МЕТОД НАИМЕНЬШИХ КВАДРАТОВ В MATLAB

В MATLAB для приближения данных в смысле наименьших квадратов используется функция polyfit, во входных аргументах которой указываются вектора с данными, а выходным является вектор коэффициентов полинома, начиная со старшей степени. Функция polyfit может быть вызвана и с большим числом входных и выходных аргументов, что может потребоваться для улучшения качества приближения и получения некоторой дополнительной информации о нем. Подробнее об этом написано в разделах алгоритм, решение системы, когда появляются предупреждения, центрирование и масштабирование и как использовать polyval.

Поставим задачу приблизить данные, которые заданы массивами x и y:

```
>> x = [0.1 0.3 0.45 0.5 0.79 1.1 1.89 2.4 2.45];
>> y = [-3 -1 0.9 2.4 2.5 1.9 0.1 -1.3 -2.6];
```

полиномами первой, третьей и пятой степени. Используем функцию polyfit, указав в ее входных аргументах вектора с данными и степени полиномов. Сами коэффициенты запишем в вектора p1, p3 и p5, соответственно:

```
>> p1 = polyfit(x, y, 1)
p1 =
   -0.6191    0.6755
>> p3 = polyfit(x, y, 3)
p3 =
    2.2872  -12.1553   17.0969   -4.5273
>> p5 = polyfit(x, y, 5)
p5 =
   -6.0193   33.9475  -62.4220   35.9698    4.7121   -3.8631
```

Итак, мы получили полиномы:

$$p^{(1)}(x) = -0.6191 \cdot x + 0.6755$$
$$p^{(3)}(x) = 2.2872 \cdot x^3 - 12.1533 \cdot x^2 + 17.0969 \cdot x - 4.5273$$
$$p^{(5)}(x) = -6.0193 \cdot x^5 + 33.9475 \cdot x^4 - 62.4220 \cdot x^3 + 35.9698 \cdot x^2 + 4.7121 \cdot x - 3.8631$$

Для построения графиков этих полиномов следует найти их значения в промежуточных точках, принадлежащих интервалу, на котором заданы данные, т.е. между x(1) и x(end). Сгенерируем 100 точек, равномерно расположенных на области определения данных, при помощи функции linspace

```
>> xx = linspace(x(1), x(end), 100);
```

вычислим в них значения полиномов $p^{(1)}(x), p^{(3)}(x), p^{(5)}(x)$ при помощи функции polyval и запишем найденные значения в вектора yy1, yy3, yy5 соответственно. Входными аргументами функции polyval в самом простом случае являются вектор коэффициентов полинома и вектор значений независимой переменной, для которых требуется вычислить значения полинома, а выходным - вектор значений полинома

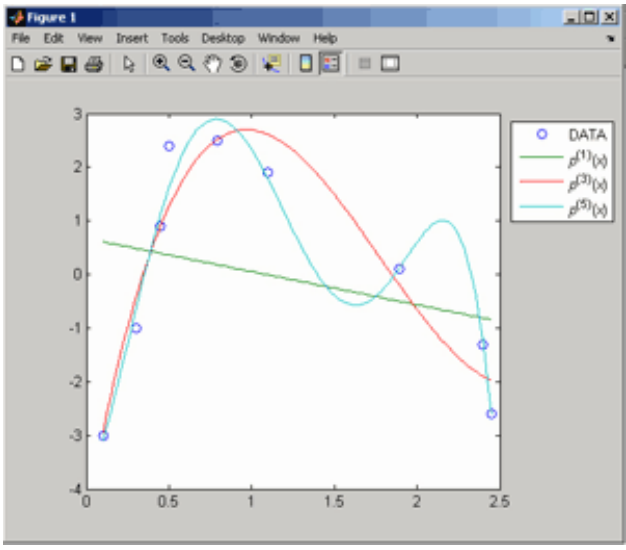
```
>> yy1 = polyval(p1, xx);
>> yy3 = polyval(p3, xx);
>> yy5 = polyval(p5, xx);
```

Для наглядности построим графики полиномов и разместим заданные массивами x и y точки круглыми маркерами

```
>> plot(x, y, 'o', xx, yy1, xx, yy3, xx, yy5)
```

Кроме этого, при помощи функции legend разместим справа от осей легенду, в которой укажем соответствие линий полиномам:

```
>> legend('DATA', '{\itp}^{(1)}({\itx})', '{\itp}^{(3)}({\itx})', '{\itp}^{(5)}({\itx})',-1)
```



При создании легенды использовались возможности интерпретатора TeX, приведенный ниже.

Для того, чтобы узнать, насколько далеко отстоит график полинома от заданных точек, т.е. какая была допущена ошибка при приближении данных, следует вызывать функцию polyfit с двумя выходными аргументами. В первый из них запишется вектор коэффициентов построенного полинома, а во второй структура с информацией о приближении, например:

```
>> [p3, S3] = polyfit(x, y, 3)
```

приводит к выводу в командное окно того же самого вектора коэффициентов кубического полинома, что получился в предыдущем примере, и структуры S3 с полями R, df и normr:

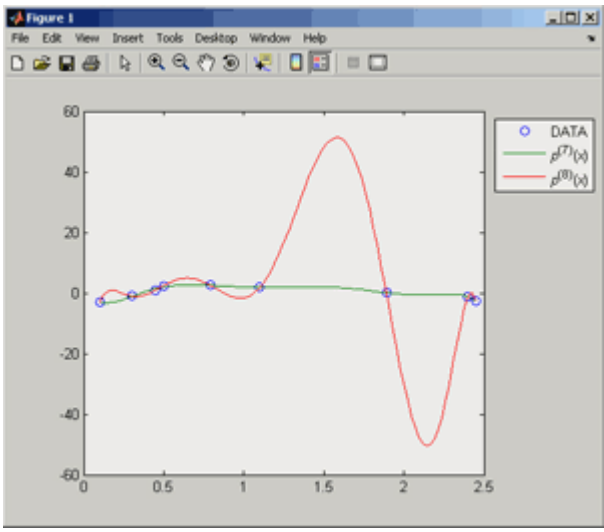
```
p3 =
   2.2872  -12.1553   17.0969   -4.5273
S3 =
      R: [4x4 double]
     df: 5
  normr: 1.7201
```

Поле normr содержит ошибку в среднеквадратичной норме, т.е. значение

$$\sqrt{\min_{p_1,p_2,\ldots,p_{n+1}} \sum_{i=1}^{N} \left(p^{(n)}(x_i) - y_i\right)^2}$$

(если работа со структурами незнакома, то достаточно понять, что для записи содержимого поля структуры в некоторую переменную надо отделять имя структуры от имени поля точкой, т.е. >> r = S3.normr). В поле df содержится разность между числом точек и числом коэффициентов полинома.

Повышение степени полинома приводит к уменьшению ошибки, однако качество приближения не всегда улучшается. Например, для наших данных полином восьмой степени обеспечивает практически нулевую ошибку (вычислительные погрешности возникают в ходе алгоритма нахождения коэффициентов)




```
>> [p8, S8] = polyfit(x, y, 8)
S8 =
      R: [9x9 double]
     df: 0
  normr: 2.2382e-010
```

однако приближает данные намного хуже, чем полином седьмой степени, что можно видеть из графика, на котором представлены данные и два приближения:

```
>> [p7, S7] = polyfit(x, y, 7)
>> yy7 = polyval(p7, xx);
>> yy8 = polyval(p8, xx);
>> plot(x, y, 'o', xx, yy7, xx, yy8)
>> legend('DATA', '{\itp}^{(7)}({\itx})', '{\itp}^{(8)}({\itx})',-1)
```

Так происходит потому, что полином восьмой степени является интерполяционным полиномом, который как правило, плохо подходит для приближения данных.

Дальнейшее увеличение степени полинома ни к чему хорошему не приведет, например при построении полинома девятой степени

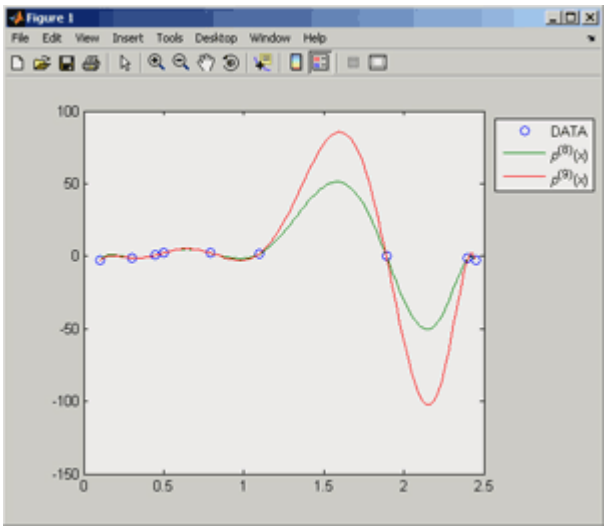
```
>> [p9, S9] = polyfit(x, y , 9)
```

выведется предупреждение о том, что такой полином не единственный

Warning: Polynomial is not unique; degree >= number of data points.

Приближение полиномом девятой степени окажется еще хуже, чем интерполяция полиномом восьмой степени:

```
>> yy9 = polyval(p9, xx);
>> plot(x, y, 'o', xx, yy8, xx, yy9)
>> legend('DATA', '{\itp}^{(8)}({\itx})', '{\itp}^{(9)}({\itx})',-1)
```



ПОЧЕМУ ВОЗНИКАЕТ ЗАДАЧА О ПРИБЛИЖЕННОМ НАХОЖДЕНИИ КОРНЕЙ

Предположим, нам задано уравнение вида $f(x) = 0$, скажем

$x^2 + 3x - 1 = 0$.

Требуется найти корни уравнения, т.е. такие величины x_*, что данное уравнение обращается в тождество, именно:

$x_*^2 + 3x_* - 1 = 0$.

Разумеется, в этом примере нет ничего сложного — мы просто решаем квадратное уравнение

$ax^2 + bx + c = 0$

по известным со школы формулам и находим ровно два корня (при обозначении корней уравнения иногда будем использовать индексы):

$$x_1 = \frac{-b - \sqrt{D}}{2a}, \quad x_1 = \frac{-b + \sqrt{D}}{2a}, \quad D = b^2 - 4ac.$$

Корни могут и совпадать, если дискриминант D равен нулю, тогда будем говорить, что корень кратный (для квадратного уравнения кратности два). Обычно, когда говорят, что у уравнения сколько-то корней, то при этом учитывают их кратности.

В случае кубических уравнений ситуация не намного сложнее. Как было установлено в середине XVI века, уравнение $ax^3 + bx^2 + cx + d = 0$ приводится к уравнению $x^3 + px + q = 0$, а его решение находится по формулам Кардано. Формулы для нахождения корней кубического уравнения, как и в случае квадратного уравнения, содержат радикалы, т.е. величины вида $\sqrt[n]{t}$. Далее было показано, что уравнение 4-ой степени может быть сведено к решению квадратного и кубического уравнений. Однако, уравнения выше четвертой степени в общем случае не разрешимы в радикалах, что было доказано норвежским математиком Нильсом Хенриком Абелем в 1826г.

Поскольку нет формул для корней полиномов выше четвертой степени, то возникает задача о приближенном нахождении корней. Их число заранее известно. Как утверждает основная теорема алгебры (теорема Гаусса, 1799г.), число всех вещественных и комплексных корней алгебраических уравнений с учетом их кратностей равно степени полинома. Для нахождения корней полиномов существуют специальные численные методы, которые позволяют приближенно найти все корни полинома.

