

Trabajo Práctico: Simulación Física en Videojuegos

Contexto

En videojuegos modernos, lograr animaciones realistas de efectos físicos (como la propagación de calor, fuego o fluidos) en tiempo real es un gran desafío computacional. Estos fenómenos se modelan a menudo mediante ecuaciones diferenciales parciales, que al discretizarse, se traducen en grandes sistemas lineales de ecuaciones.

Enunciado

En este trabajo práctico simulamos la **difusión de calor** en una superficie 2D (por ejemplo, una plataforma en un juego), utilizando el método implícito que requiere resolver un sistema lineal en cada frame (paso de tiempo).

El objetivo del trabajo práctico es realizar un análisis comparativo de diferentes métodos de resoluciones de sistemas de ecuaciones lineales en cuanto al tiempo de ejecución y a la precisión de la solución cuando son utilizados en un proceso iterativo donde el resultado de la iteración anterior es utilizada para la siguiente iteración.

Simulación de difusión de calor:

Se utilizará como base el siguiente [código](#) que resuelve numéricamente la ecuación de calor en 2D sobre una cuadrícula cuadrada, con condiciones de frontera fijas y una fuente de calor interna. El método implementado es el método implícito, que garantiza estabilidad numérica pero implica resolver un sistema lineal en cada paso.

En cada iteración (frame del juego), el sistema lineal $Ax=b$ tiene tamaño proporcional a la resolución espacial de la cuadrícula interna, donde la matriz A es una matriz tridiagonal.

Tareas

- Implementar los siguientes métodos de resolución de sistemas de ecuaciones lineales:
 - Resolución optimizada para sistemas tridiagonales,
 - Gauss con pivoteo parcial
- Ejecutar la simulación con varias resoluciones espaciales (20x20, 30x30, 50x50, 70x70, 100x100, 500x500, 1000x1000 puntos interiores).
- Comparar los tiempos de ejecución y la precisión de la solución de estas dos implementaciones con la solución directa implementada en scipy (`spsolve`).
 - Comparar el tiempo promedio de ejecución de la simulación de cada implementación para las distintas resoluciones espaciales con el método directo (`spsolve`).
 - Analizar la precisión de la simulación, comparando la solución obtenida con cada método para distintas resoluciones espaciales con el método directo (`spsolve`), calculando el error cuadrático medio (RMS) entre las soluciones.
- Documentar los experimentos realizado con gráficos que muestren:
 - Tiempo vs tamaño del sistema.
 - Error numérico vs tamaño del sistema.
- Discutir las ventajas y desventajas de cada método de solución.

Entrega

- **Grupos:** Los grupos son de 4 personas.
- **Forma de entrega:** Subir al campus el código Python + reporte (documento pdf) con los resultados de los experimentos.
- **Fecha de entrega:** Martes 10 de Junio de 2025.