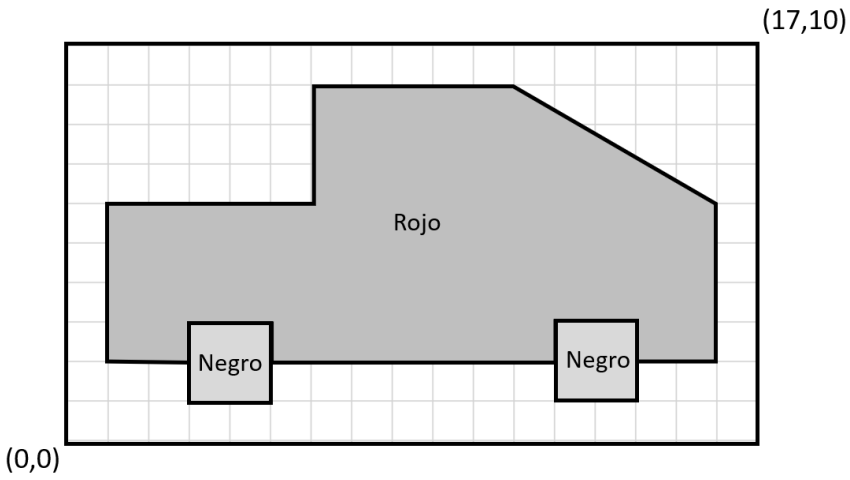
Control 1

Instrucciones Generales

- Responda cada problema en hojas separadas
- Escriba claramente su nombre y firma en cada hoja
- Responda tan breve como sea posible
- La evaluación termina a las 14 hrs., no habrá tiempo adicional
- No hay preguntas, utilice su mejor juicio para responder
- El control posee 6 puntos en total

Problema 1

1. (0.7 puntos) Queremos dibujar el siguiente modelo utilizando OpenGL. Defina una única lista ordenada de vértices y otra de índices que permitan cumplir este objetivo. Cada vértice debe especificar posición y color (en RGB). Cada 3 índices especifica un triángulo. *Observaciones:*
   - Asuma que su ventana de visualización está configurada para trabajar con objetos entre $(0, 0)$ y $(17, 10)$
   - El reticulado y líneas de borde solo cumplen fines ilustrativos, no debe dibujarlos
   - *Hint:* Puede sobreponer triángulos, los últimos triángulos se dibujarán sobre los primeros.



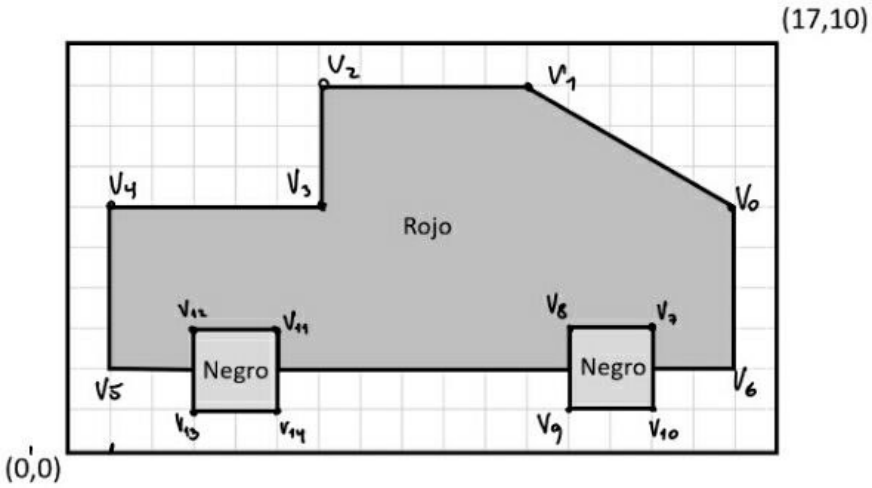
2. (0.6 puntos) Describa como almacenaría el modelo anterior en un formato de imagen vectorial y en un formato de imagen raster. ¿Cuál requeriría menos memoria de almacenamiento?, ¿por qué?.

