Laboratorio: El monorraíl

**Objetivos**

En esta actividad vas a conseguir poner en práctica aquellas cuestiones relacionadas con las diferentes técnicas de interpolación que hemos trabajado.

**Descripción**

En la ciudad de Springfield se va a instalar un monorraíl. Tras una extensa votación entre los ciudadanos, se ha decidido que debe tener paradas en los lugares marcados en la Figura 1, cuya ubicación se refleja en la Tabla 1.

Diagrama

Descripción generada automáticamente

Figura 1. Perfil de Springfield y lugares en los que tendrá parada el monorraíl. Fuente: elaboración propia con imágenes de <https://simpsons.fandom.com>

|  |  |
| --- | --- |
| **Posiciones** | |
|  |  |
| **Central nuclear** | 0 | 20 |
| **Badulaque** | 1.5 | 100 |
| **Estudios Krustylu** | 3 | 60 |
| **Estatua Jebediah Springfield** | 4 | 120 |
| **Mazmorra del Androide** | 6 | 20 |
| **Taverna de Moe** | 7 | 40 |
| **742 Evergreen Terrace** | 9 | 100 |
| **Laboratorio de Frink** | 10 | 0 |

Tabla 1. Ubicaciones de las paradas del monorraíl.

Resuelve los siguientes problemas.

