

Modelagem computacional da interseção de superfícies

Praciano-Pereira, T

Departamento de Matemática
Universidade Estadual Vale do Acaraú

13 de agosto de 2007

tarcisio@member.ams.org

pré-prints do Curso de Matemática de Sobral

no. 2007.3

Editor Tarcisio Praciano-Pereira

tarcisio@member.ams.org

Resumo

Estou mostrando neste trabalho como podemos fazer uso do `gnuplot` para visualizar alguns vetores tangentes a uma curva obtida como interseção de duas superfícies do tipo $f(x, y, z) = 0$ em que f é uma função de classe \mathcal{C}^1 . O exemplo ao final é a interseção de um parabolóide com um plano mostrando os vetores tangentes contidos no plano. Um programa em `C` é descrito na última parte do artigo e pode ser obtido para gerar esta modelagem com alteração dos parâmetros.

palavras chave: campo de vetores tangentes, modelagem computacional, `gnuplot`.

1 Interseção de variedades

Vou calcular a equação de uma reta tangente à interseção das variedades de dimensão 2 (superfícies).

Hipótese 1 (ponto comum) *existência de um ponto comum*
A interseção das variedades

$$\begin{cases} f(x, y, z) = 0 \\ g(x, y, z) = 0 \end{cases} \quad (1)$$

passa no ponto $P = (a, b, c)$, que vou supor que seja conhecido.

Esta hipótese é muito restritiva uma vez que dadas duas superfícies, o cálculo da interseção das mesmas, no fundo a resolução de um sistema de equações, é um problema que não temos meios formais de resolver salvo em alguns casos muito particulares.

O leitor logo verá que precisamos desta hipótese e naturalmente, o trabalho somente se aplicará quando ela for verdadeira. Por outro lado, se este problema tiver solução, uma varredura com uma malha suficiente fina de um domínio compacto com fronteira pode responder pelo menos se a solução existe permitindo uma varredura mais fina para encontrar uma solução aproximada posteriormente. Vou dar exemplos de como isto pode ser feito de forma prática, computacional, mais a frente.

Supor que a interseção das variedades, equação (1), tem o ponto comum $P = (a, b, c)$, significa que

$$f(a, b, c) = 0 ; g(a, b, c) = 0 \quad (2)$$

A derivada implícita do sistema de equações (1)

$$\begin{cases} \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial z} dz = 0 \\ \frac{\partial g}{\partial x} dx + \frac{\partial g}{\partial y} dy + \frac{\partial g}{\partial z} dz = 0 \end{cases} \quad (3)$$

é o modelo das variedades lineares tangentes às mesmas (planos tangentes neste caso).

O cálculo das derivadas parciais no ponto $P = (a, b, c)$ vai transformar o modelo na equação (3) nos planos tangentes a estas variedades neste ponto, e a interseção destes planos (se for uma reta)¹ será uma reta tangente à interseção das variedades (uma curva) neste ponto.

$$\begin{cases} \frac{\partial f}{\partial x}|_{(a,b,c)}(x-a) + \frac{\partial f}{\partial y}|_{(a,b,c)}(y-b) + \frac{\partial f}{\partial z}|_{(a,b,c)}(z-c) = 0 \\ \frac{\partial g}{\partial x}|_{(a,b,c)}(x-a) + \frac{\partial g}{\partial y}|_{(a,b,c)}(y-b) + \frac{\partial g}{\partial z}|_{(a,b,c)}(z-c) = 0 \end{cases} \quad (4)$$

$$\begin{pmatrix} \frac{\partial f}{\partial x}|_{(a,b,c)} & \frac{\partial f}{\partial y}|_{(a,b,c)} & \frac{\partial f}{\partial z}|_{(a,b,c)} \\ \frac{\partial g}{\partial x}|_{(a,b,c)} & \frac{\partial g}{\partial y}|_{(a,b,c)} & \frac{\partial g}{\partial z}|_{(a,b,c)} \end{pmatrix} \begin{pmatrix} x-a \\ y-b \\ z-c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (5)$$

¹a interseção de dois planos pode ser outro plano

O sistema de equações (5) é um sistema sub-determinado, tendo solução, tem um número infinito de soluções (equação de uma reta ou de um plano).

Vou considerar os dois casos abaixo.

1.1 Quando as variedades forem tangentes

Se todos os determinantes menores (de ordem 2) forem zero significa que os planos são paralelos ou coincidentes, porque os vetores perpendiculares, representados pelas derivadas parciais, são proporcionais, se encontram na mesma direção.

