# MA-0501 Análisis Numérico I
# EXAMEN 2

**Fecha:** Miércoles 13 de noviembre, 9:00am.
**Instrucciones:**

– El examen consta de **una única parte** para realizar en la casa. El trabajo es estrictamente individual. Cualquier caso de plagio será atendido según el procedimiento del Reglamento de Orden y Disciplina de la universidad.

– Debe subir a Mediación Virtual antes del lunes 18 de noviembre, 11:59pm, **dos archivos**:

  – Un archivo en formato pdf con la resolución de los ejercicios y la discusión de los resultados. El documento debe llamarse `ApellidosNombre.pdf`

  – Un archivo `.m` o `.xlm` con todos los códigos implementados. El archivo debe llamarse `ApellidosNombre.xxx`

– La prueba tiene un total de 65 puntos.

**Parte I. Respuesta corta.**

1. (2 puntos) Defina el número de condición de una matriz cuadrada $A$. ¿Qué implicaciones tiene un número de condición alto al resolver un sistema $Ax = b$?

2. (2 puntos) Considere $A \in \mathbb{R}^{m\times n}$. Demuestre que $\|A\|_F = \sqrt{\sum_j \sigma_j^2}$, donde $\{\sigma_j\}$ son los valores singulares de $A$.

3. (2 puntos) Considere $A \in \mathbb{R}^{m\times n}$. Describa un algoritmo para encontrar una base del espacio generado por las filas de $A$.

4. (2 puntos) Si $\boldsymbol{u} \in \mathbb{R}^m$, $\boldsymbol{v} \in \mathbb{R}^m$ son vectores ortogonales y de norma unitaria, determine el rango y los valores singulares de la matriz $A = \boldsymbol{u}\boldsymbol{v}^T + \boldsymbol{v}\boldsymbol{u}^T$.

5. (2 puntos) Dados tres nodos $(x_0, y_0), (x_1, y_1), (x_2, y_2)$, con $x_1, x_2, x_3$ distintos, describa un algoritmo que determine una función lineal $f$ de forma tal que se minimice el error cuadrático

$$\sum_{i=1}^{3} [y_i - f(x_i)]^2.$$

**Parte II. Desarrollo.** Debe incluir todo el procedimiento que permita concluir el resultado, así como códigos debidamente documentados y gráficas apropiadas.

6. Considere el algoritmo para obtener la factorización de Cholesky de una matriz definida positiva:

    **Data:** Matriz $A \in \mathbb{R}^{m\times m}$
    **Result:** Factorización $A = R^T R$
    $R = triu(A)$;
    **for** $k = 1 : m$ **do**
        **for** $j = k+1 : m$ **do**
            $R_{j,j:m} = R_{j,j:m} - R_{k,j:m} R_{kj}/R_{k,k}$;
        **end**
        $R_{k,k:m} = R_{k,k:m}/\sqrt{R_{k,k}}$;
    **end**

    **Algoritmo 1:** Factorización de Cholesky

    a) (3 puntos) Implemente una función `R = mychol(A)` para el algoritmo anterior. Note que al inicio $R$ se define como la parte triangular superior de $A$ con el comando `triu`.

    b) (3 puntos) Tome $m = 20$. Genere una matriz aleatoria simétrica definida positiva `A`. Calcule la factorización de Cholesky y calcule $\|A - R^T R\|_2$. Compare la factorización con la que brinda `MATLAB` con su función `chol(A)`.

    c) (3 puntos) Tome ahora $m = 20$. Defina `A = hilb(m)` como la matriz de Hilbert de tamaño $m \times m$. Repita el inciso anterior. ¿Qué ocurre en este caso? Justifique. Utilice la factorización de `mychol` para resolver el sistema $Ax = b$ (donde asumimos que la solución exacta es un vector de unos). ¿Es el error en la solución numérica del sistema el esperado?

7. Considere la ecuación diferencial dada por $y' = t - y$ en el intervalo $(0, 10)$, con condición inicial $y(0) = 0$. Dado $n$ natural positivo, considere una partición uniforme

    $$0 = t_0 < t_1 < \ldots < t_n = 10.$$

    a) (2 puntos) Verifique (a mano) que $y(t) = t + e^{-t} - 1$ es una solución de la ecuación diferencial.

    b) (2 puntos) Escriba (a mano) la iteración que brinda el método de Euler.

    c) (3 puntos) Considere ahora el método del trapecio. Calcule (a mano) la iteración correspondiente para $y_{k+1}$ de forma explícita (en función de $y_k, t_k, t_{k+1}$).

    d) (3 puntos) Implemente en `MATLAB` ambas iteraciones (Euler y trapecio) para obtener una aproximación de la solución de la ecuación diferencial. En un mismo gráfico visualice la solución exacta y las aproximaciones para $n = 20$.

*e*) (2 puntos) Para el vector de errores $e^n \in \mathbb{R}^{n+1}$ con entradas $e^n_k = |y(t_k) - y_k|$, grafique la norma $\|e^n\|_\infty$ en función de $h = 10/n$, para valores de $n$ apropiados y para ambos métodos. ¿Cuáles son los órdenes de convergencia observados?

