

1.

辐射长度

因轫致辐射而造成的平均能量损失率由下式给出^[1]

$$-\frac{dE}{dx} = 4\alpha N_A \frac{Z^2}{A} \left(\frac{1}{4\pi\epsilon_0} \frac{e^2}{mc^2} \right)^2 E \ln \frac{183}{Z^{1/3}} \quad (2.1.19)$$

Z 和 A 为物质的原子序数和原子量, z 、 m 和 E 分别为入射粒子的电荷量(以电子电荷 e 为单位)、质量和能量, α 为精细结构常数。对于电子, 其轫致辐射能损可由式(2.1.19)推得

$$-\frac{dE}{dx} = 4\alpha N_A \frac{Z^2}{A} r_e^2 E \ln \frac{183}{Z^{1/3}} \quad (2.1.20)$$

式(2.1.20)要求 $E \gg m_e c^2 / \alpha Z^{1/3}$ 成立。

由于电子是最轻的带电粒子, 其轫致辐射能损非常显著。将式(2.1.20)改写为以下形式

$$-\frac{dE}{dx} = \frac{E}{X_0} \quad (2.1.21)$$

此式定义了一个新的物理量 X_0 , 称为辐射长度, 它表示的是一个高能电子通过轫致辐射能量损失到 $1/e$ (此处 e 为自然对数常数) 所经过的平均路程。由式(2.1.20)和(2.1.21), 显然有

$$X_0 = \frac{A}{4\alpha N_A Z^2 r_e^2 \ln(183/Z^{1/3})} [\text{g/cm}^2] \quad (2.1.22)$$

辐射长度与物质原子序数的关系大致为 $X_0 \propto Z^{-2}$, 这来源于粒子与核的库仑场的相互作用。

计及这些因素后, 常用的经验公式是

$$X_0 = \frac{716.4 \cdot A}{Z(Z+1) \ln(287/\sqrt{Z})} [\text{g/cm}^2]$$

核作用强度

类似于光子束,可以根据强子束强度在物质中的衰减定义核相互作用长度 λ_I

$$N = N_0 e^{-x/\lambda_I} \quad (2.3.2)$$

λ_I 可以通过强子截面的非弹性部分计算

$$\lambda_I = \frac{A}{N_A \rho \sigma_{\text{inelastic}}} [\text{cm}] \quad (2.3.3)$$

若以 g/cm^2 为单位,只需将式(2.3.3)乘以密度 ρ 即可。核碰撞长度 λ_T 则和总截面相关

电子能量损失

公式

$\gamma m_e c^2$, 取 $z = 1$, 则电子的能损公式可近似为

$$-\frac{dE}{dx} = K \frac{Z}{A} \frac{1}{\beta^2} \left[\ln \frac{\gamma m_e c^2}{2I} - \beta^2 - \frac{\delta^*}{2} \right]$$

由于电子是最轻的带电粒子,其轫致辐射能损非常显著。将式(2.1.20)改写为以下形式

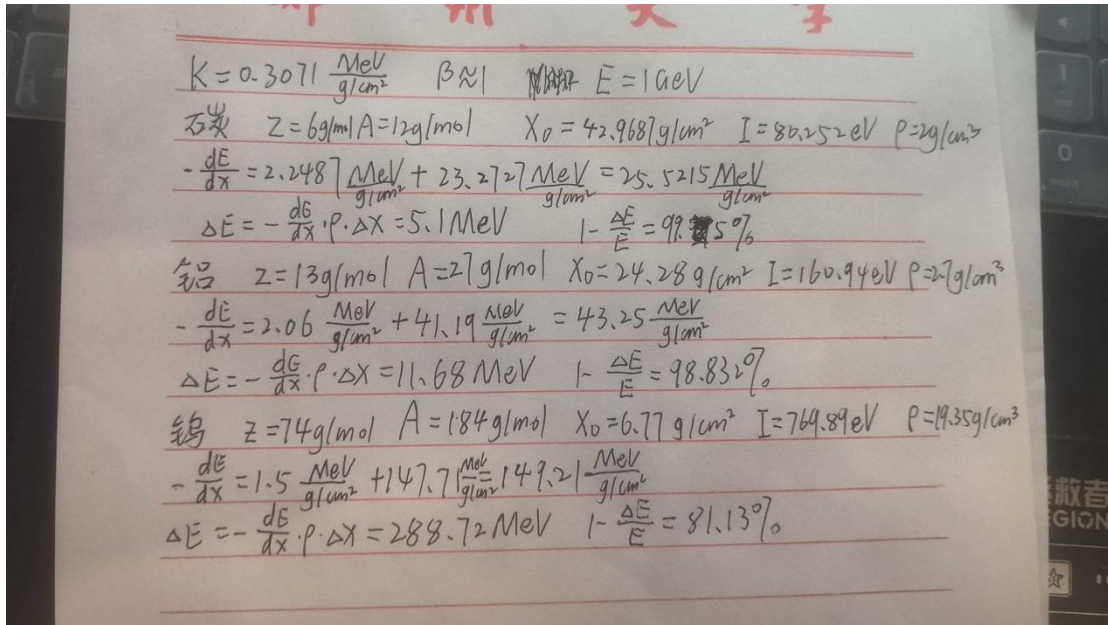
$$-\frac{dE}{dx} = \frac{E}{X_0} \quad (2.1.21)$$

计及这些因素后,常用的经验公式是

$$X_0 = \frac{716.4 \cdot A}{Z(Z+1) \ln(287/\sqrt{Z})} [\text{g/cm}^2] \quad (2.1.24)$$