3. (0.7 puntos) Considere que dispone de la función cuadrado(color, transformación) que dibuja un cuadrado de arista 1 centrado en $(0, 0)$ del color dado luego de aplicar la matriz de transformación especificada. Utilizando solo llamadas a dicha función, modele exactamente la misma figura de la parte 1 de este problema. *Observaciones:*

   - No es necesario efectuar los productos matriciales, basta con especificar las matrices y el orden correcto en el cual se deben operar
   - Ejemplo: cuadrado($Azul$, $R_x(45^o)S(2, 2, 2)$), aplicará primero la matriz S, y luego R_x con los valores dados (consultar formulario en la última página)

Solución Problema 1

Parte 1



Nuestro formato de vértices es el siguiente $Vertice = (x, y, z, r, g, b)$. Utilizaremos los siguientes vértices:

1. Partes del auto de color rojo:

   - $V_0 = (16,\ 6,\ 0,\ 1,\ 0,\ 0)$
   - $V_1 = (11,\ 9,\ 0,\ 1,\ 0,\ 0)$
   - $V_2 = (6,\ 9,\ 0,\ 1,\ 0,\ 0)$
   - $V_3 = (6,\ 6,\ 0,\ 1,\ 0,\ 0)$

- $V_4 = (1, 6, 0, 1, 0, 0)$
- $V_5 = (1, 2, 0, 1, 0, 0)$
- $V_6 = (16, 2, 0, 1, 0, 0)$

2. Ruedas de color negro:

- $V_7 = (14, 3, 0, 0, 0, 0)$
- $V_8 = (12, 3, 0, 0, 0, 0)$
- $V_9 = (12, 1, 0, 0, 0, 0)$
- $V_{10} = (14, 1, 0, 0, 0, 0)$
- $V_{11} = (5, 3, 0, 0, 0, 0)$
- $V_{12} = (3, 3, 0, 0, 0, 0)$
- $V_{13} = (3, 1, 0, 0, 0, 0)$
- $V_{14} = (5, 1, 0, 0, 0, 0)$

De acuerdo a lo anterior, nuestra lista de vértices será la siguiente:

$$[V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}, V_{12}, V_{13}, V_{14}]$$

Ahora uniremos los vértices para formar las figuras, por lo que nuestra lista de indices será:

$$[0, 1, 3 \mid 1, 2, 3 \mid 4, 5, 6 \mid 0, 4, 6 \mid 7, 8, 9 \mid 9, 10, 7 \mid 11, 12, 13 \mid 13, 14, 11]$$

En donde los | son para leer los triángulos formados de una manera más agradable a la vista, pero deberían ser **,** (comas).

Parte 2

Para almacenar el modelo como una imagen vectorial se requeriría al menos almacenar los contornos involucrados. Esto implacaria un contorno de 7 puntos para el chasis y dos de 4 puntos para las ruedas. Cada contorno necesitaría información de color. Luego, como aproximación podemos considerar que se requieren 15x2 = 30 números (15 puntos de 2 coordenadas) más 4x3 = 12 (4 colores RGB, uno para cada contorno y otro para el fondo). En resumen, se requieren a lo menos 42 números.
Para almacenar el modelo como imagen raster necesitamos primero rasterizar a un número de píxeles. Como ejemplo consideremos 17x10 píxeles, donde cada uno de ellos debe tener asociado un color (esquema directo) o tag (esquema indirecto). Para un esquema directo necesitaríamos 170x3 = 510 números, mientras que para un esquema indirecto necesitaríamos al menos 170 + 3x3 = 179 números (i.e. una tabla de sólo 3 colores).
Luego, el formato vectorial utilizaría menor memoria en general, a menos que la resolución de la imagen sea demasiado pequeña, donde se perdería la forma que se intenta representar.

Parte 3

También es posible realizar esta parte usando shearings.

