Diffusion of Titanium Oxide into Sapphire – ORC

1 Introduction

The ORC group diffuse titanium sesqui-oxide, Ti_2O_3, into sapphire in order to produce desired optical properties in the doped sapphire. This is achieved by placing a layer of Ti_2O_3 on the surface of the sapphire and then baking the system in a furnace. The aim is to achieve a doped region in the sapphire as shown on the left of Figure 1; however in practice the doped region resembles that shown on the right of the figure.

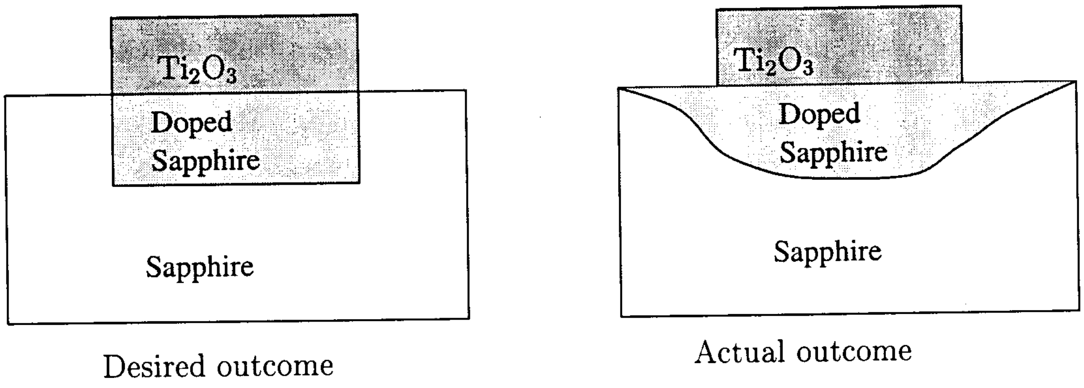
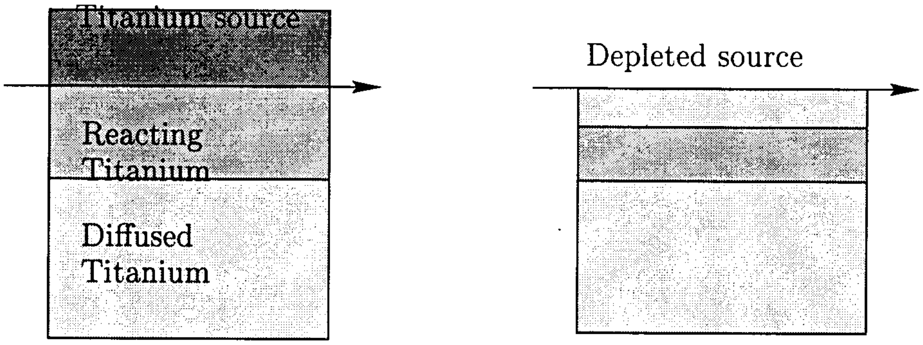


Figure 1: Geometry of the sapphire doping process

As well as bulk diffusion of Ti_2O_3 into the sapphire, there is also rapid surface diffusion across the top of the sapphire wafer during the early stages of the baking process.

Also observed during the early stages of the baking process is a region of very high Ti_2O_3 concentration in which the transport of the Ti_2O_3 is facilitated by a reaction with the sapphire wafer. Initially this region lies directly below the surface source of Ti_2O_3, but after the source is exhausted it shrinks and moves into the sapphire, eventually vanishing; see Figure 2. ORC have observed Ti concentration profiles beneath the position of the initial Ti_2O_3 source, as shown schematically in Figure 3, after long periods of baking. Such profiles are associated with the enhanced diffusivity and exhaustion of the surface source of Ti_2O_3.



(a)
Early stages of baking

(b)
Later stages of baking

Figure 2: Geometry of the sapphire doping process

All the diffusion mechanisms involve movement of Ti_2O_3 molecules, Ti ions and local defects in the sapphire matrix and surface, and are therefore probably best described by a system of (non-linear) diffusion equations. Although this was discussed during the study group, it was decided that in order to understand better the qualitative behaviour of the problem, only simple piecewise linear diffusion models would be treated in depth.

2 Two dimensional diffusion

2.1 Initial rapid surface diffusion

In the initial stages of the baking process, there is evidence to suggest a rapid diffusion of the Ti_2O_3 along the surface of the sapphire wafer. The following model was proposed to analyse this process near to the edge of the surface deposit of the Ti_2O_3. There is a bulk concentration, $\bar{C}(x, y, t)$, of Ti_2O_3 in the sapphire and a surface concentration, $\bar{S}(x, t)$, of Ti_2O_3 on the surface of the sapphire wafer. The diffusivity in the bulk is D_b and the diffusivity on the surface is D_s, with $D_s \gg D_b$. The surface of the wafer is $\bar{y} = 0$ and to the left of the origin it is covered by Ti_2O_3, to the right it is initially clear. It is assumed that, as diffusion proceeds there is a partition co-efficient which relates the surface concentration $\bar{S}(\bar{x}, \bar{t})$ to the bulk concentration $\bar{C}(\bar{x}, 0, \bar{t})$;

$$\bar{C}(\bar{x}, 0, \bar{t}) = \phi \bar{S}(\bar{x}, \bar{t}), \quad \bar{x} > 0, \ \bar{t} > 0.$$

(The units of ϕ are m^{-1}.) Further. Ti_2O_3 from the surface may diffuse into the bulk from the surface, giving rise to a sink term proportional to

