MÉTODOS NUMÉRICOS: UN ENLACE ENTRE EL CÁLCULO Y LA MATEMÁTICA DISCRETA

Edison De Faria Campos
Universidad de Costa Rica, Costa Rica
edefaria@cariari.ucr.ac.cr

Resumen

En este artículo destacamos la importancia de los métodos numéricos como un elemento integrador entre el cálculo diferencial e integral, el álgebra lineal y la matemática discreta. Trataremos con métodos clásicos para la búsqueda de raíces de ecuaciones algebraicas, integración numérica mediante sumas de Riemann y métodos de Monte Carlo, curva de ajuste vía regresión y fractales. Utilizaremos las potencialidades de programación y del sistema computacional algebraico (CAS) de la calculadora graficadora voyage 200 para desarrollar las aplicaciones y los programas correspondientes.

Introducción

Las Escuela de Matemática y de Ciencias de la Computación e Informática de la Universidad de Costa Rica iniciaron durante el año 2002 un proceso de revisión de los cursos de matemática para computación, tendiente a adecuar los contenidos de los cursos a las necesidades reales de la carrera, incorporar las nuevas tecnologías de la información y comunicación como herramientas didácticas, y desarrollar proyectos conjuntos entre los docentes de las dos escuelas.

Los argumentos utilizados para proponer una reforma curricular fueron los siguientes:

1. Falta de conocimientos básicos que presentan los estudiantes al ingresar a la Universidad. Esto lleva a un alto nivel de deserción y de reprobación en los primeros cursos de matemática. Hemos sugerido un examen de ubicación para determinar las deficiencias matemáticas en los estudiantes y la necesidad de ofrecer un curso de nivelación para aquellos que presenten deficiencias en el manejo de los conocimientos básicos de matemática.
2. Falta de correlación entre los contenidos desarrollados en los cursos de matemática y los cursos propios de la carrera de computación. En este sentido hemos sugerido la introducción de ejes transversales que relacionen los contenidos de ambas disciplinas y la formación de una comisión compartida para la búsqueda de temas y metodologías adecuadas que permitan obtener este acercamiento.

   Entre los ejes transversales hemos propuesto los métodos numéricos, pues estos permiten establecer un puente entre lo continuo y el discreto.
3. Ausencia del uso de tecnologías. Creemos que el uso de tecnologías digitales en los cursos de matemática para computación facilitará el trabajo en conjunto de las dos disciplinas, logrará un aprendizaje significativo y funcionará como un elemento motivador para los estudiantes de computación (Noguera, 1998).
4. Exceso de contenidos. Sugerimos una reducción de los contenidos y una mayor profundización en aquellos que consideramos pertinentes.
5. Énfasis en procesos memorísticos. Llegamos a un consenso de que el énfasis debe de ser puesto en conjeturar, desarrollar habilidades superiores del pensamiento y resolver problemas (De Faria, 1998).

Las actividades propuestas en este curso corresponden a los contenidos de Matemática para Computación 2,3 y 4 (siglas MA0229, MA 0329, MA0429): Cálculo Diferencial e Integral para funciones de una variable, Cálculo Diferencial e Integral para funciones de varias variables y Álgebra Lineal y procuran hacer el enlace entre el cálculo continuo y el discreto.

Actividad 1 : Raíces de la ecuación $f(x)=0$
Objetivo: Programar el algoritmo de bisección, para determinar las raíces de ecuaciones algebraicas.

En esta actividad programaremos algunos de los algoritmos clásicos para determinar el valor aproximado de una raíz de una ecuación de la forma $f(x)=0$.

1. **Método de bisección**.

Este método se basa en el teorema del valor medio (Bolzano): Si f es continua en el intervalo $[a,b]$ tal que $f(a)$ y $f(b)$ tienen signos opuestos (es decir $f(a)f(b)<0$) entonces existe (al menos) un número $p \in]a,b[$ tal que $f(p)=0$. El método de bisección requiere dividir varias veces a la mitad los subintervalos de $[a,b]$ y, en cada paso, localizar la mitad que contenga p. Para simplificar supongamos que existe una única raíz en $[a,b]$, y construyamos las siguientes sucesiones: $\{a_n\}, \{b_n\}, \{p_n\}$ con $a_1 = a$, $b_1 = b$, $p_1 = \dfrac{a_1+b_1}{2}$. Si $f(p_1)=0$ tome $p=p_1$. Caso contrario $f(p_1)$ tiene el mismo signo de $f(a_1)$ o de $f(b_1)$. Si $f(p_1)f(a_1)<0$ entonces $p \in]a_1, p_1[$. En este caso tomamos $a_2 = a_1, b_2 = p_1$. Repetimos el procedimiento en el intervalo $[a_2, b_2]$. Caso contrario, $p \in]p_1, b_1[$ y $a_2 = p_1$, $b_2 = b_1$ y repetimos el procedimiento en el intervalo $[a_2, b_2]$. Podemos utilizar alguno de los siguientes criterios de paro. Dado $\varepsilon > 0$ detenga cuando:

