CHAPTER 3

Numerical Series and Power Series

This chapter demonstrates the wide range of features that MATLAB offers which can be used to treat numerical series. These include the determination of the radius of convergence of a power series, summation of convergent series, alternating series and so on.

3.1 Series. Convergence Criteria

We distinguish between numerical series which have non-negative terms and series which have alternating terms. Some examples of the more classical criteria to determine whether a series of non-negative terms converges or diverges will be presented first, then we will go on to analyze alternating series.

3.2 Numerical Series with Non-Negative Terms

Among the most common of the convergence criteria is the ***ratio test*** or ***d'Alembert criterion***, which reads as follows:

$$\sum_{n=1}^{\infty} a(n) \text{ is convergent if } \lim_{n\to\infty} \frac{a(n+1)}{a(n)} < 1$$

$$\sum_{n=1}^{\infty} a(n) \text{ is divergent if } \lim_{n\to\infty} \frac{a(n+1)}{a(n)} > 1$$

If the limit is 1, we don't know whether the series converges or diverges.

Another widely used criterion is the ***Cauchy criterion*** or ***root test***, which reads as follows:

$$\sum_{n=1}^{\infty} a(n) \text{ is convergent if } \lim_{n\to\infty} \sqrt[n]{a(n)} < 1$$

$$\sum_{n=1}^{\infty} a(n) \text{ is divergent if } \lim_{n\to\infty} \sqrt[n]{a(n)} > 1$$

Again, if the limit is 1, we cannot say whether the series diverges or converges.

If a limit of 1 is obtained in both the ratio and root test, we can often use the ***criterion of Raabe or Duhamel***, which reads as follows:

$$\sum_{n=1}^{\infty} a(n) \text{ is convergent if} \lim_{n\to\infty} \left[n \left(1 - \frac{a(n+1)}{a(n)} \right) \right] > 1$$

$$\sum_{n=1}^{\infty} a(n) \text{ is divergent if } \lim_{n\to\infty}\left[n\left(1-\frac{a(n+1)}{a(n)}\right)\right]<1$$

However, if the limit is 1, we still cannot conclude anything about the convergence or divergence of the series. Another useful criterion is the following:

$$\sum_{n=1}^{\infty} a(n) \text{ and } \sum_{n=1}^{\infty} 2^n a\left(2^n\right) \text{ either both diverge or both converge.}$$

Other criteria such as Gauss majorization, comparison tests and so on can also be implemented via Maple. We will see some examples below:

EXERCISE 3-1

Study the convergence and, if possible, find the sum of the following series:

$$\sum_{n=1}^{\infty} \frac{1+n}{n(2+n)(3+n)}$$

We will apply the ratio test:

```
>> maple('a:=n -> (n+1)/(n*(n+2)*(n+3)):limit(a(n+1)/a(n), n=infinity);')

ans =

1
```

We see that the limit is 1, so we can't yet say anything about the convergence or divergence of the series. We will apply Raabe's criterion:

```
>> maple('a:=n-> (n+1) / (n * (n + 2) * (n)): limit (n * (1-a (n + 1) /a (n)),
n = infinity);')

ans =

2
```

As the limit is greater than 1, the series converges and will be summable:

```
>> maple('sum(a(n),n=1..+infinity)')

ans =

17/36
```

The MATLAB command *symsum* can be used to sum series, but it is not as strong as the command *maple('sum')*. The syntax of both is:

```
syms v, a, b
symsum(S,v,a,b) sums the series with general term S where the variable v ranges between a and b
```

symsum(S,v) sums the series with general term S in the variable v

maple('sum(S,v=a..b)') sums the series with general term S where the variable v ranges between a and b

The sum of the series can also be found as follows:

```
>> syms n
>> symsum((n+1)/(n*(n+2)*(n+3)),1,Inf)

ans =

17/36
```

EXERCISE 3-2

Study the convergence and, if possible, find the sum of the following series:

$$\sum_{n=1}^{\infty}\frac{n}{2^n},\quad \sum_{n=1}^{\infty}\frac{n^n}{n!},\quad \sum_{n=1}^{\infty}\frac{n^n}{3^n\,n!}$$

We apply the ratio test to the first series:

```
>> maple('a:=n -> n/2^n: limit(a(n+1)/a(n), n=infinity)')

ans =

1/2
```

The limit is less than 1, so the series converges. We find its sum:

```
>> maple('sum(a(n),n=1..+infinity)')

ans =

2
```

We apply the ratio test to the second series:

```
>> maple('a:=n -> n^n/n!: limit(a(n+1)/a(n), n=infinity)')

ans =

exp (1)
```

As the limit is greater than 1, the series diverges.

We apply the ratio test to the third series:

```
>> maple('a:=n -> n^n/((n!)*(3^n)): limit(a(n+1)/a(n), n=infinity)')

ans =

1/3 * exp (1)
```

The limit is less than 1, so the series is convergent. Trying to find the exact sum, we see that MATLAB does not solve the problem:

```
>> maple('sum(a(n),n=1..+infinity)')

ans =

sum(n^n/n!/(3^n),n = 1 .. inf)
```

EXERCISE 3-3

Study the convergence and, if possible, find the sum of the following series:

$$\sum_{n=1}^{\infty} \frac{3+2n}{7^n n(1+n)}, \quad \sum_{n=1}^{\infty} \frac{n}{p^n}, \quad \sum_{n=1}^{\infty} \frac{(n+p)!}{p^n n! p!}, \quad p = \textit{parameter}$$