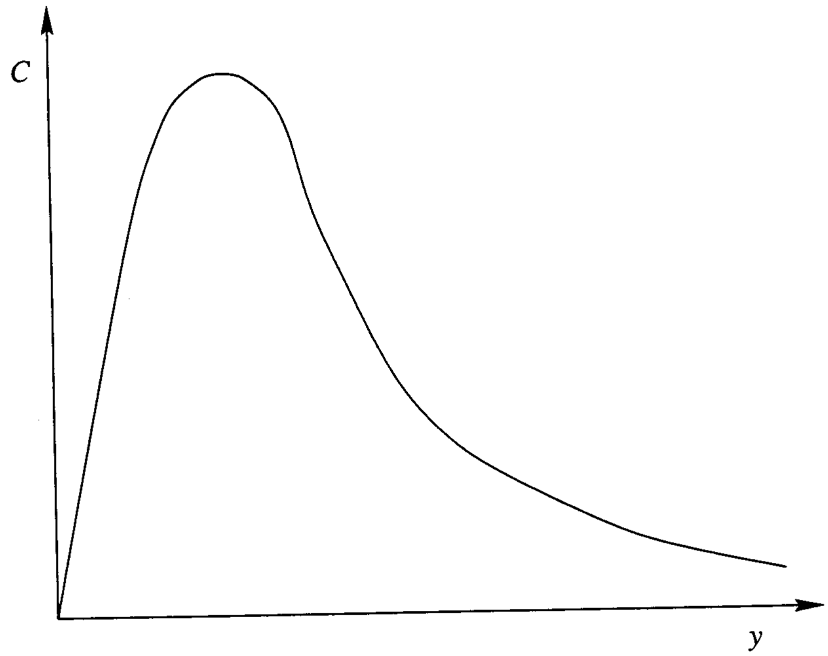


Figure 3: Schematic of Ti_2O_3 concentration profiles after long baking

$(\partial\bar{C}/\partial\bar{y})(\bar{x},0,\bar{t})$ in the diffusion equation for the surface concentration. The dimensional equations are

$$\frac{\partial\bar{C}}{\partial\bar{t}} = D_b\left(\frac{\partial^2\bar{C}}{\partial\bar{x}^2} + \frac{\partial^2\bar{C}}{\partial\bar{y}^2}\right), \quad \bar{y} > 0,\ \bar{t} > 0,$$
$$\frac{\partial\bar{S}}{\partial\bar{t}} = D_s\frac{\partial^2\bar{S}}{\partial\bar{x}^2} + D_b\frac{\partial\bar{C}}{\partial\bar{y}}, \quad \bar{y} = 0,\ \bar{x} > 0,\ \bar{t} > 0.$$

with the initial conditions

$$\bar{C}(\bar{x},\bar{y},0) = 0, \quad \bar{y} > 0,$$
$$\bar{S}(\bar{x},0) = 0, \quad \bar{x} > 0,$$

and the boundary conditions

$$\bar{C}(\bar{x},0,\bar{t}) = \begin{cases} \bar{C}_0 & \text{if } \bar{x} < 0 \\ \phi\bar{S}(\bar{x},\bar{t}) & \text{if } \bar{x} > 0 \end{cases}$$
$$\nabla\bar{C} \to 0, \quad \frac{\partial\bar{S}}{\partial\bar{x}} \to 0, \quad \text{at infinity.}$$

The geometry is illustrated in Figure 4.

We non-dimensionalise $\bar{t}$ with respect to a time-scale T, to be determined, $\bar{x}$ with the surface diffusion length scale $\sqrt{D_sT}$, $\bar{y}$ with the bulk diffusion

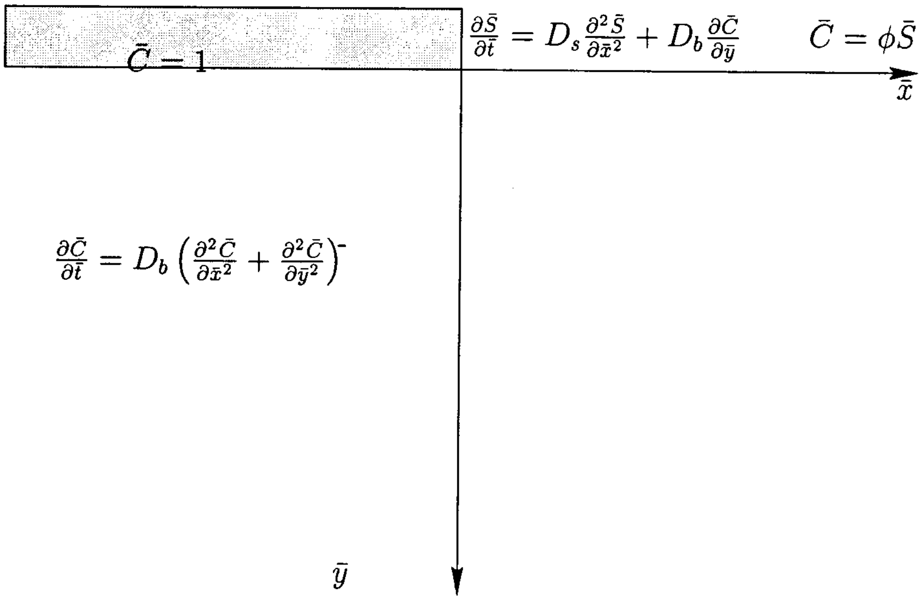


Figure 4: Geometry of the problem near the corner of the Ti_2O_3 surface deposit

length scale $\sqrt{D_b T}$, $\bar{C}$ with respect to $\bar{C}_0$ and $\bar{S}$ with respect to $\bar{C}_0/\phi$,

$$\bar{t} = T\,t, \quad \bar{x} = \sqrt{D_s T}\,x, \quad \bar{y} = \sqrt{D_b T}\,y$$
$$\bar{C} = \bar{C}_0\,C, \quad \bar{S} = \bar{C}_0\,S/\phi.$$

Note that as $D_s \gg D_b$, the non-dimensionalisation effectively compresses distances in the x direction relative to the y direction. We find that

$$\frac{\partial C}{\partial t} = \frac{D_b}{D_s}\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}, \quad \frac{\partial S}{\partial t} = \frac{\partial^2 S}{\partial x^2} + \phi\sqrt{D_b T}\frac{\partial C}{\partial y}$$

so that the choice $T = 1/D_b \phi^2$ gives us

$$\frac{\partial C}{\partial t} = \frac{\partial^2 C}{\partial y^2} + \frac{D_b}{D_s}\frac{\partial^2 C}{\partial x^2}, \quad y > 0, t > 0,$$
$$\frac{\partial S}{\partial t} = \frac{\partial^2 S}{\partial x^2} + \frac{\partial C}{\partial y}, \quad y = 0,\ x > 0,\ t > 0.$$