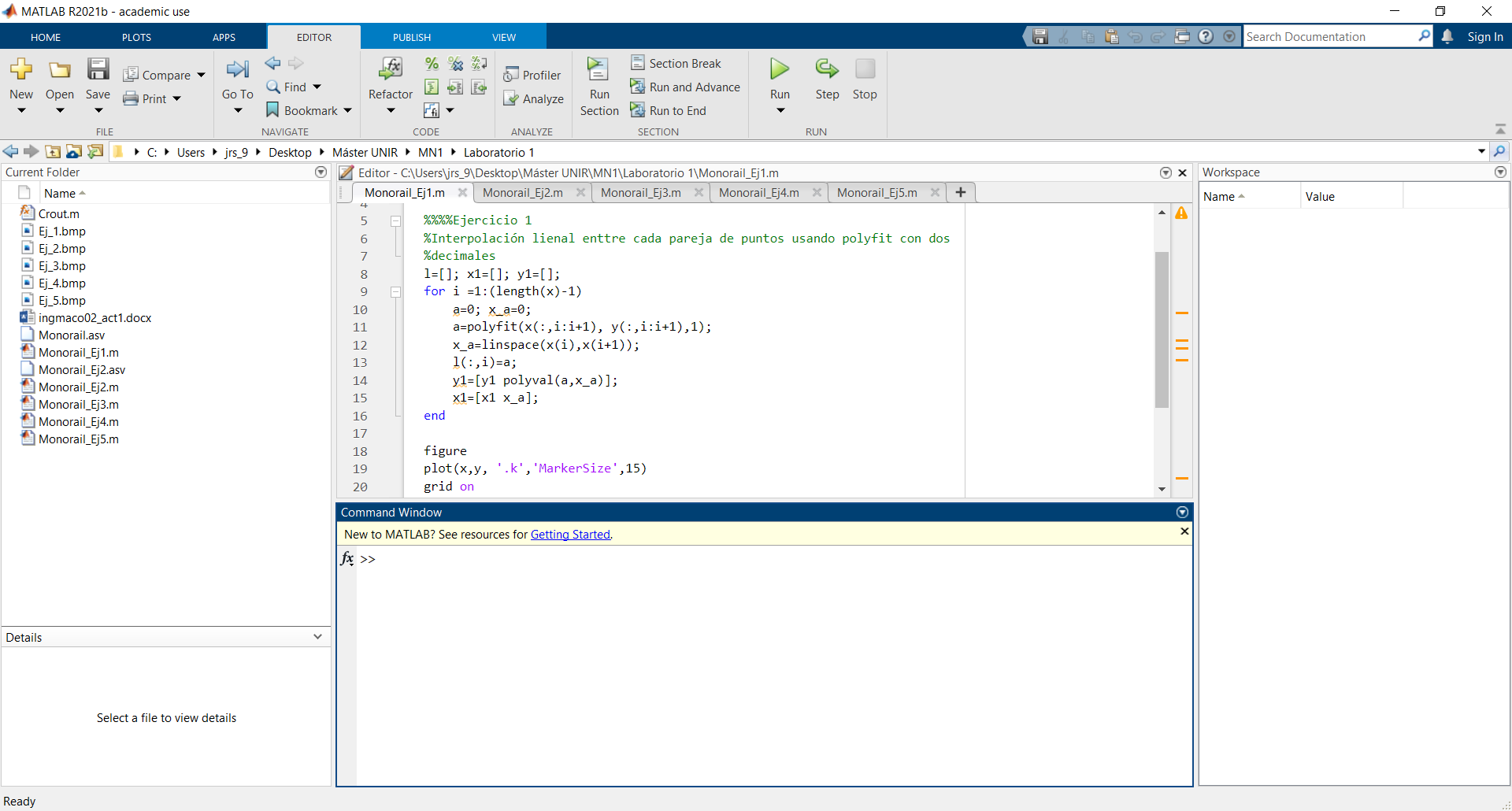
Problema 1

Como primera aproximación, vamos a hacer una interpolación lineal entre cada pareja de puntos contiguos. Utiliza el comando de Matlab polyfit utilizando dos decimales. Obtén la expresión de , es decir:

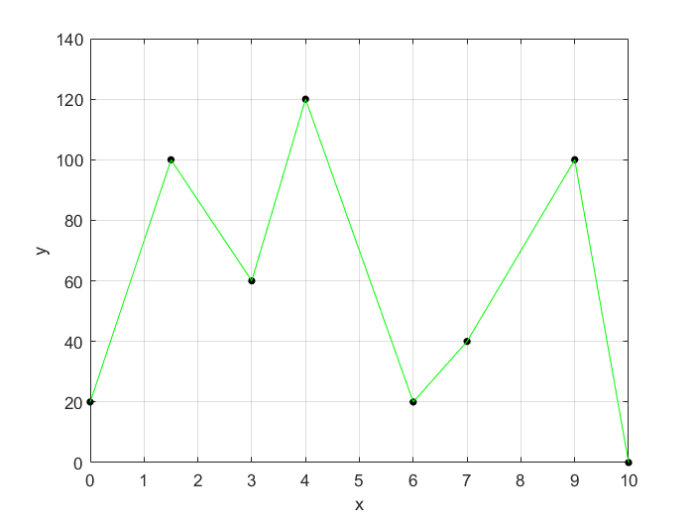
Representa el polinomio a trozos con líneas; utiliza puntos para las ubicaciones de las paradas.

Solución:

Para desarrollar este ejercicio, lo primero que hacemos es declarar dos vectores fila *x* e *y* con los valores del enunciado. A partir de esto, vamos recorriendo con un bucle los pares de valores *i* e *i+1*, y utilizando la función *polyfit* obtenemos los parámetros de ajuste de la recta entre esos puntos. Luego defino un vector genérico entre las dos posiciones correspondientes, y guardo los valores de esa función a trozos en un vector *x1* e *y1* inicializados vacíos, que serán los que pintaré como ajuste. Además, durante el bucle, voy almacenando los parámetros del ajuste de *polyfit* por columnas en una matriz *l* para luego tener acceso a ellos.

Todo esto viene mostrado en el esqueleto principal del código:

A partir de esto, obtengo la solución y la gráfica:



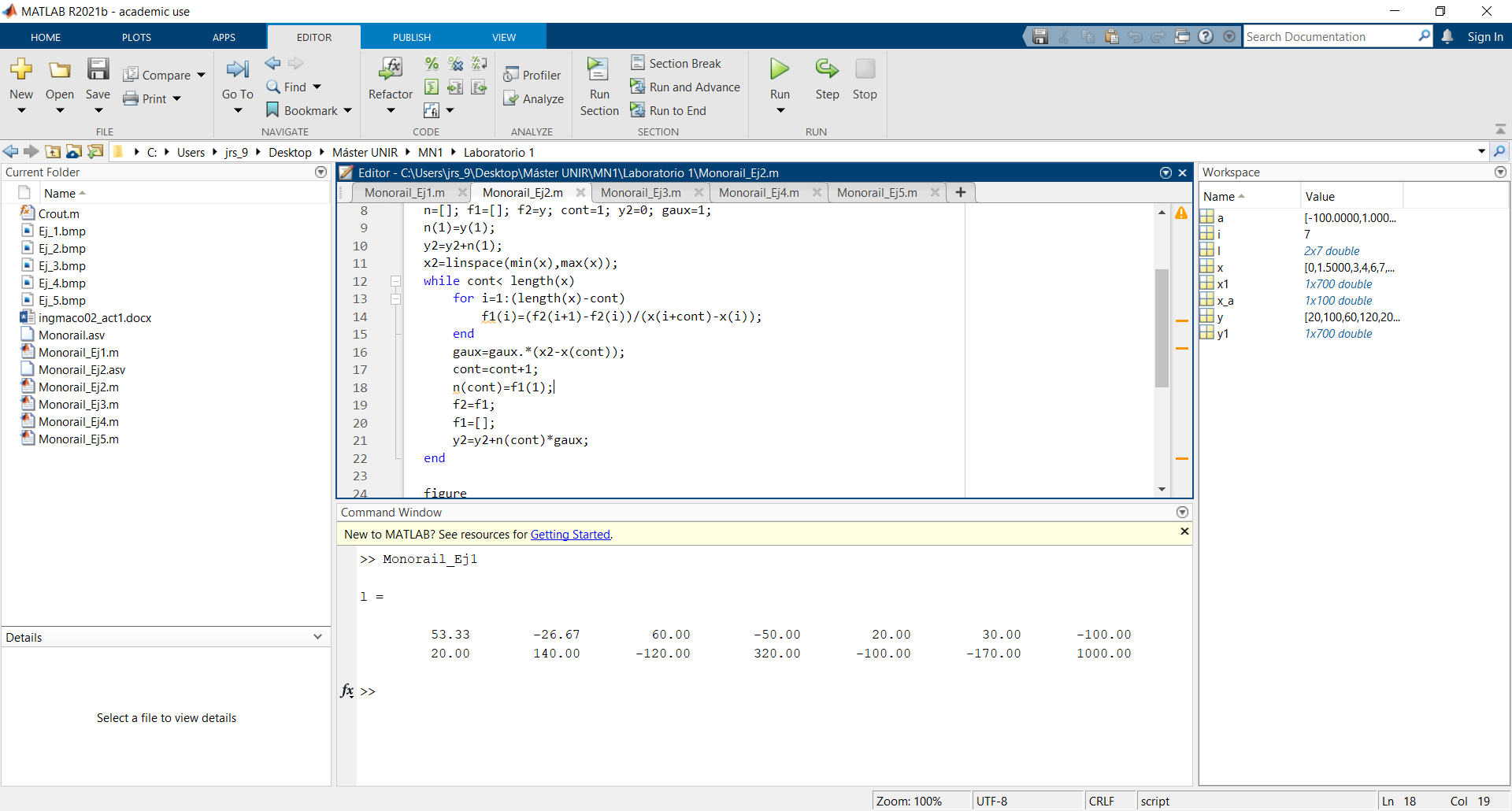
*Fig. 1: Gráfica correspondiente al polinomio a trozos representada con líneas verdes y de las ubicaciones de las paradas marcadas con puntos en negro.*

Problema 2

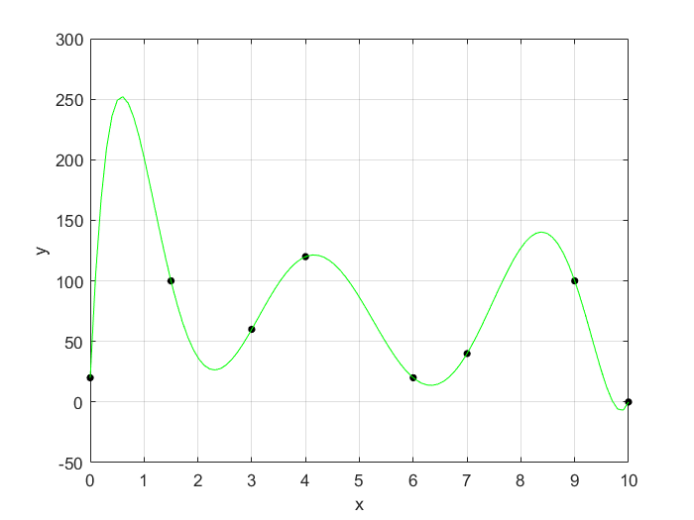
Si utilizamos interpolación lineal entre las paradas, observamos que hay cambios muy bruscos de la pendiente, o lo que es equivalente, la función no es derivable. Obtén el polinomio de interpolación de Newton de grado 7 que pasa por las ocho paradas, utilizando dos decimales. Representa el polinomio; utiliza puntos para las ubicaciones de las paradas.

