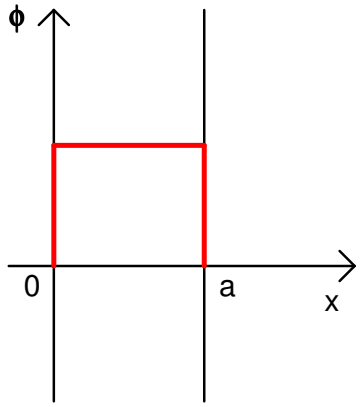
PROBLEMA DI DIFFUSIONE MONODIMENSIONALE IN TRANSITORIO

- **Problema fisico:**

  evoluzione della distribuzione del flusso neutronico in uno slab assorbente a partire da una distribuzione iniziale piatta con condizioni di annullamento agli estremi



- **Formulazione matematica:**

$$\frac{1}{v}\frac{\partial \phi}{\partial t} = D\frac{\partial^2 \phi}{\partial x^2} - \Sigma_a \phi$$

$$\phi(0,t) = \phi(a,t) = 0 \qquad \phi(x,0) = 1$$

Valore dei parametri:

$\Sigma_a = 0.08\ cm^{-1}$ $\qquad D = 0.4\ cm$

$v = 2.2 \times 10^5\ cm$ $\qquad a = 10\ cm$

- **Formulazione numerica:**

$$\frac{1}{v}\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \alpha\left(D\frac{\phi_{i+1}^{n+1} - 2\,\phi_i^{n+1} + \phi_{i-1}^{n+1}}{\Delta x^2} - \Sigma_a\,\phi_i^{n+1}\right)$$

$$+(1-\alpha)\left(D\frac{\phi_{i+1}^{n} - 2\,\phi_i^{n} + \phi_{i-1}^{n}}{\Delta x^2} - \Sigma_a\,\phi_i^{n}\right)$$

$$(i = 1, ..., N) \quad ; \quad (n = 0, 1, ...)$$

$$\phi_0^n = \phi_{N+1}^n = 0 \quad (n = 0, 1, ...) \; ; \quad = 1 \qquad (i = 1, ..., N)$$

Si ha:

- $\alpha = 0$: **metodo esplicito**
- $\alpha = 0.5$: **metodo Crank-Nicolson**
- $\alpha = 1$: **metodo implicito**

Ponendo:

$$a_i = c_i = -\frac{\alpha\, v\, \Delta t\, D}{\Delta x^2} \qquad b_i = 1 + \alpha\, v\, \Delta t\left(\frac{2\,D}{\Delta x^2} + \Sigma_a\right)$$

$$d_i = \phi_i^n + (1-\alpha)\, v\, \Delta t\left(D\frac{\phi_{i+1}^{n} - 2\,\phi_i^{n} + \phi_{i-1}^{n}}{\Delta x^2} - \Sigma_a\,\phi_i^{n}\right)$$

Si ha:

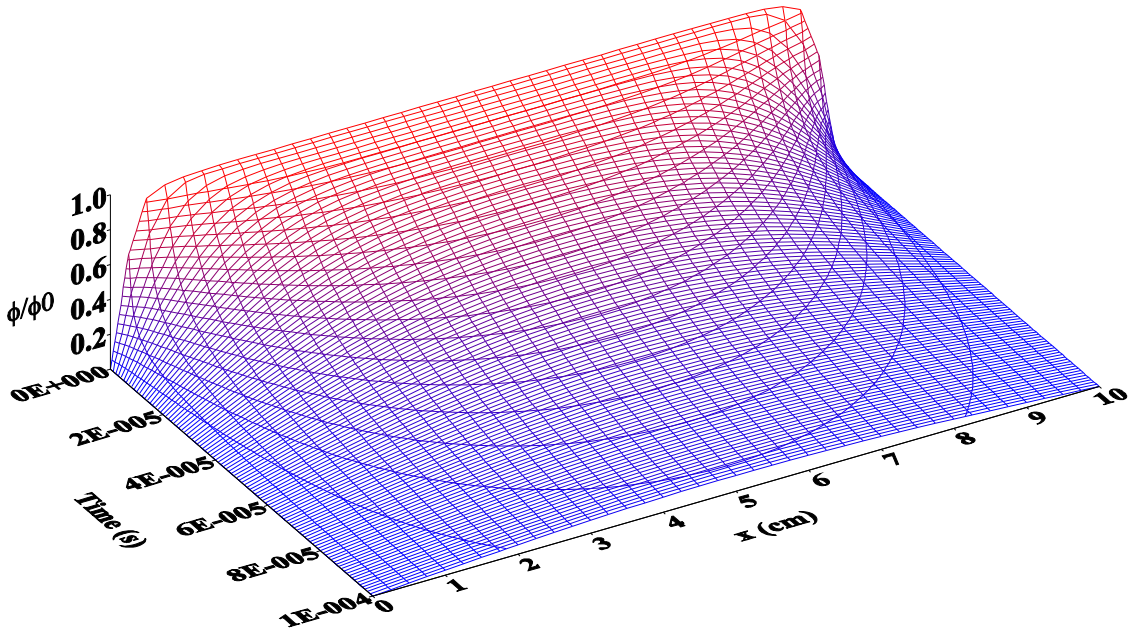
$$a_i\,\phi_{i-1}^{n+1} + b_i\,\phi_i^{n+1} + c_i\,\phi_{i+1}^{n+1} = d_i \qquad (i = 1, ..., N)$$

