Uma introdução Teórica e Numérica em alguns problemas clássicos de EDP's (PARTE 2)

Prof.: Pitágoras Pinheiro de Carvalho - **(UESPI)**

(Escola de Verão 2021 - UFPB)

26 de fevereiro de 2021

Programa de Pós Graduação em MATEMÁTICA

Sumário

Equação do Calor:

Suponha que:

- Temos $\Omega \subset \mathbb{R}^2$ um conjunto aberto, ocupado por um meio condutor de calor;
- Temos uma função u a qual descreve a temperatura em uma determinada posição (espacial) (x, y), com $((x, y); t) \in \Omega \times (0, T)$.
- Esta função $u((\underbrace{x, y}_{\text{EspaÇo}}); \underbrace{t}_{\text{Tempo}})$, indicará alteração de temperatura.

$$\boxed{u_t - k\Delta u = F((x, y); t)}$$

- Sobre este meio Ω, durante o intervalo temporal $(0, T)$, atua uma fonte de calor $F = F((x, y); t)$.

Condições de Contorno

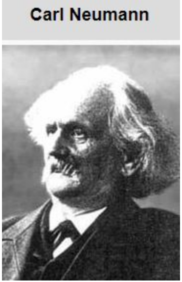


Figura: Especifica os valores da derivada normal de uma solução na fronteira do domínio.

Figura: Especifica os valores que uma solução necessita tomar na fronteira do domínio.

O problema do calor:

- Considere $\partial\Omega$ a fronteira de $\Omega \subset \mathbb{R}^2$.

A equação do calor: $\boxed{u_t + k\Delta u = F((x,y);t)}$

Condição de Dirichlet: $\boxed{u\Big|_{\partial\Omega} = g(x,y)}$

Condição de Neumann: $\boxed{\frac{\partial u}{\partial n}\Big|_{\partial\Omega} = h(x,y)}$

- As funções F, g e h são dadas, k é constante e a solução u é desconhecida.

Sistemas de equação do calor:

- EXEMPLOS:

$$\begin{cases} u_t - \Delta u = f & \text{in } \Omega \times (0, T), \\ u = 0 & \text{on } \partial\Omega \times (0, T), \\ u(x, 0) = u_0(x) & \text{in } \Omega. \end{cases} \tag{1}$$

$$\begin{cases} u_t - \Delta u = f & \text{in } \Omega \times (0, T), \\ \dfrac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega \times (0, T), \\ u(x, 0) = u_0(x) & \text{in } \Omega. \end{cases} \tag{2}$$

$$\begin{cases} u_t - \Delta u = f & \text{in } \Omega \times (0, T), \\ u = 0 & \text{on } \partial\Omega_1 \times (0, T), \\ \dfrac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega_2 \times (0, T), \qquad \big(\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2\big) \\ u(x, 0) = u_0(x) & \text{in } \Omega. \end{cases} \tag{3}$$

Solução Fraca:

Vamos considerar o problema de calor:

$$\begin{cases} u_t - \Delta u = f & \text{em } \ \Omega \times (0, T), \\ u = 0 & \text{sobre } \ \partial\Omega \times (0, T), \\ u(x,0) = g & \text{em } \ \Omega := \Omega \times \{t = 0\}, \end{cases} \tag{4}$$

onde:

- $\Omega \subset \mathbb{R}^n$ é um conjunto aberto, e o tempo $T > 0$;
- $\Omega_T = \Omega \times (0,\ T)$;
- $f : \Omega \times (0,\ T) \to \mathbb{R}$ é dada;
- $g : \Omega \to \mathbb{R}$ é dada;
- $u = u(x,t)$ é desconhecida, com $\underbrace{\mathbf{x} \in \mathbb{R}^n}_{x=(x_1,\ldots,x_n)}$ e $t \in (0,\ T)$.

Função Teste ϕ

- $C_0^\infty(\Omega)$ denota o espaço das funções infinitamente diferenciáveis

$$\phi : \Omega \to \mathbb{R},$$

com suporte compacto em Ω. **Obs.:** O suporte de uma função é o menor subconjunto fechado do domínio onde a função não é nula. Se o suporte for compacto, a função é de suporte compacto.

- Uma função $\phi \in C_0^\infty(\Omega)$ será chamada de função teste.

E como vamos usar isso?

