

## ЛИНЕЙНЫЕ МЕТОДЫ ИНТЕРПОЛИРОВАНИЯ КРИВЫХ ТРЕТЬЕГО ПОРЯДКА ДЛЯ СТАНКОВ С ПРОГРАММНЫМ УПРАВЛЕНИЕМ \*

Б. ТАММ

Непосредственное управление фрезерным станком быстродействующей цифровой вычислительной машиной нецелесообразно в основном по двум причинам.

Во-первых, потому, что в этом случае цифровая машина должна будет для достижения требуемой точности обработки вычислить огромное количество точек, находящихся на контуре обрабатываемой детали (примерно одна точка через каждые 0,01 мм длины), что связано с затратой времени, превышающей в среднем в семь раз время, требуемое для самой обработки детали по этой программе (при скорости подачи рабочего инструмента 200 мм/мин). Кроме того, в этом случае на программно-носителе (на перфоленду, магнитную ленту или перфокарты) нужно печатать программу, содержащую чрезвычайно много информации, что усложняет ее проверку и неизбежно увеличивает возможность появления ошибок. Во-вторых, непосредственное соединение фрезерного станка с вычислительной машиной требовало бы массового производства последних, удорожало бы весь агрегат (вычислительная машина гораздо дороже фрезерного станка) и требовало бы расширения производственных площадей и увеличения обслуживающего персонала.

В связи с указанными обстоятельствами от этого метода отказались и перешли к более рациональному способу изготовления программы, где подсчеты совершаются в два этапа. Этот метод позволяет отделить вычислительную машину от станка. Кроме того, в этом случае одна вычислительная машина успевает обслужить много станков и вообще нет необходимости, чтобы вычислительные машины находились на заводе.

На первом этапе программного управления пользуются универсальной цифровой вычислительной машиной, при помощи которой первоначальная информация, определяющая контур детали, перерабатывается в данные, служащие входной информацией для второго этапа. Последняя, которая по своему объему должна быть как можно меньше, определяет траекторию движения фрезы.

На втором этапе пользуются специальной приставкой (интерполятором), электрически не связанной с вычислительной машиной. Результаты вычисления, полученные в универсальной вычислительной машине, записываются на программно-носителе, где они могут сохраняться как угодно долго, а в случае необходимости могут быть переданы в интерpolator. В интерполаторе по сравнительно небольшому числу так называемых опорных точек, координаты которых зафиксированы на программно-носителе, рассчитывается полная непрерывная программа взаимных перемещений рабочих органов станка. Эти результаты либо вводятся непосредственно в станок, либо записываются

\* Работа выполнена в Институте автоматизации и телемеханики АН СССР под руководством кандидата технических наук В. А. Котельникова.



на ленту. В последнем случае новая программа в дальнейшем прочитывается командоаппаратом, который является управляющим органом станка [2].

Совершенно очевидно, что проблема рационального программирования и, в частности, нахождения наилучших математических методов для программирования контура обрабатываемой детали, представляет собой вопрос первостепенной важности.

Требования для получения оптимальной программы часто противоречивы. Так, чем меньшая информация записана на ленту, тем меньше и вероятность появления ошибок, тем быстрее можно изготовить программу и проверить ее. С другой стороны, однако, программа с большим количеством информации позволяет рациональнее использовать станок и часто с большей точностью аппроксимировать контур обрабатываемой детали.

Если, кроме того, учесть еще и многообразие изготавливаемых изделий и то обстоятельство, что одну и ту же форму изделия можно достигнуть разными способами (последовательность ходов фрезы может быть самой различной), то станет ясным, что процесс обработки изделия может быть запрограммирован самыми различными методами. Нет возможности дать всем встречающимся в практике случаям общее, универсальное решение, пользование которым всегда обеспечивало бы получение наилучшей программы. Поэтому программист всегда должен относиться к задаче творчески и выбирать метод программирования в зависимости от конкретных условий [4].

Ниже рассматриваются некоторые математические методы, связанные с линейным интерполированием, при помощи которых целесообразно аппроксимировать детали аэродинамической и гидродинамической формы (контур которых состоит из отрезков парабол) при их программировании. Рассматривается обработка в двух измерениях.

### Постановка задачи

Исходные данные о контуре изготавливаемой детали даются конструкторами аналитически, в виде зависимости  $y = f(x)$  или же с помощью таблицы, где указаны координаты так называемых опорных точек  $x_1, x_2, x_3, \dots, x_n$  и  $y_1, y_2, y_3, \dots, y_n$ . Опорные точки выбираются обычно так, чтобы абсцисса каждой последующей точки отличалась от абсциссы каждой предыдущей на постоянную величину. В той же таблице даны еще разности ординат первого, второго и третьего порядка. Кроме того, задается форма кривой профиля изделия, например кубическая парабола

$$y = Ax^3 + Bx^2 + Cx + D$$

причем коэффициенты параболы выражаются через указанные выше разности. Так как профиль изготавливаемого изделия состоит из целого ряда парабол, то коэффициенты между различными парами опорных точек обычно различны и взаимная зависимость координат выражается различными формулами (фиг. 1).

Аппроксимирование кривой отрезками прямых линий должно произойти так, чтобы максимальное отклонение прямой от заданной кривой не превышало некоторой величины  $\delta$ , которая зависит от требуемой точности обработки и также задается конструкторами.

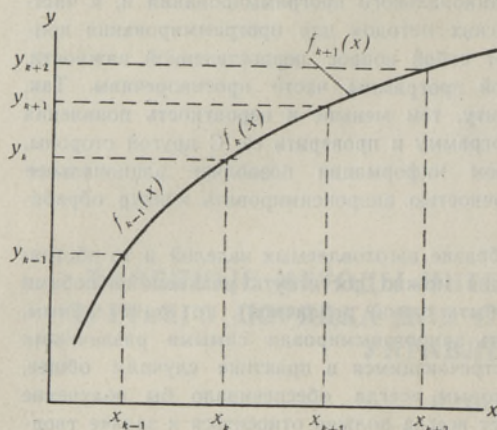
**Первый способ** интерполирования\* состоит в том, что находится целый ряд аппроксимирующих прямых с различными углами наклона и из них выбирается наилучшая, данные которой затем печатаются.

