Korišćenje Furijeovih redova za crtanje pomoću epiciklusa

Seminarski rad u okviru kursa
Naučno izračunavanje
Matematički fakultet

Ivan Ristović

septembar 2019.

Sažetak

Najjednostavniji epiciklus predstavlja krug čiji se centar kreće po kružnici drugog kruga. Složeni epiciklusi nastaju rekurzivnim dodavanjem krugova u pomenuti sistem. Epiciklusi su poznati još od vremena starih Grka i korišćeni su da opišu složena kretanja nebeskih tela. Pokazano je da se slaganjem dovoljnog broja epiciklusa odgovarajućih dimenzija i njihovim kretanjem odgovarajućim brzinama mogu iscrtati najrazličitije orbite bez obzira na njihovu složenost. U ovom radu se opisuje veza izmedju epiciklusa i diskretne Furijeove transformacije i ista se koristi za crtanje proizvoljnih neprekidnih linija dajući fascinantno geometrijsko shvatanje Furijeove transformacije koje se krilo u epiciklusima poznatim od davnina.

Sadržaj

1 Uvod

1.1 Furijeova transformacija

Furijeova transformacija [4] razlaže funkciju u vremenskom domenu (tzv. *signal*) u frekvencije od kojih je sačinjena. Tradicionalna oznaka za Furijeovu transformaciju funkcije f je $\hat{f}$.

Neka je funkcija f periodična na intervalu $[a, b]$ [1]. Tada se f može razviti u *Furijeov red* [2]:

$$f(t) = \frac{a_0}{2} \sum_{k=1}^{\infty} a_k \cos \frac{2\pi kt}{b-a} + \sin \frac{2\pi kt}{b-a}$$

gde važi:

$$a_k = \frac{2}{b-a} \int_a^b f(t) \cos \frac{2\pi kt}{b-a} dt, k = 0, 1, \ldots$$

$$b_k = \frac{2}{b-a} \int_a^b f(t) \sin \frac{2\pi kt}{b-a} dt, k = 1, 2, \ldots$$

U praksi, signal je obično diskretan, pa se neprekidni Furijeov red zamenjuje diskretnom varijantom. Takodje, moguće je formirati kompleksnu reprezentaciju Furijeovog reda koristeći jednakosti:

$$e^{i\theta} = \cos\theta + i \sin\theta$$

$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2}$$

$$\sin\theta = i \frac{e^{-i\theta} - e^{i\theta}}{2}$$

Takva kompleksna reprezentacija Furijeovog reda će biti korišćena kasnije u implementaciji, u narednoj formi:

$$\hat{f}_k = \frac{1}{n} \sum_{j=0}^{n} c_j e^{2\pi ikj/n}, k = 0, 1, \ldots, n-1$$

1.2 Epiciklusi

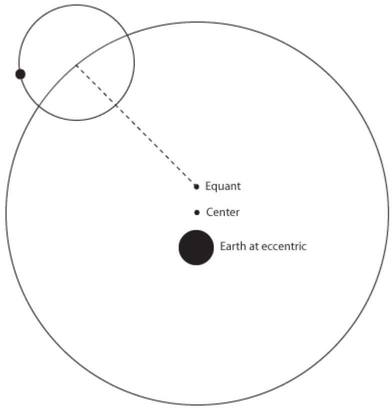
Reč *epiciklus*, u prevodu sa grčkog, znači *na krugu*, tj. *krug na krugu*. Formalno, to je geometrijski model korišćen da objasni varijacije u brzini i smeru kretanja Sunca, Meseca i ostalih planeta. Takav model je od grčkih astronoma preuzeo i unapredio Ptolomej [2]. Po Ptolomeju [3], svaka planeta se kreće po manjoj kružnoj putanji, zvanoj *epiciklus*, dok se ta putanja kreće po većoj kružnoj putanji, zvanoj *deterent* (videti sliku 1.1).

