

РОССИЙСКАЯ ФЕДЕРАЦИЯ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Физический факультет
Кафедра астрономии и космической геодезии

АВДЮШЕВ Виктор Анатольевич
ИНТЕГРАТОР ГАУССА–ЭВЕРХАРТА
НОВЫЙ ФОРТРАН–КОД

Томск — 2006

Введение

В 1973 г. Э. Эверхарт [1] предложил интегратор, разработанный автором специально для численного исследования орбит, и продемонстрировал его высокую эффективность в задачах кометной динамики. По-видимому, обнаружив в дальнейшем принадлежность своего интегратора к семейству интеграторов типа Батчера, Эверхарт акцентировал внимание на оригинально реализованный им алгоритм интегрирования и обобщил его для численного решения любых обыкновенных дифференциальных уравнений первого и второго порядков [2,3]. Тем самым ему удалось расширить область применения своего интегратора, который, тем не менее, остается одним из самых популярных именно в решении задач небесной механики.

Интегратор Эверхарта (RA15) основан на видоизмененных формулах неявных методов Рунге–Кутты батчеровского типа, поэтому он наследует все их замечательные свойства [4]. Более того, именно благодаря оригинальному представлению вычислительной схемы интегратор Эверхарта обрел ряд преимуществ с точки зрения численного интегрирования: 1) алгоритм интегрирования универсален для любого порядка; 2) интегратор имеет простой критерий для выбора шага интегрирования; а также 3) в нем реализован достаточно точный предиктор решения, что позволяет выполнять численное интегрирование всего с 2 итерациями на шаге.

Несмотря на это, программный код Эверхарта RA15 [3] (впрочем, как и любая его модификация типа RADAU_27), на наш взгляд, довольно существенно ограничивает возможности интегратора и поэтому нуждается в дополнительной редакции. Среди главных недостатков в программной реализации интегратора можно перечислить следующие: 1) трудночитаемый и громоздкий код; 2) много констант, связанных с порядком интегратора, что затрудняет обобщение кода на другие порядки; 3) интегратор реализован только для определенных порядков, причем для нечетных с разбиением Гаусса–Радо, хотя известно, что неявные методы Рунге–Кутты, построенные на симметричных разбиениях Гаусса–Лобатто и Гаусса–Лежандра, обладают геометрическими свойствами [5]; 4) алгоритм выбора шага для уравнений первого порядка используется такой же как и для уравнений второго порядка, поэтому шаг при интегрировании уравнений первого порядка выбирается неверно; 5) стартовый шаг интегрирования в режиме переменного шага выбирается независимо от дифференциальных уравнений, поэтому не всегда оптимально; 6) ограничения на величину выбираемого переменного шага, по-видимому, заданы в интеграторе просто из эмпирических соображений и они, кроме того, не зависят от порядка интегратора.

В данной работе мы представляем новый код интегратора Эверхарта GAUSS_15, который разрешает названные выше трудности: 1) используя возможности Фортран 90, нам удалось сократить программный код почти в 2 раза; 2) устранены все константы, связанные с порядком метода (оставлены лишь константы узловых значений на шаге); 3) код позволяет получать решение 2–15 порядка точности (хотя при необходимости код без изменений можно обобщить на любой другой порядок: для этого нужно лишь получить соответствующие узловые значения); 4) исправлен алгоритм выбора переменного шага; 5) стартовый шаг выбирается по оценке интегрирующей схемы второго порядка с учетом поведения правых частей уравнений; 6) накладываются ограничения на выбираемый шаг в соответствии с порядком интегратора.

Кроме того, мы наделили интегратор новыми возможностями: 1) интегрирование на шаге до полной сходимости итерационного процесса; 2) запоминание величины предпоследнего шага после выполнения процедуры интегрирования, что весьма полезно при многократном использовании программного кода в режиме переменного шага; и 3) быстрый выбор стартового шага, требуемый лишь для первого обращения к интегратору (при повторном обращении используется запоминаемый шаг предыдущего обращения).

Для лучшего понимания программного кода GAUSS_15 мы также кратко излагаем общую теорию интегратора Эверхарта с внесенными нами незначительными коррективами. В дальнейшем по причине того, что интегратор использует гауссовы разбиения, мы будем называть его интегратором Гаусса–Эверхарта (хотя обычно такие интеграторы называют гауссовыми).

1 Основные формулы

Предположим, на шаге h мы решаем задачу:

$$\mathbf{x}' = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{x}_0 = \mathbf{x}(t_0). \quad (1.1)$$

Здесь t — независимая переменная, \mathbf{x} — интегрируемые переменные, а \mathbf{f} — заданная вектор-функция t и \mathbf{x} . Введем переменную $\tau = (t - t_0)/h$ и представим правую часть уравнений (1.1) в виде полинома степени k :

$$\mathbf{x}' = \mathbf{x}'_\tau/h = \mathbf{f} = \mathbf{f}_0 + \mathbf{A}_1\tau + \mathbf{A}_2\tau^2 + \mathbf{A}_3\tau^3 + \dots + \mathbf{A}_k\tau^k, \quad (1.2)$$

где коэффициенты \mathbf{A}_i пока не определены. Интегрируя (1.2) по τ , получаем решение

$$\mathbf{x} = \mathbf{x}_0 + h \left(\mathbf{f}_0\tau + \frac{1}{2}\mathbf{A}_1\tau^2 + \frac{1}{3}\mathbf{A}_2\tau^3 + \frac{1}{4}\mathbf{A}_3\tau^4 + \dots + \frac{1}{k+1}\mathbf{A}_k\tau^{k+1} \right). \quad (1.3)$$

Перепишем (1.2) в виде интерполяционного многочлена Ньютона на сетке $\tau_0, \tau_1, \dots, \tau_k$ ($\tau_0 = 0$):

$$\mathbf{f} = \mathbf{f}_0 + \alpha_1\tau + \alpha_2\tau(\tau - \tau_1) + \alpha_3\tau(\tau - \tau_1)(\tau - \tau_2) + \dots + \alpha_k\tau(\tau - \tau_1)\dots(\tau - \tau_{k-1}). \quad (1.4)$$

Из соотношений

$$\begin{aligned} \mathbf{f}_1 &= \mathbf{f}_0 + \alpha_1\tau_1, \\ \mathbf{f}_2 &= \mathbf{f}_0 + \alpha_1\tau_2 + \alpha_2\tau_2(\tau_2 - \tau_1), \\ \mathbf{f}_3 &= \mathbf{f}_0 + \alpha_1\tau_3 + \alpha_2\tau_3(\tau_3 - \tau_1) + \alpha_3\tau_3(\tau_3 - \tau_1)(\tau_3 - \tau_2), \\ &\dots \end{aligned} \quad (1.5)$$

получаем конечные разности α

$$\begin{aligned} \alpha_1 &= (\mathbf{f}_1 - \mathbf{f}_0)/\tau_1, \\ \alpha_2 &= ((\mathbf{f}_2 - \mathbf{f}_0)/\tau_2 - \alpha_1)/(\tau_2 - \tau_1), \\ \alpha_3 &= (((\mathbf{f}_3 - \mathbf{f}_0)/\tau_3 - \alpha_1)/(\tau_3 - \tau_1) - \alpha_2)/(\tau_3 - \tau_2), \\ &\dots \end{aligned} \quad (1.6)$$