Se os planos tangentes coincidirem, as variedades $f(x, y, z) = 0$ e $g(x, y, z) = 0$ são tangentes neste ponto, e há uma infinidade de retas tangentes, qualquer reta pertencendo ao único plano tangente.

Para obter a equação paramétrica de *uma* reta pertencente ao plano

$$f_x|_{(a,b,c)}(x-a) + f_y|_{(a,b,c)}(y-b) + f_z|_{(a,b,c)}(z-c) = 0 \quad (6)$$

basta reduzir a dimensão da variedade² escrevendo

$$(z-c) = -\frac{f_x|_{(a,b,c)}}{f_z|_{(a,b,c)}}(x-a) - \frac{f_y|_{(a,b,c)}}{f_z|_{(a,b,c)}}(y-b) \quad (7)$$

$$y-b = k(x-a) \quad (8)$$

A equação (8) torna z, y dependentes de x e portanto na equação (7) tem uma única *variável livre*, é uma variedade linear de dimensão 1.

Podemos agora escrever o valor de $z-c$ como função de $x-a$ e determinar a equação da reta a menos da escolha do parâmetro k (há uma infinidade de retas)

$$(x-a, k(x-a), -\frac{f_x|_{(a,b,c)}}{f_z|_{(a,b,c)}}(x-a) - k\frac{f_y|_{(a,b,c)}}{f_z|_{(a,b,c)}}(x-a)) \quad (9)$$

É fácil ver que esta reta não passa no ponto $P = (a, b, c)$ se substituirmos x por a , vamos obter $(0, 0, 0)$.

É uma reta paralela³ à reta procurada, basta fazer a translação somando P e usando o parâmetro t em lugar de $x-a$ por uma questão de elegância:

$$(a+t, b+kt, c - (\frac{f_x|_{(a,b,c)}}{f_z|_{(a,b,c)}} - k\frac{f_y|_{(a,b,c)}}{f_z|_{(a,b,c)}})t) \quad (10)$$

O ponto P será obtido na equação (10) quando usarmos o valor $t = 0$.

1.2 Quando as variedades não forem tangentes.

Se houver um determinante menor diferente de zero, significa que os vetores perpendiculares aos planos tangentes tem direções não coincidentes (não são

²impossível fazer isto! Correto: escolher uma subvariedade de dimensão menor ao estabelecer uma dependência entre variáveis feito na segunda equação

³as derivadas produzem apenas objetos paralelos passando na origem

paralelos) neste caso os planos tangentes a cada uma das variedades determinam uma reta tangente (a interseção dos planos).

Quando resolvermos o sistema de equações (5) já estaremos calculando uma variedade de dimensão menor do que as outras duas, uma variedade linear de dimensão 1 (uma reta).

Para simplificar a notação passo a escrever f_x em vez de $f_x|_{(a,b,c)}$ deixando implícito que estou calculando as derivadas parciais no ponto $P = (a, b, c)$.

Os cálculos feitos no caso dos planos coincidentes, nos mostram que temos tres variáveis

$$t = x - a, r = y - b, s = z - c$$

e que a solução encontrada estará na origem. Será necessário transladar a solução usando o vetor $P = (a, b, c)$. Vou usar também esta simplificação ao escrever as equações.

Para eliminar a variável \underline{s} igualo os seus coeficientes nas duas equações o que vai conduzir à expressão de um determinante.

$$\begin{pmatrix} f_x & f_y & f_z \\ g_x & g_y & g_z \end{pmatrix} \begin{pmatrix} t \\ r \\ s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (11)$$

$$\begin{pmatrix} g_z f_x & g_z f_y & g_z f_z \\ f_z g_x & f_z g_y & f_z g_z \end{pmatrix} \begin{pmatrix} t \\ r \\ s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (12)$$

$$\begin{pmatrix} f_x & f_y & f_z \\ g_z f_x - f_z g_x & g_z f_y - f_z g_y & 0 \end{pmatrix} \begin{pmatrix} t \\ r \\ s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (13)$$

$$r = -\frac{g_z f_x - f_z g_x}{g_z f_y - f_z g_y} t = -\frac{\frac{\partial(f,g)}{\partial(x,z)}}{\frac{\partial(f,g)}{\partial(y,z)}} t \quad (14)$$

$$f_x t + f_y r f_z s = 0 \Rightarrow s = -\frac{f_x}{f_z} t - \frac{f_y}{f_z} r \quad (15)$$

$$s = -\frac{f_x}{f_z} t + \frac{f_y}{f_z} \frac{g_z f_x - f_z g_x}{g_z f_y - f_z g_y} t = \left(-\frac{f_x}{f_z} + \frac{f_y}{f_z} \frac{g_z f_x - f_z g_x}{g_z f_y - f_z g_y}\right) t \quad (16)$$