Si pone: $N = 39 \quad \Rightarrow \quad \Delta x = 0.25$ cm

Esercizi proposti

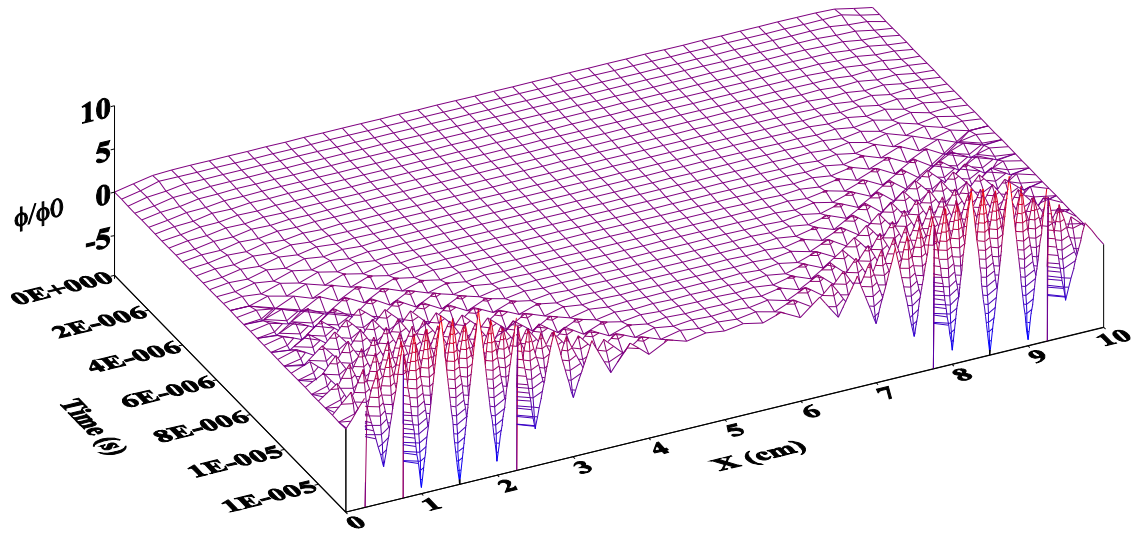
1. Studiare il transitorio riguardante la diffusione di neutroni a partire da una distribuzione di flusso uniforme nello strato:
   a) con il metodo esplicito ($\alpha = 0$) con passi temporali di 2×10^{-7} s;
   b) con il metodo esplicito ($\alpha = 0$) con passi temporali di 4×10^{-7} s (superiore al limite di stabilità ($\approx 3.54 \times 10^{-7}$ s);
   c) con il metodo implicito ($\alpha = 1$) ed il metodo di Crank-Nicholson ($\alpha = 0.5$) con passi temporali di 1×10^{-6} s o maggiori.

Risultati attesi:



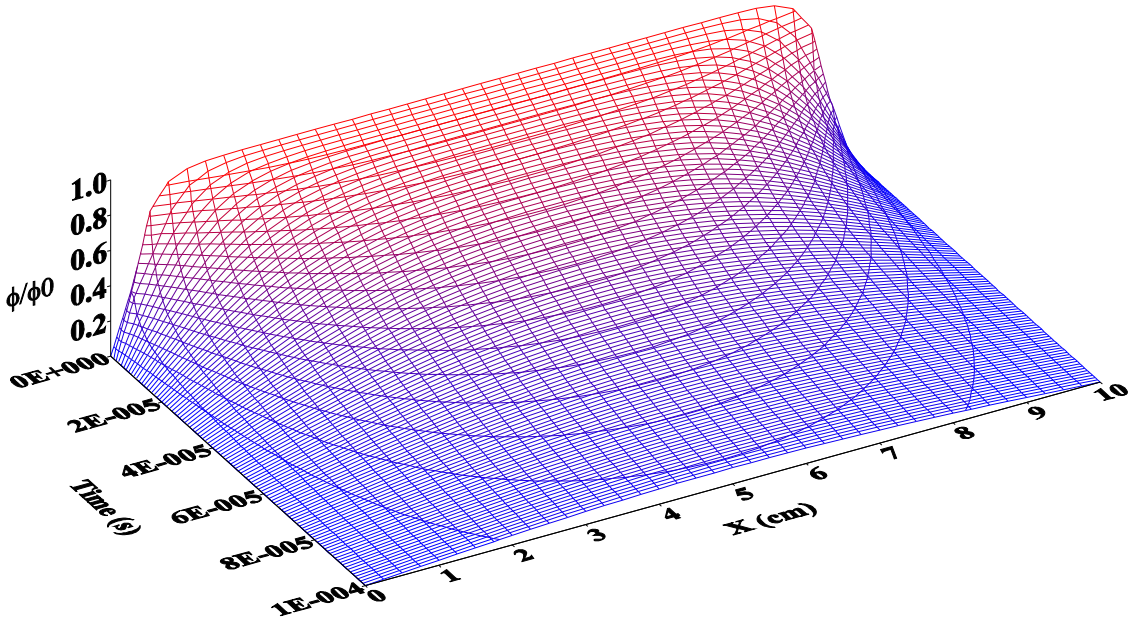
Explicit Method - Time step = 2.e-7 s

Stability Limit: Time step < 3.54e-7 s



Explicit Method Time step = 4.e-7 s

Stability Limit: Time step < 3.54e-7 s



Crank Nicolson Method - Time step = 1.e-6 s

Unconditionally Stable

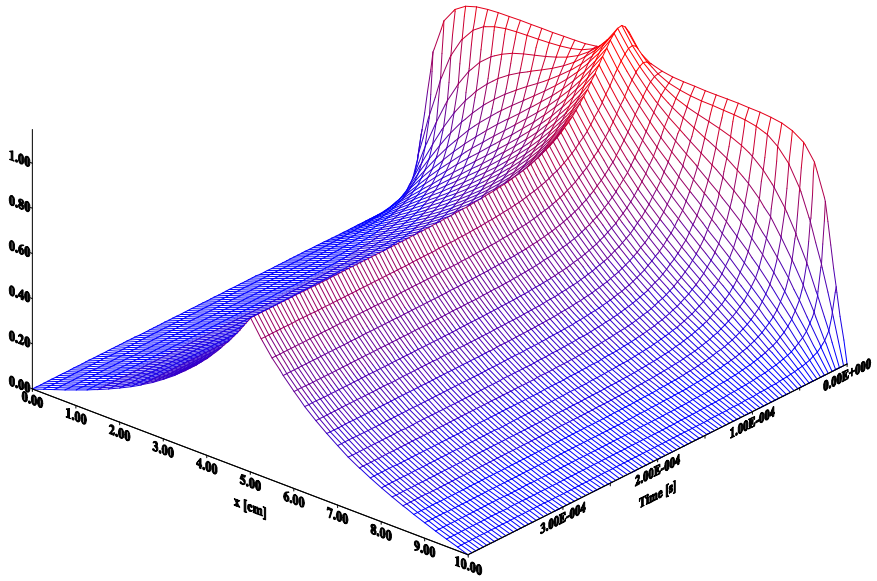
2. Studiare il transitorio riguardante la diffusione di neutroni a partire da una distribuzione di flusso uniforme nello strato nel caso di sorgente a delta di Dirac posta in diverse posizioni nello strato. Confrontare i risultati con la predizione analitica data da (v. Fisica del Reattore):

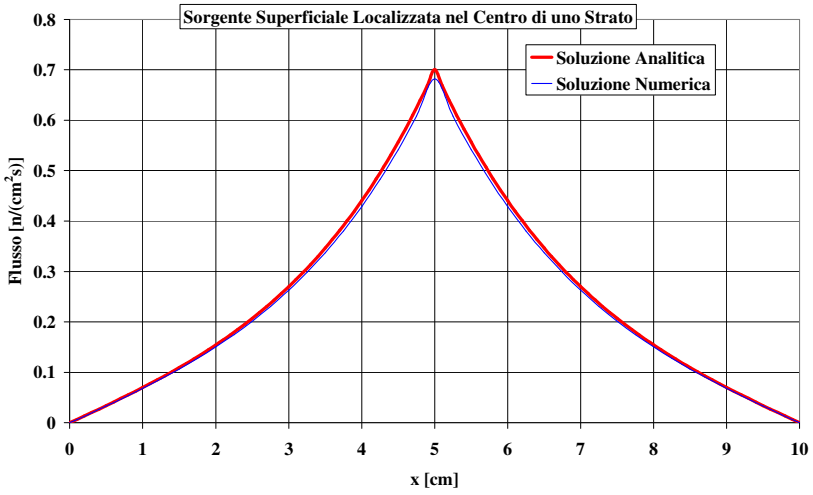
$$\phi^{+} = \frac{S}{\frac{D}{L}\sinh\frac{a}{L}} \sinh\frac{x'}{L}\sinh\frac{a-x}{L}, \qquad 0 \le x' \le x \le a$$