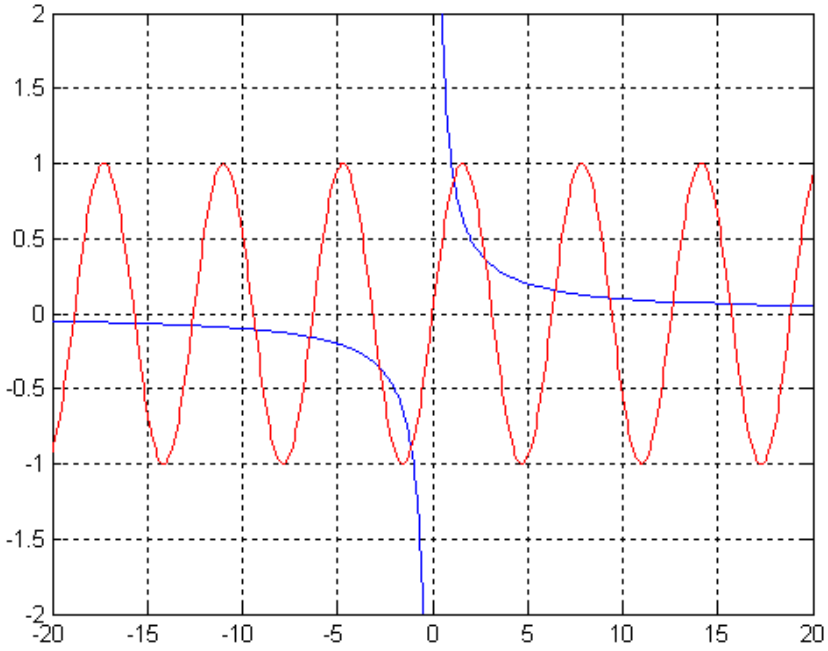
В случае трансцендентных уравнений число корней может быть любым, их может быть и бесконечно много. В этом убеждает простой пример, в котором требуется найти корни уравнения

$$\sin x - \frac{1}{x} = 0.$$

Действительно, корни данного уравнения являются абсциссами бесконечного числа точек пересечения синусоиды и гиперболы. Построим графики в MATLAB, используя функцию fplot. Ее первым входным аргументом является выражение для функции, записанное в апострофах, а вторым — отрезок, на котором строится график. Третий входной аргумент может задавать цвет линии, в нашем примере вторая линия рисуется красным цветом:

```
figure
axes
```

```
fplot('1/x', [-20 -0.5])
hold on
fplot('1/x', [0.5 20])
fplot('sin(x)', [-20 20], 'r')
grid on
```



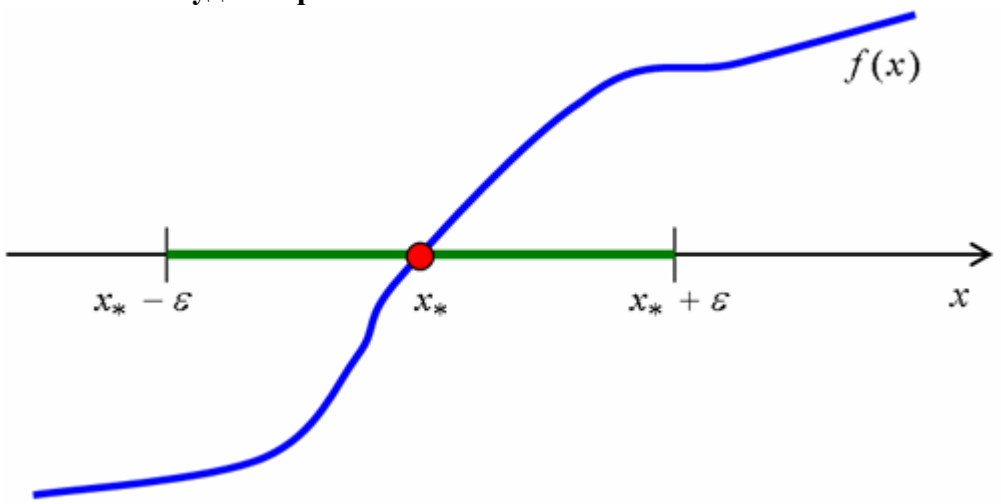
Графики функций sin *x* и $\frac{1}{x}$.

Уравнение может и вовсе не иметь корней или корни могут быть комплексными. Итак, в общем случае нахождение корней уравнений представляет целое исследование. Но прежде чем решать уравнение необходимо уточнить, что значат слова «приближенно найти корень уравнения» и какие проблемы с этим связаны.

В ЧЕМ ЗАКЛЮЧАЕТСЯ ЗАДАЧА О ЧИСЛЕННОМ РЕШЕНИИ УРАВНЕНИЙ

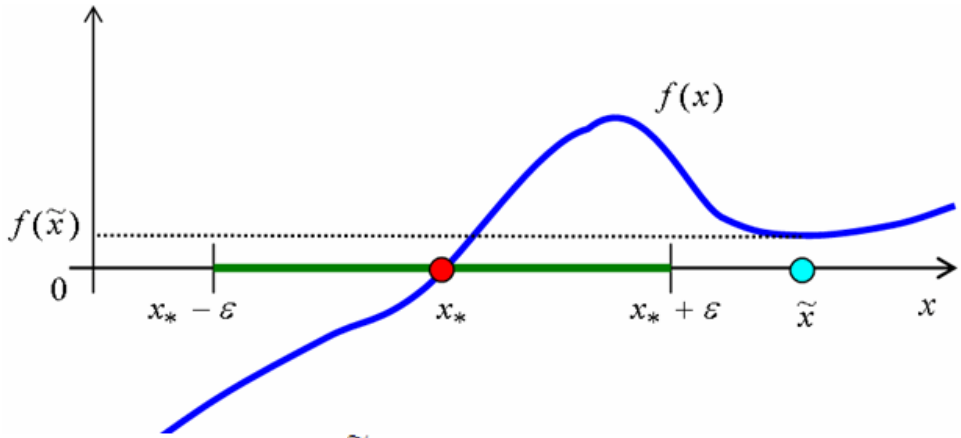
В предыдущем разделе мы заметили, что в общем случае у уравнения вида $f(x) = 0$ может быть сколько угодно корней. Определим теперь, что значит приближенно найти какой-то корень. Обычно требуется найти корень с заданной точностью, обозначим ее ε. На практике это значение может быть каким угодно, т.е. например 10^{-3}, 10^{-11} и т.д. Предположим, что x_* является точным корнем уравнения $f(x) = 0$, т.е $f(x_*) = 0$. Разумеется, значение x_* заранее неизвестно. Тогда задача о нахождении корня x_* с точностью ε заключается в нахождении какой-либо точки $\tilde{x}$, которая отстоит от точки x_* не более чем на ε.

Сказанное демонстрируется на рисунке ниже. Для нас **любая точка $\tilde{x}$ из отрезка $[x_* - \varepsilon, x_* + \varepsilon]$ будет корнем с точностью ε.**



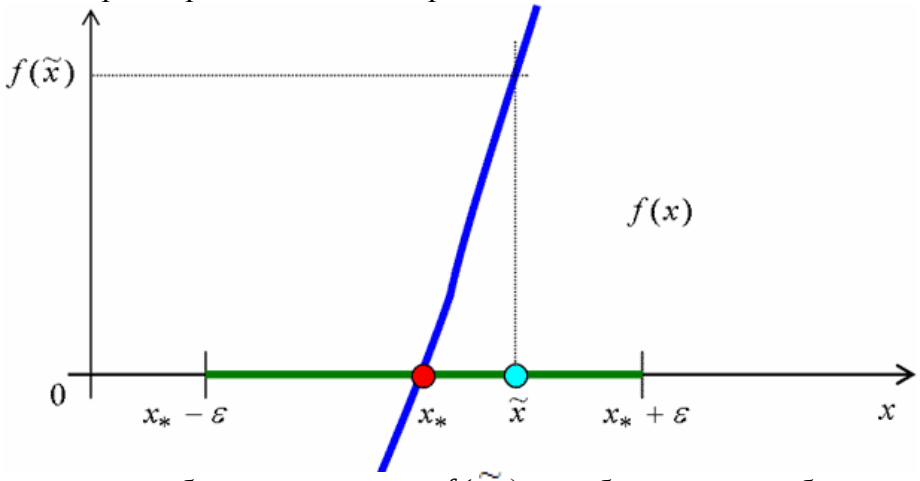
Задача о нахождении корня уравнения с заданной точностью.

Такая постановка задачи порождает сразу несколько вопросов. Первый из них — предположим, что в результате работы некоторого алгоритма мы нашли точку $\tilde{x}$, как тогда определить ее расстояние до точного корня x_*, значение которого заранее неизвестно. Достаточно маленькое значение $f(\tilde{x})$ вообще говоря не значит, что точка $\tilde{x}$ близко к корню x_*. Точно также, достаточно большое значение $f(\tilde{x})$ вообще говоря не значит, что точка $\tilde{x}$ далеко от корня x_*. Здесь в слова «достаточно маленькое значение» и «достаточно большое значение» не вкладывается точный смысл. Но, например, если $f(\tilde{x}) < \varepsilon$, то это не вовсе не значит, что точка $\tilde{x}$ отстоит от x_* не больше, чем на ε. Действительно, график функции может проходить близко к оси абсцисс вдали от корня. В этом убеждает пример, приведенный ниже на рисунке, где $f(\tilde{x}) < \varepsilon$, но $\tilde{x}$ не является корнем с точностью ε.



Пример, в котором малость $f(\tilde{x})$ ошибочно может быть принята за близость $\tilde{x}$ к x_*.

Для достаточно большого значения $f(\tilde{x})$ так же можно привести пример, в котором функция быстро возрастает вблизи корня



Пример, в котором большое значение $f(\tilde{x})$ ошибочно может быть принята за большое удаление $\tilde{x}$ от x_* .

Второй вопрос — надо убедиться, что в ε-окрестности корня x_* нет других корней. Иначе, если мы найдем $\tilde{x}$, отстоящий отx_* не более, чем на ε (и как-то проверим это), то будем ошибочно думать, что больше вблизи $\tilde{x}$ корней нет.

Для успешного решения задачи о нахождении корня необходимо понимать перечисленные выше проблемы и, кроме того, знать алгоритм, заложенный в используемой функции MATLAB.

Обычно, перед уточнением корня с заданной точностью стараются найти такие отрезки, каждый из которых содержит ровно один корень. Эта задача называется задачей об отделении корней, ей посвящен следующий раздел.

ОТДЕЛЕНИЕ КОРНЕЙ

Для алгебраических уравнений известны теоремы, позволяющие установить верхнюю и нижнюю границы отрезка, на котором расположены корни, а также определить число действительных корней на заданном отрезке. В этом разделе мы рассмотрим подходы к отделению корней (т.е. нахождению отрезков, содержащих ровно один корень) для произвольных уравнений.

По теореме Больцано-Коши, непрерывная функция, принимающая на границах отрезка значения разных знаков, имеет на этом отрезке хотя бы один корень. Следствием из этой теоремы является то, что для непрерывных монотонных на некотором отрезке функций, изменяющих знак на его границах, существует ровно один корень.

Важно понимать, что и условие смены знака и условие непрерывности одинаково важны. Невыполнение одного из условий может привести к совершенно неверному результату. Приведем простой пример уравнения

$$\frac{1}{x-\sqrt{2}}=0.$$

Ясно, что данное уравнение не имеет корней, на отрезке [1, 2] его левая часть изменяет знак, но непрерывности у левой части уравнения на этом отрезке нет. Попытаемся решить данное уравнение при помощи функции fzero (про использование функции fzero написано в разделе Основные способы обращения к fzero и следующих за ним):

```
x = fzero('1/(x-sqrt(2))', [1 2])
x = 1.4142
```

Вроде в этом примере функция fzero выдала результат и все замечательно, за исключением того, что нашлась точка разрыва (корней нет). Почему так происходит, станет понятно при детальном рассмотрении алгоритма, заложенного в функцию fzero. Ему посвящен раздел Алгоритм, по которому работает fzero.

