### **Example 3.8a Cylindrical Catalyst Pellet**

> restart:

> with(linalg):with(plots):

> N:=4;

> L:=1;



> eq:=diff(y(x),x$2)+1/x\*diff(y(x),x)-phi^2\*y(x);



> bc1:=diff(y(x),x);



> bc2:=diff(y(x),x)-1+y(x);



The central difference expression for the second and first derivatives are

> d2ydx2:=(y[m+1]-2\*y[m]+y[m-1])/h^2;



> dydx:=(y[m+1]-y[m-1])/2/h;



The three point forward and backward difference expressions for the derivative are:

> dydxf:=(-y[2]+4\*y[1]-3\*y[0])/(2\*h);



> dydxb:=(y[N-1]-4\*y[N]+3\*y[N+1])/(2\*h);



The governing equation in finite difference form is:

> Eq[m]:=subs(diff(y(x),x$2)=d2ydx2,diff(y(x),x)=dydx,y(x)=y[m],x=m\*h,eq);



The boundary conditions in finite difference form are:

> Eq[0]:=subs(diff(y(x),x)=dydxf,y(x)=y[0],bc1);



> Eq[N+1]:=subs(diff(y(x),x)=dydxb,y(x)=y[N+1],bc2);



A 'for loop' can be written for the interior node points as

> for i to N do Eq[i]:=subs(m=i,Eq[m]);od;



> y[0]:=solve(Eq[0],y[0]);



> y[N+1]:=solve(Eq[N+1],y[N+1]);



> h:=L/(N+1);



> for i to N do Eq[i]:=eval(Eq[i]);od;



> eqs:=[seq(Eq[i],i=1..N)];



> vars:=[seq(y[i],i=1..N)];



> A:=genmatrix(eqs,vars,'B1');



> evalm(B1);



Maple generates a row vector, which can be converted to a column vector as:

> B:=matrix(N,1):for i to N do B[i,1]:=B1[i]:od:evalm(B);



The solution is obtained as

> X:=evalm(inverse(A)&\*B);



> for i to N do y[i]:=X[i,1];od;



> y[0]:=eval(y[0]);y[N+1]:=eval(y[N+1]);



Now the result obtained is plotted for different values of Thiele modulus Φ:

> pars:=[0.1,0.5,1,2,5];

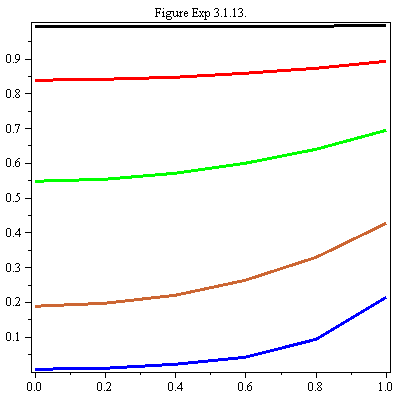


> clr:=[black,red,green,gold,blue];



> for j from 1 to 5 do p[j]:=plot([seq([i\*h,subs(phi=pars[j],y[i])],i=0..N+1)],thickness=3,title="Figure Exp 3.1.13.",axes=boxed,color=clr[j]):od:

> display({seq(p[i],i=1..5)});



>