© Jean-Baptiste APOUNG KAMGA <jean-baptiste.apoung@math.u-psud.fr>

Fiche de TP: Transport Equation, Conservation laws

Thème - 1 *Transport equations and finite differences methods*

We wish to numerically solve the following scalar transport equation

$$\begin{cases} \dfrac{\partial u}{\partial t} + c\dfrac{\partial u}{\partial x} & = & 0, \quad x \in \mathbb{R} \quad t > 0 \\ \\ u(0,x) & = & u_0(x) \quad x \in \mathbb{R} \end{cases} \tag{1}$$

where c is a fixed real and $u_0 \in C^1$ is a given scalar function. To this end we require the finite difference method consisting of computing u_j^n, the approximate values of the solution of equation (1) at positions $(x_j = jh, t_n = nk)(j \in \mathbb{Z}, n \geq 0)$, where h (resp. k) denotes the space grid size (resp. the time grid size). The initial solution is taken to be the Gaussian : $u_0(x) = e^{-x^2}, x \in \mathbb{R}$.

Exercice-1 **:** Consider the following explicit schemes:

$$\frac{1}{k}\left(u_j^{n+1} - u_j^n\right) + \frac{c}{h}\left(u_{j+1}^n - u_j^n\right) = 0, \quad j \in \mathbb{Z}, n \geq 0 \tag{2}$$

$$\frac{1}{k}\left(u_j^{n+1} - u_j^n\right) + \frac{c}{h}\left(u_j^n - u_{j-1}^n\right) = 0, \quad j \in \mathbb{Z}, n \geq 0 \tag{3}$$

$$\frac{1}{k}\left(u_j^{n+1} - u_j^n\right) + \frac{c}{2h}\left(u_{j+1}^n - u_{j-1}^n\right) - \frac{kc^2}{2h^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right) = 0, \quad j \in \mathbb{Z}, n \geq 0. \tag{4}$$

Perform the l^2 stability analysis of each of them and specify the influence of the sign of c.

Exercice-2 **:** Write a **`matlab`** program using these schemes on a bounded domain $]-A, A[$ with a fixed $\frac{k}{h} := \lambda$. Display the approximate solution at a given final time T. *Whenever necessary, consider a fixed $h = A/M$ where M is prescribed and zero or periodic boundary conditions.*

Exercice-3 **:** Recall the stability results when $c = \pm 1$ and comment the results obtained when $c\lambda = \pm 1$.

Remark 0.0.1.

- Observe that all the above schemes can be written as :

$$u_j^{n+1} = \alpha u_{j-1}^n + \beta u_j^n + \gamma u_{j+1}^n, \quad j \in \mathbb{Z}, n \geq 0 \tag{5}$$

  where α, β, γ depend on h, k with $\alpha + \beta + \gamma = 1$.

- Since $\alpha + \beta + \gamma = 1$ we have :

$$|\alpha e^{-i\theta} + \beta + \gamma e^{+i\theta}|^2 = 1 - 4\Big(\beta(\alpha+\gamma) + 4\alpha\gamma\Big)\sin^2(\frac{\theta}{2}) + 32\alpha\gamma\sin^4(\frac{\theta}{2}) \quad \forall\theta \in [-\pi, \pi]$$

- **Definition:** When $\alpha + \beta + \gamma = 1$, the scheme is said to **`preserve constants`**.

- **Definition:** When $\alpha \geq 0, \beta \geq 0, \gamma \geq 0$, the scheme is said to be **`monotone`**:

$$(v_j^n > w_j^n \quad \forall j \in \mathbb{Z}) \Longrightarrow (v_j^{n+1} > w_j^{n+1} \quad \forall j \in \mathbb{Z}).$$

- **Proposition:** When the scheme is monotone and preserves constants, it is l^∞ **stable**:

$$\forall T > 0,\ \exists C_T > 0,\ \sup_{nk<T}\left(\sup_{j\in\mathbb{Z}}|v_j^n|\right) \leq C_T\left(\sup_{j\in\mathbb{Z}}|v_j^0|\right).$$

  (*See Theme 5* for other informations).