Конечно, не всегда можно воспользоваться приведенными выше утверждениями относительно существования корня и его единственности, поскольку функция может оказаться такой, что исследование на непрерывность или на монотонность достаточно сложное. Однако, есть случаи, когда исследование вполне можно провести.

Например, функция $f(x) = x^3 + x + 1$ знак на границах отрезка [-1, 1], непрерывна на этом отрезке и монотонна, т.к. $f'(x)=3x^2+1>0$. Следовательно на отрезке [-1, 1] существует ровно один корень.

Рассмотрим, еще функцию $f(x)=e^x-7\cdot x$. Она непрерывна на всей вещественной оси, но не монотонна. Действительно, ее производная $f'(x)=e^x-7$ меняет знак с минуса на плюс при переходе через точку x такую, что $e^x = 7$, т.е. при $x = \ln 7$. Значит, при $x < \ln 7$ исследуемая функция убывает, а при $x > \ln 7$ она возрастает. Заметим, что $f(-1)>0$ и $f(\ln 7)=7-7\ln 7<0$, следовательно на отрезке есть ровно один корень. Аналогично, т.к. $f(\ln 7)<0$ и $f(4)=e^4-28>0$, то на отрезке есть так же ровно один корень. Итак, простое исследование убедило нас в наличии ровно двух корней у данной функции.

Заметим, что функция fzero, предназначенная для решения уравнений в MATLAB, позволяет либо задать отрезок, на котором надо искать корень (на его границах функция должна менять знаки), либо задать начальное приближение к корню. Если задано начальное приближение, то функция fzero пытается определить отрезок, на границах которого исследуемая функция меняет знак. Иногда эта задача может оказаться достаточно тяжелой, и необходимый отрезок найден не будет. А может получиться и так, что будет найден отрезок, соответствующий не тому корню, который нам требуется

уточнить, а другому. Поэтому для надежности перед уточнением корня, как правило, желательно отделить его.

Для отделения корней в общем случае нет определенного алгоритма. Исследование можно проводить, например графически. Самый простой способ заключается в построении графика функции и указания отрезков, на которых есть ровно один корень.

Предположим, что нам требуется отделить корни функции

$$f(x) = x^3 + \frac{1}{5}x^2 - \frac{212}{25}x + \frac{204}{25}.$$

Ясно, что при достаточно больших положительных x и при достаточно малых отрицательных x корней нет, т.к. значение x^3 будет слагаемым, определяющим значение функции. Построим график функции $f(x)$ на отрезке . Для этого вычислим на отрезке в равномерно отстоящих точках с шагом (для вычисления вектора y со значениями функции применяем операцию поэлементного возведения в степень, обозначаемую точка с крышкой):

```
x = -5:0.1:5;
y = x.^3+1/5*x.^2-212/25*x+204/25;
```

Для построения графика применим функцию plot, а для нанесения на него сетки — команду grid с аргументом on:

```
plot(x, y)
grid on
```

В нашем примере получается следующий график

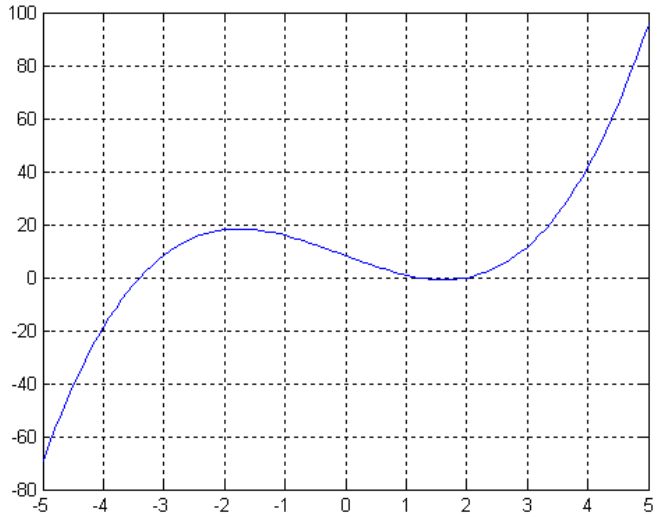
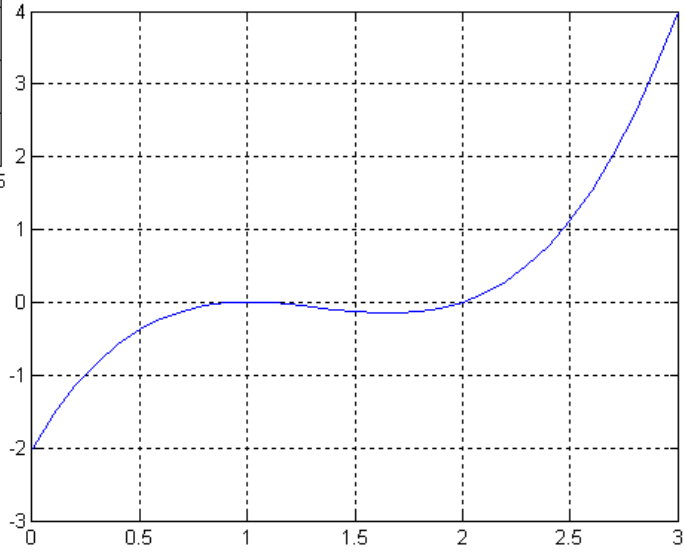


График функции

$$f(x) = x^3 + \frac{1}{5}x^2 - \frac{212}{25}x + \frac{204}{25}$$

Изучая поведение функции мы видим, что у нее есть три корня, первый около -3.5, а два других около 1.5. Однако нашей задачей было отделение корней, т.е. указание промежутков, на которых находится ровно один корень. Как видно из графика, для первого корня это промежуток [-4, -3]. Для того, чтобы отделить два других корня, удобно воспользоваться инструментом ZoomIn графического окна для увеличения масштаба просмотра вблизи точки $x = 1.5$. После увеличения масштаба просмотра становится ясно, что промежутками, содержащими ровно один корень, являются отрезки [1, 1.4] и [1.8, 2.2]. Итак, все три корня отделены.



Следующая функция представляет более сложный пример для отделения корней

$$f(x) = x^3 - \frac{4001}{1000}x^2 + \frac{5003}{1000}x - \frac{1001}{500}$$

Ее график на отрезке [0, 3], построенный по равноотстоящим точкам с шагом 0.1, приведен ниже.

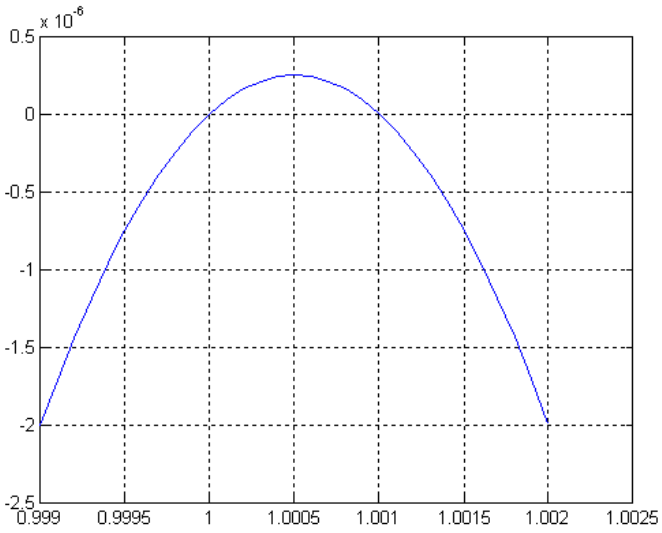
График функции

$$f(x) = x^3 - \frac{4001}{1000}x^2 + \frac{5003}{1000}x - \frac{1001}{500},$$

построенный с шагом 0.1

Из графика видно, что один корень принадлежит отрезку [1.5, 2.5]. А вблизи точки $x = 1$ ситуация не очень понятная, более того, увеличение масштаба ничего не дает. Выход состоит в построении графика вблизи точки $x = 1$ с более мелким шагом, причем в достаточно малой окрестности единицы. Например, график $f(x)$, построенный на отрезке [0.999, 1.002] с шагом 0.0001, дает хорошее представление о расположении двух оставшихся корней.

График функции $f(x) = x^3 - \frac{4001}{1000}x^2 + \frac{5003}{1000}x - \frac{1001}{500}$, вблизи $x = 1$.



Становится понятным, что у данной функции есть два корня вблизи точки $x = 1$. Один из них находится на отрезке [0.9995, 1.0005], а другой — на отрезке [1.0005, 1.0015]. Все три корня исследуемой функции отделены.

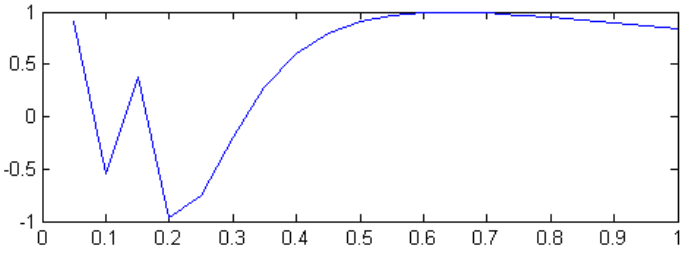
Заметим, что функция plot строит график табличной функции, заданной векторами x и y, которые мы получили по исходной непрерывной функции $f(x)$, соединяя точки отрезками. В зависимости от выбора точек для построения графика, он может оказаться слишком неточным или даже неверным. Приведем самый простой пример.

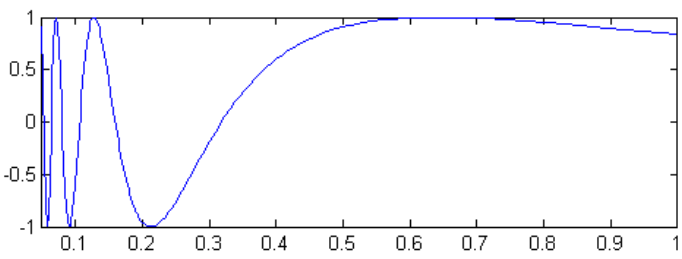
Предположим, требуется построить график функции

$$f(x) = \sin\frac{1}{x}$$

на отрезке [0.05, 1]. Сделаем это двумя способами. Первый из них состоит в вычислении функции с шагом 0.05 на данном отрезке и применении функции plot. Второй способ основан на использовании функции fplot. Для сравнения выведем результаты в одно графическое окно на разные оси, для чего применим функцию subplot

```
x = 0.05:0.05:1;
y = sin(1./x);
figure
subplot(2,1,1)
plot(x,y)
subplot(2,1,2)
fplot('sin(1/x)', [0.05 1])
```





Графики, полученные при помощи plot (верхний) и fplot (нижний).

Видим, что неудачный выбор шага для plot привел к плохому графику, который не учитывает особенности поведения функции.

Нижний график, построенный при помощи fplot, намного лучше отражает поведение исследуемой функции, хотя fplot не требует задания шага. Дело в том, что алгоритм fplot является адаптивным, т.е. fplot выбирает точки для построения графика, приспосабливаясь к особенностям поведения функции. Выбираемые функцией fplot точки для построения графика не обязательно расположены равномерно. В областях быстрого изменения функции их плотность больше, чем в областях плавного ее изменения.

Рассмотрим еще один прием, полезный при отделении корней. Перед тем, как строить график исследуемой функции $f(x)$ для отделения корней уравнения $f(x) = 0$, иногда бывает полезно представить исходное уравнение $f(x) = 0$ в виде $g(x)=h(x)$, где $g(x)$ и $h(x)$ — по возможности достаточно простые функции, поведение которых известно. Корни тогда находятся в абсциссах точек пересечения графиков этих функций.

