# Data Import

Set this to wherever this file is.

In[9]:=
```
SetDirectory["~/Library/AFS@Reed/emcmanis/Thesis/Calculations/"]
```

Out[9]=
```
/afs/reed.edu/user/e/m/emcmanis/Thesis/Calculations
```

These files contain the data from the numerical runs.

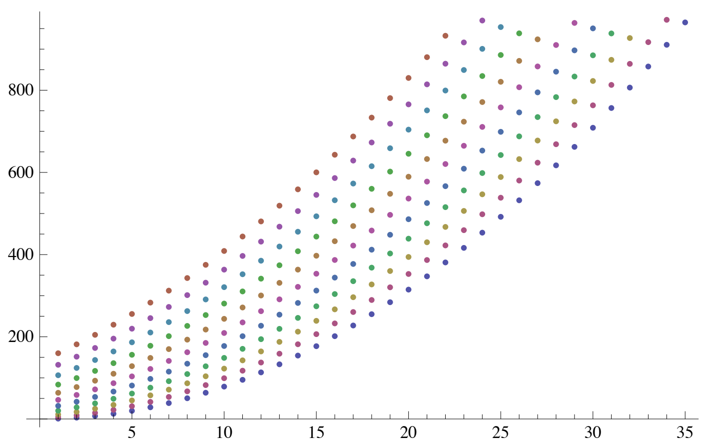
In[10]:=
```
l0 = Import["l0.csv"];
l1 = Import["l1.csv"];
l2 = Import["l2.csv"];
l3 = Import["l3.csv"];
l4 = Import["l4.csv"];
l5 = Import["l5.csv"];
l6 = Import["l6.csv"];
l7 = Import["l7.csv"];
l8 = Import["l8.csv"];
l9 = Import["l9.csv"];
l10 = Import["l10.csv"];
```

In[21]:=
```
ls = {l0, l1, l2, l3, l4, l5, l6, l7, l8, l9, l10};
```

In[22]:=
```
lsplot = Table[Table[{j, ls[[i, j, 1]]}, {j, 1, Length[ls[[i]]]}], {i, 1, Length[ls]}];
```

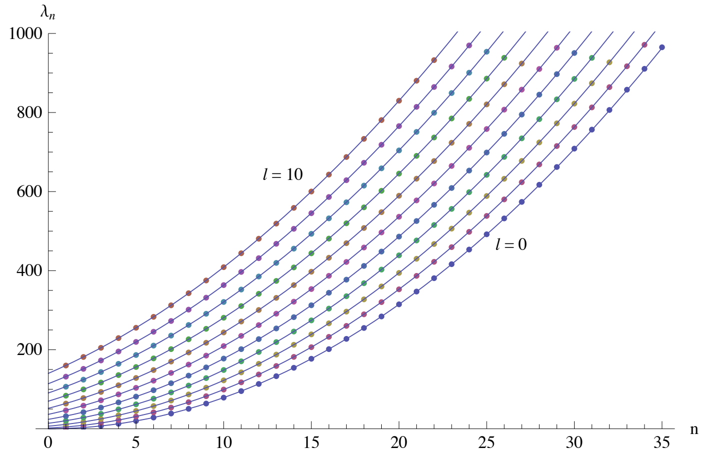
In[23]:=
```
ListPlot[lsplot]
```

Out[23]=




List plot with fits! This requires running some of the lower sections first, but it makes a nice figure up here.

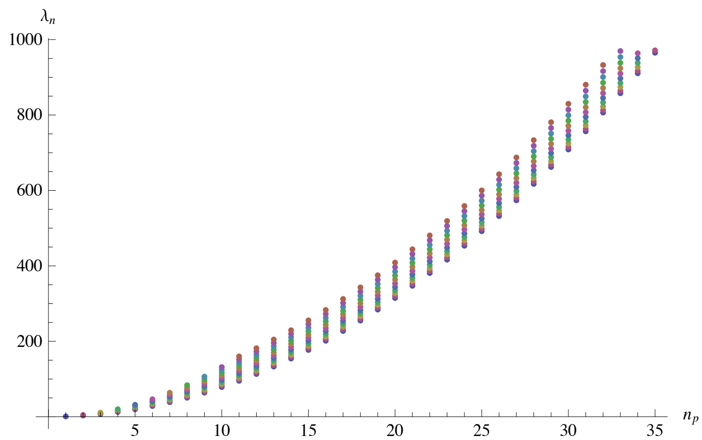
```
Show[ListPlot[lsplot, PlotRange → {0, 1000}, AxesLabel → {"n", "λn"}],
 Plot[Table[(a + b x + c x^2) /. Fits[[i]], {i, 1, Length[Fits]}], {x, 0, 35}]]
```




