





A1	- I X 4	f_x									~
A	В	C	D E	F	G H	Ι	J	К	L	М	P
2											
1											
5											
5											
			JAO			-					
3											
0				-XCE	i háci	\mathbf{CO}					
1						00					
2											
4											
5											
6			D	odro	Corou	oro					
7				earo	COLCO	ela					
8											
9			Dpto.	Mater	nática Ar	olicada					
0			Cion	aiaa da	la Comr	utooid	ś'n				
2			Ciend	Jas ue		Julaci	חו				
2			Lloiv	oroida	d do Co	ntohri	•				
4			Univ	ersiua	au ue ca	παρπ	d				
5											
6											
7				orcuor	maunica						
8			L L	ULLEI	perinca	1.69					
9											
8 F	Hoja1	(+)				4					•



- Transferencia de datos
- Organización y análisis de datos Tablas dinámicas
- Evaluación económica
- Conversión de unidades
- Programación en Excel con VBA
- Ecuaciones diferenciales ordinarias
- Ecuaciones diferenciales en derivadas parciales





 Aplicación de hojas de cálculo en problemas de ingeniería.



Transferencia de datos

- Algunas aplicaciones requieren que sean leídos o importados ficheros diferentes de Excel.
- Para leer ficheros tipo texto se siguen los siguientes pasos:
 - Asegurarse que el fichero es un fichero texto (extensión típica .txt, .csv, o .prn).
 - En Excel seleccionar Archivo→Abrir. Cuando aparece la ventana de diálogo seleccionar Archivos de texto.
 Seleccionar el archivo.
 - Aparece el Asistente. Es necesario seleccionar si el fichero tiene delimitadores entre campos o si son de ancho fijo.
 - Si hay delimitadores, seleccionar el tipo de separador.
 - Finalmente se selecciona el formato.



Transferencia de datos - Lectura

	Abrir Buscar <u>e</u> n:	Datos		_		✓ Herramientas ▼			
	Historial	datos_prueba_a datos_prueba_d datos_prueba_d	f elpc eltab			_			
	Mis documentos								
	Escritorio								
	* Favoritos								
		Nombre de archivo:			•	Abrir 🝷			
Asistente para importar texto - paso	Mis sitios de red	<u>T</u> ipo de archivo:	Archivos de texto		•	Cancelar	2 de 3		? 🛛
El asistente para convertir texto en columnas Si esto es correcto, elija Siguiente, o bien elija Tipo de los datos originales Elija el tipo de archivo que describa los datos O Delimitados - Caracteres como como O De ancho fijo - Los campos están alim	estima que sus d el tipo de datos con mayor preci as o tabulaciones eados en column	atos son Delimitado que mejor los descr sión: separan campos. as con espacios eni	is. iba. tre uno y otro.		Esta pantalla le permite Las líneas con flechas in Para CREAR un salto Para ELIMINAR un sa Para MOVER un salto	establecer el ancho ndican un salto de c o de línea, haga clic alto de línea, haga clic o de línea, haga clic	de los campos (s olumna. en la ubicación de doble clic en la líne y arrástrelo.	altos de columna). eseada. ea.	
Comenzar a importar en la fila: 1	Origen del	archivo: MS-DOS ueba_af.txt.	i (PC-8)	•	- <u>V</u> ista previa de los dato: 10	s	30	50	60
1 Edad Sexo 2 F 32 M 42 F 51 F		Casa Propia 1 1 0 0	. Cas 1 0 1 1		Edad 2 2 2 1	Sexo F M F F		Casa Propia 1 1 0 0	Casadd
	Cancelar	< Atrás		alizar			Cancelar	< At <u>r</u> ás	Einalizar



Importación de datos desde páginas Web

- Es posible importar datos desde una página Web.
 - La forma más fácil es utilizar Datos > Obtener datos externos > Desde Web. Aparece un navegador donde se puede colocar la URL deseada. Ejemplo:

http://physics.nist.gov/cgi-bin/Compositions/stand_alone.pl?ele=&ascii=html&isotype=some

 Se seleccionan los datos y se da en el botón Importar.

Transferencia de datos - Lectura

🕅 🔒 5 - 🖓 - 🗐 =				
ARCHIVO INICIO INSERTAR DIS	EÑO DE PÁGINA FÓRMU	LAS DATOS		
Desde Access Image: Constant of the sector of the sect	Actualizar todo •	Ž↓ <mark>Z Z</mark> Z↓ Ordenar		
Obtener datos externos	Conexiones	Or		
	Nueva consulta v	web		? ×
	Dirección: http:	//physics.nist.gov/cgi-bin/Composition	ons/stand_alor Ir 🗧 🔿 🦻	👌 🛃 🛃 🔚 Opciones
	Haga clic en 🕨	al lado de las tablas que desea selo	eccionar; a continuación, elija Importa	r.
	•			
	Aton	nic weights and isoto	pic Compositions for A	Il Elements
	Isotope	<u>Kelative</u> Atomic Mass	<u>Isotopic</u> <u>Stan</u> Composition Atomic	<u>dard</u> Weight Notes
		<u>intointe intaiss</u>	<u>composition</u> <u>ritomic</u>	ricigat rivicis
	1 H	1 1.007 825 032 23(9)	0.999 885(70) [1.007 84, 1	1.008 11] m
	D	2 2.014 101 778 12(12)	0.000 115(70)	
	Т	3 3.016 049 2779(24)		
	2 He	3 3.016 029 3201(25)	0.000 001 34(3) 4.002 60	2(2) g.r
		4 4.002 603 254 13(6)	0.999 998 66(3)	
	3 Li	6 6.015 122 8874(16)	0.0759(4) [6.938, 6.99	97] m ₊
				Importar Cancelar

- Para exportar datos a un fichero tipo texto.
 - En Excel seleccionar Archivo \rightarrow Guardar como...
 - Si se desea que los datos de cada línea se separen por tabuladores, seleccionar Texto (delimitado por tabulaciones). Se añade automáticamente el sufijo .txt al fichero.
 - Si se desea que los datos de cada línea se separen por comas, seleccionar CSV (delimitado por comas). Se añade automáticamente el sufijo .csv al fichero.



Transferencia de datos - Escritura





Organización y Análisis de datos – Tablas dinámicas

- Creación de listas en Excel.
 - Introducir los datos (pueden tener encabezado).
 - Para introducir un nuevo registro se puede usar un formulario de datos. Para ello:
 - Hacer clic en una celda del rango o de la tabla donde desee agregar el formulario.
 - Para agregar el botón Formulario Botón Control de formulario a la Barra de herramientas de acceso rápido:
 - Hacer clic en la flecha de la Barra de herramientas de acceso rápido y hacer clic en Más comandos. En el cuadro Comandos disponibles en, hacer clic en Todos los comandos y seleccionar el botón Formulario... de la lista. Hacer clic en Agregar y Aceptar. En la Barra de herramientas de acceso rápido, hacer clic en Formulario.
 - Ejemplo: Provincias_España.xls.



100			_					
	x∎	<u>₽</u> 5 ° ° =	;	Ŧ				
	ARC	CHIVO INICIO INSE	RT	AR DISEÑO DE PÁG	INA FÓRMULAS	DATOS	REVISAR	
	Pe	Cortar Copiar -	Ar N	rial - 11 I <u>K S</u> - 🖽 - 🏒		≫. €≣ ≇≣	🖶 Ajustar text	
		Portapapeles 5	(Provincias_España			? ×	
	A2	2		Provincia:	Álava		1 de 53	
				<u>H</u> ombres:	154376		Nuevo	
				<u>M</u> ujeres:	155259		<u>E</u> liminar	
		Α		<u>T</u> otal:	309,635		Restaurar	
	1		Н	Porcentaje:	0.67%			
	2 Provincia			Superf <u>i</u> cie (km2):	3038		uscar <u>a</u> nteri	
ľ	3	Alava Porcentaie:			0.60%	E	uscar <u>s</u> iguie	
	4	Albacete			404.00		Criterios	
	5	Alicante		Densidad (hab./km2):	101.92			
	6	Almería					Cerrar	
	7	Asturias						
	8	Avila						
	9	Badajoz				Ŧ		
	10	Barcelona		L,000,000 L,. 10,0				



- Ordenación de datos en Excel.
 - Una lista puede ser ordenada seleccionando el rango de datos y pulsando sobre una de las opciones de ordenar de A a Z (ascendente) o Z a A (descente) en la cinta de opciones de las pestañas Inicio o Datos (ordenar y filtrar).







Organización de datos - Ordenación

	Α		В	С	D	E	F	G	Н	I		
1				POBLACIÓ	N (2008)							
2	Provincia	Orde	nar	-	-	_	danili d	-		? <mark>×</mark>		
3	Àlava	*A	Agregar nivel	× Eliminar n	nivel	ar nivel		nes 🔽 Mis	datos tienen er	ncabezados		
4	Albacete		↓ <u></u> ,_,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,									
5	Alicante	Co	lumna		Ordenar	según		Criterio de or	denación			
6	Almería	Ore	denar por Total	I	 Valores 		-	De mayor a m	ienor	-		
7	Asturias											
8	Ávila											
9	Badajoz											
10	Barcelona											
11	Burgos											
12	Cáceres											
13	Cádiz											
14	Cantabria											
15	Castellón							4	ceptar	Cancelar		
16	Ceuta											
17	Ciudad Real		260,649	261,694	522,343	1.13%	19813	3.92%	26.36			
18	C⊒doba		392,658	406,164	798,822	1.73%	13771	2.72%	58.01			
19	Cuenca		109,058	106,216	215,274	0.47%	17141	3.39%	12.56			-
	< → +	loja1	Provincias_E	spaña	+		:				Þ]
LIST	го 🔠		PROMEDIO	: 255048.3226	RECUENTO: 41	5 SUMA: 928	37589.43			+ +	100 %	6



- Filtrado de datos en Excel.
 - Excel dispone de operaciones similares a las bases de datos como es la recuperación de la información que satisface ciertas condiciones.
 - El filtrado de listas permite seleccionar registros que satisfacen determinados criterios.
 - Se usará la sub-opción Filtros de números de la opción Filtro en el menú Datos.

- Ejercicios (Provincias_España.xls):
 - Las 10 provincias que tienen mayor densidad de población.
 - Qué provincias tienen superficies que exceden 15,000 km².
 - Qué provincias tienen poblaciones entre 500000 y 1 millón de habitantes.

- Las 10 provincias que tienen mayor densidad de población:
 - Seleccionar una celda arbitraria dentro de la lista.
 - Seleccionar Filtro en el menú Datos.
 - Aparecen flechas hacia abajo en el encabezado.
 - Pulsar sobre la flecha en el encabezado de densidad de población seleccionar Filtros de número y se selecciona los Diez mejores. Aparecen una ventana de diálogo en la que se puede modificar la selección.



- Qué provincias tienen areas que exceden 15,000 km²:
 - Se pulsa sobre la flecha en el encabezado Superficie \rightarrow Filtros de número.
 - En el cuadro de diálogo que aparece seleccionar Mayor que y colocar 15000 en el siguiente campo. Pulsar el botón Aceptar.
 - Aparecen los resultados en el mismo orden que los originales.

- Qué provincias tienen poblaciones entre 500000 y 1 millón:
 - Se pulsa sobre la flecha en el encabezado Total → Filtros de número.
 - En el cuadro de diálogo que aparece seleccionar Entre rellenando los campos con los valores apropiados (mayor o igual que 500000, menor o igual que 1000000). Pulsar el botón Aceptar.
 - Aparecen los resultados en el mismo orden que los originales.



Organización de datos - Filtrado

	A	В	С	D	E	F	G	Н	I.	-
1			POBLACIÓN	V (2008)						
						Superficie		Densidad		
2	Provincia 💷	Hombre -	Mujere 👻	Total 👻	Porcenta 🛫	(km2) 👻	Porcenta -	(hab./km 👻		
3	Álava	154,376	155,259	309,635	32.41	Autofiltro perso	nalizado		? ×	
4	Albacete	199,820	197,673	397,493	41.61					
5	Alicante	947,523	943,954	1,891,477	197.99	Mostrar las filas	en las cuales:			
6	Almería	343,716	323,919	667,635	69.89	Total				
7	Asturias	518,291	561,847	1,080,138	113.07	es mayor	o igual a 🔻	500000	-	
8	Ávila	86,836	84,979	171,815	17.99	QV				
9	Badajoz	340,243	345,003	685,246	71.73					
10	Barcelona	2,668,359	2,748,088	5,416,447	566.98	es menor	o igual a 💌	1000000	•	
11	Burgos	189,675	183,997	373,672	39.11					
12	Cáceres	205,949	206,549	412,498	43.18	Use ? para repre	esentar cualquie	er carácter indivi	dual	
13	Cádiz	608,616	611,851	1,220,467	127.75	Use * para repre	esentar cualquie	er serie de caract	eres	
14	Cantabria	285,469	296,669	582,138	60.94		ſ	Aceptar	Cancelar	
15	Castellón	299,829	295,086	594,915	62.27					
16	Ceuta	39,385	38,004	77,389	8.10	19	0.11%	4075.11		-
17	Ciudad Real	260,649	261,694	522,343	54.68%	5 19813	114.69%	26.36		
18	Cuenca	109,058	106,216	215,274	22.53%	5 17141	99.22%	12.56		
19	C⊡doba	392,658	406,164	798,822	83.62%	5 13771	79.72%	58.01		
20	Girona	372,266	359,598	731,864	76.61%	5910	34.21%	123.83		
21	Granada	447,280	453,940	901,220	94.34%	5 <u>12647</u>	73.21%	71.26		
22	Guadalajara	122,327	115,460	237,787	24.89%	5 12212	70.69%	19.47		
23	Guipúzcoa	344,679	356,377	701,056	73.38%	5 1980	11.46%	354.07		
24	Huelva	252,394	255,521	507,915	53.17%	6 10128	58.63%	50.15		
25	Huesca	114,939	110,332	225,271	23.58%	15636	90.51%	14.41		



- Problema: Realizar búsquedas de valores en una tabla de datos tabulares para utilizarlos en otros cálculos.
- Excel dispone de las funciones BUSCAR, BUSCARV, BUSCARH, COINCIDIR e INDICE para acceder a datos tabulares o tablas.
- Ejemplo: Busqueda_tablas.xls

- La función BUSCAR devuelve un valor procedente de un rango de una fila o columna o de una matriz.
- Tiene dos formas de sintaxis: vectorial y matricial.
- La forma vectorial de BUSCAR busca en un rango de una fila o de una columna un valor (vector) y devuelve un valor desde la misma posición en un segundo rango de una fila o de una columna.
- Sintaxis: BUSCAR(valor_buscado;vector_de_comparación;vector_resultado)
- El vector_de_comparación debe estar ordenado en orden ascendente.