Since $(D_b/D_s) \ll 1$, the bulk diffusion term $\partial^2 C/\partial x^2$ is neglected (note that this implies that mass transport in the x direction in the bulk is mainly facilitated by the surface diffusion) and the model becomes

$$\frac{\partial C}{\partial t} = \frac{\partial^2 C}{\partial y^2}, \quad y > 0, t > 0,$$

$$\frac{\partial S}{\partial t} = \frac{\partial^2 S}{\partial x^2} + \frac{\partial C}{\partial y}, \quad y = 0,\ x > 0,\ t > 0.$$

with initial conditions

$$C(x,0,0) = 1, \quad x < 0,$$
$$C(x,y,0) = 0 \text{ otherwise}$$
$$S(x,0) = 0, \quad x > 0$$

and boundary conditions

$$C(x,0,t) = 1, \quad x < 0,$$
$$C(x,0,t) = S(x,t)$$
$$\nabla C \to 0, \frac{\partial S}{\partial x} \to 0, \text{ at infinity.}$$

It is possible to show that the solution of this problem is the inverse Laplace transform of

$$\hat{C}(x,y,p) = \exp\left(-\sqrt{p}\, y - \sqrt{p + \sqrt{p}}\, x\right),$$

although no-one was able to execute the inversion analytically. Thus the model was solved numerically, using explicit finite-differences (see Appendix A). This showed that the contours of constant concentration looked suspiciously piece-wise linear (see Figures 5 & 6)—although it is relatively easily to show analytically that this can not be the case.

For $x < 0$ it is clear that

$$C(x,y,t) = \operatorname{erfc}\left(\frac{y}{2\sqrt{t}}\right), \quad (x < 0).$$

The sharp corners in the concentration contours in Figure 6 occur as a result of neglecting the $(D_b/D_s)\partial^2 C/\partial S^2$ term. Note also that the dimensional versions of Figures 5 and 6 will be greatly stretched in the x direction.

2.2 Later stages

As the surface diffusion is much more rapid than the bulk diffusion, the model proposed above is only valid during the early stages of the baking process. The surface diffusion will lead, in the long term, to a layer of Ti_2O_3

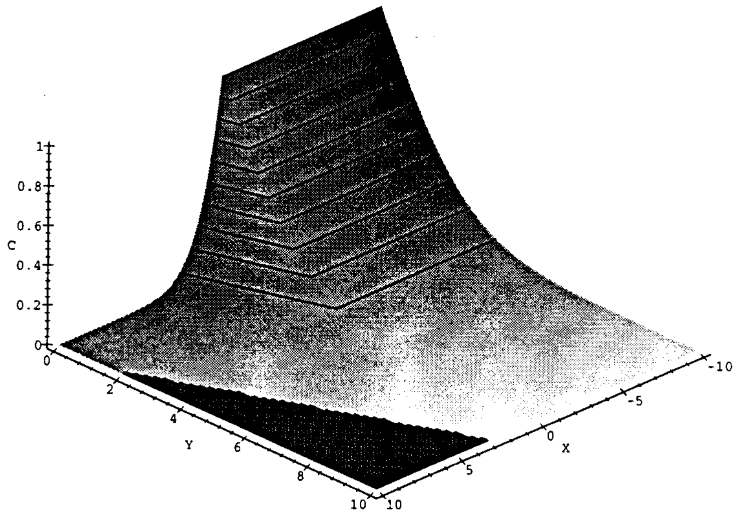


Figure 5: Numerical solution of the short time model.

of locally constant concentration, say C_S along $x > 0$, assuming that the original supply of Ti_2O_3 does not become exhausted. A simple model for this situation is to assume only bulk diffusion of Ti_2O_3

$$\frac{\partial \bar{C}}{\partial \bar{t}} = \frac{\partial^2 \bar{C}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{C}}{\partial \bar{y}^2}, \quad \bar{y} > 0$$

with zero initial concentration

$$\bar{C}(\bar{x}, \bar{y}, 0) = 0,$$

and subject to the discontinuous surface boundary condition

$$\bar{C}(\bar{x}, 0, \bar{t}) = \begin{cases} C_0 & \text{if } \bar{x} < 0, \\ C_S & \text{if } \bar{x} > 0 \end{cases}$$

where $C_s < C_0$. It is assumed here that the surface layer in $\bar{x} > 0$ loses Ti_2O_3 molecules to the bulk, but that the layer is replenished by the rapid surface diffusion from the Ti_2O_3 reserviour in $\bar{x} < 0$.

Note $u = \partial C / \partial x$ satisfies the problem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

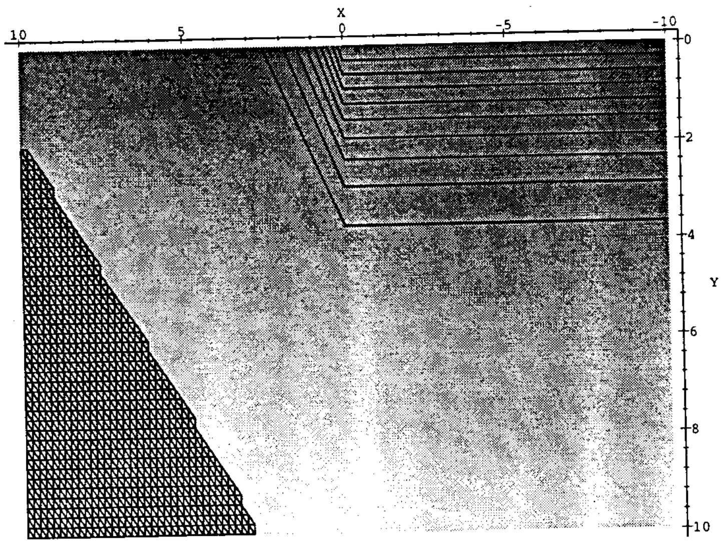


Figure 6: Concentration contour lines of the short time model.