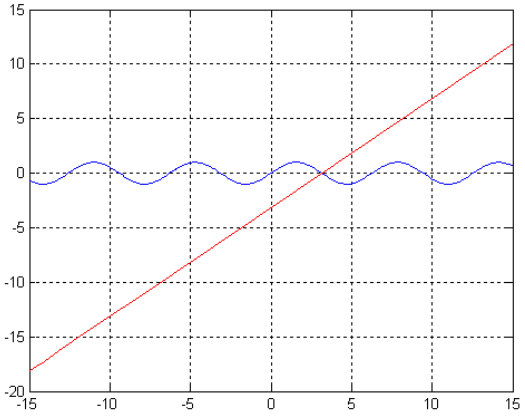
Приведем сначала очевидный пример. Рассмотрим уравнение

$$e^x - \frac{1}{x} = 0.$$

Преобразуем его к эквивалентному виду

$$e^x = \frac{1}{x}.$$

Ясно, что у данного уравнения есть только один корень — абсцисса точки пересечения графиков гиперболы $\frac{1}{x}$ и экспоненциальной функции e^x. Отделить данный корень достаточно просто, например при $x = 0.1$ левая часть уравнения $e^x - \frac{1}{x} = 0$ меньше нуля, а при $x = 1$ — больше нуля, следовательно на отрезке [0.1, 1] содержится ровно один корень.



Рассмотрим теперь уравнение

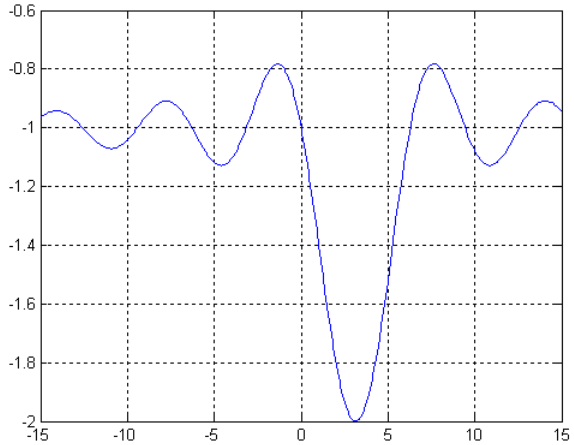
$$\frac{\sin x}{x - \pi} - 1 = 0.$$

Совершив несложное преобразование (перенос единицы в правую часть и умножение обеих частей на $x - \Pi$), приведем его к виду

$$\sin x = x - \pi.$$

На отрезке [-15, 15] строим графики функций $g(x) = \sin x$ и $h(x) = x - \pi$. Находим точку пересечения, получается, что у исходного уравнения есть корень.

```
figure
fplot('sin(x)', [-15 15])
hold on
fplot('x-pi', [-15 15], 'r')
grid on
```



Графики функций $g(x) = \sin x$ и $h(x) = x - \pi$.

Теперь строим график исходной функции $y = \frac{\sin x}{x - \pi} - 1$ на отрезке [-15, 15]. Он не пересекает ось абсцисс, следовательно корня нет.

```
figure
fplot('sin(x)/(x-pi)-1', [-15 15])
```

```
grid on
```

Графики функции $y = \frac{\sin x}{x - \pi} - 1$.

Несложно понять, что в первом случае мы совершили неэквивалентное преобразование, которое привело к появлению корня $x = \Pi$ у преобразованного уравнения. Поэтому при совершении перехода от уравнения вида $f(x) = 0$ к уравнению вида $g(x) = h(x)$ требуется выполнять только эквивалентные преобразования для того, чтобы не изменить число корней.

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

Задача численного интегрирования состоит в замене исходной подинтегральной функции *f(x)*, для которой трудно или невозможно записать первообразную в аналитике, некоторой аппроксимирующей функцией *φ(x)*. Такой функцией обычно является полином $\phi(x) = \sum_{i=1}^{n} c_i \varphi_i(x)$ (кусочный полином) .

То есть $I = \int_a^b f(x)dx = \int_a^b \varphi(x)dx + R$:

где – $R = \int_a^b r(x)dx$ *априорная погрешность метода* на интервале интегрирования,

а *r(x)* – априорная погрешность метода на отдельном шаге интегрирования.

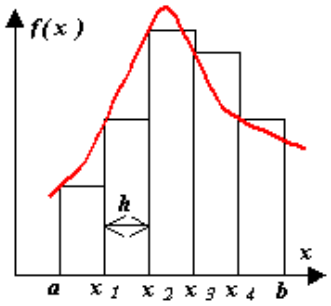
Обзор методов интегрирования.

Методы вычисления однократных интегралов называются *квадратурными* (для кратных интегралов – *кубатурными*).

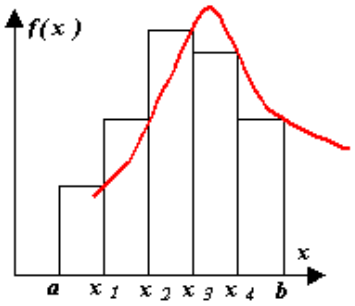
1. **Методы Ньютона-Котеса.** Здесь *φ(x)* – полином различных степеней. Сюда относятся метод прямоугольников, трапеций, Симпсона.
2. **Методы статистических испытаний (методы Монте-Карло).** Здесь узлы сетки для квадратурного или кубатурного интегрирования выбираются с помощью датчика случайных чисел, ответ носит вероятностный характер. В основном применяются для вычисления кратных интегралов.
3. **Сплайновые методы.** Здесь *φ(x)* – кусочный полином с условиями связи между отдельными полиномами посредством системы коэффициентов.
4. **Методы наивысшей алгебраической точности.** Обеспечивают оптимальную расстановку узлов сетки интегрирования и выбор весовых коэффициентов *ρ(x)* в задаче $\int_a^b \varphi(x)\rho(x)dx$. Сюда относится метод Гаусса-Кристоффеля (вычисление несобственных интегралов) и метод Маркова.

Метод прямоугольников.

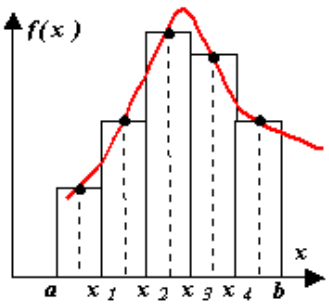
Различают метод левых, правых и средних прямоугольников. Суть метода ясна из рисунка. На каждом шаге интегрирования функция аппроксимируется полиномом нулевой степени – отрезком, параллельным оси абсцисс.



Левые прямоугольники

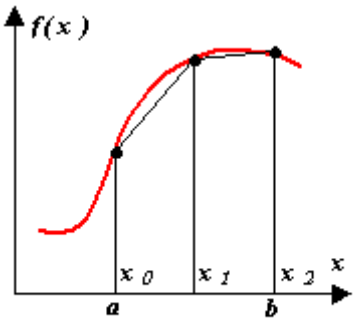


Правые прямоугольники



Средние прямоугольники

Выведем формулу метода прямоугольников из анализа разложения функции *f(x)* в ряд Тейлора вблизи некоторой точки *x = x_i*.



Метод трапеций

$$f(x)\big|_{x=x_i} = f(x_i) + (x - x_i) f'(x_i) + \frac{(x - x_i)^2}{2!} f''(x_i) + \dots$$

Рассмотрим диапазон интегрирования от x_i до x_i+h, где *h* – шаг интегрирования.

Вычислим $\int_{x_i}^{x_i+h} f(x)dx = x\cdot f(x_i)\Big|_{x_i}^{x_i+h} + \frac{(x-x_i)^2}{2} f'(x_i)\Big|_{x_i}^{x_i+h} + \frac{(x-x_i)^3}{3\cdot 2!} f''(x_i)\Big|_{x_i}^{x_i+h} + \ldots =$

$= f(x_i)h + \frac{h^2}{2} f'(x_i) + O(h^3) = \boxed{f(x_i)h + r_i}$. Получили формулу *правых (или левых) прямоугольников* и априорную оценку погрешности *r* на отдельном шаге интегрирования. Основной критерий, по которому судят о точности алгоритма – степень при величине шага в формуле априорной оценки погрешности.

В случае равного шага *h* на всем диапазоне интегрирования общая формула имеет вид

$$\int_a^b f(x)dx = h\sum_{i=0}^{n-1} f(x_i) + R$$

.

Здесь *n* – число разбиений интервала интегрирования, $R = \sum_{i=0}^{n-1} r_i = \frac{h}{2}\cdot h\sum_{i=0}^{n-1} f'(x_i) = \frac{h}{2}\int_a^b f'(x)dx$. Для справедливости существования этой оценки необходимо существование непрерывной *f'(x)*.

Метод средних прямоугольников. Здесь на каждом интервале значение функции считается в точке $\bar{x} = x_i + \frac{h}{2}$, то есть $\int_{x_i}^{x_i+h} f(x)dx = hf(\bar{x}) + r_i$. Разложение функции в ряд Тейлора показывает, что в случае средних прямоугольников точность метода существенно выше:

$$r = \frac{h^3}{24} f''(\bar{x}),\ R = \frac{h^2}{24}\int_a^b f''(x)dx$$

.

Метод трапеций.

Аппроксимация в этом методе осуществляется полиномом первой степени. Суть метода ясна из рисунка.

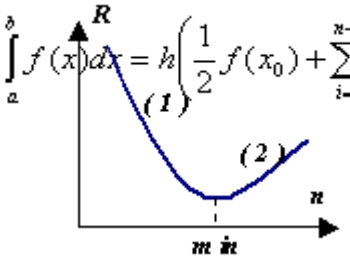
На единичном интервале

$$\int_{x_i}^{x_i+h} f(x)dx = \frac{h}{2}\left(f(x_i) + f(x_i+h)\right) + r_i$$

.

В случае равномерной сетки (*h* = const)

$$\int_a^b f(x)dx = h\left(\frac{1}{2}f(x_0) + \sum_{i=1}^{n-1} f(x_i) + \frac{1}{2}f(x_n)\right) + R$$



При этом $r_i = -\frac{h^3}{12} f''(x_i)$, а $R = -\frac{h^3}{12}\int_a^b f''(x)dx$.

Погрешность метода трапеций в два раза выше, чем у метода средних прямоугольников! Однако на практике найти среднее значение на элементарном интервале можно только у функций, заданных аналитически (а не таблично), поэтому использовать метод средних прямоугольников удается далеко не всегда. В силу разных знаков погрешности в формулах трапеций и средних прямоугольников истинное значение интеграла обычно лежит между двумя этими оценками.

Особенности поведения погрешности.

Казалось бы, зачем анализировать разные методы интегрирования, если мы можем достичь высокой точности, просто уменьшая величину шага интегрирования. Однако рассмотрим график поведения апостериорной погрешности *R*результатов численного

расчета в зависимости от числа *n* разбиений интервала (то есть при $n \to \infty$ шаг $h \to 0$. На участке (1) погрешность уменьшается в связи с уменьшением шага h. Но на участке (2) начинает доминировать вычислительная погрешность, накапливающаяся в результате многочисленных арифметических действий. Таким образом, для каждого метода существует своя R_{min}, которая зависит от многих факторов, но прежде всего от априорного значения погрешности метода *R*.

Уточняющая формула Ромберга.

