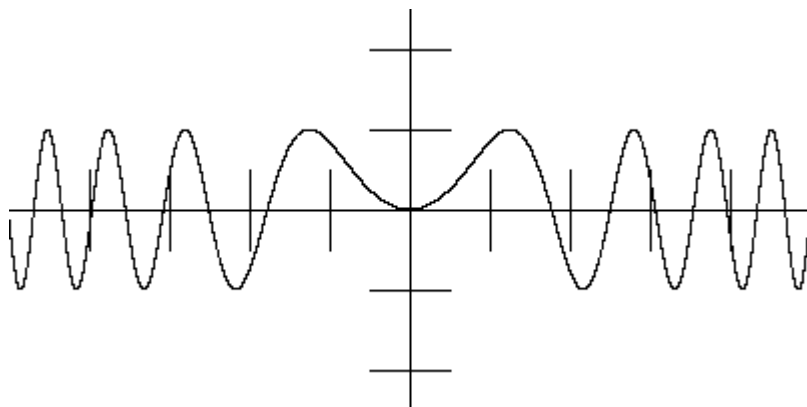


Introduktion

Med denna hemsida kan man se korta sammanfattningar för samt länkar till de senaste uppsatser i matematik. Nyhetsflöden hämtas ur arXiv.org med hjälp av RSS, version 2.0. Länken är <http://export.arxiv.org/rss/math?version=2.0>

Man kan också rita grafer på det två-dimensionella kartesiska systemet. Användaren matar in ett funktionsuttryck med syntaxen för PHP och dess inbyggda matematiska funktioner. En PNG bild genereras.



Grafen för $\sin(X*X)$

Kommentar – Del 1

Hemsidan består av en huvudsida som innehåller två navigationslänkar och en iframe. Klickar man på en navigationslänk så visas den länkade dokumentet i denna iframe. I denna del jag ska fokusera på sidan för nyhetsflöde.

[Home](#) | [Plotter](#)

Motion Reconstruction from Inertial Measurements by Functional Iteration. (arXiv:1708.05004v1 [cs.SY])
[Link to article](#)
Authors: [Yuanxin Wu](#)
Rigid motion computation or estimation is a cornerstone in numerous fields. Attitude computation can be achieved by integrating the angular velocity measured by gyroscopes, the accuracy of which is crucially important for the dead-reckoning inertial navigation. The state-of-the-art attitude algorithms have unexceptionally relied on the simplified differential equation of some kind of kinematic vector, such as the rotation vector, to obtain the attitude. This paper proposes a Functional Iteration technique using the Rodrigues vector (the so-called RodFilter method) to analytically reconstruct the attitude from gyroscope measurements. The RodFilter method is provably exact in reconstructing the incremental attitude as long as the angular velocity is exact. Notably, the Rodrigues vector is analytically obtained and can be used to update the attitude over the considered time interval. The proposed method gives birth to an ultimate attitude algorithm scheme that can be naturally extended to the general rigid motion computation. It is extensively evaluated under the attitude coning motion and compares favorably in accuracy with the mainstream attitude algorithms.

Derivations of Group Algebras. (arXiv:1708.05005v1 [math.RA])
[Link to article](#)
Authors: [A.A. Arutyunov](#), [A.S. Mishchenko](#), [A.I. Shtern](#)
In the paper, a method of describing the outer derivations of the group algebra of a finitely presentable group is given. The description of derivations is given in terms of characters of the groupoid of the adjoint action of the group.

Application of Lagrangian mechanics equations for finding of the minimum distance between smooth submanifolds in N-dimensional

Den första filen som jag kommenterar är news.php. Först "öppnar" jag en ström till <http://export.arxiv.org/rss/math?version=2.0> som innehåller nyhetsflöden i RSS 2.0 format.

