数理解析研究所講究録
第811巻 1992年 37-48

ハイブリッド積分アルゴリズムとその応用について

愛媛大学工学部電子工学科 宮広栄一 (Ei-ichi MIYAHIRO)
愛媛大学工学部情報工学科 野田松太郎 (Matu-Tarow NODA)

1 はじめに

与えられた関数を積分したいとき、数式処理で不定積分が求められればそれに越したことはない。しかし数式処理による記号積分は被積分関数の制約が非常に強く、有理関数に限ってみても不定積分が満足に求められる場合はそう多くない。そこで我々は、数式処理の積分アルゴリズムに一部数値計算を融合したハイブリッド積分アルゴリズムを提案した [2][3]。ハイブリッド積分は、記号積分で求めることが困難な浮動小数係数をもつ有理関数の不定積分を求め得る。本論では、有理関数以外の関数や実験結果のように離散点列で与えられたデータに対しても、それらを有理関数近似することにより、ハイブリッド積分を適用して近似不定積分を求める手法を検討する。特に有理関数近似において、近似gcd計算 [7] を有効に利用した新しい方法を提案する [4]。従来の有理関数近似では、極の考察が不十分であった [5][6]。したがって、分子、分母の次数を高次にするには限界があった。しかし、このような場合は分子、分母が近似共通因子を持つため、近似gcd算法の適用によりそれを除去できることを本論で示す。

以下では、**2** でハイブリッド積分アルゴリズムを簡単に記す。また、置換などによる被積分関数の拡張、及びハイブリッド積分の従来の積分法（記号積分、数値積分）に対する優位性について実例で示す。**3** では有理関数近似を用い、ハイブリッド積分で取り扱える範囲を、有理関数以外の関数や実験データへ拡張することを考える。

2 ハイブリッド積分アルゴリズム

ハイブリッド積分のアルゴリズムはすでに [2][3] などで詳しく述べているが、本論では一貫性のためこのアルゴリズムを簡単にまとめる。

- 入力: 浮動小数係数を持つ1変数有理関数 N/D ($\deg N < \deg D$, N, Dは互いに素).
- 出力: 近似不定積分 $\int N/D dx$.
- 方法:

1. 有理項と超越項の分離（2つの方法）

1-1. 近似 Horowitz のアルゴリズム

ApxHor1: 分母の決定 D_1=近似 $\gcd(D, D')$[7], $D_2 = D/D_1$.
ApxHor2: 分子の決定 未定係数法⇒ ガウス消去（数値計算）

1-2. 近似 Hermite のアルゴリズム

ApxHer1: 分母の近似無平方分解 $D = \prod_{i=1}^{k} D_i^i$.

ApxHer2: 部分分数展開（第 1 の形） $\frac{N}{D} = \sum_{i=1}^{k} \frac{N_i}{D_i^i}$, $\deg N_i < \deg D_i^i$.

ApxHer3: 部分分数展開（第 2 の形） $\frac{N}{D} = \sum_{i=1}^{k} \sum_{j=1}^{i} \frac{N_{ij}}{D_i^j}$, $\deg N_{ij} < \deg D_i$.

ApxHer4: 各項の簡略 $\int \frac{N_{ij}}{D_i^j} dx = \frac{-B_j/(j-1)}{D_i^{j-1}} + \int \frac{A_j + B_j'/(j-1)}{D_i^{j-1}} dx$

ただし、$j \geq 2$ の項に適用する

2. 超越項の計算

Hyb0: D_2をモニック多項式に
Hyb1: $D_2 = 0$ の全根を Durand-Kerner 法で決定（数値計算）

実根 : $a_1, a_2, \cdots, a_m$
共役複素根 : $b_1 \pm c_1 i, \cdots, b_n \pm c_n i$

Hyb2: 部分分数展開

$$\frac{N}{D} = \sum_{k=1}^{m} \frac{e_k}{x - a_k} + \sum_{k=1}^{n} \frac{2(f_k x - b_k f_k - c_k g_k)}{x^2 - 2b_k x + b_k^2 + c_k^2} .$$

$R(x) = N/D' \Rightarrow\ e_k = R(a_k)$,
$f_k = \Re[R(b_k + c_k i)], g_k = \Im[R(b_k + c_k i)]$.
ただし、$\Re$は実部、$\Im$は虚部を表す。