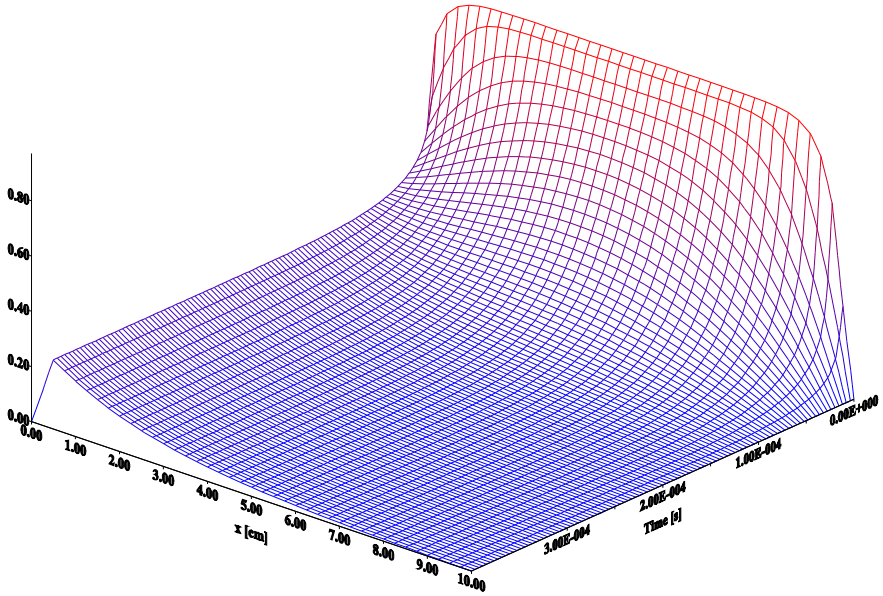
$$\phi^{-} = \frac{S}{\frac{D}{L}\sinh\frac{a}{L}} \sinh\frac{a-x'}{L}\sinh\frac{x}{L}, \qquad 0 \le x \le x' \le a$$

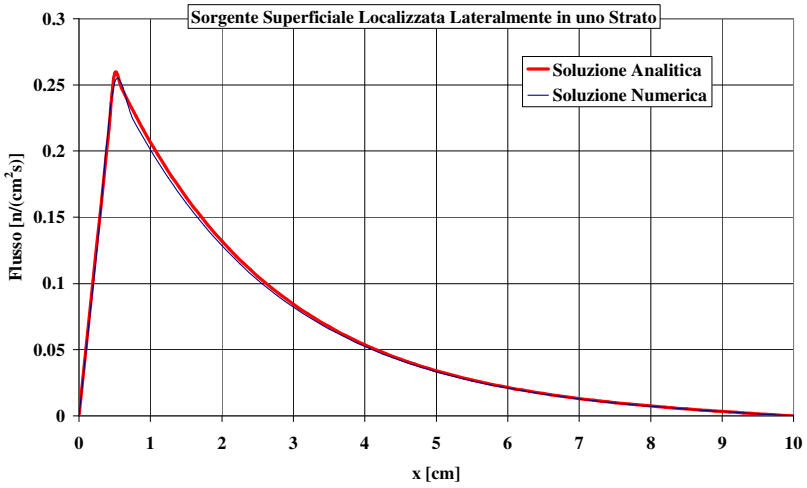
Risultati attesi:

a) <u>sorgente nel centro dello strato</u>









3. Modificare il programma considerando altri tipi di sorgente (ad es., uniforme, sinusoidale) e/o disponendo "barre di controllo" (cioè nodi con Σ_a più elevata) in varie posizioni.

   Suggerimento

   Definire una "function" che assegni il valore della sorgente in ogni nodo sulla base di diverse opzioni in ingresso. Similmente, assegnare i valori della sezione d'urto d'assorbimento in nodi particolari.