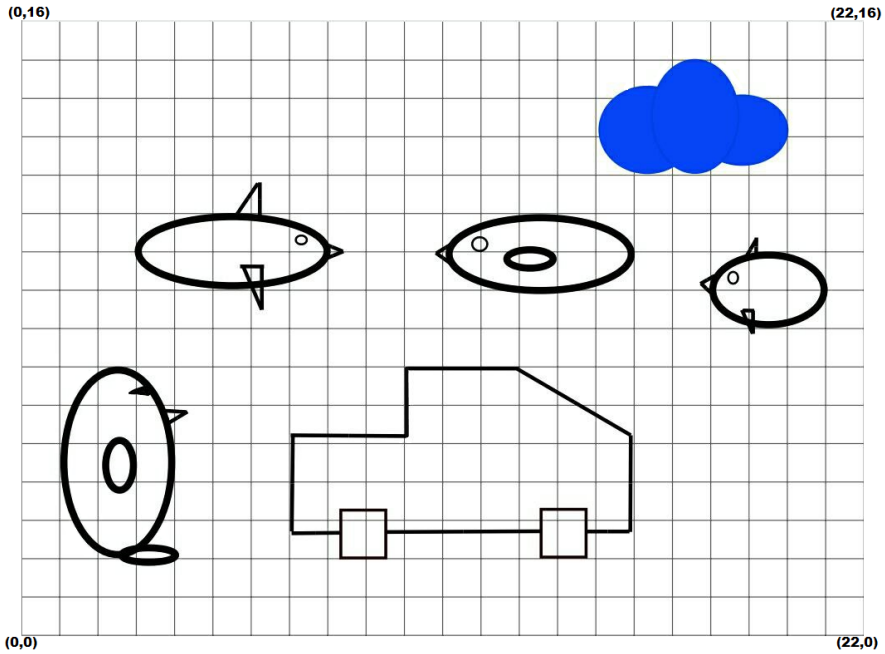
```
cuadrado(rojo, T(8.5, 4 , 0) S(15,4))                    # Parte inferior
cuadrado(rojo, T(10.5,5.5, 0) Rz(atan(5/3)) S(5,3))      # 45 grados igual sirve
cuadrado(rojo, T(8.5, 7.5, 0) S(5,3))                    # Parte superior

cuadrado(negro, T(4, 2, 0) S (2,2))    # Rueda izquierda
cuadrado(negro, T(13, 2, 0) S(2,2)))   # Rueda derecha
```

Existe un descuento mínimo de 0.1 por no considerar el desplazamiento en (0.5, 0.5). No importa si no se consideró Z en la traslación, mientras haya consistencia.

Problema 2

Diseña una aplicación gráfica interactiva que contenga una nube, cuatro pájaros, un paisaje de fondo y el modelo de vehículo de la pregunta 1 tal como se muestra en la siguiente figura.



Las formas disponibles para dibujar los pájaros, nube y vehículo son un cuadrado de lado 1 y un círculo de radio 1, ambos centrados en el origen, y un triángulo equilátero de lado 1, con uno de sus vértices en (0,0). La aplicación permite mover el vehículo hacia la izquierda (<), hacia la derecha (>) y retroceder (r). La nube se mueve continuamente de derecha a izquierda y viceversa. Las ruedas del vehículo rotan constantemente.

1. (1 punto) Dibuje el grafo de la escena, especificando las transformaciones asociadas a los arcos. Especifique una transformación por arco. Indique qué transformaciones pueden ser actualizadas por la interacción del usuario. Minimice el número de hojas del grafo.

2. (1 punto) Model-View-Controller: Identifique qué componentes/acciones de la aplicación corresponden al modelo, qué parte a la vista y qué parte al controlador. Explique qué acción(es) se realiza/ejecuta en el controlador, en el modelo y en la vista cuando el usuario presiona la flecha >.

Solución Problema 2

Grafo de escena

Parte MVC

1. Model: Contiene las figuras. Contiene las transformaciones y cálculos relacionados.

2. Vista: Lo que despliega la aplicación para que se pueda ver. Se actualiza constantemente.

3. Controller: Se encargaría de relacionarse con el usuario, recibiendo los inputs y gatillando estos eventos en el modelo y la vista.