We apply the ratio test to the first series:

```
>> maple('a:=n -> (2*n+3)/(n*(n+1)*(7^n)): limit(a(n+1)/a(n), n=infinity)')

ans =

1/7
```

As the limit is less than 1, the series is convergent. We will calculate its sum. The result that is returned will sometimes be complicated, depending on certain functions that are implemented in MATLAB (in this case *hypergeom*):

```
>> maple('sum(a(n),n=1..+infinity)')

ans =

16/3+30*log(6/7)-13/49*hypergeom([2, 2],[3],1/7)

>> maple('evalf(")')

ans =

.3833972806909634
```

Now we apply the ratio test to the second series:

```
>> maple('a:=n -> n/p^n');
>> maple('limit(a(n+1)/a(n), n=infinity)')

ans =

1/p
```

Thus, if $p > 1$, the series converges, and if $p < 1$, the series diverges. If $p = 1$, we get the series with general term n, which diverges. For p greater than 1, we find the sum of the series:

```
>> maple('sum(a(n),n=1..+infinity)')

ans =

p /(-1+p) ^ 2
```

We apply the ratio test to the third series:

```
>> maple('a:=n ->(n+p)!/((p!)*(n!)*(p^n))');
>> maple('limit(a(n+1)/a(n), n=infinity)')

ans =

limit((n+1+p)!/(n+1)!/(p^(n+1))/(n+p)!*n!*p^n,n = inf)
```

Here MATLAB has not been able to find the limit, so we will try the Raabe criterion instead:

```
>> maple('limit(n*(1-a(n+1)/a(n)),n=infinity)')

ans =

limit(n*(1-(n+1+p)!/(n+1)!/(p^(n+1))/(n+p)!*n!*p^n),n = inf)
```

Again, the limit could not be found. We try to simplify the expression before finding the limit:

```
>> maple('limit(simplify(a(n+1)/a(n)), n=infinity)')

ans =

1/p
```

Thus, if $p > 1$, the series converges, if $p < 1$, the series diverges and if $p = 1$, we get the series with general term $n + 1$, which diverges.

EXERCISE 3-4

Study the convergence and, if possible, find the sum of the following series:

$$\sum_{n=1}^{\infty}(1+\frac{1}{n})^{-n^2}, \quad \sum_{n=1}^{\infty}\left[\left(\frac{1+n}{n}\right)^{1+n}-\frac{1+n}{n}\right]^{-n}$$

For the first series we apply the ratio test:

```
>> maple('a:=n ->(1+1/n)^(-n^2)');
>> maple('simplify(limit(a(n+1)/a(n), n=infinity))')

ans =

exp (- 1)
```

As the limit is less than 1, the series converges. If we try to find the sum with the commands *sum* or *symsum*, we do not obtain an answer. MATLAB is unable to sum this series, so we can only find an approximate numerical sum:

```
>> maple ('evalf (sum ((n), n = 1.. + infinity))')

ans =

.8174194332978335
```

We apply the ratio test to the second series:

```
>> maple ('a: = n - > (((n + 1) / n) ^ (n + 1) - a (n + 1) /n) ^(-n)');
>> maple ('simplify (limit (to (n + 1) /a (n), n = infinity))')

ans =

1 / (exp (1) - 1)
```

As the limit is less than 1, the series converges. The exact sum can be found with MATLAB, but we can approximate the sum using the command *evalf*:

```
>> maple('evalf(sum(a(n),n=1..+infinity))')

ans =

1.174585575087587
```

EXERCISE 3-5

Study the convergence and, if possible, find the sum of the following series:

$$\sum_{n=1}^{\infty} \frac{5}{2^n}, \quad \sum_{n=1}^{\infty} (\sqrt[n]{n} - 1)^n, \quad \sum_{n=1}^{\infty} \left(\frac{n^2 + 2n + 1}{n^2 + n - 1} \right)^{n^2}$$

For the first series, we apply the root test:

```
>> maple('a:=n ->5/(2^n)');
>> maple('limit((a(n))^(1/n), n=infinity)')

ans =

1/2
```

As the limit is less than 1, the series is convergent:

```
>> maple('sum(a(n),n=1..+infinity)')

ans =

5
```

Now we apply the root test to the second series:

```
>> maple('a:=n ->(n^(1/n)-1)^n');
>> maple('limit(simplify((a(n))^(1/n)), n=infinity)')

ans =

0
```

As the limit is less than 1, the series is convergent, but neither the exact nor an accurate approximate sum are calculable.

Now we apply the root test to the third series:

```
>> maple('a:=n->((n^2+2*n+1)/(n^2+n-1))^(n^2)');
>> maple('simplify(limit(simplify((a(n))^(1/n)), n=infinity))')

ans =

exp (1)
```

As the limit is greater than 1, the series diverges.

EXERCISE 3-6

Study the convergence and, if possible, find the sum of the following series:

$$\sum_{n=1}^{\infty} \tan^n\left(p+\frac{q}{n}\right).$$

We apply the root test:

```
>> maple ('a: = n - > (tan(p+q/n)) ^ n');
>> simple(maple('limit(simplify((a(n))^(1/n)), n=infinity)'))

ans =

sin(p)/cos(p)
```

Then, for values of *p* such that *tan(p)* < 1 the series converges. These values are 0 < *p* < *Pi*/4. And for values of *p* such that *tan(p)* > 1 the series diverges. These values are *Pi*/4 < *p* < *Pi*/2. MATLAB does not offer the exact or an accurate approximate sum of this series.