Hyb3: 積分公式への代入

$$\begin{aligned} \int \frac{N}{D} dx &= \sum_{k=1}^{m} \{e_k \log|x - a_k|\} \\ &+ \sum_{k=1}^{n} \left\{ f_k \log|x^2 - 2b_k x + b_k^2 + c_k^2| - 2g_k \tan^{-1}\left(\frac{x}{c_k} - \frac{b_k}{c_k}\right) \right\} . \end{aligned}$$

なお前処理部分である有理項と超越項の分離は、記号積分のアルゴリズムを近似的に用いた近似 Horewiz 、近似 Hermite の 2 つのアルゴリズムを記した。この部分は、記号積分では Hermite のアルゴリズムが有名でかつ効率的 [8] とされているが、浮動小数係数の場合に、各々を近似アルゴリズムにし（gcd を近似 gcd で置き換えるなど）比較すると、必ず

しも近似 Hermite のアルゴリズムが有利にならない。計算速度でみるなら、近似 Horowitz のアルゴリズムで用いる高精度のガウス消去法が高速なため、むしろ近似 Horowitz のアルゴリズムの方が有利になる。また超越項の積分に関して、記号積分では Rothstein-Trager のアルゴリズム [1] が有名であるが、浮動小数係数をもつ有理関数に拡張しにくいため、上記の融合アルゴリズムを用いる。具体的な計算例は次の通り。

例題 1) $\int \frac{N}{D}dx = \int \frac{5x-1}{x^3-3x-2.001}dx.$

1. 有理項と超越項の分離

近似 $\gcd(D, D') = 1$ より、有理項はない。

2. 超越項の積分

Hyb1:

$D = 0$ を *Durand-Kerner* 法で数値的に求めると、
$a_1 = 2.0001111,\ b_1 \pm c_1 i = -1.0000555 \pm 0.018256996i$.

Hyb2:

$R(x) = (5x-1)/(3x^2-3)$ より、$e_1 = R(a_1) = 0.99991359$,
$R(b_1 + c_1 i) = -0.49995679 - 54.776058i$ より、
$f_1 = -0.49995679, g_1 = -54.776058$.

Hyb3:

$$\begin{aligned}\int N/D dx &= 0.99991359 \log|x - 2.0001111| \\ &- 0.49995679 \log|x^2 + 2.0001111x + 1.0004444| \\ &+ 109.55211 \tan^{-1}(54.773523x + 54.776566).\end{aligned}$$

積分範囲を-1 から 1 とした定積分値を、数値積分結果と比較すると、

ハイブリッド積分 $\rightarrow$ 164.95627
適応型ニュートンコーツ $\rightarrow$ 164.95627

となり、良好な結果を得ている。しかし微小項の存在のため数式処理システム (Maple,Reduce) による積分では満足な結果が得られない。

例題 2) $\int_0^1 \frac{-1}{x^5 - x^4 - 0.75x^3 + x^2 - 0.25x - 10^{-6}}dx \simeq 5195.24.$

これはハイブリッド積分より求めた結果である。被積分関数は、積分範囲近傍に特異点があり、かつ鋭いピークを持つため数値積分（シンプソン、ロンバーグ、ガウス型、二重指数、適応型ニュートンコーツ）の各公式で適当な結果が得られなかった。

以上のことから明らかなようにハイブリッド積分は

- 記号積分と比較すると浮動小数係数の被積分関数に強力
- 数値積分では得られない不定積分を求めることができる
- 有理関数の積分であれば被積分関数の特徴に依存されない安定した結果を与える

などの利点がある。

ハイブリッド積分の被積分関数は有理関数に限られているが、色々の数式処理を用いた手法によって有理関数に変換できる関数は多い。それらを列挙する。なお、$P(x)$ を x の多項式、$Q(x)$ を x の有理関数とする。

(1) 対数関数（部分積分の利用）

$$\int P(x)\log\left(Q(x)\right)dx = [\int P(x)dx \cdot \log\left(Q(x)\right)] - \int \left(\int P(x)dx \cdot \frac{Q(x)'}{Q(x)}\right) dx\ .$$