To implement the schemes with periodic boundary conditions, just write a Matlab function

Listing 1: Flux file : FDTransport.m

```
function [uu] = FDTransport(u,alpha,beta,gamma)
 n  = length(u);
 uu = zeros(n,1);
 Ig = [n,1:n-1];
 Id = [2:n,1];
 for j = 1:n
   uu(j) = alpha * u(Ig(j)) + beta * u(j) + gamma *  u(Id(J));
 end
 % uu = alpha * u(Ig) + beta * u + gamma * u(Id) % ceci evite boucle et initialissation de ↩
     uu
end
```

or in more vectorial way

Listing 2: Flux file : FDTransport.m

```
function [uu] = FDTransport(u,alpha,beta,gamma)
 uu = alpha * circshift(u,[0,1]) + beta * u + gamma * circshift(u,[0,-1]) ;
end
```

The main function can look like :

Listing 3: Flux file : testFDTransport.m

```
function [res] = testFDTransport(M)

clf;

c = -1;
lambda = 0.9/abs(c);
A =10;
h = A/M;
x = -A:h:A;
dt = h*lambda;

u = exp(-10*x.*x);          % condition initiale
plot(x,u,'*-');             % dessin de la solution initiale


% (Downwind)
alpha = 0;
beta  = 1 + c * lambda;
gamma = -c* lambda ;
scheme1 = @(u) FDTransport(u,alpha,beta,gamma);


% (Upwind)
alpha = c * lambda;
beta  = 1 - c * lambda;
gamma = 0 ;
scheme2 = @(u) FDTransport(u,alpha,beta,gamma);


% (Lax-Wendroff)
alpha = (1 + c * lambda)/2;
beta  = 0;
gamma = (1 - c * lambda)/2;
scheme3 = @(u) FDTransport(u,alpha,beta,gamma);


u1 = u;
u2 = u;
u3 = u;

for i =1:2000

  t = 0 + i*dt;
  u1 = scheme1(u1);
  u2 = scheme2(u2);
  u3 = scheme3(u3);

  figure(1)
  %subplot(1,2,1);
  plot(x,u1,'+', x,u3,'-');
  legend('scheme1','scheme3'); drawnow;
  s = sprintf('Downwind and Lax-wendroff c = %f',c);
  title(s);
  drawnow;

  %subplot(1,2,2);
  figure(2)
  plot(x,u2,'+', x,u3,'-');
  legend('scheme2','scheme3');
  title(sprintf('Upwind and Lax-wendroff c = %f',c));
  drawnow;

end
res = 0;
end
```

Consider now a more general equation:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial f(u)}{\partial x} &= 0, \quad x \in \mathbb{R} \quad t > 0 \\ \\ u(0,x) &= u_0(x) \quad x \in \mathbb{R} \end{cases} \tag{6}$$

As in the previous exercise, we wish to find approximate values of $u(t_n, x_i), j \in \mathbb{Z}, n \in \mathbb{N}$, using the finite volume methods. To this end, we introduce the intermediate points $x_{i+\frac{1}{2}}, i \in \mathbb{Z}$ an assume that *at each instant* t_n*, the solution is constant on each interval* $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], i \in \mathbb{Z}$.

This constant value is equal to: $u_i(t_n) = \frac{1}{h_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(t_n, x)\, dx$ where $h_i = |x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}|$.

Thus, to construct the finite volume scheme, we integrate the equation (6) over the interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ where we approximate the solution by its mean value : $u_i(t) = \frac{1}{h_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(t,x)\, dx$. We then obtain (assuming a constant mesh size $h_i = h \quad \forall i$):

$$h\frac{du_i(t)}{dt} + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial f(u)}{\partial x}\, dx = 0. \tag{7}$$