EXERCISE 3-7

Study the convergence and, if possible, find the sum of the following series:

$$\sum_{n=1}^{\infty} \frac{1}{n\,Log(n)}, \quad \sum_{n=1}^{\infty} \frac{1}{n[Log(n)]^p}, \quad p = \text{parameter} > 0$$

We apply the criterion that the series $\sum_{n=1}^{\infty} a(n)$ and $\sum_{n=1}^{\infty} 2^n a(2^n)$ either both diverge or both converge.

```
>> maple ('a: = n - > 1 / (n * log (n))');
>> maple('b:=(2^n)*a(2^n)');
>> pretty(maple('simplify(b)'))

     1
---------
n log (2)
```

As the general term is a constant multiple of the general term of the divergent harmonic series, this series diverges, and we conclude that the original series diverges.

Let us now apply the same criteria to the second series:

```
>> maple('a:=n->1/(n*(log(n))^p)');
>> maple('b:=(2^n)*a(2^n)');
>> pretty(maple('simplify(b)'))
```

```
  (- p)       (- p)
n      log(2)
```

When $p < 1$, this series dominates the series with general term $n^{-p} = 1/n^p$. Thus the initial series also diverges.

When $p < 1$, this series is dominated by the convergent series with general term $n^{-p} = 1/n^p$. Thus the initial series also converges.

When $p = 1$, the series reduces to the series studied above, i.e. it diverges.

MATLAB does not offer the sum of either of the two series of this problem.

EXERCISE 3-8

Study the convergence and, if possible, find the sum of the following series:

$$\sum_{n=1}^{\infty}\frac{(n+1)(n+2)}{n^5},\quad \sum_{n=1}^{\infty}\frac{1}{\left(1+\sqrt{n}\right)^2}$$

We will begin by studying the second series. We try to apply the ratio, Raabe and root tests:

```
>> maple('a:=n->1/(1+n^(1/2))^2');
>> maple('limit(a(n+1)/a(n), n=infinity)')

ans =

1

>> maple('limit(n*(1-a(n+1)/a(n)),n=infinity)')

ans =

1

>> maple('limit(simplify((a(n))^(1/n)), n=infinity)')

ans =

1
```

Thus all the limits are 1. Therefore, at the moment, we cannot conclude anything about the convergence of the series.

We now compare our series with the divergent harmonic series by finding the limit of the quotient of the respective general terms:

```
>> maple('limit(a(n)/(1/n),n=infinity)')

ans =

1
```

As the limit is greater than zero, the initial series is also divergent.

We will now analyze the first series of the problem directly, comparing it with the convergent series with general term $1/n^3$ by examining the limit of the quotient of the general terms:

```
>> maple('a:=n->(n+1)*(n+2)/n^5');
>> maple('limit(a(n)/(1/(n^3)),n=infinity)')

ans =

1
```

As the limit is greater than 0, the initial series is also convergent:

```
>> maple('sum(a(n),n=1..+infinity)')

ans =

2 * zeta (5) + 1/30 * pi ^ 4 + zeta (3)
```

Now, we try to approximate the result:

```
>> maple('evalf(sum(a(n),n=1..+infinity))')

ans =

6.522882114579747
```

We could have tried to determine whether the first series converges or diverges by using the ratio, root and Raabe criteria:

```
>> maple('limit(a(n+1)/a(n), n=infinity)')

ans =

1

>> maple('limit(simplify((a(n))^(1/n)), n=infinity)')

ans =

1

>> maple('limit(n*(1-a(n+1)/a(n)),n=infinity)')

ans =

3
```

The root and ratio tests tells us nothing, but since the limit is greater than 1, the Raabe criterion tells us that the series converges.

3.3 Alternating Numerical Series

We now consider numerical series that have alternating positive and negative terms.

A series ∑*a(n)* is *absolutely convergent* if the series ∑ |*a(n)*| is convergent. As the series of moduli is a series of positive terms, we already know how to analyze it.

Every absolutely convergent series is convergent.

Apart from the criteria described earlier, there are, among others, two classical criteria that allow us to analyze the nature of alternating series, which will allow us to resolve most convergence problems concerning alternating series.

The ***Dirichlet test*** says that if the sequence of partial sums of ∑*a(n)* is bounded and *{b(n)}* is a decreasing sequence that has limit 0, then the series ∑*a(n)b(n)* is convergent.

Abel's test says that if ∑*a(n)* is convergent and *{b(n)}* is a monotone convergent sequence, then the series ∑*a(n)b(n)* is convergent.

EXERCISE 3-9

Study the convergence of the following series:

$$\sum_{n=1}^{\infty} \frac{(-1)^{1+n}}{1+2n^2}$$

This is an alternating series. Let us consider the series of moduli and analyze its character:

```
>> maple ('a: = n - > 1 /(2*n^2+1)');
```

We apply to this series of positive terms the criteria of comparison of the second kind, comparing with the convergent series with general term $1/n^2$:

```
>> maple('limit(a(n)/(1/(n^2)),n=infinity)')

ans =

½
```

As the limit is greater than zero, the given series of positive terms is convergent, so the initial series is absolutely convergent and, therefore, convergent.

EXERCISE 3-10

Study the convergence of the following series:

$$\sum_{n=1}^{\infty} \frac{(-1)^{1+n}\, n}{(1+n)^2}, \quad \sum_{n=1}^{\infty} \frac{(-1)^{1+n}}{n}$$

Defining $a(n)=(-1)^{1+n}$ and $b(n)=\frac{n}{(1+n)^2}$, we have that $\sum a(n)$ has bounded partial sums and {b(n)} is monotone decreasing with limit 0.

Using the Dirichlet test we conclude that the series is convergent.

For the second series we similarly proceed as follows.