   Cuando el usuario presione la flecha ¿, el controlador gatillará un cambio en el modelo, el cual modificará la rotación de las ruedas si corresponde y trasladará al auto. Como la vista se actualiza constantemente mostrando lo que está en el modelo, permitirá que se vean los cambios en pantalla.

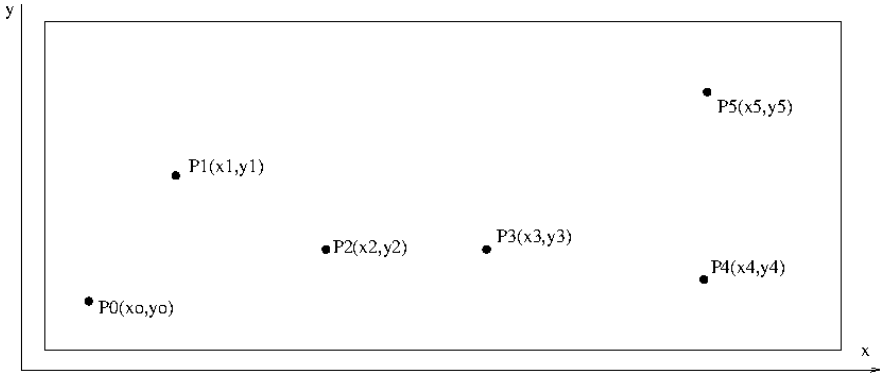
Problema 3: Curvas y lectura "Skeletal Animation"

1. (0.4 puntos) ¿Qué representa el concepto de *keyframe* y para qué sirve?

2. La siguiente figura muestra posiciones claves $P_i(x_i, y_i)$ de un monito que se mueve en un plano (una por cada *keyframe*). Dibuje la trayectoria completa aproximada/interpolada que recorrería el monito usando los métodos especificados a continuación. Explique cómo aplica cada método y formule la expresión matemática para los últimos tres métodos entre dos puntos cualquiera. Hint: ver Q(t) en Formulario adjunto.

   - (0.2 puntos) Interpolación *step*
   - (0.4 puntos) Interpolación lineal
   - (0.5 puntos) Catmull-Rom (spline)

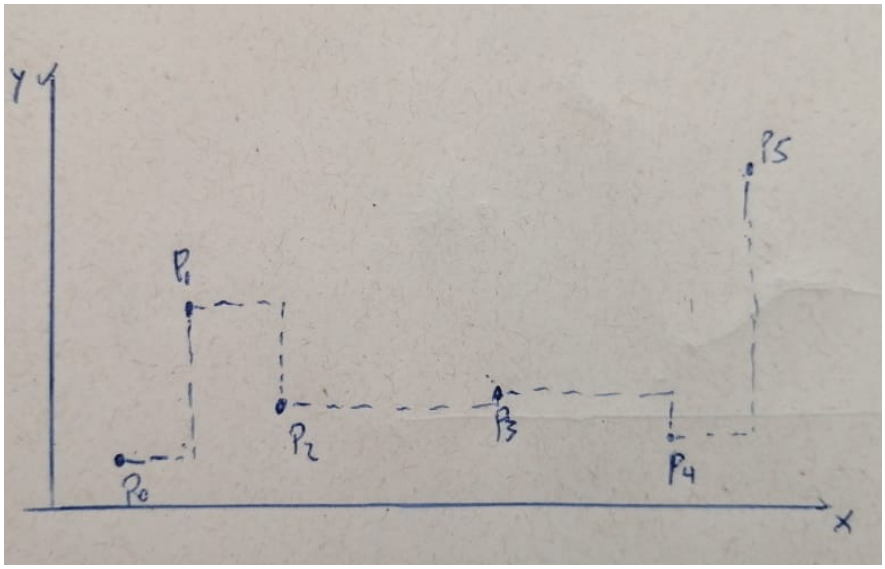
- (0.5 puntos) Bezier (spline)



Solución Problema 3

1. Un keyframe es un conjunto de parámetros que define el movimiento de un objeto, generalmente complejo, en un momento determinado. Luego, para modelar el movimiento se realizan distintas interpolaciones a varios keyframes. [opcional] Para movimientos complejos se especifican una mayor cantidad de keyframes para lograr un movimiento más cercano a lo deseado.