Longitud		Factor de diseño			
ft	ft m		Aluminio		
10	3.05	10	4		
20	6.10	15	7		
40	12.19	20	10		
80	24.38	30	15		
160	48.77	35	17		

Longitud dada =	20
Unidades =	ft
Material =	Acero
Factor de diseño =	15



- La función BUSCARV busca un valor en la primera columna de un rango de celdas y devuelve el valor en la columna especificada. Opcionalmente se puede indicar el criterio de búsqueda.
- Sintaxis:

BUSCARV(valor_buscado;matriz_buscaren;ind_columnas;ordenado)



Búsqueda en tablas - BUSCARV

1	2	3	4	5	6	7	8	9
Temperatura (C)	Densidad (kg/m3) Er	nergía Interna (kJ/kg)	Entalpía (kJ/kg)	Entropía (J/g*K)	Cv (J/g*K)	Cp (J/g*K)	Veloc.sonido (m/s)	Viscosidad (Pa*s)
10	999.7	42.018	42.119	0.15108	4.1906	4.1952	1447.3	0.0013059
20	998.21	83.906	84.007	0.29646	4.1567	4.1841	1482.3	0.0010016
30	995.65	125.72	125.82	0.43673	4.1172	4.1798	1509.2	0.00079735
40	992.22	167.51	167.62	0.57237	4.0734	4.1794	1528.9	0.00065298
50	988.04	209.32	209.42	0.70377	4.0262	4.1813	1542.6	0.00054685
60	983.2	251.15	251.25	0.83125	3.9765	4.185	1551	0.0004664
70	977.76	293.02	293.12	0.95509	3.9251	4.1901	1554.7	0.00040389
80	971.79	334.95	335.06	1.0755	3.8728	4.1968	1554.4	0.00035435
90	965.31	376.96	377.06	1.1928	3.8203	4.2052	1550.5	0.00031441

Temperatura (C)	40	< Introduce Temperatura
Densidad (kg/m3)	992.22	
Energía Interna (kJ/kg)	167.51	
Entalpía (kJ/kg)	167.62	
Entropía (J/g*K)	0.57237	
Cv (J/g*K)	4.0734	
Cp (J/g*K)	4.1794	
Veloc.sonido (m/s)	1528.9	
Viscosidad (Pa*s)	0.00065298	



- La función BUSCARH busca un valor en la primera fila de un rango de celdas y devuelve el valor en la fila especificada. Opcionalmente se puede indicar el criterio de búsqueda.
- Sintaxis:

BUSCARH(valor_buscado;matriz_buscaren;ind_filas;ordenado)



1	Temperatura (C)	10	20	30	40	50	60	70	80	90
2	Densidad (kg/m3)	999.7	998.21	995.65	992.22	988.04	983.2	977.76	971.79	965.31
3	Energía Interna (kJ/kg)	42.018	83.906	125.72	167.51	209.32	251.15	293.02	334.95	376.96
4	Entalpía (kJ/kg)	42.119	84.007	125.82	167.62	209.42	251.25	293.12	335.06	377.06
5	Entropía (J/g*K)	0.15108	0.29646	0.43673	0.57237	0.70377	0.83125	0.95509	1.0755	1.1928
6	Cv (J/g*K)	4.1906	4.1567	4.1172	4.0734	4.0262	3.9765	3.9251	3.8728	3.8203
7	Cp (J/g*K)	4.1952	4.1841	4.1798	4.1794	4.1813	4.185	4.1901	4.1968	4.2052
3	Veloc.sonido (m/s)	1447.3	1482.3	1509.2	1528.9	1542.6	1551	1554.7	1554.4	1550.5
9	Viscosidad (Pa*s)	0.001306	0.001002	0.000797	0.000653	0.000547	0.000466	0.000404	0.000354	0.000314

Temperatura (C)	30 <-	
Densidad (kg/m3)	995.65	
Energía Interna (kJ/kg)	125.72	
Entalpía (kJ/kg)	125.82	
Entropía (J/g*K)	0.43673	
Cv (J/g*K)	4.1172	
Cp (J/g*K)	4.1798	
Veloc.sonido (m/s)	1509.2	
Viscosidad (Pa*s)	0.000797	

-- Introduce Temperatura



- La función COINCIDIR devuelve la posición de un elemento en una matriz.
- Sintaxis: COINCIDIR(valor_buscado;matriz_buscaren;tipo_coincidencia)

1	Temperatura (C)	10	20	30	40	50	60	70	80	90
2	Densidad (kg/m3)	999.7	998.21	995.65	992.22	988.04	983.2	977.76	971.79	965.31
3	Energía Interna (kJ/kg)	42.018	83.906	125.72	167.51	209.32	251.15	293.02	334.95	376.96
4	Entalpía (kJ/kg)	42.119	84.007	125.82	167.62	209.42	251.25	293.12	335.06	377.06
5	Entropía (J/g*K)	0.15108	0.29646	0.43673	0.57237	0.70377	0.83125	0.95509	1.0755	1.1928
6	Cv (J/g*K)	4.1906	4.1567	4.1172	4.0734	4.0262	3.9765	3.9251	3.8728	3.8203
7	Cp (J/g*K)	4.1952	4.1841	4.1798	4.1794	4.1813	4.185	4.1901	4.1968	4.2052
8	Veloc.sonido (m/s)	1447.3	1482.3	1509.2	1528.9	1542.6	1551	1554.7	1554.4	1550.5
9	Viscosidad (Pa*s)	0.001306	0.001002	0.000797	0.000653	0.000547	0.000466	0.000404	0.000354	0.000314

Temperatura (C) Veloc.sonido (m/s)





- La función INDICE resulta útil para extraer un valor de una tabla dada la posición del valor en la tabla.
- Sintaxis:

INDICE(matriz; num_fila; num_columna)

Datos:		
0.02451	0.93331	0.84756
0.08995	0.03905	0.32620
0.94215	0.49682	0.62290
0.67168	0.60431	0.84449

Elemento 2,2 = 0.0390457

Fila 1 = 0.0245112 0.9333076 0.847564

Columna 2 = 0.9333076 0.0390457 0.4968178 0.6043112



- Las listas normalmente se organizan de manera lineal, es decir, en columnas adyacentes.
- Algunas listas se pueden organizar de manera bidimensional para mostrar una mejor interrelación entre los datos.
- Las listas bidimensionales reestructuradas se conocen como *tablas pivot o dinámicas*.



- Ejemplo: (datos_pobl_usa.xls)
 - Primero asegurarse que la lista está formada por bloques de celdas contiguas con un encabezado en cada columna.
 - Se selecciona cualquier celda dentro de la lista y seleccionar en el menú Insertar – Tablas la opción Tabla dinámica.
 - Aparece el Asistente para crear tabla dinámica con el rango de datos incluyendo los encabezados. Seleccionar Nueva hoja de cálculo. Pulsar el botón Aceptar.
 - Aparece una hoja de trabajo que incluye una ventana de Campos de tabla dinámica.
 - Seleccionar los campos según las filas y columnas que se requieran. También los campos que se quiere mostrar en valores y los filtros.



	А	В	С	D	E	F	G			
1									Campos d	le tabla di 🌄
2										
3	Suma de Población	Año <		(Seleccionar camp	os para agregar 🛛 🐇 👻
4	Etiquetas de fila 🛛 💌	1970	1980	1990	2000	Total general			al informe:	
5	🗆 Midwest	5295415	5569479	5755873	6237411	22858178			✓ Estado	
6	Missouri	4677623	4916762	5117073	5595211	20306669				
7	North Dakota	617792	652717	638800	642200	2551509				
8	■Northeast	30991880	30370039	30875562	32305902	124543383				
9	New York	18241391	17558165	17990455	18976457	72766468				
10	Penssylvania	11800766	11864720	11881643	12281054	47828183			MÁS TABLAS	
11	Rhode Island	949723	947154	1003464	1048391	3948732				
12	Northwest	3715827	4534204	5416735	6521053	20187819				
13	Alaska	302583	401851	550043	626932	1881409				
14	Washington	3413244	4132353	4866692	5894121	18306410				
15	🗏 🗉 South	16527277	20894173	25753921	31110206	94285577				
16	Florida	6791418	9746961	12937926	15982378	45458683			Arrastrar campos	entre las áreas siguientes
17	North Carolina	5084411	5800415	6628637	8049313	25562776			Anastai campos	citic las areas siguientes.
18	Virginia	4651448	5346797	6187358	7078515	23264118			T FILTROS	
19	Southwest	33379320	40783012	50040925	59024729	183227986			T TIETROS	
20	California	19971069	23667764	29760021	33871648	107270502				Año 🔻
21	Colorado	2209596	2889735	3294394	4301261	12694986				
22	Texas	11198655	14225513	16986510	20851820	63262498				
23	Total general	89909719	102150907	117843016	135199301	445102943			FILAS	∑ VALORES
24									Region	▼ Suma de Pobl ▼
25	(Estado	-
26	Suma de Població	n							Estado	
27	12000000							-		
	Hoja1	Datos Tal	ola dinámic	🕂 :	•				Aplazar actua	lización ACTUALIZAR



Tablas pivot o dinámicas

Suma de Población	Año 🗾				
Etiquetas de fila 🔄	1970	1980	1990	2000	Total general
⊟ Midwest	5295415	556 9 479	5755873	6237411	22858178
Missouri	4677623	4916762	5117073	5595211	20306669
North Dakota	617792	652717	638800	642200	2551509
🗏 Northeast	30991880	30370039	30875562	32305902	124543383
New York	18241391	17558165	17990455	18976457	72766468
Penssylvania	11800766	11864720	11881643	12281054	47828183
Rhode Island	949723	947154	1003464	1048391	3948732
Northwest	3715827	4534204	5416735	6521053	20187819
Alaska	302583	401851	550043	626932	1881409
Washington	3413244	4132353	4866692	5894121	18306410
🗏 South	16527277	20894173	25753921	31110206	94285577
Florida	6791418	9746961	12937926	15982378	45458683
North Carolina	5084411	5800415	6628637	8049313	25562776
Virginia	4651448	5346797	6187358	7078515	23264118
Southwest	33379320	40783012	50040925	59024729	183227986
California	19971069	23667764	29760021	33871648	107270502
Colorado	2209596	2889735	3294394	4301261	12694986
Texas	11198655	14225513	16986510	20851820	63262498
Total general	89909719	102150907	117843016	135199301	445102943



Tablas pivot o dinámicas

– Herramientas – Gráfico dinámico





Evaluación económica


- Una parte importante en la evaluación de proyectos es la evaluación económica.
- Se basa en el valor del dinero en el tiempo. La terminología empleada es el principal para indicar la cantidad prestada y el interés que es el pago adicional por el uso del dinero.
- Los cálculos de interés se basan en la tasa de interés *i*.
- Los cálculos económicos se basan en el uso del interés compuesto. Así para n períodos de interés, la cantidad total de dinero acumulado al final del último período de interés es: F = F_n = P(1 + i)ⁿ
- Ejemplos: Comparacion_Economica1.xls



Cálculos financieros básicos

 Problema: Calcular el capital acumulado para un depósito a un interés y período dado.



20



- Problema: Calcular el capital acumulado para un depósito a un interés y período dado con capitalización menores al año.
 - m = períodos de capitalización
 - n = número total de períodos de capitalizacion = mxnúmero de años

 $F = F_n = P(1 + i/m)^n$

Interés compuesto: Variación de la frecuencia del interés

Interés anual =	5306.60
Interés cuatrimetral =	5402.97
Interés mensual =	5425.28

Interés diario = 5436.19

 Una alternativa económica realista incluye normalmente un período de inversión inicial seguida de una serie de flujos de caja.



 Un flujo de caja típico para un préstamo consiste en una inversión inicial seguido de una serie de n pagos uniformes. En ese caso las cuotas se calculan como:

$$A = P \left[\frac{(i/m)(1+i/m)^{n}}{(1+i/m)^{n}-1} \right]$$



 Problema: Calcular el pago uniforme (amortización) para devolver una cantidad inicial P. Excel tiene una función Pago

Pago uniforme de una inversión

	<u> </u>
-	0.00

P = 1055050.92



• Problema: Calcular el valor actual de un pago uniforme A. Excel tiene una función VA.

Valor presente de una inversión

A =	140000
i =	0.08
n =	12
P =	-1,055,050.92 €



• Problema: Calcular el valor futuro de un pago uniforme A. Excel tiene una función VF.

Valor futuro de un pago

A =	140000
i =	0.08
n =	12
F =	2 656 797 70 €



 Problema: Calcular el valor presente neto de un flujo de caja no uniforme en periodos uniformes. Excel tiene una función VNA.

Valor Presente de una inversión





i =

Comparación de Alternativas Flujos de caja no uniformes

Problema: Comparar varias alternativas de flujos de caja. Se selecciona la de mayor Valor Presente Neto.

Comparación de dos oportunidades de inversión

0.1

Final año	Flujo de Caja Alternativa A	Flujo de Caja Alternativa B
0	-3500000	-3500000
1	1200000	600000
2	1200000	900000
3	1200000	1100000
4	1200000	1300000
5	1200000	1500000
6	1200000	1800000
VPN =	1,726,312.84	1,451,055.03

Excel



Comparación de Alternativas Tasa interna de retorno (TIR)

- El método de la Tasa Interna de Retorno (TIR) es otro criterio muy usado para comparar varias alternativas de inversión. A diferencia del método del Valor Presente no hay necesidad de especificar una tasa de interés.
- Si dibujamos el valor presente de un flujo de caja en función de la tasa de interés, la TIR es el punto de cruce, es decir, el valor de la tasa de interés al cual el valor presente neto se hace cero.
- Durante la comparación de alternativas mediante la TIR se escogerá aquella alternativa que tenga la *mayor* tasa interna de retorno.
- Excel tiene la función TIR que calcula la tasa interna de retorno directamente.



VPN - TIR

Valor Presente en función de la Tasa de Interés





Comparación de Alternativas Tasa interna de retorno (TIR)

Comparación de dos oportunidades de inversión

i =	0.1	
– . , ~	Flujo de Caja	Flujo de Caja
Final ano	Alternativa A	Alternativa B
0	-3500000	-3500000
1	1200000	600000
2	1200000	900000
3	1200000	1100000
4	1200000	1300000
5	1200000	1500000
6	1200000	1800000
VPN =	1,726,312.84	1,451,055.03
TIR =	26%	21%
Excel		



- Pueden existir varias TIR cuando el flujo de caja tiene más de un cambio de signo. Por tanto se recomienda usar este indicador cuando un proyecto tenga sólo un cambio de signo.
- Bajo determinadas situaciones el VPN y la TIR, se contradicen cuando se dan dos escenarios:
 - Escenario 1: Los montos de inversión de los proyectos son diferentes.
 - Escenario 2: La distribución temporal de los FC no son iguales: en dos alternativas A y B, A tiene los FC más grandes al inicio y B, los tiene más concentrados hacia el final de la vida útil.
- En caso de contradicción se debe utilizar el VPN. Se puede demostrar mediante el cálculo de la TIR marginal o tasa de indiferencia.



Conversión de Unidades



- En ingeniería es frecuente la necesidad de conversión de unidades entre diferentes sistemas de unidades.
- La forma usual de realizar esta tarea es mediante equivalencias de unidades que se obtienen a partir de factores de conversión tabulados.
- Las hojas de cálculo también se pueden usar con este propósito eliminando la necesidad de realizar cálculos manuales.



- Los sistemas de unidades más comunes son:
 - Sistema Internacional de Unidades
 - Sistema Anglosajón de Unidades
 - Sistema métrico decimal
 - Sistema Cegesimal de Unidades o cgs
 - Sistema Técnico de Unidades o mks
- Con Excel se pueden realizar conversiones simples y complejas de unidades.



- Una conversión simple incluye sólo unidades de la misma dimensión, p.e. pies a metros (longitud).
- Excel ofrece la función de Ingeniería CONVERTIR (instalar Herramientas → Complementos → Herramientas para análisis).
- En las conversiones simples se multiplica la cantidad original por el factor de equivalencia unitario apropiado.



- Ejemplo manual: Conversión simple de pies a metros Convertir 2.5 pies a su número equivalente en metros, usando el factor de conversión
 - 1 pie = 0.3048 m.
 - La conversión es: $L=2.5 ft \times \frac{0.3048 m}{1 ft} = 0.762 m$

- En Excel se usa la función CONVERTIR.
- Sintaxis:

CONVERTIR(*número*, *de_unidad*, *a_unidad*) *número* es el número a convertir. *de unidad* abreviación de la unidad original.

a unidad abreviación de la unidad final.

• Ejemplo:

=CONVERTIR(1.0, "lbm", "kg")

Convierte 1 libra masa a kilogramos (0,453592).



- Las abreviaciones de unidad se escriben entre comillas dobles y se distinguen entre mayúsculas y minúsculas (Ver ayuda).
- Ejemplos:
 - =CONVERTIR(68, "F", "C")

Convierte 68 grados Fahrenheit a Celsius (20).

=CONVERTIR(2,5, "ft", "sec")

Los tipos de datos no son iguales, por tanto se devuelve un error (#N/A).



- Se refieren a conversiones donde intervienen varias unidades.
- Ejemplo: Convertir 6.3 lb/sq-inch (psi) a newtons /m² (Pa) donde 1 lb_f/in² = 6894.8 N/m².

$$P = 6.3(lb_f / in^2) \times \frac{6894.8(N / m^2)}{1(lb_f / in^2)} = 43437 N / m^2$$

$$P = \frac{6.3 \, lb_f}{in^2} \times \frac{4.44822 \, N}{1 \, lb_f} \times \left(\frac{39.37 \, in}{1 \, m}\right)^2 = 43437 \, N \, / \, m^2$$



Ejemplo de Conversiones complejas

=CONVERTIR(6.3, "lbf", "N")*CONVERTIR(1, "m", "in")^2
Convierte 6.3 libras a newtons y se multiplica por la conversión de metros a pulgadas.