В свою очередь, сравнивая (1.2) и (1.4), будем иметь

$$\begin{aligned} \mathbf{A}_1 &= \alpha_1 + (-\tau_1)\alpha_2 + (\tau_1\tau_2)\alpha_3 + \dots + (-1)^{k-1}(\tau_1\dots\tau_{k-1})\alpha_k = c_{11}\alpha_1 + c_{21}\alpha_2 + \dots + c_{k1}\alpha_k, \\ \mathbf{A}_2 &= \alpha_2 + (-\tau_1 - \tau_2)\alpha_3 + \dots = c_{22}\alpha_2 + \dots + c_{k2}\alpha_k, \\ &\dots \dots \dots \\ \mathbf{A}_k &= \alpha_k = c_{kk}\alpha_k. \end{aligned} \quad (1.7)$$

Тогда обратный переход от \mathbf{A} к α можно представить как

$$\begin{aligned} \alpha_1 &= d_{11}\mathbf{A}_1 + d_{21}\mathbf{A}_2 + \dots + d_{k1}\mathbf{A}_k, \\ \alpha_2 &= d_{22}\mathbf{A}_2 + \dots + d_{k2}\mathbf{A}_k, \\ &\dots \dots \dots \\ \alpha_k &= d_{kk}\mathbf{A}_k. \end{aligned} \quad (1.8)$$

Коэффициенты c_{ij} и d_{ij} являются числами Стирлинга, которые вычисляются по формулам

$$\begin{aligned} c_{ii} &= d_{ii} = 1, \quad c_{i0} = d_{i0} = 0 \quad (i > 0), \\ c_{ij} &= c_{i-1,j-1} - \tau_{i-1}c_{i-1,j}, \quad d_{ij} = d_{i-1,j-1} - \tau_j d_{i-1,j} \quad (i > j > 0). \end{aligned} \quad (1.9)$$

В соответствии с (1.9) каждый коэффициент c_{ij} представляет собой сумму всевозможных произведений $i - j$ величин $\tau_1, \dots, \tau_{i-1}$ со знаком $(-1)^{i-j}$. Например,

$$c_{62} = \tau_1\tau_2\tau_3\tau_4 + \tau_1\tau_2\tau_3\tau_5 + \tau_1\tau_2\tau_4\tau_5 + \tau_1\tau_3\tau_4\tau_5 + \tau_2\tau_3\tau_4\tau_5.$$

Отсюда согласно теореме Виета значения $\tau_1, \dots, \tau_{i-1}$ будут являться корнями полинома вида

$$P_{i-1} = c_{i1} + c_{i2}\tau + c_{i3}\tau^2 + \dots + c_{ii}\tau^{i-1}. \quad (1.10)$$

2 Интегрирование на шаге

Величины α определяются по \mathbf{f} , которые, в свою очередь, вычисляются по решениям \mathbf{x} . Согласно (1.3) эти решения будем задавать как

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_0 + h \left(\mathbf{f}_0 \tau_1 + \frac{1}{2} \mathbf{A}_1 \tau_1^2 + \frac{1}{3} \mathbf{A}_2 \tau_1^3 + \dots + \frac{1}{k+1} \mathbf{A}_k \tau_1^{k+1} \right), \\ &\dots \\ \mathbf{x}_k &= \mathbf{x}_0 + h \left(\mathbf{f}_0 \tau_k + \frac{1}{2} \mathbf{A}_1 \tau_k^2 + \frac{1}{3} \mathbf{A}_2 \tau_k^3 + \dots + \frac{1}{k+1} \mathbf{A}_k \tau_k^{k+1} \right). \end{aligned} \quad (2.1)$$

Формулы (2.1) представляют собой неявные уравнения относительно \mathbf{x} , поэтому они решаются итерационным способом.

Для получения начального приближения $\bar{\alpha}$ на следующем шаге \bar{h} используется информация о коэффициентах \mathbf{A} на текущем шаге h . Безразмерная независимая переменная следующего шага будет $\bar{\tau} = (t - t_h)/\bar{h}$, где $t_h = t_0 + h$. Отсюда

$$\tau = r\bar{\tau} + 1, \quad (2.2)$$

где $r = \bar{h}/h$. Согласно (1.2)

$$\mathbf{f}_0 + \mathbf{A}_1 \tau + \mathbf{A}_2 \tau^2 + \mathbf{A}_3 \tau^3 + \dots + \mathbf{A}_k \tau^k = \bar{\mathbf{f}}_0 + \bar{\mathbf{A}}_1 \bar{\tau} + \bar{\mathbf{A}}_2 \bar{\tau}^2 + \bar{\mathbf{A}}_3 \bar{\tau}^3 + \dots + \bar{\mathbf{A}}_k \bar{\tau}^k. \quad (2.3)$$

Подставляя (2.2) в (2.3) и приравнявая коэффициенты при одинаковых степенях $\bar{\tau}$, получаем:

$$\begin{aligned} \bar{\mathbf{f}}_0 &= e_{00} \mathbf{f}_0 + e_{10} \mathbf{A}_1 + e_{20} \mathbf{A}_2 + e_{30} \mathbf{A}_3 + \dots + e_{k0} \mathbf{A}_k, \\ \bar{\mathbf{A}}_1 &= r(e_{11} \mathbf{A}_1 + e_{21} \mathbf{A}_2 + e_{31} \mathbf{A}_3 + \dots + e_{k1} \mathbf{A}_k), \\ \bar{\mathbf{A}}_2 &= r^2(e_{22} \mathbf{A}_2 + e_{32} \mathbf{A}_3 + \dots + e_{k2} \mathbf{A}_k), \\ &\dots \\ \bar{\mathbf{A}}_k &= r^k e_{kk} \mathbf{A}_k, \end{aligned} \quad (2.4)$$

где e_{ij} — числа арифметического треугольника, вычисляемые по рекуррентным формулам

$$e_{ii} = e_{i0} = 1, \quad e_{ij} = e_{i-1, j-1} + e_{i-1, j} \quad (i > j > 0). \quad (2.5)$$

Пользуясь соотношениями (1.8) для $\bar{\mathbf{A}}$, будем иметь начальное приближение $\bar{\alpha}$.

Оценку $\bar{\mathbf{A}}$ для \bar{h} можно существенно улучшить, если к ней добавлять поправку $\Delta \mathbf{A}$, получаемую как разность между значениями \mathbf{A} после итераций и оценкой $\bar{\mathbf{A}}$ на текущем шаге h .

Каждая итерация выполняется следующим образом. Сначала определяется решение \mathbf{x}_1 , из которого по первой формуле (1.5) улучшается значение α_1 . Далее определяется \mathbf{x}_2 , по которому улучшается α_2 , и так до \mathbf{x}_k . Как правило, для получения достаточно хороших α необходимо всего лишь 2 итерации, очень редко — 3.

Как только величины α получены, решение на шаге h ($\tau = 1$) будет

$$\mathbf{x}_h = \mathbf{x}_0 + h \left(\mathbf{f}_0 + \frac{1}{2} \mathbf{A}_1 + \frac{1}{3} \mathbf{A}_2 + \dots + \frac{1}{k+1} \mathbf{A}_k \right). \quad (2.6)$$

В начале интегрирования, на первом шаге, в качестве $\bar{\alpha}$ выбирают нулевые значения и запускается вышеописанный итерационный процесс. Если начальный шаг достаточно большой, чтобы обеспечить заданную локальную точность, то его следует уменьшить. При оптимально выбранном шаге высокая точность α достигается уже на 4 итерации.