Defining $a(n)=(-1)^{1+n}$ and $b(n)=\frac{1}{n}$, we have that $\sum a(n)$ has bounded partial sums and $\{b(n)\}$ is monotone decreasing with limit 0.

Using the Dirichlet test we conclude that the series is convergent.

3.4 Power Series

Given the power series $\sum a(n)x^n$, the most pressing issue is to calculate the range of convergence, i.e., the range of values of x for which the series is convergent.

The most common convergence criteria that we use are the root and ratio tests applied to the series of moduli (absolute values). If we can show the series is convergent then the original series is absolutely convergent and hence convergent.

EXERCISE 3-11

Study the range of convergence of the following power series:

$$\sum_{n=0}^{\infty} \frac{4^{2n}}{n+2}(x-3)^n$$

We apply the ratio test:

```
>> maple('a:=n->(4^(2*n))*((x-3)^n)/(n+2)');
>> maple('limit(simplify(a(n+1)/a(n)), n=infinity)')

ans =

16 * x-48
```

The series will be convergent when | 16 (- 3 + *x*) | < 1 :

```
[solve('16*x-48=1'), solve('16*x-48=-1')]

ans =

49/16  47/16
```

Thus, the condition $|16(-3+x)|<1$ is equivalent to the following:

$$47/16 < x < 49/16.$$

We already know that for values of *x* in the previous interval the series is convergent. Now we need to analyze the behavior of the series at the end points of the interval. We first consider $x = 49/16$:

```
>> maple ('a: = x - > (4 ^(2*n)) * ((x-3) ^ n) / (n + 2)');
>> maple('simplify(a(49/16))')
```

```
ans =

1/(n+2)
```

We first apply convergence tests for non-negative series to see if any of them determine the convergence or divergence of the series.

The ratio, Raabe and root tests do not solve the problem. Next we apply the criterion of comparison of the second kind, comparing the series of the problem with the divergent harmonic series with general term *1/n*:

```
>> maple('a:=n->1/(n+2)');
>> maple('limit(simplify((a(n))^(1/n)), n=infinity)')

ans =

1
```

As the limit is greater than zero, the series is divergent.

We now analyze the behavior of the series at the other end point $x = 47/16$:

```
>> maple ('a: = x - > (4 ^(2*n)) * ((x-3) ^ n) / (n + 2)');
>> maple ('simplify (a(47/16))')

ans =

(- 1) ^ n / (n + 2)
```

We have to analyze the alternating series $\sum_{n=1}^{\infty} \frac{(-1)^n}{n+2}$.

The series with general term $(-1)^n$ has bounded partial sums, and the sequence with general term $1 / (n + 2)$ is decreasing toward 0. Then, by the Dirichlet test, the alternating series converges. Therefore the interval of convergence of the power series is the half-closed interval [47/16, 49/16).

EXERCISE 3-12

Study the range of convergence of the following power series:

$$\sum_{n=0}^{\infty} \frac{1}{(-5)^n} x^{2n+1}$$

We apply the root test:

```
>> maple('a:=n->x^(2*n+1)/(-5)^n');
>> maple('limit(simplify(a(n+1)/a(n)), n=infinity)')

ans =

-1/5 * x ^ 2
```

The series is absolutely convergent when |–*x* ^ 2/5| < 1.

The condition |–*x* ^ 2/5| < 1 is equivalent to *-Sqrt (5) < x < Sqrt (5)*. Thus, we have determined a possible interval of convergence of the power series. We will now analyze the end points:

```
>> maple ('a: = x - > x ^(2*n+1) /(-5) ^ n');
>> [maple('simplify(a(sqrt(5)))'),maple('simplify(a(-sqrt(5)))')]

ans =

(- 1) ^(-n) * 5 ^(1/2)          -(-1) ^ n * 5 ^(1/2)
```

Both series are obviously divergent alternating series. Therefore, the interval of convergence of the power series is the open interval $\left(-\sqrt{5},\sqrt{5}\right)$.

3.5 Formal Power Series

MATLAB implements the package *powseries*, which can be loaded into memory using the command *with(powseries)*. This package provides commands which allow you to create, manipulate and perform symbolic calculations with formal power series. The command ***maple*** must first be used. Among these commands are the following:

powcreate(eqn,eqn1,eqn2,..,eqnn): Creates a formal power series, where *eqn* is an equation of the form *f (n) = expression (n)*, and where the equations *eqni*, which are of the form *f(ni) = value*, define the initial conditions that will give the values of the coefficients of the power series

powpoly (polynomial, variable): Creates a formal power series equivalent to the polynomial given in the specified variable

powadd(ps1,..,psn): Gives the formal power series sum of the specified series

subtract(ps1,ps2): Gives the formal power series subtraction of the specified series

negative (pserie): Gives the additive inverse of the specified power series (i.e. changes its sign)

multconst (pserie, expression): Multiplies each coefficient of the specified power series by the given expression

multiply (ps1, ps2): Gives the product of the two specified power series

quotient (ps1, ps2): Gives the quotient of the two specified power series

powsqrt (pserie): Gives the square root of the specified power series

powexp (pserie): Gives the exponential of the specified power series

powlog (pserie): Gives the natural logarithm of the specified power series

powsin (pserie): Gives the sine of the specified power series

powcos (pserie): Gives the cosine of the specified power series

powdiff (pserie): Gives the derivative of the specified power series

powint (pserie): Gives the indefinite integral of the specified power series

compose (ps1, ps2): Composes the specified power series

reversion (ps1, ps2): Calculates the reversion of the power series ps1 with respect to the power series ps2 (reversion is the inverse of composition)

inverse (pserie): Calculates the multiplicative inverse of the given series

evalpow (exprseries): Returns the power series obtained by evaluating the expression *exprseries*, where the latter is any arithmetic expression involving formal power series, polynomials, or functions that is accepted by the power series package

tpsform (pserie, variable, n): Transforms the formal power series *pserie* into a series of order *n* in the given variable

tpsform (pserie, variable): Transforms the formal series *pserie* into a series in the given variable with order defined by the global variable *Order*

op (pserie): Returns the internal structure of the formal power series *pserie* and gives access to its operands

op (op (pserie) 4): Returns a table that defines all the coefficients of the formal power series *pserie*. Any operand can be replaced with *subsop*.