VBA (Visual Basic for Applications) en Excel Programación en Excel



- VBA es la versión de Microsoft Visual Basic para los productos Office.
- Permite automatizar tareas otorgándole un lenguaje orientado a objetos a los productos Office, así como la comunicación entre las distintas aplicaciones.
- Utiliza el Editor de VBA.
- Cada objeto en VBA tiene propiedades y métodos. Las propiedades se pueden cambiar en la ventana propiedades (F4).



- Para activar el editor de VB se puede usar una de las siguientes opciones:
 - Presionar Alt + F11
 - Seleccionar Desarrollador \rightarrow Visual Basic







- En la ventana del Explorador de proyectos se gestionan los módulos.
- Los módulos pueden ser de cuatro tipos:
 - **Procedimientos Sub**. Conjuntos de instrucciones que ejecutan alguna acción.
 - **Procedimientos Function**. Es un conjunto de instrucciones que devuelven un solo valor.
 - **Procedimientos Property**. Son procedimientos especiales que se usan en módulos de clase.
 - **Declaraciones**. Es información acerca de una variable que se le proporciona a VBA.
- Un solo módulo de VBA puede guardar cualquier cantidad de procedimientos Sub, procedimientos Function y declaraciones.



- Excel incluye cerca de 200 objetos, que representan rangos de celdas, gráficos, hojas de cálculo, libros y la propia aplicación de Excel.
- Cada objeto tiene propiedades (que permiten acceder y controlar sus atributos) y métodos (funcionalidades).
- El examinador de objetos es una herramienta que permite navegar por los objetos para explorar sus propiedades y métodos.
- Para abrir el examinador de objetos en VBA pulsar F2 o seleccionar: Ver → Examinador de Objetos o seleccionar el icono respectivo





- Para referirse a la Hoja1 del Libro1 se usa Application.WorkBooks("Libro1.xls").WorkSheets("Hoja1")
- Para omitir una referencia específica a un objeto se usa los objetos activos. Si Libro1 es el libro de trabajo activo, la referencia anterior se puede simplificar a WorkSheets("Hoja1").Range("A1")
- Si se sabe que la Hoja1 es la hoja activa, se puede incluso simplificar más Range("A1")



Examinador de Objetos



📽 Examinador de objet	tos	×
<todas></todas>		
	- ▲ ×	
Clases	Miembros de 'Hoja6'	
🖾 Hoja1 🛛 📃 🔺	📕 🖘 Activate	^
🖾 Hoja2	🔊 Application	
🖾 Hoja4 🛛 🗧 📃	😼 AutoFilter	
🖏 Hoja5	🔊 AutoFilterMode	
🖾 Hoja6	🖘 Calculate	
🖾 HPageBreak	🔊 Cells	
🖾 HPageBreaks	🖘 ChartObjects	
💐 HTMLCheckbox 📃	🔊 CheckSpelling	
🛤 HTMLHidden 🛛 🞽	📕 🖘 CircleInvalid	×
<todas></todas>		



- Excel es una aplicación con un modelo de tres niveles:
 - El primer nivel es el de **servicios de cliente**, que es la interfaz que permite a los usuarios manejar la aplicación.
 - El segundo nivel es el modelo de objetos de Excel, que es el que se utiliza para realizar las operaciones en el libro de cálculo (Workbook) o en las hojas de cálculo (Worksheets). Cada comando de Excel se puede manejar mediante el modelo de objetos.
 - El tercer nivel es el de **servicios de datos**, que es el que mantiene los datos en las hojas de cálculo que son modificados por los comandos del modelo de objetos de Excel.



- El modelo de objetos de Excel contiene una gran cantidad de elementos ordenados en forma jerárquica. Algunos son:
 - **Application**: Es el objeto que se encuentra en la base de la jerarquía del modelo de objetos de Excel y representa a la aplicación en sí.
 - Workbooks: Objetos que representan los libros de cálculo o archivos de Excel. Se encuentra debajo del objeto application en la jerarquía.
 - Worksheets: Objetos que representan las hojas de cálculo de Excel. Este objeto pertenece al objeto workbook.
 - **Ranges**: Objeto que representa un rango de celdas. Este objeto pertenece al objeto worksheet.
 - Charts: Objetos que representan gráficos.
 - Pivot Tables: Objetos que representan tablas dinámicas.



- El objeto **Application** representa el programa Excel. Entrega acceso a las opciones y otras funcionalidades de Excel.
- La propiedad **ActiveSheet** se refiere a la hoja de cálculo activa. Ejemplo:

Application.ActiveSheet.Cells(1, 2) = time

- Le dice a Excel que coloque el valor de time en la celda que está en la fila 1 y columna 2.
- La propiedad ScreenUpdating le indica a Excel si debe refrescar la pantalla cuando se ejecuta código.

Application.ScreenUpdating = False



- El objeto Workbook representa un archivo Excel.
- El objeto ActiveWorkbook pertenece al objeto Application, y entrega el objeto Workbook activo. Ejemplo: Application.ActiveWorkbook.Save
- El objeto ActiveSheet pertenece al objeto Workbook y se refiere a la hoja de cálculo activa.

Application.ActiveSheet.Delete

- La propiedad Names entrega la lista de nombres que se han definido en ese Workbook.
- La propiedad Path se refiere al directorio donde se encuentra el Workbook. Ejemplo:

directorio = Application.ActiveWorkbook.path



- La colección Workbooks agrupa a todos los archivos de Excel que se encuentran abiertos.
- El método Open, Save y SaveAs le indican a Excel si debe abrir, guardar o guardar como el workbook correspondiente. Ejemplos:

Application.Workbooks("ClaseIndustrial").Save Workbooks.Open("C:\Archivo.xls") Workbooks("Libro1").SaveAs("Archivo.xls",,"clavesecreta")

• Se pueden entregar los parámetros por nombre a los métodos. Ejemplos:

Workbooks.Open FileName :="C:\Archivo.xls", _ ReadOnly:=True, Password:="clavesecreta"

Application.Workbooks("ClaseIndustrial").Save



- El objeto **Worksheet** representa una hoja de cálculo Excel. El objeto **ActiveSheet** es un subobjeto del objeto Workbook que entrega el Worksheet activo.
- Se puede copiar, pegar, imprimir, guardar, activar y borrar la hoja de cálculo. Ejemplo:

With Application.Workbooks("ClaseIndustrial") ActiveSheet.Copy ActiveSheet.Paste ActiveSheet.PrintOut ActiveSheet.SaveAs ActiveSheet.Activate ActiveSheet.Delete

End With


- La colección **Worksheets** contiene a todas las hojas de cálculo que pertenecen a algún workbook.
- Se le puede dar un nombre a un worksheet en particular para referirse a él. Ejemplo:

Dim w As Workbook, s As Worksheet

Set w = Workbooks("Libro1")

Set s = w.Worksheets("Hoja1")

MsgBox s.Range("a1").Value

• Se pueden nombrar todas las hojas de un archivo usando el comando For Each ... Next Loop.

Sub MuestraNombres() Dim w As Worksheet For Each w In Worksheets MsgBox w.Name Next End Sub



- El objeto WorksheetFunction permite usar las funciones de Excel desde VBA.
- Para ver las funciones que existen (todas en inglés) se puede ocupar el Explorador de Objetos.
- Ejemplo:

```
area = WorksheetFunction.Pi * r ^ 2
a = WorksheetFunction.Acos(b)
Set TestRange = Worksheets("Hoja1").Range("A1:A5")
TestSum = WorksheetFunction.Sum(TestRange)
```



- El objeto Range representa rangos de celdas. También es posible acceder a las celdas usando la propiedad Cells de ActiveSheet.
- Ejemplos:

```
Set notas = Worksheets("Funciones").Range("F2:F13")
prom = WorksheetFunction.Average(notas)
Worksheets("Funciones").Range("F14").Value = prom
Worksheets("Funciones").Range("F15").Formula = "=average(F2:F13)"
Worksheets("Funciones").Cells(2, 1).Select
Workbooks("Libro1").Worksheets("Hoja1").Range("A1).Value = 10
Workbooks("Libro1").Worksheets("Hoja1").Range("A2.A10").Value = 5
Workbooks("Libro1").Worksheets("Hoja1").Range("A2", "A10").Value = 5
```



 Un módulo VBA se compone de procedimientos que son códigos de ordenador que realizan alguna acción sobre los objetos o con ellos.

```
Sub Prueba()
        Sum = 1+1
       MSGBox "La respuesta es" & Sum
End Sub
Sub Hola()
        Msg = "Su nombre es " & Application.UserName & "?"
       Ans = MsgBox(Msg, vbYesNo)
        If Ans = VbNo Then
               MsgBox "No se preocupe"
        Flse
               MsgBox "Debo ser adivino!"
        End If
End Sub
```



- Para ejecutar el procedimiento Hola es necesario asegurarse que el cursor está situado en cualquier parte del texto introducido.
- Después se puede optar por uno de los siguientes métodos:
 - Presionar F5
 - Seleccionar Ejecutar, Ejecutar Sub/UserForm
 - Hacer clic sobre el botón Ejecutar, Ejecutar Sub/UserForm



- Para escribir Subrutinas en VBA se debe crear un Módulo que las contenga (Insertar → Módulo)
- Ejemplo:

```
Public Sub MiSub(A as Integer, B as Double)
'Código de la subrutina
```

End Sub

- El alcance puede ser **Public** o **Private** que especifica si el procedimiento se puede llamar de cualquier parte o sólo desde ese módulo.
- La keyword Sub especifica una subrutina, que a diferencia de una función no devuelve un resultado.



Los parámetros se definen como:

(ByValue) NombreParam as Tipo

donde Tipo puede ser un tipo de dato, por ejemplo:

- Integer: un número entero
- Double: un número real
- Boolean: un número binario (V o F)
- Byte: guarda un valor entre 0 y 255
- String: Caracteres ASCII
- Para pasar sólo los valores usar ByValue que se considera por defecto.



- Las subrutinas se pueden llamar desde otras partes del código usando su nombre y agregando los parámetros que necesita.
- Para llamar a una subrutina llamada MiSub se puede usar: MiSub 4, 2.87

Call MiSub(4, 2.87)

• También se puede agregar el nombre de la subrutina a botones u otros controles de VBA.



 Las funciones son similares a las subrutinas con la diferencia que se usa Function en vez de Sub y que retornan un valor después de ejecutarse.

```
Public Function Calc_q(y1 As Double, y3 As Double) As Double
Calc_q = 1 / ((Abs(y3 - y1)) ^ 0.74)
End Function
```

• Se pueden usar como cualquier función de Excel.

```
Public Function MiFactorial(N As Integer) As Integer
'Funcion que calcula el factorial de un numero N
MiFactorial = 1
For i = 1 To N
MiFactorial = i * MiFactorial
Next
End Function
```



• Para comentar el código se usa ' o Rem

'Declaración de variables Dim y As Double Dim x As Double Rem Declaración de Matrices Dim M(1 To 8, 1 To 8) As Double Dim N(8, 8) As Double

• Para separar múltiples líneas se usa un guión bajo (_):

```
K2(1) = dt * dy1dt(y(1) + k1(1) / 2#, y(2) + _
k1(2) / 2#, y(3) + k1(3) / 2#, y(4) + _
k1(4) /2#)
```

Tiene que haber un espacio antes del underscore.



- Los datos manipulados en VBA residen en objetos (p.e. rangos de hojas de cálculo) o en variables que se crean.
- Una variable es una localización de almacenamiento con nombre, dentro de la memoria del ordenador. Debe tener asociado un tipo de dato.
- Las reglas para nombrar las variables son:
 - Se pueden usar caracteres alfabéticos, números y algún carácter de puntuación, pero el primero de los caracteres debe ser alfabético
 - VBA no distingue entre mayúsculas y minúsculas
 - No se pueden usar espacios ni puntos
 - No se pueden incrustar en el nombre de una variable los siguientes símbolos: #, \$, %, !
 - La longitud del nombre puede tener hasta 254 caracteres



Tipos de Datos en VBA

Tipo de dato	Bytes	Rango de valores
Byte	1	0 a 255
Boolean	2	True o False
Integer	2	-32768 a 32767
Long	4	- 2147483648 y 2147483647
Currency	8	-922337203685477.5808 a 922337203685477.5807
Single	4	-3.402823E38 a 3.402823E38
Double	8	-1.79769313486231E308 a 1.79769313486232E308
Date	8	1-1-100 al 31-12-9999 y horarios de 0:00:00 a 23:59:59
String		longitud variable (2 ³¹ caracteres). longitud fija (2 ¹⁶)
Object	4	
Variant		cualquier clase de datos excepto cadena de longitud fija



• Con **Dim** o **Public** se declaran las variables:

```
Dim b As Double, a As Double
Dim n, m As Integer
Dim InerestRate As Single
Dim TodaysDate As Date
Dim UserName As String * 20
Dim x As Integer, y As Integer, z As Integer
```

Si una variable no se declara se asume de tipo Variant (tipo genérico).

• En general debe ser:

Dim NombreVariable As DataType



• El ámbito de una variable determina el módulo y el procedimiento en el que se puede usar una variable.

Ámbito	Cómo se declara una variable en este ámbito
Un procedimiento	Incluye instrucciones Dim, Static o Private
	dentro del procedimiento.
Al nivel de módulo	Incluye una instrucción Dim antes del
	primer procedimiento de un módulo.
Todos los módulos	Incluye una instrucción Public antes del
	primer procedimiento de un módulo.

• Variables locales

Una variable local es una variable declarada dentro de un procedimiento. Estas variables se pueden usar sólo en el procedimiento en que se declararon. Cuando el procedimiento finaliza, la variable deja de existir y Excel libera su memoria.



 Un array es un grupo de elementos del mismo tipo que tienen un nombre común; para referirse a un elemento específico de un array se usa el nombre de la array y un número de índice. Los arrays se definen similar a las variables, pero con el tamaño:

Arrays de una dimensión

Dim y(1 To 4) As Double Dim x(4) As Double Dim MiArray(1 To 100) As Integer Dim MiArray (100)

Arrays multidimensionales

Dim M(1 To 8, 1 To 8) As Double Dim N(8, 8) As Double Dim MiMatriz(1 To 100, 1 to 10) As Integer



- Por defecto los subíndices de los arrays de VBA empiezan en 0. Si deseamos que comience en 1 en vez de en 0, incluiremos antes del primer array y antes del primer procedimiento la expresión:
 Option Base 1 o explícitamente el rango de elementos
- Para acceder a los elementos del array:

```
y(3) = 2.983
M(1, 2) = 4.321
MiArray(1) = 20
MiMatriz(1,2) = 20
```

• Si no se sabe el tamaño, se puede usar ReDim:

```
Dim Matriz() As Double
ReDim Matriz(10)
ReDim Preserve Matriz(12) 'Mantiene lo que estaba
```



• Con **Const** se declaran las constantes:

```
Const MiConstante As Integer = 14
Const MiConstante2 As Double = 1.025
Const NumTrim As Integer = 4
Const Interés = 0.05, Periodo = 12
Const Nombre Mod as String = "Macros Presupuestos"
Public Const NombreApp As String = "Aplicación Presupuestos"
```

- Las constantes también poseen un ámbito:
 - Si se declaran después de Sub o Function es local.
 - Si se declara al inicio de un módulo está disponible para todo el módulo.
 - Si se declara con Public al inicio de un módulo está disponible para todos los módulos de una hoja de trabajo.



• Constantes predeterminadas, que se pueden usar sin necesidad de declararlas.

- Cadenas, hay dos tipos de cadenas en VBA:
 - De longitud fija, que se declaran con un número específico de caracteres. La máxima longitud es de 65.536 caracteres.
 - De longitud variable, que teóricamente pueden tener hasta 2.000 millones de caracteres.

