

Appendix A

Maple code for exercises in section A.1

1.

```
> wallis := n -> evalf( product( 4*k^2/(4*k^2-1),k=1..n));
```

> Show that the average value of the upper and lower bounds on π can be computer as

```
> wallisaverage := n -> 2*evalf( product( 4*k^2/(4*k^2-1),k=1..n)*(1+1/(4*n)) );
```

7.

```
> evalf( binomial(1,1/2),20 );
```

```
> evalf( 4/Pi, 20);
```

13.

```
> evalf( binomial(1,1/3),20);
```

```
> evalf(1/int((1-x^(3/2))^(1/3), x=0..1),20);
```

Maple code for exercises in section A.2

1.

> The n th Bernoulli number is stored in Maple as `bernoulli(n)`. The following command generates a list of the first forty Bernoulli numbers

```
> [seq(bernoulli(n),n=1..40)];
```

> We can now use the recursive formula for Bernoulli polynomials:

```
> B := (n,x) -> if n=1 then x-1/2 else n*int(B(n-1,t),t=0..x)+bernoulli(n) end if;
```

```
> [seq(B(n,x),n=1..10)];
```

2.

> for n from 1 to 8 do plot(B(n,x),x=-1..2) end do;

8 & 9

> The command NB(n) finds the numerator of the 2nth Bernoulli number. Thus, the numerator of B₂₀ is NB[10]. The command DB(n) finds the denominator.

> NB := n -> numer(bernoulli(2*n)); DB := n -> denom(bernoulli(2*n));

> The following commands list the factorizations of the numerators and of the denominators of the Bernoulli numbers. The first column gives the value of n (it is the Bernoulli number for 2n), the second column list the prime divisors, and the third column gives the powers of each of those prime divisors.

> [seq([n,ifactor(NB(n))],n=1..20)];

> [seq([n,ifactor(DB(n))],n=1..20)];

Maple code for exercises in section A.3

1.

> The following command calculates the difference between the sum of the reciprocals of the first n perfect squares and $\pi^2/6$.

> df := n -> evalf(sum(1/k^2,k=1..n) - Pi^2/6,10);

> [seq([100*n, df(100*n)],n=1..10)];

> The next command adds 1/n to the sum of the reciprocals of the first n squares.

> dfplus := n -> evalf(1/n + sum(1/k^2,k=1..n) - Pi^2/6,10);

> [seq([100*n, dfplus(100*n)],n=1..10)];

2.

> The following command calculates the difference between the sum of the reciprocals of the first n fourth and sixth powers and the exact values of the infinite series.

```
> df4 := n -> evalf( sum(1/k^4,k=1..n) - Pi^4/90,10);
```

```
> [seq( [100*n, df4(100*n)],n=1..10)];
```

```
> df6 := n -> evalf( sum(1/k^6,k=1..n) - Pi^6/945,10);
```

```
> [seq( [100*n, df6(100*n)],n=1..10)];
```

8.

```
> P := (n,x) -> 1 + sum( bernoulli(k)*x^k/k!,k=1..n);
```

> The following command superimposes the the polynomials that approximate $x/(\exp(x)-1)$.

```
> plot([P(4,x),P(6,x),P(8,x),P(10,x),P(12,x),x/(exp(x)-1)],x=-8..8);
```

11.

> The command given below will find these derivatives, but notice that the denominators are 0. Show that in the limit as x approaches 0, these all approach the correct values.

```
> for n from 1 to 3 do simplify(diff(x/(exp(x)-1),x$n)) end do;
```

14.

```
> Q := (n,x) -> 1 + sum((-1)^k * bernoulli(2*k)*(2*x)^(2*k)/(2*k)!, k=1..n);
```

```
> plot([Q(2,x),Q(4,x),Q(6,x),Q(8,x),x+cot(x)],x=-4..4);
```

17.

```
> R := (n,z) -> 1/z + 2*sum( z/(z^2-k^2*Pi^2), k=1..n);
```

```
> plot([R(3,x),R(6,x),R(9,x),R(12,x),cot(x)],x=-4..4,y=-1..1);
```

> The following command plots the differences between the cotangent and each of these series. Use these graphs to give a reasonable approximation of the error function $\cot(x)-R(n,x)$

```
> for n from 1 to 4 do plot(cot(x) - R(3*n,x),x=-4..4) end do;
```

20.