- Considere $u \in C^1(\Omega)$, $\phi \in C_0^\infty(\Omega)$ e $u\Big|_{\partial\Omega} = 0$.
Usando Gauss-Green + Integração por partes, temos que:

$$\boxed{\int_\Omega u\phi_{x_i}\, dx = -\int_\Omega u_{x_i}\phi\, dx + \underbrace{\int_{\partial\Omega} u\phi\nu^i\, dS}_{=0}, \qquad (i = 1, \ldots, n).}$$

Uma reformulação para $(u_t - \Delta u = f)$:

- Considere o operador

$$Lu = -\sum_{i,j=1}^{n} (a^{i,j}(x,t)u_{x_i})_{x_j} \ .$$

- Associado ao operador linear L, considere a forma bilinear $B[.,.;.]$ sendo

$$B[u,\ v;\ t] := \int_{\Omega} \sum_{i,j=1}^{n} a^{i,j}(\cdot,t)u_{x_i}v_{x_j}$$

- Tomemos uma função f. Dizemos que u, é uma **solução fraca** do problema (4), se para cada v-função teste, vale:

$$(u_t,\ v)\ +\ B[u,\ v;\ t] = (f,\ v),$$

onde $(\ ,\ )$ denota o produto interno em $L^2(\Omega)$.

Um breve resumo do passo a passo:

1. Considere o nosso problema:

$$\boxed{u_t - \Delta u = f, \qquad u\Big|_{\partial\Omega} = 0}$$

   com $f \in L^2(\Omega)$.

2. Multiplique o problema por uma função teste $v \in C_0^\infty(\Omega)$;
3. Integre sobre Ω;
4. Obtenha:

$$\boxed{\int_\Omega u_t v + \nabla u \nabla v \, dx = \int_\Omega f v \, dx.} \qquad (\star)$$

5. A identidade $(\star)$ é chamada de Formulação Variacional de (4).

Existência de Solução Fraca:

- (Método de Galerkin)
  1. Construção de soluções aproximadas u_m em espaços de dimensão finita;
  2. Posteriormente, passagem ao limite $m \to \infty$.
- Teoremas que garantem a possibilidade da contrução das soluções aproximadas;
- Obtenção de estimativas de Energia para as soluções aproximadas u_m;
- Teoremas de existência e unicidade de soluções.

Formulação Variacional:

- Devemos encontrar $u \in H_0^1(\Omega)$ tal que:

$$\int_\Omega u_t v - \Delta u v = \int_\Omega f v,$$

$\forall v \in H_0^1(\Omega)$.

- Ou ainda, na formulação variacional de **(4)**, devemos encontrar $u \in H_0^1(\Omega)$, tal que

$$\int_\Omega u_t v + \nabla u \cdot \nabla v - \int_\Omega f v = 0, \quad \forall v \in H_0^1(\Omega).$$

Dados da malha:

- O nome $\mathcal{T}_h$ (Th no FreeFEM) refere-se a família $\{T_k\}_{k=1,\ldots,n_t}$ de triangulos.

- Se Ω não é um domínio poligonal, um erro ínfimo permanece entre o domínio exato e sua aproximação
$$\Omega \simeq \Omega_h = \bigcup_{k=1}^{n_t} T_k.$$

- Um espaço de elemento finito é, geralmente, um espaço de funções polinomiais em elementos (triângulos aqui apenas), com propriedades correspondentes nas arestas, vértices etc...

- Aqui o fespace $Vh(Th, P\mathbf{?})$ define V_h como o espaço de funções que são afins em (x, y) para cada triângulo de T_h.

As funções de aproximação (Polinômios P?)

- A base canônica é feita de funções que são chamadas hat ϕ_k, que são contínuas por partes, iguais a 1 em um vértice e 0 em todos os outros.

- As funções hat (ϕ_k) são definidas fazendo uso das coordenadas baricêntricas em cada ponto $P \in T_k$.

- Assim:

$$V_h(\mathcal{T}_h, P?) = \left\{ w(x,y) \middle| \, w(x,y) = \sum_{k=1}^{M} w_k \phi_k(x,y), \; w_k \text{ são números reais} \right\}$$

sendo M a dimensão de V_h.

- $Vh \ u \ v$, declara que u e v são aproximadas por:

$$\boxed{u(x,y) \simeq u_h(x,y) = \sum_{k=0}^{M-1} u_k \phi_k(x,y)}$$

Quais serão os Polinômios?

