

Corpul numerelor complexe

1. Introducere

Dezvoltarea istorică a gândirii matematice a urmărit îndeaproape evoluția ideii de număr. Această evoluție, exprimată succint prin șirul de incluziuni

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C},$$

nu a fost nicidecum liniară ci foarte sinuoasă, cu etape suprapuse sau chiar în ordine inversă.

Primele au apărut *numerele naturale* $1, 2, 3, \dots$, urmate de *fracții și rapoarte*. În lumea antică deja apăruseră semne că numai acestea nu sunt suficiente; de exemplu, incomensurabilitatea diagonalei pătratului cu latura sa era cunoscută din vremea lui Pitagora.

Următoarea achiziție a fost numărul *zero* care, plecând din India, a ajuns în Europa în jurul anului 1000, odată cu scrierea pozițională a numerelor, prin intermediul cărților lui Muhammad al-Khwarizmi, persanul de la numele căruia provine termenul de algoritm.

Numerele *negative* și *imaginare* apar în același timp, în Italia secolului al XVI-lea, în parte tot sub influența surselor islamice și bizantine. Aceste numere, cu denumiri care sugerează o acceptare doar formală a existenței lor, s-au dovedit utile în rezolvarea ecuațiilor algebrice. Astfel, unitatea negativă, -1 , a fost definită ca fiind soluția ecuației $x + 1 = 0$, iar unitatea imaginară, $\sqrt{-1}$, ca soluție a ecuației $x^2 + 1 = 0$, chiar dacă printre numerele deja cunoscute nu exista niciunul care să verifice vreo una dintre aceste două ecuații.

Calculul formal cu *numere complexe*, adică cu expresii de forma $a + b\sqrt{-1}$ s-a dezvoltat prin analogie cu cel al numerelor iraționale pătratice de forma $a + b\sqrt{2}$. În ciuda numeroaselor rezultate obținute cu astfel de numere în secolele XVII-XVIII, matematicienii au avut rețineri cu privire la utilizarea lor. Îndoiala asupra “existenței” acestor numere a dispărut odată cu apariția unei interpretări geometrice foarte simple a lor, și anume reprezentarea lor ca puncte în plan. Folosirea sistematică și popularizarea reprezentării geometrice a numerelor complexe o datorăm lucrărilor lui Carl Friedrich Gauss (1777-1855).

Denumirile nefericite, care mai provoacă și astăzi confuzii, de “numere reale” și “numere imaginare” au fost introduse de René Descartes (1596 – 1650) pentru clasificarea rădăcinilor polinoamelor. Înțelegerea deplină, atât a numerelor reale cât și a celor complexe, a avut loc concomitent, în secolul al XIX-lea, odată cu fundamentarea analizei matematice începută de Augustin-Louis Cauchy (1789 – 1857).

Mulțimea numerelor complexe \mathbb{C} , apărută ca instrument de lucru în rezolvarea ecuațiilor polinomiale, s-a dovedit a fi chiar *cadru natural* al acestui domeniu, dacă luăm în considerație, de exemplu, *Teorema fundamentală a algebrei*, enunțată în 1608 și demonstrată două secole mai târziu, în 1806, care afirmă că orice polinom cu coeficienți reali sau complecși are măcar o rădăcină în \mathbb{C} . Dezvoltarea teoriei numerelor complexe a continuat cu trecerea de la funcții polinomiale la funcții oarecare de o variabilă complexă, cu studiul derivabilității și integrabilității acestora, ajungându-se în final la stabilirea unui capitol fundamental în matematică, capitol numit *Analiză complexă*, prin analogie cu *Analiza matematică* a funcțiilor de variabilă reală.

Teoria funcțiilor de o variabilă complexă a pătruns, prin intermediul ecuațiilor diferențiale mai ales, în numeroase domenii din știință și tehnică. De exemplu, în electrotehnică toate mărimile “sinusoidale” care caracterizează curentul electric sunt definite prin funcții cu valori complexe. Sub presiunea numeroaselor lor utilizări, numerele complexe au ajuns și în limbajele de programare, în C++ sub formă de clase în biblioteca standard iar în C# chiar ca tip predefinit al limbajului.

