# BAB 1 Pendahuluan

## 1.1 Latar Belakang

Poin Penting::

* Sistem persamaan Nonlinear dan relevansinya dalam kehidupan
* Metode tradisional pemecahan sistem persamaan (bahas dikit soal kelebihan) dan kekurangannya yang notable (yang bisa diselesaikan dengan metode optimasi)
* Metode optimasi dan mengapa masih kurang untuk mendapat seluruh akar persamaan optimasi
* Metode heuristik dan metaheuristik
* Evolutionary Algorithm dan mengapa bisa dipakai untuk menyelesaikan kasus sistem persamaan nonlinear
* Penjelasan umum HGA dan mengapa bisa dipakai untuk menyelesaikan kasus sistem persamaan nonlinear
* Penjelasan umum RADE dan mengapa bisa dipakai untuk menyelesaikan kasus sistem persamaan nonlinear
* Penjelasan Umum Spiral Dynamic Clustering

## 1.2 Rumusan Masalah

## 1.3 Tujuan

## 1.4 Metodologi

## 1.5 Sistematika Penulisan

# BAB 2 Metode Pencarian Akar Sistem Persamaan Nonlinear

## 2.1 Sistem Persamaan Nonlinear

* Bentuk Umum
* Metode Tradisional Penyelesaian Sistem Persamaan Nonlinear

## 2.2 Metode Optimasi

* Bentuk Umum persamaan Optimasi
* Fungsi Konveks
* Masalah Optimasi Nonlinear & Fungsi objektif SPNL

## 2.3 Metode Heuristik dan Metaheuristik

* Definisi, latar belakang
* Sekilas tentang contoh-contoh

## 2.4 Algoritma Evolusioner

* Background, prinsip
* Alur umum algoritma

## 2.5 Genetic Algorithm

* Algoritma umum
* Teknik selection
* Teknik crossover
* Teknik mutation

## 2.6 Differential Evolution

* Algoritma umum
* Teknik Mutasi
* Teknik Crossover
* Teknik seleksi atau evaluasi

## 2.7 Teknik Clustering

* Penelitian terdahulu oleh Sidarto
* Algoritma umum
* Highlight mana bagian yang dipake dan modifikasi

## 2.8 Repulsion Based Adaptive DE

* Penelitian terdahulu oleh Gong et al.
* Algoritma umum
* Penjelasan karakteristik algoritma
* Highlight mana bagian yang dipake dan modifikasi

## 2.9 Hypercube GA

* Penelitian terdahulu oleh Mastorakis
* Algoritma umum
* Penjelasan karakteristik algoritma
* Highlight mana bagian yang dipake dan modifikasi

## 2.10 Barisan Sobol

* Karakteristik
* Mengapa digunakan

# BAB 3 Komparasi Metode Pencarian Akar

## 3.1 Skema Komparasi

* Metode Algoritma yang dipakai
* Metode Pencarian akar yang dipakai
* Fungsi Objektif yang digunakan
* Spek

| Processor 12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz  Installed RAM 16.0 GB (15.7 GB usable)  Device ID 6C891826-42E1-42A9-8774-F745E58768AB  Product ID 00328-90000-00000-AAOEM  System type 64-bit operating system, x64-based processor  Pen and touch No pen or touch input is available for this display |
| --- |

## 3.2 Clustering Differential Evolution

* Algoritma
* Analisis Kompleksitas
* Uji coba pada fungsi objektif
  + SPNL 1

| **m\_cluster** | **max\_gen** |
| --- | --- |
| 50 | 50 |
| 100 | 100 |
| 250 | 250 |

| dim = 2;  mutation\_factor=0.864519198;  crossover\_rate=0.860650007;  tau\_d=0.4;  gamma = -0.2;  epsilon = 1e-6;  delta = 0.01;  k\_cluster = 10;  m = 250;  boundaries = repmat([-10, 10], dim, 1);  verbose = false;  print\_stat = false; |
| --- |

* + SPNL 2

| **m\_cluster** | **max\_gen** |
| --- | --- |
| 150 | 150 |
| 300 | 300 |
| 450 | 450 |

| dim = 3;  mutation\_factor=0.871070813;  crossover\_rate=0.853720936;  tau\_d=0.1;  gamma = -0.3;  epsilon = 1e-6;  delta = 0.01;  k\_cluster = 10;  m = 300;  verbose = false;  print\_stat = false; |
| --- |