I ——平均激发能,取决于所穿越物质的性质。可近似描述为 $I = 16Z^{0.9} \text{ eV} (Z > 1)$ 。 I 还与该物质所处的分子态有关,例如,原子态、分

碳 99.5%、铝 98.832%、钨 81.13%



散射角

其中 z 已取为 1。 θ_{plane} 考虑的是二维情况, 当需要考虑三维情况时, 有

$$\theta_{\text{space}}^{\text{rms}} = \sqrt{2} \theta_{\text{plane}}^{\text{rms}} = \frac{19.2}{\beta c p} \sqrt{\frac{x}{X_0}} \quad (2.1.18)$$

碳 $9.26\text{e-}4$ 、 $9.26\text{e-}5$ 、 $9.26\text{e-}6$

铝 $1.23\text{e-}3$ 、 $1.23\text{e-}4$ 、 $1.23\text{e-}5$

钨 $2.33\text{e-}3$ 、 $2.33\text{e-}4$ 、 $2.33\text{e-}5$

厚薄不同

薄: 朗道分布

厚：高斯分布

原因：

Bethe - Bloch 公式给出的是能量损失的平均值,当粒子穿越薄的吸收体层时,由于相互作用次数少,能量损失的统计涨落很大。在薄层物质,例如气体中,能

2.

泊松分布：描述单位时间内随机事件发生次数的概率分布

$$P(X=k) = \lambda^k \cdot e^{-\lambda} / k!$$

泊松过程的特征

平稳性：在任意等长时间内，事件发生次数的分布相同

独立增量性：不相交时间段内的事件发生次数相互独立



普通性：在极短时间，发生两次及以上事件的概率很小

泊松分布与二项分布

二项分布

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

$n \rightarrow \infty$
 $p \rightarrow 0$



泊松分布与高斯分布

当 λ 较大时（通常 $\lambda > 10$ ），泊松分布近似于正态分布 $N(\lambda, \lambda)$

母函数

定义

对于有限或无限数列 $\{a_0, a_1, a_2, \dots\}$, 用形式幂级数 $f(x) = \sum_{n=0}^{\infty} a_n x^n$ 使之成为整体, 然后通过先求 $f(x)$, 再展开 $f(x)$ 算得 x^n 的系数求出 a_n . 我们称 $f(x)$ 为数列 $\{a_n\}$ 的**母函数 (生成函数)**.

概率母函数是专门用于描述非负整数值离散随机变量概率分布的工具, 它通过幂级数的形式“封装”了整个概率分布的信息。其定义为: 对于一个取值为非负整数 $(0, 1, 2, \dots)$ 的随机变量 X , 其概率母函数 (PGF) 为:

$$G(s) = E(s^X) = \sum_{k=0}^{\infty} P(X = k) \cdot s^k$$

性质

性质	描述	公式示例 (简要)
恢复概率分布	对 $G(s)$ 求导并令 $s=0$, 可得到具体的概率值。	$P(X=k) = k! G^{(k)}(0)$
计算数字特征	利用函数在 $s=1$ 处的导数, 可以方便地计算期望和方差等。	$E[X] = G'(1), D[X] = G''(1) + G'(1) - [G'(1)]^2$
处理独立随机变量和	独立随机变量之和的概率母函数, 等于各自概率母函数的乘积。	若 $Z=X+Y$ 且 X, Y 独立, 则 $G_Z(s) = G_X(s) \cdot G_Y(s)$
与分布一一对应	一个概率分布唯一确定一个概率母函数, 反之亦然。	-

$X \sim P(\lambda_1) \quad Y \sim P(\lambda_2) \quad Z = X + Y$
 $G_X(s) = \sum_{k=0}^{\infty} \frac{\lambda_1^k e^{-\lambda_1}}{k!} \cdot s^k = e^{-\lambda_1} \cdot e^{\lambda_1 s} = e^{\lambda_1(s-1)}$
 $G_Y(s) = \sum_{k=0}^{\infty} \frac{\lambda_2^k e^{-\lambda_2}}{k!} \cdot s^k = e^{-\lambda_2} \cdot e^{\lambda_2 s} = e^{\lambda_2(s-1)}$
 $G_Z(s) = G_X(s) \cdot G_Y(s) = e^{\lambda_1(s-1)} \cdot e^{\lambda_2(s-1)} = e^{(\lambda_1 + \lambda_2)(s-1)} = e^{-(\lambda_1 + \lambda_2)} \cdot e^{(\lambda_1 + \lambda_2)s} = \sum_{k=0}^{\infty} \frac{(\lambda_1 + \lambda_2)^k e^{-(\lambda_1 + \lambda_2)}}{k!}$
 $\therefore Z \sim P(\lambda_1 + \lambda_2)$

3.landao

```
MyStyle();
TCanvas *cvs = new TCanvas("canvas", "signal", 800, 600);
TFile *f=TFile::Open("result.root");
TH2D *h2=(TH2D*)f->Get("r_strip");
TH1D *h1=h2->ProjectionY("",1,5);
h1->GetXaxis()->SetTitle("x");
h1->GetYaxis()->SetTitle("f(x)");
TF1 *fun=new TF1("fit","landau",h1->GetXaxis()->GetXmin(),h1->GetXaxis()->GetXmax());
fun->SetParameter(0, h1->GetMaximum());
fun->SetParameter(1, h1->GetMean());
fun->SetParameter(2, h1->GetRMS()/2);
h1->Fit("fit","R");
h1->Draw("HIST");
fun->Draw("SAME");
cvs->Update();
cin.get();
delete cvs;
f->Close();
```