8. Considere la ecuación diferencial dada por el sistema

$$\begin{cases} \dfrac{d^2x}{dt^2} = x + 2\dfrac{dy}{dt} - u_c \cdot \dfrac{x+u}{((x+u)^2+y^2)^{3/2}} - u \cdot \dfrac{x-u_c}{((x-u_c)^2+y^2)^{3/2}}, \\ \dfrac{d^2y}{dt^2} = y - 2\dfrac{dx}{dt} - u_c \cdot \dfrac{y}{((x+u)^2+y^2)^{3/2}} - u \cdot \dfrac{y}{((x-u_c)^2+y^2)^{3/2}}, \end{cases}$$

donde $x(t), y(t)$ son funciones $x, y : [0, \infty) \to \mathbb{R}$. Dicho sistema modela el movimiento bidimensional de un satélite, donde $(x(t), y(t))$ representa su ubicación en el tiempo $t$. Defina las constantes $u = 0.012277471$ y $u_c = 1 - u$. Considere además las condiciones iniciales de posición y velocidad en $t = 0$ dadas por $x(0) = 0.994$, $y(0) = 0$, $x'(0) = 0$, $y'(0) = -2.001585106$. Sabemos a priori que el movimiento tiene periodo $T = 17.0652166$.

*a*) (3 puntos) Haga $z = x'$, $w = y'$ y obtenga un sistema de grado uno de ecuaciones diferenciales con cuatro ecuaciones de la forma $\boldsymbol{u}' = \boldsymbol{f}(t, \boldsymbol{u}(t))$, $\boldsymbol{u}(0) = \boldsymbol{u}_0$.

*b*) (3 puntos) Resuelva el sistema obtenido mediante el comando

```
[t,u] = ode113(f,[0 40],u0).
```

Grafique la curva paramétrica $\{(x(t), y(t)) : t \in [0, T_{\text{máx}}]\}$. ¿Qué observa en este caso?

*c*) (3 puntos) Defina el vector de opciones

```
options = odeset('RelTol',1e-10,'Stats','on','AbsTol',1e-10).
```

Ejecute ahora el comando `[t,u] = ode113(f,[0 40],u0,options)`. Grafique la curva paramétrica $\{(x(t), y(t)), t \in [0, T_{\text{máx}}]\}$. ¿Qué observa en este caso?

9. Este ejercicio continuamos lo realizado en el ejercicio 7 de la Tarea 3. Para lograr mejores resultados, considere el sistema

$$\left[J(\boldsymbol{c}^{(k)})^T J(\boldsymbol{c}^{(k)}) + \lambda_k \mathrm{Diag}(J(\boldsymbol{c}^{(k)})^T J(\boldsymbol{c}^{(k)}))\right] \Delta\boldsymbol{c}^{(k)} = J(\boldsymbol{c}^{(k)})^T(\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{c}^{(k)})), \qquad (1)$$

donde $\lambda_k$ se conoce como parámetro de *damping* (que garantiza que la norma del residuo disminuye en cada iteración) y $\mathrm{Diag}(J(\boldsymbol{c}^{(k)})^T J(\boldsymbol{c}^{(k)}))$ es una matriz diagonal cuyas entradas son la diagonal de $J(\boldsymbol{c}^{(k)})^T J(\boldsymbol{c}^{(k)})$. Se puede demostrar que si $\lambda_k$ toma valores altos, el método converge (pero lentamente), mientras que si $\lambda_k$ es pequeño, el método - en caso de convergencia - converge rápidamente. Note que si $\lambda_k = 0$, entonces el sistema corresponde a las ecuaciones normales del problema resuelto en la tarea. Considere así el siguiente algoritmo:

- Tome $\boldsymbol{c}^{(0)}$ dado y defina $\lambda = 1$.

- Para $k = 0, 1, \ldots$ hasta convergencia:
  - Evalúe el residuo $r_k = (\boldsymbol{g}(\boldsymbol{c}^{(k)}) - \boldsymbol{y})^T(\boldsymbol{g}(\boldsymbol{c}^{(k)}) - \boldsymbol{y})$.
  - Usando $\lambda_k = \lambda$, resuelva el sistema (1) y calcule un nuevo valor de prueba $\boldsymbol{c}^{(k+1)}$. Para este valor $\boldsymbol{c}^{(k+1)}$, calcule el nuevo residuo $r_{k+1} = (\boldsymbol{g}(\boldsymbol{c}^{(k+1)}) - \boldsymbol{y})^T(\boldsymbol{g}(\boldsymbol{c}^{(k+1)}) - \boldsymbol{y})$.
  - Usando $\lambda_k = \lambda/2$, resuelva el sistema (1) y calcule un nuevo valor de prueba $\tilde{\boldsymbol{c}}^{(k+1)}$. Para este valor $\tilde{\boldsymbol{c}}^{(k+1)}$, calcule el nuevo residuo $\tilde{r}_{k+1} = (\boldsymbol{g}(\tilde{\boldsymbol{c}}^{(k+1)}) - \boldsymbol{y})^T(\boldsymbol{g}(\tilde{\boldsymbol{c}}^{(k+1)}) - \boldsymbol{y})$.
  - Si $r_{k+1}$ es menor que $r_k$ y $\tilde{r}_{k+1}$, acepte $\boldsymbol{c}^{(k+1)}$.
  - Si $\tilde{r}_{k+1}$ es menor que $r_k$ y $r_{k+1}$, tome $\boldsymbol{c}^{(k+1)} := \tilde{\boldsymbol{c}}^{(k+1)}$ y haga $\lambda \leftarrow \lambda/2$.
  - Si $r_k$ es menor que $\tilde{r}_{k+1}$ y $r_{k+1}$, haga $\lambda \leftarrow 2\lambda$ y calcule el nuevo residuo. Continúe duplicando $\lambda$ hasta que el nuevo residuo sea menor que $r_k$. Acepte $\boldsymbol{c}^{(k+1)}$.