3 Формулы интегратора как одно из представлений неявного метода Рунге–Кутты

Пользуясь (1.6), (1.7) и (2.1), нетрудно показать, что решение (2.6) представимо в виде

$$\mathbf{x}_h = \mathbf{x}_0 + h \sum_{i=0}^k b_i \mathbf{k}_i, \quad \text{где} \quad \mathbf{k}_i = \mathbf{f}(t_0 + h\tau_i, \mathbf{x}_0 + h \sum_{j=0}^k a_{ij} \mathbf{k}_j) \quad (i = 0, \dots, k),$$

а коэффициенты a_{ij} и b_i — постоянные, зависящие только от τ_i . Таким образом, интегратор Гаусса–Эверхарта фактически основан на видоизмененных формулах неявного метода Рунге–Кутты.

4 Повышение порядка интегратора

Значения узлов τ_1, \dots, τ_k — свободные параметры и их можно выбирать как угодно, лишь бы они не были равны между собой. В общем случае интегратор будет иметь порядок $k + 1$, однако существуют такие узловые значения, которые позволяют значительно повысить точность решения, а именно до порядка $2k$, либо $2k + 1$.

Приближенное решение порядка $2k + 1$ на сетке τ_1, \dots, τ_{2k} будет

$$\mathbf{x}_h = \mathbf{x}_0 + h \left(\mathbf{f}_0 + \frac{1}{2} \mathbf{A}'_1 + \frac{1}{3} \mathbf{A}'_2 + \dots + \frac{1}{2k+1} \mathbf{A}'_{2k} \right). \quad (4.1)$$

Определим τ_1, \dots, τ_k так, чтобы решение (2.6) имело порядок $2k + 1$. Это возможно в том случае, если разность решений (2.6) и (4.1) будет равна нулю:

$$\Delta \mathbf{x}_h = h \left(\frac{\mathbf{A}'_1 - \mathbf{A}_1}{2} + \frac{\mathbf{A}'_2 - \mathbf{A}_2}{3} + \dots + \frac{\mathbf{A}'_k - \mathbf{A}_k}{k+1} + \frac{\mathbf{A}'_{k+1}}{k+2} + \dots + \frac{\mathbf{A}'_{2k}}{2k+1} \right) = \mathbf{0}. \quad (4.2)$$

Согласно (1.6) коэффициенты $\alpha_1, \dots, \alpha_k$ в (2.6) и (4.1) совпадают. Тогда

$$\begin{aligned} \mathbf{A}'_1 - \mathbf{A}_1 &= c_{k+1,1} \alpha_{k+1} + \dots + c_{2k,1} \alpha_{2k}, \\ \mathbf{A}'_2 - \mathbf{A}_2 &= c_{k+1,2} \alpha_{k+1} + \dots + c_{2k,2} \alpha_{2k}, \\ &\dots \\ \mathbf{A}'_k - \mathbf{A}_k &= c_{k+1,k} \alpha_{k+1} + \dots + c_{2k,k} \alpha_{2k}, \\ \mathbf{A}'_{k+1} &= c_{k+1,k+1} \alpha_{k+1} + \dots + c_{2k,k+1} \alpha_{2k}, \\ \mathbf{A}'_{k+2} &= c_{k+2,k+2} \alpha_{k+2} + \dots + c_{2k,k+2} \alpha_{2k}, \\ &\dots \\ \mathbf{A}'_{2k} &= c_{2k,2k} \alpha_{2k}. \end{aligned} \quad (4.3)$$

Используя рекуррентные соотношения (1.9) выразим в (4.3) коэффициенты $c_{k+i,j}$ через $c_{k+1,j}$:

$$\begin{aligned} c_{k+2,1} &= -\tau_{k+1} c_{k+1,1}, & c_{k+2,j} &= c_{k+1,j-1} - \tau_{k+1} c_{k+1,j} & (j > 1), \\ c_{k+3,1} &= \tau_{k+2} \tau_{k+1} c_{k+1,1}, & c_{k+3,j} &= c_{k+1,j-2} + (-\tau_{k+1} - \tau_{k+2}) c_{k+1,j-1} + \tau_{k+2} \tau_{k+1} c_{k+1,j} & (j > 1), \\ &\dots & &\dots \end{aligned} \quad (4.4)$$

Подставляя (4.4) в (4.3), получаем

$$\begin{aligned} \mathbf{A}'_1 - \mathbf{A}_1 &= \mathbf{B}_1 c_{k+1,1}, \\ \mathbf{A}'_2 - \mathbf{A}_2 &= \mathbf{B}_1 c_{k+1,2} + \mathbf{B}_2 c_{k+1,1}, \\ &\dots \\ \mathbf{A}'_k - \mathbf{A}_k &= \mathbf{B}_1 c_{k+1,k} + \mathbf{B}_2 c_{k+1,k-1} + \dots + \mathbf{B}_k c_{k+1,1}, \\ \mathbf{A}'_{k+1} &= \mathbf{B}_1 c_{k+1,k+1} + \mathbf{B}_2 c_{k+1,k} + \dots + \mathbf{B}_k c_{k+1,2}, \\ \mathbf{A}'_{k+2} &= \mathbf{B}_2 c_{k+1,k+1} + \dots + \mathbf{B}_k c_{k+1,3}, \\ &\dots \\ \mathbf{A}'_{2k} &= \mathbf{B}_k c_{k+1,k+1}, \end{aligned} \quad (4.5)$$

где \mathbf{B} имеют форму (1.7):

$$\begin{aligned} \mathbf{B}_1 &= \alpha_{k+1} + (-\tau_{k+1}) \alpha_{k+2} + (\tau_{k+1} \tau_{k+2}) \alpha_{k+3} + \dots + (-1)^{k-1} (\tau_{k+1} \dots \tau_{2k-1}) \alpha_{2k}, \\ \mathbf{B}_2 &= \alpha_{k+2} + (-\tau_{k+1} - \tau_{k+2}) \alpha_{k+3} + \dots, \\ &\dots \\ \mathbf{B}_k &= \alpha_{2k}. \end{aligned}$$

Подставляя далее (4.5) в (4.2) и уравнивая коэффициенты при одинаковых величинах \mathbf{B} , будем иметь уравнения для $c_{k+1,j}$

$$\begin{aligned} \frac{c_{k+1,1}}{2} + \frac{c_{k+1,2}}{3} + \dots + \frac{c_{k+1,k}}{k+1} + \frac{1}{k+2} &= 0, \\ &\dots \\ \frac{c_{k+1,1}}{k+1} + \frac{c_{k+1,2}}{k+2} + \dots + \frac{c_{k+1,k}}{2k} + \frac{1}{2k+1} &= 0. \end{aligned} \quad (4.6)$$

Решения (4.6) однозначно определяют значения τ_1, \dots, τ_k , которые вычисляются из уравнения

$$c_{k+1,1} + c_{k+1,2}\tau + c_{k+1,3}\tau^2 + \dots + c_{k+1,k}\tau^{k-1} + \tau^k = 0. \quad (4.7)$$

Узловые значения τ_i , определяемые из (4.6) и (4.7), задают (*левое*) разбиение Гаусса–Радо. Их также можно получить из уравнения

$$(\tau^{k+1}(\tau - 1)^k)_\tau^{(k)} = 0. \quad (4.8)$$

В (4.8) левый корень равен нулю ($\tau_0 = 0$). Если потребуем, чтобы правый корень был равен единице ($\tau_k = 1$), то для повышения порядка численного решения получим разбиение Гаусса–Лобатто, узловые значения которого также удовлетворяют уравнению

$$(\tau^k(\tau - 1)^k)_\tau^{(k-1)} = 0. \quad (4.9)$$

Разбиение Гаусса–Лобатто симметрично относительно середины шага и дает порядок $2k$.

