_] := 0.68 - 0.000982 * Day + 1.004 * 10^-6 * Day^2;
(*Pu239VsTime[day_]:= 0.22 +day*(0.36-0.22)/220; *)
Pu239VsTime[day_] := 0.21845630524445778 +
   0.0009830227186944146 day - 1.3407534926252103 * 10^-6 * day^2;
U238VsTime[day_] := 0.075;
Pu241VsTime[day_] := 1. - U235VsTime[day] - Pu239VsTime[day] - U238VsTime[day];
NusU235[E_] := Exp[a0U235 + a1U235 * E + a2U235 * E^2];
NusP239[E_] := Exp[a0P239 + a1P239 * E + a2P239 * E^2];
NusU238[E_] := Exp[a0U238 + a1U238 * E + a2U238 * E^2];
NusP241[E_] := Exp[a0P241 + a1P241 * E + a2P241 * E^2];
```

Fuel burn up plot.

In[•]:=
```
Plot[{U235VsTime[d], Pu239VsTime[d], U238VsTime[d], Pu241VsTime[d]},
 {d, 0, 200}, PlotTheme → "Scientific", LabelStyle → {Bold, Medium, Italic},
 FrameStyle → {Bold , GrayLevel[0.1]}, FrameLabel → {"days", "Power Fraction"}]
```

Out[•]=



Calculate the anti-neutrino spectrum on a particular day after turn on per GW rate is normalized to be per MeV per day per m^2 at the given distance in m. E is in MeV

In[•]:=

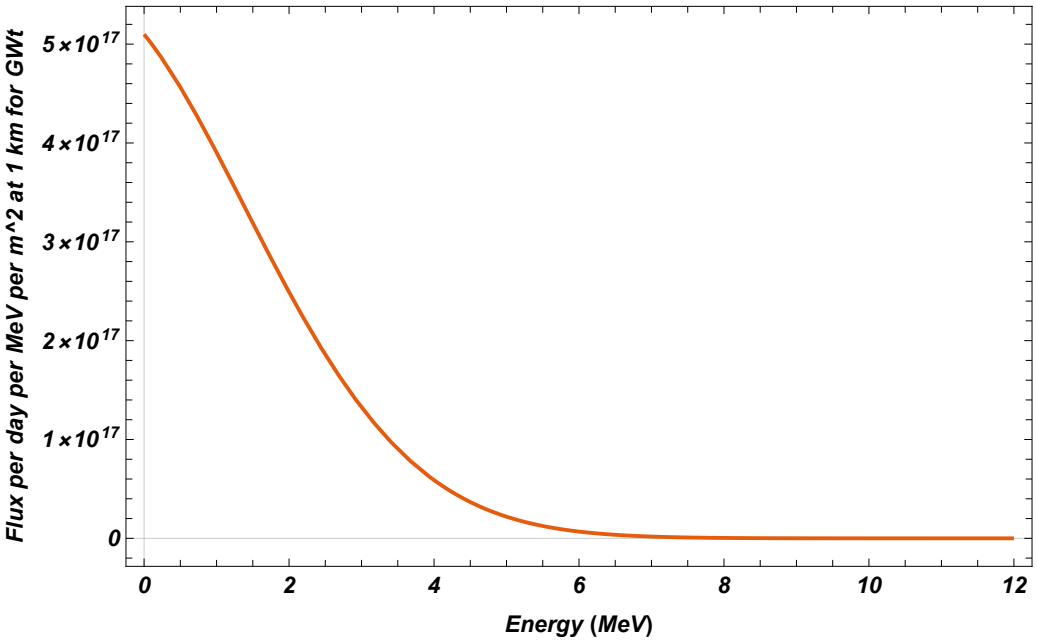
```
AnuSpecStd[day_, Dist_, E_] :=
  Module[{f1, f2, f3, f4, n1, n2, n3, n4, r1, r2, r3, r4, T},
   f1 = U235VsTime[day];
   f2 = Pu239VsTime[day];
   f3 = U238VsTime[day];
   f4 = Pu241VsTime[day];
   n1 = f1 * GWtoMeVperDay / EperFisU235;
   n2 = f2 * GWtoMeVperDay / EperFisP239;
   n3 = f3 * GWtoMeVperDay / EperFisU238;
   n4 = f4 * GWtoMeVperDay / EperFisP241;
   r1 = n1 * NusU235[E];
   r2 = n2 * NusP239[E];
   r3 = n3 * NusU238[E];
   r4 = n4 * NusP241[E];
   T = (r1 + r2 + r3 + r4) / (4 * Pi * Dist^2);
   Return[T]];
(* This is just the compiled version of the above *)
AnuSpec = Compile[{{day, _Real}, {Dist, _Real}, {E, _Real}},
   Module[{f1, f2, f3, f4, n1, n2, n3, n4, r1, r2, r3, r4, T},
    f1 = U235VsTime[day];
    f2 = Pu239VsTime[day];
    f3 = U238VsTime[day];
    f4 = Pu241VsTime[day];
    n1 = f1 * GWtoMeVperDay / EperFisU235;
    n2 = f2 * GWtoMeVperDay / EperFisP239;
    n3 = f3 * GWtoMeVperDay / EperFisU238;
    n4 = f4 * GWtoMeVperDay / EperFisP241;
    r1 = n1 * NusU235[E];
    r2 = n2 * NusP239[E];
    r3 = n3 * NusU238[E];
    r4 = n4 * NusP241[E];
    T = (r1 + r2 + r3 + r4) / (4 * Pi * Dist^2);
    Return[T]]];
```

Let's check the absolute normalization. This is per second per GW. The number is somewhat on the low side here. But recall that threshold is 1.8 MeV.