*a*) (6 puntos) Implemente un programa que ejecute el algoritmo descrito anteriormente. Para detener la iteración, utilice la condición $\|\Delta \boldsymbol{c}_k\|_\infty < 10^{-8}$. Además, contabilice el número de sistemas de ecuaciones que son resueltos en su código.

*b*) (3 puntos) Corra su nuevo programa para $\boldsymbol{c}_0 = [1.1; 0.4; 2.1; 0.2]$. Reporte el número de iteraciones requeridas. ¿Cuántos sistemas de ecuaciones se resolvieron?

*c*) (4 puntos)Ahora considere $\boldsymbol{c}_0 = [1, 1, 1, 1]^T$. Reporte el número de iteraciones requeridas. Grafique los nodos $\{x_i, y_i\}_{i=1}^m$, la función exacta $f(x, \boldsymbol{c}_{\text{exact}})$ y la función aproximada $f(x, \boldsymbol{c}^{(k)})$. Grafique además $\|\Delta \boldsymbol{c}^{(k)}\|_\infty$ en función de $k$. ¿Qué puede deducir sobre la velocidad de convergencia?

10. (Runge-Kuta adaptivo) En clase discutimos el método explícito de Runge-Kuta de cuarto orden, dado por el esquema:

$$\begin{array}{c|cccc}
0 & & & & \\
1/2 & 1/2 & & & \\
1/2 & 0 & 1/2 & & \\
1 & 0 & 0 & 1 & \\
\hline
 & 1/6 & 1/3 & 1/3 & 1/6
\end{array} \tag{2}$$

Para ese caso asumimos que $h$ es constante en cada iteración. A continuación describimos un método adaptivo, en donde en cada paso se determina si el valor $h$ debe ser aumentado o disminuido, según consideraciones locales de error. Asuma deseamos resolver el problema

$$\begin{cases} y' = f(t, y), & t \in [0, T_{\text{máx}}], \\ y(t_0) = y_0 \end{cases}$$

utilizando el método de Runge-Kuta explícito de orden cuatro, con incremento en el tiempo $h$ variable.

El programa incluye tres parámetros: $h_{\text{máx}}$ (el máximo posible valor de $h$, por consideraciones de estabilidad), *tol* (la tolerancia prescrita) y $h_0$ (valor inicial de $h$).

En cada iteración $k$, el error local $\Delta$ se estima de la siguiente manera. Primero, se calcula la aproximación $y_1$ (en el tiempo $t_k+h$) mediante una iteración de Runge Kuta con incremento $h$. Luego, se calcula una aproximación $y_2$ (en el tiempo $t_k+h$) utilizando dos iteraciones de Runge-Kuta con incremento $h/2$ cada una. Se toma así $\Delta = y_2 - y_1$.

El valor de $h$ se modifica de la siguiente manera. Calcule

$$\widetilde{h} = \begin{cases} Sh\left|\frac{tol}{\Delta}\right|^{0.20}, \text{si } |\Delta| \leq tol \\ Sh\left|\frac{tol}{\Delta}\right|^{0.25}, \text{si } |\Delta| > tol \end{cases}$$

donde $S$ es un factor de *seguridad* cercano a 1; en nuestro caso tome $S = 0.9$. Luego, se toma $h_{\text{new}} = \text{mín}\{h_{\text{máx}}, \widetilde{h}\}$. Finalmente, si $|\Delta| < tol$, se toma el valor $y_{k+1} = y_2 + \Delta/15$; en caso contrario se repite el ciclo. Note que en cualquier caso se continua con el valor $h_{\text{new}}$.

a) (8 puntos) Escriba una función que implemente el método adaptivo descrito anteriormente. Las entradas deben ser $f$, $y_0$, $t_0$, $T_{\text{máx}}$, $h_0$, $h_{\text{máx}}$, $tol$.

b) (4 puntos) Utilice la función anterior para el caso $f(t) = \log(t+0.01)+1/(t-4)$, $y_0 = 0.8$, $t_0 = 0$, $T_{\text{máx}} = 3.99$. Tome $h_0 = 10^{-5}$, $h_{\text{máx}} = 10^{-2}$, $tol = 10^{-12}$. Dibuje la solución $y(t)$ y $h$ en función del tiempo (en una escala adecuada). Comente las variaciones observadas en las variaciones de $h$.