### **Example 3.2.5 Multiple Steady States in a Catalyst Pellet**

> restart:

> with(plots):

The governing equation is entered here (after substituting the parameter values):

> eq:=diff(y(x),x$2)-0.04\*y(x)\*exp(16\*(1-y(x))/(1+0.8\*(1-y(x))));



> eqalpha:=subs(y(x)=Y(x,alpha),eq):

> eqalpha:=diff(eqalpha,alpha):

> eqalpha:=subs(diff(Y(x,alpha),alpha)=y2(x),eqalpha):

The sensitivity equation is:

> eqalpha:=subs(Y(x,alpha)=y(x),eqalpha);



The variables are stored in vars:

> vars:=(y(x),y2(x));



The governing equations are stored in eqs:

> eqs:=(eq,eqalpha);



The boundary value problem has multiple solutions. The solution obtained depends on the initial guess provided for α. An initial guess of 0.9 is given:

> alpha0:=0.9;



> ICs:=(y(0)=alpha0,D(y)(0)=0,y2(0)=1,D(y2)(0)=0);



> sol:=dsolve({eqs,ICs},{vars},type=numeric,abserr=1e-10);



> sol(1);



> ypred:=rhs(sol(1)[2]);



> y2pred:=rhs(sol(1)[4]);



The new value of α is obtained as:

> alpha1:=alpha0+(1-ypred)/y2pred;



For this example, the error is calculated based on the boundary condition at x = 1.

> err:=abs(1-ypred);



> alpha0:=alpha1;



> k:=1;



The iteration is performed until the error becomes less than the tolerance limit 1e - 10.

> tol:=1e-10;



> while err> tol do

> ICs:=(y(0)=alpha0,D(y)(0)=0,y2(0)=1,D(y2)(0)=0);

> sol:=dsolve({eqs,ICs},{vars},type=numeric);

> ypred:=rhs(sol(1)[2]);

> y2pred:=rhs(sol(1)[4]);

> alpha1:=alpha0+(1-ypred)/y2pred;

> err:=abs(1-ypred);

> alpha0:=alpha1;k:=k+1;

> end:

> k;



The problem has converged after six iterations. The concentration at the center of the particle (x = 0) is given by:

> alpha1;



The error obtained is:

> err;



Next, the solution obtained is plotted and stored in p1.

> p1:=odeplot(sol,[x,y(x)],0..1,axes=boxed,thickness=3,color=blue):

The same steps are performed for a different initial guess of 0.5. The solution obtained is stored in p2.

> alpha0:=0.5;



> ICs:=(y(0)=alpha0,D(y)(0)=0,y2(0)=1,D(y2)(0)=0);



> sol:=dsolve({eqs,ICs},{vars},type=numeric,abserr=1e-10);



> sol(1);



> ypred:=rhs(sol(1)[2]);



> y2pred:=rhs(sol(1)[4]);



> alpha1:=alpha0+(1-ypred)/y2pred;



> err:=abs(1-ypred);



> alpha0:=alpha1;



> k:=1;



> while err> tol do

> ICs:=(y(0)=alpha0,D(y)(0)=0,y2(0)=1,D(y2)(0)=0);

> sol:=dsolve({eqs,ICs},{vars},type=numeric);

> ypred:=rhs(sol(1)[2]);

> y2pred:=rhs(sol(1)[4]);

> alpha1:=alpha0+(1-ypred)/y2pred;

> err:=abs(1-ypred);

> alpha0:=alpha1;k:=k+1;

> end:

> k;



The problem has converged after eight iterations. The concentration at the center of the particle (x = 0) is given by:

> alpha1;



> err;



> p2:=odeplot(sol,[x,y(x)],0..1,axes=boxed,thickness=3,color=green):

Next, an initial guess of 1e - 4 is used. For this case the updated α becomes a negative. Hence, a scaling factor of ρ=0.2 is used:

> alpha0:=1e-4;



> ICs:=(y(0)=alpha0,D(y)(0)=0,y2(0)=1,D(y2)(0)=0);



> sol:=dsolve({eqs,ICs},{vars},type=numeric,abserr=1e-10);



> sol(1);



> ypred:=rhs(sol(1)[2]);



> y2pred:=rhs(sol(1)[4]);



> alpha1:=alpha0+(1-ypred)/y2pred;



> rho:=0.2;



> alpha1:=alpha0+rho\*(1-ypred)/y2pred;



> err:=abs(1-ypred);



> alpha0:=alpha1;



> k:=1;



> while err> tol do

> ICs:=(y(0)=alpha0,D(y)(0)=0,y2(0)=1,D(y2)(0)=0);

> sol:=dsolve({eqs,ICs},{vars},type=numeric);

> ypred:=rhs(sol(1)[2]);

> y2pred:=rhs(sol(1)[4]);

> alpha1:=alpha0+rho\*(1-ypred)/y2pred;

> err:=abs(1-ypred);

> alpha0:=alpha1;k:=k+1;

> end:

The problem has converged after 93 iterations. The concentration at the center particle (x = 0) is given by:

> k;



> alpha1;

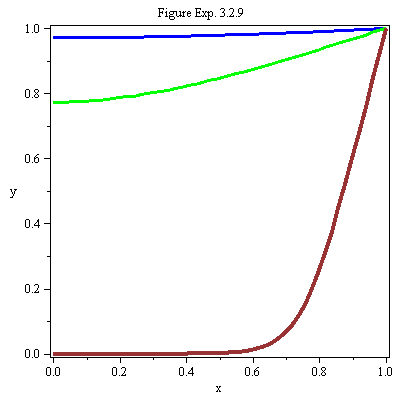


> err;



> p3:=odeplot(sol,[x,y(x)],0..1,title="Figure Exp. 3.2.9",axes=boxed,thickness=4,color=brown):

> display({p1},{p2},{p3});



>

>

Hence, we observe that the shooting technique can predict three multiple states in a catalyst pellet. The number of iterations required to obtain a converged solution depends on the initial guess and the scaling factor ρ.

>