- P0 piecewise constant discontinuous finite element. The degrees of freedom are the barycenter element value.
- P1 piecewise linear continuous finite element. The degrees of freedom are the vertices values.
- P2 piecewise continuous quadratic finite element. The degrees of freedom are the vertices values and the middle point of each edge.
- P3 piecewise continuous cubic finite element. (need : load "Element_P3")
- P4 piecewise continuous quartic finite element. (need : load "Element_P4")
- P1b piecewise linear continuous plus bubble. [1]
- P2b piecewise quadratic continuous plus bubble. [1]
- P1dc piecewise linear discontinuous finite element.
- P2dc piecewise P2 discontinuous finite element.

A lista completa dos polinônios na documentação do Freefem++.
Para os espaços H^s ver as referências [3] e [4].

Problema Aproximado (Método de Galerkin):

- Para o problema do calor, devemos encontrar $u_h \in V_h$ tal que:

$$\int_\Omega (u_h)_t v_h - \Delta u_h v_h = \int_\Omega f v_h,$$

$\forall v_h \in V_h$.

- Ou ainda, no problema aproximado para **(4)** devemos encontrar $u_h \in V_h$, tal que

$$\int_\Omega (u_h)_t v_h + \nabla u_h \cdot \nabla v_h - \int_\Omega f v_h = 0, \quad \forall v_h \in V_h,$$

sendo

$$V_h = \left\{ (u_h,\ v_h)\Big|_T \ para\ cada\ T \in T_h \right\}.$$

Observação: T_h é a malha do domínio decomposto em triângulos T.

Equação do Calor - 2D

- Considere a equação do calor atuando em $\Omega = (\sin(t), \cos(t))$, com $t \in (0, 2\pi)$.

$$\begin{cases} \dfrac{\partial u}{\partial t} - \mu \Delta u = F \quad \text{em} \quad \Omega \times (0, T), \\ u(x, y, \underbrace{0}_{\textit{tempo}}) = f \quad \text{em} \quad \Omega, \\ u = g \quad \text{sobre} \quad \partial\Omega. \end{cases} \qquad \textbf{(2)}$$

onde temos: $\underbrace{F = x \cdot y}_{\text{termo força}}, \quad \underbrace{f = 1}_{\text{condição inicial}}, \quad \underbrace{g = 0}_{\text{condição de fronteira}}, \quad \underbrace{t = m \cdot \tau}_{\text{tempo}}.$

- A derivada parcial da solução na variável temporal é dada por:

$$\frac{\partial u}{\partial t}(x, y, t) = \lim_{\tau \to 0} \frac{u(x, y, t) - u(x, y, t - \tau)}{\tau} .$$

- Como $t = m\tau$, logo $t - \tau = (m - 1)\tau$.

Discretização (**Diferenças Finitas em t**):

- Isto indica que $u^m(x,y) = u(x,y,m\tau)$, deverá satisfazer (aproximadamente):

$$\frac{\partial u}{\partial t}(x,y,m\tau) \simeq \frac{u^m(x,y) - u^{m-1}(x,y)}{\tau}.$$

Assim, $(\forall m = 0, \ldots, T/\tau)$ **a discretização da equação do calor no tempo é dada por**:

$$\begin{cases} \dfrac{u^{m+1}-u^m}{\tau} - \mu\Delta u^{m+1} = F^{m+1} & \text{em} \quad \Omega \times (0,3), \\ u^0(x,y) = u_0(x,y) & \text{em} \quad \Omega, \\ u^{m+1} = g & \text{sobre} \quad \partial\Omega. \end{cases} \qquad \textbf{(3)}$$

Observação: A discretização em **(3)** é chamada de método de **Euler** "regressivo no tempo" para **(2)**. Mais detalhes na referência [2] (Pags: 26 e 27).