with

$$u(x,y,0) = 0, \quad u(x,0,t) = (C_0 - C_S)\delta(x)$$

which admits a similarity solution of the form

$$u(x,y,t) = \frac{1}{\sqrt{t}} U\left(\frac{x}{\sqrt{t}}, \frac{y}{\sqrt{t}}\right).$$

Indeed, it is easy to see that U depends only on

$$r = \frac{x^2 + y^2}{t},$$

reducing the problem to an ODE. Thus it is possible to find the exact solution

$$C(x,y,t) = C_S \mathrm{erfc}\left(\frac{y}{2\sqrt{t}}\right) + (C_0 - C_S)\psi\left(\frac{x}{2\sqrt{t}}, \frac{y}{2\sqrt{t}}\right)$$

where

$$\psi(\xi, \eta) = \frac{1}{\pi}\eta e^{-\eta^2} \int_{-\infty}^{\xi} \frac{e^{-s^2}}{\eta^2 + s^2} ds.$$

Solutions and concentration contours are shown in Figures 7–10.

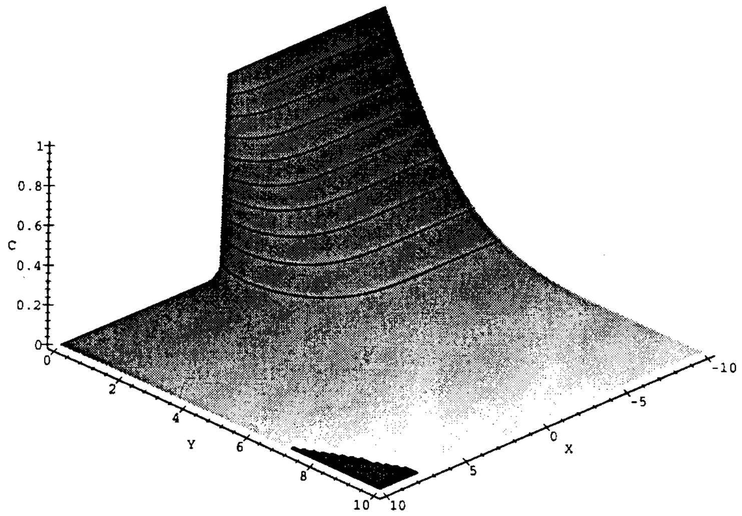


Figure 7: Numerical solution of the long-term model with $C_0 = 1$, $C_S = 0$.

3 Concentration dependent diffusivity

As two-dimensional edge effects are unlikely to be the cause of the concentration profiles illustrated in Figure 3, only diffusion in one spatial dimension was considered. Depletion of the Ti_2O_3 source and a concentration dependent diffusivity are the suspected mechanisms. Although more complicated mechanisms, involving coupled non-linear diffusion of molecular species and defects, were discussed, the following model seemed adequate to explain the observed profiles, at least qualitatively.

It was assumed that Ti_2O_3 molecules diffused according to the linear equation

$$\frac{\partial C}{\partial t} = D_b \frac{\partial^2 C}{\partial y^2}$$

with bulk diffusivity D_b, for $C < C_{\text{sat}}$, where C_{sat} is a "saturation concentration". For concentrations $C > C_{\text{sat}}$, the Ti_2O_3 molecules react with the sapphire, facilitating faster diffusion. The details of the reaction mechanisms were considered but, for this model, it is assumed that the effect can be

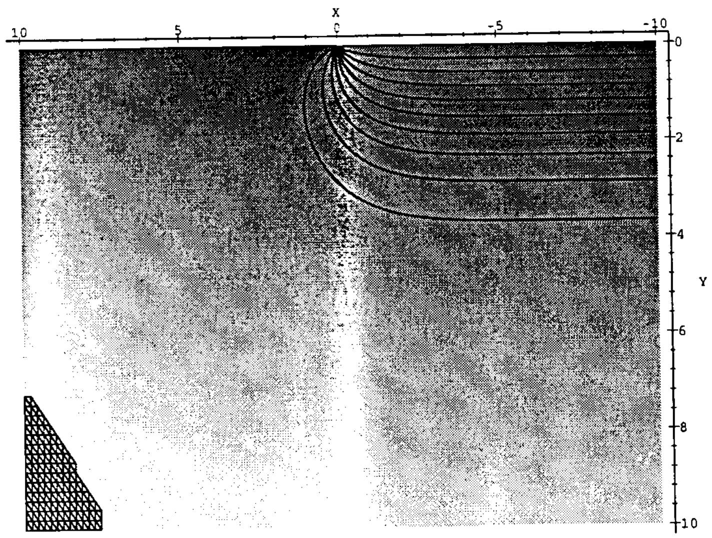


Figure 8: Concentration contours for the long-term model with $C_0 = 1$, $C_S = 0$.

described by a constant "reaction diffusivity", $D_r > D_b$. Thus,

$$\frac{\partial C}{\partial t} = D_b \frac{\partial^2 C}{\partial y^2} \qquad \text{for } C < C_{\text{sat}},$$
$$\frac{\partial C}{\partial t} = D_r \frac{\partial^2 C}{\partial y^2} \qquad \text{for } C \geq C_{\text{sat}}.$$

A sharp boundary, $y = s(t)$, is assumed to separate the saturated and unsaturated regions of the wafer,

$$C\big(s(t), t\big) = C_{\text{sat}}.$$

Mass conservation requires that

$$D_r \frac{\partial C}{\partial y}(s(t)^-, t) = D_b \frac{\partial C}{\partial y}(s(t)^+, t).$$