Here are some examples:

```
>> maple('with(powseries):powcreate(t(n)=1/n!,t(0)=1):')
>> pretty(simple(sym(maple('tpsform(t, x, 7)'))))

               2         3          4            5             6       7
1 + x + 1/2 x   + 1/6 x   + 1/24 x   + 1/120 x   + 1/720 x   + O(x )

>> maple('powcreate(v(n)=(v(n-1)+v(n-2))/4,v(0)=4,v(1)=2): ')
>> pretty(simple(sym(maple('tpsform(v, x) '))))

                 2         3   19  4   47  5        6
4 + 2 x + 3/2 x   + 7/8 x   + -- x   + --- x   + O(x )
                              32       128
>> maple('with(powseries): ')
>> maple('t:=powpoly(2*x^5+4*x^4-x+5, x): ')
>> pretty(simple(sym(maple('tpsform(t, s, 5) '))))

           4       5
5 - s + 4 s   + O(s )

>> pretty(simple(sym(maple('tpsform(t, s, 4)'))))

           4
5 - s + O(s )

>> maple('powcreate(t(n)=t(n-1)/n,t(0)=1): ')
>> maple('powcreate(v(n)=v(n-1)/2,v(0)=1): ')
>> maple('s := powadd(t, v): ')
>> pretty(simple(sym(maple('tpsform(s, x, 7) '))))
```

```
                    2           3           4     19  5      49  6       7
2 + 3/2 x + 3/4 x  + 7/24 x  + 5/48 x  + --- x  + ---- x  + O(x )
                                          480       2880
>> maple('s := multiply(t, v): ')
>> pretty(simple(sym(maple('tpsform(s, x) '))))

                    2   19  3          4   109  5       6
1 + 3/2 x + 5/4 x  + -- x  + 7/16 x  + --- x  + O(x )
                     24                480

>> maple('s := powexp(x): ')
>> pretty(simple(sym(maple('tpsform(s, x, 7) '))))

               2         3          4           5            6       7
1 + x + 1/2 x  + 1/6 x  + 1/24 x  + 1/120 x  + 1/720 x  + O(x )

>> maple('t := powexp( exp(x) ): ')
>> pretty(simple(sym(maple('tpsform(t, x, 4) '))))

                                       2                  3       4
exp (1) + exp (1) x + exp (1) x + 5/6 exp (1) x + or (x)

>> maple('u := powexp( powdiff( powlog(1+x) ) ): ')
>> pretty(simple(sym(maple('tpsform(u, x, 5) '))))

                                       2                  3    73          4     5
exp(1)- exp(1) x + 3/2 exp(1) x  - 13/6 exp(1) x  +  -- exp(1) x + O(x)
                                                      24
>> maple('powcreate(t(n)=t(n-1)/n,t(0)=0,t(1)=1): ')
>> maple('powcreate(v(n)=v(n-1)/2,v(0)=0,v(1)=1): ')
>> maple('s := reversion(t,v): ')
>> pretty(simple(sym(maple('tpsform(s,x,11) '))))

          3          5            7             9       11
x + 1/12 x  + 1/80 x  + 1/448 x  + 1/2304 x  + O(x   )

>> maple('ts := compose(t,s): ')
>> pretty(simple(sym(maple('tpsform(ts,x) '))))

         2         3         4          5       6
x + 1/2 x  + 1/4 x  + 1/8 x  + 1/16 x  + O(x )

>> pretty(simple(sym(maple('tpsform(v,x) '))))

         2         3         4          5       6
x + 1/2 x  + 1/4 x  + 1/8 x  + 1/16 x  + O(x )

>> maple('powcreate(f(n)=f(n-1)/n,f(0)=1): ')
>> maple('powcreate(g(n)=g(n-1)/2,g(0)=0,g(1)=1): ')
>> maple('powcreate(h(n)=h(n-1)/5,h(0)=1): ')
>> maple('k:=evalpow(f^3+g-quotient(h,f)): ')
>> pretty(simple(sym(maple('tpsform(k,x,5) '))))
```

```
          233  2  7273  3  52171  4      5
24/5 x + --- x + ---- x + ----- x + o (x)
          50      1500      15000
```

3.6 Power Series Expansions and Functions

A local approximation of a real function of a real variable at a point replaces the definition of the function in a neighborhood of the point by a simpler function. The most commonly used local approximations replace an arbitrary function *f(x)* by a polynomial *p(x)* (a truncated power series), so that for any *x* close to the point, *f(x)* is close to *p(x)*.