Epicikluse možemo zakomplikovati rekurzivnim dodavanjem drugih epiciklusa. Dodati krugovi ne moraju (i često u primenama nisu) istih dimenzija. Postoje tvrdnje da je Ptolomejev model imao u opisima kretanja nekih planeta i do 80 epiciklusa, u poredjenju sa Kopernikovim sistemom, koji je imao do 34 epiciklusa [3]. Pokazano je da se bilo kakva

[1] U definiciji neprekidne Furijeove transformacije takodje pretpostavljamo da je f integrabilna sa kvadratom na intervalu $[a, b]$

[2] Ptolomejev model je, kao i ostali modeli pre njega, pretpostavljao da je Zemlja u centru svemira. Medjutim, uprkos očito pogrešnoj pretpostavci, njegov model je prvi precizno objasnio kretanje svemirskih tela

[3] Nikola Kopernik je za cilj imao da smanji kompleksnost Ptolomejevog modela, što je i uspeo u svom heliocentričnom sistemu.



Slika 1.1: Ptolomejev model kretanja svemirskih tela.

putanja - bilo periodična ili ne, zatvorena ili otvorena - može proizvoljno dobro aproksimirati *slaganjem* dovoljnog mnogo epiciklusa odgovarajućih dimenzija (videti sliku 1.2). U literaturi se takodje često za orbitu koju poslednjeg kruga koristi termin *epiciklus*.

Epiciklusi se, naravno, mogu predstaviti potpuno formalno matematičkim formulama. Deferent možemo predstaviti kao kompleksan broj

$$z_0 = a_0 e^i k_0 t,$$

gde su a_0 i k_0 konstante, i imaginarna jedinica, a t vreme. Deferent je stoga centriran u koordinatnom početku kompleksne ravni, sa poluprečnikom a_0 i ugaonom brzinom k_0 sa periodom T:

$$k_0 = \frac{2\pi}{T}.$$

Ukoliko je z_1 epiciklus, onda zbir deferenta i epiciklusa predstavlja periodičnu funkciju [4]

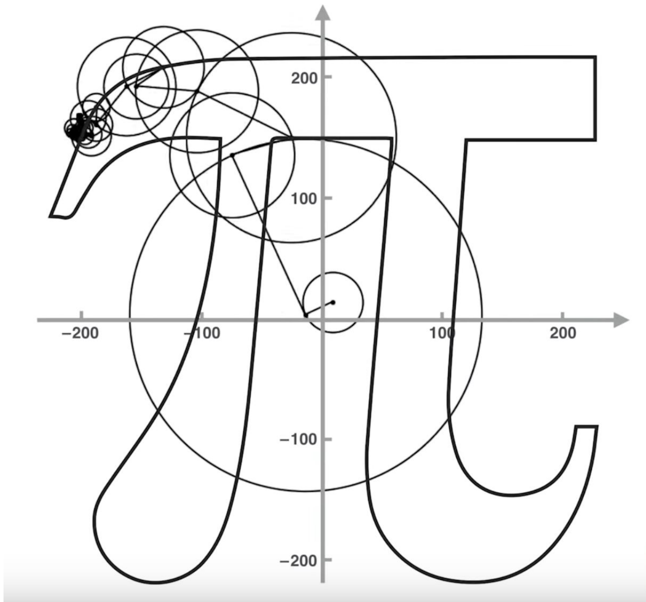
$$z_2 = z_0 + z_1 = a_0 e^i k_0 t + a_1 e^i k_1 t.$$

Generalizovanjem dobijamo opšti član:

$$z_n = \sum_{j=0}^{n} a_j e^i k_j t$$

Postavlja se pitanje: *Šta je zapravo putanja koju najmanji epiciklus pravi, i gde se ona krije u ovoj formuli*? Problem pronalaženja *orbite* najmanjeg epiciklusa se sastoji u pronalaženju vrednosti koeficijenata a_j kako bi se reprezentovao vremenski zavisan put u kompleksnoj ravni. Više o tome u poglavljima koja slede.

[4] Zapravo, ovakve funkcije se nazivaju *skoro periodične funkcije* - zainteresovani čitalac može pročitati više u [1]. Ova funkcija je periodična samo kada je udeo konstanti k_j racionalan.