Следует заметить, что решение (2.6) при разбиении Гаусса–Радо вычисляется вне интервала интерполяции, поскольку $\tau_k < 1$. Поэтому ошибка экстраполированного решения при $\tau = 1$ может быть больше ошибки интерполяции.

5 Выбор шага

В интеграторе Гаусса–Эверхарта контроль шага интегрирования осуществляется по величине последнего члена в (2.6).

Пусть $\|\mathbf{e}_{tol}\|$ — заданная точность. Потребуем, чтобы на следующем шаге выполнялось равенство

$$\frac{\bar{h}}{k+1} \|\bar{\mathbf{A}}_k\| = \|\mathbf{e}_{tol}\|.$$

Отсюда, используя последнее соотношение в (2.4), получаем оценку

$$\bar{h} = hr = h \left(\frac{k+1}{h} \frac{\|\mathbf{e}_{tol}\|}{\|\mathbf{A}_k\|} \right)^{\frac{1}{k+1}}. \quad (5.1)$$

Очевидно, при разбиениях Гаусса–Радо и Гаусса–Лобатто недостаток такой оценки состоит в том, что шаг по ней выбирается как для решения порядка k , поэтому, вообще говоря, она не обеспечивает сохранение локальной точности.

Во избежание слишком больших (и малых) локальных ошибок на r следует наложить ограничение:

$$\frac{1}{\sigma} < r^{k+1} < \sigma. \quad (5.2)$$

Для того чтобы величина последнего члена в (2.6) была ограничена в пределах одного порядка, значение σ должно быть равно $\sqrt{10}$. Это следует из того факта, что $\bar{h} \|\bar{\mathbf{A}}_k\| \sim r^{k+1}$.

Выполнение обоих неравенств проверяется лишь в начале интегрирования при выборе стартового шага: если (5.2) не выполняется, то интегрирование повторяется с новым шагом $\bar{h} = hr$ и так далее, пока не выполнится условие (5.2). Обычно для получения стартового шага требуется не более 4 итераций. В дальнейшем для ограничения r проверяется только правое неравенство: если неравенство не выполняется, то r принимает значение правого предела.

Начальное приближение стартового шага получается из оценки (5.1) для $k = 1$:

$$\bar{h} = \sqrt{\frac{2h\|\mathbf{e}_{tol}\|}{\|\mathbf{f}_1 - \mathbf{f}_0\|}}, \quad \mathbf{f}_1 = \mathbf{f}(t_0 + h, \mathbf{x}_0 + h\mathbf{f}_0), \quad (5.3)$$

где h — малая величина. Если h настолько мала, что в компьютерной арифметике $\mathbf{f}_1 = \mathbf{f}_0$, она увеличивается в 10 раз и оценка повторяется снова.

6 Фортран код GAUSS_15

Входные (I) и выходные (O) параметры интегратора Гаусса–Эверхарта

Параметр	Статус	Описание
X	I/O	Интегрируемые переменные. До выполнения процедуры начальные для значения TS, после — конечные для TF.
TS	I	Начальное значение независимой переменной.
TF	I	Конечное значение независимой переменной.
STEP	I/O	Стартовый шаг интегрирования. Условие $ERR \neq 0$ задает режим переменного шага. Если $STEP = 0$, стартовый шаг выбирается по формуле (5.3). После выполнения процедуры STEP — величина предпоследнего шага. При $ERR = 0$ интегрирование выполняется с постоянным шагом STEP.
ERR	I	Задаваемая точность $\ e_{tol}\ $ для выбора переменного шага (5.1).
N	I	Число уравнений.
NOR	I	Порядок интегратора.
NI	I	Максимальное число итераций на шаге. При $NI \leq 0$ итерационный процесс выполняется до сходимости.
NS	O	Число шагов интегрирования за выполнение процедуры.
NBS	O	При $NI \leq 0$ число шагов, на которых итерационный процесс не сходится. В процедуре несходимость итерационного процесса регистрируется, когда число итерации достигает 100.
NF	O	Число обращений к процедуре правых частей.
FUN	I	Название процедуры правых частей f.

Интегратор Гаусса–Эверхарта (начало)

```

SUBROUTINE GAUSS_15(X,TS,TF,STEP,ERR,N,NOR,NI,NS,NBS,NF,FUN)
  ! Gauss-Everhart Integrator (Radau & Lobatto Spacings)
  ! Written by Avdyushev V. (2006)
  IMPLICIT REAL*8(A-H,O-Z)
  PARAMETER (IPSO=100, EPS=1.D-15, SGM=3.16D0)
  REAL*8, ALLOCATABLE :: A(:, :), B(:, :), C(:, :), D(:, :), E(:, :)
  REAL*8, ALLOCATABLE :: SMS(:, :), SPC(:, :), BL(:, :)
  REAL*8, DIMENSION(N) :: X, Y, F, FO, P, XK
  LOGICAL LS, VS, INC, CNV
  COMMON/SPACING/SPACING(56)
  EXTERNAL FUN

  IF(TS.EQ.TF) RETURN
  IF(NOR.LT.2.OR.NOR.GT.15)
    *STOP 'Inadmissible Order!'

```

k и τ_i

```

K=NOR/2
ALLOCATE(SPC(0:K), SMS(K,K))
ALLOCATE(A(K,N), B(K,N), BL(K,N))
ALLOCATE(C(0:K,0:K), D(0:K,0:K), E(0:K,0:K))
SPC(0)=0.DO; L=K*(NOR-K-1); SPC(1:K)=SPACING(L+1:L+K)

```

$$1/(\tau_i - \tau_j)$$

```
DO I=1,K; DO J=1,I-1
SMS(I,J)=1.DO/(SPC(I)-SPC(J))
END DO; END DO
```

$$c_{ij} \text{ и } d_{ij} \text{ (1.9)}$$

```
C=0.DO; C(0,0)=1.DO
D=0.DO; D(0,0)=1.DO
DO J=1,K; DO I=J,K
C(I,J)=C(I-1,J-1)-SPC(I-1)*C(I-1,J)
D(I,J)=D(I-1,J-1)+SPC(J)*D(I-1,J)
END DO; END DO
```

$$e_{ij} \text{ (2.5)}$$

```
E=0.DO; E(0:K,0)=1.DO
DO I=1,K; DO J=1,I
E(I,J)=E(I-1,J-1)+E(I-1,J)
END DO; END DO
```

Нормализация c_{ij} , d_{ij} и e_{ij}

```
DO I=1,K
FLT=DFLOAT(I+1)
C(:,I)=C(:,I)/FLT
D(I,:)=D(I,:)*FLT
E(I,:)=E(I,:)*FLT
END DO
```

$$1/(k+1), \sqrt[k+1]{\sigma} \text{ и } \sqrt[k+1]{1/\sigma}$$

```
DEG=1.DO/DFLOAT(K+1)
RMX=SGM**DEG; RMN=1.DO/RMX
```

Направление интегрирования ± 1 , $|h|$, h , инициализация **A**

```
DIR=(TF-TS)/DABS(TF-TS); H_=DABS(STEP)
VS=ERR.NE.0.DO; CNV=NI.LE.0; INC=.TRUE.
T=TS; NS=0; NBS=0; NST=0; NF=0; IPS=IPSO
3 H=H_*DIR; R=1.DO; LS=.FALSE.
B=0.DO; XK=0.DO
```

