

Chapter 6

Maple code for exercises in section 6.1

8.

```
> a := k -> piecewise(k=0, 1/2*int(x^2, x = -1 .. 1), int(x^2*cos(k*Pi*x), x = -1 .. 1));
```

```
> b := k -> int(x^2*sin(k*Pi*x), x = -1 .. 1) ;
```

11.

```
> riemann := proc(x)
```

```
> local pos::list, neg::list, negi, posi, i, l::list, sum, summands::list, j, posl::list,
    negl::list; negi := 1; posi := 1; posl := []; negl := []; i := 1; sum := 0; pos := []; neg
    := []; summands := []; l := [];
```

```
> summands := [seq((-1)^(j-1)*cos((2*j-1)*x)/(2*j-1), j=1..100)];
```

```
> for j from 1 to 100 do if op(j, summands) >= 0 then pos :=
    [op(pos), op(j, summands)]; posl := [op(posl), j]; else neg :=
    [op(neg), op(j, summands)]; negl := [op(negl), j] end if; end do;
```

```
> while i <= 20 do while sum < 1 do sum := sum + op(posi, pos); l :=
    [op(l), op(posi, posl)]; posi := posi + 1; i := i + 1; end do; while sum >= 1 do sum
    := sum + op(negi, neg); l := [op(l), op(negi, negl)]; negi := negi + 1; i := i + 1; end
    do; end do; print(l); print(sum);
    end proc;
```

```
> riemann(.5);
```

The list shows the order in which the summands have been rearranged. thus, with $x = 0.5$, we take the first summand, then the fourth, then the second, then the seventh, and so on.

13.

The following command will calculate a numerical approximation to the value of the integral:

```
> F := n -> evalf(evalf(int(sin((2*n+1)*u)*sqrt(9+2*u)/sin(u), u = 0 .. 1/2*Pi))/Pi) ;
```

```
> plots[listplot]([seq([5*n, F(5*n)],n = 1 .. 30)],style = POINT,symbol = POINT,view
= 1.5 .. 1.51);
```

17.

```
> a := k -> piecewise(k=0, 1/2*int(2*x+1,x = -Pi .. 0)/Pi+int(1/3*x-2/3,x = 0 ..
Pi)/Pi, int((2*x+1)*cos(k*x),x = -Pi .. 0)/Pi+int(1/3*(x-2)*cos(k*x),x = 0 .. Pi)/Pi);
b := k -> int((2*x+1)*sin(k*x),x = -Pi .. 0)/Pi+int(1/3*(x-2)*sin(k*x),x = 0 .. Pi)/Pi ;
```

```
> [a(k), b(k)];
```

Simplify Mathematica's answer by using your knowledge that k is an integer and therefore $\sin(k\pi) = 0$ and $\cos(k\pi) = (-1)^k$.

Maple code for exercises in section 6.2

1.

```
> S1 := n -> sum(j^3/n^4-2*j^2/n^3+j/n^2,j = 0 .. n-1) ;
```

```
> evalf(seq([n, S1(n)], n=1..20),10);
```

2.

In the argument of ApproxS, enter the list of points in the partition in increasing order, starting with 0 and ending with 1.

```
> ApproxS := proc (P::list) local j ;
sum((op(j,P)^3-2* op(j,P)^2+ op(j,P))*( op(j+1,P)- op(j,P)),j = 1 .. nops(P)-1) end
proc;
```

```
> ApproxS([0, .25, .5, .75, 1]);
```

```
> ApproxS([0, .1, .35, .6, .85, 1]);
```

3.

```
> S2 := n -> sum(sin(n/j)/n,j = 1 .. n-1) ;
```

```
> evalf(seq([5*n, S2(5*n)], n=1..20),10);
```

5.

```
> int(cos(100*Pi*x)^2,x = 0 .. 1);

> S3 := n -> sum(cos(100*Pi*j/n)^2/n,j = 0 .. n-1) ;

> evalf(seq([n, S3(n)], n=1..30),10);
```

Maple code for exercises in section 6.3

15.

The first command defines the numerator function $((x))$.

```
> num := x -> piecewise(x < floor(x)+1/2,x-floor(x),x = floor(x)+1/2,0,floor(x)+1/2 <
  x,x-floor(x)-1);
```

This next command simply looks at the plot of this function.

```
> plot(num(x),x = -4 .. 4);
```

The command `points[n]` generates a table of the approximations to $f(x)$ at the values $x = j/1000$, $j = 1, 2, \dots, 1000$, using a summation with n summands (and thus an error that is bounded by $1/2n$).

```
> points := n -> [seq([1/1000*j, sum(num(1/1000*k*j)/k^2,k = 1 .. n)],j = 1 .. 1000)] ;

> plots[listplot](points(10),style = POINT,symbol = POINT);

> plots[listplot](points(100),style = POINT,symbol = POINT);

> plots[listplot](points(1000),style = POINT,symbol = POINT);
```

19.

```
> fun := (k,d)-> piecewise( k mod d = 0, (-1)^d*d , 0 ) ;

> psi := k -> sum(fun(k,d),d = 1 .. k) ;

> seq([k, psi(k)], k=1..100);
```

Maple code for exercises in section 6.4

2.

```
> distance := x -> piecewise(x <= floor(x)+1/2,x-floor(x),floor(x)+1/2 <
  x,floor(x)+1-x) ;
```

```
> plot(distance(x),x = -2 .. 2);
```

3.

```
> plotF := n -> plot(distance(4^n*x)/(4^n),x = -4^(1-n) .. 4^(1-n)) ;
```

Notice what happens to the scale on both the x- and y-axes.

```
> plotF(2);
```

```
> plotF(3);
```

```
> plotF(4);
```

4.

```
> S4 := (n, x) -> sum(distance(4^k*x)/(4^k),k = 0 .. n) ;
```

```
> plot(S4(2,x),x = -2 .. 2);
```

```
> plot(S4(3,x),x = -2 .. 2);
```

```
> plot(S4(4,x),x = -2 .. 2);
```

Why don't these look any different?

11.

```
> S5 := (n, x) -> sum((6/7)^k*cos(7^n*Pi*x),k = 0 .. n) ;
```

Notice what happens to the scale on both the x- and y-axes.

```
> plot(S5(1,x),x = -1 .. 1);
```

```
> plot(S5(2,x),x = -1/7 .. 1/7);
```

```
> plot(S5(3,x),x = -1/49 .. 1/49);
```