This is equivalent to

$$h\frac{du_i(t)}{dt} + f(t)^*_{i+\frac{1}{2}} - f(t)^*_{i-\frac{1}{2}} = 0, \tag{8}$$

where $f(t)^*_{i\pm\frac{1}{2}}$ called **numerical flux** is the approximate value of $f(u)$ at position $x_{i\pm\frac{1}{2}}$ at time t. The main difficulty carried out by the finite volume scheme is the effective construction of those numerical fluxes. Since u is constant on the left (resp. on the right) of $x_{i+\frac{1}{2}}$ with value u_i (resp. u_{i+1}), the numerical flux $f^*_{i+\frac{1}{2}}$ will thus depend on $u^L_{i+\frac{1}{2}}$ and $u^R_{i+\frac{1}{2}}$ which are the best approximate value of u on the left and the right of $x_{i+\frac{1}{2}}$. This lead to the following expression:

$$f^*_{i+\frac{1}{2}} \equiv f^*_{i+\frac{1}{2}}(u^L_{i+\frac{1}{2}}, u^R_{i+\frac{1}{2}}). \tag{9}$$

We are then left with the problem of constructing for each i, the quantities $u^L_{i+\frac{1}{2}}, u^R_{i+\frac{1}{2}}$ and $f^*_{i+\frac{1}{2}}$, under the particular constraint of **consistency** of the numerical fluxes, which states that $f^*(v,v) = f(v)\,\forall v$. It is not unusual that more properties such as monotony or Lipschitz character of these fluxes are sought.

Let us now enumerate particular numerical fluxes and the corresponding schemes.

Exercice-1 **:** **Centered Scheme**. Here the numerical flux is given by

$$f^*_{i+\frac{1}{2}} = f\left(\frac{u_i + u_{i+1}}{2}\right).$$

Implement this scheme and evaluate your implementation over the transport equation (i.e. $f(u) = cu$). Try to justify the observed behavior.

Exercice-2 : **Upwind Scheme :** this scheme corresponds to the following expression of the numerical fluxes

$$f^*_{i+\frac{1}{2}} = \begin{cases} f(u_i) & \text{if } \frac{\partial f}{\partial u}(\frac{u_i+u_{i+1}}{2}) \geq 0, \\ f(u_{i+1}) & \text{if } \frac{\partial f}{\partial u}(\frac{u_i+u_{i+1}}{2}) < 0. \end{cases}$$

Implement this scheme and validate your implementation on the transport equation.

Exercice-3 : **Scheme with fluxes of Lax-Friedrichs type**

It is based on the observation that if one has a good approximation of $u^L_{i+\frac{1}{2}}$ et $u^R_{i+\frac{1}{2}}$, then one can define the fluxes as

$$f^*_{i+\frac{1}{2}} = \frac{1}{2}\left(f(u^L_{i+\frac{1}{2}}) + f(u^R_{i+\frac{1}{2}}) - a_{i+\frac{1}{2}}(u^R_{i+\frac{1}{2}} - u^L_{i+\frac{1}{2}})\right),$$

with $a_{i+\frac{1}{2}} = \max\left(\varrho\left(\frac{\partial f(u_i)}{\partial u}\right), \varrho\left(\frac{\partial f(u_{i+1})}{\partial u}\right)\right)$, where $\varrho(A)$ denotes the spectral radius of the matrix A.

Depending on the choice of $u^L_{i+\frac{1}{2}}$ and $u^R_{i+\frac{1}{2}}$, various schemes are generated, the most common being:

1. **Scheme with local Lax-Friedrichs flux** : $u^L_{i+\frac{1}{2}} = u_i, \quad u^R_{i+\frac{1}{2}} = u_{i+1}$.
2. **Centered Kurganov-Tadmor scheme with flux limiter**

$$\begin{cases} u^L_{i+\frac{1}{2}} &= u_i + \frac{\phi(r_i)}{2}(u_{i+1} - u_i), \\ u^R_{i+\frac{1}{2}} &= u_{i+1} - \frac{\phi(r_{i+1})}{2}(u_{i+2} - u_{i+1}), \\ r_i &= \frac{u_i - u_{i-1}}{u_{i+1} - u_i}. \end{cases}$$

Various famous flux limiters are provided in the literature and some are listed below :

	Van-Albada (2003)	Ospre (1995)	charm (1995)	minmod (1986)
$\phi(r)$	$\frac{2r}{1+r^2}$	$1.5\frac{r^2+r}{r^2+r+1}$	$\begin{cases} \frac{r(3r+1)}{(r+1)^2} & \text{si } r > 0 \\ 0 & \text{si } r \leq 0 \end{cases}$	$\max(0, \min(1, r))$

Implement these schemes and evaluate your implementation on the transport equation where $f(u) = cu$.

