Binary Degraders

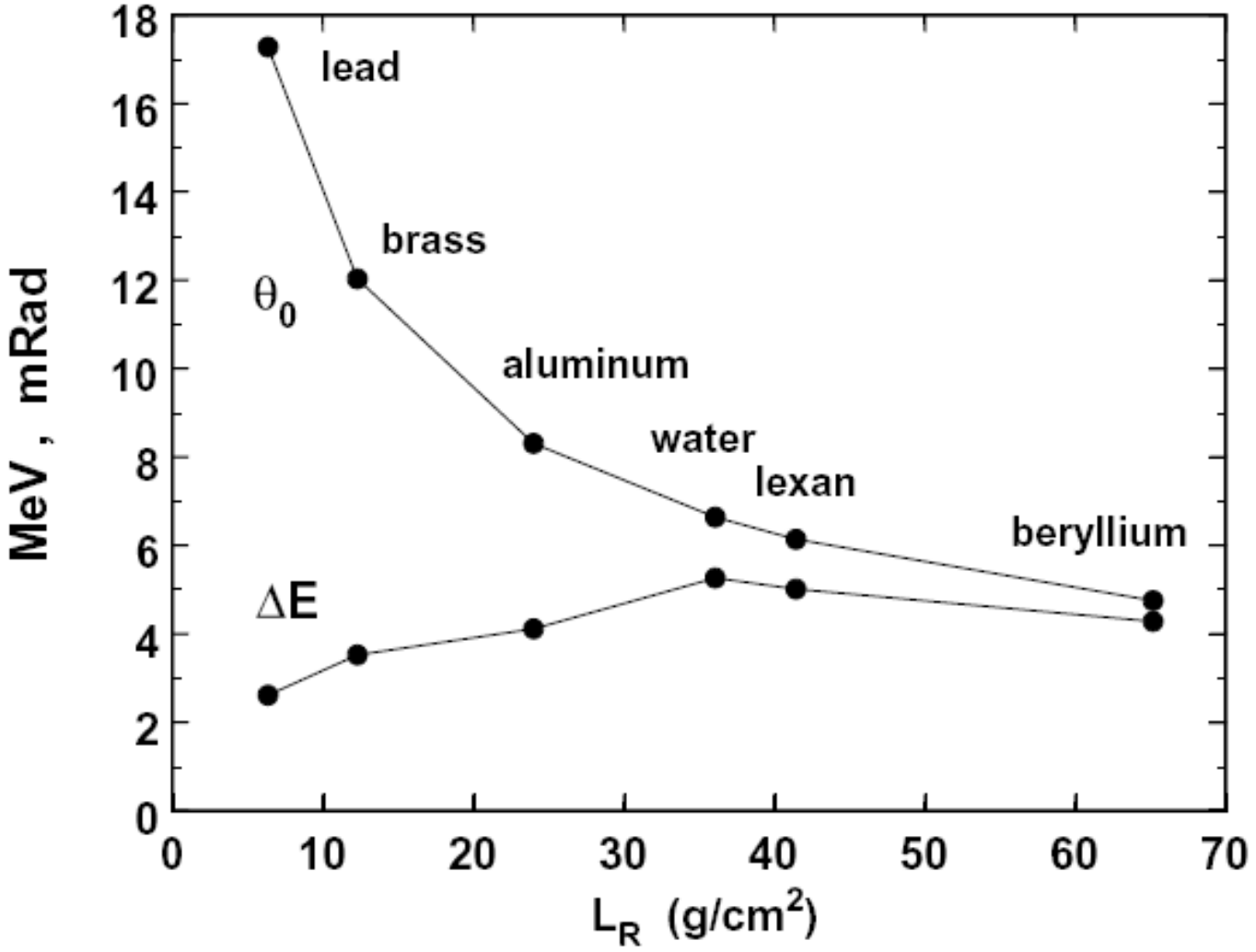
Often we want to control both scattering and energy loss in a beam line element. For instance, we might want a contoured scatterer with an energy loss independent of radius, or a modulator with scattering angle independent of modulator step (thickness).

We call such devices `compensated' and one could make them by scouring the periodic table for an appropriate material. That is usually impractical. Instead, we combine high-Z and low-Z materials (e.g. lead and plastic) in 'sandwiches' having the correct ratio of materials.

Finding that combination (if it exists) is basically trial-and-error, tedious by hand but relatively trivial on a computer if we use some standard 'root finder' technique.