MATLAB implements several commands which can be used to work with local approximations. The command ***maple*** must first be used. The syntax of these commands is presented below:

series(expr,var=a): Returns a truncated power series expansion of expr in a neighborhood of the point *a*. The series is truncated in the order specified by the global variable *Order*. The expansion can be Taylor, Laurent or other more generalized series. The expansion may have fractional exponents, in which case the series will be represented in ordinary sum-of-products form.

series (expr, var): Equivalent to the above with *a* = 0

series (int (expr, var) var = a): Returns a generalized series expansion of the specified integral in a neighborhood of the point *a*

series (leadterm (expr), var = a): Returns the smallest degree term in the generalized series expansion of *expr* in a neighborhood of *a*

series(expr,var=a,n): Returns the generalized series expansion of *expr* in a neighborhood of *a* up to order *n*. In order to evaluate the expansion at some input, it must first be converted to a polynomial using the command *convert (polynom).*

convert (s, polynom): Converts the series *s* to a polynomial, eliminating the order term

convert (s, ratpoly): Converts the series *s* to a rational polynomial expression. If *s* is a Taylor or Laurent series, then the rational polynomial expression is the Padé approximation. If *s* is a Chebyshev series, then the rational polynomial expression is the Chebishev-Padé approximation.

convert(s,ratpoly,m,n): Converts the series *s* to a rational polynomial where the numerator polynomial has degree *m* and the denominator polynomial has degree *n*

op (s): Shows the internal structure of the series *s* and allows access to its operands. It returns a sequence of *2n* operands where *n* is the number of terms of the series, including the order. The i^{th} coefficient is the $2i\text{-}1^{th}$ term and the i^{th} exponent is the $2i^{th}$ term. If the expansion is in a neighborhood of *x* = *a*, *op(0,series)* extracts the expression *var* = *a*. Any operand can be substituted with *subsop.*

order (s): Determines the order of the truncated expansion of the series *s*

Order: = n sets the global variable *Order* to the value *n*. By default, the value is 6. This variable determines the order of all future truncated series expansions.

Order returns the current value of the order at which all truncated series expansions are given. The default order is 6.

type (expr, series): Determines whether the expression is a series

type (expr, taylor): Determines whether the expression is a Taylor series

type (expr, laurent): Determines whether the expression is a Laurent series

Here are some examples:

```
>> pretty(simple(sym(maple('series(x/(1-x-x^2), x=0) '))))

                    2      3      4      5      6
             x + x  + 2 x  + 3 x  + 5 x  + O(x )

>> pretty(simple(sym(maple('series(exp(x)/x, x=0, 8 ) '))))

 -1                    2         3           4          5           6     7
x   + 1 + 1/2 x + 1/6 x  + 1/24 x  + 1/120 x  + 1/720x + 1/5040 x + O(x)

>> pretty(simple(sym(maple('series(GAMMA(x), x=0, 2 ) '))))

           -1                    2             2          2
          x   - gamma + (1/12 Pi  + 1/2 gamma ) x + O(x )

>> pretty(simple(sym(maple('series(x^3/(x^4+4*x-5),x=infinity) '))))

                       4      5        1
                1/x - ---- + ---- + O(----)
                        4      5        7
                       x      x        x

>> pretty(simple(sym(maple('int(exp(x^3), x ) : series(", x=0) '))))

                            4      7
                   x + 1/4 x  + O(x )

>> pretty(simple(sym(maple('series(x^x, x=0, 3) '))))

                                       2  2      3
             1 + ln(x) x + 1/2 ln(x)  x  + O(x )

>> pretty(simple(sym(maple('convert(" ,polynom) '))))

                                         2  2
                 1 + ln(x) x + 1/2 ln(x)  x

>> pretty(simple(sym(maple('series(exp(x), x) '))))

                    2        3         4          5      6
        1 + x + 1/2 x  + 1/6 x  + 1/24 x  + 1/120 x  + O(x )
```

```
>> pretty(simple(sym(maple('convert(", ratpoly) '))))

                         2         3
         1 + 3/5 x + 3/20 x  + 1/60 x
         ----------------------------
                                 2
              1 - 2/5 x + 1/20 x
```

3.7 Taylor, Laurent, Padé and Chebyshev Expansions

Taylor's theorem gives us a way of approximating a function *f(x)* by a polynomial at a point $x = x_0$. The series that is truncated to give such a polynomial is called the Taylor series of the function *f(x)* at the point $x = x_0$. The Taylor series of *f(x)* at $x = 0$ is called the MacLaurin series of *f(x)*.

Taylor's theorem says that if *f(x)* is an $n + 1$times differentiable function defined on an interval that *contains the point a*, then *f(x)* can be approximated in a neighborhood of *a* by the following polynomial:

$$f(x)=\sum_{k=0}^{n}\frac{f^{(k)}(a)}{k!}(x-a)^k+\frac{f^{(n+1)}(z)}{(n+1)!}(x-a)^{n+1}$$

where *z* is a number between *a* and *x*.

The above expression is the Taylor series expansion of *f(x)* at the point $x = a$. The last term of the expression is the remainder term, and represents a measure of the error of the approximation.

The Laurent expansion allows a finite number of negative powers in the series expansion of a function.

The Padé expansion approximates the function as a ratio of polynomials whose numerator and denominator have fixed degrees *m* and *n*.

The Chebyshev expansion approximates the function using Chebyshev polynomials. The *Chebyshev-Padé expansion* mixes both types.