Метод Ромберга заключается в последовательном уточнении значения интеграла при кратном увеличении числа разбиений. В качестве базовой может быть взята формула трапеций с равномерным шагом *h*.Обозначим интеграл с числом разбиений *n* = 1 как

$$R(1;1) = \frac{h}{2}\left(f(a) - f(b)\right).$$

Уменьшив шаг в два раза, получим

$$R(2;1) = \frac{h}{2}\left(f(a) + f(b)\right) + hf(a+h).$$

Если последовательно уменьшать шаг в 2^n раз, получим рекуррентное соотношение для расчета

$$R(n+1;1) = \frac{1}{2}R(n;1) + h\sum_{i=1}^{2^n-1} f\left(a + (2i-1)h\right).$$

Пусть мы вычислили четыре раза интеграл с *n* от 1 до 4. Представим следующий треугольник:

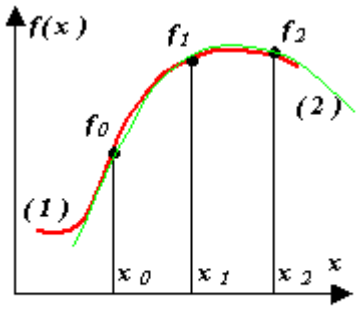
R(1;1)

R(2;1) R(2;2)

R(3;1) R(3;2) R(3;3)

R(4;1) R(4;2) R(4;3) **R(4;4)**

В первом столбце стоят значения интеграла, полученные при последовательном удвоении числа интервалов. Следующие столбцы – результаты уточнения значения интеграла по следующей рекуррентной формуле:



Метод Симпсона

$$R(n+1;m+1) = R(n+1;m) + \frac{R(n+1;m) - R(n;m)}{4^m - 1}.$$

Правое нижнее значение в треугольнике – искомое уточненное значение интеграла.

Метод Симпсона**.**

Подынтегральная функция *f(x)* заменяется интерполяционным полиномом второй степени*P(x)* – параболой, проходящей через три узла, например, как показано на рисунке ((1) – функция, (2) – полином).

Рассмотрим два шага интегрирования (*h* = const = $x_{i+1} - x_i$), то есть три узла x_0, x_1, x_2, через которые проведем параболу, воспользовавшись уравнением Ньютона:

$$P(x) = f_0 + \frac{x - x_0}{h}\left(f_1 - f_0\right) + \frac{(x - x_0)(x - x_1)}{2h^2}\left(f_0 - 2f_1 + f_2\right).$$

Пусть $z = x - x_0$,

тогда

$$P(z) = f_0 + \frac{z}{h}\left(f_1 - f_0\right) + \frac{z(z-h)}{2h^2}\left(f_0 - 2f_1 + f_2\right) =$$

$$= f_0 + \frac{z}{2h}\left(-3f_0 + 4f_1 - f_2\right) + \frac{z^2}{2h^2}\left(f_0 - 2f_1 + f_2\right)$$

Теперь, воспользовавшись полученным соотношением, сосчитаем интеграл по данному интервалу:

$$\int_{x_0}^{x_2} P(x)dx = \int_0^{2h} P(z)dz = 2hf_0 + \frac{(2h)^2}{4h}(-3f_0 + 4f_1 - f_2) + \frac{(2h)^3}{6h^2}(f_0 - 2f_1 + f_2) =$$

$$= 2hf_0 + h(-3f_0 + 4f_1 - f_2) + \frac{4h}{3}(f_0 - 2f_1 + f_2) = \frac{h}{3}(6f_0 - 9f_0 + 12f_1 - 3f_2 + 4f_0 - 8f_1 + 4f_2)$$

.В итоге $\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}(f_0 + 4f_1 + f_2) + r$

.Для равномерной сетки и четного числа шагов n формула Симпсона принимает вид:

$$\int_a^b f(x)dx = \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \ldots + 2f_{n-2} + 4f_{n-1} + f_n) + R$$

Здесь $r = -\frac{h^5}{90} f^{IV}(x_i)$, а $R = -\frac{h^4}{180}\int_a^b f^{IV}(x)dx$ в предположении непрерывности четвертой производной подынтегральной функции.

ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ В MATLAB

Символьные вычисления неопределенных интегралов в MATLAB осуществляется при помощи функции:int(fun, var),гдefun – символьное выражение, представляющее собой подынтегральную функцию, a var – переменная интегрирования.Пример вычисления неопределенного интеграла:

```
syms x %Определение символьной переменной
f=sym('exp(x) -x'); %Определение символьной функции
int(f,x) %Вычисление неопределенного интеграла
```

Результатом будет:

```
ans =
exp(x)-1/2*x^2
```

Для того чтобы вычислить определенный интеграл, можно использовать функцию:

int(fun, var, a, b),

где fun –подынтегральная функция, a var – переменная интегрирования, a, b – пределы интегрирования.

Пример вычисления определенного интеграла:

```
I1=int('exp(x)-x','x',-1,0); %Символьное решение
vpa(I1,5)%Численное решение
```

Результатом будет:

```
ans =
1.1321
```

Вычислительный алгоритм метода Симпсона с автоматическим выбором шага реализован функцией quad(name, a, b[,tol, trace]), где

- name – имя M-функции, задающей подынтегральное выражение;
- a, b – пределы интегрирования;
- tol – точность вычисления;
- trace – параметр, позволяющий получить информацию о ходе вычислений в виде таблицы, в столбцах представлены значение количества вычислений, начальная точка текущего промежутка интегрирования, его длина и значение интеграла.

Пример:

В M-файле с именем Simpson.m пишем:

```
function y=G(x)
y=exp(x)-x;
end
```

Потом в командном окне вызываем функцию quad:

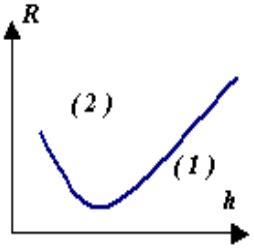
```
format long %Формат вывода значений
quad('Simpson',-1,0,1.0e-05)
```

Результатом будет:

```
ans =
1.13212056020538
```

ЧИСЛЕННОЕ ДИФФЕРЕНЦИРОВАНИЕ

Методы численного дифференцирования применяются, если исходную функцию f(x) трудно или невозможно продифференцировать аналитически. Например, эта функция может быть задана таблично. Задача численного дифференцирования – выбрать легко вычисляемую функцию (обычно полином) $\varphi(x,\vec{a})$, для которой приближенно полагают $y'(x) = \varphi'(x,\vec{a})$.



Численное дифференцирование – некорректная задача, так как отсутствует устойчивость решения. При численном дифференцировании приходится вычитать друг из друга близкие значения функции. Это приводит к уничтожению первых значащих цифр, т.е. к потере части достоверных знаков числа. А так как значения функции обычно известны с определенной погрешностью, то все значащие цифры могут быть потеряны. На графике кривая (1) соответствует уменьшению погрешности дифференцирования при уменьшении шага; кривая (2) представляет собой неограниченно возрастающий (осциллирующий) вклад неустранимой погрешности исходных данных – значений функции y(x). Критерий выхода за оптимальный шаг при его уменьшении – «разболтка» решения: зависимость результатов вычислений становится нерегулярно зависящей от величины шага. Пусть $\varphi(x,\vec{a})$ введена как интерполяционный многочлен Ньютона. В этом случае для произвольной неравномерной сетки:

$y'(x_i) \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$, для i = 0, 1…n-1, интерполяция полиномом первой степени.

$y''(x_i) \approx \frac{2}{x_{i+2} - x_i}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_{i+2} - y_i}{x_{i+2} - x_i}\right)$, интерполяция полиномом второй степени.

$$y^{(k)}(x) \approx k!\sum_{p=0}^{k} y_p \prod_{\substack{i=0 \\ i\neq p}}^{k} \left(x_p - x_i\right)^{-1}$$

В общем случае . Минимальное число узлов, необходимое для вычисления k-й производной, равно k+1.

Оценка погрешности при численном дифференцировании может быть осуществлена по

$$R_n^{(k)} \le \frac{\max\left|y^{(n+1)}\right|}{(n+1-k)!}\max_i\left|x - x_i\right|^{n+1-k}$$

формуле , где n – число узлов функции, k – порядок производной. На практике чаще всего используются упрощенные формулы для равномерной сетки, при этом точность нередко повышается. Часто используются следующие формулы для трех узлов:

$y'(x_{i+1}) \approx \frac{y_{i+2} - y_i}{2h}$,. $y''(x_{i+1}) \approx \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2}$, где $h = x_1 - x_0 = \text{const}$

Исходя из общего вида интерполяционного полинома можно вывести формулы для более высокого порядка точности или для более высоких производных.

ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ И СИСТЕМЫ УРАВНЕНИЙ В MATLAB

Для решения дифференциальных уравнений и систем в MATLAB предусмотрены следующие функции ode45(f, interval, X0 [, options]), ode23(f, interval, X0 [, options]), ode113(f, interval, X0 [, options]), odel5s(f, interval, X0 [, options]), ode23s(f, interval, X0 [, options]), ode23t (f, interval, X0 [,options]) и ode23tb(f, interval, X0 [, options]).

Входными параметрами этих функций являются:

- f - вектор-функция для вычисления правой части уравнения системы уравнений
- interval - массив из двух чисел, определяющий интервал интегрирования дифференциального уравнения или системы;
- X0 - вектор начальных условий системы дифференциальных систем
- options - параметры управления ходом решения дифференциального уравнения или системы.

Все функции возвращают:

- массив T - координаты узлов сетки, в которых ищется решение;
- матрицу X, i-й столбец которой является значением вектор-функции решения в узле Ti

В функции ode45 реализован метод Рунге-Кутта 4-5 порядка точности, в функции ode23 также реализован метод Рунге-Кутта, но 2-3 порядка, а функция ode113 реализует метод Адамса.

Пример:

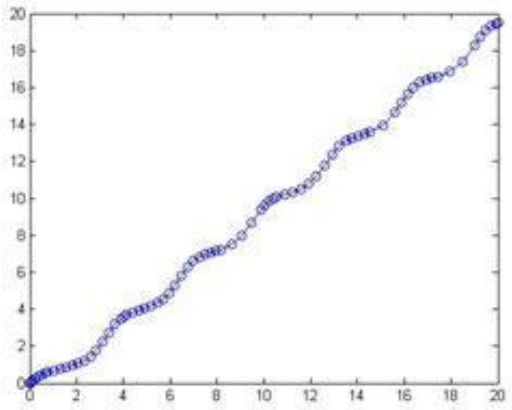
В М-файле с именем pr7.m пишем:

```
function f=pr7(x,y)
f=cos(x+y)+(3/2)*(x-y);
end
```

Потом в командном окне вызываем функцию ode113:

```
ode113(@pr7,[0  20],0) %Метод Адамса: @pr7 – ссылка на М-функцию, [0 20]-интервалы интегрирования,0 - условие: y(0)=0
```

Результатом будет график:



Пример: необходимо реализовать метод Рунге-Кутта 4 порядка и решить задачу Коши для предложенной системы дифференциальных уравнений:

$$\begin{cases} \frac{d}{dx}y_1 = y_2 \\ \frac{d}{dx}y_2 = \left(\frac{y_1}{x} - y_2\right)\frac{1}{x} - y_1 \end{cases}$$

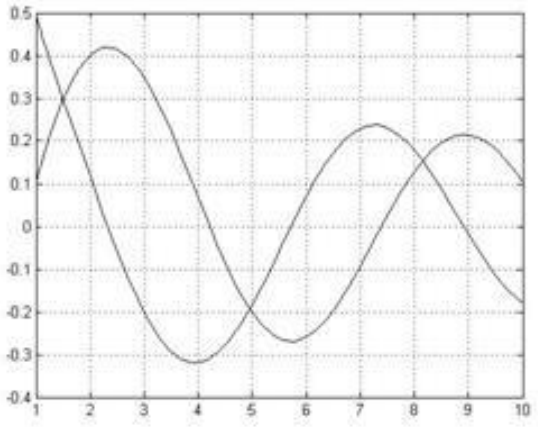
$y_1(0) = 0.1$; $y_2(0) = 0.5$
В М-файле с именем pr8.m пишем:

```
function dy=pr8(x,y)
dy=zeros(2,1);
dy(1)=y(2);
dy(2)=((y(1)/x)-y(2))*(1/x)-y(1);
end
```

Потом в командном окне вызываем функцию ode45:

```
[x,y]=ode45(@pr8,[1 10], [0.1 0.5]);
plot(x,y,'-k')
grid;
```

Результатом будет график:



КЛАССИФИКАЦИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ТРАНСПОРТНЫХ СИСТЕМ

Транспортная инфраструктура — одна из важнейших инфраструктур, обеспечивающих жизнь городов и регионов. В последние десятилетия во многих крупных городах исчерпаны или близки к исчерпанию возможности экстенсивного развития транспортных сетей. Поэтому особую важность приобретает оптимальное планирование сетей, улучшение организации движения, оптимизация системы маршрутов общественного транспорта. Решение таких задач невозможно без математического моделирования транспортных сетей. Главная задача математических моделей — определение и прогноз всех параметров функционирования транспортной сети, таких как интенсивность движения на всех элементах сети, объемы перевозок в сети общественного транспорта, средние скорости движения, задержки и потери времени и т.д.