(2) 指数関数（$t \leftarrow e^x$ と置換）

$$\int Q(e^x)dx = \int \frac{Q(t)}{t}dt\ .$$

(3) 三角関数（$t \leftarrow \tan(x/2)$ と置換）

$$\int Q\left(\sin(x),\cos(x)\right)dx = \int \left(Q(\frac{2t}{1+t^2},\frac{1-t^2}{1+t^2})\cdot\frac{2}{1+t^2}\right)dt\ .$$

(4) 代数関数（$t \leftarrow ((ax+b)/(cx+d))^{1/n}$ と置換）

$$\int Q(x,\sqrt[n]{\frac{ax+b}{cx+d}})dx\ \ (ad \neq bc) = \int \left(Q(\frac{dt^n-b}{-ct^n+a},t)\cdot\frac{n(ad-bc)t^{n-1}}{(-ct^n+a)^2}\right)dt\ .$$

また、被積分関数の

- 分子の多項式の係数
- 積分範囲の上下限

にパラメタを含む被積分関数の近似不定積分を得ることもできる。

これらの変換を用いて、例えば以下の積分が求められる。

$$\int_{-1}^{1}(x^2+2)\log\frac{x^2+2.22}{x^2+2.2}dx \simeq 0.03665\ ,\quad \int_0^1 \frac{e^{2x}(1.4e^x-10)^2}{e^x+2}\sqrt[3]{\frac{7.8e^x}{e^x-0.9}}dx \simeq 115.070.$$

3 有理関数近似とハイブリッド積分

ハイブリッド積分の適用範囲をさらに広げることを考える。有理関数以外の関数の積分や実験結果のように点列で与えられるデータの積分である。特殊関数の積分などは、これまでの数式処理研究では拡大体の計算や新しい規則の追加などで対応しているが、積分可能な関数の枠はそうは増えない。しかし、このような複雑な積分を現実に要求する工学問題などでは厳密な関数形を知るより定積分の近似値や積分関数のおおまかな挙動を知る方が重要である場合が多い。すなわち、実用的に意味ある積分計算では「実用に耐え得る精度で定性的に正しい積分値」を得ることが重要になる。これは積分されるべき関数がデータ列に置き変わっても同じである。数値積分はこの条件を満たさない。

そこでハイブリッド積分の利用を考える。ハイブリッド積分が近似積分であることから、被積分関数を有理関数で高精度に近似できれば実用的な問題にも十分ハイブリッド積分を利用できる。以下、複雑な関数や与えられたデータ列を有理関数で近似し、ハイブリッド積分を適用するアルゴリズムを考える。ここで以下の点に注意する。

有理関数近似は2つの多項式の比で表される。これを、$f = n/d$ とする。一般に多項式 n, d は互いに素であると解釈され、かつその場合のみが文献にも示され議論されている[5][6]。しかし近似精度を上げようとすると多くの点を用いて近似有理関数を求めなければならない。当然 n, d の次数は高くなり、多くの不必要な零点を含む可能性がある。このとき、近似有理関数は本来の関数やデータ列にはないような多くの極を持ち、定性的にも異なった近似を与え、実用に耐えられなくなる。これを避けるため、近似gcd計算を用いて分子、分母の多項式が近似的に互いに素であるような有理関数近似を得る方法を[4]に示した。ハイブリッド計算による有理関数近似とその積分の計算手順は次のようになる。

[有理関数近似とその積分]

- 被積分関数の有理関数近似 $f_1 = n_1/d_1$ を得る。

  RatApx1: サンプル点を多めに取って通常の有理関数近似を行う。
  RatApx2: 分子と分母の近似gcd g を求め近似除算をする。
  すなわち、$f_2 = n_2/d_2$, ただし $n_2 = n_1/g, d_2 = d_1/g$.
  RatApx3: 上で変換した分母の近似無平方分解をする。
  よって $f_3 = n_2/(d_{21}d_{22}^2 \cdots d_{2k}^k)$ が被積分関数になる。