Dim MiCadena As String * 50 Dim SuCadena As String



Fechas y Expresiones

• Trabajar con Fechas

Dim Hoy As Date Dim HoraInicio As Date Const PrimerDía As Date = #1/1/2001# Const MedioDía As date = #12:00:00#

 Expresiones de asignación, expresión que realiza evaluaciones matemáticas y asigna el resultado a una variable o a un objeto. Se usa el signo igual "=" como operador de asignación.

```
x = 1
x = x + 1
x = (y * 2) / (z * 2)
FileOpen = true
Range("Año"). Value = 1995
```



OPERADORES ARITMÉTICOS

+ Suma, - Resta, * Multiplicación, / División, \ División entera, Mod Resto, ^ exponencial, & Concatenación

- OPERADORES COMPARATIVOS
- = Igual, < Menor, > Mayor, <= Menor o igual, >= Mayor o igual, <> Distinto
- OPERADORES LÓGICOS

Not (negación lógica, And (conjunción lógica), Or (disyunción lógica), XoR (exclusión lógica), Eqv (equivalencia en dos expresiones), Imp (implicación lógica)



- VBA ofrece dos estructuras que simplifican el trabajo con objetos y colecciones.
- Con WITH...END WITH se permite realizar múltiples operaciones en un solo objeto.

```
Sub CambiarFuente()
With Selection.Font
.Name = "Times New Roman"
.FontStyle = "Bold Italic"
.Size = 12
.Underline = xlSingle
.ColorIndex = 5
End With
End Sub
```



 Para una colección no es necesario saber la cantidad de elementos que existen en ella para usar la estructura For Each...Next.

```
Sub ContarHojas()
'Muestra el nombres de las hojas del libro de trabajo activo
       Dim Item As Worksheet
       For Each Item In ActiveWorkbook.Sheets
               MsgBox Item.Name
       Next Item
End Sub
Sub VentanasAbiertas()
'Cuenta el número de ventanas abiertas
       Suma = 0
       For Each Item In Windows
               Suma = Suma + 1
       Next Item
       MsgBox "Total de ventanas abiertas", & Suma
End Sub
```



• Los tests lógicos en VBA tienen la siguiente sintaxis:

```
If (time = 32000) Then
   MsgBox "time vale 32000"
End If
If (MiCondicion = True) Then
   MsgBox "Mi Condición es Verdad"
Else
   MsgBox "Mi Condición No es Verdad"
End If
If (contador < 10) Then
   MsgBox "El Contador es menor a 10"
ElseIf (contador < 20) Then
   MsgBox "El Contador es mayor que 10 y menor que 20"
ElseIf (contador < 30) Then
   MsgBox "El Contador es mayor que 20 y menor que 30"
End If
```



 La estructura Select Case es útil para elegir entre tres o más opciones

```
Sub Positivos_Negativos_Cero()
a = InputBox("Ingrese un número")
Select Case a
Case Is > 0
Msg = "Número Positivo"
Case Is < 0
Msg = "Número negativo"
Case Else
Msg = "Cero"
End Select
MsgBox Msg
End Sub</pre>
```



Esta sentencia de iteración se ejecuta un número determinado de veces. Su sintaxis es:

```
For contador = empezar To finalizar [Step valorincremento]
    [Instrucciones]
    [Exit For]
    [instrucciones]
Next [contador]
```

```
Sub SumaNúmeros

Sum = 0

For Count = 0 To 10

Sum = Sum + Count

Next Count

MsgBox Sum

End Sub
```



Bucles For...Next

For i = 1 To n 'Código
Next i
For i = 1 To n Step 2
'Código
Next i
For i = 1 To n
'Código
If tiempo >10 Then
Exit For
End If
'Más Código
Next i



• El bucle se ejecuta hasta que la condición llegue a ser verdadera. Do Until tiene la sintaxis.

```
Do Until [condicion]
[instrucciones]
[Exit Do]
[instrucciones]
```

Loop

```
Sub DoUntilDemo()
Do
ActiveCell.Value = 0
ActiveCell.Offset(1, 0).Select
Loop Until Not IsEmpty(ActiveCell)
End Sub
```





Do While (tiempo < 10) 'Código Loop

Do

'Código Loop While (tiempo < 10)</pre>

Do Until (tiempo > 10) 'Código Loop

Do 'Más Código Loop Until (tiempo > 10)

- Problema: Se requiere realizar operaciones con vectores como el producto vectorial o escalar.
- Ejemplo: vector examples.xlsm
- Excel no dispone de tales operaciones pero se pueden definir usando VBA.
- Ejemplo: para calcular la magnitud o módulo de un vector
 3d se define una función v_Mag a la que se pasa como parámetro el rango de las celdas, es decir su uso es:

=v_Mag(rango_celdas)

donde rango_celdas consta de tres celdas (x, y, z)



Public Function v_Mag(v As Range) As Double

Dim x As Double

Dim y As Double

Dim z As Double

 $v_Mag = Sqr(x ^ 2 + y ^ 2 + z ^ 2)$ End Function

- Ejemplo: para calcular el producto vectorial de dos vectores 3d se define una función v_CrossProduct a la que se pasa como parámetros dos rangos de las celdas.
- Como el resultado de esta función es un array, es necesario seleccionar el rango de celdas del resultado y teclear simultáneamente Ctrl – Mayúscula – Enter. Es decir su uso es:

$$\{ = v_CrossProduct(rango1, rango2) \}$$

$$u(x_1, y_1, z_1) = \begin{vmatrix} i & j & k \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix} u \times v = \left(\begin{vmatrix} y_1 & z_1 \\ y_2 & z_2 \end{vmatrix}, - \begin{vmatrix} x_1 & z_1 \\ x_2 & z_2 \end{vmatrix}, \begin{vmatrix} x_1 & y_1 \\ x_2 & z_2 \end{vmatrix} \right)$$

X Código para la función v_CrossProduct

```
Public Function v CrossProduct(u As Range, v As Range)
    ' Declaracion de variables locales:
    Dim ux As Double
    Dim uy As Double
    Dim uz As Double
    Dim vx As Double
    Dim vy As Double
    Dim vz As Double
    Dim ReturnArray(3)
    Dim DoTranspose As Boolean
    ' Determina si el rango de resultado seleccionado es un array fila o columna:
    If Application.Caller.Rows.Count > 1 Then
        DoTranspose = True
    Else
        DoTranspose = False
    Fnd Tf
    ' Obtiene los vectores componentes:
    ux = u.Cells(1).Value
    uy = u.Cells(2).Value
    uz = u.Cells(3).Value
```



vx = v.Cells(1).Value vy = v.Cells(2).Value vz = v.Cells(3).Value

' Calculo del producto vectorial: ReturnArray(0) = (uy * vz - uz * vy) ReturnArray(1) = (uz * vx - ux * vz) ReturnArray(2) = (ux * vy - uy * vx)

' Si el rango resultado seleccionado es una columna de celdas se transpone: If DoTranspose Then

v_CrossProduct = Application.WorksheetFunction.Transpose(ReturnArray)
Else

```
v_CrossProduct = ReturnArray
End If
```

End Function

Uso de función Excel



Para desplegar menú de depuración:
 Ver → Barras de Herramientas → Depuración

Para crear un breakpoint haga click en la zona que se

- encuentra al lado del código (donde está el punto rojo).
- Para poder ir paso a paso se puede usar F8 sobre un procedimiento.
- Se pueden ver las variables con las ventanas Locales, Inmediato e Inspección.
- Para imprimir en Inmediato: Debug.Print x



Depuración

🚰 Microsoft Visual Basic - controles_VBA.xls						
<u>A</u> rchivo <u>E</u> dición <u>V</u> er Ins	′tar Eormato Depuración Ejecuṯar Herramientas ⊆omplementos Ventana					
Ayuda						
ĨX 🔄 - 🖬 👗 🛤 🗠 ↔ → 👔 🖬 😫 😭 😤 🖓 📿 🛛 🐥						
Proyecto - VBAProject 🗙	🖇 controles_VBA.xls - Módulo1 (Código)					
	(General) CreaSerie	=				
🗄 😻 funcres (FUNCRES.XL						
🖃 😻 VBAProject (control	Sub Crease Depuración 🔹 🗸	ĸ				
🖃 😁 Microsoft Excel Obj						
Hoja1 (Controle		-				
Hoja2 (Ejemplo:	Macro grabada el 11/09/2009 por Pedro					
Hoja4 (Controle	· · · · · · · · · · · · · · · · · · ·					
Hoja5 (MacroBu	Acceso directo: CIRL+r					
Hoja6 (Macros)	ActiveColl FerryloD1C1 = 505					
ThisWorkbook	Bange (#12#) Salagt					
	hctiveCell FormuleP1C1 = "0.5"					
	Pange("11:101mutakici = 0.5					
H S UDADucie of (ellpse.x	Selection AutoFill Destination:=Pange("A1:A30	71 J				
Harroject (mielips	Range ("A1: A30"), Select	1				
	Selection.AutoFill Destination:=Range("A1:A31	71 J				
間 Hoja1 (Hoja1)	Range("A1:A31").Select					
Hoja2 (Hoja2)	End Sub					
ThisWorkbook						
	mediato	Y				
Il UserForm1		읙				
		-				
<						



- Se puede agregar **Formularios** a las hojas de cálculo para realizar algunas tareas específicas y como GUI.
- Para insertar un formulario ir a Insertar \rightarrow UserForm.
- Se le pueden agregar controles a los formularios para darles distintas funcionalidades.
- Para mostrar un Formulario primero se diseña con un nombre (p.e. UserForm1) y después se usa el método Show. Para esconderlo se usa el método Hide. Ejemplo:

Sub MostrarForm()

UserForm1.Show

End Sub


Formularios

	🚈 Microsoft Visual Basic - mielipse.xls
	Archivo Edición ⊻er Insertar Eormato Depuración Ejecutar Herramientas ⊆omplementos Ventana Ayuda
Insertar Formato Depur Modulo Modulo Modulo Modulo Modulo Archivo Modulo Modulo	Ayuda Normal Andrew An
	Propiedades - UserForm X UserForm1 UserForm Alfabética Por categorías (Name) UserForm1 Alfabética Por categorías (Name) UserForm1 Alfabética Por categorías SackColor & H8(V BorderColor & H8(000) BorderStyle 0 - fmBorder V Costion UserForm1 V



- Se le puede insertar código a cada uno de los objetos que componen el formulario.
- Presionando F7 se muestra el código detrás del formulario.
- En el ComboBox de la izquierda (el que dice CommandButton1) se muestran los objetos que componen el formulario.
- En el ComboBox de la derecha (el que dice Click) se muestran los eventos que pueden ocurrir para cada objeto donde se puede insertar código.



Formularios





- Controles_VBA.xlsx, donde se muestran controles y macros.
- primos.xlsm, para calcular números primos y descomposición en factores primos
- lissajous.xlsm, donde se genera curvas lissajous con vba.
- proyectil.xlsm, que muestra una aplicación para el cálculo de la trayectoria de un proyectil.
- formularios.xlsm, muestra el uso de formularios con VBA.
- Ecuaciones_VBA.xls, que muestra la invocación de funciones Excel desde VBA.



- En cualquier programa MS Office se puede agregar **controles** que pueden interactuar con los datos y con otros controles.
- Se utiliza la barra de controles para insertarlos y manipularlos.
- Cada objeto tiene asociado una serie de propiedades que se pueden modificar y se les puede asociar una macro o programa VBA.
- Ejemplo: controles_VBA.xls



Controles - Diseño

⊻er	Insertar Eormato Herrami	ientas	Datos FlashPaper Ventana <u>?</u> Adobe P	21	ontroles V	/RA		
▦	<u>N</u> ormal		- ⊂ - 🍓 Σ - 🔂 ≜↓ Z↓ 🕍 🛷		B		D	E E G
	Vist <u>a</u> previa de salto de página		厨 爱 £ % 00 t& % 住 住 厚	10				
	Panel de tareas			11		<u> </u>	Propiedades	×
	Barras de berramientas	•	Earbios Terminar revision	12	0 0	n Rala asián	Botón CommandBu	tton 🗾
	Parra da fármulas			13	o Suma	ir Selección O	Alfabética Por cat	egorías
	barra de Lornidias		Formato	14	0	0 0	(Name)	Botón
~	Barra de es <u>t</u> ado		Auditoría de fórmulas	15	Eu ▼ X	1	Accelerator	
	En <u>c</u> abezado y pie de página		Bordes	17		ra de Decelazar	AutoLoad	False
83 5 .	Comentarios		Quadro de controles	18		la de Desplazan	AutoSize	False
1000				19	Q.		BackColor	8H800000F&
	<u>V</u> istas personalizadas		Datos externos	20	S 🔽 🔤	Cosono	BackStyle	1 - fmBackStyleOpaque
	Pantalla completa		V Dibujo .	20	3 1 101		Caption	Sumar Selección
-			Formularios	21			Enabled	
	<u>Z</u> oom		Formularios	22		5 -0.49026082	Font	
				23		7 -0.51928865	ForeColor	■ &H80000012&
	Cu 🔻 🗙	X	Cortar	24		0.99943459	Height	31.5
				25	- 🗄 A	1 -0.46067859	Lerc	
		Ľ۵	⊆opiar	26	-	4 -0.54772926		(Ningupo)
		<u>م</u>	Pegar	27	- 43.	4 0.99773898	MousePointer	0 - fmMousePointerDefault
	Modo Diseno	-	<u>r</u> egai	28	- ~	17 -0.4305754	Picture	(Ninguno)
	I abl	r 😭	Propiedades	29	-0.373052	77 -0.57555048	PicturePosition	7 - fmPicturePositionAboveCenter
	= 0		r	30	0.047704	0.99491511	Placement	2
		S.	<u>V</u> er código	31	0.0177013	93 -0.39998531	PrintObject	True
			Obiete Batés de comando A	32	0.215967.	27 -0.60272065	Shadow	False
	큰 좀			33	0.405661	88 0.99096615	TakeFocusOnClick	True
			Agrupar 🕨	34	0.579164	03 -0.36894291	Тор	143.25
	∄ A		ng <u>c</u> apa, v	35	0.729576	74 -0.62920964	Visible	True
	2		Ordenar 🕨 🕨	36	0.850903	62 U.98689668	Width	108
	2.00	8		37	0.938307	47 -0.33748329	WordWrap	False
	*	S.	Formato de co <u>n</u> trol	38	0.988304	06 -0.65498691		
	*	-		39	0.998900	09 0.97971213		



Tipos de Controles – de formulario y **ActiveX**

Cu 🔻 🗙
👱 🔗
ą
🗹 🖬
2
🗄 A
<u>~</u>
×

- Casilla de Verificación •
- Botón de Comando •
- Cuadro Combinado •
- Botón o Control de Número
- Slider •
- Etiqueta •
- Ejemplo Texto CheckBox1 Botón Opción Botón Fútbol Fútbol • Baloncesto Tenis ToggleButton1 Etiqueta

- Cuadro de Texto
- Botón de Opción
- Cuadro de Lista
- Botón de Alternar
- Etiqueta



- Las **macros** de Excel permite automatizar tareas repetitivas utilizando un grabador de macros.
- Mientras se graba la macro, todas las acciones que se realicen en Excel quedan guardadas en un módulo asociado al libro en lenguaje Visual Basic para Aplicaciones (VBA).
- Se puede crear la macro con referencia absoluta o referencia relativa.
- Las macros se pueden ejecutar directamente o con combinación de teclas. Para ver las macros pulsar Alt + F8.



- Un modo de crear una macro es grabar la macro.
 - En el menú Herramientas→Macro →Grabar nueva macro.
- Una vez grabada la macro, se puede examinar el código generado en el Editor de Visual Basic para ver cómo funciona el código.
 - En el menú Herramientas →Macro→Macros para buscar la macro grabada. Seleccionarla y hacer clic en Editar para ver el código.



Macros - Diseño

Her	ramientas <u>P</u>	resentación	Flash <u>P</u> aper	Ve <u>n</u> t
ABC	Ortografía F7		=7	
13	Re <u>f</u> erencia		Alt+C	lic
	Sinónim <u>o</u> s		Mayús+F	=7
	I <u>d</u> ioma			
	Á <u>r</u> ea de trab	ajo compartić	la	
	Comparar y combinar presentaciones			
	C <u>o</u> laboración en línea			•
	<u>M</u> acro			►
	Compl <u>e</u> mento	os		
3	Opcione <u>s</u> de	Autocorrecció	ón	
	Personali <u>z</u> ar.			
	Opcio <u>n</u> es			



De 🔻 🗙



- El Editor de Visual Basic es una herramienta para escribir y modificar código escrito en VBA
- Para abrir el Editor de Visual Basic: En el menú Herramientas → Macro → Editor de Visual Basic o Alt+F11.
- Las macros se almacenan en módulos de un libro de trabajo.
- Los módulos se agregan en el Editor de Visual Basic seleccionando Módulo en el menú Insertar del editor.
- Debe aparecer una ventana de módulo vacía dentro de la ventana principal del Editor de Visual Basic.