Slika 1.2: Demonstracija praćenja orbite simbola π pomoću epiciklusa.

2 Crtanje proizvoljnih kontura

Epiciklusi se mogu posmatrati kao krive definisane sledećim jednačinama:

$$\begin{aligned} x(t) &= \sum_{i}^{N} R_i \cos(\omega_i t + \phi_i) \\ y(t) &= \sum_{i}^{N} R_i \sin(\omega_i t + \phi_i) \end{aligned} \tag{1}$$

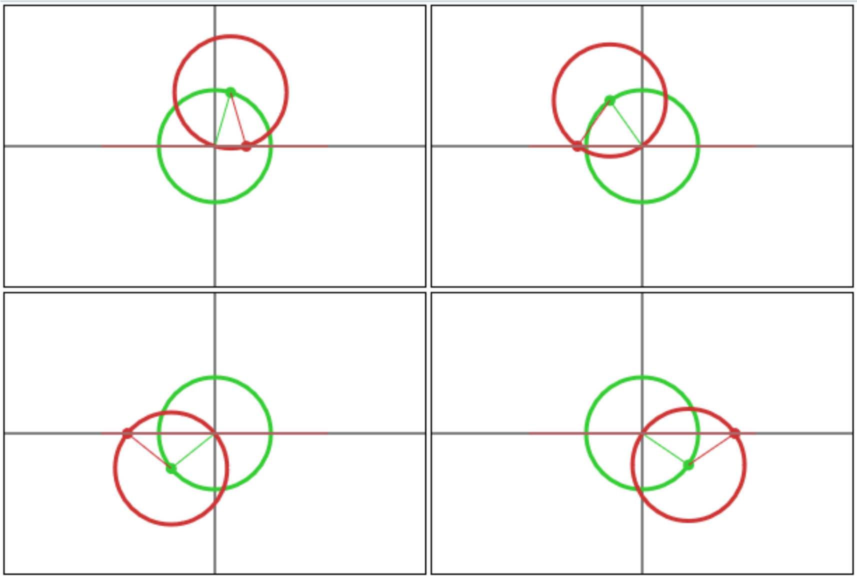
gde N predstavlja broj krugova u epiciklusu, R_i predstavlja radijus kruga i, ω_i predstavlja ugaonu brzinu pridruženu krugu i, a ϕ_i označava za koliki ugao je krug i zarotiran u trenutku $t = 0$ (u daljem tekstu *fazni pomeraj*).

Postavlja se pitanje: *Kakve sve oblike, tj. orbite, možemo opisati epiciklusima*? Iz odeljka 1.2 znamo da se bilo kakva neprekidna linija može opisati epiciklusom, pa čak i prave linije ukoliko su krugovi istih dimenzija i rotiraju se u suprotnim smerovima tako da se y komponente poništavaju (videti sliku 2.1). Jedini problem je zapravo odrediti vrednosti za R, ω i ϕ tako da prate proizvoljnu orbitu.

2.1 Računanje epiciklusa za datu orbitu

Najpre formalizujmo problem. Signal je dostupan u diskretnom obliku, kao skup tačaka (x_j, y_j). Zadatak je pronaći epicikluse tako da se date tačke najbolje spoje. Tačnije, tražimo R_i, ω_i i ϕ_i tako da $x(t)$ i $y(t)$ iz jednakosti 1 što bliže odgovaraju signalu. R_i, ω_i i ϕ_i se onda mogu dobiti rešavanjem odgovarajućeg sistema jednačina.

Primećujemo da u formalizaciji problema moramo *približiti* aproksimaciju signalu. To se može formalizovati na razne načine, najčešće pojmom *srednjekvadratne greške* [2].



Slika 2.1: Iscrtavanje prave linije zahteva samo dva kruga.