Стартовый шаг \bar{h} (5.3) (при STEP = 0)

```
IF(H_.EQ.0.DO) THEN
IF(.NOT.VS)
*STOP 'Zero Constant Step!'
CALL FUN(TS,X,FO); NF=NF+1
H_=EPS; B_=0.DO
DO WHILE(B_.EQ.0.DO)
H_=H_*10.DO; H=H_*DIR; Y=X+H*FO
IF(DABS(TF-TS).LE.H_) GOTO 1
CALL FUN(TS+H,Y,F); NF=NF+1
DO J=1,N; B_=B_+(F(J)-FO(J))**2; END DO
END DO
H_=DSQRT(2.DO*ERR*H_/DSQRT(B_))
H=H_*DIR
END IF
```

Пошаговый процесс (начало)

```

1 IF(DABS(TF-T).LE.H_) THEN
  STEP=H_; H=TF-T; H_=DABS(H)
  R=R*H_/STEP; LS=.TRUE.; END IF

  IF(NS.LE.1) BL=B

```

Интегрирование на шаге (начало). $\bar{A} + \Delta A$ (2.4)

```

Q=1.DO
DO L=1,K
  P=0.DO
  DO M=L,K
    P=P+E(M,L)*B(M,:)
  END DO
  B(L,:)=B(L,)-BL(L,:)
  Q=Q*R; BL(L,:)=P*Q/DFLOAT(L+1)
  B(L,:)=BL(L,)+B(L,:)
END DO

```

 $\bar{\alpha}$ (1.8)

```

DO L=1,K; P=0.DO; DO M=L,K
  P=P+D(M,L)*B(M,:)
END DO; A(L,:)=P; END DO

```

 f_0

```

CALL FUN(T,X,F0); NF=NF+1

```

Итерационный процесс (начало)

```

DO IT=1,IPS

```

```

  DO I=1,K

```

 x_i

```

    S=SPC(I); P=0.DO
    DO J=K,1,-1; P=S*(B(J,)+P); END DO
    Y=X+H*S*(F0+P)

```

 f_i

```

  CALL FUN(T+H*S,Y,F); NF=NF+1

```

 α_i и A

```

  P=(F-F0)/S
  DO J=1,I-1; P=(P-A(J,))*SMS(I,J); END DO
  DO J=1,I; B(J,:)=B(J,)+C(I,J)*(P-A(I,)); END DO
  A(I,:)=P

  END DO

```

Проверка на сходимость итерационного процесса

```

IF(ALL(DABS(Y-XK).LE.DABS(EPS*Y))) EXIT
XK=Y

```

```

END DO

```

Итерационный процесс (завершение)

Регистрация шагов, на которых итерационный процесс не сходится (при $NI \leq 0$)

```
IF(IT.GT.IPSO) THEN
IF(CNV.AND.NS.NE.0) NBS=NBS+1
END IF
```

r (5.1) (при $ERR \neq 0$)

```
IF(VS) THEN
B_=0.DO
DO J=1,N
B_=B_+B(K,J)**2
END DO
IF(B_.NE.0.DO) THEN
R=(ERR/DSQRT(B_)/H_)**DEG
ELSE; R=RMX; END IF
IF(NS.EQ.0) THEN
INC=.NOT.LS.AND.INC
IF(INC.AND.R.GT.RMX.OR.R.LT.RMN) THEN
NST=NST+1; IF(NST.GT.100.AND.R.GT.RMX) GOTO 4
H_=H_*R; GOTO 3
END IF; END IF
4 IF(R.GT.RMX) R=RMX
END IF
```

x_h и \bar{h}

```
P=0.DO
DO J=K,1,-1; P=B(J,:)+P; END DO
X=X+H*(FO+P); T=T+H; NS=NS+1
H=H*R; H_=DABS(H); XK=0.DO
```

Максимальное число итераций на шаге (100 для $NI \leq 0$)

```
IF(CNV) THEN; IPS=IPSO
ELSE; IPS=NI; END IF
```

Интегрирование на шаге (завершение)

```
IF(LS) GOTO 2; GOTO 1
```

Пошаговый процесс (завершение)

```
2 DEALLOCATE(A,B,C,D,E,BL,SMS,SPC)
```

```
END SUBROUTINE GAUSS_15
```

Интегратор Гаусса-Эверхарта (завершение)

```

*0.39535039104876056561567136982732,
*0.60464960895123943438432863017268,
*0.79585009071657115107225536569898,
*0.93587007425480330766872288061033,
*1.000000000000000000000000000000,
*0.05626256053692214646565219103231,
*0.18024069173689236498757994280918,
*0.35262471711316963737390777017124,
*0.54715362633055538300144855765235,
*0.73421017721541053152321060830661,
*0.88532094683909576809035976293249,
*0.97752061356128750189117450042915/
END BLOCK DATA

```

Процедура FUN для задачи двух тел (пример)

```

SUBROUTINE FUN(T,X,F)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION X(4),F(4)
R=DSQRT(X(1)**2+X(2)**2)
F(1:2)= X(3:4)
F(3:4)=-X(1:2)/R**3
END SUBROUTINE FUN

```

Возможные сообщения при выполнении процедуры

Сообщение	Описание
'Inadmissible Order!'	Недопустимый порядок NOR (либо меньше 2, либо больше 15).
'Zero Constant Step!'	Нулевой стартовый шаг в режиме ERR = 0.

7 Интегратор Гаусса–Эверхарта в задаче двух тел

Интегратор тестировался на дифференциальных уравнениях задачи двух тел:

$$\mathbf{r}' = \mathbf{v}, \quad \mathbf{v}' = -\frac{\mu}{|\mathbf{r}|^3}\mathbf{r}.$$

Здесь $\mathbf{r} = (r_1, r_2)$ и $\mathbf{v} = (v_1, v_2)$ — векторы положения и скорости соответственно, а $\mu (= 1)$ — гравитационный параметр. Начальные условия задачи были

$$r_1 = 1 - e, \quad r_2 = 0, \quad v_1 = 0, \quad v_2 = \sqrt{(1+e)/(1-e)}.$$

Они соответствуют однопараметрическому семейству орбит с эксцентриситетом e в качестве параметра и с единичной большой полуосью a . Интегрирование выполнялось на интервале 1000 оборотов для различных эксцентриситетов. При этом исследовались те или иные характеристики интегратора.

7.1 Круговой случай $e = 0$

Как уже было замечено, шаг в интеграторе выбирается таким образом, чтобы сохранялась величина члена $k + 1$ порядка. Если бы интегратор имел порядок k , то таким способом можно было бы обеспечить сохранение локальной точности на всем интервале интегрирования. Однако порядок интегратора Гаусса–Эверхарта существенно выше k . На примере круговой задачи мы исследовали зависимость реальной локальной точности от задаваемой для выбора переменного шага.