Математические модели, применяемые для анализа транспортных сетей, весьма разнообразны по решаемым задачам, математическому аппарату, используемым данным и степени детализации описания движения. Поэтому не представляется возможным дать исчерпывающую классификацию этих моделей. Основываясь на функциональной роли моделей, т.е. на тех задачах, для решения которых они применяются, можно условно выделить три основные класса: 1) прогнозные модели, 2) имитационные модели, 3) оптимизационные модели.

Прогнозные модели предназначены для решения следующей задачи. Пусть известны геометрия и характеристики транспортной сети, а также размещение потоко-образующих объектов в городе. Необходимо определить, какими будут транспортные потоки в этой сети. Более подробно, прогноз загрузки транспортной сети включает в себя расчет усредненных характеристик движения, таких как объемы межрайонных передвижений, интенсивность потока, распределение автомобилей и пассажиров по путям движения и др. При помощи этих моделей можно прогнозировать последствия изменений в транспортной сети или в размещении объектов.

В отличие от этого имитационное моделирование ставит своей целью воспроизведение всех деталей движения, включая развитие процесса во времени. При этом усредненные значения потоков и распределение по путям считаются известными и служат исходными данными для этих моделей. Кратко это отличие можно сформулировать так: прогнозные модели отвечают на вопрос: «сколько и куда» будут ехать в данной сети, а имитационные модели отвечают на вопрос: как в деталях будет происходить движение, если известно в среднем, «сколько и куда». Таким образом, прогноз потоков и имитационное моделирование являются дополняющими друг друга направлениями. Из сказанного следует, что к классу имитационных по их функциональной роли можно отнести широкий спектр моделей, известных под названием **модели динамики транспортного потока.** В моделях этого класса может применяться разная техника — от имитации движения каждого отдельного автомобиля до описания динамики функции плотности автомобилей на дороге.

Для динамических моделей характерна значительно большая детализация описания движения и, соответственно, потребность в больших вычислительных ресурсах. Применение этих моделей позволяет оценить динамику скорости движения, задержки на перекрестках, длины и динамику образования «очередей» или «заторов» и другие характеристики движения. Основные области практического применения динамических имитационных моделей — улучшение организации движения, оптимизация светофорных циклов и др. В настоящее время актуальной задачей является разработка систем автоматизированного оперативного управления движением, работающих в режиме реального времени. Такие системы должны использовать информацию с датчиков в сочетании с динамическим имитационным моделированием. Однако помимо практических применений, развитие динамических моделей представляет большой

научный интерес в связи с изучением транспортного потока как физического явления со сложными и нетривиальными свойствами. Среди таких свойств - спонтанная потеря устойчивости, явления самоорганизации и коллективного поведения и др.
Модели прогноза потоков и имитационные модели ставят своей целью адекватное **воспроизведение** транспортных потоков. Существует, однако, большое количество моделей, предназначенных для **оптимизации** функционирования транспортных сетей. В этом классе моделей решаются задачи оптимизации маршрутов пассажирских и грузовых перевозок, выработки оптимальной конфигурации сети и др.

ОСОБЕННОСТИ ТРАНСПОРТНЫХ СИСТЕМ

Отличительными особенностями автотранспортных систем являются:

1. Транспортная система представляет собой распыленную дискретную систему, состоящую из целого множества элементов, определяющих сложность ее структуры и/или поведения: погрузочно-разгрузочные пункты; транспортные средства; предприятия, осуществляющие грузовые и/или пассажирские перевозки; транспортная сеть; грузовой склад (терминал), логистические центры и т.п.; пассажирские вокзалы (станции, остановки и т.п.); перевозочный (транспортный) процесс; поток транспортных средств и др. Такая система с течением времени меняет свое состояние, последовательно переходя из состояния *S* в состояние *S*

2. Процессы в транспортных сетях можно рассматривать как случайные события, ход и исход которых зависит от многих причин случайного характера. Переход системы из одного состояния в другое происходит «скачком», а так как каждую ездку, рейс можно перечислить (пронумеровать), то транспортный процесс является процессом с дискретным состоянием.

3. Ежесменно (ежесуточно) такая система приходит в первоначальное состояние *S*. Этот переход может осуществляться с
любого состояния *Si*. Следовательно, транспортный процесс, происходящий в автотранспортных системах, является также
циклическим случайным процессом с дискретным состоянием.

4. Принятым практикой является использование коэффициентов 0 и 1 при поставках *x* в связи с предположением об отсутствии потерь.

5. При невыполнении условия баланса поставки вводится дополнительный узел (отправитель или получатель), которого называют фиктивным.

6. В качестве теоретических основ для моделирования перечисленных объектов применяются чаще всего аппараты теории линейного (нелинейного, динамического) программирования, массового обслуживания, управления запасами, игр и т.д. При этом делаются предположения либо о полностью детерминированном функционировании систем (например, в случае решения разнообразных транспортных задач линейного программирования), либо о вероятностном характере происходящих процессов (в большинстве других случаев).

ОСНОВНЫЕ ПРИНЦИПЫ МОДЕЛИРОВАНИЯ ЗАГРУЗКИ

Транспортные потоки складываются из отдельных передвижений, совершаемых участниками движения, или пользователями транспортной сети. В общем случае, говоря о передвижениях, мы включаем в это понятие не только поездки различными видами транспорта, но и пешие передвижения. Основными факторами, определяющими количество совершаемых передвижений и их распределение по транспортной сети города, являются:

• Потокообразующие факторы, т.е. размещение объектов, порождающих передвижения, таких как места проживания, места приложения труда, культурно-бытового обслуживания и др.

• Характеристики транспортной сети, такие как количество и качество улиц и дорог, параметры организации движения, маршруты и провозные способности общественного транспорта и др.

• Поведенческие факторы, такие как мобильность населения, предпочтения при выборе способов и маршрутов передвижений и др.

Для построения математических моделей необходимо формальное описание указанных факторов. Основа такого описания — транспортный граф, узлы которого соответствуют перекресткам и станциям внеуличного транспорта, дуги — сегментам улиц и линий внеуличного транспорта. В число дуг также включаются дуги, изображающие пересадки с уличного на внеуличный транспорт. Отдельной составляющей транспортного графа является маршрутный граф общественного транспорта. Узлами маршрутного графа являются остановочные пункты, дугами — сегменты маршрутов между остановочными пунктами. С обычными узлами графа узлы-остановки соединены дугами-посадками и дугами-высадками.

Для описания распределения потокообразующих объектов необходимо разделить город на некоторое количество условных районов прибытия и отправления (ПО). Каждый район ПО включается в граф как узел, соединенный с обычными узлами графа специальными дугами-связями. Общий объем передвижений из одного района ПО в другой (независимо от конкретных путей передвижения) называется *межрайонной корреспонденцией.*

Основой для моделирования поведения пользователей является математическая формулировка критерия, на основании которого пользователь оценивает альтернативные пути и способы передвижения. Данный критерий принято называть *обобщенной ценой пути.* Увеличение обобщенной цены снижает привлекательность пути. Обобщенная цена пути складывается из обобщенных цен входящих в него дуг. Кроме того, в цену пути может добавляться цена переходов с дуги на дугу, например цена поворота при движении по улично-дорожной сети (УДС) или цена посадки при переходе с дуги-пересадки на дугу, соответствующую поездке.

Обобщенная цена определяется как взвешенная сумма слагаемых, выражающих влияние факторов различной природы на оценку пути. В общем случае она может включать в себя следующие слагаемые:

• время передвижения, которое вычисляется на основе заданной функции зависимости скорости движения от загрузки дуги. Используются различные функции скорости для дорог с разными физическими характеристиками и условиями регулировки движения;

• дополнительные задержки на различных элементах транспортной сети (время парковки, время ожидания и др.);

• денежные затраты (платные магистрали, плата за въезд в определенные зоны города и др.);

• условные штрафные добавки времени, используемые для моделирования различных особенностей транспортной сети и мер по управлению транспортом.

Как показывают обследования, время — основной фактор, определяющий цену пути. Другие факторы являются корректирующими и количественно выражаются в условных минутах, добавляемых к времени передвижения. Поэтому путь между двумя точками сети, имеющий минимальную обобщенную цену среди всех возможных путей, часто для простоты называют *кратчайшим* путем.

Важнейшей и фундаментальной особенностью формирования загрузки транспортной сети является то, что выбор способов и путей передвижения пользователями сети влияет на тот же выбор, осуществляемый другими пользователями. Математически это взаимное влияние описывается функциями зависимости цены дуги от суммарного потока по этой дуге. Данное обстоятельство создает обратную связь в процессе формирования загрузки: выбор путей, формирующих загрузку, основан на сопоставлении цен различных путей, в то время как цены сами определяются сложившейся загрузкой. Транспортные потоки, реально наблюдающиеся в сети, представляют собой некоторое *равновесное состояние* этого процесса. Для поиска этого равновесного состояния применяются итеративные алгоритмы, описанные ниже.

В задаче моделирования транспортных потоков в сети крупного города традиционно выделяют четыре основных этапа:

• оценка общих объемов прибытия и отправления из каждого района города (Trip generation);

• расщепление по способам передвижений, таким как пешие передвижения, передвижения с использованием общественного транспорта, передвижения на личном автомобиле и др. (Modal split);

• определение матриц корреспонденций, определяющих объем передвижений между каждой парой расчетных районов города (Trip distribution);

• распределение корреспонденций по транспортной сети, т.е. определение всех путей, выбираемых участниками движения, и определение количества передвижений по каждому пути (Trip assignment).

Разделение задачи моделирования на эти четыре этапа является условным, так как все этапы взаимосвязаны и не могут, вообще говоря, быть решены как отдельные задачи в силу отмеченных выше обратных связей. Так, большинство моделей расчета корреспонденций используют в качестве важного фактора обобщенные цены межрайонных передвижений. Аналогично, расщепление передвижений по видам (например, между частным и общественным транспортом) зависит от соотношения цен при использовании этих видов транспорта. Следовательно, расчет корреспонденций и их расщепление может быть выполнено корректно, если уже известна итоговая загрузка сети. Все это приводит к необходимости решать задачу последовательными приближениями, повторяя все шаги в итеративном режиме.

МОДЕЛИ РАСЧЕТА КОРРЕСПОНДЕНЦИЙ

Количественной характеристикой структуры передвижений по сети служит матрица корреспонденций, элементами которой являются объемы передвижений (автомобилей или пассажиров в час) между каждой парой условных районов ПО. Все многообразие передвижений, совершаемое в сети, может быть разбито на разные группы передвижений по следующим критериям:

- по различию в целях передвижений;
- по различию в выборе способов передвижения;
- по различию в предпочтениях при выборе путей передвижения.

Среди групп передвижений с различными целями наиболее важными и многочисленными являются

- передвижения от мест жительства к местам приложения труда и обратно (так называемые трудовые корреспонденции);
- передвижения от мест жительства к местам культурно-бытового обслуживания и обратно;
- передвижения, совершаемые между местами приложений труда (деловые поездки);
- передвижения, совершаемые между объектами культурно-бытового обслуживания.