```
In[•]:= SecondsPerDay = 24 * 3600;
        Dist = 1; (* set distance to be 1 m*)
        TotalNumberOfNus =
          (1 / SecondsPerDay) * (4 * Pi * Dist^2) * NIntegrate[AnuSpecStd[1, Dist, x], {x, 0, 12}]
Out[•]= 1.61358 × 10^20
```

```
In[•]:= plt1 =  Plot[AnuSpec[1, 1000, x], {x, 0, 12}, PlotTheme → Scientific,
          FrameLabel → {"Energy (MeV)", "Flux per day per MeV per m^2 at 1 km for GWt"},
          FrameStyle → Bold, LabelStyle → {Bold, Medium, Italic}, PlotStyle → Thick]
```

Out[•]=



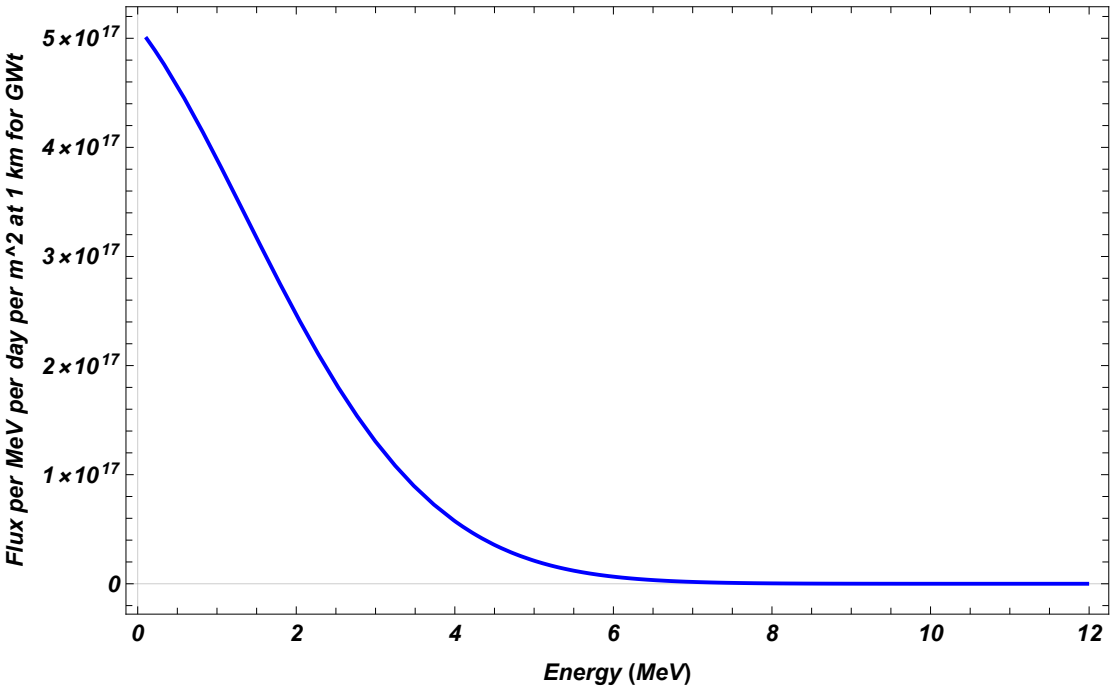
Now we will read a tabulated flux model. The energy is in GeV, and the flux is normalized per GeV in the file and extra factor of 5*10-6 for 1 km match. We will change this as we read it in. (you will need to change the location of the file Reactor.dat when you use it)

The final units are per MeV per day per m^2 at 1 km for 1 GWt

```
In[•]:= FluxData =
          ReadList["/Users/diwan/Documents/Projects/basic-reactor-calc/Reactor.dat",
           {Number, Number, Number, Number, Number, Number, Number}];
        Earray = Table[FluxData[[i]][[1]] * 1000, {i, 1, 501}]; (* convert to MeV *)
        Anuearray = Table[FluxData[[i]][[5]] / 1000 / (5 * 10^-6), {i, 1, 501}];
        (* convert to per MeV *)
        AnueFunc = Interpolation[Table[{Earray[[i]], Anuearray[[i]]}, {i, 1, 501}]];
```

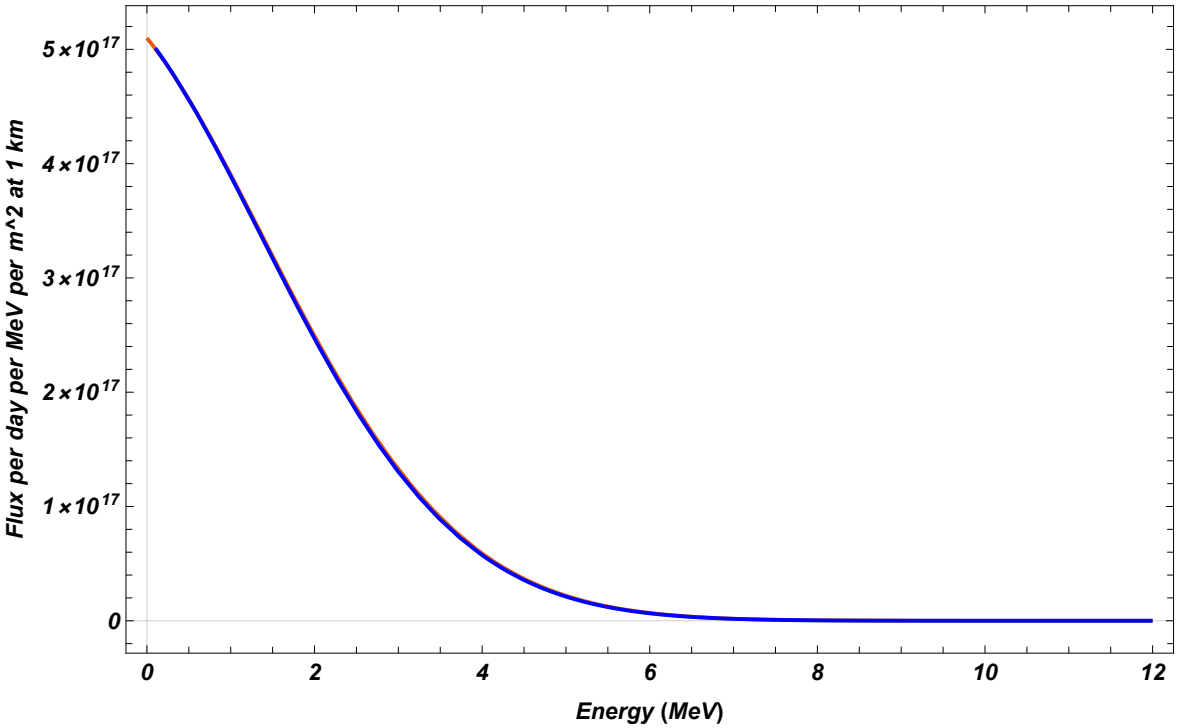
In[•]:=
```
plt2 = Plot[AnueFunc[x], {x, 0.1, 12.0}, PlotTheme → Scientific,
