### **Example 3.6**

> restart:

> with(linalg):with(plots):

> N:=2;

For brevity, only four terms are used for calculating the matrizant in this example.

> nvars:=4;

> Eq:=1/x\*diff(x\*diff(c(x),x),x)=phi^2\*c(x);



Enter the A matrix (equation 3.22).

> A:=matrix(2,2,[0,1/x,phi^2\*x,0]);



> Y0:=matrix(2,1,[c[1],0]);



> id:=Matrix(N,N,shape=identity);



> X1:=matrix(N,N);X2:=matrix(N,N);



> X1:=map(int,subs(x=x1,evalm(A)),x1=0..x);



To avoid the singularity, in X1, integrate from x0 to x and later find the limit as x0 goes to zero.

> X1:=map(int,subs(x=x1,evalm(A)),x1=x0..x)assuming x>0,x0>=0,x>=x0;



> mat := evalm(id + X1) ;



> for i from 2 to nvars do

S:=evalm( subs(x=x1,evalm(A))&\*subs(x=x1,evalm(X1)) ):X2:=

map(int,S,x1=x0..x):mat := evalm(mat +X2) :

X1:=evalm(X2):od :

> evalm(mat)assuming x>0,x0>=0,x>=x0;



> sol:=evalm(mat&\*Y0);



> C:=sol[1,1];



> dCdx:=1/x\*sol[2,1];



To find c1 apply the boundary condition at x=1:

> bc2:=eval(subs(x=1,C))=1 assuming x>0,x0>=0,x>=x0;

Warning, unable to determine if 0 is between x0 and x1; try to use assumptions or set \_EnvAllSolutions to true



> c[1]:=solve(bc2,c[1]);



> C:=eval(C);

Warning, unable to determine if 0 is between x0 and x; try to use assumptions or set \_EnvAllSolutions to true

Warning, unable to determine if 0 is between x0 and x1; try to use assumptions or set \_EnvAllSolutions to true



Now apply the limit command for x0.

>

Warning, premature end of input, use <Shift> + <Enter> to avoid this message.

> C:=limit(C,x0=0);



Divide both numerator and denominator by 64. (Note when different values of 'nvars' are used, this number has to be changed accordingly.)

> n1:=numer(C)/64;



> d1:=denom(C)/64;



> C:=n1/d1;



One can verify that both the numerator and the denominator of C are modified Bessel functions of the order zero by using Maple.

> series(BesselI(0,phi\*x),x);



> series(BesselI(0,phi),phi);



Next, plots can be obtained by substituting the parameters for the Thiele modulus .



> pars:=[0.1,1,2,10];



> clr:=[red,green,blue,brown];



> for i to 4 do p[i]:=plot(subs(phi=pars[i],C),x=0..1,color=clr[i]):od:

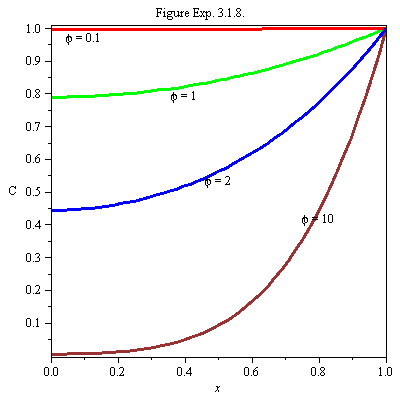
> pt[1]:=textplot([0.1,evalf(subs({x=0.1,phi=pars[1]},C)),'phi=pars[1]'], align=below):

pt[2]:=textplot([0.4,evalf(subs({x=.4,phi=pars[2]},C)),'phi=pars[2]'],align=below):

pt[3]:=textplot([0.5,evalf(subs({x=.5,phi=pars[3]},C)),'phi=pars[3]'],align=below):

pt[4]:=textplot([0.8,evalf(subs({x=0.8,phi=pars[4]},C)),'phi=pars[4]'],align=below):

> display({seq(p[i],i=1..4),seq(pt[i],i=1..4)},axes=boxed,thickness=3,title="Figure Exp. 3.1.8.",labels=[x,"C"]);



For higher values of , more terms (nvars) in the matrizant series solution are needed for higher accurance.



>