Исходя из очередной опорной точки  $(x_k, y_k)$ , координаты которой известны, задается некоторый  $\Delta x$  и вычисляется соответствующая орди-

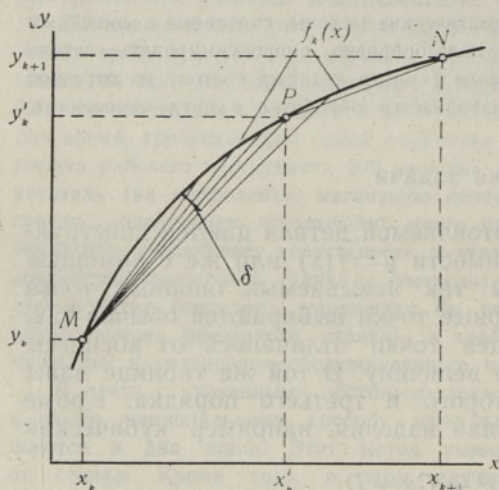
\* Разработан ведущим инженером ИАТ АН СССР В. В. Карибским.



ната  $y_k + \Delta y$  кривой  $y = f(x)$  и максимальное отклонение  $d$  прямой, соединяющей эти точки с кривой, выражающей контур детали (фиг. 2).



Фиг. 1.



Фиг. 2.

В зависимости от того, насколько точно удалось выбрать  $\Delta x$ , чтобы допустимое отклонение  $|\delta|$  возможно ближе совпадало с  $|d|$ , может представиться два различных случая. Во-первых, может оказаться, что отрезок  $\Delta x$  слишком велик и  $|d| > |\delta|$ . В таком случае нужно уменьшить  $\Delta x$  и произвести вторичный расчет; если снова  $|d| > |\delta|$ , то нужно еще раз уменьшить  $\Delta x$ , и так до тех пор, пока  $|d|$  не будет меньше  $|\delta|$ . Затем значение, при котором это условие выполняется, выводят в печать. Во втором случае  $\Delta x$  может оказаться таким, что условие  $|\delta| > |d|$  сразу выполняется. Это, однако, еще не означает, что  $\Delta x$  имеет оптимальную величину. Необходимо постепенно увеличивать  $\Delta x$  и снова произвести все подсчеты до тех пор, пока не будет удовлетворяться условие  $|\delta| < |d|$ , после чего печатать результат предыдущих вычислений (на фиг. 2  $x'_k$  и  $y'_k$ ).

С другой стороны, надо сравнить приращение ординаты  $\Delta y = y'_k - y_k$  с максимально допустимым приращением  $\Delta y_{\max}$ , так как пределы изменения ординаты ограничены из-за способности интерполятора выдавать лишь ограниченное число импульсов.

Если окажется, что

$$\Delta y > \Delta y_{\max}$$

то при помощи специальной команды, предусмотренной в программе, происходит уменьшение приращения  $\Delta x$ .

Кроме указанных, в логическую схему программы включен еще ряд вспомогательных команд, например для проверки того, не пересекла ли аппроксимирующая прямая зону, ограниченную ординатами опорных точек, и т. д. Отрезок кривой  $f_k(x)$  между опорными точками  $M$  и  $N$ , приведенный в виде примера на фиг. 2, аппроксимируется описанным методом при помощи прямых  $MP$  и  $PN$ .

**Второй способ.** Достоинством описанного метода является сравнительно несложный математический аппарат, при помощи которого проводятся вычисления. Однако необходимость многократного повторения всех подсчетов для нахождения каждой аппроксимирующей прямой существенно увеличивает общее количество вычислений, а следовательно, и число

арифметических действий и логических операций на вычислительной машине. Необходимость применения различных критериев для правильного выбора оптимальной прямой усложняет логическую схему программы.

В связи с этим представляется целесообразным найти такой способ интерполяции, при помощи которого кривая, выражающая форму изделия; аппроксимируется сразу прямой максимально возможной длины, у которой  $|d| = |\delta|$ .

Принцип нахождения подобного рода прямой состоит в следующем. Зная координаты начальной точки аппроксимирующей прямой и уравнение полинома, характеризующее кривую профиля изделия, а также максимально допустимое отклонение  $\delta$  прямой от кривой, находим требуемый наклон искомой прямой. Как видно из фиг. 3, наклон искомой прямой равен наклону касательной полинома в точке  $(x_0, y_0)$  или производной полинома в этой точке. Сначала вычисляем координаты этой точки, потом при помощи их наклон и уравнение искомой прямой.

Наконец, решая систему из уравнений прямой и полинома, находим координаты точки  $(x_1, y_1)$  пересечения прямой и кривой, которая служит начальной точкой для следующего отрезка интерполяции.

Итак, уравнение кривой контура детали выражается кубической параболой

$$y = Ax^3 + Bx^2 + Cx + D \quad (1)$$

Уравнение искомой прямой (фиг. 3) запишется в следующем виде:

$$y' - b = k(x' - a) \quad (2)$$

или

$$y - y' = k(x - x') \quad (3)$$

или

$$y_1 - b = k(x_1 - a) \quad (4)$$

где наклон прямой  $k$  выражается через

$$k = \frac{dy}{dx} = 3Ax_0^2 + 2Bx_0 + C \quad (5)$$

Перепишем уравнение (3) в нормальную форму

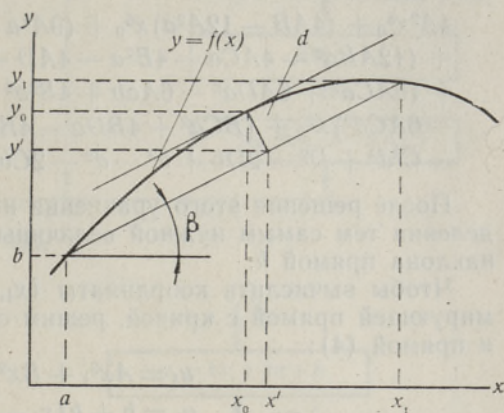
$$\frac{y}{\sqrt{1+k^2}} - \frac{kx}{\sqrt{1+k^2}} - \frac{y' - kx'}{\sqrt{1+k^2}} = 0$$

и представим максимальное отклонение прямой от кривой в виде

$$d = \frac{y_0 - kx_0 - (y' - kx')}{\sqrt{1+k^2}} \quad (6)$$

Как следует из уравнения (2),

$$y' = b + k(x' - a) \quad (7)$$



Фиг. 3.