Solución:

De manera análoga, inicio primero los valores de la función en el primer nodo, y creo un vector *x2* genérico entre el menor y el mayor nodo. A partir de esto, vamos recorriendo con un bucle los distintos nodos y anido con otro bucle para calcular las diferencias divididas del correspondiente orden. De esto, almaceno en una variable auxiliar el correspondiente valor de (*x-xi*), aumento una unidad al contador del bucle, guardo el coeficiente que corresponde del polinomio de Newton y actualizo variables del bucle y el vector *y2* que es el que voy a graficar.

Añado el esqueleto principal del código:

A partir de esto, obtengo la solución y la gráfica:



*Fig. 2: Gráfica correspondiente al polinomio de Newton de grado 7 representado en verde y de las ubicaciones de las paradas marcadas con puntos en negro.*

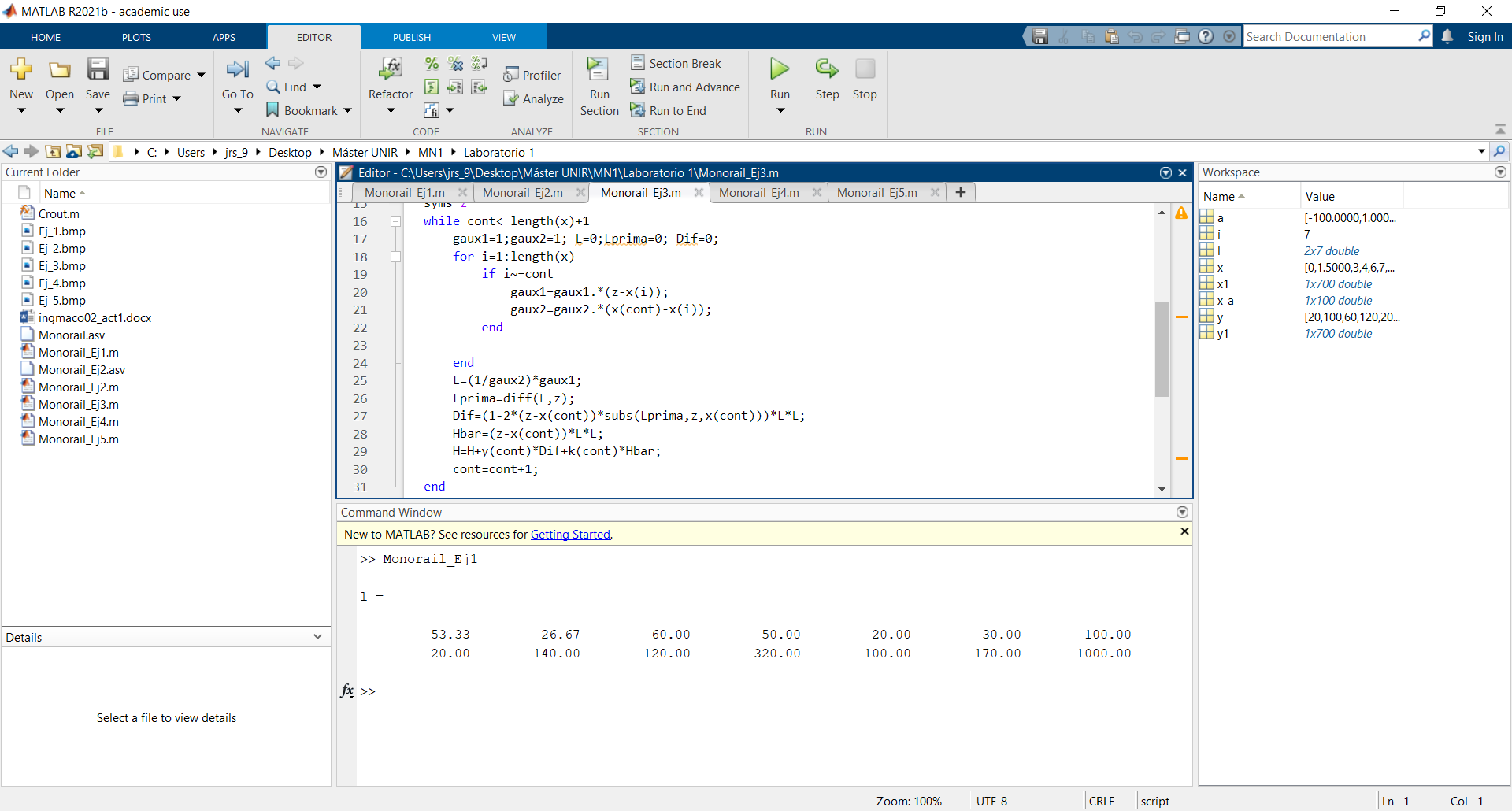
Problema 3

El polinomio de grado 7 de Newton hace que para ir de la Central Nuclear al Badulaque recorramos mucha distancia. Obtén el polinomio de Hermite de grado 15 que pasa por las ocho paradas, tomando como derivada en cada punto el valor 0. Redondea los coeficientes al entero más próximo (acabarás con un polinomio de grado 12). Representa el polinomio; utiliza puntos para las paradas.