4. Modificare il programma includendo una sorgente di fissione ($\nu\Sigma_f\phi$) e studiare la dinamica del sistema in assenza di neutroni ritardati.

FORTRAN ROUTINE

```
c--------------------------------------------------------------------------c
c                                                                          c
c    1D Trasient Neutron diffusion with sources                            c
c    Program set up for teaching purposes only                             c
c    W. Ambrosini - Università di Pisa                                     c
c                                                                          c
c--------------------------------------------------------------------------c
      program nonsta
      implicit double precision (a-h,o-z)
      character*80 riga
c
      parameter (m = 50)
      dimension phi(0:m+1),x(0:m+1),xsour(10),sour(10),isour(10)
      dimension a(m),b(m),c(m),d(m),v(m),alef(m),bet(m)
c
      open (unit=5,file='nonsta.dat')
      open (unit=6,file='nonsta.txt')
c
      read (5,100) riga
      read (5,*) aleng,diff,sigma,vel,n
c
      read (5,100) riga
      read (5,*) dt,tend,alpha
c
      read (5,100) riga
      read (5,*) nsour
c
      read (5,100) riga
c
      do 5 is = 1,nsour
      read (5,*) xsour(is),sour(is)
    5 continue
c
      read (5,100) riga
      read (5,*) xbar,sigbar
c
      np1 = n + 1
      dx = aleng / dfloat (np1)
      cmpalp = 1.d00 - alpha
      veldt = vel * dt
      sigdt = sigma * dt
      dx2 = dx * dx
c
      phi(0) = 0.d00
      phi(np1) = 0.d00
      x(0) = 0.d00
      x(np1) = aleng
c
      do 10 i = 1,n
c
      phi(i) = 1.d00
      x(i) = dx * dfloat(i)
c
      halfdx = 0.5d00 * dx
c
        do j = 1,nsour
        if(dabs(xsour(j)-x(i)).le.halfdx) isour(j) = i
        enddo
c
        if(dabs(xbar-x(i)).le.halfdx) ibar = i
c
```

```
      auxsig = 0.d00
      if(i.eq.ibar) auxsig = sigbar
c
      a(i) = - veldt * alpha * diff / dx2
      b(i) = 1.d00 + veldt * alpha * ( 2.d00 * diff / dx2
     &                  + sigma + auxsig )
      c(i) = a(i)
c
   10 continue
c
c   Loop on time advancement
c
      time = 0.d00
      do 50 it = 1,5000000
      time = time + dt
c
      do 20 i = 1,n
      im1 = i - 1
      ip1 = i + 1
      d2phi = phi(ip1) - 2.d00 * phi(i) + phi(im1)
C
      auxsig = 0.d00
      if(i.eq.ibar) auxsig = sigbar
      d(i) = phi(i) + veldt * cmpalp * ( diff * d2phi / dx2
     &                      - ( sigma + auxsig ) * phi(i) )
c
       do j = 1,nsour
       if(i.eq.isour(j)) d(i) = d(i) + veldt * sour(j)
       enddo
c
   20 continue
c
      call tdma (a,b,c,d,v,alef,bet,n,m)
c
      phimax = 0.d00
      do 30 i = 0,np1
      if((i.ne.0).and.(i.ne.np1)) phi(i) = v(i)
      write(6,110) time,x(i),phi(i)
      if(phimax.lt.dabs(phi(i))) phimax = phi(i)
   30 continue
      if((time.ge.tend).or.(phimax.gt.10.d00)) goto 60
c
   50 continue
   60 continue
c
      stop
  100 format(a80)
  110 format(3(1x,e14.7))
      end
c-----------------------------------------------------------------------c
c                                                                       c
c   TDMA Algorithm                                                      c
c                                                                       c
c-----------------------------------------------------------------------c
      subroutine tdma (a,b,c,d,v,alef,bet,n,ld)
      implicit double precision (a-h,o-z)
      dimension a(ld),b(ld),c(ld),d(ld),v(ld),alef(ld),bet(ld)
      ub=1.d00/b(1)
      alef(1)=c(1)*ub
      bet(1)=d(1)*ub
      do 10 i=2,n
      l=i-1
      qz=b(i)-a(i)*alef(l)
```

```
      uqz=1.d00/qz
      alef(i)=c(i)*uqz
   10 bet(i)=(d(i)-a(i)*bet(l))*uqz
      nm1=n-1
      v(n)=bet(n)
      do 20 i=1,nm1
      ii=n-i
      l=ii+1
   20 v(ii)=bet(ii)-alef(ii)*v(l)
      return
      end
```