Подставляя (7) в (6), исключаем координаты точки  $(x', y')$  и получим

$$d\sqrt{1+k^2} = y_0 - k(x_0 - a) - b \quad (8)$$

Далее, имея в виду уравнение (5), а также то, что

$$y_0 = Ax^3_0 + Bx^2_0 + Cx_0 + D$$

исключаем из (8)  $k$  и  $y_0$ . После проведения соответствующих преобразований находим уравнение для координаты  $x_0$ , которое имеет следующий вид:

$$\begin{aligned} & 4A^2x^6_0 + (4AB - 12A^2a)x^5_0 + (9A^2a - 14ABa + B^2 - 9A^2d^2)x^4_0 + \\ & + (12ABa^2 - 4ACa - 4B^2a - 4AD + 4Ab - 12ABd^2)x^3_0 + \\ & + (6ACa^2 + 6ADa^2 - 6Aab + 4B^2a^2 - 2BD + 2Bb - 2BCa - 4B^2d^2 - \\ & - 6ACd^2)x^2_0 + (BCa^2 + 4BDa - 4Bab - 4BCd^2)x_0 + C^2a^2 - \\ & - C^2d^2 + D^2 - 2Db + b^2 - d^2 - 2Cba + 2CDa = 0 \end{aligned} \quad (9)$$

После решения этого уравнения на вычислительной машине и определения тем самым нужной величины для  $x_0$ , находим из (5) величину наклона прямой  $k$ .

Чтобы вычислить координаты  $(x_1, y_1)$  точки пересечения аппроксимирующей прямой с кривой, решим совместно уравнения параболы (1) и прямой (4):

$$y_1 = Ax^3_1 + Bx^2_1 + Cx_1 + D$$

$$y_1 = b + k(x_1 - a)$$

Обозначая

$$M = C - k \quad \text{и} \quad N = D - ka - b$$

получим

$$Ax^3_1 + Bx^2_1 + Mx_1 + N = 0 \quad (10)$$

В результате решения на машине кубического уравнения (10) получим численное значение для  $x_1$  и, подставляя его в (4), находим вторую координату  $y_1$  искомой точки. С целью воспроизведения соответствующего количества импульсов в направлениях координатных осей для того, чтобы сдвинуть рабочий инструмент от точки  $(a, b)$  до точки  $(x_1, y_1)$  по прямой линии, в интерполятор нужно передать приращения координат

$$\Delta x = x_1 - a \quad \text{и} \quad \Delta y = y_1 - b$$

Эти математические действия также предвидятся в программе, и результаты печатаются на ленту.

Машинное решение вышеизложенной задачи происходит следующим образом. После того, как по разностям координат опорных точек, данным конструкторами, найдены коэффициенты параболы

$$y = Ax^3 + Bx^2 + Cx + D$$

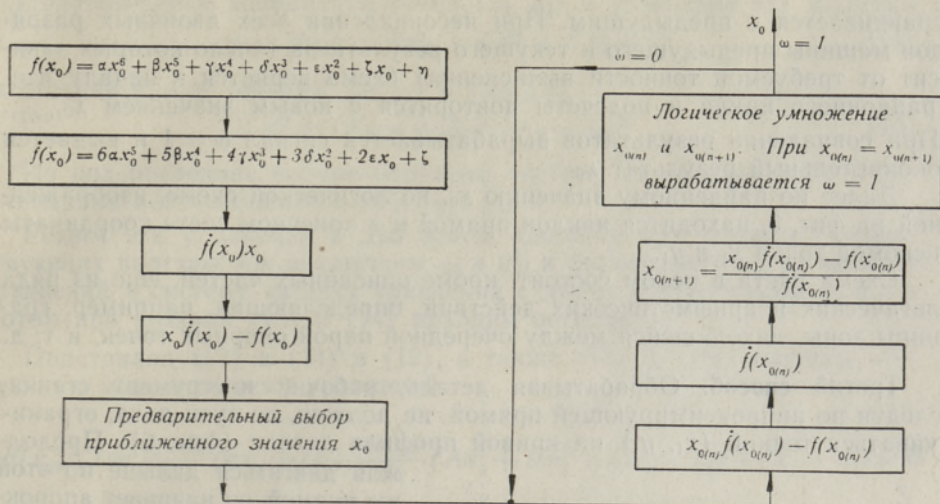
между очередной парой опорных точек машина переключается для вычисления коэффициентов уравнения шестого порядка. Определив коэффициенты, машина перейдет к решению этого уравнения методом последовательных приближений Ньютона, логическая схема которого представлена на фиг. 4.

Обозначая коэффициенты уравнения (9), начиная с высших порядков, соответственно

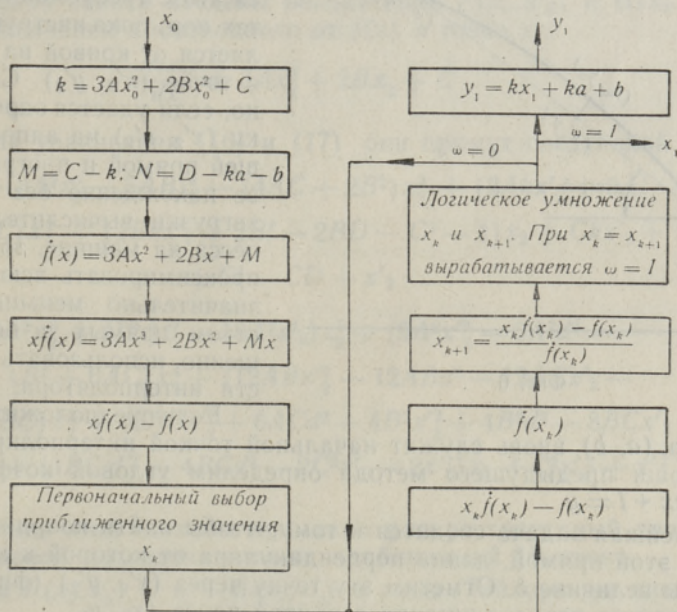
$$\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta,$$

можем его переписать следующим образом:

$$f(x_0) = \alpha x^6_0 + \beta x^5_0 + \gamma x^4_0 + \delta x^3_0 + \epsilon x^2_0 + \zeta x_0 + \eta.$$



Фиг. 4.



Фиг. 5.