- 得られた有理関数に対し、**1** のハイブリッド積分を適用する。

ハイブリッド積分というとき、RatApx3をアルゴリズムの中に含むが(ApxHer1)、有理関数近似と積分を区別するため分離して考える。

以上によって複雑な関数を被積分関数とする場合や離散データ列が与えられた場合の積分に対し近似不定積分を得ることができる。これを次の例で示す。

$$\int f_0 dx = \int \frac{1}{x \sin x} dx \ .$$

この一見簡単そうな積分は、通常の手法で不定積分を求めることはできない。そこで被積分関数 f_0(cf. Fig.1) を $0 < x < \pi$で、

$$x_1 = 0.001, \cdots, x_{21} = 3.141; \quad \Delta x = 0.157$$

のような、21 点の等間隔のデータ列に変換する。これを単純に有理関数で近似し f_1(cf. Fig.2) とすると、

$$f_1 = \frac{1.879 \cdot 10^{-3} x^{10} - 2.736 \cdot 10^{-2} x^9 + \cdots - 5.374 \cdot 10^4}{x^{10} - 29.22 x^9 + 104.4 x^8 + \cdots + 2.908 \cdot 10^{-10}}$$

となる。f_1は数値的には高精度な近似式である。しかし関数の振る舞いに着目したとき、いくつかの問題が生じてくる。分母の多項式の次数が大きくなると不必要な零点を持ち有理関数が特異になることが多い。この場合も、f_0は $0 < x < \pi$で連続な関数であるが、f_1は $x \simeq 0.3210$ に特異点を持ち不連続な関数になる。この特異点は本来取り除かれるべきである、分子と分母の近似共通因子が引き起こす。そこで、近似 gcd 計算により f_1の分子と分母の近似 gcd を求めると、

$$\text{近似 gcd}(\text{分子}, \text{分母}) = x - 0.3210 \cdots$$

となり、これを取り除くと、

$$f_2 = \frac{1.879 \cdot 10^{-3} x^9 - 2.676 \cdot 10^{-2} x^8 + \cdots + 1.674 \cdot 10^5}{x^9 - 28.90 x^8 + 9.507 \cdot 10 x^7 + \cdots - 3.576 \cdot 10^{-9}}$$

となる。f_2(cf. Fig.3) は $0 < x < \pi$で連続となる。

次に、$x = 0$ での関数の振る舞いに着目する。f_0は $x = 0$ に 2 位の極を持つが、f_2は関数近似の誤差により、$x = 0$ のごく近傍に 1 位の極を 2 つ持つ。多重極を持つ関数を有理関数で近似すると、誤差のためにそれが厳密な多重極になることはありえず、いくつかの 1 位の極に分散したり、極がなくなったりする。しかし分母の近似無平方分解をすることにより、f_2は、

$$f_3 = \frac{1.879 \cdot 10^{-3} x^9 - 2.676 \cdot 10^{-2} x^8 + \cdots + 1.674 \cdot 10^5}{(x + 2.866 \cdot 10^{-12})^2 (x^7 - 28.90 x^6 + \cdots + 1.674 \cdot 10^5)}$$

となり、$x = 0$ 近くに 2 位の極を得る。このようなハイブリッド処理により、f_3(cf. Fig.4) の関数の振る舞いは f_0に近づく。これは次のことからも確認できる。$f_0 \simeq f_3$ならば、

$$\frac{1}{x \sin x} \simeq \frac{1.879 \cdot 10^{-3} x^9 - 2.676 \cdot 10^{-2} x^8 + \cdots + 1.674 \cdot 10^5}{(x + 2.866 \cdot 10^{-12})^2 (x^7 - 28.90 x^6 + \cdots + 1.674 \cdot 10^5)}.$$

両辺逆数をとって x を消去すると、

$$\sin x \simeq \frac{x(x^7 - 28.90 x^6 + \cdots + 1.674 \cdot 10^5)}{1.879 \cdot 10^{-3} x^9 - 2.676 \cdot 10^{-2} x^8 + \cdots + 1.674 \cdot 10^5}$$

となる。上式左辺をgとすると、これはFig.5に示したように$0<x<\pi$の間できれいなサインカーブを描く。

f_3の不定積分（すなわち区間$[0,\pi]$でのf_0の不定積分,cf. Fig.6）はハイブリッド積分で求められる。

$$\begin{aligned}\int f_3 dx = &-1.000/(x+2.866\cdot 10^{-12}) + 0.054\ln|x-21.58| - 0.078\ln|x-9.170| \\ &+0.159\ln|x-6.283| - 0.318\ln|x-3.141| + 0.322\ln|x+3.151| - 0.112\ln|x+5.617| \\ &+5.742\cdot 10^{-4}\ln|x+2.437| - 7.589\cdot 10^{-10}\ln|x+2.592\cdot 10^{-12}| + 1.879\cdot 10^{-3}x\ .\end{aligned}$$