* + SPNL 7

| **mcluster** | **max\_gen** |
| --- | --- |
| 100 | 100 |
| 250 | 250 |
| 400 | 400 |

| dim = 3;  mutation\_factor=0.831987414;  crossover\_rate=0.887292888;  tau\_d=0.1;  gamma = -0.4;  epsilon = 1e-6;  delta = 0.01;  k\_cluster = 10;  m = 300;  seed = 'shuffle';  verbose = false;  print\_stat = false; |
| --- |

## 3.3 Repulsion-based Adaptive Differential Evolution

* Algoritma
* Analisis Kompleksitas
* Uji coba pada fungsi objektif untuk semua sistem persamaan nonlinear:
  + SPNL 1

| **pop\_size** | **max\_gen** |
| --- | --- |
| 50 | 50 |
| 100 | 100 |
| 250 | 250 |

| F\_init=0.5;  CR\_init=0.5;  num\_l=20;  theta=1e-6;  tau\_d=0.4;  s\_max=20;  print\_gen=false;  Hm = 50;  dim = 2;  beta = 1;  rho = 0.01; |
| --- |

* + SPNL 2

| **pop\_size** | **max\_gen** |
| --- | --- |
| 100 | 100 |
| 500 | 200 |
| 2000 | 300 |

| F\_init=0.5;  CR\_init=0.5;  num\_l=20;  theta=1e-6;  tau\_d=0.4;  s\_max=20;  print\_gen=false;  Hm = 50;  dim = 2;  beta = 1;  rho = 0.01; |
| --- |

* + SPNL 7

| **pop\_size** | **max\_gen** |
| --- | --- |
| 100 | 100 |
| 250 | 250 |
| 400 | 400 |

| F\_init=0.5;  CR\_init=0.5;  num\_l=10;  theta=1e-6;  tau\_d=0.5;  s\_max=20;  print\_gen=true;  Hm = 50;  dim = 3;  seed = 'shuffle';  beta = 1;  rho = tau\_d;  boundaries = [0,2;-10,10;-1,1]; |
| --- |

## 3.4 Hypercube Differential Evolution

* Algoritma
* Analisis Kompleksitas
* Uji coba pada fungsi objektif
  + SPNL 1

| **parts** | **max\_gen** |
| --- | --- |
| 100 | 100 |
| 250 | 250 |
| 400 | 400 |

| squared\_parts = [50,100,250];  parts = round(sqrt(squared\_parts));  max\_gen = [50,100,250];  mutation\_factor=0.864519198;  crossover\_rate=0.860650007;  epsilon = 1e-6;  delta = 0.01;  pop\_size = 250;  dim = 2;  seed = 'shuffle';  print\_stat = false;  verbose = false;  visual\_properties = struct('show\_visual',false, ...  'save\_visual', false, ...  'file\_name', 'hde.avi');  boundaries = repmat([-10, 10], dim, 1); |
| --- |

* + SPNL 2

| **Parts (squared)** | **max\_gen** |
| --- | --- |
| 150 | 150 |
| 300 | 300 |
| 450 | 450 |

| epsilon = 1e-6;  delta = 0.01;  dim = 2;  mutation\_factor=0.871070813;  crossover\_rate=0.853720936;  seed = 'shuffle';  print\_stat = false;  verbose = false;  visual\_properties = struct('show\_visual',false, ...  'save\_visual', false, ...  'file\_name', 'hde.avi'); |
| --- |

* + SPNL 7

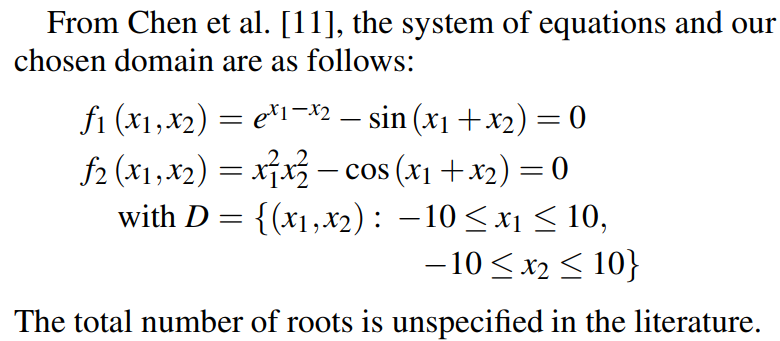
| epsilon = 1e-6;  delta = 0.01;  pop\_size = 300;  dim = 3;  mutation\_factor=0.831987414;  crossover\_rate=0.887292888;  seed = 'shuffle';  print\_stat = false;  verbose = false;  visual\_properties = struct('show\_visual',false, ...  'save\_visual', false, ...  'file\_name', 'hde.avi'); |
| --- |