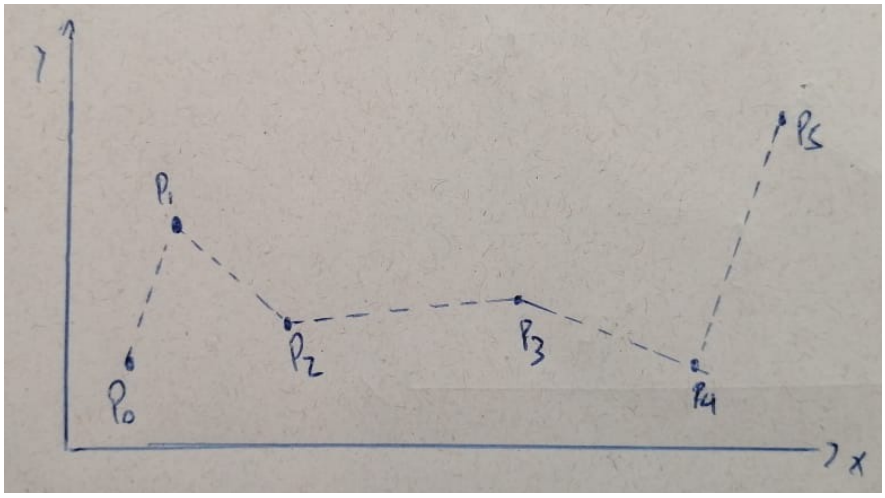
2. [Interpolación Step] *(Nota: todo t' se usará como en el formulario)*



Para la interpolación step se deben modelar 5 curvas, en las que:

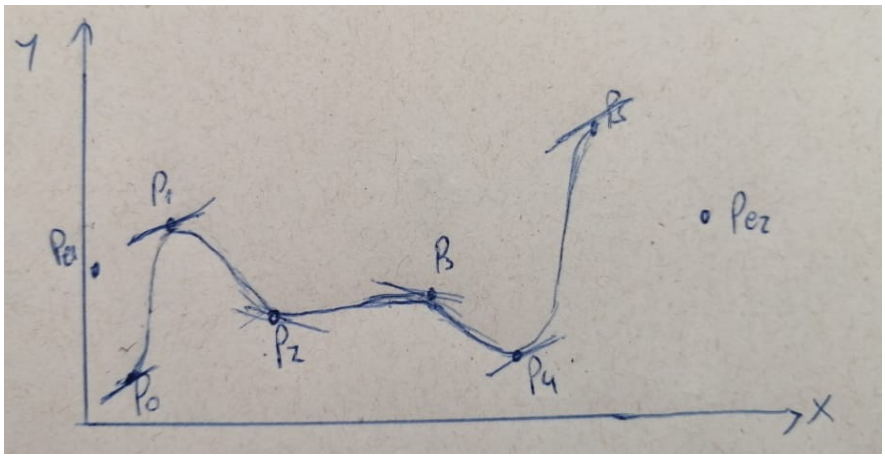
$$C_i(t') = P_i, t_1 = \frac{i}{5}, t_2 = \frac{i+1}{5}$$

[Interpolación Linear]
Para la interpolación linear, basta con generar las líneas correspondientes usando la función lineal:



$$C_i(t') = P_i \cdot (1 - t') + P_{i+1} \cdot t', t_1 = \frac{i}{5}, t_2 = \frac{i+1}{5}$$

[Interpolación Catmull-Rom]
Para esta interpolación, es necesario generar 2 puntos más para poder formar una curva en todos los puntos pedidos, para ello se usarán P_{e1} y P_{e2} en los extremos, formando lo siguiente:

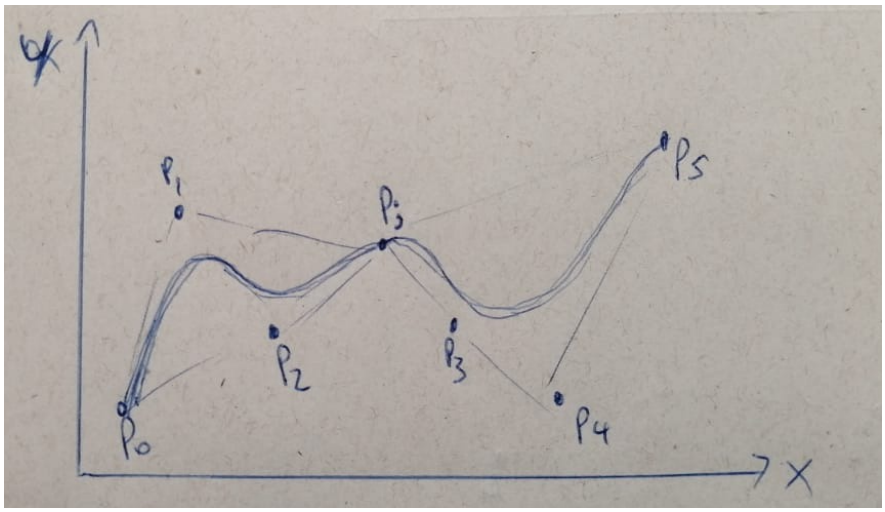


$$C_i(t') = [P_{i-1}, P_i, P_{i+1}, P_{i+2}]M_{CR}T(t'), t_1 = \frac{i}{5}, t_2 = \frac{i+1}{5}$$

En donde $C_0(t') = [P_{e1}, P_0, P_1, P_2]M_{CR}T(t')$ y $C_4(t') = [P_3, P_4, P_5, P_{e2}]M_{CR}T(t')$ Recordando que dependiendo de los puntos e1 y e2 es cómo serán formadas las tangentes en

P_0 y P_5.

[Interpolación Bézier]
Para el caso de Bézier, es necesario añadir un punto, que será llamado P_j, el cual estará entre P_2 y P_3, notando que este punto tiene que cumplir una condición.
Tomando las 2 curvas que se pueden generar: $C_0(P_0, P_1, P_2, P_j)$ y $C_1(P_j, P_3, P_4, P_5)$ se tiene:



Así, imponemos la condición de que las tangentes de las curvas C_0 y C_1 en P_j debe tener la misma dirección:

$$P_3 - P_j = c(P_j - P_2), c > 0$$

Reduciendo:

$$P_j = \frac{P_3 - cP_2}{1 + c}$$

Finalmente,

$$C_0(t') = [P_0, P_1, P_2, P_j] M_B T(t'), t_1 = 0, t_2 = 0{,}5$$
$$C_1(t') = [P_j, P_3, P_4, P_5] M_B T(t'), t_1 = 0{,}5, t_2 = 1$$

Formulario

$$S(\alpha,\beta,\gamma) = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{bmatrix}$$

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad \theta \text{ anti-horario}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$H_{xz}(s) = \begin{bmatrix} 1 & 0 & s \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(T_x,T_y,T_z) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$Q(t) = \underbrace{\begin{bmatrix} g_1 & g_2 & g_3 & g_4 \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}}_{T(t)}$$

$$\begin{bmatrix} P_1 & P_2 & T_1 & T_2 \end{bmatrix}$$

$$M_H = \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$T_i = \frac{1}{2}(P_{i+1} - P_{i-1})$$

$$C_i(t) = \begin{bmatrix} P_{i-1} & P_i & P_{i+1} & P_{i+2} \end{bmatrix} M_{CR} T(t)$$

$$M_{CR} = \frac{1}{2}\begin{bmatrix} 0 & -1 & 2 & -1 \\ 2 & 0 & -5 & 3 \\ 0 & 1 & 4 & -3 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$M_B = \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_1 = P_0 + \frac{T_0}{3}$$

$$P_2 = P_3 - \frac{T_3}{3}$$

$$t' = \frac{t - t_1}{t_2 - t_1}, \quad t \in [t_1, t_2], \quad t' \in [0,1]$$