Для каждой группы передвижений рассчитывается своя матрица межрайонных корреспонденций. Входной информацией к модели расчета корреспонденций являются общие объемы прибытия и отправления в каждом районе ПО. Оценка объема прибытий и отправлений по разным группам связана с пространственным размещением потокопорождающих объектов и подвижностью населения, т.е. средним количеством поездок, совершаемых с теми или иными целями. Эта оценка строится на основе имеющихся демографических и социально-экономических данных и результатов обследований и в основном предшествует собственно математическому моделированию. Под различными способами передвижений понимают, например, передвижение пешком, с использованием общественного транспорта или личного автомобиля. С точки зрения методики расчета смысл деления на способы передвижения следующий: избранный способ передвижения не меняется на этапе распределения корреспонденций по сети. Процедура выбора пользователем пути передвижения разбивается тем самым на два этапа: выбор способа передвижения *(модальный выбор)* и выбор конкретного пути (путей) передвижения, осуществляемый на основе некоторого критерия оценки путей *(критериальный выбор)*. Модальный выбор реализуется на стадии расчета корреспонденций, критериальный выбор реализуется на стадии распределения корреспонденций по сети.

Деление участников движения на группы по предпочтению приводит к понятию класса пользователей транспортной сети. В общем случае пользователей транспортной сети относят к разным классам, если они используют разные критерии оценки путей. В понятие «разные критерии» в данном случае включается также тот факт, что разные классы пользователей могут использовать для движения разные элементы транспортной сети. Вот некоторые примеры различных классов пользователей:

- В сети, содержащей платные участки дорог, или платный въезд в определенные районы, или платные и бесплатные парковки в различных районах люди разного достатка и социального статуса будут предпочитать разные пути.
- Пользователи сети общественного транспорта могут различаться по предпочтениям. Например, предпочитающие комфортное движение с минимальным количеством пересадок и пеших проходов или предпочитающие минимизировать время достижения цели.

Для моделирования комплексной загрузки сети с учетом всех факторов такого рода все пользователи разделяются на классы, для каждого класса рассчитывается отдельная матрица корреспонденций и производится распределение корреспонденций по сети. При этом для каждого класса используется свой критерий оптимальности путей.

К числу наиболее распространенных моделей расчета корреспонденций относятся гравитационные модели, энтропийные модели, модели конкурирующих возможностей и некоторые другие.

МОДЕЛИ РАСПРЕДЕЛЕНИЯ ПОТОКОВ

Загрузка транспортной сети определяется количеством транспортных средств или пассажиров, использующих для движения каждый элемент сети (дугу, поворот, перегон на маршруте общественного транспорта). Моделирование загрузки состоит в распределении межрайонных корреспонденций по конкретным путям, соединяющим пары районов. Входом к модели загрузки является матрица корреспонденций или в общем случае набор матриц, относящихся к передвижениям разных видов или разных пользовательских классов. Целью моделирования является определение для каждой пары районов прибытия-отправления

- набора путей, которые используются для передвижений между этими районами;
- коэффициентов расщепления (долей) корреспонденции между этими путями.

После расчета всей системы путей загрузка любого элемента сети может быть получена суммированием вкладов всех корреспонденций, использующих данный элемент. Таким образом, моделирование загрузки подразумевает более подробное описание движения, чем просто определение загрузки всех элементов. В иностранной литературе модели распределения корреспонденций по транспортной сети объединяются общим термином *traffic assignment.*

Существующие модели загрузки транспортной сети могут быть разбиты на классы по следующим основным признакам:

- модели, основанные на *нормативном* и *дескриптивном* подходе;
- *статические* и *динамические* модели.

В нормативных моделях распределение корреспонденций осуществляется на основе оптимизации некоторого глобального критерия эффективности работы транспортной сети. Таким критерием могут служить, например, суммарные затраты времени всеми участниками движения, суммарный пробег (авт*км или пасс*км) и др. При дескриптивном подходе предполагается, что структура транспортных потоков формируется в результате индивидуальных решений участников движения, основанных на оптимизации ими их личных критериев. Традиционно считается, что для моделирования загрузки реальных транспортных сетей следует применять дескриптивный подход. Нормативные модели могут применяться при планировании передвижений в тех случаях, когда планирующий орган имеет возможность директивного влияния на выбор маршрутов (например, при планировании централизованных грузовых перевозок). В последние годы, однако, интерес к нормативным моделям возрос в связи с началом разработки проектов о централизованном управлении движением частных автомобилей с использованием бортовых компьютеров и спутниковой связи.

Модель относится к классу *статических,* если загрузка моделируется в терминах усредненных характеристик движения на выбранный период моделирования (например, утренний час пик). В частности, если некоторая доля αFij корреспонденции использует выбранный маршрут движения, то предполагается, что эта доля дает вклад αF_{ij} в загрузку каждого элемента маршрута на протяжении всего периода моделирования. Такое предположение оправдано, если среднее время всех маршрутов не превышает характерное время, за которое сама корреспонденция успеет заметно поменяться. В случае, если динамика выезда меняется достаточно быстро, а маршруты достаточно длинные, необходимо учитывать, что представители той или иной корреспонденции загружают каждый участок избранного маршрута в разное время. При этом как сама корреспонденция, так и объемы прибытия-отправления в каждом районе должны задаваться как функции времени. Модели, в которые явно введен фактор времени и явно

описывается динамика расчетных величин в течение периода моделирования, называют *динамическими.*

Термин *динамический* употребляется в транспортном моделировании в разных смыслах. Здесь имеются в виду динамические модели прогноза загрузки (dynamic assignment); их не следует смешивать с динамическими имитационными моделями. Также динамическими иногда называют модели, описывающие долговременную эволюцию транспортной сети.

Наиболее простым способом распределения корреспонденций по сети является наложение каждой корреспонденции на единственный оптимальный маршрут, соединяющий два района (метод «все или ничего»). Поскольку такой способ не учитывает естественного рассеяния, а также слишком чувствителен к характеристикам отдельных дуг графа, предложены различные способы расчета нескольких альтернативных путей и рассеивания корреспонденции по этим путям. Основная трудность таких моделей состоит в методике построения разумных альтернативных путей.

Выбор пути некоторыми пользователями увеличивает загрузку элементов сети, входящих в данных путь. В результате происходит увеличение обобщенной цены этих элементов. Это, в свою очередь, влияет на оценку и выбор путей другими пользователями. Таким образом, выбор, совершаемый одними участниками движения, косвенно влияет на выбор, совершаемый другими. Особенно важно учитывать этот фактор при расчете загрузки улично-дорожной сети, поскольку время движения на каждом элементе этой сети очень сильно зависит от загрузки элемента.

Особенностью передвижений с использованием системы маршрутов общественного транспорта является то, что, начиная движение, пользователь может принимать решение не о конкретном пути, а скорее о стратегии своего поведения. Конкретное продолжение пути может в этом случае зависеть от посадок на тот или иной маршрут в процессе движения (попросту говоря от того, «какой автобус подойдет первым» в пересадочном узле).

Наиболее эффективной моделью, в полной мере учитывающей фактор взаимного влияния пользователей, является модель, основанная на поиске *равновесного распределения (user-equilibrium assignment).* Модель, определяющая загрузку транспортной сети на основе расчета стратегий поведения, называется моделью *оптимальных стратегий (optimal strategy).*

ИСТОРИЧЕСКИЕ СВЕДЕНИЯ

Основы математического моделирования закономерностей дорожного движения были заложены в 1912 году русским ученым, профессором Г.Д. Дубелиром.

Первостепенной задачей, послужившей развитию моделирования транспортных потоков, стал анализ пропускной способности магистралей и пересечений. Под пропускной способностью понимают максимально возможное число автомобилей, которое может пройти через сечение дороги за единицу времени. В специальной литературе встречаются такие модификации понятия пропускной способности, как теоретическая, номинальная, эффективная, собственная, практическая, фактическая и другие. В настоящее время пропускная способность является важнейшим критерием оценки качества функционирования путей сообщения.

Первая макроскопическая модель, в которой движение транспортного потока рассматривалось с позиций механики сплошной среды, была предложена в 1955 году Лайтхиллом (Ligthill) и Уиземом (Whitham). Они показали, что методы описания процессов переноса в сплошных средах могут быть использованы для моделирования заторов.

Выделение математических исследований транспортных потоков в самостоятельный раздел прикладной математики впервые было осуществлено Ф. Хейтом.

В 60 – 70-е годы вновь возник интерес к исследованию транспортных систем. Эта заинтересованность проявилась в том числе, в финансировании многочисленных контрактов, обращении к авторитетным ученым – специалистам в области математики, физики, процессов управления, таким как Нобелевский лауреат И. Пригожин, специалист по автоматическому управлению М.Атанс, автор фундаментальных работ по статистике Л.Брейман. В нашей стране движение автотранспорта активно изучалось в конце 70-х годов в связи с подготовкой к Олимпийским играм 1980 года в Москве. Результаты этих исследований неоднократно докладывались на научно-исследовательском семинаре И.И. Зверева на механико-математическом факультете МГУ им. М.В. Ломоносова.

Сегодня имеется обширная литература по изучению и моделированию автотранспортных потоков. Несколько академических журналов посвящены исключительно динамике автомобильного движения. Наиболее крупными являются Transportation Research, Transportation Science, Mathematical Computer Simulation, Operation Research, Automatica, Physical Review E, Physical Reports. Количество публикуемых статей исчисляется сотнями.

В конце 80-х и начале 90-х в США проблемы исследования транспортных систем были возведены в ранг проблем национальной безопасности. К решению этой задачи были привлечены лучшие „физические умы“ и компьютерная техника Национальной исследовательской лаборатории Лос-Аламос – Los Alamos National Lab (LANL).

КЛАССИФИКАЦИЯ МОДЕЛЕЙ ТРАНСПОРТНЫХ ПОТОКОВ

В моделировании дорожного движения исторически сложилось два основных подхода:

- детерминистический;
- вероятностный (стохастический).

В основе детерминированных моделей лежит функциональная зависимость между отдельными показателями, например, скоростью и дистанцией между автомобилями в потоке. В стохастических моделях транспортный поток рассматривается как вероятностный процесс.

Все модели транспортных потоков можно разбить на три класса: модели-аналоги, модели следования за лидером и вероятностные модели.

В моделях-аналогах движение транспортного средства уподобляется какому либо физическому потоку (гидро- и газодинамические модели). Этот класс моделей принято называть **макроскопическими.**

В моделях следования за лидером существенно предположение о наличии связи между перемещением ведомого и головного автомобиля. По мере развития теории в моделях этой группы учитывалось время реакции водителей, исследовалось движение на многополосных дорогах, изучалась устойчивость движения. Этот класс моделей называют **микроскопическими.**

В вероятностных моделях транспортный поток рассматривается как результат взаимодействия транспортных средств на элементах транспортной сети. В связи с жестким характером ограничений сети и массовым характером движения в транспортном потоке складываются отчетливые закономерности формирования очередей, интервалов, загрузок по полосам дороги и т. п. Эти закономерности носят существенно стохастический характер.

В последнее время в исследованиях транспортных потоков стали применять междисциплинарные математические идеи, методы и алгоритмы нелинейной динамики. Их целесообразность обоснована наличием в транспортном потоке устойчивых и неустойчивых режимов движения, потерь устойчивости при изменении условий движения, нелинейных обратных связей, необходимости в большом числе переменных для адекватного описания системы.

ГИДРОДИНАМИЧЕСКИЕ МОДЕЛИ ТРАНСПОРТНЫХ ПОТОКОВ.

Транспортный поток можно рассматривать как поток одномерной сжимаемой жидкости, допуская, что поток сохраняется и существует взаимно однозначная зависимость между скоростью и плотностью транспортного потока.