В (2.6) величину $k + 1$ члена $\|\mathbf{e}_k\|$ можно оценить как

$$\|\mathbf{e}_k\| \equiv \frac{h}{k+1} \|\mathbf{A}_k\| \approx \frac{h^{k+1}}{(k+1)!} \|\mathbf{x}^{(k+1)}\|, \quad (7.1)$$

тогда как ошибка численного решения порядка p будет

$$\|\mathbf{e}_p\| \approx \frac{h^{p+1}}{(p+1)!} \|\mathbf{x}^{(p+1)}\|. \quad (7.2)$$

В круговом случае решение \mathbf{x} имеет вид

$$x_1 = r_1 = a \cos nt, \quad x_2 = r_2 = a \sin nt, \quad x_3 = v_1 = -an \sin nt, \quad x_4 = v_2 = an \cos nt,$$

где $n = \sqrt{\mu/a^3}$ — среднее движение. Тогда

$$\|\mathbf{x}^{(i)}\| = a\sqrt{1+n^2}n^i. \quad (7.3)$$

Подставляя (7.3) в (7.1) и (7.2), получаем

$$\|\mathbf{e}_p\| \approx \frac{(a\sqrt{1+n^2})^{-\frac{p-k}{k+1}}}{(p+1)!} ((k+1)! \|\mathbf{e}_k\|)^{\frac{p+1}{k+1}}.$$

В частности, для разбиения Гаусса–Радо ($p = 2k + 1$)

$$\|\mathbf{e}_p\| \approx \frac{1}{a\sqrt{1+n^2}} \prod_{i=1}^{k+1} \frac{i}{k+1+i} \|\mathbf{e}_k\|^2. \quad (7.4)$$

На рис. 7.1 представлены оценки реальной локальной ошибки

$$\|\mathbf{e}_p\| = \|\Delta\mathbf{x}\| = \sqrt{\Delta r_1^2 + \Delta r_2^2 + \Delta v_1^2 + \Delta v_2^2}$$

в зависимости от различных ERR(= $\|\mathbf{e}_k\|$) для $p = 11$. Оценивание выполнялось в режиме STEP = 0 на первом шаге, подбираемом в соответствии с задаваемой точностью ERR.

Не трудно видеть, что результаты хорошо согласуются с (7.4). В частности, экспериментальная оценка подтверждает квадратичную зависимость $\|\mathbf{e}_p\| \propto \|\mathbf{e}_k\|^2$, отчего, главным образом, реальная точность численного решения существенно выше задаваемой.

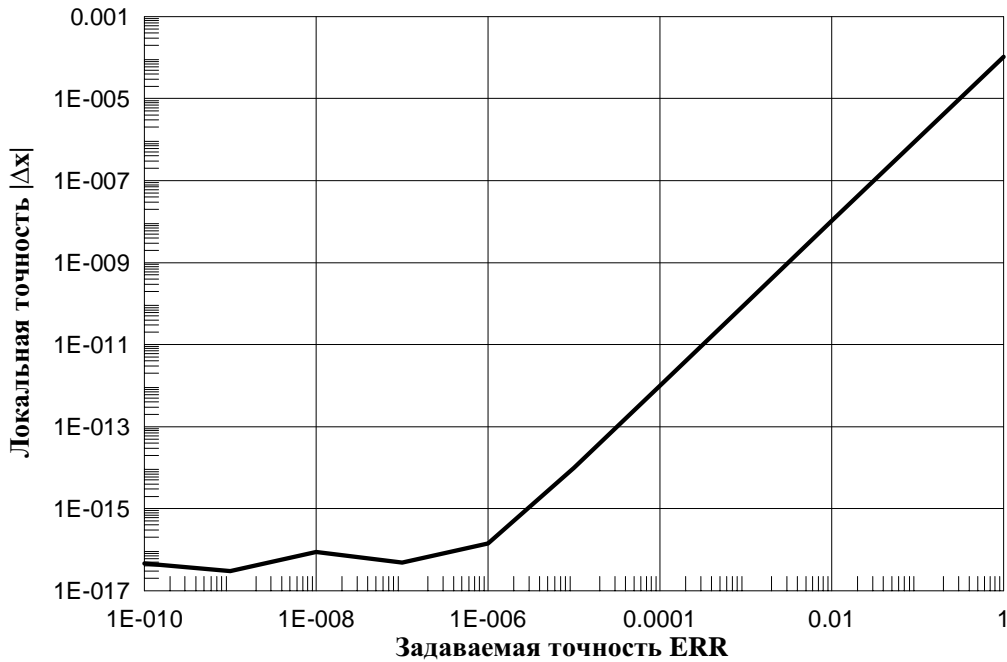


Рисунок 7.1 — Зависимость локальной точности от задаваемой

7.2 Слабоэксцентричный случай $e = 0.1$

Для оценки оптимизации составленного фортран кода мы сравнили быстродействие (в затратах процессорного времени) интегратора GAUSS_15 и широко используемого среди небесных механиков интегратора RADAU_27. При этом в обоих интеграторах выполнялись все операции, определяемые основными формулами. В то же время расхождение двух численных решений, полученных интеграторами, оказалось меньше методической ошибки на несколько порядков.

На рис. 7.2 показаны отношения временных затрат, которые потребовались интеграторам для получения численных решений 7, 11 и 15 порядков. Как видно, интегратор GAUSS_15 работает быстрее, однако этот выигрыш по быстродействию с увеличением порядка уменьшается. Впрочем, следует заметить, что в задачах со сложной функцией \mathbf{f} оперативность интеграторов должна быть одинаковой, поскольку большая часть времени будет затрачиваться на процедуру ее вычисления FUN и тогда быстродействие будет, главным образом, определяться числом обращения к этой процедуре.

Далее мы оценили ошибки интегрирования $|\Delta \mathbf{r}| = \sqrt{\Delta r_1^2 + \Delta r_2^2}$ в зависимости от величины постоянного шага h для порядков 2–11. Интегрирование выполнялось с 6 итерациями (на шаге) на интервале 1000 оборотов.

В случае $k = 1-3$ представленные на рис. 7.3 характеристики показывают, что при достаточно больших h (глобальная) ошибка решения p -порядка, как и ожидалось, с уменьшением шага ведет себя как $|\Delta \mathbf{r}| \sim h^p$, что, собственно, и говорит о соответствии ошибки методу порядка p . Для $k = 4, 5$ ситуация несколько иная. Это объясняется тем, что 6 итераций недостаточно для сходимости итерационного процесса на шаге и получаемое решение не соответствует порядку интегрирующих формул. Случайное поведение характеристик ниже $|\Delta \mathbf{r}| = 10^{-6}$ обусловлено влиянием ошибок округления.

Особо следует заметить, что несмотря на крутой скат характеристики нечетного порядка $2k + 1$ в сравнении с характеристикой четного порядка $2k$ (для $k = 1-3$), определенная точность для четного порядка достигается при большем шаге, т.е. быстрее. Это связано с тем, что при симметричном разбиении Гаусса–Лобатто интегратор Гаусса–Эверхарта обладает геометрическими свойствами и методическая ошибка вдоль независимой переменной t эволюционирует медленнее, нежели при разбиении Гаусса–Радона.