結果の良さを見るため、被積分関数が滑らかな範囲で定積分を求めて数値積分結果と比較する。

$$\begin{aligned}\int_1^2 f_0 dx\ [\text{数値積分}] &= 0.7303864072 \\ \int_1^2 f_3 dx\ [\text{ハイブリッド積分}] &= 0.7303864072\end{aligned}$$

ハイブリッド積分は、高精度の値を得ていることがわかる。このように、有理関数近似を用いたハイブリッド積分は、有理関数近似さえ高精度であれば、複雑な被積分関数の積分や、離散データ列を与えた場合の積分に強力である。

有理関数近似において近似共通因子を取り除くApxRat2の処理により悪条件性を除去できることを示したが、ApxRat2の利点を別の視点から見るため次の例を考える。

$$f_0 = \frac{1}{1+25x^2}\ .$$

これは多項式近似が困難なRungeの例として知られている。有理関数を有理関数で近似するのは一見無意味に見えるが、より高次の有理関数で近似を行い、その分子と分母から近似共通因子を取り除くとどうなるかに着目する。-1から1の範囲で等間隔に5点、9点、17点のデータ列に変換し、それを単純に有理関数近似した式をそれぞれf_{1-5}、f_{1-9}、f_{1-17}とすると、

$$f_{1-5} = \frac{1.0+2.220\cdot 10^{-16}x^2}{1.0+25.0x^2}\ ,$$

$$f_{1-9} = \frac{1.0-0.5733x-1.304x^2-1.850\cdot 10^{-17}x^3+1.319\cdot 10^{-15}x^4}{1.0-0.5733x+23.69x^2-14.33x^3-32.60x^4}\ ,$$

$$f_{1-17} = \frac{1.0+0.4189x-1.225x^2+0.2264x^3-\cdots+2.402\cdot 10^{-15}x^8}{1.0+0.4189x+23.77x^2+10.70x^3-\cdots-26.47x^8}$$

となる。各近似式は、一見全然別の形をしているが、分子と分母の近似gcdを求めるとそれぞれ0次、2次、6次の多項式となり、これを除去して小数第7位以下を四捨五入すると、

$$f_{2-5} = f_{2-9} = f_{2-17} = \frac{1.0}{1.0+25.0x^2}$$

となり、結局同じ関数になりしかもそれはもとの関数 f_0 と一致している。すなわち不必要なデータを多く取るほど近似 gcd の次数が高くなり、それを除去することで必要最低限の次数の有理関数近似が得られる可能性を表している。逆に、近似 gcd の次数が高いときデータの取り方に無駄があるとも考えられ、このことを RatApx1 の選点法にフィードバックすればより良い近似が得られるものと思われる。

4 おわりに

以上で述べたように、与えられた関数または実験結果などのデータ列から作られた近似有理関数を被積分関数とすることによって、ハイブリッド積分の適用範囲を拡大することができた。近似不定積分から得られた定積分の値は、数値積分結果と比較して良好である。ここで行った有理関数近似は伝統的な有理関数近似の枠を越え、今後広い応用分野を持つと思われる。この算法をまとめると以下のようになる。

1. 関数またはデータ点列を単純な有理関数で近似する。有理関数の分子、分母多項式の係数はガウス消去法により数値的に求める。
2. 分子、分母の多項式の近似 gcd を計算し、近似共通因子を求める。
3. 分子、分母多項式を近似共通因子で近似除算して近似的に互いに素なものにし、有理関数を再構成する。この段階で有理関数の次数は最初のものより低次となり、かつ不必要な極が除去される。
4. さらに分母の近似無平方分解を行い、高位の極を取り出す。

このようにして元の関数やデータ列を、定量的にも定性的にもうまく近似できる。実際の工学問題の多くは極めて高精度の計算結果を要求するものではないので、ここで用いる近似 gcd のカットオフ値もさして小さくとる必要はない。本論では触れなかったが、極を含むデータ列の積分値を求めたい実際の設計問題に本手法を適用し、数式処理により特異点近傍でコーシー主値積分を併用することで、従来の手法では満足な結果が得られなかった問題に良い結果を与えることができたことをつけ加えておく。