Thème - 3 *Some ideas on the implementation*

For a problem like (6), the function $f(t, u)$ (we consider the more general case where f also depends explicitly on t) will be called the flux function. Its implementation will be supplied in a file **`flux.m`** and defined as follows:

Listing 4: Flux file: flux.m

```
function [f,fprime] =flux(t,x)
c = 1;
f = x*x/2.;%c * x;
fprime = x;%c;
```

Note that we have chosen to return also partial derivatives of $f(\cdot, \cdot)$ with respect to its second argument. This will simplify the implementation of upwind schemes.

The equation (8) can be written as a system of ordinary differential equations.

$$\frac{du}{dt} = F(t,u), \quad \text{où} \quad u(t) = (u_1(t), \ldots, u_n(t))^T.$$

In this form, one can apply any numerical schemes for ODE to find an approximate solution.This approach, consisting of freezing the space during the discretization, in order to obtain a system of ODEs is referred to as **Method Of Lines (MOL)**.

In order to implement the second member of the transport equation discretized using the finite volumes method with periodic boundary conditions ($F(t,u)$ in our case), one can consider the following function, which takes a limiter as its third argument:

Listing 5: Transport file: transport.m

```
function [ures] = transport(t,u,limit)

  n = length(u); % nombre de mailles
  ures = u;
  for i=1:n
    % TRAITEMENT DANS LA MAILLE I de numero i
    im = indexg(i,n);   % numero de la maille à  gauche c-a-d  i-1
    ip = indexd(i,n);   % numero de la maille à  droite c-a-d  i+1
    uim = u(im);        % valeur de la solution dans la maille à  gauche
    ui  = u(i);         % valeur de la solution dans la maille i
    uip = u(ip);        % valeur de la solution dans la maille à  droite
    [fi,fi_prime] = flux(t,ui); % f(u_i) et df/du (u_i)

    % FLUX A DROITE  NOTE ICI fstarp
    [fip,fip_prime] = flux(t,uip);
    ap = max(abs([fi_prime,fip_prime]));
    uipp = u(indexd(ip,n));
    %ri = (ui - uim)/(uip - ui);
    riup = (ui - uim);
    ridown = (uip - ui);
    %rip = (uip -ui)/(uipp - uip);
    ripup = (uip -ui);
    ripdown = (uipp - uip);
    % calcul des valeurs à gauche (upL) et à  droite (upR) de u en xp
    upL = ui + 0.5*limit(riup,ridown)*(uip - ui);
    upR = uip - 0.5*limit(ripup,ripdown)*(uipp - uip);

    fstarp = 0.5 *( flux(t,upR)+flux(t,upL) - ap *(upR-upL));

    % FLUX A GAUCHE NOTE ICI fstarm
    [fim,fim_prime] = flux(t,uim);
    am = max(abs([fi_prime,fim_prime]));
    uimm = u(indexg(im,n));
    %rim = (uim -uimm)/(ui - uim);
    rimup = (uim -uimm);
    rimdown = (ui - uim);
    %ri = (ui - uim)/(uip - ui);
    riup = (ui - uim);
    ridown = (uip - ui);
    umL = uim + 0.5 * limit(rimup,rimdown)*(ui - uim);
    umR = ui  - 0.5 * limit(riup,ridown)*(uip - ui);

    fstarm =    0.5 * ( flux(t,umR)+flux(t,umL) - am *(umR-umL));

    % STOCKAGE
    ures(i) =-1.0 *(fstarp - fstarm);
  end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% functions retournant les numeros des mailles voisines
% sous hypothèse des conditions aux limites périodiques

%%%%%%%%%%%% NUMERO MAILLE A GAUCHE %%%%%%%%%%%%%%%%%%%%%%
function j=indexg(i,n)
  if(i>=2)
```

```
    j = i-1;
  else
    j = n;
  end
end
%%%%%%%%%%%% NUMERO MAILLE A DROITE %%%%%%%%%%%%%%%%%%%%%
function j = indexd(i,n)
  if(i<n)
    j=i+1;
  else
    j=1;
  end
end
```