When you use the command "series", without further specification, MATLAB uses by default the type of expansion most suitable in each case. There are commands that allow you to specify which type of expansion should be used (all require the prior use of the *maple* command). The syntax is as follows:

taylor(expr,var=a): Returns the Taylor series expansion of the expression expr at the point *a* where the order is determined by the global variable *Order*

taylor (expr, variable): Returns the Taylor series expansion of the expression expr in a neighborhood of the origin (i.e. the MacLaurin expansion)

taylor(expr,var=a,n): Returns the Taylor expansion of expr at *a* up to order *n*.

taylor(int,var=a): Returns the Taylor expansion of the indefinite integral int in a neighborhood of *a*

taylor(leadterm(expr),var=a): Returns the lowest degree term in the Taylor series expansion of expr in a neighborhood of *a*

readlib (mtaylor): Loads into the memory a package that allows you to work with multivariate Taylor expansions

mtaylor (expression, [var1,..., varn] or mtaylor(expression,{var1,...,varn})):

Returns the multivariate Taylor expansion of *expr* given in the specified variables up to the order determined by the global variable *Order*

mtaylor(expression,[var1,...,varn],n):

Returns the multivariate Taylor expansion of the specified expression up to order *n*

mtaylor(expression,[var1,...,varn],n,[w1,...,wn]):

Returns the multivariate Taylor expansion of the given expression by assigning weights *w1* to *wn* to the variables. The default weight is 1. A weight of 2 will halve the order of the corresponding variable to which the expansion is given.

readlib (coeftayl): Loads into the memory a package that allows you to find coefficients of Taylor expansions

coeftayl(expr,variable=a,n): Returns the coefficient of x^n in the Taylor expansion of *expr* in a neighborhood of the point *a*.

coeftayl(expr,[var1,..,varm]=[a1,...,an],[n1,...,nm]):

Returns the coefficient of the term $(var1\text{-}a1)^{n1}$ *...* $(varm\text{-}am)^{nm}$ in the Taylor expansion of *expr* in a neighborhood of the point *(a1,..., am)*

readlib (poisson): Loads into the memory a package that allows you to work with multivariate Taylor series in Poisson form

poisson(expression,[var1,...,varn]) or poisson(expr,{var1,...,varn}):

Returns the multivariate Taylor expansion of the expression in Poisson form. The Poisson form combines the terms of the series in terms of sines and cosines, forming the Fourier canonical form. The order of the expansion is specified by the global variable *Order.*

poisson(expr,[var1,...,varn],n):

Returns the multivariate Taylor expansion of *expr* in Poisson form up to order *n*

poisson(expr,[var1,...,varn],n,[w1,...,wn]):

Returns the multivariate Taylor expansion of order *n* of *expr* in Poisson form using the specified weights

chebyshev(expr,var=a..b):

Returns the Chebyshev expansion of an analytical expression *expr* in the interval *[a, b]*. The result is a sum based on orthogonal Chebyshev polynomials in *t*. The orthopoly package needs to be loaded via the command *with(orthopoly,t)* before applying the Chebyshev command.

chebyshev(expr,var): Returns the Chebyshev expansion of an analytical expression *expr* in the interval *[- 1,1]*

chebyshev(expr,var=a..b, error):

Returns the Chebyshev expansion with the specified tolerance, which by default is *10* ^*(-Digits)*

asymt (expr, var): Returns the asymptotic expansion *(var* -> *infinity)* of the given expression expr up to the order specified by the global variable *Order*

asymt (expr, var, n): Returns the asymptotic expansion *(var* -> *infinity)* of the given expression expr up to order *n*

with (numapprox): Loads into the memory a package that allows you to work with Padé, Padé-Chebyshev and Laurent expansions

chebdeg (series): Returns the degree of the specified Chebyshev series

chebmult (s1, s2): Returns the product of the two specified Chebychev series in the same variable

chebpade(expr,var=a..b,[n,d]): Finds the Chebyshev-Padé approximation of the given expression *expr* in the specified variable, in the interval *[a, b]*, with numerator and denominator polynomials of degrees *n* and *d* respectively

chebpade(expr,var,[n,d]): Returns the Chebyshev-Padé approximation of the given expression *expr* in the specified variable, in the interval *[- 1,1]*, with numerator and denominator polynomials of degrees *n* and *d* respectively

chebpade(expr,var=a.. b, [n, 0]) or chebpade(expr,var=a..b, n): Simply returns the Chebyshev polynomial of degree *n* in *[a, b]*

chebsort (series): Orders the terms of the given Chebyshev series

pade(expr,var=a,[n,d]): Returns the Padé approximation of the given expression *expr* in the specified variable at the point *a*, with numerator and denominator polynomials of degrees *n* and *d* respectively

pade(expr,var,[n,d]): Returns the Padé approximation of the given expression *expr* in the specified variable, at 0, with numerator and denominator polynomials of degrees *n* and *d* respectively

pade(expr,var=a,[n,0]) or pade(expr,var=a,n):

Simply returns the Taylor series expansion up to order n of the expression at the given point

laurent(expr,var=a,n):

Returns the Laurent series of the given expression *expr* at the given point up to order *n*

laurent(expr,var,n): Returns the Laurent series of the given expression *expr* at the point 0 up to order *n*

readlib (eulermac) loads into the memory a package whose commands allow you to find Euler-MacLaurin expansions

eulermac(expr,var,n): Finds the Euler-MacLaurin expansion of the given expression up to degree *n* in the given variable

eulermac (expr, var): Finds the Euler-MacLaurin expansion of the given expression up to the degree specified by the global variable *Order*

EXERCISE 3-13

Find the Taylor series up to 13th order of *sinh (x)* at the point $x = 0$. Also find the Taylor expansion of $1/(1+x)$ up to 10th order at the point $x = 0$. Find the corresponding Laurent and Euler-McLaurin series. Find the Chebychev-Padé approximation of degree (4,3) in the interval [0,1].

```
>> pretty(simple(sym(maple('taylor(sinh(x),x=0,13) '))))
```

```
         3         5          7            9              11      13
x + 1/6 x + 1/120 x + 1/5040 x  + 1/362880 x  + 1/39916800 x   + O(x  )
```