1. $|p_n - p_{n-1}| \le \varepsilon$
2. $\dfrac{|p_n - p_{n-1}|}{|p_n|} \le \varepsilon,\ p_n \ne 0$
3. $|f(p_n)| \le \varepsilon$
4. Cuando se ha alcanzado un número máximo de iteraciones predefinido, $n = N$.

De esta forma hemos transformado un problema de calcular las raíces de una función continua definida en un intervalo compacto en un problema discreto que consiste en determinar los valores de $\{p_n\}$, n entero positivo tal que se cumpla la condición de paro.

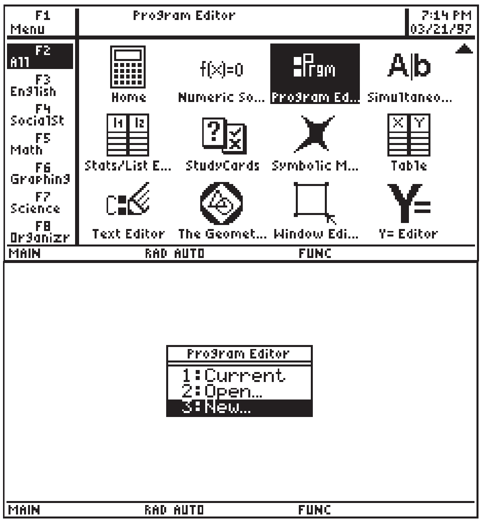
Código del programa bisección (Para la calculadora Voyage 200)
(Digitar las instrucciones separadas por / en cada línea por separado)

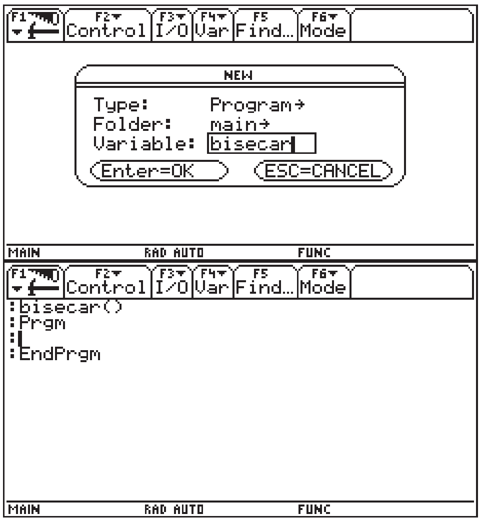
```
bisecar( )  / Prgm  / Local ex, aa, bb, ee, mm, n, p / Dialog  / Title
“bisección” /
Request “función f(x)=”,ex /  Request “extremo inferior a=”,aa /  Request
“extremo superior b=”,bb / Request “tolerancia e=”,ee / Request
“Num. Máx. Iterac.=”,mm /
EndDlog / expr(ex)→ex / expr(aa) →aa / expr(bb) →bb /
expr(ee) →ee /
expr(mm) →mm / ClrIO / 0→n / If (ex|x=aa)*(ex|x=bb)>0 Then
/
Disp “el método no se aplica en [a,b]” / Stop / EndIf / If
abs(ex|x=aa)<ee / Then / Disp “raíz aproximada” /
SetMode(“Exact/Approx”,”APROXIMATE”) /
Disp aa / Disp “Número de iteraciones” / Disp n /
SetMode(“Exact/Approx”,”AUTO”) / Stop / ElseIf
abs(ex|x=bb)<ee Then /
Disp “raíz aproximada” /  SetMode(“Exact/Approx”,”APROXIMATE”)
/ Disp bb /
Disp “Número de iteraciones” / Disp n /
SetMode(“Exact/Approx”,”AUTO”) /
Stop / EndIf / (aa+bb)/2→p / While n<mm and abs(ex|x=p)>ee
/
If (ex|x=p)*(ex|x=bb)>0 Then / p→bb / Else / p→aa / Endif
/
(aa+bb)/2→p / n+1→n / EndWhile / If abs(ex|x=p)<ee Then /
Disp “raíz aproximada” /
SetMode(“Exact/Approx”,”APROXIMATE”) /
Disp p / Disp “Número de iteraciones” / Disp n
SetMode(“Exact/Approx”,”AUTO”)
Stop / EndIf / EndPrgm
```

Pasos

1. Abra el editor de programas. Presione la tecla O, seleccione el ícono correspondiente al editor de programas y presione ÷. Abra un nuevo programa, y digite el nombre del programa: bisecar

2.