În cursul pe care îl vom parcurge în acest semestru vom prezenta elementele de bază ale analizei complexe, cu accent pe aplicațiile acestora în principal în grafica pe calculator, dar nu numai. Deoarece în C#, limbajul de programare pe care l-am ales pentru programele noastre ilustrative, modul de lucru cu tipul `Complex` este destul de greoi, după cum se vede, de exemplu, din secvența de cod

```
Complex z = new Complex(12, 6);
Console.WriteLine(z.Imaginary);
```

am preferat să ne definim pentru numere complexe propriul tip de dată, și anume structura `Complex` din fișierul `Complex.cs` atașat.

2. Forma algebrică a unui număr complex

Mulțimea numerelor complexe \mathbb{C} se definește ca fiind mulțimea perechilor de numere reale

$$\mathbb{C} = \mathbb{R} \times \mathbb{R} = \{(x, y) \mid x, y \in \mathbb{R}\}$$

dotată cu două operații binare, adunarea:

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2), \quad \forall (x_1, y_1), (x_2, y_2) \in \mathbb{C},$$

și înmulțirea

$$(x_1, y_1)(x_2, y_2) = (x_1x_2 - y_1y_2, x_1y_2 + x_2y_1), \quad \forall (x_1, y_1), (x_2, y_2) \in \mathbb{C}.$$

Teoremă. $(\mathbb{C}, +, \cdot)$ este un corp comutativ. Prin corespondența $x \rightarrow (x, 0)$, \mathbb{R} devine un subcorp al lui \mathbb{C} .

Demonstrație. Este evident că adunarea și înmulțirea sunt operații interne. Asociativitatea și comutativitatea adunării sunt imediate, $(0, 0)$ este element neutru iar opusul $-z$ al lui $z = (x, y)$ este $(-x, -y)$. Asociativitatea înmulțirii se verifică prin calcul, comutativitatea este evidentă, iar $(1, 0)$ este elementul unitate. Pentru un $z = (a, b) \neq (0, 0)$ găsirea inversului $z^{-1} = (x, y)$ revine la rezolvarea sistemului liniar

$$\begin{cases} ax - by = 1 \\ bx + ay = 0, \end{cases}$$

care, deoarece $\Delta = a^2 + b^2 \neq 0$, are soluția unică $z^{-1} = \left(\frac{a}{a^2+b^2}, \frac{-b}{a^2+b^2}\right)$. Distribuțivitatea înmulțirii față de adunare se verifică ușor prin calcul, și astfel am arătat că $(\mathbb{C}, +, \cdot)$ este un corp comutativ.

În sfârșit, este clar că aplicația $x \rightarrow (x, 0)$, definită pe \mathbb{R} cu valori în \mathbb{C} este injectivă și are proprietățile

$$(x_1, 0) + (x_2, 0) = (x_1 + x_2, 0)$$

$$(x_1, 0)(x_2, 0) = (x_1x_2, 0),$$

de unde rezultă imediat că imaginea sa $\{(x, 0) \mid x \in \mathbb{R}\} \subset \mathbb{C}$, este un subcorp al lui \mathbb{C} izomorf cu corpul numerelor reale \mathbb{R} .

Observație. Fiind dat un număr complex $z = (x, y)$, convenim să numim numerele reale x și y ca fiind *partea reală* și, respectiv, *partea imaginară* a lui z , și vom utiliza notațiile $x = \operatorname{Re} z$, $y = \operatorname{Im} z$. Aplicațiile $z \rightarrow \operatorname{Re} z$ și $z \rightarrow \operatorname{Im} z$, definite pe \mathbb{C} cu valori în \mathbb{R} păstrează operația de adunare, adică

$$\operatorname{Re}(z_1 + z_2) = \operatorname{Re} z_1 + \operatorname{Re} z_2$$

$$\operatorname{Im}(z_1 + z_2) = \operatorname{Im} z_1 + \operatorname{Im} z_2,$$

dar nu și operația de înmulțire.