Первое допущение выражается уравнением неразрывности. В то рое – функциональной зависимостью между скоростью и плотностью для учета уменьшения скорости движения автомобилей с ростом плотности потока. Это интуитивно верное допущение теоретически может привести к отрицательной величине плотности или скорости. Очевидно, одному значению плотности может соответствовать несколько значений скорости. Поэтому для второго допущения средняя скорость потока в каждый момент времени должна соответствовать равновесному значению при данной плотности автомобилей на дороге. Равновесная ситуация – чисто теоретическое допущение и может наблюдаться только на участках дорог без пересечений. Поэтому часть исследователей отказались от непрерывных моделей, часть рассматривает их как слишком грубые.

Среди гидродинамических моделей различают модели с учетом и без учета эффекта инерции. Последние могут быть получены из уравнения неразрывности, если скорость рассматривать как функцию плотности. Модели, учитывающие инерцию, представляются уравнениями Навье-Стокса со специфическим членом, описывающим стремление водителей ехать с комфортной скоростью.

ЗАКОН СОХРАНЕНИЯ ТРАНСПОРТНОГО ПОТОКА

Рассмотрим поток транспорта на однополосной дороге, т.е. при движении без обгонов. Плотность автомобилей (количество автомобилей на единицу длины дороги) $\rho(x, t)$, $x \in \square$ в момент времени $t \geq 0$. Число автомобилей в интервале (x_1, x_2) в момент времени t равно

$$\int_{x_1}^{x_2} \rho(x, t)\, dx .$$

Пусть $\upsilon(x, t)$–скорость автомобилей в точке x в момент t. Число проходящих через x (единицу длины) автомобилей в момент t, есть $\rho(x, t)\upsilon(x, t)$. Найдем уравнение изменения плотности. Число автомобилей в интервале (x_1, x_2) за время t изменяется в соответствии с числом въезжающих и выезжающих машин:

$$\frac{d}{dt}\int_{x_1}^{x_2} \rho(x, t)\, dx = \rho(x_1, t)\upsilon(x_1, t) + \rho(x_2, t)\upsilon(x_2, t).$$

(1)

Интегрируя (1) по времени и полагая, что ρ и υ – непрерывные функции, получим

$$\int_{t_1}^{t_2}\int_{x_1}^{x_2} \rho(x, t)\, dx dt = \int_{t_1}^{t_2} \left(\rho(x_1, t)\upsilon(x_1, t) + \rho(x_2, t)\upsilon(x_2, t)\right) = -\int_{t_1}^{t_2}\int_{x_1}^{x_2} \rho(x, t)\, dx dt .$$

Поскольку $x_1, x_2 \in \square$, $t_1, t_2 > 0$ произвольны,

$$\rho_t + \left(\rho(\upsilon)\right)_x = 0, \quad x \in \square, \; t > 0 .$$

(2)

Дополним уравнение (2) начальными условиями

$$\rho(x_0, 0) = \rho_0(x), \quad x \in \square .$$

Найдем уравнение для скорости. Положим, что скорость зависит только от плотности. Если дорога пуста ($\rho = 0$), автомобили едут с максимальной скоростью $\upsilon = \upsilon_{\max}$. При наполнении дороги, скорость падает вплоть до полной остановки ($\upsilon = 0$), когда машины расположены „бампер-к-бамперу“ ($\rho = \rho_{\max}$). Эта простейшая модель выражается следующим линейным соотношением (рис. 1)

$$\upsilon(\rho) = \upsilon_{\max}\left(1 - \frac{\rho}{\rho_{\max}}\right), \quad 0 \leq \rho \leq \rho_{\max} .$$

Тогда уравнение (2) примет вид

$$\rho_t + \left(\upsilon_{\max}\left(1 - \frac{\rho}{\rho_{\max}}\right)\right)_x = 0, \quad x \in \square, \; t > 0 .$$

(3)

Очевидно, это **закон сохранения количества автомобилей**. В самом деле, интегрируя (3) по $x \in \square$, получим

$$\frac{d}{dt}\int_{\square} \rho(x, t)\, dx = -\int_{\square} \frac{\partial}{\partial x}\left[\upsilon_{\max}\rho(x, t)\left(1 - \frac{\rho(x, t)}{\rho_{\max}}\right)\right] dx = 0,$$

(4)

и, следовательно, количество автомобилей в $\square$ постоянно для любых значений $t \geq 0$.

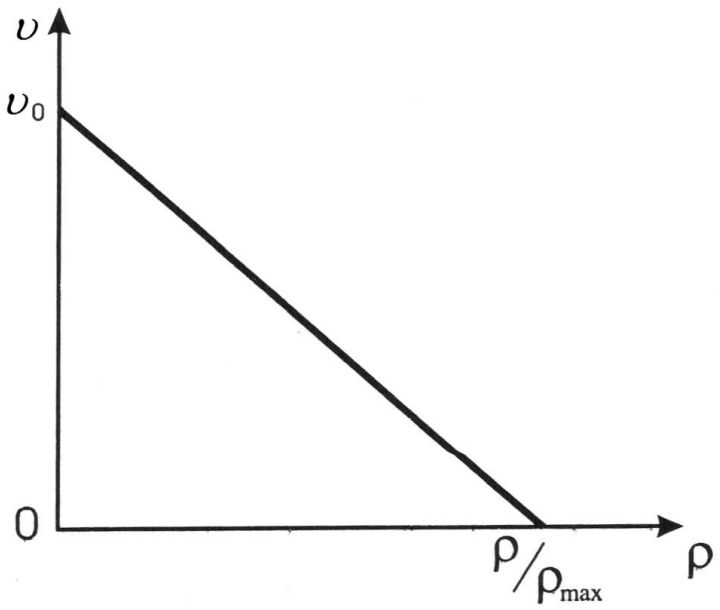


Рис. 1: Линейная аппроксимация Гриншилдса.

Предположим, что $\upsilon(x,t)=\upsilon(\rho(x,t))$ и $\upsilon'(\rho)<0$. Обозначим

$$Q(\rho)=\rho\upsilon(\rho)$$

- интенсивность потока АТС (количество АТС, проходящих в единицу времени через заданное сечение). Зависимость $Q(\rho)$ часто называют фундаментальной (или основной) диаграммой. Отметим также, что и зависимость $\upsilon(\rho)$ иногда называют фундаментальной диаграммой. Для однополосного потока принято считать:

$$Q''(\rho)<0.$$

Это условие можно понимать следующим образом: движение по двум одинаковым и независимым полосам с разными плотностями менее эффективно, чем движение по этим полосам с одинаковой плотностью, равной среднему арифметическому первоначальных плотностей. Однако, если агрегировать несколько полос в одну (иначе говоря заменить несколько полос одной агрегированной, на которой уже использовать рассматриваемую модель, то как показывают наблюдения за реальными транспортными потоками. От вогнутости функции $Q(\rho)$, вообще говоря, придется отказаться.

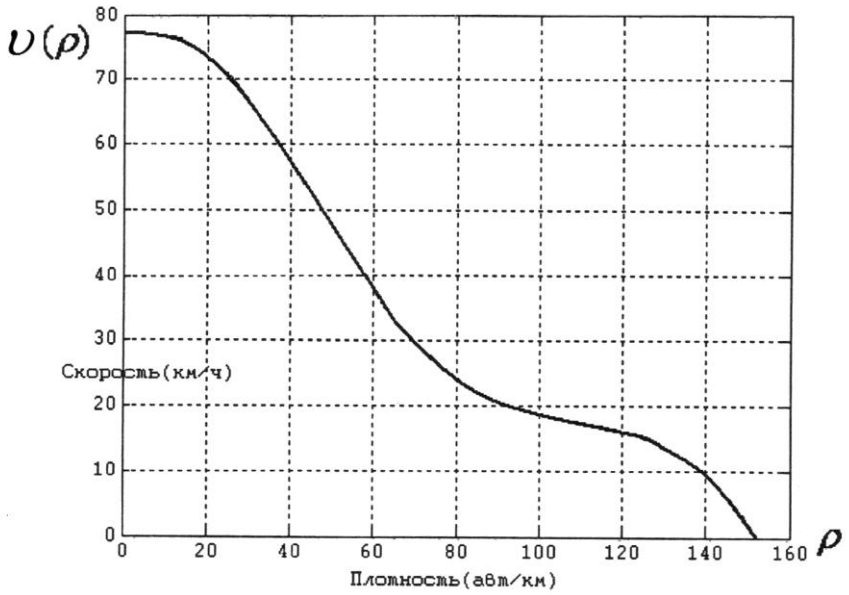


Рис. 2 Уравнение состояния транспортного потока

На рис. 2,3 отображены экспериментальные данные «Центра исследования транспортной инфраструктуры» г. Москвы (собранные в течение одного дня в 2005 году) по четырем полосам движения. Следует отметить, что в действительности измерялась зависимость $\upsilon(\rho)$.

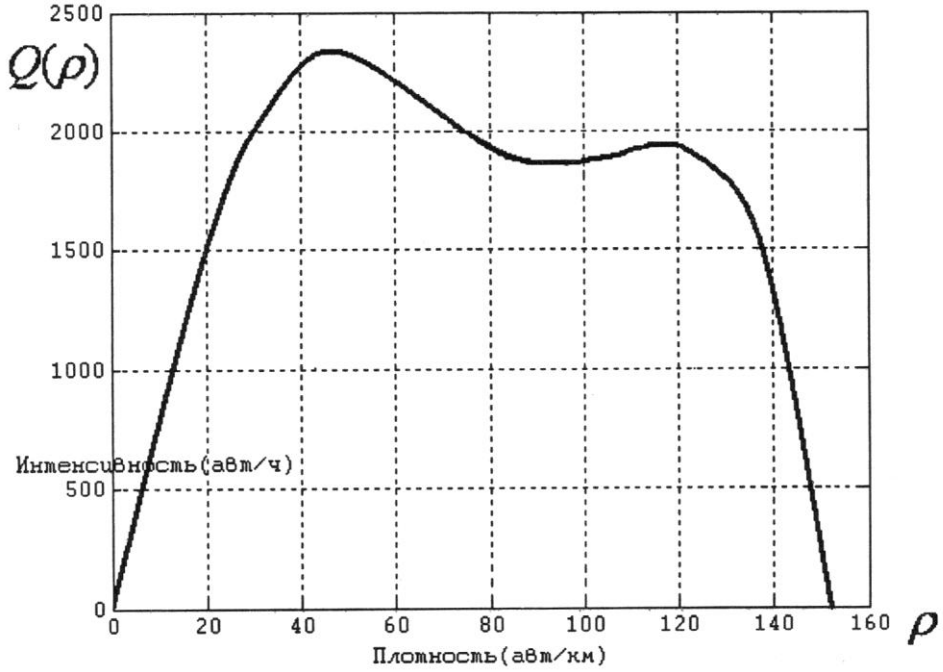


Рис. 3. Фундаментальная диаграмма

Объяснить небольшой провал интенсивности потока $Q(\rho)$ при плотностях $\rho \approx 60-115$ АТС/км можно, по-видимому, тем, что при этих плотностях существенное влияние на интенсивность потока оказывают перемещения АТС с одной полосы на другую. Перестраивания АТСиз одной полосы в другую полосу при этих плотностях снижают интенсивность потока. С одной стороны, за счет перемещения из полосы в полосу можно двигаться быстрее (так оно и происходит при плотностях $\rho \approx 30-50$ АТС/км), но, с

другой стороны в среднем такие перемещения при $\rho \approx 50-120$ АТС/км приводят к дополнительным затратам на самоперестраивание и замедление тех АТС, перед которыми встраивается новое АТС. Другое объяснение этого наблюдения связано с тем, что при $\rho \approx 50-120$ АТС/км само понятие «фундаментальная диаграмма» не совсем корректно. Иначе говоря, при этих плотностях нет четкой зависимости величины потока (скорости) от плотности. Одному значению плотности соответствует целый промежуток возможных значений потока (скорости).