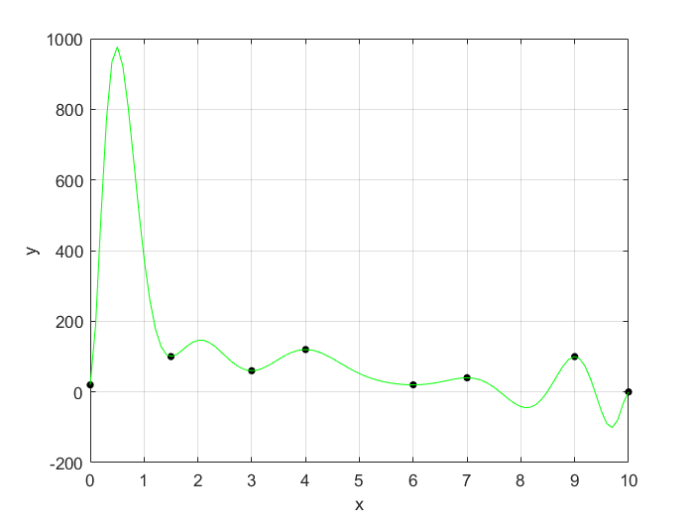
Solución:

En este caso, como quiero hacer un programa genérico, voy a definir además un vector *k* del mismo tamaño que *x* e *y* que represente el valor de la derivada, pero con todo 0s en este caso y defino una variable simbólica *z*, y creo un vector *x3* genérico entre el menor y el mayor nodo. A partir de esto, vamos recorriendo con un bucle los distintos nodos y anido con otro bucle para calcular la correspondiente función de Lagrange *L*. Una vez tengo esto, ya tengo todo lo necesario para calcular la ecuación:

que es la que voy a evaluar con la función *subs* para obtener *y3* que representaré.

 Añado el esqueleto principal del código.

A partir de esto, obtengo la solución y la gráfica:



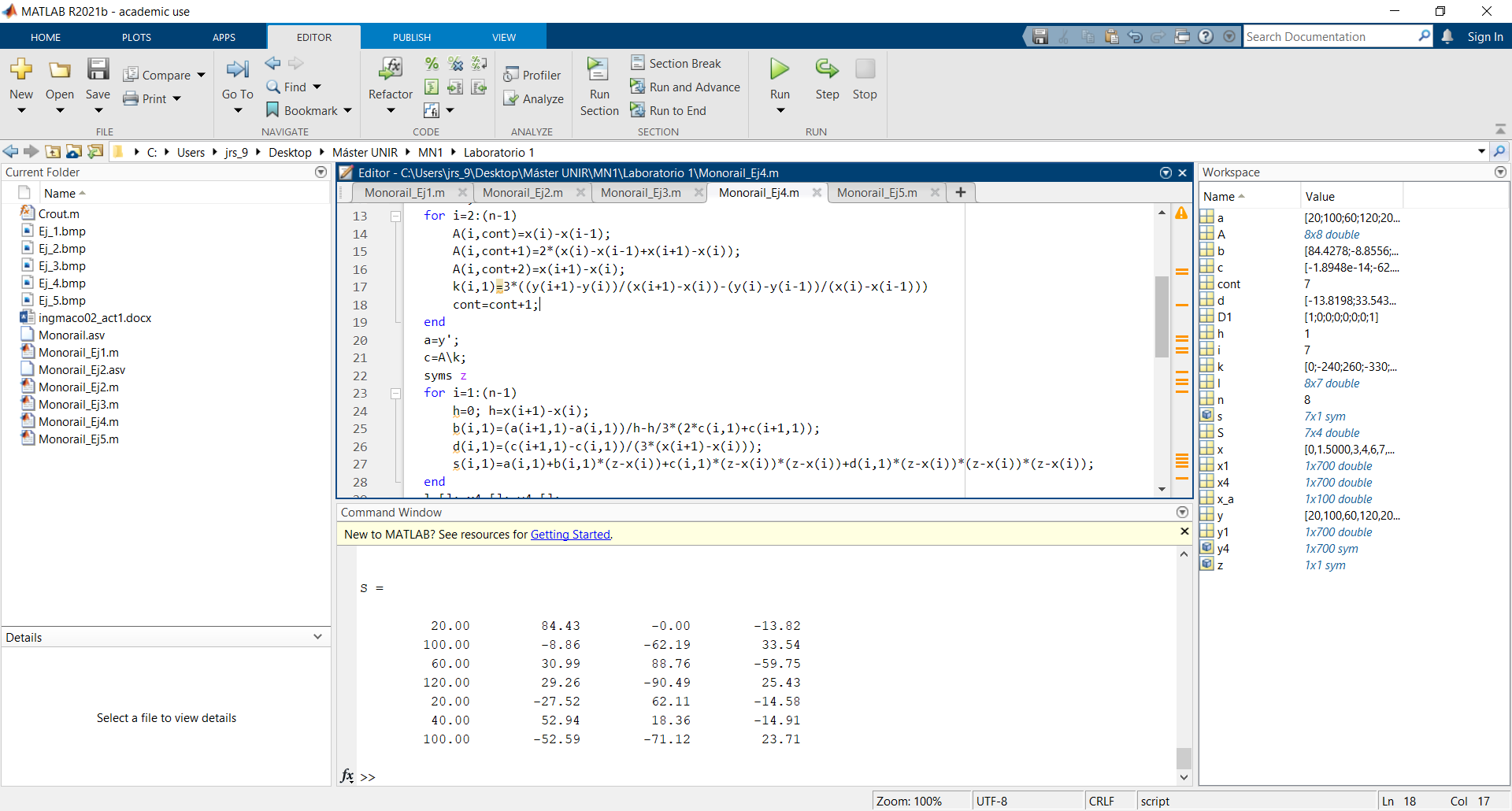
*Fig. 3: Gráfica correspondiente al polinomio de Hermite de grado 15 representado en verde y de las ubicaciones de las paradas marcadas con puntos en negro.*