3. La barra de herramientas ha cambiado como en otros editores y ahora nos permite seleccionar las instrucciones de entrada salida o de control.
4. Digite el código del programa bisecar.El comando exp(string)$\Rightarrow$ expresión devuelve la cadena de caracteres contenida en string como una expresión y la ejecuta inmediatamente.El símbolo $\rightarrow$ se obtiene al presionar la tecla ♣ Para poner comentarios presione 2 X. Aparecerá e símbolo de comentario f. El símbolo $\subseteq$, tal que, se obtiene al presionar 2 K.
5. Ejecutar el programa.

Actividad 2: Raíces de la ecuación $f(x)=0$
Objetivo: Programar el algoritmo de Newton-Raphson, para determinar las raíces de ecuaciones algebraicas.

Método de Newton-Raphson

Para determinar una raíz de la ecuación $f(x)=0, x\in[a,b]$, con f diferenciable en $[a,b]$. Consideremos la sucesión $\{p_n\}$construida de la siguiente forma: Sea $p_0\in[a,b]$ una aproximación inicial para la raíz, y $\{p_1\}$ el punto en dónde la recta tangente a la curva $y=f(x)$ en el punto $(p_0, f(p_0))$ corta el eje x. La pendiente de la recta tangente es $f'(p_0)=\dfrac{f(p_0)}{p_0-p_1}$, si $p_1\neq p_0$. Despejando p_1 obtenemos: $p_1=p_0-\dfrac{f(p_0)}{f'(p_0)}$ si $f'(p_0)\neq 0$. Si repetimos el procedimiento y trazamos la recta tangente a la curva en el punto $(p_1, f(p_1))$ y si $f'(p_1)\neq 0$ entonces dicha recta tangente cortará el eje x en el punto p_2 tal que $p_2=p_1-\dfrac{f(p_1)}{f'(p_1)}$ si $f'(p_1)\neq 0$. De esta forma construiremos recursivamente la sucesión

$$\begin{cases} p_0 \quad \textit{aproximación inicial} \\ p_{n+1}=p_n-\dfrac{f(p_n)}{f'(p_n)} \qquad si\ \ f'(p_n)\neq 0,\ n\geq 0 \end{cases}$$

Este es el método de Newton-Raphson para aproximar una raíz de la ecuación $f(x)=0,\ x\in[a,b]$.

Código del programa Newton (para la calculadora Voyage 200)
(Digitar las instrucciones separadas por / en cada línea por separado)

```
newton( )  /  Prgm  /  Local ex, x0, ee, mm,n, p  /  Dialog  /  Title
"newton"  /
Request "función f(x)=",ex  /  Request "valor inicial x0=",x0  /
Request "tolerancia e=",ee  /  Request "Num. Máx. Iterac.=",mm  /  EndDlog
/
expr(ex)→ex  /  expr(x0) →x0  /  expr(ee) →ee  /  expr(mm) →mm  /
ClrIO  /
0→n  / x0→p / While n<mm and abs(ex|x=p)>ee  /
x0-(ex|x=x0)/(nDeriv(ex,x)|x=x0) →p  /  n+1→n  /  If abs(ex|x=p)<ee
Then  /
p→x0  /  EndIf  /  EndWhile  / Disp "raíz aproximada"  /
SetMode("Exact/Approx","APROXIMATE")  /  Disp p  /
Disp "Número de iteraciones"  /  Disp n  /
SetMode("Exact/Approx","AUTO")  /
Stop  /  EndPrgm
```

Pasos

1. Abra el editor de programas. Abra un nuevo programa, y digite newton en el campo correspondiente al nombre del programa.

2. Digite el programa.
3. Ejecute el programa.

Actividad 3: Sumas de Riemann
Objetivo: Programar el algoritmo sumas de Riemann , para determinar el valor aproximado de una integral definida.