Prior to depletion of the surface source of Ti_2O_3, there is a constant concentration of Ti_2O_3 at $y = 0$, so

$$C(0, t) = 1.$$

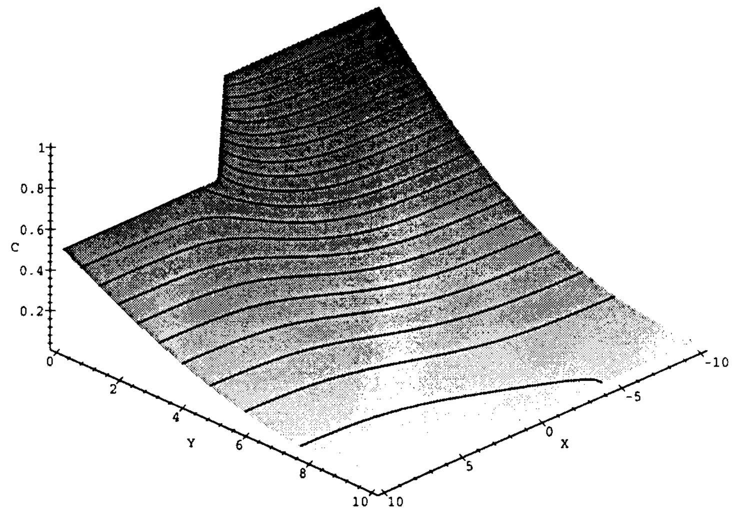


Figure 9: Numerical solution of the long-term model with $C_0 = 1$, $C_S = 0.5$.

Initially, no Ti_2O_3 has diffused into the wafer,

$$C(y,0) = 0, \quad s(0) = 0.$$

This problem is self-similar and has a solution in terms of the variable $y/\sqrt{t}$. Denoting the concentration C by C_b if $C \leq C_{\text{sat}}$ and by C_r if $C \geq C_{\text{sat}}$, the similarity solution is

$$C_r = \operatorname{erfc}\left(\frac{y}{2\sqrt{D_r t}}\right)$$

$$C_b = A\operatorname{erfc}\left(\frac{y}{2\sqrt{D_b t}}\right)$$

$$s(t) = \alpha\sqrt{t}$$

where A is given by

$$A = \exp\left(\frac{\alpha^2(D_r - D_b)}{2D_r D_b}\right) > 0,$$

and α is the unique positive solution of the transcendental equation

$$\operatorname{erfc}\left(\frac{\alpha}{\sqrt{2D_r}}\right) = A\operatorname{erfc}\left(\frac{\alpha}{\sqrt{2D_b}}\right).$$

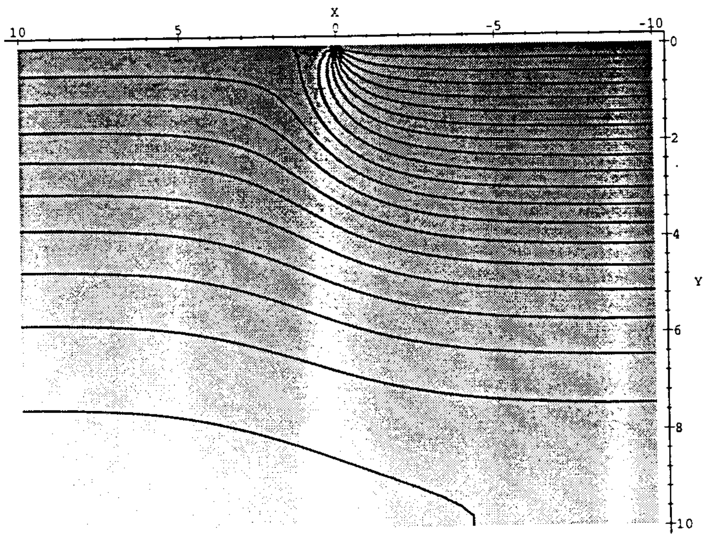


Figure 10: Concentration contours for the long-term model with $C_0 = 1$, $C_S = 0.5$.

Before the Ti_2O_3 source at $y = 0$ has been depleted, we can calculate the rate at which it is depleted, as the mass flow rate is given by by Fick's law,

$$-D_r \frac{\partial C_r}{\partial y}(0, t).$$

(In view of the initial and boundary conditions, it is clear that a reaction zone forms immediately below the surface, hence the concentration C at the surface is given by C_r for $t > 0$ until the surface source is depleted.) Thus the time to depletion of the surface source of Ti_2O_3 t^*, is given implicitly by

$$\int_0^{t^*} D_r \frac{\partial C_r}{\partial y}(0, t)\, dt = \rho H$$

where ρ is the density of the Ti_2O_3 source and H is the height of the initial surface deposit of Ti_2O_3.

After the surface source is depleted, we assume that Ti_2O_3 can evaporate out of the sapphire wafer across its surface and that it does so according to the linear law

$$D(C)\frac{\partial C}{\partial y}(0, t) = hC(0, t),$$

where

$$D(C) = \begin{cases} D_r & \text{if } C > C_{\text{sat}} \\ D_b & \text{if } C < C_{\text{sat}} \end{cases} \tag{1}$$

(The "reaction" between high concentrations of Ti_2O_3 and sapphire is reversible in the sense that the diffusivity returns to D_b if C falls below C_{sat}.)

The problem can be written in conservation form

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial y}\left(D(C)\frac{\partial C}{\partial y}\right)$$

with

$$C(y,0) = 0,$$

$$C(0,t) = 1, \quad t < t^*,$$

or

$$D(C)\left.\frac{\partial C}{\partial y}\right|_{y=0} = hC(0,t), \quad t > t^*,$$

and

$$C \to 0 \text{ as } y \to \infty.$$

This form of the problem is easily solved numerically (see Appendix B). The solutions for $C_{\text{sat}} = 0.5$, $D_b = 1$, $D_r = 10$, $\rho H = 5$ and $h = 1$ are shown at various times in Figures 11–15. Note the re-emergence of a low diffusivity (low concentration) region after the surface source has been exhausted (see Figures 13–15).

Report submitted by Jeff Dewynne.

Contributions from: Alec Buck, Jon Chapman, Peter Howell, John King, Sharon Kirkham, Tim Lattimer, John Lister, Andrew Miller, Mphaka Mphaka, John Ockendon, Martin Pope, Graham Veitch.