以上の手法はいまだ提案されたばかりで、多くの課題を残している。それらは、

- 初めの有理関数近似に別の方法（連分数近似、パデ近似他）を用いた場合との比較、
- データ列の選び方をある種の適応型にした場合との比較、
- データ数と近似共通因子の次数の関係、
- 有理関数近似とハイブリッド積分の誤差の関係及び得られた関数の近似度の考察、

などである。

なお、本研究は部分的に科学研究費補助金（一般研究 (c), 課題番号 03808003 ）の補助によって行った。

参考文献

[1] Davenport, J. H., Siret, Y. and Tournier, E., Computer Algebra, Academic Press, 1988

[2] Noda, M. T. and Miyahiro, E., On the Symbolic/Numerical Hybrid Integration, Proc. ISSAC90, ACM, 1990, 304.

[3] Noda, M. T. and Miyahiro, E., A Hybrid Approach for the Integration of a Rational Function, JCAM, March, 1992 (to appear).

[4] 野田, 宮広, 甲斐, 近似 GCD を用いた有理関数近似, 数理解析研究所講究録「非線形連立方程式の大域における数値解法とその応用」, 1992（掲載予定）.

[5] Revlin, An Introduction to the Approximation of Functions, Blaisdell Pub. Co., 1969, 120-141.

[6] Rice, J. R., The Approximation of Functions 2, Addison-Wesley Pub, Co., 1969, 76-122.

[7] Sasaki, T. and Noda, M. T., Apporoximate Square-free Decomposition and Root-finding Ill-conditioned Algebraic Equations, J. Inf. Proc, 1989, 159-168.

[8] Yun, Y. Y., Fast Algorithm for Rational Function Integration, IFIP, 1977, 493-498.

Fig.1 $\mathrm{Plot}[f_0, \{x, -1, 4\}, \{f_0, -40, 100\}]$

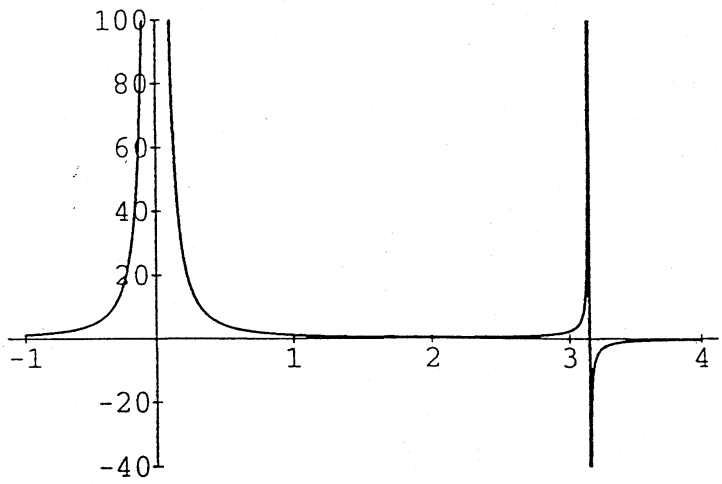
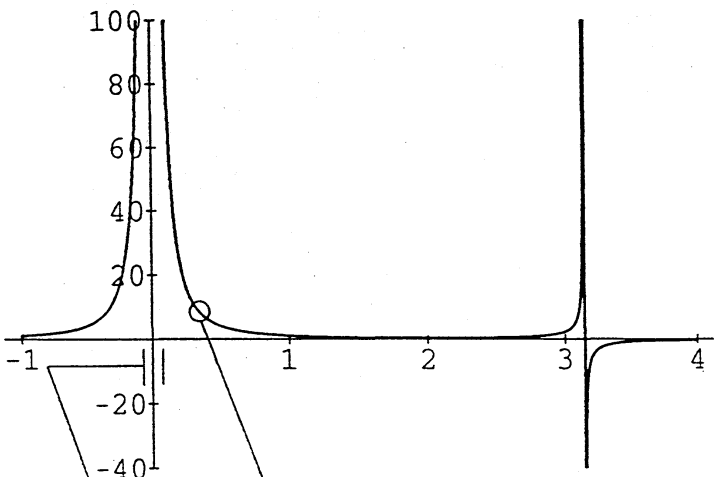
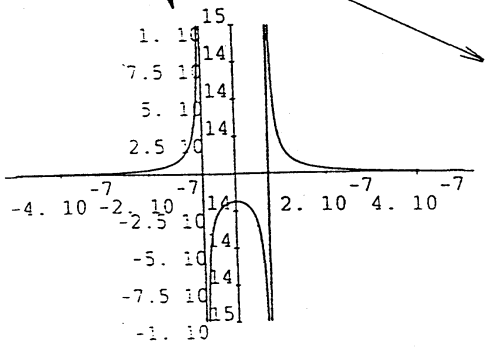
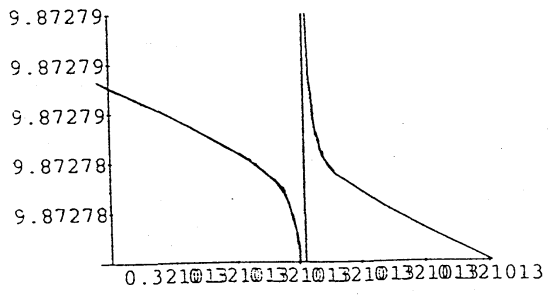