A doua parte a teoremei de mai sus afirmă că mulțimea numerelor reale \mathbb{R} poate fi identificată, prin izomorfismul de corpuri dat de *aplicația de scufundare* $x \in \mathbb{R} \rightarrow (x, 0) \in \mathbb{C}$, cu mulțimea numerelor complexe cu partea imaginară nulă

$$\mathcal{R} = \{z \mid \operatorname{Re} z = 0\} = \{(x, 0) \mid x \in \mathbb{R}\} \subset \mathbb{C}.$$

Aceasta înseamnă că, în loc de

$$\mathbb{R} \cong \mathcal{R} \subset \mathbb{C}$$

vom scrie

$$\mathbb{R} \subset \mathbb{C},$$

iar numerele complexe de forma $(x, 0)$ vor fi notate cu x atunci când nu există pericol de confuzie.

Mai mult, se verifică imediat că orice număr complex $z = (x, y)$ poate fi descompus sub forma

$$z = (x, y) = (x, 0) + (0, y) = (x, 0) + (0, 1)(y, 0),$$

și, prin urmare, cu notația

$$i = (0, 1),$$

obținem așa numita *formă algebrică* a lui z :

$$z = x + iy,$$

unde, subliniem, x , y și i sunt numere complexe. Din definiția înmulțirii avem că

$$(0, 1)(0, 1) = (-1, 0) = -(1, 0),$$

relație care, cu convențiile făcute, se scrie

$$i^2 = -1.$$

Aplicând regulile de calcul dintr-un corp comutativ, deducem mai departe că

$$\begin{aligned}(x_1 + iy_1)(x_2 + iy_2) &= x_1x_2 + ix_1y_2 + iy_1x_2 + i^2y_1y_2 \\ &= x_1x_2 - y_1y_2 + i(x_1y_2 + y_1x_2),\end{aligned}$$

justificând astfel de ce înmulțirea numerelor complexe a fost definită atât de complicat.

Să urmărim cum au fost implementate cele de mai sus în structura `Complex` definită în fișierul `Complex.cs`.

Egalitatea $\mathbb{C} = \mathbb{R} \times \mathbb{R}$ se reflectă direct în structura unui obiect de tip `Complex`, care este format numai din două câmpuri de tip `double` inițializate de constructorul obiectului:

```
public struct Complex
{
    public double x;
    public double y;

    public Complex(double x, double y)
    {
        this.y = y;
        this.x = x;
    }
    ...
}
```

Definițiile operațiilor cu numere complexe apar, exact sub forma dată, la supraîncărcarea operatorilor corespunzători:

```

public static Complex operator +(Complex zst, Complex zdr)
{
    return new Complex(zst.x + zdr.x, zst.y + zdr.y);
}
public static Complex operator *(Complex zst, Complex zdr)
{
    return new Complex(zst.x * zdr.x - zst.y * zdr.y,
                       zst.y * zdr.x + zst.x * zdr.y);
}

```

Mai mult, este implementată chiar și aplicația de scufundare a lui \mathbb{R} în \mathbb{C} , și anume la definirea operatorului de conversie implicită de la `double` la `Complex`:

```

public static implicit operator Complex(double x){
    return new Complex(x, 0);
}

```

Această conversie implicită ne permite, printre altele, să lucrăm în cod cu numere complexe sub formă algebrică:

```

Complex i = new Complex(0,1);
Complex z1 = 1 + 2*i;

```

Definiție. *Conjugatul* lui $z = x + iy$ este numărul complex $\bar{z} = a - ib$.