$$s = \left(-\frac{f_x}{f_z} + \frac{f_y(g_z f_x - f_z g_x)}{f_z(g_z f_y - f_z g_y)}\right) t \quad (17)$$

$$s = \frac{-f_x(g_z f_y - f_z g_y) + f_y(g_z f_x - f_z g_x)}{f_z(g_z f_y - f_z g_y)} t = \frac{f_z(f_x g_y - f_y g_x)}{f_z(g_z f_y - f_z g_y)} t \quad (18)$$

$$s = \frac{(f_x g_y - f_y g_x)}{(g_z f_y - f_z g_y)} t = \frac{\frac{\partial(f,g)}{\partial(x,y)}}{\frac{\partial(f,g)}{\partial(y,z)}} t \quad (19)$$

$$\left(t, -\frac{\frac{\partial(f,g)}{\partial(x,z)}}{\frac{\partial(f,g)}{\partial(y,z)}} t, \frac{\frac{\partial(f,g)}{\partial(x,y)}}{\frac{\partial(f,g)}{\partial(y,z)}} t\right) \quad (20)$$

Na equação (11) temos a derivação implícita do sistema na qual anulamos o coeficiente z para obter a equação (13) com a qual deduzimos o valor r na equação (14).

Retornamos a equação de um dos planos tangentes, na equação (15) para obter agora o valor de s usando o de r e com isto deixamos as três variáveis dependendo do único parâmetro t na equação (20). Estas são as equações paramétricas

de uma reta que passa na origem e podemos assim multiplicar todas as coordenadas pelo denominador diferente de zero na segunda e terceira coordenada e depois somar o ponto $P = (a, b, c)$ para obter as equações paramétricas da reta que passa no ponto P

$$\left(a + \frac{\partial(f, g)}{\partial(y, z)}t, b - \frac{\partial(f, g)}{\partial(x, z)}t, c + \frac{\partial(f, g)}{\partial(x, y)}t\right) \quad (21)$$

Isto pode ser feito se o determinante $g_z f_y - f_z g_y$ e f_z forem diferentes de zero, sendo zero explicitamos outra variável, em vez de z o que será possível porque agora estamos trabalhando com a hipótese de que um destes determinantes menores é diferente de zero.

O ponto de tangência, $P = (a, b, c)$ é obtido quando $t = 0$.

2 O algoritmo computacional

Num primeiro exemplo numérico, usando `gnuplot`, vou encontrar um vetor tangente à interseção de um plano e um parabolóide.

A escolha do plano, como uma das superfícies tem como objetivo apresentar um exemplo bem visível, uma vez que os vetores tangentes à curva interseção estarão localizados no plano e `gnuplot` permite rotações da superfícies o que deixa isto bem evidente, é didático, portanto.

Vou considerar

$$\begin{cases} x^2 + y^2 - z & = f(x, y, z) = 0 \\ 3x - 4y - z + 7 & = g(x, y, z) = 0 \end{cases} \quad (22)$$

a interseção de um parabolóide com um plano. Para encontrar um ponto

$$P = (a, b, c)$$

que pertença a interseção destas figuras onde desejo calcular uma reta tangente, tenho que resolver a equação

$$x^2 + y^2 = 3x - 4y + 7 \quad (23)$$

$$x^2 + y^2 - 3x + 4y - 7 = 0 \quad (24)$$

sendo a equação (24) obtida eliminando z , igualando as primeiras partes das equações em (23).

Como quero uma solução (particular) qualquer posso selecionar um valor de x , por exemplo, $x = 1$, uma vez que tenho agora duas variáveis livres (elimino uma).

Este item é essencial no método (na solução computacional). A idéia consiste em partir de uma solução, como se fosse uma *condição inicial* e seguir calculando *soluções aproximadas*, vizinhas, desta *condição inicial*. Portanto interessa encontrar uma solução exata, se possível para depois escolher outra solução, exata ou aproximada, que melhor sirva a determinado objetivo.