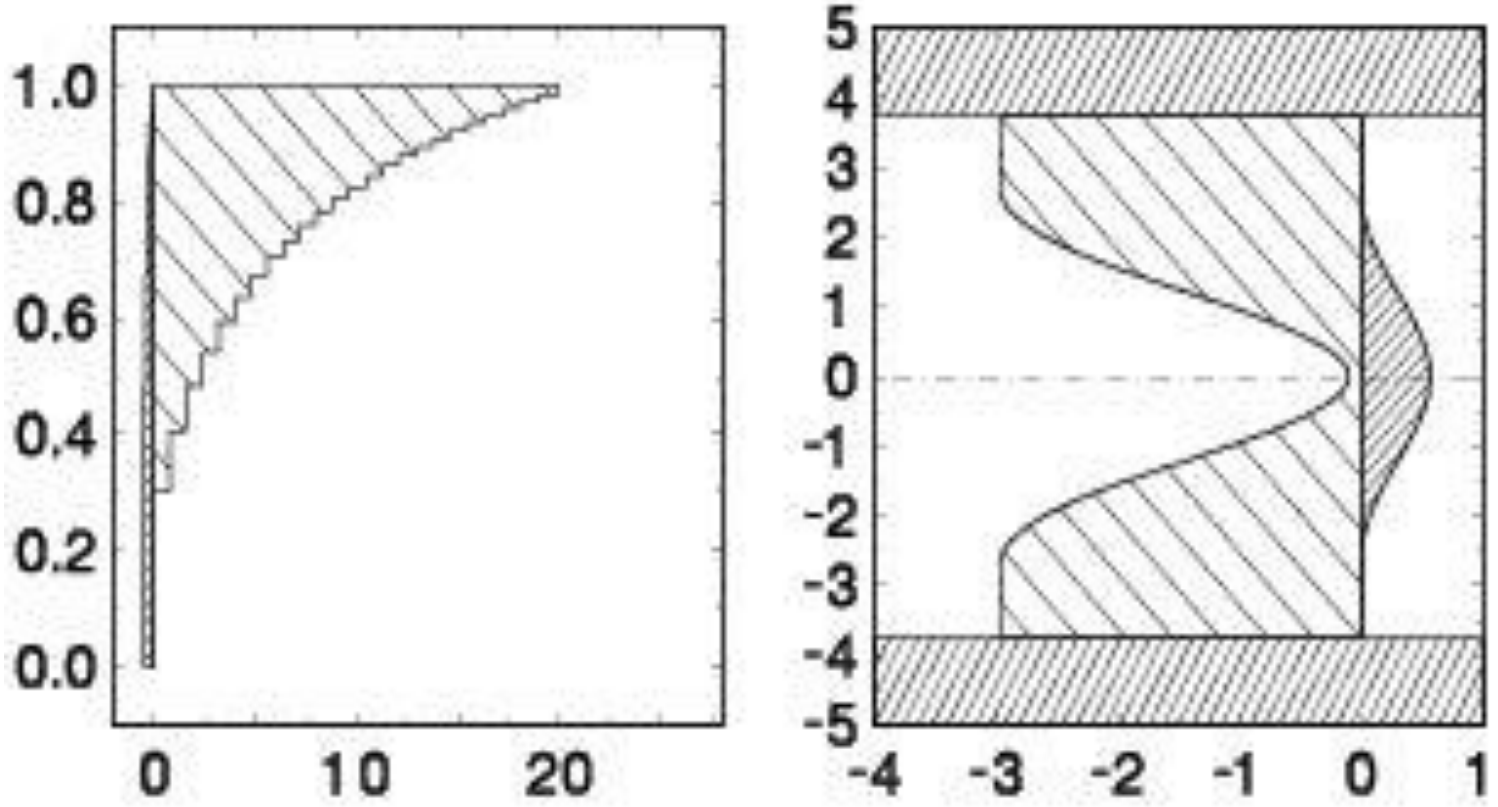
BGware has subprograms to solve such problems in beam line design. **LOOKUP** is a convenient user interface for by-hand design. This lecture explains how these programs work.

Ingredients



Energy loss and multiple scattering for 160 MeV protons incident on 1 g/cm^2 of various materials. $L_R \equiv$ radiation length, a property of the material which you can look up. Use lead (Pb) for scattering, plastic or beryllium for degrading.