Macros - Editor VB

Herra	mientas Da <u>t</u> os	; Fla	shĘ						
	Proteger								
	Eu <u>r</u> oconversión								
	Sol <u>v</u> er								
	B <u>u</u> scar objetivo								
	<u>M</u> acro		Þ	•	Macros	Microsoft Visual Bas	ic -	- controles_VBA.xls	4
	Compl <u>e</u> mentos			•	<u>G</u> rabar nueva macro	Archivo Edicion Ver	Inse	isertar Eormato Depuración Ejecutar Herramientas Complementos Ventana Ayuda	
	Opcio <u>n</u> es				<u>S</u> eguridad	Proyecto - VBAProje 🗙 👖	00		
	Análisis de datos.		4		Editor de Visual Basic 🛛 Al			ś controles_VBA.xls - Módulo1 (Código)	
						Solution functions (FUNCRI WBAProject (cor Wapping Hoja1 (Cor Hoja2 (Eje Hoja4 (Cor Hoja5 (Hoj ThisWorkb Módulos Módulo1		<pre>Sub CreaSerie() ' CreaSerie Macro ' Macro grabada el 11/09/2009 por Pedro ' Acceso directo: CTRL+r ' ActiveCell.FormulaR1C1 = "0" Range("A2").Select ActiveCell.FormulaR1C1 = "0.5" Range("A1:A2").Select Selection.AutoFill Destination:=Range("A1:A30"), Type:=xlFillDefault Range("A1:A30").Select Selection.AutoFill Destination:=Range("A1:A31"), Type:=xlFillDefault Range("A1:A31").Select End Sub</pre>	
							=		



- En la ventana del módulo se escribe la palabra Sub seguido del nombre de la macro.
- Por ejemplo, Sub MiMacro, crearía una nueva macro Ilamada MiMacro.
- El Editor de Visual Basic insertará automáticamente los paréntesis y End Sub.

🚜 Libro 1- Module 1 (Código) 🛛 🔲 🔀					
(9	general)	•	MiMacro	-	
	Sub MiMacro ()	l			
	End Sub				



- Si se desea mostrar un mensaje simple escribir MsgBox "Mi primera macro".
- MsgBox es la palabra que VBA utiliza para los cuadros de mensaje.
- Si se ejecuta la macro, Excel mostraría un mensaje con el texto Mi primera macro y un botón Aceptar para cerrar el mensaje.

Excel





- Las macros de bucle funcionan recorriendo los datos de celdas para realizar acciones automáticamente de manera repetida.
- Hay varias instrucciones que permiten crear este tipo de macros:
 - For Each...Next
 - For ... Next
 - For ... Next Loop With Step
 - Do While ... Loop
 - Do Until ... Loop
 - Do ... Loop While
 - Do ... Loop Until



- For Each...Next se usa para realizar una acción en cada celda de un grupo de datos.
- Ejemplo: Código para que la palabra "Aceptar" aparezca en negrita en cualquier lugar del grupo de datos seleccionado

```
Sub MacroBucle()
For Each MyCell In Selection
If MyCell.Value Like "Aceptar" Then
MyCell.Font.Bold = True
End If
Next
End Sub
```



- Hay dos métodos para llevar los datos de una celda a código VBA: propiedad Cells y Range.
- La propiedad Range identifica las filas y columnas mediante los números y las letras de la hoja de cálculo.
- La propiedad Cells utiliza números para las filas y las columnas (Herramientas → Opciones → General → "Estilo de referencia F1C1")



Ejemplos de Macros

• Ejemplo1: Macro que usa Do While...Loop para combinar dos columnas en otra.

Pepelu	Rodriguez	Pepelu Rodriguez
lker	Casillas	Iker Casillas
Cristiano	Ronaldo	Cristiano Ronaldo
Jaime	Peñafiel	Jaime Peñafiel
Teresa	Berlanga	Teresa Berlanga
Ainoha	Arteta	Ainoha Arteta
Pedro	Mamani	Pedro Mamani

```
Sub EjMacro1()
'x empieza en 1
x = 1
'Seguir hasta encontrar una fila vacia
Do While Cells(x, 1).Value <> ""
'Colaca los valores delas columnas
'1 y 2 juntas con un espacio entre ellas
'en la 3 columna
Cells(x, 3).Value = Cells(x, 1).Value + " "
+ Cells(x, 2).Value
x = x + 1
Loop
End Sub
```



Ejemplos de Macros

• Ejemplo2: Bucle For Each...Next lee cada celda y le aplica color según el contenido.

		Sub EjMacro2()
	1.11	'Indica a VBA que la variable MyCell es un rango
LIDIO	Libro	Dim MiCelda As Range
Pelicula	Pelicula	For Each MiCelda In Selection
		lf MiCelda.value Like "*Libro*" Then
Revista	Revista	'Define el color de fondo de la celda en rojo Miceldo Interior CelerIndex = 2
laglibra	Loc Libro	FigeIf MiCelde Velue Like "*Deligule*" Then
		Define el color de fondo de la celda en verde
Ver pelicula	Ver pelicula	MiCelda.Interior.ColorIndex = 4
\/inc		ElseIf MiCelda.Value Like "" Then
VINO	VIIIO	'Borra el color de fondo de la celda
Texto	Texto	MiCelda.Interior.ColorIndex = xlNone
ΓΟΛΙΟ		Else
Libro texto	Libro texto	'Define el color de fondo de la celda en azul
		MiCelda.Interior.ColorIndex = 5
		End If
		Next
		End Sub



Ejemplos de Macros

 Ejemplo3: Elimina filas que contienen duplicados en las columnas D y F.

Elemento 1 Azul Elemento 2 Verde Elemento 1 Rojo Elemento 2 Mediano En existencia Sin existencias En existencia Sin existencias

```
Sub EjMacro3()
'Quita duplicados - empieza en la celda seleccionada
x = ActiveCell.Row
v = x + 1
Do While Cells(x, 4).Value <> ""
  Do While Cells(v, 4).Value <> ""
  'Comprueba duplicados: Si en la 4ta columna D y
  'la 6ta columna F coinciden en dos filas, borrar la
  'segunda del par sino seguir con la siguiente
  If (Cells(x, 4).Value = Cells(y, 4).Value)
      And (Cells(x, 6).Value = Cells(y, 6).Value) Then
      Cells(y, 4).EntireRow.Delete
  Else
    v = v + 1
  End If
Loop
x = x + 1
v = x + 1
Loop
End Sub
```



Ecuaciones diferenciales ordinarias

X Ecuaciones diferenciales de primer orden

- Hay varios métodos disponibles para resolver numéricamente ecuaciones diferenciales.
- Los métodos de solución más utilizados para
 - Problemas de valor inicial son: Euler, Euler mejorado, Runge Kutta.
 - Problemas de contorno: método de disparo, método de diferencias finitas, extrapolación Richardson
- Ejemplo: ec_diferenciales.xlsx

X Ecuaciones diferenciales de primer orden y valor inicial

- Problema: se requiere hallar la solución de una ecuación diferencial de primer orden de la forma:
- $\frac{dy}{dx} = f(x, y)$ $y = e^{x} - x - 1$ y(0) = a• Ejemplo: $\frac{dy}{dx} = x + y$ 07 0.6 0.5 04 y(0) = 00.3 0.2 Solución: $y = e^x - x - 1$ 0.1 0
- Usaremos esta ecuación para comparar

0.8

0.4

0.6



- No es el mejor método (requiere un tamaño de paso pequeño para mejorar la precisión), pero es simple para mostrar la mecánica de su implementación en Excel y VBA.
- El método de Euler se basa en considerar los dos primeros términos de la expansión en Serie de Taylor de una función para predecir el valor de la función en algún punto, conociendo el valor de la función en otro punto y la derivada de esa función.



• La expansión en series de Taylor de y es:

$$y(x + \Delta x) = y(x) + \Delta x \frac{dy}{dx} + \frac{(\Delta x)^2}{2!} \frac{d^2 y}{dx^2} + \frac{(\Delta x)^3}{3!} \frac{d^3 y}{dx^3} \cdots$$

- El método de Euler considera los dos primeros términos de la expansion, descartando los términos de orden superior que constituyen el error de truncamiento del método.
- Se conoce *dy/dx* (la ecuación diferencial) y se calcula y para valores de x empezando en el valor inicial.



- La expansión Taylor es fácil de implementar en una hoja o haciendo uso de VBA para hacerlo más versátil.
- Usando solo hoja de cálculo:
 - La solución se implementa como una tabla con una columna x en la que se calcula en cada fila un valor según el valor anterior más el paso. La columna y contiene la fórmula de Euler para el siguiente valor de y, según:

$$y(x + \Delta x) = y(x) + \Delta x \frac{dy}{dx}$$



- El error disminuye conforme disminuye el tamaño de paso para la variable paso x. Esto significa que se necesitan más cálculos con un paso pequeño para cubrir un rango equivalente en x. La cantidad de filas necesarias para pasos pequeños crece bastante, siendo engorroso si se quiere graficar los resultados, cambiar el rango o cambiar el tamaño de paso. En ese caso es mejor usar VBA.
- Usando VBA:
 - La subrutina diseñada calcula los valores estimados de y para un rango de valores x y coloca los resultados en la hoja a partir de la fila 2 en las columnas A y B



Método de Euler

Código:

Public Sub DoEuler1stOrder()

Dim yn As Double Dim yn1 As Double Dim xn As Double Dim dx As Double Dim n As Integer yn = 0xn = 0dx = 0.001n = 1000

```
For i = 1 To n

yn1 = yn + (xn + yn) * dx

xn = xn + dx

yn = yn1

ActiveSheet.Cells(i + 1, 1) = xn

ActiveSheet.Cells(i + 1, 2) = yn

Next i

End Sub
```

El problema con el código anterior es el mismo que se mencionó para la hoja, el número de filas puede ser muy grande cuando se usa un paso pequeño.



Se puede añadir una condición para que imprima los resultados cada cierta cantidad de pasos (p.e. 100) con el fin de mantener un número de filas de resultados manejable.

Public Sub DoEuler1stOrder()

Dim yn, yn1, xn, dx As Double Dim n, C, k As Integer

yn = 0 xn = 0 dx = 0.0001 n = 11000 C = n / 10 k = 1

```
For i = 1 To n
    yn1 = yn + (xn + yn) * dx
    xn = xn + dx
    yn = yn1
    If C >= (n / 10) Then
       ActiveSheet.Cells(k + 1, 1) = xn
       ActiveSheet.Cells(k + 1, 2) = yn
       k = k + 1
       C = 0
     Else
       C = C + 1
     End If
  Next i
End Sub
```



• Problema: se requiere resolver numéricamente la ecuación diferencial de segundo orden de la forma:

$$A\frac{d^2y}{dx^2} + B\frac{dy}{dx} = f(x, y)$$
$$y(0) = \propto$$
$$\frac{dy}{dx}(0) = \beta$$

 Se puede utilizar cualquier técnica de integración numérica usando Excel o VBA



Considerar la siguiente ecuación y condiciones iniciales:

$$m\frac{d^{2}s}{dt^{2}} + C_{d}\frac{ds}{dt} = T$$
$$s(0) = 0$$
$$\frac{ds}{dt}(0) = 0$$

 Físicamente representa la ecuación del movimiento de un objeto sujeto a un empuje *T. m* es la masa, *C_d* un factor de rozamiento y s la posición del objeto.



 Para resolver la ecuación de movimiento se reescribe para obtener dos ecuaciones de primer orden:

si hacemos:
$$v = \frac{ds}{dt}$$

 $m \frac{dv}{dt} = T - C_d v$
 $\frac{ds}{dt} = v$
 $s_{t=0} = 0$
 $v_{t=0} = 0$

• Se obtiene dos ecuaciones de primer orden acopladas, a las que se aplican técnicas numéricas.

Método Runge-Kutta aplicado a problemas de valor inicial de 2do orden

- El método de Runge Kutta se basa en tomar más términos de la serie de Taylor de la función, que se traduce en expandir más series de Taylor para estimar las derivadas de mayor orden.
- El enfoque RK reduce el error de truncamiento a un orden de (*dt*)⁵ en oposición a (*dt*)² del método de Euler, con lo que se puede aumentar el paso manteniendo la precisión.
- El compromiso es que hay que realizer más cálculos en cada paso.



• Las ecuaciones generales de Runge Kutta para la integración son:

$$k_{1} = y'(x, y)(\Delta x)$$

$$k_{2} = y'(x + \frac{\Delta x}{2}, y + \frac{k_{1}}{2})(\Delta x)$$

$$k_{3} = y'(x + \frac{\Delta x}{2}, y + \frac{k_{2}}{2})(\Delta x)$$

$$k_{4} = y'(x + \Delta x, y + k_{3})(\Delta x)$$

$$y(x + \Delta x) = y(x) + \frac{(k_{1} + 2k_{2} + 2k_{3} + k_{4})}{6}$$
donde: y' representa dy/dx

Método Runge-Kutta aplicado a problemas de valor inicial de 2do orden

	For i = 1 To n 'Start iterations
Public Sub DoRK2ndOrder()	F = (t - (Cd * Vn)) ' Compute k1
Dim t, Cd, M, dt As Double ' Thrust, Drag coefficient, Mass	A = F / M
Dim dt, F, A As Double 'Time step size, Force, Acceleration	k1 = dt * A
Dim Vn As Double 'Velocity at time t	F = (t - (Cd * (Vn + k1 / 2))) ' Compute k2
Dim Vn1 As Double 'Velocity at time t + dt	A = F / M
Dim Sn As Double 'Displacement at time t	k2 = dt * A
Dim Sn1 As Double 'Displacement at time t + dt	F = (t - (Cd * (Vn + k2 / 2))) ' Compute k3
Dim time As Double ' Total time	A = F / M
Dim k1, k2, k3, k4 As Double 'RK k1, RK k2, RK k3, RK k4	k3 = dt * A
Dim n As Integer 'Counter controlling total number of time steps	F = (t - (Cd * (Vn + k3))) ' Compute k4
Dim C As Integer ' Counter controlling output of results to spreadsheet	A = F / M
Dim k As Integer ' Counter controlling output row	k4 = dt * A
Dim r As Integer 'Number of output rows	Vn1 = Vn + (k1 + 2 * k2 + 2 * k3 + k4) / 6 'Compute velocity at t + dt
With ActiveSheet 'Extract given data from the active spreadsheet:	Sn1 = Sn + Vn1 * dt 'Compute displacement at t + dt using Euler
dt = .Range("dt")	time = time + dt
t = .Range("T")	Vn = Vn1
M = .Range("M")	Sn = Sn1
Cd = .Range("Cd")	If $C \ge n / r$ Then 'Output results to the active spreadsheet
n = .Range("n")	ActiveSheet.Cells(k + 1, 1) = time
r = .Range("r_")	ActiveSheet.Cells(k + 1, 2) = Sn
End With	ActiveSheet.Cells(k + 1, 3) = Vn
k = 1 'Initialize variables	k = k + 1
time = 0	C = 0
C = n / r	Else
Vn = 0	C = C + 1
Sn = 0	End If
	Nexti
	End Sub

Ecuaciones diferenciales con condiciones de contorno o de frontera

 Hay problemas que se modelizan mediante una ecuación diferencial de segundo orden con condiciones en sus dos extremos [a, b], que se denomina ecuación diferencial ordinaria con valores en la frontera o contorno. Se formula como:

 $y'' = f(t, y, y'), \quad y(a) = \alpha, \quad y(b) = \beta$

- Hay varios métodos para hallar la solución:
 - Método del disparo (shooting method)
 - Métodos de las diferencias finitas
 - Método de colocación y de elementos finitos
• Se requiere resolver la siguiente ecuación diferencial ordinaria con condiciones de frontera:

$$\frac{d^2 u}{dt^2} = 2t \quad u(0) = 0, \ u(1) = 1 \quad 0 < t < 1$$

es una ecuación de segundo orden sujeta a condiciones de contorno

• Si hacemos: $y_2 = u y y_1 = du/dt = dy_2/dt$ para reducir la ecuación de segundo orden en dos de primer orden:

$$\frac{dy_2}{dt} = y_1, \frac{dy_1}{dt} = 2t \quad y_2(0) = 0 \quad y_2(1) = 1 \quad 0 < t < 1$$



- Se basa en encontrar la ecuación diferencial que tiene la misma solución que la del contorno que se estudia.
- Para ello se resuelve como un problema de valor inicial usando un algoritmo de avance como Euler o Runge Kutta, asumiendo un valor inicial para las condiciones iniciales desconocidas.
- Cuando el proceso se completa se verifica si los valores obtenidos satisfacen las condiciones de contorno. Sino se repite con otros valores.



