

Orientation Homework Report

10-bar truss optimization

徐若瑄

August 24, 2021

1 問題描述

十桿桁架 (10-bar truss) 由十個長桿件組成並且有 6 個端點，其中第 5、6 號端點為固定端，桿件的配置如圖 1 所示。

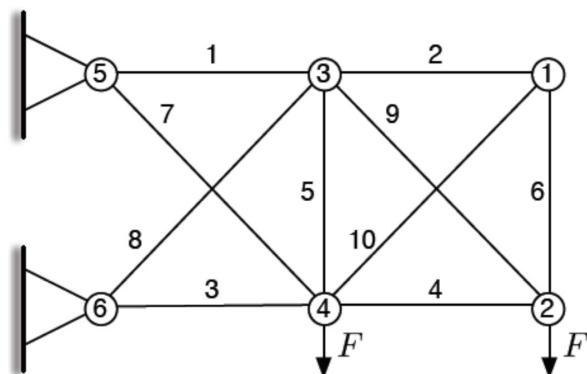


Figure 1: 10-bar truss

所有桿件的截面皆為圓形，桿件 1 到桿件 6 的截面半徑同為 r_1 且長度為 9.14 m ；桿件 7 到桿件 8 的截面半徑同為 r_2 。桿件所使用的材料為剛，相關的材料性質有：

1. 密度 $\rho = 7860\text{ kg/m}^3$
2. 楊氏模數 (Young's Modulus) $E = 200\text{ GPa}$
3. 降伏係數 (Yield stress) $\sigma_y = 250\text{ MPa}$

端點 4 及端點 5 有一向下的外力 $F = 1.0 \times 10^{-7}\text{ N}$ ，在所有桿件的應力不超過降伏應力且端點 2 的位移小於 0.02 m 的條件下，求桿件截面半徑 r_1 與 r_2 的值，使得十桿桁架整體重量為最輕。

2 計算方法

計算十桿衍架最小重量主要使用兩個方法，首先利用有限元素法 (Finite element method) 計算出所有桿件的應力值、各個端點的位移量，接著進行最佳化即可得到最佳值與最佳解。

以下所有計算過程皆使用 Matlab 程式輔助計算。

2.1 有限元素分析

有限元素分析的關鍵步驟如下：

1. 建立元素表 (Element Table)：元素表內包含各桿件的長度、截面積、角度等
2. 建立剛性矩陣 (Stiffness Matrix)：利用元素表計算出剛性矩陣 K ，表示出 6 個端點的 12 個自由度上下位移與受力之間的關係
3. 計算各個節點的位移：透過 $F = KQ$ 的關係式求出各點在 x,y 兩方向的位移量
4. 計算各桿件的應力：透過 $\sigma = E\varepsilon$ 的關係式求出各桿件的應力
5. 計算反作用力：同樣使用 $F = KQ$ 的關係式求出固定端點 5、6 的反作用力
6. 最佳化：將問題條件轉換為數學式，最佳化後得到最佳值與最佳解

2.1.1 建立端點與桿件參數

為了快速建立元素表，首先將桿件長度轉換為各端點的座標位置 (以端點 6 為原點) 並存於 `nodeTable`

```
%% Create Node Table
nodeTable = [length*2 length;
             length*2 0;
             length*1 length;
             length*1 0;
             0 length;
             0 0];
```

接著設定桿件 1 到桿件 10 對應到的左右兩截點編號。利用 `nodeInfo` 紀錄各端點連接的桿件編號，再將桿件兩端點的編號存於 `elementToNode`

```
%% Create element to node array
nodeInfo = {[2 6 10],[4 6 9],[1 2 5 8 9],[3 4 5 7 10],[1 7],[3 8]};
elementToNode = zeros(10,2);
index = ones(10,1);

for i = 1:6
    for j = 1:size(nodeInfo{i},2)
        element = nodeInfo{i}(j);
        elementToNode(element,index(element)) = i;
        index(element) = index(element) + 1;
    end
end
```

2.1.2 建立元素表

元素表 `elementTable` 為 10x4 的矩陣，每一欄依序紀錄各桿件的長度 L 、截面積 A 、 \cos 及 \sin 。以 `elementToNode` 搭配 `nodeTable` 計算各項數值

```
%% Create the elementTable using elementToNode
%% (10,4) = [A L cos sin]
elementTable = zeros(10,4);

for i = 1:10
```

```

% Node of Element
nodei_x = nodeTable(elementToNode(i,1),1);
nodei_y = nodeTable(elementToNode(i,1),2);
nodej_x = nodeTable(elementToNode(i,2),1);
nodej_y = nodeTable(elementToNode(i,2),2);

% Area
if i < 7
    elementTable(i,1) = pi * r1.^2;
else
    elementTable(i,1) = pi * r2.^2;
end

% Length of element
elementTable(i,2) = sqrt(power(nodej_x - nodei_x
,2)+power(nodej_y - nodei_y ,2));

% cos
elementTable(i,3) = (nodej_x - nodei_x)/elementTable(i,2);

% sin
elementTable(i,4) = (nodej_y - nodei_y)/elementTable(i,2);
end

```

3 結論