This plot is for $n_p$.

```
lsplot2 = Table[Table[{j + i - 1, ls[[i, j, 1]]}, {j, 1, Length[ls[[i]]]}], {i, 1, Length[ls]}];
```

```
ListPlot[lsplot2, AxesLabel → {"n_p", "λ_n"}]
```




# Least Squares Fit

- **Weighted linear least-squares fit (ultimately unsuccessful)**
- **Polynomial least-squares fit functions**

Implementation of a polynomial least squares fit. Appendix A of the thesis has more information on the theory behind this.

The PolyFit functions take in a list of the form $\{\{n, \lambda_n\} \ldots\}$, and the WeightedPolyFit functions expect a list of the form $\{\{\lambda_n, \text{err}\} \ldots\}$

```
In[2]:= PolyFit[lambdas_] := Module[{M, len, res},
   len = Length[lambdas];
   M =
    (                 len                      Sum[lambdas[[i, 1]], {i, 1, len}]   Sum[lambdas[[i, 1]]
     Sum[lambdas[[i, 1]], {i, 1, len}]   Sum[lambdas[[i, 1]]^2, {i, 1, len}]   Sum[lambdas[[i, 1]]
     Sum[lambdas[[i, 1]]^2, {i, 1, len}]   Sum[lambdas[[i, 1]]^3, {i, 1, len}]   Sum[lambdas[[i, 1]]

     );
   res = Inverse[M].{Sum[lambdas[[i, 2]], {i, 1, len}],
      Sum[lambdas[[i, 1]] * lambdas[[i, 2]], {i, 1, len}],
      Sum[lambdas[[i, 1]]^2 * lambdas[[i, 2]], {i, 1, len}]};
   Return[{a → res[[1]], b → res[[2]], c → res[[3]]}]
  ]

In[3]:= WeightedPolyFit[lambdas_] := Module[{M, len, res},
   len = Length[lambdas];
   M =
    (Sum[1/lambdas[[i,2]]^2, {i, 1, len}]   Sum[i/lambdas[[i,2]]^2, {i, 1, len}]   Sum[i^2/lambdas[[i,2]]^2, {i, 1, le
     Sum[i/lambdas[[i,2]]^2, {i, 1, len}]   Sum[i^2/lambdas[[i,2]]^2, {i, 1, len}]   Sum[i^3/lambdas[[i,2]]^2, {i, 1, le
     Sum[i^2/lambdas[[i,2]]^2, {i, 1, len}]   Sum[i^3/lambdas[[i,2]]^2, {i, 1, len}]   Sum[i^4/lambdas[[i,2]]^2, {i, 1, le

     );
   res = Inverse[M].{Sum[lambdas[[i, 1]]/lambdas[[i, 2]]^2, {i, 1, len}],
      Sum[i * lambdas[[i, 1]]/lambdas[[i, 2]]^2, {i, 1, len}],
      Sum[i^2 * lambdas[[i, 1]]/lambdas[[i, 2]]^2, {i, 1, len}]};
   Return[{a → res[[1]], b → res[[2]], c → res[[3]]}]
  ]

In[4]:= DeltaYs[lambdas_, fit_] := Sqrt[1/(Length[lambdas] - 3)
    Sum[(lambdas[[i, 2]] - (a + b lambdas[[i, 1]] + c lambdas[[i, 1]]^2) /. fit)^2,
     {i, 1, Length[lambdas]}]]
```

```
In[5]:= DeltaABCs[lambdas_, fit_, dy_] := Module[{M, len, res},
   len = Length[lambdas];
   M =
    (                len                      Sum[lambdas[[i, 1]], {i, 1, len}]  Sum[lambdas[[i, 1]]
     Sum[lambdas[[i, 1]], {i, 1, len}]  Sum[lambdas[[i, 1]]^2, {i, 1, len}]  Sum[lambdas[[i, 1]]
     Sum[lambdas[[i, 1]]^2, {i, 1, len}]  Sum[lambdas[[i, 1]]^3, {i, 1, len}]  Sum[lambdas[[i, 1]]
    );
   res = Sqrt[Sum[(Inverse[M].{1,
           lambdas[[i, 1]],
           lambdas[[i, 1]]^2} * dy) ^2, {i, 1, len}]];
   Return[res]
  ]

In[6]:= WeightedDeltaABCs[lambdas_] := Module[{M, len, res},
   len = Length[lambdas];
   M =
    ( Sum[1/lambdas[[i,2]]^2, {i, 1, len}]  Sum[i/lambdas[[i,2]]^2, {i, 1, len}]  Sum[i^2/lambdas[[i,2]]^2, {i, 1, le
      Sum[i/lambdas[[i,2]]^2, {i, 1, len}]  Sum[i^2/lambdas[[i,2]]^2, {i, 1, len}]  Sum[i^3/lambdas[[i,2]]^2, {i, 1, le
      Sum[i^2/lambdas[[i,2]]^2, {i, 1, len}]  Sum[i^3/lambdas[[i,2]]^2, {i, 1, len}]  Sum[i^4/lambdas[[i,2]]^2, {i, 1, le
    );
   res = Table[Sqrt[Inverse[M][[i, i]]], {i, 1, 3}];
   Return[res]
  ]
```