Propoziție. Aplicația $z \rightarrow \bar{z}$ are următoarele proprietăți imediate:

- (i) $\overline{z_1 + z_2} = \bar{z}_1 + \bar{z}_2$
- (ii) $\overline{z_1 z_2} = \bar{z}_1 \bar{z}_2$
- (iii) $\overline{\bar{z}} = z$
- (iv) $\bar{z} = z \Leftrightarrow z \in \mathbb{R}$
- (v) $\operatorname{Re} z = \frac{z + \bar{z}}{2}$
- (vi) $\operatorname{Im} z = \frac{z - \bar{z}}{2i}$

Verificăm de exemplu (ii)

$$\overline{z_1 z_2} = (x_1 - iy_1)(x_2 - iy_2) = (x_1 x_2 - y_1 y_2) - i(x_1 y_2 + x_2 y_1) = \overline{z_1 z_2}.$$

Observație. În corpul comutativ $(\mathbb{C}, +, \cdot)$ prin *scădere* se înțelege

$$z_1 - z_2 = z_1 + (-z_2),$$

iar prin *împărțire*

$$\frac{z_1}{z_2} = z_1 \cdot z_2^{-1}.$$

Este clar că din (i) și (ii) urmează și

$$(vii) \overline{z_1 - z_2} = \overline{z_1} - \overline{z_2}$$

$$(viii) \overline{\left(\frac{z_1}{z_2}\right)} = \frac{\overline{z_1}}{\overline{z_2}}$$

$$(ix) \overline{z^n} = \overline{z}^n$$

Definiție. Modulul lui $z = x + iy$ este numărul real $|z| = \sqrt{x^2 + y^2}$.

Propoziție. Aplicația $z \rightarrow |z|$ are următoarele proprietăți:

$$(i) |z|^2 = z \cdot \bar{z}$$

$$(ii) |z| = |\bar{z}|$$

$$(iii) |z| = 0 \Leftrightarrow z = 0$$

$$(iv) |z_1 z_2| = |z_1| |z_2|$$

$$(v) |z_1 + z_2| \leq |z_1| + |z_2|$$

Verificăm de exemplu *inegalitatea triunghiulară* (v):

$$\begin{aligned} |z_1 + z_2|^2 &= (z_1 + z_2)\overline{(z_1 + z_2)} = (z_1 + z_2)(\bar{z}_1 + \bar{z}_2) = \\ &= z_1\bar{z}_1 + z_1\bar{z}_2 + z_2\bar{z}_1 + z_2\bar{z}_2 = |z_1|^2 + (z_1\bar{z}_2 + \overline{z_1\bar{z}_2}) + |z_2|^2 \\ &= |z_1|^2 + 2\operatorname{Re} z_1\bar{z}_2 + |z_2|^2 \leq |z_1|^2 + 2|z_1\bar{z}_2| + |z_2|^2 \\ &= |z_1|^2 + 2|z_1||z_2| + |z_2|^2 = (|z_1| + |z_2|)^2. \end{aligned}$$

3. Forma trigonometrică a unui număr complex

Identificând \mathbb{R} cu o dreaptă pe care se consideră o “axă a numerelor” Ox , urmează că $\mathbb{C} = \mathbb{R} \times \mathbb{R}$ se identifică cu un plan raportat la un sistem de coordonate cartezian xOy , asociind fiecărui punct M de coordonate (x, y) numărul complex $z = x + iy$, numit *afixul* lui M . Vom trece acum la coordonate polare.

Propoziție. Fie $z \in \mathbb{C}$, $z \neq 0$. Există și este unic $\theta \in (-\pi, \pi]$, numit *argumentul* numărului complex z și notat $\arg z$, astfel încât

$$\cos \theta = \frac{\operatorname{Re} z}{|z|} \text{ și } \sin \theta = \frac{\operatorname{Im} z}{|z|}.$$