Ukoliko posmatramo tačke kao kompleksne brojeve, moguće je iskoristiti Ojlerovu formulu,

$$e^i x = \cos x + i \sin x,$$

i DFT proceduru opisanu u 1.1 kako bismo kreirali algoritam koji je veoma jednostavan za računar. Najpre formirajmo kompleksni broj koristeći jednačine 1:

$$p_j = x_j + iy_j x(t) + iy(t) = \sum_i^N R_i(\cos(\omega_i t + \phi_i) + i \sin(\omega_i t + \phi_i)).$$

Primećujemo da nismo ništa izgubili prelaskom na kompleksnu reprezentaciju, naime x i y su zapravo realni i imaginarni deo komplesnog rešenja. Sada, uz pomoć Ojlerove formule, shvatamo da tražimo promenljive takve da izraz

$$\sum_j^N R_j e^{i\omega_j t + \phi_j}$$

što bliže odgovara kompleksnim brojevima p_j. Štaviše, ukoliko dozvolimo da X_j bude kompleksan broj, izraz postaje:

$$\sum_j^N X_j e^{i\omega_j t}$$

Radi simplifikacije, umesto biranja N ugaonih brzina ω, postavićemo fiksne brzine: $0, 1x, -1x, 2x, -2x, \ldots, N/2x, -N/2x$. Sa ovom pretpostavkom, problem poprima finalni oblik:

Problem 2.1. Pronaći kompleksne brojeve X_j (krugove epiciklusa) minimizujući srednjekvadratnu grešku izmedju kompleksnih tačaka p_j i krive

$$p(t) = \sum_{j=0}^N X_j e^{\frac{2\pi i j}{N} t}.$$

Problem opisan iznad skoro u potpunosti odgovara DFT formi. Stoga, dovoljno je iskoristiti DFT algoritam (videti A), dobiti kompleksne brojeve X_j i zatim izračunati nepoznate parametre:

$$R_j = |X_j|,$$

$$\omega_j = \frac{2\pi i j}{N},$$

$$\phi_j = \operatorname{arctan2}(\Im X_j, \Re X_j).$$

3 Implementacija

Program koji automatizuje proces opisan u poglavlju 2 je dostupan na GitHub-u: `https://github.com/ivan-ristovic/epicyclez`. Implementiran u `C#`-u 7.3, trenutno dostupan samo za `Windows` operativni sistem [5]. Prikaz rada programa se može videti na slici 3.1.