As the limiters are concerned, examples are furnished below. It should be noted that in order to prevent errors in the evaluation of $\phi(r)$ with $r = (v_1 - v_2)/(v_3 - v_4)$, we have chosen to define ϕ as a function of two variables, the first being the numerator and the second the denominator. So instead of writing $\phi((v_1 - v_2)/(v_3 - v_4))$ we have chosen to write $\phi(v_1 - v_2, v_3 - v_4)$. This can help handle singular situations such as those where $v_3 = v_4$.

Listing 6: Minmod file: minmod.m

```
function y=minmod(rup,rdown)
  if(abs(rdown) <1e-16)
    y=1;
  else
    r= rup/rdown;
    y = max(0,min(1,r));
  end
end
```

Listing 7: Ospre file: ospre.m

```
function y=ospre(rup,rdown)
  if(abs(rdown) <1e-16)
    y=1.5;
  else
    r= rup/rdown;
    y = 1.5*(r*r+r)/(r*r + r + 1);
  end
end
```

In order to test the programs on the transport equation with explicit Euler time advancing, a sample main program could be :

Listing 8: Test file: testtransport.m

```
function testtransport()
  clear all;
  x =linspace(-10,10,200); % sommets du maillage
  dx = x(2)-x(1);          % pas uniforme d'espace
  dt = 0.5 * dx;           % pas de temps
  lambda = dt/dx;          %
  xplot = (x(1:end-1) + x(2:end))/2; % centres des mailles
  u = exp(-10*xplot.*xplot);         % condition initiale
  uu = u;                            %
  plot(xplot,u,'*-');                % dessin de la solution initiale

  for i =1:2000
    t = 0 + i*dt
    u  = u  + lambda * transport(t,u, @minmod);
    uu = uu + lambda * transport(t,uu,@ospre);
    plot(xplot,u,'+',xplot,uu,'-');
    legend('minmod','opspre');
    drawnow;
    %pause();
  end
end
```

Note: *Two functions are furnished in file* **`transport.m`** *in order to handle the connectivity between cells for periodic boundary conditions. For extended usage, It would be more practical to move them in two files* **`indexg.m`** *and* **`indexd.m`**.

Exercice-1 : Modify the function **`function [uu] =transport(t,u, limit)`** to make it more vectorial (i.e suppress loops)

Exercice-2 **:** Modify the main program in order to use the **`Matlab`** ODE solvers.

Thème - 4 *Applications*

Apply the Centered Kurganov-Tadmor scheme to one of the following problems:

Exercice-1 **: Reactive transport in porous media**

Consider

$$\begin{cases} \partial_t C_1 + V\partial_x C_1 - D\partial^2_{xx}C_1 + K_1(C_1,x)C_1 = f_1(C_2,x) \;\; sur \;\;]0,T[\times]0,L[, \\ \partial_t C_2 + V\partial_x C_2 - D\partial^2_{xx}C_2 + K_2(C_2,x)C_2 = f_2(C_1,x) \;\; sur \;\;]0,T[\times]0,L[, \end{cases}$$

where,

$$\begin{array}{ll} K_1(C,x) = \frac{V_m^1 x}{K_h^1+C}\delta_1, & f_1(C,x) = -\kappa_{12}K_2(C,x)C, \\ K_2(C,x) = \frac{V_m^2 x}{K_h^2+C}\delta_2, & f_2(C,x) = -\kappa_{21}K_1(C,x)C. \end{array}$$