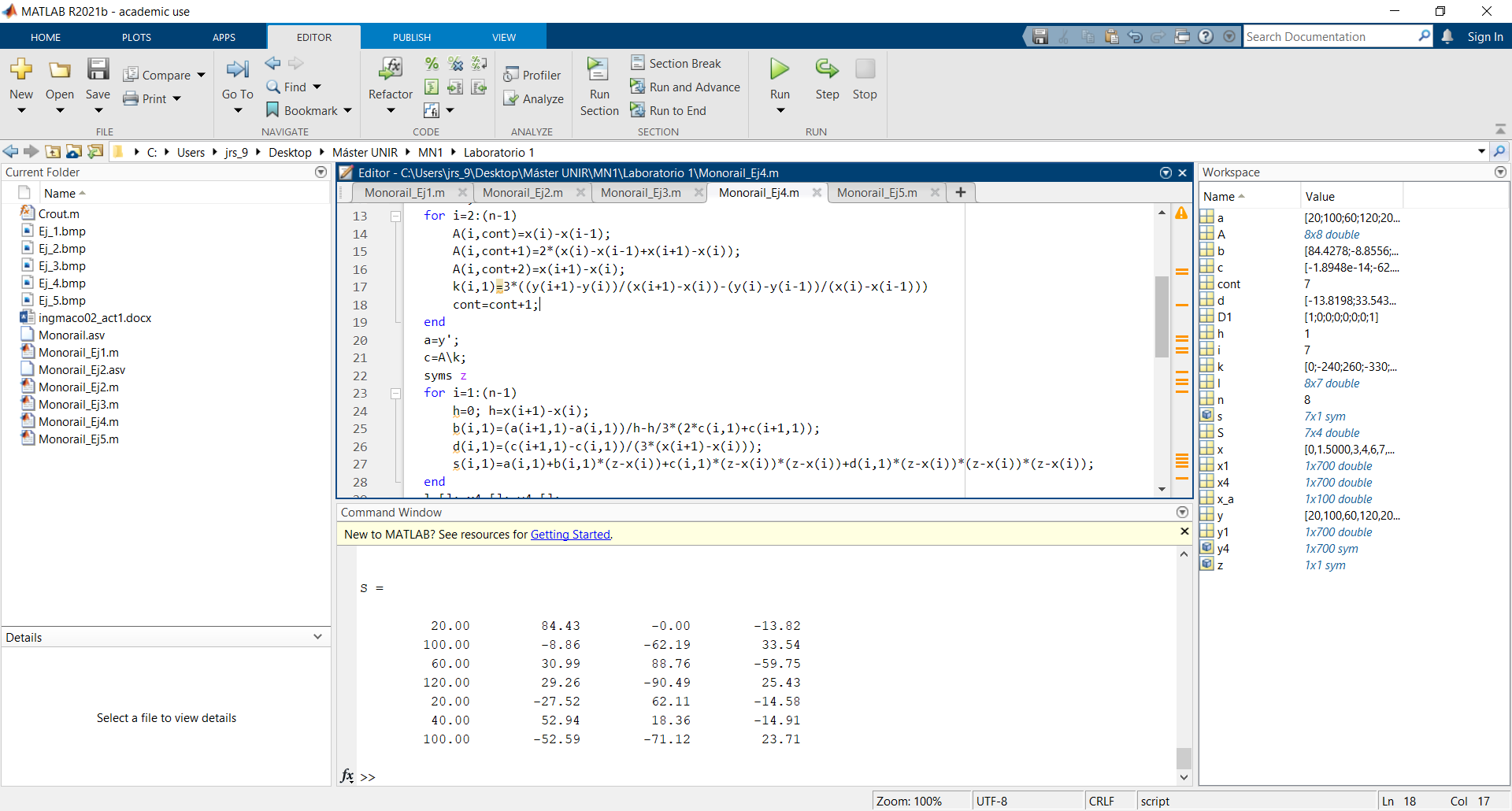
Problema 4

Parece que utilizar el polinomio de Hermite empeora todavía más el trayecto entre la Central Nuclear y el Badulaque. Así que vamos a utilizar splines naturales cúbicos . Obtén los splines naturales cúbicos que pasan por las ocho paradas, es decir, obtén la expresión de , donde:

Utiliza dos decimales. Representa el polinomio; utiliza puntos para las ubicaciones de las paradas.

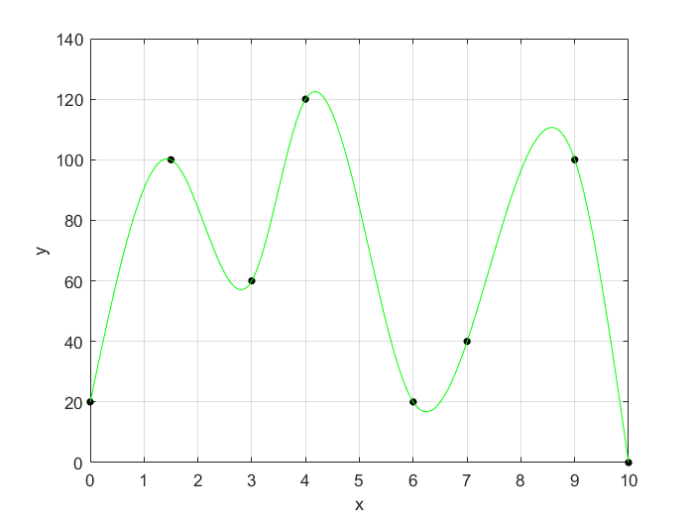
Solución:

En este caso, creo primero un bucle para definir la matriz tridiagonal *A* y el vector correspondiente *k* tal y como viene en los apuntes.

A partir de esto, tomo la columna de soluciones *a* como la transpuesta del vector *y*, calculo el vector *c* como solución del sistema *Ax=k*, y con esto ya puedo calcular con otro bucle los vectores columna *b* y *d*, y el polinomio a trozos que guardo en *s*. Guardo los valores de esa función a trozos en un vector *x4* e *y4* inicializados vacíos, que serán los que pintaré como ajuste. Añado el esqueleto principal del código.

De estos cuatro vectores ya puedo calcular la solución correspondiente por trozos que me dará todos los parámetros para calcular los correspondientes polinomios a trozos con:

A partir de esto, obtengo la solución y la gráfica:

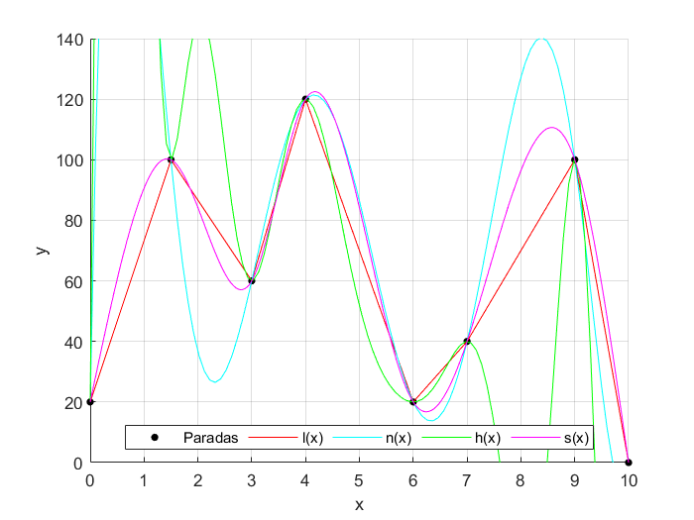


*Fig. 4: Gráfica correspondiente al polinomio de splines naturales cúbicos representada con líneas verdes y de las ubicaciones de las paradas marcadas con puntos en negro.*

Problema 5

Representa en una sola gráfica la superposición de los polinomios de interpolación , , , y los puntos de las paradas, ajustando los ejes a . Comenta los resultados.

Solución:



*Fig. 5: Gráfica correspondiente a las ubicaciones de las paradas marcadas con puntos en negro, al polinomio a trozos representada con líneas rojas, al polinomio de Newton de grado 7 representado en cian, al polinomio de Hermite de grado 15 representado en verde y al polinomio de splines naturales cúbicos representada en magenta.*

Como ya hemos ido viendo, definir un polinomio a trozos lineal nos permitiría ir de un punto a otro lo más directo posible, pero el inconveniente es que nos crea cambios muy bruscos en las paradas, cosa que no sería muy viable para recorrer con un monorraíl.

De manera análoga, hemos obtenido el polinomio de Newton de grado 7 y el de Hermite de grado 15 (aunque al final nos ha dado uno de grado 12). Matemáticamente podríamos decir que esperábamos que estos dos polinomios fuesen buenos ya que hemos utilizado polinomios de grado “elevado” con muchas raíces, y, hemos visto que en las paradas el cambio de pendiente es mucho más suave que el caso anterior. Sin embargo, estos polinomios nos han metido un gran viaje entre la Central Nuclear y el Badulaque, cosa que tampoco sería muy útil para el trayecto, ya que nos incorporaría mucho tiempo de viaje.

Por último, hemos obtenido el polinomio de splines cúbicos naturales, que pese a ser un polinomio de grado 3, nos ha permitido obtener una curva suave, sin cambios bruscos y que no mete mucho viaje entre paradas. Todo esto se ha conseguido gracias a que hemos añadido la definición de que los valores del polinomio a trozos deben ser los mismos en los nodos, que las derivadas en los nodos deben ser la misma, y que la segunda derivada debe ser también igual en los nodos, imponiendo además que sea 0 por ser splines naturales.

Podemos concluir así que no necesariamente un polinomio de grado elevado nos va a dar siempre la mejor aproximación pese a tener más raíces, ya que, en nuestro caso, la mejor aproximación ha sido obtenida a partir de un polinomio de grado 3 que es más eficiente en el viaje por el hecho de haber impuesto condiciones a la primera y segunda derivada en los nodos.

**Rúbrica**

|  |  |  |  |
| --- | --- | --- | --- |
| El monorraíl | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | Calidad en la presentación. | 1 | 10% |
| Criterio 2 | Problema 1. Desarrollo. | 0.3 | 3% |
| Criterio 3 | Problema 1. Resultado. | 0.5 | 5% |
| Criterio 4 | Problema 1. Representación. | 0.2 | 2% |
| Criterio 5 | Problema 2. Desarrollo. | 0.6 | 6% |
| Criterio 6 | Problema 2. Resultado. | 1 | 10% |
| Criterio 7 | Problema 2. Representación. | 0.4 | 4% |
| Criterio 8 | Problema 3. Desarrollo. | 0.6 | 6% |
| Criterio 9 | Problema 3. Resultado. | 1 | 10% |
| Criterio 10 | Problema 3. Representación. | 0.4 | 4% |
| Criterio 11 | Problema 4. Desarrollo. | 0.9 | 9% |
| Criterio 12 | Problema 4. Resultado. | 1.5 | 15% |
| Criterio 13 | Problema 4. Representación. | 0.6 | 6% |
| Criterio 14 | Problema 5. Representación. | 0.5 | 5% |
| Criterio 15 | Problema 5. Reflexión. | 0.5 | 5% |
|  |  | **10** | **100 %** |

**Extensión máxima de la actividad**: 10 páginas, fuente Calibri 12 e interlineado 1,5.