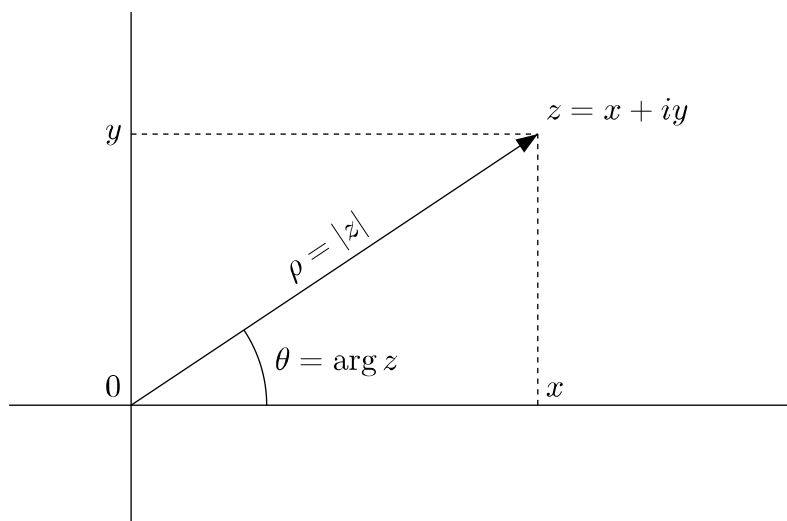
Demonstrație. Fie $z = x + iy$, cu $|z| \neq 0$. Din $|z| = \sqrt{x^2 + y^2} \geq \sqrt{x^2} = |x|$ rezultă că $\frac{x}{|z|} \in [-1, 1]$, deci există

$$\theta_0 = \arccos \frac{x}{|z|} \in [0, \pi].$$

Avem $\sin^2 \theta_0 = 1 - \cos^2 \theta_0 = \left(\frac{y}{|z|}\right)^2$. Dacă $y \geq 0$ definim $\theta = \theta_0$, altfel $\theta = -\theta_0$. Unicitatea lui $\theta = \arg z$ decurge din echivalența

$$\begin{cases} \cos \theta = \cos \varphi \\ \sin \theta = \sin \varphi \end{cases} \Leftrightarrow \exists k \in \mathbb{Z} \text{ a. î. } \theta - \varphi = 2k\pi,$$

prin urmare, $\theta, \varphi \in (-\pi, \pi]$ implică $\theta = \varphi$.



Observație. Fie punctul $M(x, y)$ cu afixul $z = x + iy$. Raza $\rho = |z| \geq 0$ și unghiul $\theta = \arg z \in (-\pi, \pi]$ reprezintă chiar coordonatele polare ale lui M . Legătura dintre cele două tipuri de coordonate este dată de formulele

$$\begin{cases} x = \rho \cos \theta \\ y = \rho \sin \theta \end{cases} \Leftrightarrow \begin{cases} \rho = \sqrt{x^2 + y^2} \\ \theta = \pm \arccos \frac{x}{\sqrt{x^2 + y^2}} \end{cases}$$

unde semnul din fața lui \arccos este dat de semnul lui y .

Este ușor de văzut că argumentul unui număr complex z cu $\operatorname{Re} z \neq 0$ se poate afla și cu formula

$$\theta = \operatorname{arctg} \frac{y}{x} + k\pi,$$

unde k are una din valorile 0, 1 sau -1 , în funcție de cadranul în care cade z .

Cu aceste notații obținem imediat *forma trigonometrică* a lui z ;

$$z = x + iy = \rho \cos \theta + i\rho \sin \theta = \rho(\cos \theta + i \sin \theta).$$

Să observăm că aceste două modalități de exprimare a unui număr complex au fost implementate în structura `Complex` prin metodele statice

```

public static Complex setReIm(double x, double y)
{
    //forma algebrica
    return new Complex(x, y);
}

```

și

```

public static Complex setRoTheta(double Ro, double Theta)
{
    //forma trigonometrica
    return new Complex(Ro * Math.Cos(Theta), Ro * Math.Sin(Theta));
}