Examples



Left: compensated modulator, principally of plastic. Energy loss increases with step # but multiple scattering angle remains constant thanks to the layer of lead. Right: compensated contoured scatterer, principally of lead. Scattering decreases with radius but total energy loss remains constant thanks to the plastic. The lead/plastic mix must respect both the required energy loss and the required scattering angle in both designs. The scatterer shows failure of compensation at large radius. We can tolerate it in this case.

Software Building Blocks

Assume we have four functions at our disposal :

```
Range(energy, material)
Energy(range, material)
Dedx(energy, material)
Theta0(mode, target thickness, outgoing energy, material)
```

where

material	a code e.g. 'WATER' or 'polystyrene'
mode	a code e.g. 'HIGHLAND' or 'MOLIERE'
energy	MeV
range	g/cm^2
thickness	g/cm^2

Abbreviated Notation for Problems

Let

a	denote	scattering angle (rad)
g		thickness (g/cm^2)
m		material
t		kinetic energy (MeV)
1		'entering the first element'
2		'entering the second element' ...

Then

$$a1,\ a2,\ t2 \ \rightarrow\ g1,\ t1$$

denotes the *single degrader* problem 'Given the incoming and outgoing angles and the outgoing energy, find the degrader thickness and the incoming energy.' *Materials are always given.* Similarly,

$$a1,\ a3,\ t1,\ t3 \ \rightarrow\ g1,\ g2,\ t2$$

denotes the *binary degrader* problem 'Given the incoming and outgoing energies and angles, find the required thicknesses and the energy at the interface.' This is the binary problem most commonly encountered in beam line design.

Solved Problems

Module BDG.FOR has subroutines for solving two single degrader problems:

```
1   a1,a2,t2   ->   g1,t1
2   a1,a2,t1   ->   g1,t2
```

and five binary degrader problems:

```
1   (a1,a3) t1,t3   ->   g1,g2 (t2)
2   (a1,a3) g1,t3   ->   g2,t1 (t2)
3   (a1,a3) g1,t1   ->   g2,t3 (t2)
4   (a1,a3) g2,t1   ->   g1,t3 (t2)
5   (a1,a3) g2,t3   ->   g1,t1 (t2)
```

All can be run by hand from LOOKUP, and all sometimes arise in the design of passive beam spreading systems !

Single Degrader Forward Problem

A 'forward' problem is one where all thicknesses are known:

$$m1,\ g1,\ t1,\ a1 \rightarrow t2,\ a2$$

The solution is straightforward:

1. Find the proton range in material $m1$ at energy $t1$.
2. Subtract $g1$ to obtain the residual range.
3. Find the energy in $m1$ for that range. That equals $t2$.
4. Use `Theta0` to find the scattering.
5. Add it to $a1$ in quadrature to find $a2$.

You can repeat this for as many degraders as you have. TOUT.FOR runs this procedure for a stack of degraders. TINP.FOR does it backwards.

Single Degrader Inverse Problem

In an *inverse* problem at least one thickness is unknown:

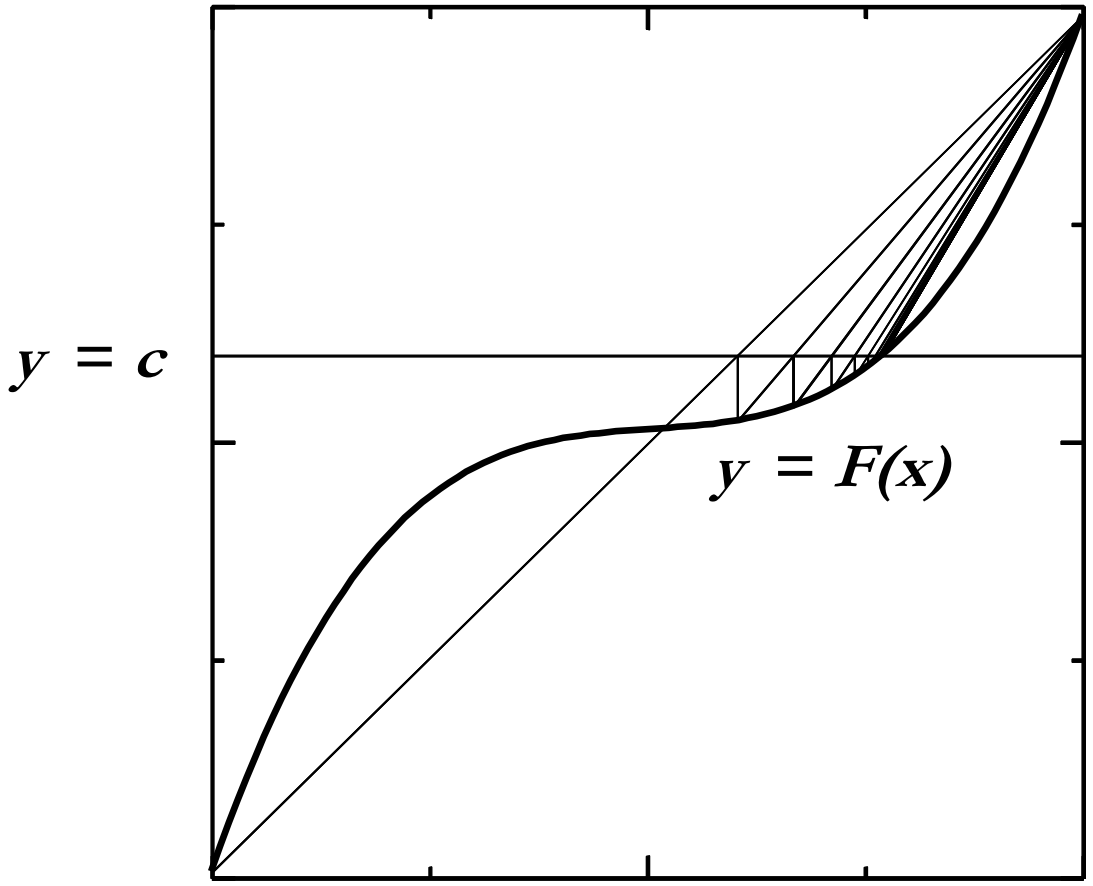
$$a1,\ a2,\ t1 \rightarrow g1,\ t2$$

We solve it by iteration (trial and error) on a related *forward* problem, after testing that a solution is possible. For instance, we can construct a function **a2 = F(g1...)** as follows. **g1** is called **x** here because it is the cardinal unknown.

```
rr = Range(t1,m1) - x
t2 = Energy(rr,m1)
F  = SQRT(a1**2 + Theta0(mode,x,t2,m1,bb)**2)
```

We now need to solve **F(x) – c = 0** where **c** is the desired value of **a2** and **F(x)** is not an algebraic expression, but a known numerical function. This is the standard numerical problem of finding the root of a function (see Numerical Methods). The ***Method of False Position*** (next slide) is often a good choice. Programming note: because the root finder iterates the whole sequence of statements, once it converges ancillary variables (such as **t2**) are already known. We don't have to recompute them.

Method of False Position (*Regula Falsa*)



Find ***x*** such that ***F(x) – c = 0*** . In this method (***Vaishali Ganit***, ca. 3rd century BC) a line is drawn between the last two points to bracket the root, and its intersection with ***y = c*** is used for the next bracket. The method is robust --- the root always stays bracketed --- and reasonably fast. We usually use it or the even more robust ***method of bisection***.

Binary Degrader Inverse Problem

The following inverse problem is common in beam line design:

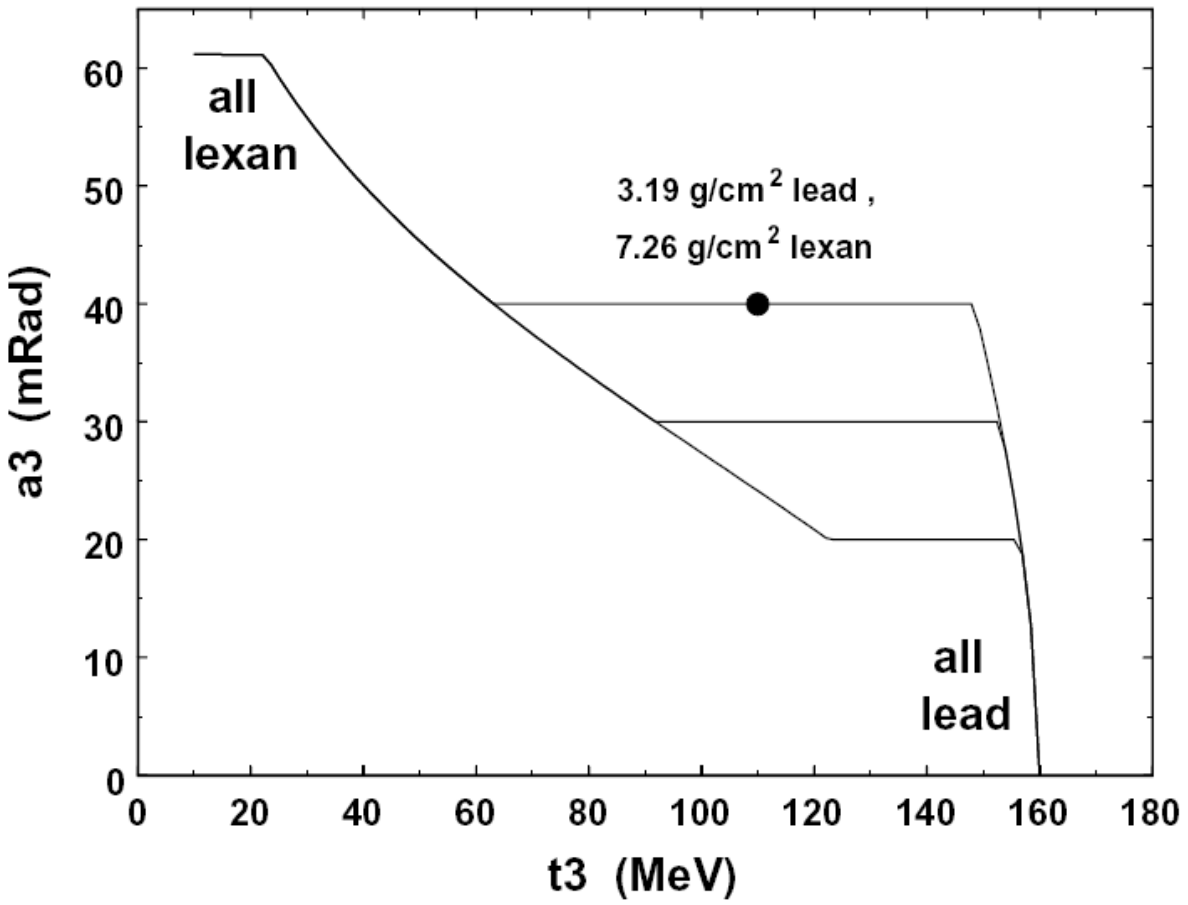
$$a1\ a3,\ t1,\ t3 \rightarrow g1,\ g2,\ t2$$

The solution is similar. Focusing on one of the unknowns, say t2 because its brackets are known, we write down the forward solution a3(t2), then use a root finder to locate the value of t2 that yields the desired a3.

We must check whether a solution exists and decide what to do if it does not. **Energy trumps: t1 and t3 will be satisfied even if a3 cannot be.** (In beam line design we usually need the correct energy, but we can often tolerate a small error in multiple scattering angle.)

Suppose 160 MeV protons enter a lead/Lexan degrader. We want to reduce the energy to 110 MeV while obtaining exactly 40 mrad of multiple scattering. The solution, and the range of energies over which a solution exists, are shown in the next slide.

Binary Degrader Problem (Graphical)



This graph shows the solution as well as the range of incoming and outgoing energies over which we can hold any given scattering angle a3. 'All Lexan' means we are asking for too much energy degradation given the desired angle; 'all lead' means we are asking for too little. Eventually the angle saturates at the maximum possible value for Lexan (see lecture on multiple scattering).

Does the Order of Materials Matter?

The order of materials makes a difference. If the lead is second it sees a lower energy and scatters more, so less lead is needed. In our example:

order	$t2$ MeV	lead g/cm^2	lexan g/cm^2
lead/lexan	151.6	3.186	7.261
lexan/lead	116.0	1.837	7.985

However, the range over which a solution exists does *not* change, because once the binary degrader is all one or the other, 'which comes first' is meaningless. We can't extend the range by changing the order! In our example the t3 range over which we can hold 40 mrad is 64 to 148 MeV.

Summary

High-Z elements scatter more and low-Z materials stop more. By combining e.g. lead and plastic we can (over a finite range) control both the multiple scattering and the energy loss in a beam line element.

It is convenient to categorize problems using a compact notation. Forward problems, where all thicknesses are known, are solved by applying subroutines Range, Energy and Theta0.

Inverse problems, where at least one thickness is to be found, are solved by focusing on one variable, writing down as a Fortran function the solution to the corresponding forward problem, and applying a root finder.

Module BDG.FOR comprises solutions to two single- and five binary-degrader problems. LOOKUP is a convenient user interface to this module for by-hand designs.

BDG.FOR is also used by our beam line design programs. The order of two materials affects how much of each material will be used, but cannot affect the range over which a solution satisfying all requirements exists. If no solution exists, energy trumps angle, and the program usually issues a 'failure in compensation' warning.