>> pretty(simple(sym(maple('taylor(1/(1+x),x=0,10) '))))

```
         2    3    4    5    6    7    8    9     10
1 - x + x  - x  + x  - x  + x  - x  + x  - x  + O(x  )
```

>> maple('with(numapprox): ')
>> pretty(simple(sym(maple('laurent(sinh(x),x=0,13) '))))

```
         3         5          7            9              11      13
x + 1/6 x + 1/120 x + 1/5040 x  + 1/362880 x  + 1/39916800 x   + O(x  )
```

>> pretty(simple(sym(maple('laurent(1/(1+x),x=0,10) '))))

```
         2    3    4    5    6    7    8    9     10
1 - x + x  - x  + x  - x  + x  - x  + x  - x  + O(x  )
```

Note that the Laurent and Taylor series agree in this case.

>> maple('readlib(eulermac) : ')
>> pretty(simple(sym(maple('eulermac(sinh(x),x,13) '))))

```
314416268077
------------ cosh(x) - 1/2 sinh(x) + O(cosh(x))
290594304000
```

>> pretty(simple(sym(maple('eulermac(1/(1+x),x,10) '))))

```
                 1             1                1
ln(1 + x) - 1/2 ----- - 1/12 -------- + 1/120 --------
                1 + x              2                4
                            (1 + x)          (1 + x)

               1                1                 1             1
     - 1/252 -------- + 1/240 -------- - 1/132 --------- + O(---------)
                    6                8                10            12
             (1 + x)          (1 + x)          (1 + x)       (1 + x)
```

>> maple(' with(numapprox): ')
>> pretty(simple(sym(maple(' with(orthopoly,T):chebpade(sinh(x),x=0..1,[4,3]) '))))

```
                                                                2
(.03672350156 + .9734745062 x - .02546155828 (2. x - 1.)

                                  3                                  4
     + .009531362262 (2. x - 1.)  - .001730582994 (2. x - 1.) )  /
                                                                     2
    (1.157361890 - .3056450150 x - .009078765811 (2. x - 1.)

                                  3
     + .001929812884 (2. x - 1.) )
```

```
>> pretty(simple(sym(maple('with(orthopoly,T):chebpade(1/(1+x),x=0..1,[4,3]) '))))

         .6666666665
--------------------------
.6666666666 + .6666666668 x
```

EXERCISE 3-14

Find the Laurent expansion of 1 / *(x sin (x))* at the point *x* = 0 up to degree 10. Also find the Padé approximation of degree (3,4) at the same point and the Taylor expansion of *log (x)* at the point *x* = 2 up to degree 5.

```
>> pretty(simple(sym(maple('with(numapprox):laurent(1/(x*sin(x)),x=0,10) '))))

   -2                 2    31     4    127     6      7
  x   + 1/6 + 7/360 x  + ----- x  + ------ x  + O(x )
                         15120      604800

>> pretty(simple(sym(maple('with(numapprox):pade(1/(x*sin(x)),x=0,[3,4]) '))))

                                          2
                                -60 - 3 x
                               ------------
                                   4      2
                               7 x  - 60 x

>> pretty(simple(sym(maple('taylor(log(x),x=2,4) '))))

                                2                3              4
 ln(2) + 1/2 (x - 2) - 1/8 (x - 2)  + 1/24 (x - 2)  + O((x - 2) )
```

EXERCISE 3-15

Find the Taylor series at the origin, both in its normal form and in its Poisson form, up to order 5, of the function:

$$f(x,y) = e^{x+y^2}$$

```
>> maple('readlib(mtaylor): ')
>> pretty(simple(sym(maple('mtaylor(exp(x+y^2),[x,y],5) '))))

          2          2       2         3         4        2  2          4
 1 + x + y  + 1/2 x  + x y  + 1/6 x  + 1/2 y  + 1/2 y  x  + 1/24 x

>> maple('readlib(poisson): ')
>> pretty(simple(sym(maple('poisson(exp(x+y^2),[x,y],5) '))))
```

```
           2        2       2        3        4        2  2        4
 1 + x + y  + 1/2 x  + x y  + 1/6 x  + 1/2 y  + 1/2 y  x  + 1/24 x
```

We see that the two forms coincide.

EXERCISE 3-16

Find the Taylor expansion of the function 1 /(2–*x*) at the point *x* = 1 up to order 10. Also find the Taylor expansion of *sin*(*x*) at *pi*/2 up to order 8, and the same for the function *log* (*x*) at the point *x* = 2 up to degree 5.

```
>> pretty(simple(sym(maple('taylor(1/(2-x),x=1,10)'))))

1 + x - 1 + (x - 1)  + (x - 1)  + (x - 1)  + (x - 1)  + (x - 1)  + (x - 1)  +

           8          9           10
    (x - 1)  + (x - 1)  + O((x - 1)  )

>> pretty(simple(sym(maple('taylor(sin(x),x=pi/2,8)'))))
                       2                   4                    6               8
1 - 1/2(x - 1/2 pi)  + 1/24(x - 1/2 pi)  - 1/720(x - 1/2 pi) + O((x - 1/2 pi) )

>> pretty(simple(sym(maple('taylor(log(x),x=2,5)'))))

                                  2               3               4            5
log(2) + 1/2 (x - 2) - 1/8 (x - 2)  + 1/24 (x - 2)  - 1/64 (x - 2)  +  O((x - 2) )
```