```

în timp ce modulul și argumentul unui număr complex sunt date de proprietățile

```

public static Complex setRoTheta(double Ro, double Theta)
{
    //forma trigonometrica
    return new Complex(Ro * Math.Cos(Theta), Ro * Math.Sin(Theta));
}

```

Observație. Conform implemetării de mai sus, argumentul $\theta = \arg z$ returnat de proprietatea `Theta` este în intervalul $(-\pi, \pi]$, conform cerințelor. Acest fapt este utilizat, de exemplu, în următorul test care decide dacă punctul z se află sau nu în semiplanul stâng determinat de dreapta ab parcursă de la a la b :

```

public bool zEsteLaStangaLuiAB(Complex z, Complex a, Complex b)
{
    Complex r=(z-a)/(b-a);
    if (r.Theta > 0) return true;
    else return false;
}

```


4. Structura Complex

Prezentăm în continuare codul sursă al implementării numerelor complexe în C#.

```
public struct Complex
{
    private double x;
    private double y;

    public Complex(double x, double y)
    {
        this.y = y;
        this.x = x;
    }

    public override string ToString()
    {
        double eps = 1.0e-12;
        if (Math.Abs(y) < eps)           // z=real
            return string.Format("{0}", x);
        if (Math.Abs(x) < eps)           // z=imaginar
        {
            if (Math.Abs(y - 1) < eps)    //z=i
                return "i";
            if (Math.Abs(y + 1) < eps)    //z=-i
                return "-i";
            return string.Format("{0}i", y);
        }
        if (y > 0) return string.Format("{0}+{1}i", x, y);
        return string.Format("{0}{1}i", x, y);
    }

    public void show()
    {
        Console.WriteLine(this);
    }

    public double Ro
    {
        get
        {
            return Math.Sqrt(x * x + y * y);
        }
    }
}
```

```

public double Ro2
{
    get
    {
        return x * x + y * y;
    }
}
public double Theta
{
    get
    {
        return Math.Atan2(y, x);
    }
}
public double Re
{
    get
    {
        return x;
    }
    set
    {
        x = value;
    }
}
public double Im
{
    get
    {
        return y;
    }
    set
    {
        y = value;
    }
}

public Complex Conj
{
    get
    {
        return new Complex(x, -y);
    }
}

```

```

public static Complex setRoTheta(double Ro, double theta)
{
    return new Complex(Ro * Math.Cos(theta),
        Ro * Math.Sin(theta));
}
public static Complex setReIm(double x, double y)
{
    return new Complex(x, y);
}

public static Complex operator +(Complex zst, Complex zdr)
{
    return new Complex(zst.x + zdr.x, zst.y + zdr.y);
}
public static Complex operator +(Complex zst)
{
    return new Complex(zst.x, zst.y);
}
public static Complex operator -(Complex zst, Complex zdr)
{
    return new Complex(zst.x - zdr.x, zst.y - zdr.y);
}
public static Complex operator -(Complex zst)
{
    return new Complex(-zst.x, -zst.y);
}
public static Complex operator *(Complex zst, Complex zdr)
{
    return new Complex(zst.x * zdr.x - zst.y * zdr.y,
        zst.y * zdr.x + zst.x * zdr.y);
}
public static Complex operator /(Complex zst, Complex zdr)
{
    double r = zdr.Ro2;
    return new Complex((zst.x * zdr.x + zst.y * zdr.y) / r,
        (zst.y * zdr.x - zst.x * zdr.y) / r);
}
public static implicit operator Complex(double x)
{
    return new Complex(x, 0);
}
public static bool operator ==(Complex zst, Complex zdr)
{
    return (zst - zdr).Ro2 < 1.0e-16;
}

```

```
public static bool operator !=(Complex zst, Complex zdr)
{
    return (zst - zdr).Ro2 >= 1.0e-16;
}
public override bool Equals(object o)
{
    if (o.GetType() != this.GetType()) return false;
    else return this == (Complex)o;
}
public override int GetHashCode()
{
    return x.GetHashCode()+y.GetHashCode();
}
}
```