Fig.2 $\mathrm{Plot}[f_1, \{x, -1, 4\}, \{f_1, -40, 100\}]$



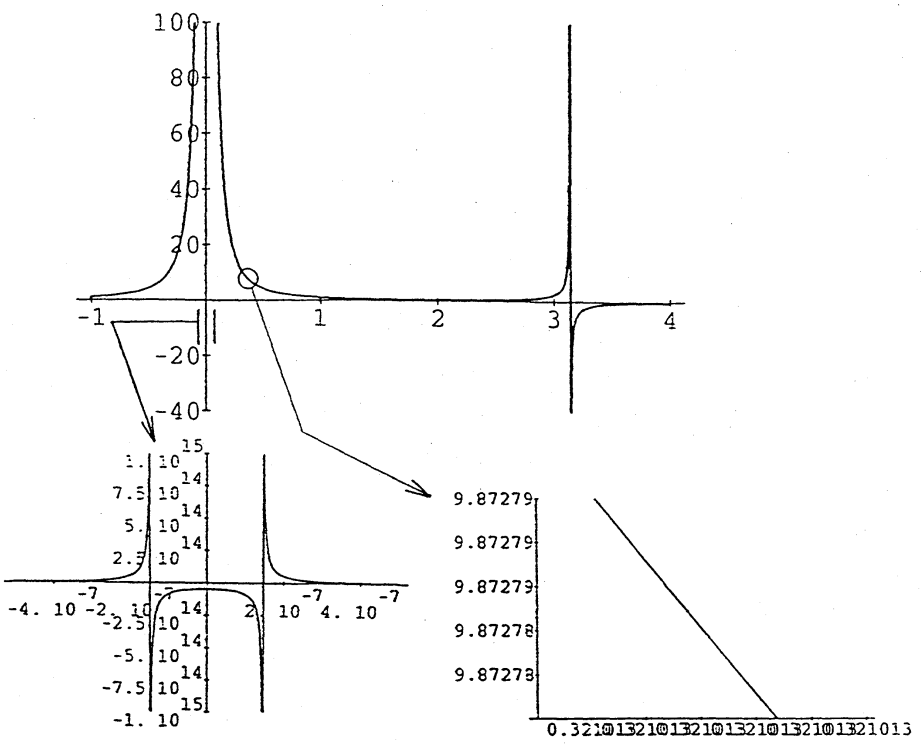


$\mathrm{Plot}[f_1, \{x, -0.5 * 10^{-6}, 0.5 * 10^{-6}\}, \{f_1, -10^{15}, 10^{15}\}]$