Método de disparo

- El proceso iterativo es ideal para Solver.
- Todo el proceso se realiza combinando Solver y el método de Euler.
- En Solver solo se especifica la celda variable y la condición que debe cumplirse (cond. frontera)

arámetros de Solver			×
Es <u>t</u> ablecer objetivo:			
Para: 💿 <u>M</u> áx 💿 Mír	n () <u>V</u> alor de:	0	
Cambiando <u>l</u> as celdas de variable	s:		
\$D\$3			
S <u>uj</u> eto a las restricciones:			
\$E\$112 = \$D\$6		× [<u>A</u> gregar
		(<u>C</u> ambiar
		[<u>E</u> liminar
		[<u>R</u> establecer todo
		-	<u>C</u> argar/Guardar
Convertir variables sin restrice	iones en no negativas		
Método d <u>e</u> resolución:	GRG Nonlinear	•	O <u>p</u> ciones
Método de resolución Seleccione el motor GRG Nonlir el motor LP Simplex para proble problemas de Solver no suaviza	ear para problemas de So mas de Solver lineales, y s dos.	olver no lineales su eleccione el motor	avizados. Seleccione Evolutionary para
Ayuda		<u>R</u> esolver	Cerrar



Ecuaciones diferenciales en derivadas parciales

Ordinary Differential Equations have only one independent variable

$$3\frac{dy}{dx} + 5y^2 = 3e^{-x} , y(0) = 5$$

Partial Differential Equations have more than one independent variable

$$3\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = x^2 + y^2$$

 subject to certain conditions: where u is the dependent variable, and x and y are the independent variables.



Clasificación de EDPs de 2 orden

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D = 0$$

• where *A*, *B*, and *C* are functions of *x* and *y*, and *D* is a function of

$$x, y, u$$
 and $\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}$.

• can be: Elliptic if $B^2 - 4AC < 0$ Parabolic if $B^2 - 4AC = 0$ Hyperbolic if $B^2 - 4AC = 0$



Ejemplos de EDPs de 2 orden

• Elliptic A = 1, B = 0, C = 1 $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$

Laplace equation

- Parabolic A = k, B = 0, C = 0 $\frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}$ Heat equation
- Hyperbolic $A = 1, B = 0, C = -\frac{1}{c^2}$ $\frac{\partial^2 y}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 y}{\partial t^2}$ Wave equation

Ejemplo físico de una PDE elíptica



- Schematic diagram of a plate with specified temperature boundary conditions $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial v^2} = 0$
- The Laplace equation governs the temperature:



Discretizando la PDE elíptica



$$\frac{\partial^2 T}{\partial x^2}(x, y) \cong \frac{T(x + \Delta x, y) - 2T(x, y) + T(x - \Delta x, y)}{(\Delta x)^2}$$
$$\frac{\partial^2 T}{\partial x^2} = \frac{T(x, y + \Delta y) - 2T(x, y) + T(x, y - \Delta y)}{(\Delta x)^2}$$

$$\frac{\partial^2 T}{\partial y^2}(x, y) \cong \frac{T(x, y + \Delta y) - 2T(x, y) + T(x, y - \Delta y)}{(\Delta y)^2}$$



Discretizando la PDE elíptica





Discretizando la PDE elíptica

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

• Substituting these approximations into the Laplace equation yields:

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} = 0$$

- if, $\Delta x = \Delta y$
- the Laplace equation can be rewritten as

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = 0$$
 (Eq. 1)

there are several numerical methods that can be used to solve the problem:

Direct Method

Gauss-Seidel Method

Lieberman Method



• Consider a plate $2.4 m \times 3.0 m$ that is subjected to the boundary conditions shown below. Find the temperature at the interior nodes using a square grid with a length of 0.6 m by using the direct method.





Ejemplo 1: Método directo

• We discretize the plate by taking, $\Delta x = \Delta y = 0.6m$

$$m = \frac{L}{\Delta x} = 4$$
 $n = \frac{W}{\Delta y} = 5$



• The nodal temperatures at the boundary nodes are given by:





• the equation for the temperature at the node (2,3)





• We can develop similar equations for every interior node leaving us with an equal number of equations and unknowns.



• For this problem the number of equations generated is 12



- The corner nodal temperature of $T_{0,5}, T_{4,5}, T_{4,0}, T_{0,0}$ are not needed
- To get the temperature at the interior nodes we have to write Equation 1 for all the combinations of *i* and *j*, *i* = 1,...,*m*−1; *j* = 1,...,*n*−1

i=1 and *j*=1 $-4T_{11} + T_{12} + T_{21} = -125$ *i*=1 and *j*=2 $T_{1,1} - 4T_{1,2} + T_{1,3} + T_{2,2} = -75$ *i*=1 and *j*=3 $T_{1,2} - 4T_{1,3} + T_{1,4} + T_{2,3} = -75$ $T_{1,3} - 4T_{1,4} + T_{2,4} = -375$ i=1 and j=4 $T_{11} - 4T_{21} + T_{22} + T_{31} = -50$ *i*=2 and *j*=1 $T_{12} + T_{21} - 4T_{22} + T_{23} + T_{32} = 0$ *i*=2 and *j*=2 *i*=2 and *j*=3 $T_{13} + T_{22} - 4T_{23} + T_{24} + T_{33} = 0$ $T_{14} + T_{23} - 4T_{24} + T_{34} = -300$ *i*=2 and *j*=4 $T_{21} - 4T_{31} + T_{32} = -150$ *i*=3 and *j*=1 $T_{22} + T_{31} - 4T_{32} + T_{33} = -100$ i=3 and j=2 $T_{2,3} + T_{3,2} - 4T_{3,3} + T_{3,4} = -100$ i=3 and j=3 $T_{24} + T_{33} - 4T_{34} = -400$ *i*=3 and *j*=4



Ejemplo 1: Método directo

• We can use Excel and matrix operations to solve the linear equations system

T _{1,1}	T _{1,2}	T _{1,3}	T _{1,4}	T _{2,1}	T _{2,2}	T _{2,3}	T _{2,4}	Т _{3,1}	T _{3,2}	Т _{3,3}	T _{3,4}	RHE			
-4	1	0	0	1	0	0	0	0	0	0	0	-125	-	Т _{1,1}	74.8719
1	-4	1	0	0	1	0	0	0	0	0	0	-75	-	Т _{1,2}	95.8959
0	1	-4	1	0	0	1	0	0	0	0	0	-75	-	Т _{1,3}	127.8036
0	0	1	-4	1	0	0	1	0	0	0	0	-375	-	T _{1,4}	196.9288
1	0	0	0	-4	1	0	0	1	0	0	0	-50	-	T _{2,1}	78.5917
0	1	0	0	1	-4	1	0	0	1	0	0	0	-	T _{2,2}	105.9082
0	0	1	0	0	1	-4	1	0	0	1	0	0	-	T _{2,3}	143.3896
0	0	0	1	0	0	1	-4	0	0	0	1	-300	-	T _{2,4}	206.3200
0	0	0	0	1	0	0	0	-4	1	0	0	-150	-	T _{3,1}	83.5868
0	0	0	0	0	1	0	0	1	-4	1	0	-100	-	T _{3,2}	105.7554
0	0	0	0	0	0	1	0	0	1	-4	1	-100	-	Т _{3,3}	133.5267
0	0	0	0	0	0	0	1	0	0	1	-4	-400	-	T ₂₄	184.9617

	300.0	300.0	300.0	
75.0	196.9	206.3	185.0	100.0
75.0	127.8	143.4	133.5	100.0
75.0	95.9	105.9	105.8	100.0
75.0	74.9	78.6	83.6	100.0
	50.0	50.0	50.0	



• Recall the discretized equation

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = 0$$

• This can be rewritten as

$$T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{4}$$

 For the Gauss-Seidel Method, this equation is solved iteratively for all interior nodes until a pre-specified tolerance is met.



• Consider a plate $2.4 m \times 3.0 m$ that is subjected to the boundary conditions shown below. Find the temperature at the interior nodes using a square grid with a length of 0.6 m using the Gauss-Siedel method. Assume the initial temperature at all interior nodes to be $0^{\circ}C$.



• Discretizing the plate by taking, $\Delta x = \Delta y = 0.6m$

$$m = \frac{L}{\Delta x} = 4$$
 $n = \frac{W}{\Delta y} = 5$



• The nodal temperatures at the boundary nodes are given by:





- Now we can begin to solve for the temperature at each interior node using $T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{4}, i = 1,2,3,4; j = 1,2,3,4,5$
- Assume all internal nodes to have an initial temperature of zero.
- Iteration 1:

$$i=1 \text{ and } j=1$$
 $T_{1,1} = 31.25 \,^{\circ}C$ $i=2 \text{ and } j=3$ $T_{2,3} = 9.27735 \,^{\circ}C$ $i=1 \text{ and } j=2$ $T_{1,2} = 26.5625 \,^{\circ}C$ $i=2 \text{ and } j=4$ $T_{2,4} = 102.344 \,^{\circ}C$ $i=1 \text{ and } j=3$ $T_{1,3} = 25.3906 \,^{\circ}C$ $i=3 \text{ and } j=1$ $T_{3,1} = 42.5781 \,^{\circ}C$ $i=1 \text{ and } j=4$ $T_{1,4} = 100.098 \,^{\circ}C$ $i=3 \text{ and } j=2$ $T_{3,2} = 38.5742 \,^{\circ}C$ $i=2 \text{ and } j=1$ $T_{2,1} = 20.3125 \,^{\circ}C$ $i=3 \text{ and } j=3$ $T_{3,3} = 36.9629 \,^{\circ}C$ $i=2 \text{ and } j=2$ $T_{2,2} = 11.7188 \,^{\circ}C$ $i=3 \text{ and } j=4$ $T_{3,4} = 134.827 \,^{\circ}C$

Iteration 2: we take the temperatures from iteration 1 and calculate the approximated error. $|\mathcal{E}_a|_{i,j} = \left| \frac{T_{i,j}^{present} - T_{i,j}^{previous}}{T_{i,j}^{present}} \right| \times 100$ *i*=2, *j*=3 $T_{2,3} = 56.4881^{\circ}C |\varepsilon_a|_{2,3} = 83.58\%$ *i*=1, *j*=1 $T_{1,1} = 42.9688 \,^{\circ} C |\varepsilon_a|_{1,1} = 27.27\%$ *i*=1, *j*=2 $T_{1,2} = 38.7596 \,{}^{\circ}C |\varepsilon_a|_{1,2} = 31.49\%$ *i*=2, *j*=4 $T_{24} = 156.150 \,^{\circ}C |\varepsilon_a|_{24} = 34.46\%$ *i*=1, *j*=3 $T_{1,3} = 55.7862 \circ C |\varepsilon_a|_{1,3} = 54.49\%$ *i*=3, *j*=1 $T_{3,1} = 56.3477 \,{}^{\circ}C |\varepsilon_a|_{3,1} = 24.44\%$ *i*=1, *j*=4 $T_{1,4} = 133.283 \,^{\circ} C |\varepsilon_a|_{1,4} = 24.90\%$ *i*=3, *j*=2 $T_{3,2} = 56.0425 \,^{\circ}C |\varepsilon_a|_{3,2} = 31.70\%$ *i*=2, *j*=1 $T_{2,1} = 36.8164 \circ C |\varepsilon_a|_{2,1} = 44.83\%$ *i*=3, *j*=3 $T_{3,3} = 86.8394 \circ C |\varepsilon_a|_{3,3} = 57.44\%$ *i*=2, *j*=2 $T_{2,2} = 30.8594 \circ C |\varepsilon_a|_{2,2} = 62.03\%$ *i*=3, *j*=4 $T_{3,4} = 160.747 \circ C |\varepsilon_a|_{3,4} = 16.12\%$

Ejemplo 2: Método Gauss-Seidel

Node	Temperature Distribution in the Plate (°C) Number of Iterations								
	1	2	10						
$T_{1,1}$	31.2500	42.9688	73.0239						
$T_{1,2}$	26.5625	38.7695	91.9585						
$T_{1,3}$	25.3906	55.7861	119.0976						
$T_{1,4}$	100.0977	133.2825	172.9755						
$T_{2,1}$	20.3125	36.8164	76.6127						
$T_{2,2}$	11.7188	30.8594	102.1577						
$T_{2,3}$	9.2773	56.4880	137.3802						
$T_{2,4}$	102.3438	156.1493	198.1055						
$T_{3,1}$	42.5781	56.3477	82.4837						
$T_{3,2}$	38.5742	56.0425	103.7757						
$T_{3,3}$	36.9629	86.8393	130.8056						
<i>T</i> _{3,4}	134.8267	160.7471	182.2278						

X Ejemplo 2: Método Gauss-Seidel in Excel

- The numerical solution of Laplace equation at a point is the average of four neighbors $T_{i+1,i} + T_{i-1,i} + T_{i,i+1} + T_{i,i-1}$
- Example for cell S8: =(S7+S9+R8+T8)/4



- Enter the boundary conditions in the appropriate cells.
- Copy and paste to cover the cells where values of the potential are to be calculated. This calculation contains a "circular reference".



• To allow circular references and enable iterations:

 $\mathsf{File} \to \mathsf{Options} \to \mathsf{Formulas}$

On the "Calculations options" form select "Enable iterative calculation"

Opcione	es de Excel			in the second seco	Madding.	8 ×
Gene	eral nulas	$f_{\hat{x}}$ Cambie las opc	iones relativas al cálculo d	e fórmulas, rendimiento y	tratamiento de errores.	•
Revi	sión	Opciones de cálculo:				
Guar	rdar	Cálculo de libro 🛈		✓ Hab <u>i</u> litar cá	culo iterativo	
Idior	ma	Automático	ento nara tablas de datos	Iteraciones i	má <u>x</u> imas: 100	
Avar	nzadas	 Manual 		<u>C</u> ambio má	ximo: 0.001	

We can increase the Maximum Iterations (100 is the deafult) and reduce the Maximum Change (0.001 is the default). Iterations will stop when the maximum iteration is reached or the change is less than the maximum change.

• F9 to recalculate.



Color cell based on value
 To achieve the cell color based on value: Inicio → Estilos →
 Formato condicional → Escalas de color → Más reglas
 We can chose a 3 color scale with blue for mimimum, white or gray for midpoint and red for maximum.





 Plotting the results Normally we use the chart type Surface or Contour.





 Consider the steady state temperature distribution in a square plate. The left, right, and bottom edges of the plate are maintained at a 100°C, whereas the top of the plate is maintained at a constant 0°C. A finite difference mesh for this problem might look something like:



- The residual at each node is: $T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} 4T_{i,j} = 0$
- For each node, take the residual and square them. Then sum the squares to end up with a least-squares minimization problem.



