

```
[ > restart;
> with(SharedFuncs);
[ DoB, DoP, K, NsB, NsP, WB, WP, fCiB, fCiP, fClimB, fClimP, fDB, fDP, fNB, fNP, fQB, fQP]
```

Кремний, двухстадийная диффузия бора

1. Дано в задании:

Tz - температура загонки, С

Tr - температура разгонки, С

tz, tr - время загонки и время разгонки, часов

ro - удельное сопротивление исходного материала при комнатной температуре, Ом*см²

step - шаг расчёта, см, здесь не применяется

depth - глубина расчёта, см

```
> Tz := 1100; Tr := 1200; tz := 0.8; tr := 2.9; ro := 6.5; step := 0.5*1e-4;
depth := 15*1e-4;
```

$T_z := 1100$

$T_r := 1200$

$t_z := .8$

$t_r := 2.9$

$ro := 6.5$

$step := .00005$

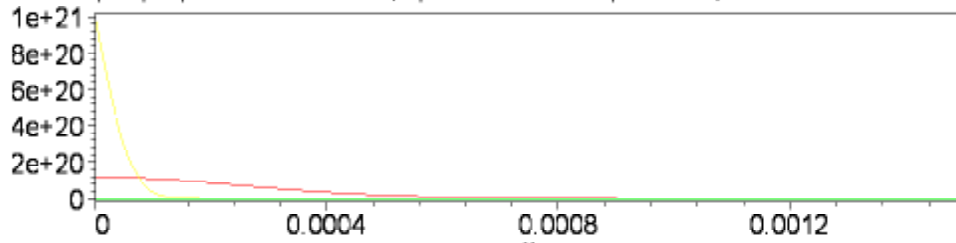
$depth := .0015$

```
> Q := evalf(2*NsB*sqrt(fDB(Tz)*tz*3600/Pi));
```

$Q := .3762668911 \cdot 10^{17}$

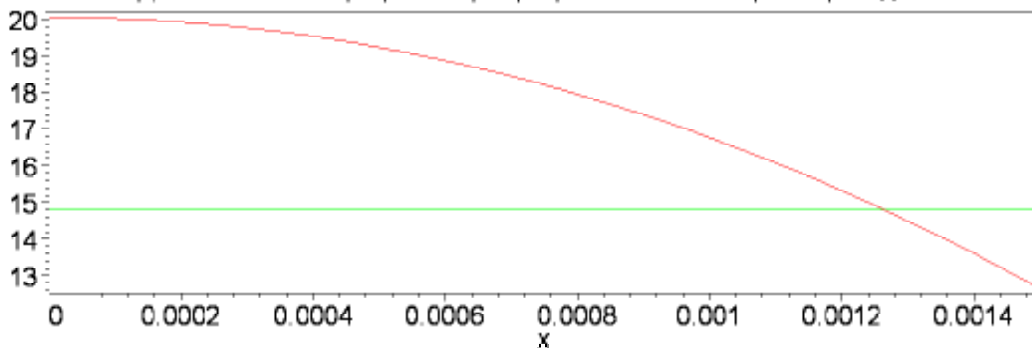
```
> plot([fNP(ro), fCiB(Tz, x, tz), fClimB(Tr, Q, x, tr)], x=0..depth, color=[green, yellow, red],
axes=box, title="Жёлтый - концентрация после загонки, красный - после разгонки, зелёный - начальная бора");
```

Жёлтый - концентрация после загонки, красный - после разгонки, зелёный - начальная бора



```
> plot([log10(fClimB(Tr, Q, x, tr)), log10(fNP(ro))], x=0..depth, color=[red, green], axes
=box, title="Десятичный логарифм концентраций - положение p-n перехода");
```

Десятичный логарифм концентраций - положение p-n перехода



```
>
```