При решении  $\dot{f}(x_0)$  машина возьмет производное от него, умножит это производное на  $x_0$  и вычит  $\dot{f}(x_0)$  из  $x_0\dot{f}(x_0)$ . После первоначального выбора решения для  $\dot{f}(x_0)$  начинается итерационный процесс. Делением

$$\frac{x_n\dot{f}(x_n) - f(x_n)}{\dot{f}(x_n)}$$

получается очередной результат для искомого корня  $x_0$ , который



сравнивается с предыдущим. При несовпадении всех двоичных разрядов машины предыдущего и текущего результатов (число которых зависит от требуемой точности вычисления) схема вернется к началу итерационного цикла и подсчеты повторятся с новым значением  $x_{0(n+1)}$ . При совпадении результатов вырабатывается сигнал  $\omega = 1$  и выдается окончательный результат  $x_0$ .

Далее по найденному значению  $x_0$ , по логической схеме, изображенной на фиг. 5, находится наклон прямой и в конечном счете координаты искомой точки  $x_1$  и  $y_1$ .

Схема счета в целом состоит, кроме описанных частей, еще из ряда логических и арифметических действий, определяющих, например, границы зоны, находящейся между очередной парой опорных точек, и т. д.

**Третий способ.** Обработывая деталь, рабочий инструмент станка, следуя по аппроксимирующей прямой, не должен, по сути дела, ограничиваться точкой  $(x_1, y_1)$  на кривой профиля детали (фиг. 6). Продол-

жая двигаться дальше по этой же прямой, он начинает аппроксимировать профильную кривую с другой стороны. Продолжается это с требуемой точностью до тех пор, пока инструмент не удаляется от кривой на величину  $\delta$  в точке  $(x'_2, y'_2)$ . Следовательно, если удастся определить точку  $(x'_2, y'_2)$  на аппроксимирующей прямой и запрограммировать ее нахождение без чрезмерной загрузки вычислительного процесса на машине, то можно аппроксимировать всю кривую со значительно меньшим количеством прямых и более совершенно использовать возможности интерполятора.

Если предположить, что опорная точка  $(a, b)$  вновь служит начальной точкой интерполирования, то при помощи предыдущего метода определим угловой коэффициент  $k$  прямой  $kx + l = y$ .

Дальнейшая задача сводится к тому, чтобы найти координаты такой точки на этой прямой, длина перпендикуляра от которой к кривой равнялась бы величине  $\delta$ . Отметим эту точку через  $(x'_2, y'_2)$  (фиг. 6) и точку пересечения перпендикуляра с кривой через  $(x_2, y_2)$ .

Координаты этих точек удовлетворяют следующим уравнениям: касательная кривой профиля детали  $y = f(x)$  в точке  $(x_2, y_2)$

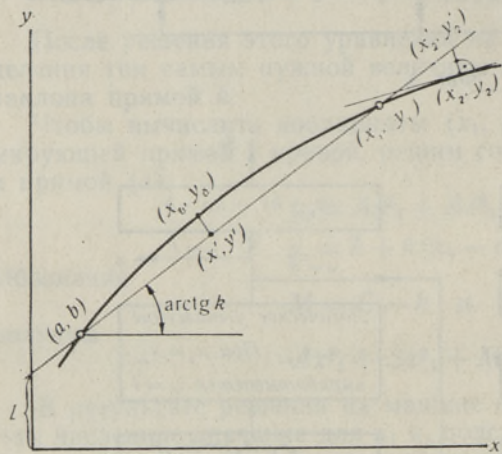
$$y - y_2 = \frac{df(x_2)}{dx} (x - x_2) \quad (11)$$

перпендикуляр к этой касательной

$$y'_2 - y_2 = -\frac{1}{\frac{df(x_2)}{dx}} (x'_2 - x_2) \quad (12)$$

аппроксимирующая прямая

$$kx'_2 + l = y'_2 \quad (13)$$



Фиг. 6.

Подставлением координаты точки  $(x_2, y_2)$  в уравнение (1) получим

$$y_2 = f(x_2) = Ax_2^3 + Bx_2^2 + Cx_2 + D \quad (14)$$

расстояние между точками  $(x_2, y_2)$  и  $(x'_2, y'_2)$

$$d^2 = (y'_2 - y_2)^2 + (x'_2 - x_2)^2 \quad (15)$$

Из них последние четыре образуют систему из четырех уравнений с четырьмя неизвестными:  $x_2, x'_2, y_2$  и  $y'_2$ .

Решим эти уравнения в два этапа. Сначала при помощи соответствующих подстановок исключаем  $y_2$  и  $y'_2$  и сводим систему из четырех уравнений в систему из двух уравнений с двумя неизвестными  $x_2$  и  $x'_2$ . Потом при помощи (13) и (14) находим остальные.

Подставляя (13) и (14) в (12), а также (12) в (15), получим следующую новую систему уравнений:

$$\begin{cases} F(x_2, x'_2) \equiv \frac{df(x_2)}{dx} (kx'_2 + l) - \frac{df(x_2)}{dx} (Ax_2^3 + Bx_2^2 + Cx_2 + D) + (x'_2 - x_2) = 0 \\ G(x_2, x'_2) \equiv (x'_2 - x_2)^2 + (x'_2 - x_2)^2 \left( \frac{df(x_2)}{dx} \right)^2 - d^2 \left( \frac{df(x_2)}{dx} \right)^2 = 0 \end{cases} \quad (16)$$

Для сокращения записи введены обозначения  $F(x_2, x'_2)$  и  $G(x_2, x'_2)$ .

После исключения производного от  $f(x)$  в точке  $x_2$

$$\frac{df(x_2)}{dx} = 3Ax_2^2 + 2Bx_2 + C$$

в предыдущих уравнениях (16) и (17) они примут следующий вид:

$$F(x_2, x'_2) = -3A^2x_2^5 - 5ABx_2^4 - (4AC + 2B^2)x_2^3 + (3Akx'_2 + 3Al - 3AD - 3BC)x_2^2 + (2Bkx'_2 + 2Bl - 2BD - C^2 - 1)x_2 + Ckx'_2 + Cl - CD + x'_2$$

$$G(x_2, x'_2) = 9A^2x_2^6 + (12AB - 18A^2x'_2)x_2^5 + (9A^2x'^2_2 - 9A^2d^2 - 24ABx'_2 + 4B^2 + 6AC)x_2^4 + (12ABx'^2_2 - 12ABd^2 - 12ACx'_2 - 8B^2x'_2 + 4BC)x_2^3 + (6ACx'^2_2 + 6ACd^2 + 4B^2x'^2_2 + 4B^2d^2 - 8BCx'_2 + C^2 + 1)x_2^2 + (4BCx'^2_2 - 4BCd^2 - 2C^2x'_2 - 2x'^2_2)x_2 + C^2x'^2_2 + x'^2_2 - C^2d^2 \quad (19)$$

Для решения этой системы применим распространенный для систем уравнений способ последовательных приближений Ньютона.