  FrameLabel → {"Energy (MeV)", "Flux per MeV per day per m^2 at 1 km for GWt"},
  FrameStyle → Bold, LabelStyle → {Bold, Medium, Italic}, PlotStyle → {Blue, Thick}]
```

Out[•]=




```
Show[plt1, plt2] (* these two are a close match *)
```

Out[•]=



We also have tabulated spectrum as detected in the Daya Bay experiment

without oscillations. This has the cross section built in. It is empirical and therefore has features that are not exponential. These features are real due to the numerous beta decay contributions. This spectrum is not normalized for any distance, detector or reactor power. One can choose to normalize it for 1 event as we have done for the plot.

In[•]:=
```
FluxData2 = ReadList[
   "/Users/diwan/Documents/lbn_review/Review/LBN_review/plot/reactor_spectrum.txt"
, {Number, Number}]; (* This is detected spectra according to Daya Bay model *)
Earray2 = Table[FluxData2[[i]][[1]], {i, 1, 1200}];
Anuearray2 = Table[FluxData2[[i]][[2]], {i, 1, 1200}];
AnueFunc2 = Interpolation[
   Table[{Earray2[[i]], Anuearray2[[i]]}, {i, 1, 1200}], InterpolationOrder → 8];
```

In[•]:=
```
Totalanue2 = NIntegrate[AnueFunc2[x], {x, 1.5, 11}];
Plot[AnueFunc2[x] / Totalanue2, {x, 0.1, 12}, PlotRange → {{0, 12}, {0, 0.4}},
 Frame → True, FrameStyle → Bold, FrameLabel → {"Energy (MeV)", "Events/MeV"},
 LabelStyle → {Bold, Medium, Italic}, Epilog -> Inset["Daya Bay Model", {8, 0.3}]]
```

Out[•]=