> If we evaluate the first 1000 terms of this series, then we obtain a value that is between

> **evalf(1/(2*1001^2), 10);**

> and

> **evalf(1/(2*1000^2), 10);**

> below the true value of $\zeta(3)$, and therefore $\zeta(3)$ lies in the interval

> **[evalf(sum(1/k^3,k=1..1000)+1/(2*1001^2),12),evalf(sum(1/k^3,k=1..1000)+1/(2*1000^2),12)];**

Maple code for exercises in section A.4

2.

> The command `StirlingTriple` produces three numbers: n , $n!$ (to 10-digit accuracy), and Stirling's formula for n with the first two terms of the asymptotic series (to 10-digit accuracy).

> **StirlingTriple := x -> [x,evalf(x!,10),evalf((x/exp(1))^x*sqrt(2*Pi*x)*exp(1/(12*x) - 1/(30*x^3)), 10)];**

> **[seq(StirlingTriple(x),x in [5, 10, 20, 50, 100])];**

3.

> The following command finds the absolute value of the m th summand in the asymptotic series, evaluated at n .

> **Summand := (m,n) -> abs(bernoulli(2*m)/((2*m)*(2*m-1)*n^(2*m-1)));**

> The next command lists the absolute values of the summands, taking 20 summands between 1 and $10n$.

> **ListSummands := n -> [seq([floor(m*n/2), evalf(Summand(floor(m*n/2),n),10)],m=1..20)];**

> **ListSummands(10);**

> To refine your search, the following command lists all the summands in the range a to b

> **RefineSearch := (n,a,b) -> [seq([m,evalf(Summand(m,n),10)],m=a..b)];**

> **RefineSearch(10,25,40);**

> This command allows to compare the value of the approximation to $n!$ with the best asymptotic estimate (to m terms):

> **Cf := (n,m) -> [n!,evalf((n/exp(1))^n
*sqrt(2*Pi*n)*exp(sum(bernoulli(2*k)/(2*k*(2*k - 1)*n^(2*k - 1)),
k=1..m))),ceil(log10(n!))];**

> **Cf(10,3);**

5.

> The first command calculates $1/2 +$ the sum of the first n terms. The second produces a table of these values as n goes from 1 to 10. The third command gives the actual value of Euler's gamma to 10-digit accuracy for purposes of comparison.

> **GammaApprox := n -> evalf(1/2 + sum(bernoulli(2*k)/(2*k), k=1..n), 10);**

> **[seq([n,GammaApprox(n)],n=1..10)];**

> **evalf(gamma,10);**

6.

> The following command finds the absolute value of the m th summand in the asymptotic series, evaluated at n .

> **HarmonicSummand := (m,n) -> abs(bernoulli(2*m)/(2*m*n^(2*m))) ;**

> The next command lists the absolute values of the summands, taking 20 summands between 1 and $10n$.

> **ListHarmonicSummands := n -> [seq([floor(m*n/2), evalf(
Summand(floor(m*n/2),n), 10)],m=1..20)];**

> **ListHarmonicSummands(10);**

> To refine your search, the following command lists all the summands in the range a to b

> **RefineHarmonicSearch := (n,a,b) -> [seq([m,evalf(Summand(m,n),10)],m=a..b)];**

> **RefineHarmonicSearch(10,25,40);**

> This command allows you to compare the value of the approximation to the harmonic series with the best asymptotic estimate (to m terms):

> **HarmonicCf := (n,m) -> [evalf(sum((1/k),k=1..n),n), evalf(ln(n) + gamma + 1/(2*n) - sum(bernoulli(2*k)/(2*k*n^(2*k)), k=1..m) , n)];**

> **HarmonicCf(10,3);**