Разложим  $F(x_2, x'_2)$  и  $G(x_2, x'_2)$  в ряды Тейлора до членов первого порядка и предположим, что  $(x_2, x'_2)$  есть решение, т. е.  $F(x_2, x'_2) = G(x_2, x'_2) = 0$ . Предположим, что точка  $(x_{2_1}, x'_{2_1})$ , полученная в результате первоначального выбора, близка к решению, и пусть

$$x_{2(n+1)} - x_{2_n} = \Delta x_{2(n+1)} \quad \text{и} \quad x'_{2(n+1)} - x'_{2_n} = \Delta x'_{2(n+1)}$$

Обозначим для краткости

$$F = F(x_2, x'_2), \quad G = G(x_2, x'_2)$$

$$F_x = \frac{\partial F}{\partial x_2}, \quad F_{x'} = \frac{\partial F}{\partial x'_2}, \quad G_x = \frac{\partial G}{\partial x_2}, \quad G_{x'} = \frac{\partial G}{\partial x'_2}$$

в точке  $x_2 = x_{2_1}, \quad x'_2 = x'_{2_1}$  и т. д. Тогда приближенно



$$F + F_x \Delta x_{2(n+1)} + F_{x'} \Delta x'_{2(n+1)} = 0 \quad (20)$$

$$G + G_x \Delta x_{2(n+1)} + G_{x'} \Delta x'_{2(n+1)} = 0 \quad (21)$$

Решим уравнения (20) и (21) относительно

$$\Delta x_{2(n+1)} \quad \text{и} \quad \Delta x'_{2(n+1)} :$$

$$-\Delta x_{2(n+1)} = \frac{\begin{vmatrix} F & F_{x'} \\ G & G_{x'} \end{vmatrix}}{\begin{vmatrix} F_x & F_{x'} \\ G_x & G_{x'} \end{vmatrix}} \quad (22)$$

$$\Delta x'_{2(n+1)} = \frac{\begin{vmatrix} F_x & -F \\ G_x & -G \end{vmatrix}}{\begin{vmatrix} F_x & F_{x'} \\ G_x & G_{x'} \end{vmatrix}} \quad (23)$$

Отсюда новые приближения для решения определяются выражениями

$$x_{2(n+1)} = x_{2n} + \Delta x_{2(n+1)} \quad \text{и} \quad x'_{2(n+1)} = x'_{2n} + \Delta x'_{2(n+1)}$$

Итерационный процесс нахождения величин  $x_2$  и  $x'_2$  продолжается до тех пор, пока они не будут вычислены с определенной точностью.

Логическая схема счета начинается с вычислений коэффициентов функций  $F(x_2, x'_2)$  и  $G(x_2, x'_2)$ , а также коэффициентов частных производных  $F_x, F_{x'}, G_x$  и  $G_{x'}$  этих функций. Далее, после первоначального выбора  $x_{2_1}$  и  $x'_{2_1}$ , начинается повторяющийся итерационный процесс, где происходит численное вычисление функций (18) и (19) и их производных, вычисление определителей (22), (23) и, наконец, очередных приближений искомых величин  $x_2$  и  $x'_2$ . Упрощенная логическая схема вычислительного процесса приведена на фиг. 7.

После нахождения  $x_2$  и  $x'_2$  уже легко вычислить ординату искомой точки из уравнения (13)

$$y'_2 = kx'_2 + l$$

Если окажется нужным найти величину  $y_2$ , то она выражается через (14)

$$y_2 = Ax_2^3 + Bx_2^2 + Cx_2 + D$$

и логическая схема вычисления ее аналогична схеме, показанной на фиг. 4. Как при использовании предыдущего метода, так и здесь в интерполятор передаются приращения координат, то есть

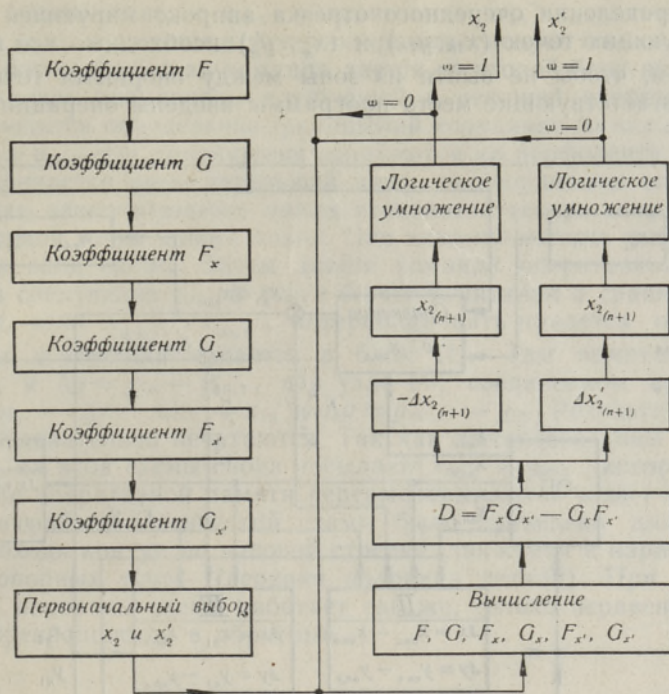
$$\Delta x = x'_2 - a$$

и

$$\Delta y = y'_2 - b$$

Изменим теперь систему обозначения координат интересующих нас точек, как показано на фиг. 8, более строго систематизируя их, и будем в дальнейшем придерживаться этих обозначений.