$\mathrm{Plot}[f_1, \{x, 0.3210128, 0.3210131\}, \{f_1, 9.87278, 9.87279\}]$

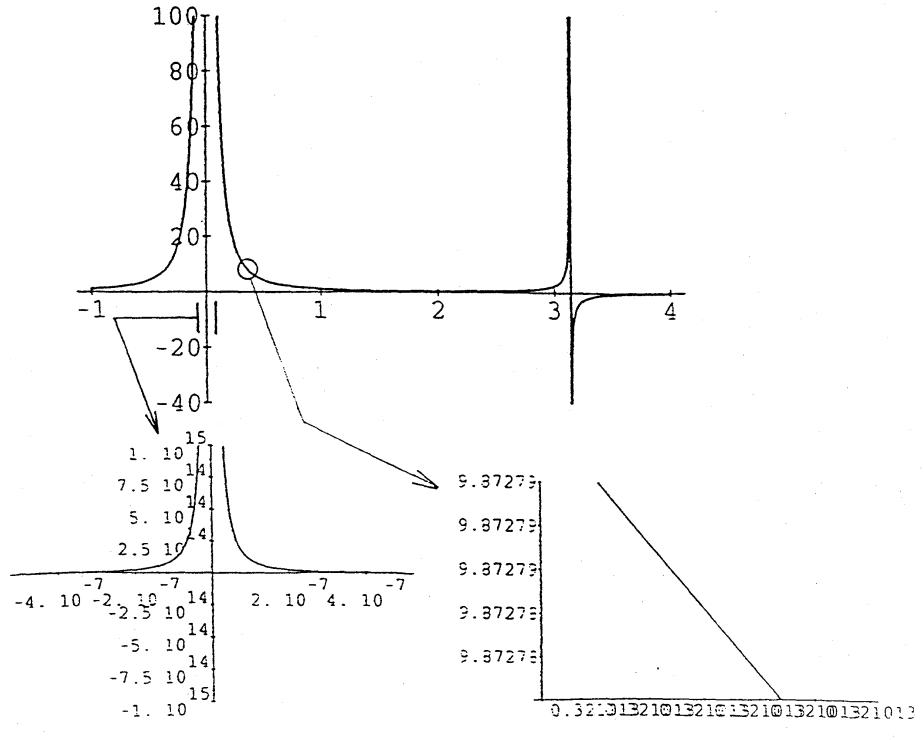
Fig.3 $\mathrm{Plot}[f_2, \{x, -1, 4\}, \{f_2, -40, 100\}]$



$\mathrm{Plot}[f_2, \{x, -0.5 * 10^{-6}, 0.5 * 10^{-6}\}, \{f_2, -10^{15}, 10^{15}\}]$

$\mathrm{Plot}[f_2, \{x, 0.3210128, 0.3210131\}, \{f_2, 9.87278, 9.87279\}]$

Fig.4 $\mathrm{Plot}[f_3, \{x, -1, 4\}, \{f_3, -40, 100\}]$



$\mathrm{Plot}[f_3, \{x, -0.5 * 10^{-6}, 0.5 * 10^{-6}\}, \{f_3, -10^{15}, 10^{15}\}]$

$\mathrm{Plot}[f_3, \{x, 0.3210128, 0.3210131\}, \{f_3, 9.87278, 9.87279\}]$

Fig.5

Plot[$g, \{x, -6, 9\}, \{g, -1, 1\}$]

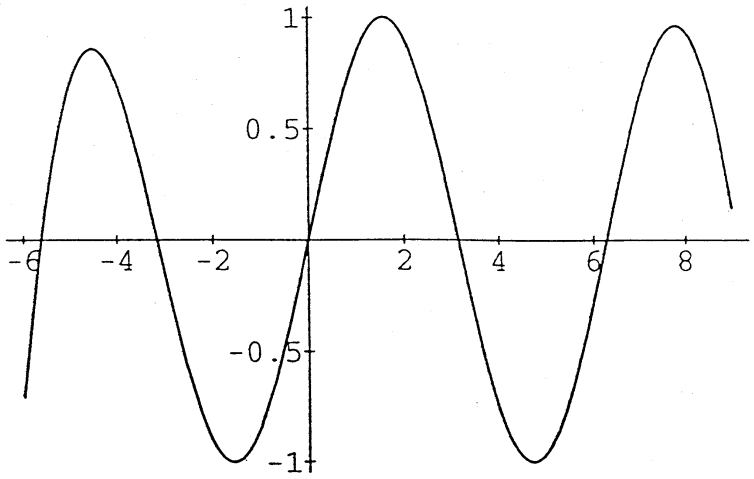


Fig.6

Plot[$f_4, \{x, -1, 4\}, \{f_4, -8, 8\}$]