# BAB 4 Kesimpulan dan Saran

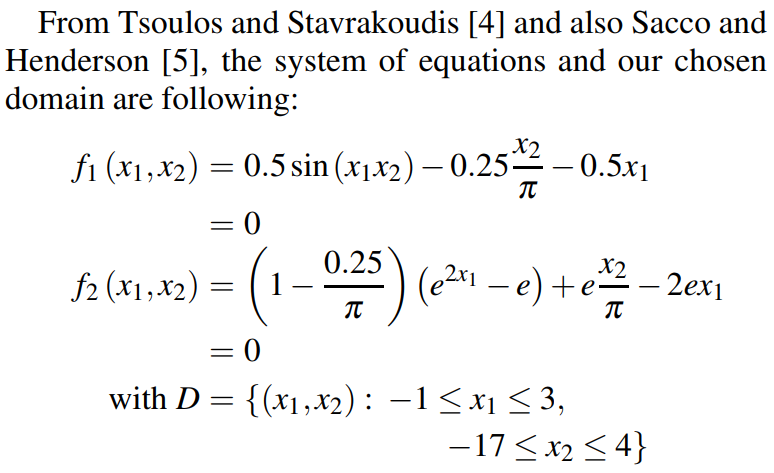
# Daftar Pustaka

***Nb***

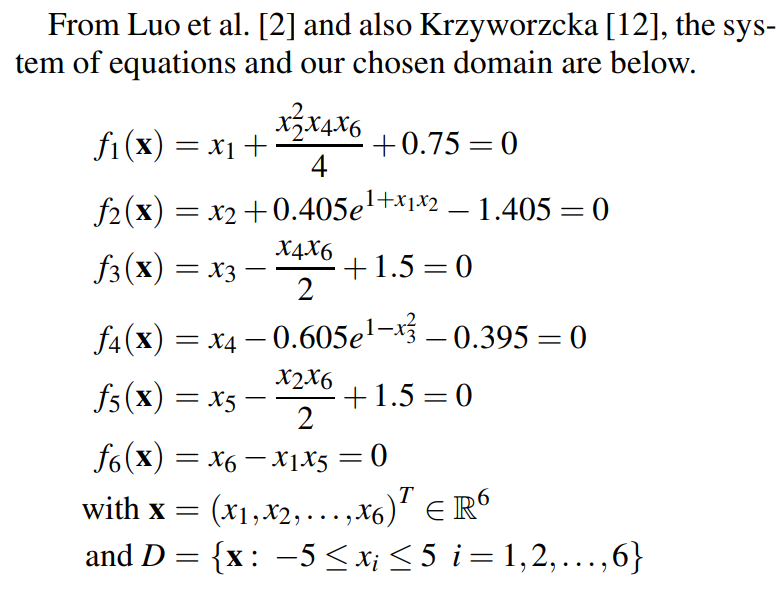
* **SPNL 1**



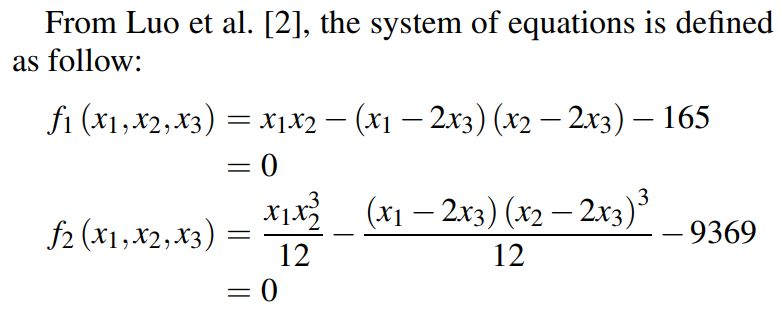
* **SPNL 2**

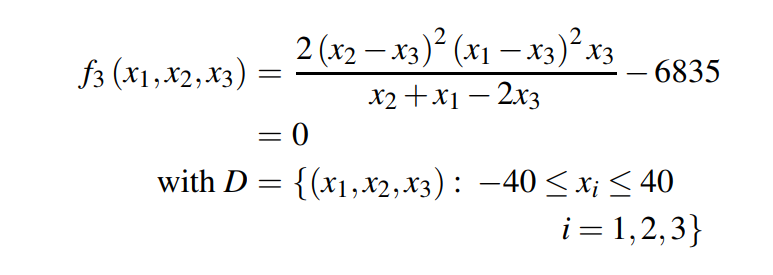


* **SPNL 3**

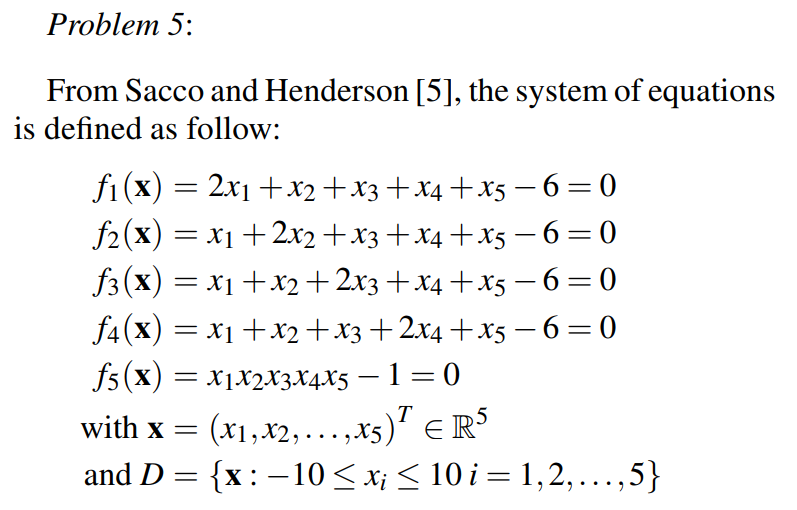
****

* **SPNL 4**