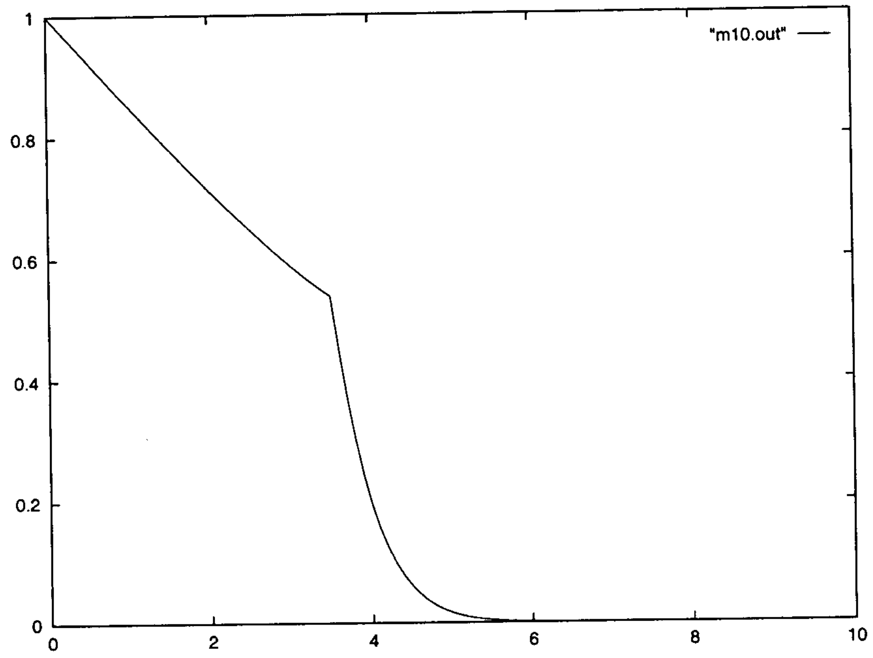


Figure 11: Concentration profile for $C_{\text{sat}} = 0.5$, $D_b = 1$, $D_r = 10$, $\rho H = 5$ and $h = 1$ at time $t = 1$.

A Finite difference code I

```
      program darren
      implicit real*8 (a-h,o-z)
      parameter(ndim=100)
      dimension cold(-ndim:ndim, 0:ndim)
      dimension cnew(-ndim:ndim, 0:ndim)

100   write(*,'(a)') 'dimension '
105   read(*,*,end=999,err=105) n
      if (n.gt.ndim) goto 100

*     set initial values

      do 205 i=-n,n
         do 200 j=0, n
            cold(i,j) = 0.0
200      continue
205   continue
```

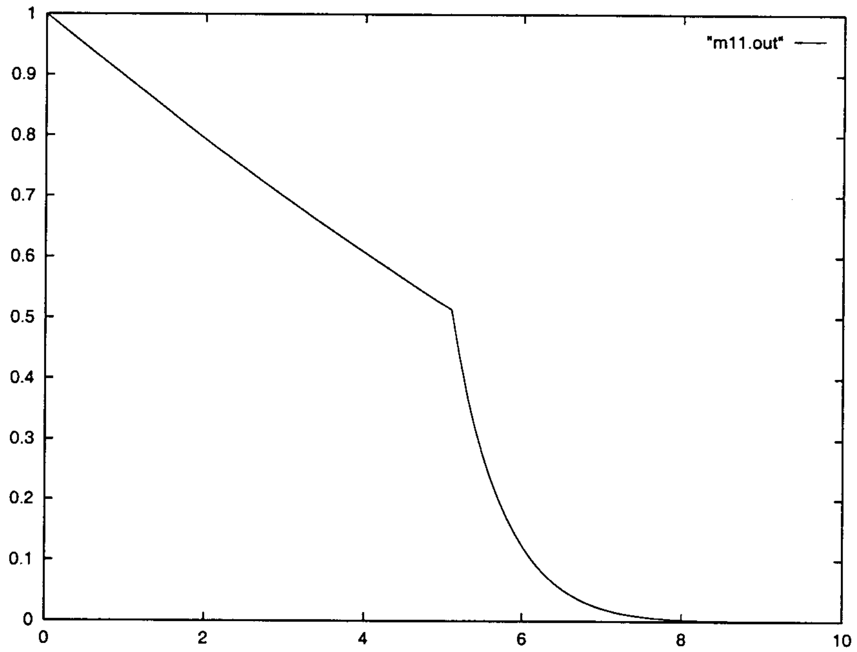


Figure 12: Concentration profile for $C_{\text{sat}} = 0.5$, $D_b = 1$, $D_r = 10$, $\rho H = 5$ and $h = 1$ at time $t = 2$.

```
      do 210 i=-n,0
         cold(i,0) = 1.0
210   continue

300   write(*,'(a)') 'alpha '
      read(*,*,end=999) alpha
      write(*,'(a)') 'xmax '
      read(*,*,end=999) xmax
      write(*,'(a)') 'number of time steps '
      read(*,*,end=999) nt
      dx = xmax/n
      dt = alpha*dx*dx
      write(*,*) 'n  = ',n
      write(*,*) 'xmax = ',xmax
      write(*,*) 'dx = ',dx
      write(*,*) 'dt = ',dt
      alpha2 = 1.0-2.0*alpha
      alpha3 = 2.0*alpha
      beta = dt/dx
```