Sumas de Riemann

Este programa es un poco más complejo que los dos anteriores. El usuario ingresa la función integrando, el intervalo de integración y el número de subintervalos de la partición del intervalo dado. Posteriormente el usuario escoge si el punto de cada subintervalo es e extremo izquierdo, el punto medio o el extremo derecho. Para lograr esto, se construye un menú con el comando **DropDown**. Este comando recibe como entrada un título (hilera de caracteres), una lista con las opciones a seleccionar, y una variable que contendrá los valores de los parámetros ingresados. Al hacer la partición el intervalo $[a,b]$ y aproximar el área bajo la curva por rectángulos, transformamos el problema de cálculo del área bajo una curva representada por una función continua por otro problema de una suma discreta de área de rectángulos.

Código del programa riemann

```
riemann( ) /  Prgm /  local a,b,c,h,n,i,r,s,v,x,y,pic1,xf,xa,xb,xn /  ClrHome /
Dialog / Text "Sumas de Riemann" / Request "f(x)=", xf / Request "a=", xa /
Request "b=", xb / Request "n=", xn / EndDlog / expr(xa) →a / expr(xb) → b /
expr(xn) → n / expr("Define y1(x)="&xf) / FnOff / ClrDraw / a → xmin /
b → xmax / (b-a)/n → h / h → xscl / FnOn / 1:ZoomFit /
min(0,ymin) → ymin / max(0,ymax) → ymax / DispG / stoPic pic1 /
While n>0 / Dialog / Text "Número subdivisiones" / Request "n=", xn /
Text "Seleccionar puntos" / DropDown "Posición", {"Inicio","Centro","Fin"},c /
EndDlog / expr(xn) → n / setMode("Exact/Approx","APPROXIMATE") /
FnOff / ClrDraw / RclPic pic1 / (b-a)/n → h / a → x /
a+(c-1)*h/2 → v / 0 → s / For i,1,n / y1(v) → y / s+y*h → s /
Line v,0,v,y / Line x,0,x,y / Line x,y,x+h,y / Line x+h,y,x+h,0 /
PxlText "    ",5,5 / PxlText "Σ="&string(s),5,5 / x+h → x / v+h → v /
EndFor / ∫(y1(x),x,a,b) → r / PxlText "∫f(x)dx="&string(r),15,5 /
PxlText " Error="&string(s-r),25,5 / setMode("Exact/Approx","AUTO") / Pause /
EndWhile / EndPrgm
```

Pasos

Abra el editor de programas. Abra un nuevo programa, y digite riemann en el campo correspondiente al nombre del programa.

Digite el programa.

Ejecute el programa.

Algunas otras actividades que vimos en el curso pero que no podemos describirlas completamente por falta de espacio son:

Actividad 4: Fractales

Objetivo: Programar Sistemas de Funciones Iteradas (IFS), como una aplicación de transformaciones lineales.

Actividad 5 : Regresión lineal
Objetivo: Obtener curvas de mejor ajuste para un conjunto de datos.

Actividad 6: Integración numérica por métodos de Monte Carlo
Objetivo: Aproximar integrales mediante métodos de Monte Carlo (simulación)

Actividad 11 : Multiplicadores de Lagrange
Objetivo: Programar el método de los multiplicadores de Lagrange para resolver problemas de máximos y mínimos para funciones de varias variables con restricciones.

Conclusiones

Creemos que la introducción de los métodos numéricos como un eje transversal en los cursos de matemática para computación servirá de motivación para los y las estudiantes y funcionará como un elemento de enlace entre las matemáticas y los cursos de programación propios de la carrera. Esto también permitirá la utilización de tecnologías digitales para programar los algoritmos correspondientes y abrirá nuevas vías de diálogo entre los docentes de ambas disciplinas. Finalmente, podemos aprovechar el potencial de las tecnologías digitales para profundizar el estudio de ciertos contenidos de los cursos mencionados e introducir aplicaciones novedosas como por ejemplo los fractales, integración mediante simulación, regresión y modelización mediante ecuaciones en derivadas parciales.

Bibliografía

Castillo W., González J., Arce C. (1997) *Álgebra Lineal*. Universidad de Costa Rica. páginas 235, 236.

De Faria E. (1998). Calculadoras gráficas, geometría y el constructivismo. Costa Rica: Revista Innovaciones Educativas, año V, No. 9, EUNED.

Noguera, N. (1998). *A Description on Tenth Grade Student's Attitudes and Cognitive Development When Leaning Algebra Using Symbolic Manipulators (TI-92's).* Tesis. Ohio: Ohio University.