Ejemplo 3: Uso de Solver

Finite Differe	ence Solution									Parámetros de Solver		x
8	8	0	0	0	0	0	0	0		Es <u>t</u> ablecer objetivo:	SCS27	
-	7 100	51.74	33.90	26.95	25.07	26.95	33.90	51.74	100	Para: <u>M</u> áx O	Mín 🔘 <u>V</u> alor de:	ונ
(6 100	73.07	56.89	48.82	46.39	48.82	56.89	73.07	100	Cambiando <u>l</u> as celdas de vari SDS5:SJS11	ables:	a
Ę	5 100	83.64	71.80	65.03	62.86	65.03	71.80	83.64	100	Sujeto a las restricciones:	Land	2
4	4 100	89.71	81.62	76.65	75.00	76.65	81.62	89.71	100		^ <u>A</u> gregar	
	3 100	93.56	88.31	84.97	83.83	84.97	88.31	93.56	100		Cambiar	
2	2 100	96.23	93.11	91.07	90.37	91.07	93.11	96.23	100		Eliminar	
	1 100	98.26	96.80	95.85	95.52	95.85	96.80	98.26	100		<u>R</u> establecer todo	
(0	100	100	100	100	100	100	100		Convertir variables sin res		1
x\y	0	1	2	3	4	5	6	7	8	Método d <u>e</u> resolución:	GRG Nonlinear	
										Método de resolución Seleccione el motor GRG No	onlinear para problemas de Solver no lineales suavizados. Seleccione	
Residuals-so	quared	. =(-	4*D5+C)4+D6+	$C5 + F5)^{4}$	^2				el motor LP Simplex para pro problemas de Solver no sua	oblemas de Solver lineales, y seleccione el motor Evolutionary para ivizados.	
8	8	↓ \	1 00 0		00 20/	-				Ayuda	Resolver Cerrar	5
-	7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000	ociones	? <mark>.</mark>	ciones	×
	6	0 0000	0 0000	0 0000	0 0000	0 0000	0 0000	0 000				
t	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000	Todos los métodos GRG N	Ionlinear Evolutionary	Todos los métodos GRG Nonlinear Evolutionary	
Ę	5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000	Todos los métodos GRG N Precisión de restriccione	Ionlinear Evolutionary	Todos los métodos GRG Nonlinear Evolutionary	
	5	0.0000	0.0000	0.0000	0.0000	0.0000 0.0000	0.0000	0.000	Todos los métodos GRG N Precisión de restriccione V Usar escala automátic	Ionlinear Evolutionary S: 0.000001 Ca	Todos los métodos GRG Nonlinear Evolutionary Convergencia: Derivados	
	5 4 3	0.0000 0.0000 0.0000	0.0000 0.0000 0.0000	0.0000 0.0000 0.0000	0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000	0.000 0.000 0.000 0.000	Todos los métodos GRG N Precisión de restriccione V Usar escala automátic Mostrar resultados de	konlinear Evolutionary 5: 0.000001 ca e iteraciones	Todos los métodos GRG Nonlinear Evolutionary Convergencia: 0.00001 Derivados O Adelantada O Central]
	5 4 3 2	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.000 0.000 0.000 0.000 0.000	Todos los métodos GRG N Precisión de restriccione Usar escala automátic Mostrar resultados de Resolviendo restriccion	ionlinear Evolutionary s: 0.000001 ca e iteraciones nes de enteros	Todos los métodos GRG Nonlinear Evolutionary Convergencia: 0.00001 Derivados O Adelantada O Central Inicio múltiple Usar inicio múltiple	
	5 4 3 2 1	0.0000 0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000 0.0000	0.000 0.000 0.000 0.000 0.000 0.000	Todos los métodos GRG N Precisión de restriccione Usar escala automáti Mostrar resultados de Resolviendo restriccione Omitir restricciones Optimalidad de entero	Ionlinear Evolutionary s: 0.000001 ca e iteraciones nes de enteros : de enteros (%): 1	Todos los métodos GRG Nonlinear Evolutionary Cgnvergencia: 0.00001 Derivados O Adelantada O Cgntral Inicio múltiple Usar inicio múltiple Tamaño de goblación: 100	
	5 4 3 2 1 0	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000	0.000 0.000 0.000 0.000 0.000	Todos los métodos GRG N Precisión de restriccione Usar escala automátii Mostrar resultados de Resolviendo restricciones Optimalidad de entero I Resolviendo límites	ionlinear Evolutionary 5: 0.000001 ca e iteraciones nes de enteros	Todos los métodos GRG Nonlinear Evolutionary Cgnvergencia: 0.00001 Derivados O Adelantada O Cgntral Inicio múltiple Usar inicio múltiple Tamaño de goblación: 100 Valor de inicialización aleatorio: 0	
x\y	5 4 3 2 1 0 0	0.0000 0.0000 0.0000 0.0000 0.0000 1	0.0000 0.0000 0.0000 0.0000 0.0000 2	0.0000 0.0000 0.0000 0.0000 0.0000 3	0.0000 0.0000 0.0000 0.0000 0.0000 4	0.0000 0.0000 0.0000 0.0000 0.0000 5	0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 6	0.000 0.000 0.000 0.000 0.000	Todos los métodos GRG N Precisión de restriccione Usar escala automáti Mostrar resultados de Resolviendo restriccione Omitir restricciones Optimalidad de entero l Resolviendo límites Tiempo máximo (segund	ionlinear Evolutionary s: 0.000001 ca e e iteraciones e nes de enteros e e: de enteros e (%): 1	Todos los métodos GRG Nonlinear Evolutionary Cgnvergencia: 0.00001 Derivados O Adelantada O Cgntral Inicio múltiple Usar inicio múltiple Tamaño de goblación: 100 Valor de inicialización aleatorio: 0 Image: Requerir Jímites en variables 0	
x\y	5 4 3 2 1 0 0	0.0000 0.0000 0.0000 0.0000 0.0000 1	0.0000 0.0000 0.0000 0.0000 0.0000 2	0.0000 0.0000 0.0000 0.0000 0.0000 3	0.0000 0.0000 0.0000 0.0000 0.0000 4	0.0000 0.0000 0.0000 0.0000 0.0000 5	0.0000 0.0000 0.0000 0.0000 0.0000 6	0.000 0.000 0.000 0.000 0.000	Todos los métodos Precisión de restriccione Usar escala automáti Mostrar resultados de Resolviendo restricciones Omitig restricciones Optimalidad de entero l Resolviendo límites Jiempo máximo (seguno Jieraciones:	lonlinear Evolutionary 5: 0.000001 ca e Iteraciones nes de enteros : de enteros (%): 1 dos): 1000 1000	Todos los métodos GRG Nonlinear Evolutionary Cgnvergencia: 0.00001 Derivados O O Adelantada O Cgntral Inicio múltiple Usar inicio múltiple Tamaño de goblación: 100 Valog de inicialización aleatorio: 0 Image: Requerir límites en variables 0	
x\y sum =	5 4 3 2 1 0 0 2.7745E-13	0.0000 0.0000 0.0000 0.0000 0.0000 1 1 =SUM	0.0000 0.0000 0.0000 0.0000 0.0000 2 A(D17:	0.0000 0.0000 0.0000 0.0000 0.0000 3 J23)	0.0000 0.0000 0.0000 0.0000 0.0000 4	0.0000 0.0000 0.0000 0.0000 0.0000 5	0.0000 0.0000 0.0000 0.0000 0.0000 6	0.000 0.000 0.000 0.000 0.000	Todos los métodos Precisión de restriccione Usar escala automátii Mostrar resultados de Resolviendo restricciones Omitig restricciones Optimalidad de entero Resolviendo límites Tiempo máximo (segund Iteraciones: Restricciones de entero	Ionlinear Evolutionary 5: 0.000001 ca e iteraciones nes de enteros : de enteros (%6): 1 dos): 1000 1000 s y Evolutionary:	Todos los métodos GRG Nonlinear Evolutionary Cgnvergencia: 0.00001 Derivados O Q Adelantada © Cgntral Inicio múltiple Usar inicio múltiple Tamaño de goblación: 100 Valog de inicialización aleatorio: 0 Imicio múltiple Inicialización aleatorio:	
x\y sum =	5 5 4 3 2 1 0 0 0 2.7745E-13	0.0000 0.0000 0.0000 0.0000 0.0000 1 =SUM	0.0000 0.0000 0.0000 0.0000 0.0000 2 A(D17:	0.0000 0.0000 0.0000 0.0000 0.0000 3 J23)	0.0000 0.0000 0.0000 0.0000 0.0000 4	0.0000 0.0000 0.0000 0.0000 0.0000 5	0.0000 0.0000 0.0000 0.0000 0.0000 6	0.000 0.000 0.000 0.000 0.000	Todos los métodos GRG N Precisión de restriccione Usar escala automáti Mostrar resultados de Resolviendo restricciones Omitig restricciones Optimalidad de entero I Resolviendo límites Tiempo máximo (segund Iteraciones: Restricciones de entero Máximo de subproblem	Ionlinear Evolutionary S: 0.000001 Ca e iteraciones nes de enteros s de enteros (%): 1 dos): 1000 1000 s y Evolutionary: nas:	Todos los métodos GRG Nonlinear Evolutionary Cgnvergencia: 0.00001 Derivados O O Adelantada O Cgntral Inicio múltiple Usar inicio múltiple Tamaño de goblación: 100 Valog de inicialización aleatorio: 0 Imites en variables Inicialización aleatorio:	
x\y sum =	5 4 3 2 1 0 0 2.7745E-13	0.0000 0.0000 0.0000 0.0000 0.0000 1 =SUM	0.0000 0.0000 0.0000 0.0000 0.0000 2 A(D17:	0.0000 0.0000 0.0000 0.0000 0.0000 3 J23)	0.0000 0.0000 0.0000 0.0000 0.0000 4	0.0000 0.0000 0.0000 0.0000 0.0000 5	0.0000 0.0000 0.0000 0.0000 0.0000 6	0.000 0.000 0.000 0.000 0.000	Todos los métodos Precisión de restriccione Usar escala automátii Mostrar resultados de Resolviendo restricciones Omitig restricciones Optimalidad de entero I Resolviendo límites Tiempo máximo (segund Iteraciones: Restricciones de entero Máximo de soluciones ;	lonlinear Evolutionary S: 0.000001 Ca e iteraciones nes de enteros : de enteros (%): 1 dos): 1000 1000 s y Evolutionary: nas:	Todos los métodos GRG Nonlinear Evolutionary Cgnvergencia: 0.00001 Derivados O O Adelantada O Central Inicio múltiple Usar inicio múltiple Tamaño de goblación: 100 Valor de inicialización aleatorio: 0 Image: Requerir Jimites en variables 0	
x\y sum =	5 4 3 2 1 0 0 2.7745E-13	0.0000 0.0000 0.0000 0.0000 1 1 =SUM	0.0000 0.0000 0.0000 0.0000 0.0000 2 A(D17:	0.0000 0.0000 0.0000 0.0000 0.0000 3 J23)	0.0000 0.0000 0.0000 0.0000 0.0000 4	0.0000 0.0000 0.0000 0.0000 0.0000 5	0.0000 0.0000 0.0000 0.0000 0.0000 6	0.000 0.000 0.000 0.000 0.000	Todos los métodos Precisión de restriccione Usar escala automátic Mostrar resultados de Resolviendo restricciones Omitir restricciones Optimalidad de entero Resolviendo límites Tiempo máximo (segund Jteraciones: Restricciones de entero Máximo de subproblem Máximo de soluciones ;	Ionlinear Evolutionary 5: 0.000001 ca e iteraciones nes de enteros : de enteros (%6): 1 dos): 1000 1000 s y Evolutionary: nas: yiables:	Todos los métodos GRG Nonlinear Evolutionary Cgnvergencia: 0.00001 Derivados O Adelantada O Cgntral Inicio múltiple Inicio múltiple Tamaño de goblación: 100 Valog de inicialización aleatorio: 0 Image: Requerir jimites en variables Imites en variables	



• Recall the equation used in the Gauss-Siedel Method,

$$T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{4}$$

• Because the Gauss-Siedel Method is guaranteed to converge, we can accelerate the process by using overrelaxation. In this case,

$$T_{i,j}^{relaxed} = \lambda T_{i,j}^{new} + (1 - \lambda) T_{i,j}^{old}$$

• The λ is known as the "overrelaxation parameter" and is in the range $0 < \lambda < 2$.



- In the past examples, the boundary conditions on the plate had a specified temperature on each edge.
 What if the conditions are different? For example, what if one of the edges of the plate is insulated.
- In this case, the boundary condition would be the derivative of the temperature. Because if the right edge of the plate is insulated, then the temperatures on the right edge nodes also become unknowns.



The finite difference equation in this case for the right edge for the nodes (*m*, *j*) for *j* = 2,3,..*n*−1

$$T_{m+1,j} + T_{m-1,j} + T_{m,j-1} + T_{m,j+1} - 4T_{m,j} = 0$$

• However the node (m+1, j) is not inside the plate. The derivative boundary condition needs to be used to account for these additional unknown nodal temperatures on the right edge. This is done by approximating the derivative at the edge node (m, j) as





• Rearranging this approximation gives us,

$$T_{m+1,j} = T_{m-1,j} + 2(\Delta x) \frac{\partial T}{\partial x} \Big|_{m,j}$$

• We can then substitute this into the original equation gives us,

$$2T_{m-1,j} + 2(\Delta x)\frac{\partial T}{\partial x}\Big|_{m,j} + T_{m,j-1} + T_{m,j+1} - 4T_{m,j} = 0$$

• Recall that is the edge is insulated then,

$$\left.\frac{\partial T}{\partial x}\right|_{m,j} = 0$$

• Substituting this again yields,

$$2T_{m-1,j} + T_{m,j-1} + T_{m,j+1} - 4T_{m,j} = 0$$



Ecuaciones en derivadas parciales parabólicas

 The general form for a second order linear PDE with two independent variables and one dependent variable is

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D\frac{\partial u}{\partial x} + E\frac{\partial u}{\partial y} + Fu + G = 0$$

- The criteria for an equation of this type to be considered parabolic: $B^2 - 4AC = 0$
- Examine the heat-conduction equation given by $\alpha \frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t} \quad \text{where } \alpha = \frac{k}{\rho C} \begin{cases} k = \text{thermal conductivity of rod material,} \\ \rho = \text{density of rod material,} \\ C = \text{specific heat of the rod material.} \end{cases}$ where $A = \alpha, B = 0, C = 0, D = 0, E = -1, F = 0, G = 0$ thus we can classify this equation as parabolic.



- Consider the flow of heat within a metal rod of length L, one end of which is held at a known high temperature, the other end at a lower temperature.
 - Heat will flow from the hot end to the cooler end.
 - We'll assume that the rod is perfectly insulated, so that heat loss through the sides can be neglected.



• We want to calculate the temperature along the length of the rod as a function of time.

X Discretización de una EDP Parabólica

- For a rod of length *L* divided into n+1 nodes $\Delta x = \frac{L}{n}$
- The time is similarly broken into time steps of Δt
- Hence T_i^j corresponds to the temperature at node *i*, that is,

$$x = (i)(\Delta x) \text{ and time } t = (j)(\Delta t)$$

$$x \quad \Delta x \quad \Delta x \quad dx$$

$$i - 1 \quad i \quad i + 1$$
Schematic diagram showing interior nodes


Solución EDP Parabólica: Método explícito

• If we define $\Delta x = \frac{L}{n}$ we can then write the finite central divided difference approximation of the left hand side at a general interior node (i) as $\frac{\partial^2 T}{\partial x^2}\Big|_{i,j} \approx \frac{T_{i+1}^j - 2T_i^j + T_{i-1}^j}{(\Delta x)^2}$ where (j) is the node number along

the time.

• The time derivative on the right hand side is approximated by the forward divided difference method as, $\partial T \Big| = T_i^{j+1} - T_i^j$



Ec. derivadas parciales



Solución EDP Parabólica: Método explícito

• Substituting these approximations into the governing equation yields

$$\alpha \frac{T_{i+1}^{j} - 2T_{i}^{j} + T_{i-1}^{j}}{(\Delta x)^{2}} = \frac{T_{i}^{j+1} - T_{i}^{j}}{\Delta t}$$

• Solving for the temp at the time node j+1 gives

$$T_{i}^{j+1} = T_{i}^{j} + \alpha \frac{\Delta t}{(\Delta x)^{2}} \left(T_{i+1}^{j} - 2T_{i}^{j} + T_{i-1}^{j} \right)$$

- choosing, $\lambda = \alpha \frac{\Delta t}{(\Delta x)^2}$
- we can write the equation as, $T_i^{j+1} = T_i^j + \lambda \left(T_{i+1}^j 2T_i^j + T_{i-1}^j \right)$
- we can be solved explicitly: for each internal location node of the rod for time node *j*+1 in terms of the temperature at time node *j*. If we know the temperature at node *j* = 0, and the boundary temperatures, we can find the temperature at the next time step. We continue the process until we reach the time at which we are interested in finding the temperature.



• Consider a steel rod that is subjected to a temperature of $100^{\circ}C$ on the left end and $25^{\circ}C$ on the right end. If the rod is of length 0.05m, use the explicit method to find the temperature distribution in the rod from t = 0 and t = 9 seconds. Use $\Delta x = 0.01m$, $\Delta t = 3s$.

• Given:
$$k = 54 \frac{W}{m-K}$$
, $\rho = 7800 \frac{kg}{m^3}$, $C = 490 \frac{J}{kg-K}$

• The initial temperature of the rod is $20^{\circ}C$.