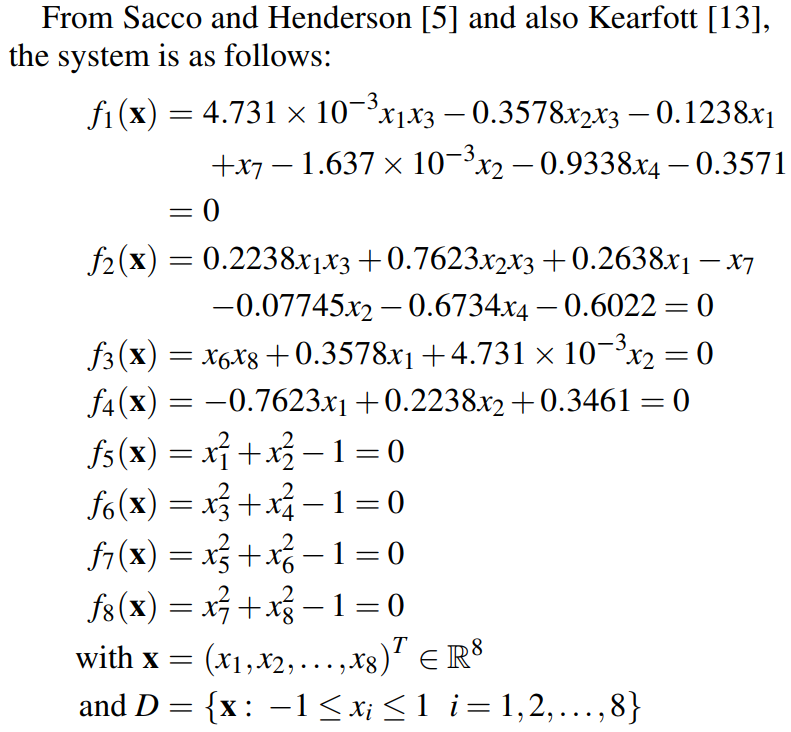
****

****

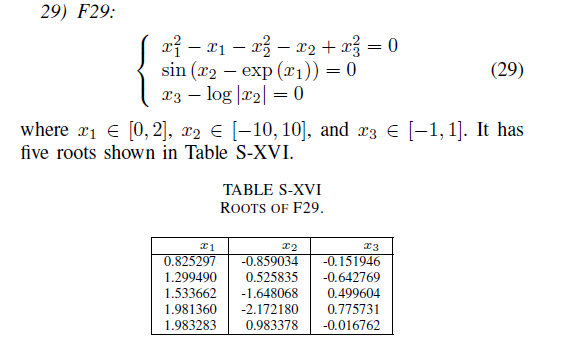
* **SPNL 5**

****

* **SPNL 6**

****

* **SPNL 7**

****

### Hasil Perbandingan

| Indikator | SDDE | HDE | RADE |
| --- | --- | --- | --- |
| Deterministik | T | T | F |
| Update akar? | F | F | T |
|  |  |  |  |
|  |  |  |  |