Что касается дальнейшего пути резца от точки  $(x'_1, y'_1)$ , то для наилучшего интерполирования нужно искать прямую, которая, проходя через точки  $(x'_1, y'_1)$  и  $(x'_2, y'_2)$  (фиг. 8), аппроксимировала бы контур



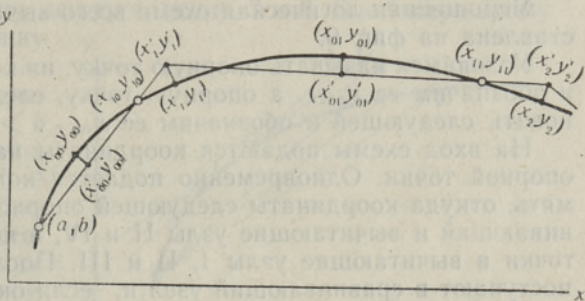
Фиг. 7.

детали с двух сторон с заданной точностью. Это удобно сделать в два этапа.

На первом этапе определяем требуемый наклон прямой и точку пересечения  $(x_{11}, y_{11})$ . На втором этапе — точку  $(x'_2, y'_2)$  на той же прямой.

Оказывается, что процесс нахождения наклона новой прямой и точки  $(x_{11}, y_{11})$  совершенно аналогичен процессу, описанному во втором способе интерполирования, несмотря на то, что здесь начальная точка отрезка прямой не находится на кривой. Можно указать даже, что процессы не только аналогичны, но их алгоритмы полностью совпадают. Тот факт, что мы простым повторением вычислительного процесса (изменяя координаты  $(a, b)$  на  $(x'_1, y'_1)$ ) можем высчитать наклон новой прямой и точку  $(x_{11}, y_{11})$ , существенно упрощает программу для машины. Определение же точки  $(x'_2, y'_2)$  производится при помощи повторения процесса, логическая схема которого изображена на фиг. 7.

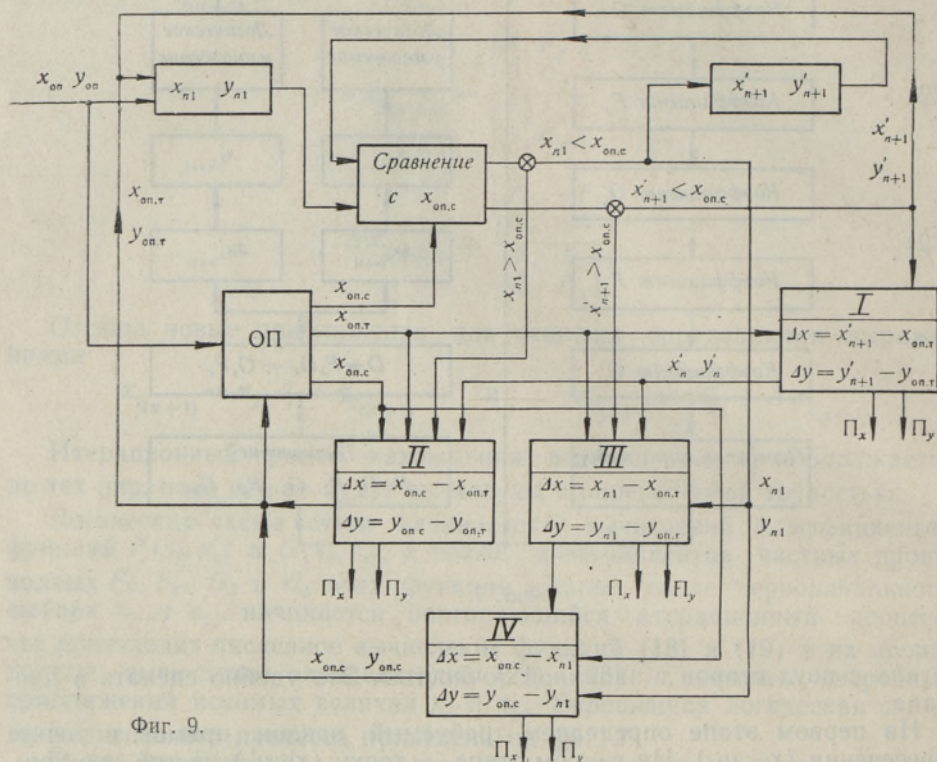
Таким образом, весь цикл интерполяции начинает повторяться независимо от того, исходим ли мы от точки  $(x'_n, y'_n)$  или от опорной точки на кривой. Сам цикл состоит из двух вышеуказанных этапов.



Фиг. 8.



При определении очередного отрезка аппроксимирующей прямой и соответствующих точек  $(x_{1n}, y_{1n})$  и  $(x'_n, y'_n)$  необходимо все время следить за тем, чтобы не выйти из зоны между опорными точками. Для этого в соответствующие места программы введены операции контроля.



Фиг. 9.

Упрощенная логическая схема всего вычислительного процесса представлена на фиг. 9.

Условимся называть опорную точку, из которой мы исходим, текущей и обозначим ее  $x_{оп,т}$ , а опорную точку, следующую за ней, будем именовать следующей и обозначим ее  $x_{оп,с}$ .

На вход схемы подаются координаты начальной точки, т. е. первой опорной точки. Одновременно подается команда и в оперативную память, откуда координаты следующей опорной точки посылаются в сравнивающий и вычитающие узлы II и IV, а координаты текущей опорной точки в вычитающие узлы I, II и III. После вычисления  $x_{1n}$  и  $y_{1n}$  они поступают в сравнивающий узел и, если окажется, что  $x_{1n} < x_{оп,с}$ , т. е. находится в зоне между текущей и следующей опорными точками, то координаты передаются в узел, где вычисляются  $x'_{n+1}$  и  $y'_{n+1}$ . Одновременно они запоминаются в вычитающем узле III. После вычисления  $x'_{n+1}$  и  $y'_{n+1}$  происходит аналогичное прежнему сравнению абсциссы этой точки с абсциссой опорной точки и, если опять  $x'_{n+1} < x_{оп,с}$ , то найденные величины передаются с одной стороны к началу схемы, откуда начинается новый вычислительный цикл. С другой стороны они попадают в узел I, где из них вычитают координаты текущей опорной точки. Результаты вычитания  $\Delta x$  и  $\Delta y$  печатаются. В узлах I, II и III  $x'_{n+1}$  и  $y'_{n+1}$  запоминаются в качестве координат  $x_{оп,т}$ .