- Number of time steps $=\frac{t_{final} t_{initial}}{\Delta t} = \frac{9 0}{3} = 3$ Recall, $\alpha = \frac{k}{\rho C} = \frac{54}{7800 \times 490} = 1.4129 \times 10^{-5} m^2 / s$
- Then, $\lambda = \alpha \frac{\Delta t}{(\Lambda x)^2} = 1.4129 \times 10^{-5} \frac{3}{(0.01)^2} = 0.4239$
- Boundary Conditions $T_0^j = 100^{\circ}C$ $T_5^j = 25^{\circ}C$ for all j = 0,1,2,3
- All internal nodes are at 20°C for $t = 0 \sec$: $T_i^0 = 20^\circ C$, for all i = 1, 2, 3, 4 $T_0^0 = 100^{\circ}C$
 - $T_1^0 = 20^{\circ}C$ $\begin{array}{c|c} T_2^0 = 20^{\circ}C \\ T_3^0 = 20^{\circ}C \end{array}$ Interior nodes $T_4^0 = 20^{\circ}C$ $T_{5}^{0} = 25^{\circ}C$

We can now calculate the temperature at each node explicitly using the equation formulated earlier,

$$T_{i}^{j+1} = T_{i}^{j} + \lambda \left(T_{i+1}^{j} - 2T_{i}^{j} + T_{i-1}^{j} \right)$$



Nodal temperatures vs. Time $t = 6 \sec i = 2$ $t = 9 \sec i = 3$ $t = 0 \sec j = 0$ $t = 3 \sec i = 1$ $T_0^2 = 100^{\circ}C \qquad \qquad T_0^3 = 100^{\circ}C$ $T_0^0 = 100^{\circ}C$ $T_0^1 = 100^{\circ}C - Boundary Condition$ $T_1^0 = 20^{\circ}C$ $T_1^2 = 59.073^{\circ}C$ $T_1^3 = 65.953^{\circ}C$ $T_1^1 = 53.912^{\circ}C$ $T_2^0 = 20^{\circ}C$ $T_2^2 = 34.375^{\circ}C$ $T_2^3 = 39.132^{\circ}C$ $T_2^1 = 20^{\circ}C$ Interior nodes $T_{3}^{0} = 20^{\circ}C$ $T_3^2 = 20.889^{\circ}C$ $T_3^3 = 27.266^{\circ}C$ $T_{3}^{1} = 20^{\circ}C$ $T_4^0 = 20^{\circ}C$ $T_4^2 = 22.442^{\circ}C$ $T_4^3 = 22.872^{\circ}C$ $T_{A}^{1} = 22.120^{\circ}C$ $T_5^3 = 25^{\circ}C$ $T_{5}^{0} = 25^{\circ}C$ $T_5^1 = 25^{\circ}C - \text{Boundary Condition}$ $T_5^2 = 25^{\circ}C$ 120 Temperature distribution along the length of the rod 100 Γemperature, T (°C) 80 60 =3 secs 6 secs 40 t=9 secs 20 0 0 0.01 0.02 0.03 0.04 0.05 Location on rod, x(m)



- Consider an insulated 10-cm brass rod, initially at a temperature of 0°C. One end of the rod is heated to 100°C. Assume that there is no heat loss through the sides of the rod. The coefficient of thermal conductivity k is 0.26 cal s⁻¹ cm⁻¹ deg⁻¹, the heat capacity C is 0.094 cal g⁻¹ deg⁻¹ and the density ρ is 8.4 g cm⁻³. The coefficient λ is 3.04 s cm⁻². Calculate the temperature along the rod, in 1-second and 1-cm intervals.
- The stencil is: j + 1

• File: edp_parabolica.xlsx



Solución EDP Parabólica: Método implícito

- Using the explicit method, we were able to find the temperature at each node, one equation at a time.
- However, the temperature at a specific node was only dependent on the temperature of the neighboring nodes from the previous time step. This is contrary to what we expect from the physical problem.
- The implicit method allows us to solve this and other problems by developing a system of simultaneous linear equations for the temperature at all interior nodes at a particular time.
- The second derivative is approximated by the CDD and the first derivative by the BDD scheme at time level j+1 at node (i) as

$$\alpha \frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t} \xrightarrow{\partial T} \frac{\partial^2 T}{\partial x^2} \bigg|_{i,j+1} \approx \frac{T_{i+1}^{j+1} - 2T_i^{j+1} + T_{i-1}^{j+1}}{(\Delta x)^2}$$
$$\xrightarrow{\partial T} \frac{\partial T}{\partial t} \bigg|_{i,j+1} \approx \frac{T_i^{j+1} - T_i^j}{\Delta t}$$



Solución EDP Parabólica: Método implícito

 Substituting these approximations into the heat conduction equation yields

$$\alpha \frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t} \longrightarrow \alpha \frac{T_{i+1}^{j+1} - 2T_i^{j+1} + T_{i-1}^{j+1}}{(\Delta x)^2} = \frac{T_i^{j+1} - T_i^j}{\Delta t}$$

• Rearranging yields

$$-\lambda T_{i-1}^{j+1} + (1+2\lambda)T_i^{j+1} - \lambda T_{i+1}^{j+1} = T_i^{j}$$

given that

$$\lambda = \alpha \, \frac{\Delta t}{(\Delta x)^2}$$

 The rearranged equation can be written for every node during each time step. These equations can then be solved as a simultaneous system of linear equations to find the nodal temperatures at a particular time.



• Consider a steel rod that is subjected to a temperature of $100^{\circ}C$ on the left end and $25^{\circ}C$ on the right end. If the rod is of length 0.05m, use the implicit method to find the temperature distribution in the rod from t = 0 and t = 9 seconds. Use $\Delta x = 0.01m$, $\Delta t = 3s$.

• Given:
$$k = 54 \frac{W}{m-K}$$
, $\rho = 7800 \frac{kg}{m^3}$, $C = 490 \frac{J}{kg-K}$

• The initial temperature of the rod is $20^{\circ}C$.





- Number of time steps $=\frac{t_{final} t_{initial}}{\Delta t} = \frac{9 0}{3} = 3$ Recall, $\alpha = \frac{k}{\rho C} = \frac{54}{7800 \times 490} = 1.4129 \times 10^{-5} m^2 / s$
- Then, $\lambda = \alpha \frac{\Delta t}{(\Lambda x)^2} = 1.4129 \times 10^{-5} \frac{3}{(0.01)^2} = 0.4239$
- Boundary Conditions $T_0^j = 100^{\circ}C$ $T_5^j = 25^{\circ}C$ for all j = 0,1,2,3
- All internal nodes are at 20°C for $t = 0 \sec$: $T_i^0 = 20^\circ C$, for all i = 1, 2, 3, 4
 - $T_0^0 = 100^{\circ}C$ $T_1^0 = 20^{\circ}C$ $T_4^0 = 20^{\circ}C$ $T_{5}^{0} = 25^{\circ}C$

We can now form the system of equations for the first time step by writing the approximated heat $T_2^0 = 20^{\circ}C$ Interior nodes $T_3^0 = 20^{\circ}C$ Interior nodes

$$-\lambda T_{i-1}^{j+1} + (1+2\lambda)T_i^{j+1} - \lambda T_{i+1}^{j+1} = T_i^{j}$$



- Nodal temperatures when $t = 3 \sec t$
- For the first time step we can write four such equations with four unknowns, expressing them in matrix form yields

1.8478	-0.4239	0	0	$\left[T_1^1\right]$		62.390	
-0.4239	1.8478	-0.4239	0	T_2^1		20	
0	-0.4239	1.8478	-0.4239	T_3^1	=	20	
0	0	-0.4239	1.8478	T_4^1		_30.598_	

 The above coefficient matrix is tri-diagonal, so special algorithms (e.g.Thomas' algorithm) can be used to solve. The solution is given by

$$\begin{bmatrix} T_1^1 \\ T_2^1 \\ T_3^1 \\ T_4^1 \end{bmatrix} = \begin{bmatrix} 39.451 \\ 24.792 \\ 21.438 \\ 21.477 \end{bmatrix} = \begin{bmatrix} 100 \\ T_1^1 \\ T_2^1 \\ T_3^1 \\ T_4^1 \end{bmatrix} = \begin{bmatrix} 100 \\ 39.451 \\ 24.792 \\ 21.438 \\ T_4^1 \\ T_5^1 \end{bmatrix} = \begin{bmatrix} 24.792 \\ 21.438 \\ 21.477 \\ 25 \end{bmatrix}$$



Nodal temperatures when: $t = 3 \sec t$





Solución EDP Parabólica: Método Crank-Nicolson

- Using the implicit method our approximation of $\frac{\partial^2 T}{\partial x^2}$ was of accuracy $O(\Delta x)^2$, while our approximation of $\frac{\partial T}{\partial t}$ was of $O(\Delta t)$ accuracy.
- One can achieve similar orders of accuracy by approximating the second derivative, on the left hand side of the heat equation, at the midpoint of the time step. Doing so yields

$$\frac{\partial^2 T}{\partial x^2}\Big|_{i,j} \approx \frac{\alpha}{2} \left[\frac{T_{i+1}^{j} - 2T_{i}^{j} + T_{i-1}^{j}}{(\Delta x)^2} + \frac{T_{i+1}^{j+1} - 2T_{i}^{j+1} + T_{i-1}^{j+1}}{(\Delta x)^2} \right]$$

• The first derivative, on the right hand side of the heat equation, is approximated using the forward divided difference method at time level j+1, $\partial T | = T_i^{j+1} - T_i^j$

$$\left. \frac{\partial T}{\partial t} \right|_{i,j} \approx \frac{T_i^{j+1} - T_i^j}{\Delta t}$$



Solución EDP Parabólica: Método Crank-Nicolson

Substituting these approximations into the governing equation for heat conductance yields

$$\frac{\alpha}{2} \left[\frac{T_{i+1}^{j} - 2T_{i}^{j} + T_{i-1}^{j}}{\left(\Delta x\right)^{2}} + \frac{T_{i+1}^{j+1} - 2T_{i}^{j+1} + T_{i-1}^{j+1}}{\left(\Delta x\right)^{2}} \right] = \frac{T_{i}^{j+1} - T_{i}^{j}}{\Delta t}$$

• giving

$$-\lambda T_{i-1}^{j+1} + 2(1+\lambda)T_i^{j+1} - \lambda T_{i+1}^{j+1} = \lambda T_{i-1}^j + 2(1-\lambda)T_i^j + \lambda T_{i+1}^j$$

• where

$$\lambda = \alpha \, \frac{\Delta t}{(\Delta x)^2}$$

• Having rewritten the equation in this form allows us to discretize the physical problem. We then solve a system of simultaneous linear equations to find the temperature at every node at any point in time.



• Consider a steel rod that is subjected to a temperature of $100^{\circ}C$ on the left end and $25^{\circ}C$ on the right end. If the rod is of length 0.05m, use the Crank-Nicolson method to find the temperature distribution in the rod from t = 0 and t = 9 seconds. Use $\Delta x = 0.01m$, $\Delta t = 3s$.

• Given:
$$k = 54 \frac{W}{m-K}$$
, $\rho = 7800 \frac{kg}{m^3}$, $C = 490 \frac{J}{kg-K}$

• The initial temperature of the rod is $20^{\circ}C$.





- Number of time steps $=\frac{t_{final} t_{initial}}{\Delta t} = \frac{9 0}{3} = 3$ Recall, $\alpha = \frac{k}{\rho C} = \frac{54}{7800 \times 490} = 1.4129 \times 10^{-5} m^2 / s$
- Then, $\lambda = \alpha \frac{\Delta t}{(\Lambda x)^2} = 1.4129 \times 10^{-5} \frac{3}{(0.01)^2} = 0.4239$
- Boundary Conditions $T_0^j = 100^{\circ}C$ $T_5^j = 25^{\circ}C$ for all j = 0,1,2,3
- All internal nodes are at $20^{\circ}C$ for $t = 0 \sec$: $T_i^0 = 20^{\circ}C$, for all i = 1, 2, 3, 4

 $T_0^0 = 100^{\circ}C$ We can now form the system of equations for the $T_1^0 = 20^{\circ}C$ first time step by writing the approximated heat $T_2^0 = 20^{\circ}C$ Interior nodes
Interior nodes
Inst time step by writing the approx
conduction equation for each node $T_{4}^{0} = 20^{\circ}C \Big| \qquad -\lambda T_{i-1}^{j+1} + 2(1+\lambda)T_{i}^{j+1} - \lambda T_{i+1}^{j+1} = \lambda T_{i-1}^{j} + 2(1-\lambda)T_{i}^{j} + \lambda T_{i+1}^{j}$ $T_{5}^{0} = 25^{\circ}C$



- Nodal temperatures when $t = 3 \sec t$
- For the first time step we can write four such equations with four unknowns, expressing them in matrix form yields

2.8478	-0.4239	0	0	$\left[T_1^1\right]$	[116.30]
-0.4239	2.8478	-0.4239	0	T_2^1	40.000
0	-0.4239	2.8478	-0.4239	T_3^1	40.000
0	0	-0.4239	2.8478	T_4^1	52.718

• The above coefficient matrix is tri-diagonal, so special algorithms (e.g.Thomas' algorithm) can be used to solve. The solution is given by

$$\begin{bmatrix} T_1^1 \\ T_2^1 \\ T_3^1 \\ T_4^1 \end{bmatrix} = \begin{bmatrix} 44.372 \\ 23.746 \\ 20.797 \\ 21.607 \end{bmatrix} \qquad \begin{bmatrix} T_0 \\ T_1^1 \\ T_2^1 \\ T_3^1 \\ T_4^1 \\ T_5^1 \end{bmatrix} = \begin{bmatrix} 100 \\ 44.372 \\ 23.746 \\ 20.797 \\ 21.607 \\ 25 \end{bmatrix}$$

100



Nodal temperatures when: $t = 3 \sec t$



Ec. derivadas parciales



Comparación de métodos: temperaturas en 9 seg.

• The table below allows you to compare the results from all three methods discussed in juxtaposition with the analytical solution.

Node	Explicit	Implicit	Crank- Nicolson	Analytical	
T_1^3	65.953	59.043	62.604	62.510	
T_{2}^{3}	39.132	36.292	37.613	37.084	
T_{3}^{3}	27.266	26.809	26.562	25.844	
T_{4}^{3}	22.872	24.243	24.042	23.610	



Ecuaciones en derivadas parciales hiperbólicas

 The general form for a second order linear PDE with two independent variables and one dependent variable is

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D\frac{\partial u}{\partial x} + E\frac{\partial u}{\partial y} + Fu + G = 0$$

- The criteria for an equation of this type to be considered hyperbolic: $B^2 - 4AC > 0$
- The wave equation (oscillatory systems) given by $\frac{\partial^2 y}{\partial t^2} = k \frac{\partial^2 y}{\partial x^2} \quad \text{where } k = \frac{Tg}{w} \quad \begin{array}{l} T = \text{tension}, \\ g = \text{gravitational constant}, \\ w = \text{weight/unit} = W/L, W = \text{weight}, L = \text{length} \end{array}$ where A = 1, B = 0, C = -1thus we can classify this equation as hyperbolic.



 A string of certain length and weight is under a fixed tension. Initially the mid-point of the string is displaced some distance from its equilibrium position and released.



• We want to calculate the displacement as a function of time at fixed intervals along the length of the string.



Solución EDP hiperbólica: Método explícito

Once again, we can solve the problem by replacing derivatives by finite differences.

$$\frac{T_i^{j+1} - 2T_i^{j} + T_i^{j-1}}{\left(\Delta t\right)^2} = \frac{Tg}{w} \left[\frac{T_{i+1}^{j} - 2T_i^{j} + T_{i-1}^{j}}{\left(\Delta x\right)^2} \right]$$

• which, when rearranged, yields

$$T_{i}^{j+1} = \frac{Tg}{w} \frac{(\Delta t)^{2}}{(\Delta x)^{2}} (T_{i+1}^{j} + T_{i-1}^{j}) - T_{i}^{j-1} + 2\left(1 - \frac{Tg}{w} \frac{(\Delta t)^{2}}{(\Delta x)^{2}}\right) T_{i}^{j}$$

- If we set $Tg(\Delta t)^2/w(\Delta x)^2 = 1$, the above equation is simplified to $T_i^{j+1} = T_{i+1}^j + T_{i-1}^j - T_i^{j-1}$
- When employing the simplified equation, the value of Δt is determined by the expression $\Delta t = \frac{\Delta x}{\sqrt{Tg/w}}$
- To begin the calculations (value at t₁), it is required values of the function at t₀ = 0 and also a value at t₋₁. We can get a value for the function at t₋₁ by making use of the fact that the function is periodic.



 A string 50 cm long and weighing 0.5 g is under a tension of 33 kg. Initially the mid-point of the string is displaced 0.5 cm from its equilibrium position and released. We want to calculate the displacement as a function of time at 5 cm intervals along the length of the string, using equation

$$T_i^{j+1} = T_{i+1}^j + T_{i-1}^j - T_i^{j-1}$$

- From equation $\Delta t = \frac{\Delta x}{\sqrt{Tg/w}}$ the Δt must be 8.8 x 10⁻⁵ seconds.
- File: edp_hiperbolica.xlsx