Formulação Variacional:

- Multiplicando o sistema **(3)** pela função teste v,

$$\left(\frac{u^{m+1}-u^m}{\tau}-\mu\Delta u^{m+1}\right)v = F^{m+1}v$$

e integrando sobre Ω, obtemos

- $$\int_\Omega u^{m+1}v+\tau\mu(\nabla u^{m+1}\cdot\nabla v) = \int_\Omega u^m v+\tau F^{m+1}v \ .$$

- Ou melhor, a formulação variacional a ser programada no Freefem++:

$$\int_\Omega\{u^{m+1}v+\tau\mu(\nabla u^{m+1}\cdot\nabla v)\}-\int_\Omega\{u^m v+\tau F^{m+1}v\} = 0 \ ,$$

com $\quad u^{m+1}\Big|_{\partial\Omega} = g.$

O problema e o programa:

- O problema:

$$\int_\Omega \{u^{m+1}v + \tau\mu(\nabla u^{m+1}\cdot\nabla v)\} - \int_\Omega \{u^m v + \tau F^{m+1} v\} = 0\ ,$$

com $\quad u^{m+1}\Big|_{\partial\Omega} = g.$

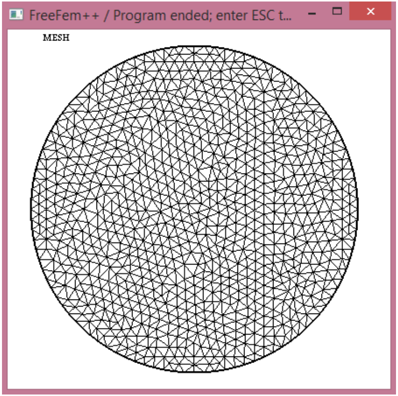
- O programa:

problem **Heat** (u, v) = int2d(Th)(u*v + τ*mu*(dx(u)*dx(v) + dy(u)*dy(v)))
- int2d(Th)(τ* fff*v)
+ on($\partial\Omega$, u=g);

fff será justificado no que segue...

```
//——— Heat-Linear.edp ———
load "medit"
verbosity=0;

//———- NUMERICAL VALUES TO PROVIDE ———
int C0; // the label
int Ntimes=100;
real T=2., tau=T/Ntimes, error, errFP=1.e-10, y00=1.;
int itermax=125;

// ———- AUXILIAR VARIABLES ———
int kiter=0, jiter=0, itermm1=itermax-1, Nt1=Ntimes+1,Ntm1=Ntimes-1, n1;
real[int] Uerror(1000);
```

```
// —— DESCRIPTION OF THE BOUNDARIES ——
border C00(t=0,2*pi){x=10*cos(t); y=10*sin(t); label=C0;}

//—— MESH ——
mesh Th=buildmesh(C00(80));
```



```
//—SOME FUNCTIONS ——
func fcont=x*y;
func y0=y00;

//——— THE FINITE ELEMENT SPACE AND FUNCTIONS ———
fespace Vh(Th,P1);
Vh sol, fff, phi, soly;
Vh[int] yy(Nt1), yyn(Nt1), ff(Nt1);

//———THE SYSTEM ——-
problem
states(sol,phi)=int2d(Th)(sol*phi+tau*(dx(sol)*dx(phi)+dy(sol)*dy(phi)))
    - int2d(Th)(fff*phi) + on(C0,sol=0.);
```

```
// —SOLVING THE PROBLEM (ITERATES)——
for (int k=0;k < itermax;k++)
{
    kiter=k; yy[0]=y00;
    for (int n=0;n < Ntimes;n++)
    {
    n1=n+1; ff[n]=fcont;
    fff=yy[n]+tau*ff[n1];
    states;
    yy[n1]=sol;
    error= sqrt(int2d(Th)(sol*sol));
    }
    Uerror[kiter]=error;
    cout << "ITER="<< kiter << "ERROR="<< error << endl;
    if (error <= error*error)
    {
    cout << "CONVERGENCE - ITER="<< kiter << endl;
    break;
    }
}
```

```
//—- PLOTS —–
for (int n=0; n <= Ntimes; n++)
{
plot(yy[n], cmm="STATE - TIME STEP"+n, value=true, fill=true, dim=2,
nbiso=30, wait=false);
}

plot(yy[1], cmm="STATE - TIME STEP 1", value=true, fill=true, dim=2,
nbiso=30, wait=true);

plot(yy[50], cmm="STATE - TIME STEP 50", value=true, fill=true, dim=2,
nbiso=30, wait=true);

plot(yy[100], cmm="STATE - TIME STEP 100", value=true, fill=true, dim=2,
nbiso=30, wait=true);
```

Nossas primeiras figuras:

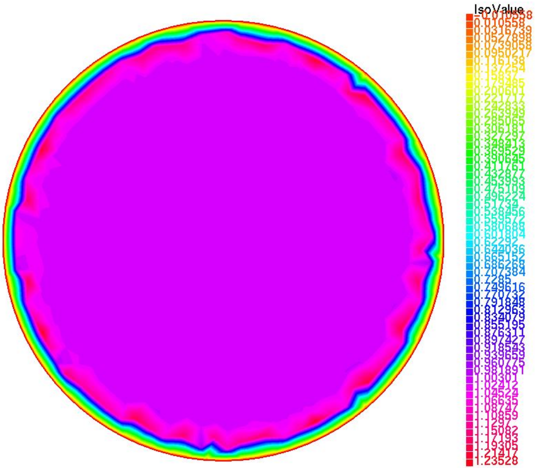


Figura: Plot u no tempo $t = 0.02$

Nossas primeiras figuras:

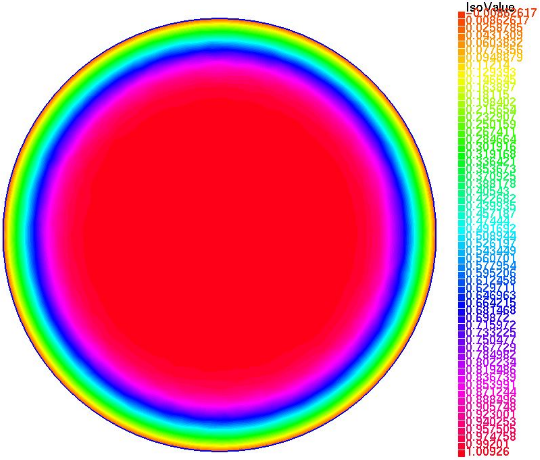


Figura: Plot u no tempo $t = 1$.

Nossas primeiras figuras:

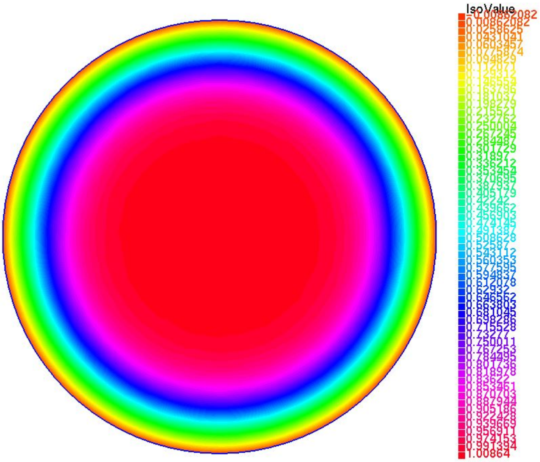
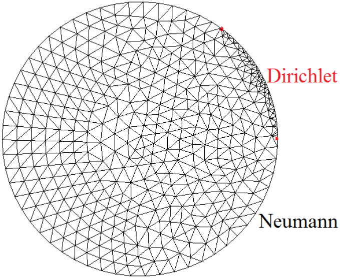


Figura: Plot u no tempo $t = 2$.

Dirichlet + Neumann (2D):

// Parametros (D e N):

```
load "medit"

int C0, C1;

border C00(t=0,0.3*pi){x=10*cos(t); y=10*sin(t); label=C0;}

border C10(t=0.3*pi,2*pi){x=10*cos(t); y=10*sin(t); label=C1;}

plot(C00(50)+C10(50));

mesh Th=buildmesh(C00(50) + C10(50));

plot(Th);

medit("2D Mesh", Th);
```



Referências Bibliográficas

F. Hecht. New development in freefem++. J. Numer. Math., 20(3-4):251–265, 2012.

M. Rincon, I. Liu. Introdução ao método de elementos finitos. Computação e análise em equações diferenciais parciais, 2020.

Ph . CLÉMEN T. APPROXIMATION BY FINITE ELEMENT FUNGTIONS USING LOCAL REGULARIZATION. https://www.esaim-m2an.org/articles/m2an/abs/1975/02/m2an197509R200771/m2an197509R200771.html

P. CIARLET, Jr. Analysis of the Scott–Zhang interpolation in the fractional order Sobolev spaces. https://sci-hub.se/https://doi.org/10.1515/jnum-2013-0007