## ■ Polynomial fits and analysis

```
In[24]:= Fits = Table[PolyFit[lsplot[[i]]], {i, 2, Length[lsplot]}];

In[25]:= ftest = WeightedPolyFit[l0];
```

Different data collection methods meant using different fit types for the $l = 0$ data and the rest.

```
In[26]:= Fits = Join[{ftest}, Fits];

In[27]:= dys = Table[DeltaYs[lsplot[[i]], Fits[[i]]], {i, 2, Length[Fits]}];

In[28]:= dys = Join[{{}}, dys];
```

In[29]:=
```
TableForm[
 Table[Join[{"l = " <> ToString[i - 1]}, Fits[[i]], {dys[[i]]}], {i, 1, Length[Fits]}]]
```

Out[29]//TableForm=

| | | | | |
|---|---|---|---|---|
| l = 0 | a → 0.0437591 | b → 0.0180773 | c → 0.78578 | |
| l = 1 | a → 2.08389 | b → 1.87181 | c → 0.783359 | 0.061561 |
| l = 2 | a → 6.69914 | b → 3.7623 | c → 0.780989 | 0.0999442 |
| l = 3 | a → 13.9713 | b → 5.66987 | c → 0.778668 | 0.126059 |
| l = 4 | a → 23.916 | b → 7.58959 | c → 0.776349 | 0.141452 |
| l = 5 | a → 36.5484 | b → 9.5174 | c → 0.774072 | 0.148596 |
| l = 6 | a → 51.8423 | b → 11.4602 | c → 0.77144 | 0.140831 |
| l = 7 | a → 69.8763 | b → 13.3975 | c → 0.769258 | 0.136714 |
| l = 8 | a → 90.6213 | b → 15.3367 | c → 0.767164 | 0.129699 |
| l = 9 | a → 114.081 | b → 17.2771 | c → 0.765162 | 0.12079 |
| l = 10 | a → 140.221 | b → 19.2291 | c → 0.762649 | 0.101198 |

Goodness of fit for $l = 0$:

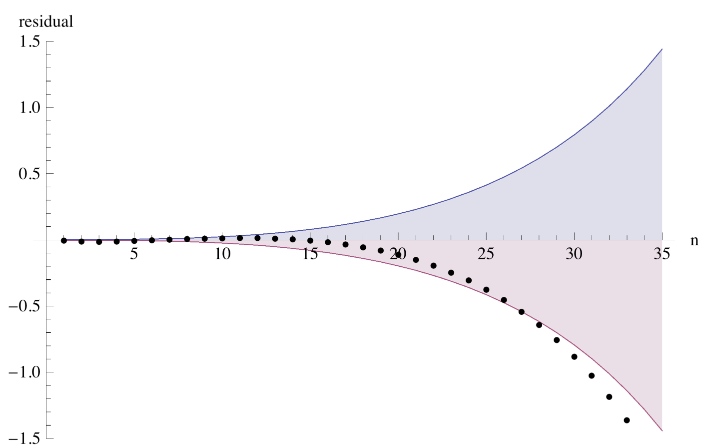
In[30]:=
```
G[x_] := c x^2 + b x + a /. Fits[[1]]
```

In[31]:=
```
quadfitvals = Table[G[i], {i, 1, Length[l0]}];
```

In[32]:=
```
quadresiduals = Table[quadfitvals[[i]] - l0[[i, 1]], {i, 1, Length[l0]}];
```

In[33]:=
```
Show[
 ListPlot[{Table[l0[[i, 2]], {i, 1, Length[l0]}], Table[-l0[[i, 2]], {i, 1, Length[l0]}]},
  Joined → True, Filling → Axis, AxesLabel → {"n", "residual"}],
 ListPlot[quadresiduals, PlotStyle → Black, PlotMarkers → {Automatic, 5}]]
```

Out[33]=




This fit is much better than the linear one! Additional info:

```
dabcs = Table[DeltaABCs[lsplot[[i]], Fits[[i]], dys[[i]]], {i, 2, Length[lsplot]}]
```

```
{{0.0336319, 0.00443055, 0.000122796}, {0.0564978, 0.0078931, 0.00023205},
 {0.0725534, 0.0104523, 0.000316912}, {0.0829445, 0.0123341, 0.000386055},
 {0.0888369, 0.0136497, 0.000441511}, {0.0877272, 0.01444, 0.000500517},
 {0.0870456, 0.0148573, 0.000534104}, {0.0844883, 0.0149742, 0.000559067},
 {0.0805912, 0.0148535, 0.000576805}, {0.0710914, 0.0142391, 0.000601176}}
```

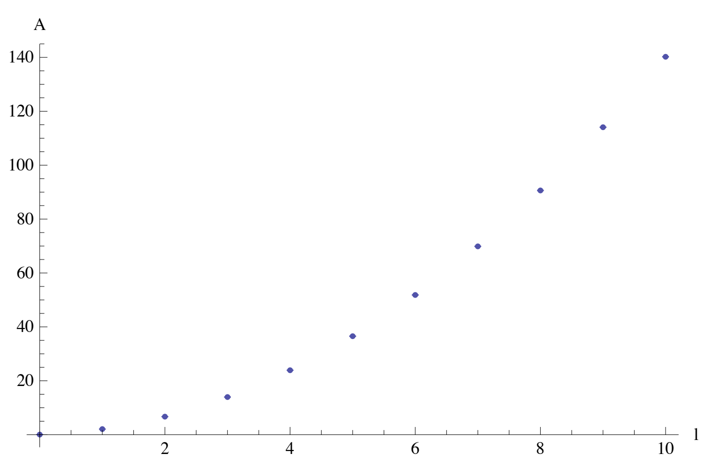
```
dabcs = Join[{WeightedDeltaABCs[l0]}, dabcs]
```

```
{{0.0028406, 0.00145906, 0.000149063},
 {0.0336319, 0.00443055, 0.000122796}, {0.0564978, 0.0078931, 0.00023205},
 {0.0725534, 0.0104523, 0.000316912}, {0.0829445, 0.0123341, 0.000386055},
 {0.0888369, 0.0136497, 0.000441511}, {0.0877272, 0.01444, 0.000500517},
 {0.0870456, 0.0148573, 0.000534104}, {0.0844883, 0.0149742, 0.000559067},
 {0.0805912, 0.0148535, 0.000576805}, {0.0710914, 0.0142391, 0.000601176}}
```