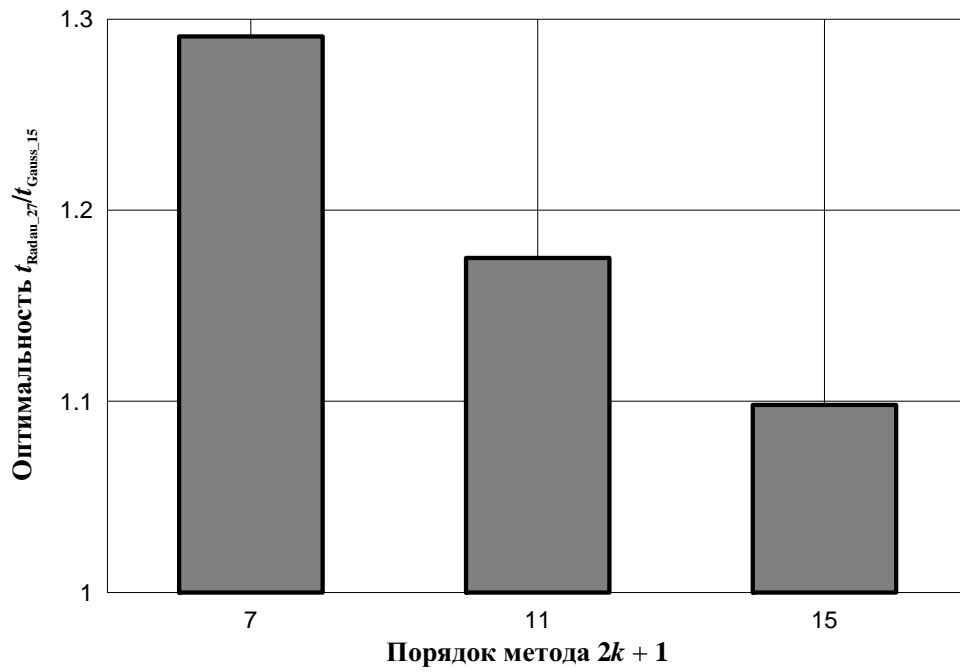


Рисунок 7.2 — Проверка оптимизации интегратора GAUSS_15 в сравнении с RADAU_27



Рисунок 7.3 — Зависимость точности от величины шага интегрирования

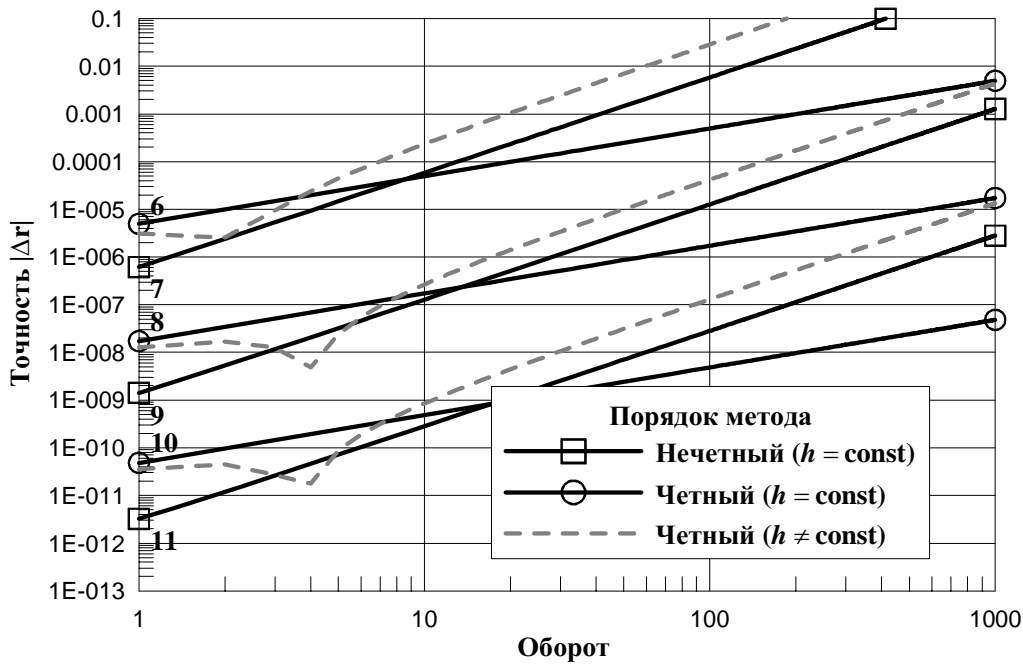


Рисунок 7.4 — Поведение ошибки для решений четных и нечетных порядков

На рис. 7.4 показано поведение ошибки с изменением t для четных и нечетных порядков ($k = 3-5$). Результаты, приведенные здесь, были получены в режиме $NI \leq 0$, т.е. до полной сходимости итерационного процесса. При этом на шаге требовалось от 11 до 16 итераций.

Как видно из рисунка, при постоянном шаге ($h = 2\pi/16$) ошибка интегрирования для четных порядков ведет себя линейно, тогда как для нечетных порядков — квадратически. Даже несмотря на то, что интегратор порядка $2k + 1$ в начале интегрирования дает решение точнее, нежели интегратор порядка $2k$, в конце интегрирования вследствие разного роста ошибок первый уже уступает второму в точности почти на 2 порядка. Очевидно, с увеличением интервала интегрирования это преимущество для каждого интегратора с симметричным разбиением будет только возрастать.

К сожалению, такие замечательные свойства интеграторов четных порядков имеют место только при использовании постоянного шага. При автоматическом выборе шага ошибка также как и для интеграторов нечетных порядков ведет себя квадратичным образом.

Далее мы исследовали вопрос о соотношении точности и быстродействия при увеличении числа итерации на шаге. Путем вариации ERR оценивалась точность и быстродействие интегратора 11-го порядка с переменным шагом на интервале 1000 оборотов для $NI = 1-6$. Результаты приведены на рис. 7.5.

Рисунок показывает, что уже на 2 итерации можно получить довольно хорошее решение, хотя соответствующая характеристика заметно отклоняется от степенной зависимости. Поэтому для уверенного результата необходимо использовать, по крайней мере, 3 итерации на шаге. Следует также заметить, что увеличение числа итераций не существенно понижает эффективность интегрирования.

7.3 Сильноэксцентричный случай $e = 0.9$

Очевидно, сильноэксцентричные орбиты необходимо интегрировать с переменным шагом. Мы проверили, каким образом выбирается шаг для случая $e = 0.9$. Поведение выбираемого шага на одном обороте представлено на рис. 7.6, где также приведены функции пропорциональные $|\mathbf{r}|$, $|\mathbf{r}|^{3/2}$, $|\mathbf{r}|^2$ и $|\mathbf{v}|^{-1}$. Как видно, изменение шага очень близко к поведению $|\mathbf{r}|^{3/2}$. Это означает, что шаг переменной величины по t соответствует почти постоянному шагу по так называемой эллиптической аномалии.

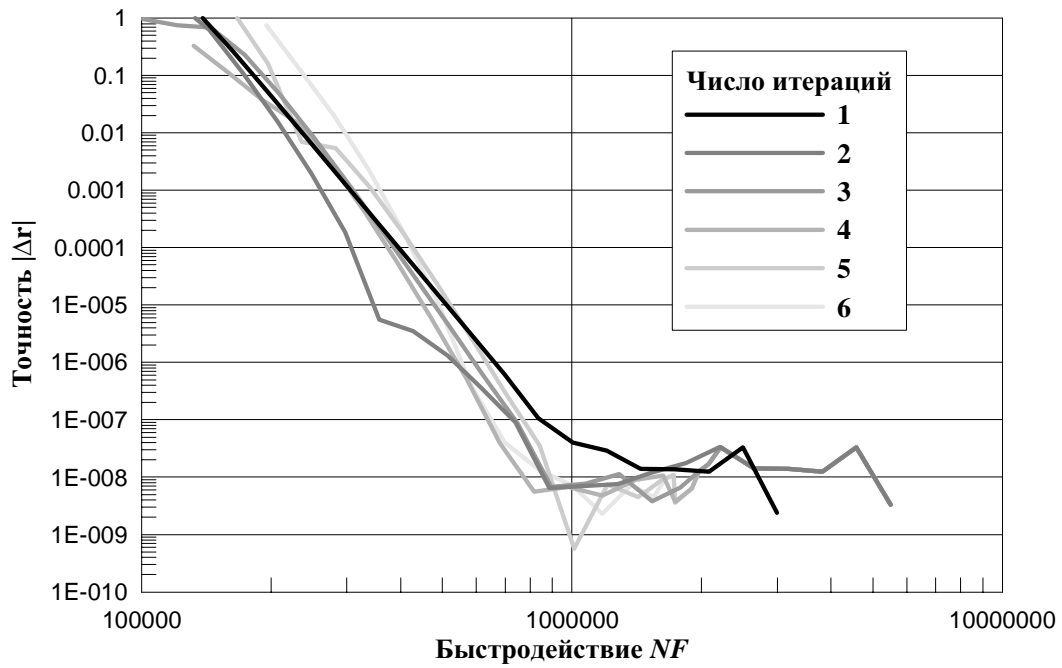


Рисунок 7.5 — Характеристики точность–быстродействие в зависимости от числа итераций на шаге