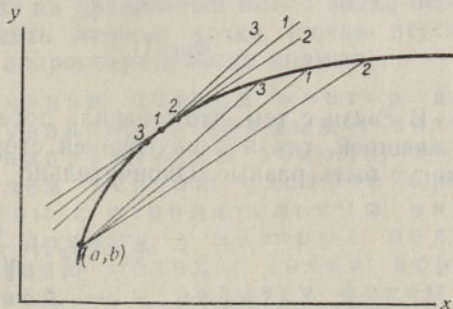
Если при сравнении  $x_{1n}$  с  $x_{оп.с}$  выяснится, что  $x_{1n} > x_{оп.с}$ , то это означает, что мы вышли из зоны и интерполировать следует только до опорной точки. Соответствующая команда дается во второй блок вычитания, где запоминались координаты следующей и текущей опорных точек. Здесь производится определение приращений координат  $\Delta x = x_{оп.с} - x_{оп.т}$  и  $\Delta y = y_{оп.с} - y_{оп.т}$ . Эти приращения печатаются на перфоленду. Так как на этом и кончается вычислительный цикл, то координаты следующей опорной точки закончившегося цикла становятся теперь координатами текущей опорной точки нового цикла. Они запоминаются в узле II и передаются на вход схемы. Затем дается команда оперативной памяти для отправки следующих  $x_{оп.с}$  и  $x_{оп.т}$  в блоки вычитания и сравнения.

В случае, если  $x'_{n+1} > x_{оп.с}$ , интерполировать придется отрезками двух прямых и команда подается в блок III. Там вычитают  $\Delta x = x_{1n} - x_{оп.т}$  и  $\Delta y = y_{1n} - y_{оп.т}$ , а в узле IV, соединенном последовательно с ним, —  $\Delta x = x_{оп.с} - x_{1n}$  и  $\Delta y = y_{оп.с} - y_{1n}$ . Результаты обоих блоков последовательно печатаются. Так как вычислительный цикл заканчивается, на вход схемы снова посылают  $x_{оп.с}$  и  $y_{оп.с}$ , которые после срабатывания оперативной памяти переименовываются в  $x_{оп.т}$  и  $y_{оп.т}$ .

Описание работы логической схемы было приведено для случая, когда мы, обходя контур по часовой стрелке, движемся к нарастающим абсциссам опорных точек (верхняя половина детали). При вычислении нижней половины схема работает так же, только неравенства при сравнении превращаются в обратные.

### Устранение погрешностей вычислений

В связи с тем, что цифровая вычислительная машина имеет конечное число двоичных разрядов, она не может дать результаты вычисления с любой точностью. Чтобы не нарушать заданный допуск, необходимо дать качественную оценку этой погрешности и принять меры для ее ликвидации. При нахождении методом итерации точки  $x_{оп}$  мы в результате некоторой погрешности можем попасть не на точку 1 (фиг. 10), а куда-нибудь в зону между точками 3 и 2. Благодаря этому мы получим ложную величину наклона и, в конечном счете, не аппроксимируем кривую с прямой, у которой  $d = \delta$ . Если  $d < \delta$  (зона 3—1), то это допустимо, но прямых, при которых  $d > \delta$  (зона 1—2), надо избегать. Следовательно, зная, в какую сторону нельзя ошибаться, всегда можно округлить последнюю цифру  $x_{оп}$  в соответствующую сторону так, чтобы эта ошибка не возникла.



Фиг. 10.

Обходя контур по часовой стрелке, значения абсцисс точек  $x_{оп}$ , находящиеся в верхней половине контура, необходимо уменьшить, а находящиеся в нижней половине — увеличить. Таким образом, в верхней половине последняя цифра абсциссы остается без изменений (отбрасывается остаток). В нижней же половине к последней цифре прибавляется единица.



Ввиду того, что новое значение  $x_1$  (фиг. 6) найдется также с некоторой погрешностью, возникает опасение, что эта погрешность увеличится при нахождении  $x_2$ . Этого можно избежать, если уточнить значение наклона прямой  $k$  по фактически найденному  $x_1$  и опять округлить его согласно вышеуказанному правилу.

Аналогичные проблемы, хотя и в меньшей мере, возникают и при вычислении других точек. Для устранения подобного рода погрешностей в алгоритме вычисления предвидятся специальные команды.

### О расчете траектории центра фрезы

Установлением координат отрезков аппроксимирующих прямых обрабатываемой поверхности и соответствующих приращений процесс подготовки программы для станка еще не заканчивается.

Для правильного передвижения фрезы по контуру обрабатываемого изделия в систему управления станка интерполятором должны быть переданы координаты движения центра фрезы, которые в зависимости от ее диаметра находятся на определенном расстоянии от установленных выше координат — на эквидистанте.

Зная координаты точек профиля детали, можно найти координаты соответствующих точек на эквидистанте. Обозначим расстояние эквидистанты от профиля детали  $y = f(x)$  через  $R$ . Тогда (фиг. 11)

$$x_g^* = x_g + R \cos \alpha$$

и

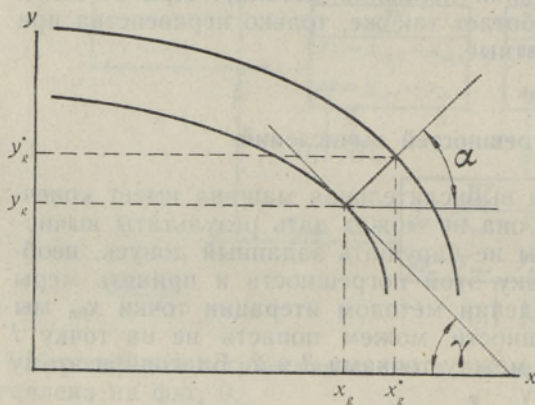
$$y_g^* = y_g + R \sin \alpha$$

$$\cos \alpha = \frac{1}{\sqrt{1 + \operatorname{tg}^2 \gamma}} = \frac{\operatorname{tg} \gamma}{\sqrt{1 + \operatorname{tg}^2 \gamma}}$$

$$\sin \alpha = \sqrt{1 - \frac{\operatorname{tg}^2 \gamma}{1 + \operatorname{tg}^2 \gamma}} = \frac{1}{\sqrt{1 + \operatorname{tg}^2 \gamma}}$$

где

$$\operatorname{tg} \gamma = \frac{df(x_g)}{dx}$$



Фиг. 11.