$$y^2 + 4y + 1 - 3 - 7 = 0 \quad (25)$$

$$y = \frac{-4 \pm \sqrt{16+36}}{2} = \frac{-4 \pm 2\sqrt{13}}{2} = -2 \pm \sqrt{13} \quad (26)$$

$$y_1 = -2 - \sqrt{13} \quad (27)$$

$$P = (a, b, c) \quad (28)$$

$$a = 1.0 ; b = -2 - \sqrt{13} ; c = 32.42220510185595717 \quad (29)$$

$$a = 1.0 ; b = -2 + \sqrt{13} ; c = 3.57779489814404282752 \quad (30)$$

Os cálculos aqui foram feitos usando uma linguagem e programação chamada `calc` que é interpretada portanto de uso fácil, pode ser usada como uma máquina de calcular. Os comandos podem ser rapidamente obtidos marcando e colando do texto em \LaTeX diretamente no terminal do `calc` com uma pequena alteração da sintaxe:

```
; define f(x,y) {return 3*x - 4*y + 7;}
f(x,y) defined
; a = 1; b = -2 - sqrt(13)
; f(a,b)
    32.42220510185595717248
; define g(x,y) {return x**2 + y**2;}
g(x,y) defined
; g(a,b)
    ~32.42220510185595717249
```

Outro ponto, vou usar agora $x = -1$

$$x = -1 \quad (31)$$

$$y^2 + 4y + x^2 - 3x - 7 \quad (32)$$

$$y^2 + 4y + 1 + 3 - 7 \quad (33)$$

$$y^2 + 4y - 3 = 0 \quad (34)$$

$$y^2 + 4y - 3 = 0 \quad (35)$$

$$y = \frac{-4 \pm \sqrt{16+12}}{2} = \frac{-4 \pm 2\sqrt{7}}{2} \quad (36)$$

$$y_1 = -2 - \sqrt{7} \quad (37)$$

$$P = (a, b, c) \quad (38)$$

$$a = -1.0 ; b = -2 - \sqrt{7} ; c = 22.58300524425836236 \quad (39)$$

$$a = -1.0 ; b = -2 + \sqrt{7} ; c = 1.41699475574163763799 \quad (40)$$

Vou calcular as derivadas parciais no ponto $P = (a, b, c)$ e escrever as equações paramétricas da reta, agora usando `gnuplot`, isto é, traduzindo para `gnuplot` os cálculos feitos acima,

No `script` do `gnuplot` eu deixei, protegidos por comentários, os comandos que servem para criar a figura em arquivo que foi incluído neste documento em

```

f(x,y,z) = x**2 + y**2 - z
g(x,y,z) = 3.0*x - 4.0*y + 7 - z
fx(x,y,z) = 2.0*x
fy(x,y,z) = 2.0*y
fz(x,y,z) = -1.0
gx(x,y,z) = 3.0
gy(x,y,z) = -4.0
gz(x,y,z) = -1.0
D1(x,y,z) = fx(x,y,z)*gy(x,y,z) - fy(x,y,z)*gx(x,y,z)
D2(x,y,z) = fx(x,y,z)*gz(x,y,z) - fz(x,y,z)*gx(x,y,z)
D3(x,y,z) = fy(x,y,z)*gz(x,y,z) - fz(x,y,z)*gy(x,y,z)
a = 1.0
b = -2.0 - sqrt(13)
c = 3*a - 4*b + 7
x1(x) = a + D3(a,b,c)*x
x2(x) = b - D2(a,b,c)*x
x3(x) = c + D1(a,b,c)*x
set arrow from x1(0),x2(0),x3(0) rto x1(2),x2(2),x3(2)
a = 1.0
b = -2.0 + sqrt(13)
c = 3*a - 4*b + 7
set arrow from x1(0),x2(0),x3(0) rto x1(2),x2(2),x3(2)
a = -1.0
b = -2.0 - sqrt(7)
c = 3*a - 4*b + 7
set arrow from x1(0),x2(0),x3(0) rto x1(2),x2(2),x3(2)
a = -1.0
b = -2.0 + sqrt(7)
c = 3*a - 4*b + 7
set arrow from x1(0),x2(0),x3(0) rto x1(2),x2(2),x3(2)
set arrow from 1.000000,-5.605551,32.512203 rto 1.072111,-5.615551,32.255870
## set terminal postscript portrait enhanced color
## set output 'modelagem_intersecao01.01.eps'
splot x**2 + y**2, 3.0*x - 4.0*y + 7.0

```

L^AT_EX . O resultado do `script` é a representação de quatro vetores tangentes à curva, interseção das duas superfícies mostrando inclusive a orientação da curva, que pode ser visto, muito pobremente, na na figura (1) página 7,