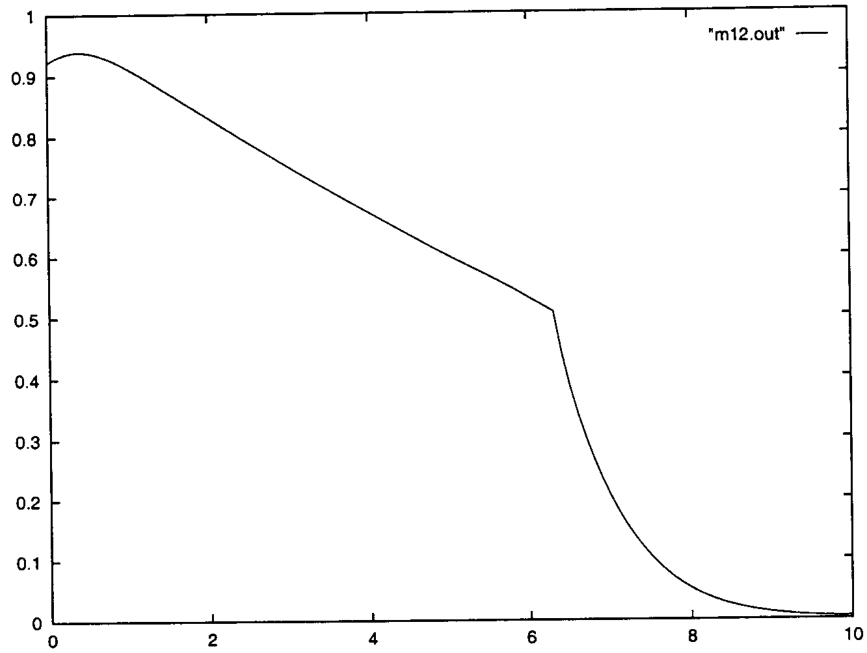


Figure 13: Concentration profile for $C_{sat} = 0.5$, $D_b = 1$, $D_r = 10$, $\rho H = 5$ and $h = 1$ at time $t = 3$.

```
      write(*,*) 'alpha= ',alpha
      write(*,*) 'beta = ',beta
      write(*,*) 'time = ',nt*dt

      do 800 it=1,nt

         do 405 i=-n,0
            cnew(i,0) = 1.0
            do 400 j= 1, n-1
               cnew(i,j) = alpha*(cold(i,j+1)+cold(i,j-1))
               cnew(i,j) = cnew(i,j) + alpha2*cold(i,j)
400         continue
            cnew(i,n) = alpha2*cold(i,n)+alpha3*cold(i,n-1)
405      continue

         do 505 i=1,n-1
            cnew(i,0) = alpha*(cold(i-1,0)+cold(i+1,0))
            cnew(i,0) = cnew(i,0)+alpha2*cold(i,0)
            cnew(i,0) = cnew(i,0)-beta*(cold(i,0)-cold(i,1))
            do 500 j= 1, n-1
```

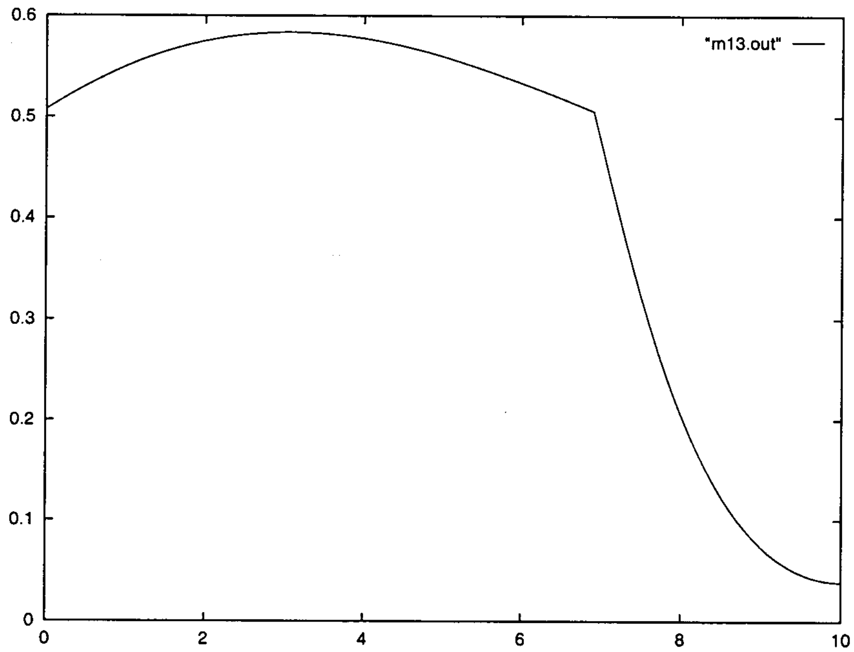


Figure 14: Concentration profile for $C_{\text{sat}} = 0.5$, $D_b = 1$, $D_r = 10$, $\rho H = 5$ and $h = 1$ at time $t = 4$.

```
                 cnew(i,j) = alpha*(cold(i,j+1)+cold(i,j-1))
                 cnew(i,j) = cnew(i,j) + alpha2*cold(i,j)
500            continue
               cnew(i,n) = alpha2*cold(i,n)+alpha3*cold(i,n-1)
505          continue

             cnew(n,0) = alpha2*cold(n,0)+alpha3*cold(n-1,0)
             cnew(n,0) = cnew(n,0)-beta*(cold(n,0)-cold(n,1))

             do 600 j=1,n-1
                cnew(n,j) = alpha*(cold(n,j+1)+cold(n,j-1))
                cnew(n,j) = cnew(n,j)+alpha2*cold(n,j)
600             continue
             cnew(n,n) = alpha2*cold(n,n)+alpha3*cold(n,n-1)

             do 705 i=-n,n
                do 700 j=0,n
                   cold(i,j)=cnew(i,j)
700                continue
705           continue
```