Literatura

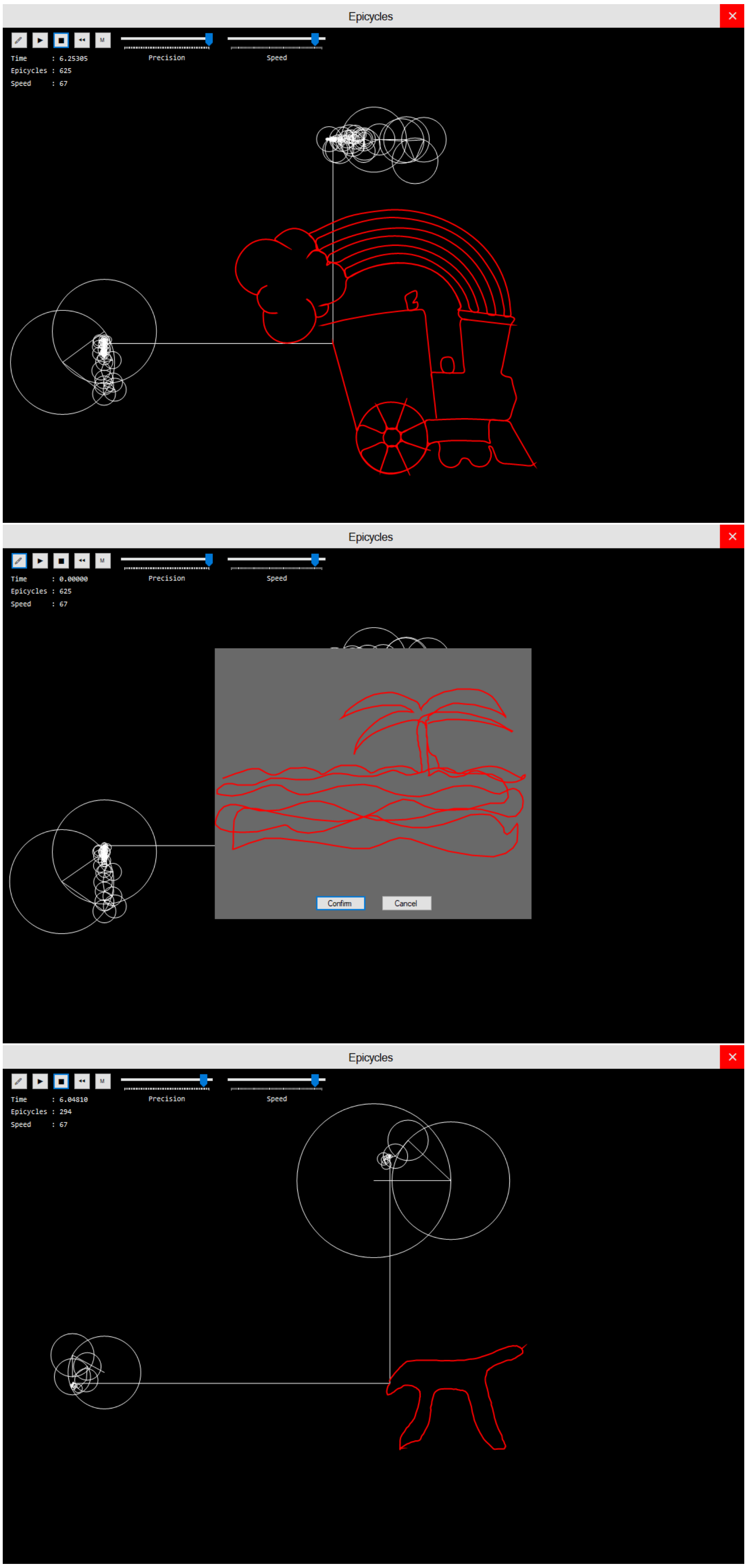
[1] A. S. Besicovitch. *Almost periodic functions.* on-line at: https://pdfs.semanticscholar.org/372e/c6e7c30f961e1bbab0da36839d235b4b01e3.pdf.

[2] Mladen Nikolić and Andjelka Zečević. *Naučno izračuvanje.* on-line at: http://ni.matf.bg.ac.rs/materijali/ni.pdf.

[3] University of Iowa. The Ptolemaic Model. on-line at: http://www.polaris.iastate.edu/EveningStar/Unit2/unit2_sub1.htm.

[4] Stanford Prof. Brad Osgood. *Fourier Transform.* on-line at: https://see.stanford.edu/materials/lsoftaee261/book-fall-07.pdf.

[5] Sa izdanjem 3.0 .NET Core radnog okvira, program će biti moguće pokrenuti na svim operativnim sistemima koji imaju instaliran .NET Core Runtime.



Slika 3.1: Prikaz rada programa. Moguće je crtati proizvoljne ručno iscrtane konture ili učitati signal u `json` formatu.

Dodatak A Implementacija DFT algoritma

```
public sealed class Epicycle
{
    public double Frequency { get; }
    public double Amplitude => this.value.Magnitude;
    public double Phase => this.value.Phase;
    private readonly Complex value;

    public Epicycle(int freq, Complex c)
    {
        this.value = c;
        this.Frequency = freq;
    }
}
```

```
IReadOnlyList<Epicycle>
DFT(IEnumerable<float> values, int? N = null)
{
    var ys = values.ToList();
    if (N is null || N < 1)
        N = ys.Count;
    var dft = new List<Epicycle>(N.Value);

    for (int i = 0; i < N; i++) {
        double re = 0, im = 0;

        for (int n = 0; n < ys.Count; n++) {
            double phi = 2 * Math.PI * i * n / ys.Count;
            re += ys[n] * Math.Cos(phi);
            im -= ys[n] * Math.Sin(phi);
        }

        dft.Add(
            new Epicycle(
                i,
                new Complex(
                    re / ys.Count,
                    im / ys.Count
                )
            )
        );
    }

    return dft
        .OrderByDescending(c => c.Amplitude)
        .ToList()
        .AsReadOnly()
        ;
}
```