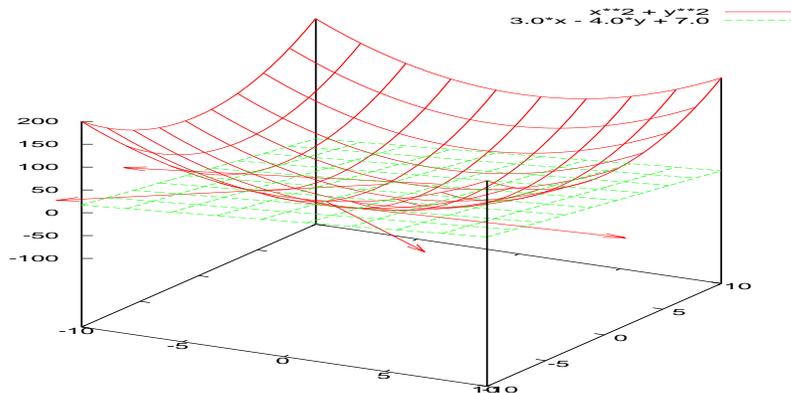


Figura 1: Alguns vetores tangentes a uma curva interseção

A representação dentro de um terminal do `gnuplot` permite rotações que tornam mais viva a representação gráfica.

Este exemplo é interessante porque como uma das superfícies é um plano, então o segmento de reta tangente à interseção está contido no plano o que é visível no gráfico do `gnuplot`.

Pela direção dos vetores você pode ver o sentido de percurso da curva.

Este caso é mais interessante ainda porque a curva obtida é uma elipse portanto podemos escrever a equação da curva e comparar com os resultados obtidos com `gnuplot`. Entretanto o método computacional serve para os casos em que não poderemos encontrar algebricamente a interseção.

Na próxima seção vamos mostrar como podemos gerar uma malha fina de vetores tangentes, com um programa em C tendo como saída de dados um `script` para o `gnuplot` e que chama `gnuplot` passando-lhe o `script`.

3 Modelagem automatizada

Neste momento o programa que estou construindo ainda tem erros e estou fazendo esta pre-publicação com o objetivo mesmo de encontrar leitores interessados na construção num trabalho de equipe. Não tenha dúvidas, rode o programa, analise e se quiser se juntar comigo no trabalho seja bem vindo. O programa se chama `modelagem_intersecao.c` e se encontra pre-publicado junto com este artigo na página.

O programa que complementa este artigo tem uma um objetivo didático, ele mostra como fazer e não tem uma finalidade de conseguir precisão e nem mesmo

de resolver problemas de qualquer natureza. Digamos que esta é a forma com que o autor se exime de qualquer responsabilidade relativamente a resultados que alguém espere obter com o programa e não o conseguir.

Vou dividir o trabalho nesta seção em tres subseções.

Na primeira subseção vou mostrar o planejamento do programa que fará toda a modelagem, esta subseção pode ser útil tanto para que os que desejarem entender o programa e mesmo alterá-lo para seus objetivos específicos, como para quem desejar apenas usar o programa.

A segunda subseção mostra parte da implementação do programa se prendendo aos pontos críticos. O programa pode ser encontrado (o código fonte) na bibliografia em uma página de onde ele pode ser baixado, portanto não interessaria ter aqui todo o programa.

A terceira seção, que vou chamar de *manual do programa* é dirigida aos que desejarem fazer uso direto do programa sem nenhum envolvimento com ele a não ser para produzir alterações pontuais.

Você poderá assim combinar o que lhe interessa na leitura, certamente, a primeira e a terceira, ou a primeira e a segunda.

3.1 O Planejamento do programa

O programa tem três seções:

- entrada de dados, dentro da função *principal*, `main()`;
- uma função `cria_transfere` que tem por objetivo criar um arquivo chamado `transfere` contendo comandos do `gnuplot`
- chamada de `gnuplot` passando-lhe o arquivo `transfere`, no final da função `main()`.

Além destes três pontos essenciais, o programa tem uma função chamada `descricao()` que pode ser chamada ou não por decisão do usuário, que descreve o funcionamento do programa.

Todos os meus programas usam uma biblioteca `ambiente.h` que deve ser baixada junto com o programa que contém funções auxiliares, em geral indicadas nos programas como definidas nesta biblioteca.

Analisando o `script` (tabela 2), página 6, vemos que grande parte do trabalho vai se concentrar na produção dos comandos

```
set arrow from a, b, c rto c,d,e
```

em que `(a,b,c)`, `(c,d,e)` são os extremos do segmento de reta tangente à curva obtidos pelo deslocamento do vetor tangente. Este comando se encontra dentro de um laço localizado numa função que chamada `cria_transfere` porque *cria* um arquivo chamado *transfere* contendo comandos do `gnuplot`.

3.2 Algumas subrotinas do programa

3.3 Manual do programa

Referências

- [1] Buck, R. C. and Buck E. F. *Advanced Calculus* McGraw-Hill - 1965