В связи с тем, что профиль детали может быть обведен фрезой как с внешней, так и с внутренней стороны, знаки перед радиусом фрезы могут быть разные. Окончательно

$$x_g^* = x_g + \frac{(\pm R) f'(x_g)}{\sqrt{1 + [f'(x_g)]^2}}$$

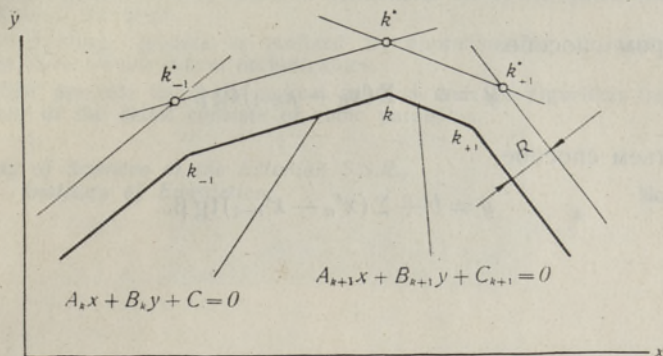
$$y_g^* = y_g + \frac{\pm R}{\sqrt{1 + [f'(x_g)]^2}}$$

Хотя вышеуказанный метод всегда ведет к цели, в нашем случае, учитывая, что весь профиль детали аппроксимировался отрезками прямых, более разумно идти по несколько иному пути.

Для обработки детали с требуемой точностью центр фрезы должен передвигаться параллельно отрезкам прямых, полученных в результате интерполирования, т. е. по эквидистанте прямых, которая находится от них на расстоянии  $R$ . Для того, чтобы обеспечить такое пере-

движение, надо определить точки пересечения прямых, образующих эквидистанту, с соседними прямыми и передать в интерполятор приращения координат следующей точки по отношению к предыдущей.

Ввиду того, что все аппроксимирующие прямые, найденные изложенными методами, вполне определены, мы можем представить их в нормальной форме. Приравняем уравнения прямых к радиусу фрезы  $R$  и составим систему уравнений из двух соседних прямых (фиг. 12):



Фиг. 12.

$$\begin{cases} \frac{A_k}{\sqrt{A_k^2 + B_k^2}} x + \frac{B_k}{\sqrt{A_k^2 + B_k^2}} y + \frac{C}{\sqrt{A_k^2 + B_k^2}} = \pm R \\ \frac{A_{k+1}}{\sqrt{A_{k+1}^2 + B_{k+1}^2}} x + \frac{B_{k+1}}{\sqrt{A_{k+1}^2 + B_{k+1}^2}} y + \frac{C}{\sqrt{A_{k+1}^2 + B_{k+1}^2}} = \pm R \end{cases}$$

Решение этой системы дает координаты точки пересечения эквидистантных прямых  $k^*$ .

Здесь следует обратить внимание на правильный выбор знака перед радиусом фрезы, чтобы не находить ложные точки (точки пересечения эквидистант на удлинениях аппроксимирующих прямых).

При наружном фрезеровании, обходя контур детали по часовой стрелке, уравнения прямых, у которых ордината второй (по направлению обхода) точки пересечения больше, чем первой, следует приравнять к радиусу фрезы с отрицательным знаком, уравнения же таких прямых, у которых ордината второй (по направлению обхода) точки пересечения меньше, чем первой, — к радиусу фрезы с положительным знаком. При внутреннем фрезеровании, обходя контур детали по тому же направлению, знаки следует выбирать обратные предыдущим.

### Заключение

Последние два способа интерполирования входят в группу таких методов, у которых количество выходной информации сводится к минимуму, и поэтому длина ленты с программой будет наименьшей.



При этом, как видно, последний метод интерполяции кривой с двух сторон является расширением предыдущего (второго) способа, куда он входит как составная часть. При наличии достаточно быстродействующей цифровой машины интерполяция по схеме третьего способа не представляет трудностей, а получаемая программа является кратчайшей для линейной интерполяции.

Огранка деталей будет неравномерной, и заменяющая кривую профиля ломаная линия имеет вид:

при втором способе

$$y = b + \Sigma (x_n - x_{n-1}) \operatorname{tg} \beta_n$$

при третьем способе

$$y = b + \Sigma (x'_n - x'_{n-1}) \operatorname{tg} \beta_n.$$

#### ЛИТЕРАТУРА

1. В. Е. Милн, Численный анализ, М., ИЛ, 1951.
2. И. М. Кучер, Цифровое программное управление металлорежущими станками, Л., 1957.
3. А. В. Чернышев, Технологические расчеты, связанные с внедрением программного управления, МВТУ им. Баумана. Технология приборостроения, 90, 1958.
4. P. J. Farmer, Programming, Aircraft Production, Vol. 19, No. 9, 1957.

*Институт энергетики  
Академии наук Эстонской ССР*

*Поступила в редакцию  
6. XI 1959.*

#### KOLMANDAT JÄRKU KÕVERATE LINEAARSE INTERPOLATSIOONI MEETODID PROGRAMMUHTIMISEGA METALLILÕIKEPINKIDE JAOKS

B. Tamm

Resümee

Artiklis esitatakse kaks uut lineaarse interpoleerimise meetodit, millede abil on võimalik otstarbekalt aproksimeerida kuupparaboolide lõikudest koosnevate aero- ja hüdrodünaamiliste kujudega detailide kontuure. On antud nende meetodite matemaatiline põhjendus, välja töötatud algoritmid ülesannete lahendamiseks masinal, iseloomustatud põhilisi arvutuslikke ebatäpsusi ja näidatud teed nende kõrvaldamiseks. Artikli viimases osas käsitletakse tööseadme tsentri trajektoori koordinaatide arvust ning antud juhendid ekvidistantsete sirglõikude õigete lõikepunktide kindlaksmääramiseks.

*Eesti NSV Teaduste Akadeemia  
Energeetika Instituut*

*Saabus toimetusse  
6. XI 1959*

METHODS OF LINEAR INTERPOLATION OF CURVES OF THE THIRD ORDER FOR  
NUMERICALLY CONTROLLED MILLING MACHINES

B. Tamm

*Summary*

The present paper describes two new methods of linear interpolation for numerically controlled milling machines.

The interpolating process is realized by approximating a smooth curve with straight lines either on one side or on both sides.

The author presents the mathematical solution and the algorithm for programming if the contour of the detail consists of cubic parabolas.

Academy of Sciences of the Estonian S.S.R.,  
Institute of Energetics

Received  
Nov. 6th, 1959