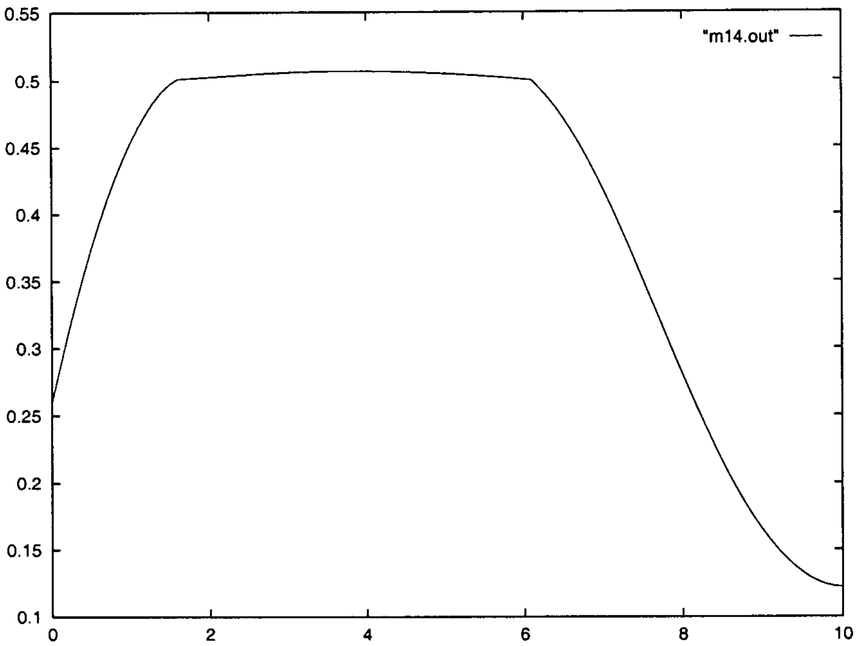


Figure 15: Concentration profile for $C_{\text{sat}} = 0.5$, $D_b = 1$, $D_r = 10$, $\rho H = 5$ and $h = 1$ at time $t = 5$.

```
800        continue

        do 855 i=-n,n
           do 850 j=0,n
              write(*,860) cold(i,j)
850        continue
855     continue
860     format(F16.8)

         goto 100

999      stop
         end
```

B Finite difference code II

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MAXSIZE 1000

static double cold[MAXSIZE+1];
static double cnew[MAXSIZE+1];

int  getpar(double *ccrit,
            double *db,
            double *dr,
            double *xmax,
            double *time,
            int *nx,
            int *nt,
            double *height,
            double *rho,
            double *h);


void main(void)
{
  double ccrit;      // critical value of conc
  double db;         // diffusivity if c < ccrit
  double dr;         // diffusivity if c > ccrit
  double xmax;       // maximum x value
  double time;       // time to solve for
  double dt;         // time step
  double dx;         // space step
  double diff1;      // diffusivity at n-1/2
  double diff2;      // diffusivity at n+1/2
  int nx;            // number of space steps
  int nt;            // number of time steps
  double a;          // dt/dx^2
  double height;     // height of source pile
  double rho;        // density of mass pile
  double h;          // D(C(0,t)) C_x = -h C0
```

```
double mass;     // mass of TiO in initial pile
double integral; // integral of surface mass flux

int ix,it;     // loop variables
int flag;      // flag to indicate exhausted surface TiO

if ( getpar(&ccrit,&db,&dr,&xmax,&time,&nx,&nt,&height,&rho,&h) != EOF ) {

  dt = time/(double)nt;
  dx = xmax/(double)nx;
  a = dt/(dx*dx);
  mass = height*rho;
  flag = 0;

  // set initial conditions

  cold[0] = 1.0;
  for(ix=1; ix <= nx; ++ix) {
    cold[ix] = 0.0;
  }

  // now time step

  integral = 0.0;

  for(it=0; it<nt; ++it) {

    if ( cold[0]+cold[1] > 2*ccrit ) {
      diff1 = dr;
    }
    else {
      diff1 = db;
    }

    //solve pde between end points

    for(ix=1; ix<nx; ++ix) {

      if (cold[ix]+cold[ix+1] > 2*ccrit) {
        diff2 = dr;
      }
```

```
      else {
        diff2 = db;
      }

      cnew[ix] = cold[ix] + a * (
           diff2 * cold[ix+1] -
           ( diff1 + diff2 ) * cold[ix] +
           diff1*cold[ix-1]);

       diff1 = diff2;

   }

// now apply boundary condition at xmax, namely c_x=0.

    if (cold[nx] > ccrit) {
      diff1 = dr;
    }
    else {
      diff1 = db;
    }

    cnew[nx] = cold[nx]+2.0*a*diff1*(cold[nx-1]-cold[nx]);

    // apply boundary condition at x=0

    if (integral < mass) {    // haven't exhausted the TiO source
      cnew[0] = 1.0;
      integral += dr*(cnew[0]-cnew[1])*dt/dx;
    }
    else {          // have exhausted the TiO source
      if (flag == 0) {
        printf("Surface TiO exhausted at time %lf\n",it*dt);
        flag = 1;
      }
      if (cold[0] < ccrit ) {
        diff1=db;
      }
      else {
        diff1=dr;
      }
```

```
      cnew[0] =  cold[0] + 2*diff1*a*(cold[1]-cold[0]);
      cnew[0] -= 2*cold[0]*h*dt/dx;
    }

    for(ix=0; ix <= nx; ++ix) {
      cold[ix] = cnew[ix];
    }
  }

  for(ix=0; ix <= nx; ++ix) {
    printf("%lf\t%lf\n",ix*dx, cold[ix]);
  }
 }
}


int  getpar(double *ccrit,
            double *db,
            double *dr,
            double *xmax,
            double *time,
            int *nx,
            int *nt,
            double *height,
            double *rho,
            double *h)
{
  do {
    printf("Enter critical concentration (0,1) ");
    if (scanf("%lf",ccrit) == EOF) return(EOF);
    if (*ccrit<=0 || *ccrit >= 1) {
      printf("Must be in the range 0--1\n");
    }
  } while (*ccrit <= 0 || *ccrit >= 1);

  printf("Enter bulk diffusivity ");
  if (scanf("%lf",db) == EOF) return(EOF);

  printf("Enter reaction diffusivity ");
  if (scanf("%lf",dr) == EOF) return(EOF);
```

```
  printf("Enter xmax ");
  if (scanf("%lf",xmax) == EOF) return(EOF);

  printf("Enter max time ");
  if (scanf("%lf",time) == EOF) return(EOF);

  printf("Number of space steps ");
  if (scanf("%d",nx)==EOF) return(EOF);

  printf("Number of time steps ");
  if (scanf("%d",nt)==EOF) return(EOF);

  printf("Height of pile of TiO ");
  if (scanf("%lf",height)==EOF) return(EOF);

  printf("Density of TiO in pile ");
  if (scanf("%lf",rho)==EOF) return(EOF);

  printf("Surface mass transfer coefficient h ");
  if (scanf("%lf",h)==EOF) return(EOF);

  return(!EOF);

}
```