`$in_item` är en boolesk variabel som indikerar om, under läsningen, programmet befinner sig i ett `<item>` element. Om programmet är det, försöker programmet sedan att hitta `<title>`, `<link>` och `<description>` elementen och extrahera innehåll från respektive element genom `get()` metoden.

get() metoden tar emot taggens ord (som *title*), sträng-linjen som har lästs och programmet är på, och strömmen *\$file*. *\$file* behövs eftersom ett element kan sträcka sig över flera linjer. I metoden, programmet läser elementets innehåll och sedan returnerar en lista (eng. array) som lagrar innehållet och den resterande strängen i en linje (efter att elementet har dragits bort).

Tillbaka till huvudprogrammet. Programmet kontrollerar att rest-strängen (*values[1]*) inte är tom, i vilket fall det hoppar över resten av while-iterationen, för att inte överskriva eller förlora rest-strängen. Å andra hand om strängen (*values[1]*) har ingen betydelse, så fortsätter programmet att läsa nästa rad. De sista två *continue* satserna i den sista if-else kedjan har samma roll, för att undvika tappa information.

Kommentar – Del 2

Här presenteras grafitaren. Det finns fyra fält. Det första fältet är för funktionens uttryck. Med det andra och tredje fälten specificerar man storleken på bilden. I "scale factor" kan man mata in ett flyttal för att "zooma" in eller ut.

[Home](#) | [Plotter](#)

Enter f(X)

Enter width

Enter height

Enter scale factor

Plot

Instructions:

- Use X as a variable; this is case sensitive
- Use PHP syntax to write the expression
- [Here](#) you find the math functions PHP provides

I det första fältet skriver man ett matematiskt uttryck för funktionen. Uttrycket måste stämma överens med syntaxen för PHP. Det är *eval* som ska "räkna ut" uttrycket. Därför överför programmet först parametrarna till *check.php* som använder *safereval* mappen.

safereval innehåller kod för att filtrera bort uttryck som är en potentiell risk. Jag har inte skrivit *safereval*, men jag har modifierat *config.safereval.php* filen för att tillåta flera matematiska funktioner. När *check.php* är klar, så går programmet vidare till *plot.php*, där själva ritningen sker, genom att skriva: *header("Location: plot.php");*

Notera att *check.php* startar en session med en "hemlig" ID och sätter sessionsvariablerna. Det förra gör jag för att försäkra att det är bara *check.php* som kan använda *plot.php* för att rita en graf.

I *plot.php*, kontrollerar körningen först om ID:et stämmer. Sedan kontrollerar den om uttrycket är giltigt, annars avslutar körningen sessionen och stoppar programmet.

Huvudkoden består nu av:

```
$img = setup();  
plot($img, EQUATION);  
imagepng($img);  
imagedestroy($img);
```

setup() metoden ritar bakgrunden och axlarna med dess respektive markörer. Det är *draw_markers()* som ritar själva markörerna på axlarna. *plot()* metoden ritar själva grafen.

```
2 function plot($img, $equation) {  
3     $color = imagecolorallocate($img, 0, 0, 0);  
4  
5     for ($i = 0; $i < WIDTH/2; $i+=0.01) {  
6         $x0 = $i/(INCREMENT/SCALE);  
7         @eval('$y = ' . str_replace('X', '$x0', $equation) . ';'');  
8         $x0 = $i + WIDTH/2;  
9         $y = HEIGHT/2 - $y*(INCREMENT/SCALE);  
10        imagesetpixel($img, $x0, $y, $color);  
11  
12        $x1 = -$i/(INCREMENT/SCALE);  
13        @eval('$y = ' . str_replace('X', '$x1', $equation) . ';'');  
14        $x1 = -$i + WIDTH/2;  
15        $y = HEIGHT/2 - $y*(INCREMENT/SCALE);  
16        imagesetpixel($img, $x1, $y, $color);  
17    }  
18 }
```

Det är denna for-slingan som utgör kodens kärna. (*INCREMENT/SCALE*) är den faktor som används för att omvandla mellan kartesiska koordinaterna och pixlar. *\$x0* går i den positiva rikningen i x-led. *\$x1* går i den negativa rikningen i x-led. Varje iteration ritar två punkter som är *\$i* pixlar från origo i x-led. Mindre tal som *\$i* ska ökas med (som *0.01* i det här fallet) gör det möjligt att bättre rita grafen när det är en större lutning i grafen.