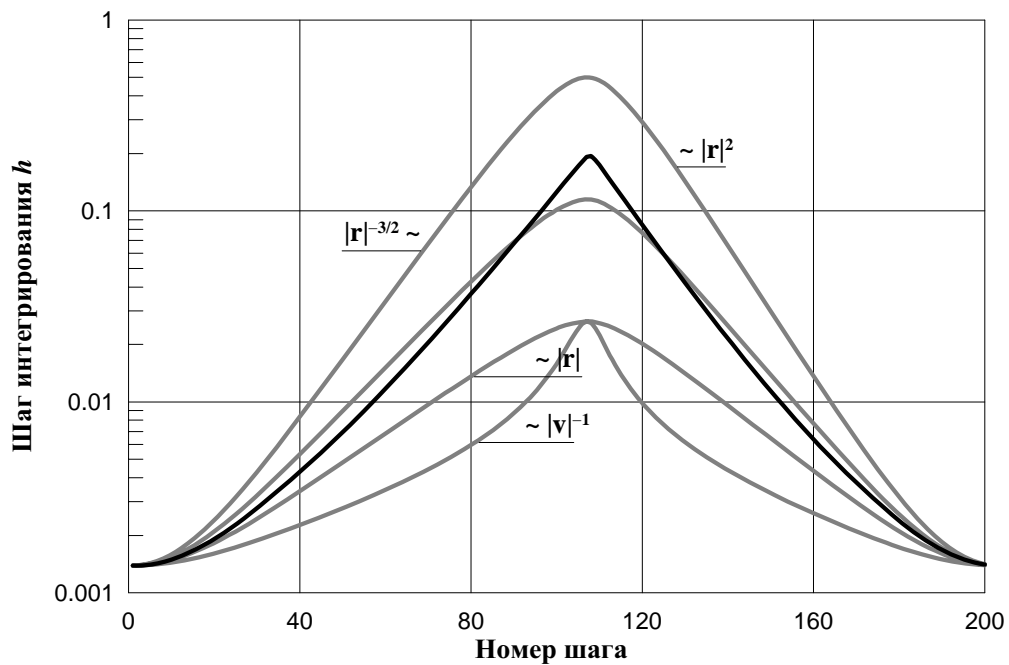


Рисунок 7.6 — Выбор шага интегрирования в случае сильновытянутой орбиты

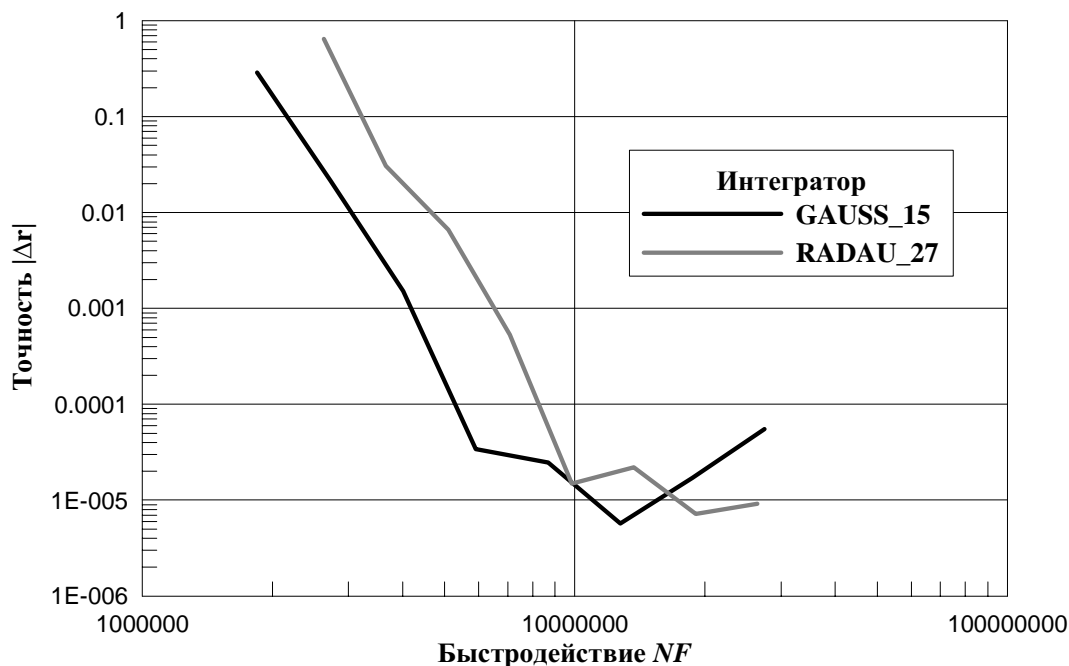


Рисунок 7.7 — Характеристики точность–быстродействие для интеграторов GAUSS_15 и RADAU_27 в экстраэксцентричном случае

7.4 Экстраэксцентричный случай $e = 0.999$

Наконец мы исследовали алгоритм выбора шага при долгосрочном интегрировании очень вытянутой орбиты с эксцентриситетом $e = 0.999$. Интегрирование выполнялось на интервале 1000 оборотов двумя интеграторами: GAUSS_15 и RADAU_27. Результаты показали (рис. 7.7), что RADAU_27 заметно менее эффективен, нежели GAUSS_15. Так, при одинаковом быстродействии первый интегратор дает решение с худшей точностью, ниже на порядок и более. Это связано с тем, что в RADAU_27 заложен неверный алгоритм выбора шага для интегрирования систем с уравнениями первого порядка, а именно алгоритм не точно соответствует формуле (5.1): вместо степени $1/(k+1)$ в RADAU_27 используется $1/(k+2)$. Очевидно, эта ошибка легко устраняется.

Литература

1. Everhart E. A New Method for Integrating Orbits // Bulletin of the American Astronomical Society. 1973. V. 5. P. 389.
2. Everhart E. Implicit Single Sequence Methods for Integrating Orbits // Cel. Mech. 1974. V. 10. P. 35–55.
3. Everhart E. An Efficient Integrator That Uses Gauss–Radau Spacings // Dynamics of Comets: Their Origin and Evolution, Proceedings of IAU Colloq. 83, held in Rome, Italy, June 11–15, 1984. Edited by A. Carusi and G. B. Valsecchi. Dordrecht: Reidel, Astrophysics and Space Science Library. V. 115. 1985. P. 185–202.
4. Butcher J.C. Implicit Runge–Kutta Processes // Math. Comput. 1964. V. 18. P. 50–64.
5. Hairer E., Lubich C., Wanner G. Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations // Springer Series in Computational Mathematics. Springer. V. 31. 2002.