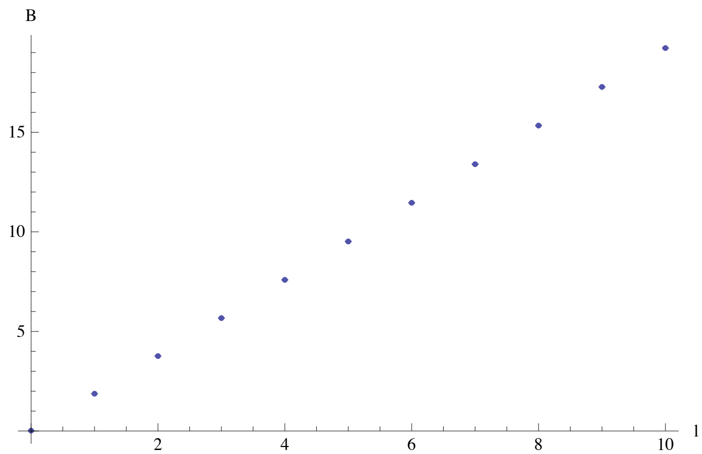
```
dabcs // TableForm
```

| | | |
|---|---|---|
| 0.0028406 | 0.00145906 | 0.000149063 |
| 0.0336319 | 0.00443055 | 0.000122796 |
| 0.0564978 | 0.0078931 | 0.00023205 |
| 0.0725534 | 0.0104523 | 0.000316912 |
| 0.0829445 | 0.0123341 | 0.000386055 |
| 0.0888369 | 0.0136497 | 0.000441511 |
| 0.0877272 | 0.01444 | 0.000500517 |
| 0.0870456 | 0.0148573 | 0.000534104 |
| 0.0844883 | 0.0149742 | 0.000559067 |
| 0.0805912 | 0.0148535 | 0.000576805 |
| 0.0710914 | 0.0142391 | 0.000601176 |

```
Needs["ErrorBarPlots`"]

as = Table[{i - 1, a /. Fits[[i, 1]]}, {i, 1, Length[Fits]}]

{{0, 0.0437591}, {1, 2.08389}, {2, 6.69914}, {3, 13.9713}, {4, 23.916},
 {5, 36.5484}, {6, 51.8423}, {7, 69.8763}, {8, 90.6213}, {9, 114.081}, {10, 140.221}}

aserr = Table[{{i - 1, a /. Fits[[i, 1]]}, ErrorBar[dabcs[[i, 1]]]}, {i, 1, Length[Fits]}];

bs = Table[{i - 1, b /. Fits[[i, 2]]}, {i, 1, Length[Fits]}]

{{0, 0.0180773}, {1, 1.87181}, {2, 3.7623}, {3, 5.66987}, {4, 7.58959},
 {5, 9.5174}, {6, 11.4602}, {7, 13.3975}, {8, 15.3367}, {9, 17.2771}, {10, 19.2291}}

bserr = Table[{{i - 1, b /. Fits[[i, 2]]}, ErrorBar[dabcs[[i, 2]]]}, {i, 1, Length[Fits]}];

cs = Table[{i - 1, c /. Fits[[i, 3]]}, {i, 1, Length[Fits]}]

{{0, 0.78578}, {1, 0.783359}, {2, 0.780989}, {3, 0.778668}, {4, 0.776349}, {5, 0.774072},
 {6, 0.77144}, {7, 0.769258}, {8, 0.767164}, {9, 0.765162}, {10, 0.762649}}

cserr = Table[{{i - 1, c /. Fits[[i, 3]]}, ErrorBar[dabcs[[i, 3]]]}, {i, 1, Length[Fits]}];

ErrorListPlot[aserr, AxesLabel → {"l", "A"}]
```



```
ErrorListPlot[bserr, AxesLabel → {"l", "B"}]
```




```
ErrorListPlot[cserr, AxesLabel → {"l", "C"}]
```