Boundary conditions are given by:

$$\begin{array}{ll} C_1(t,0) = 3.0, & \partial_x C_1(t,L) = 0, \\ C_2(t,0) = 10., & \partial_x C_2(t,L) = 0. \end{array}$$

The initial conditions are:

$$C_1(0,x) = 3.0, \qquad C_2(0,x) = 0.0$$

In this exercise, we consider the following parameters:

$$\begin{array}{ll} V_m^i = 1.0/day, i = 1,2 & \kappa_{12} = 2.0, \\ K_h^i = 1.0mg/L, i = 1,2 & \kappa_{21} = 0.5. \end{array}$$

We also consider the following values

$$\begin{array}{lll} V = 1.0m/day, & D = 0.2m^2/day, & L = 100m. \\ \delta_1 = 1, & \delta_2 = 1. & \end{array}$$

Display in the same picture the curves of C_1 and C_2 at time $t = 68$ days.

Exercice-2 **: A drying model problem**

Let $\mathbf{A} \in \mathbb{R}^4 \times \mathbb{R}^4$ and $\mathbf{g} \in R^4$. We wish to find $\mathbf{u} : [0,T] \times [0,Z] \to \Omega \in \mathbb{R}^4$ such that:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{u}}{\partial z} & = & \mathbf{g}(\mathbf{u}), \in]0,T]\times]0,Z] \\ u(0,z) & = & u^0(z), \quad z \in [0,Z] \\ u(t,0) & = & u_0(t), \quad t \in [0,T] \end{cases} \tag{10}$$

where $Z > 0, T > 0, \mathbf{u} = (u_1,u_2,u_3,u_4)^t$ is the vector of states variables ; $\mathbf{A} = diag(1, \frac{\mu}{\nu}, 1, \frac{\mu}{\nu})$ and each component of the application $\mathbf{u} \to \mathbf{g}(\mathbf{u})$ is defined by:

$$
\begin{aligned}
g_{1S}(\mathbf{u}) &= \theta_{1S} - \frac{\beta_S(u_4 + \lambda u_2 - \lambda\zeta)}{\sigma_G - u_2} + \frac{\gamma_S u_3}{\sigma_S + u_1} \\
g_{1G}(\mathbf{u}) &= \theta_{1G} - \frac{\beta_G(u_4 + \lambda u_2 - \lambda\zeta)}{\sigma_G - u_2} + \frac{\gamma_G u_3}{\sigma_S + u_1} \\
g_{1G}(\mathbf{u}) &= \theta_{2S} - \frac{\delta_S(u_4 + \lambda u_2 - \lambda\zeta)}{\sigma_G - u_2} - \frac{\varepsilon_S u_3}{\sigma_S + u_1} \\
g_{2G}(\mathbf{u}) &= \theta_{2G} - \frac{\delta_G(u_4 + \lambda u_2 - \lambda\zeta)}{\sigma_G - u_2} + \frac{\varepsilon_G u_3}{\sigma_S + u_1}
\end{aligned}
\tag{11}
$$

Sample values for the physical parameters are given by the table below:

Listing 9: Paramètres physiques

```
%       mu =  18.548
%       nu =  1
%       betaS =  5.6180
%       gammaS =  5.6180
%       sigmaS =  0.20000
%       detaS =  0.044440
%       gammaS =  5.6180
%       sigmaS =  0.20000
%       deltaS =  0.044440
%       epsilonS =  0.044440
%       betaG = -5.6180
%       gammaG = -5.6180
%       sigmaG = -0.70370
%       deltaG = -0.098770
%       epsilonG = -0.098770
%       theta1S =  3.3307e-16
%       theta1G = -3.3307e-16
%       theta2S = -1.7347e-18
%       theta2G = 0
%       Z =  2
%       T =  0.75586
%       tauS = 0.17778;
%       tauG = 0.17778;
%       lambda = 0.01758
%       zeta   = 0.11111
```

For this study the initial and the boundary conditions are the following:

$$u^0 = u_0 = (1, 0, 0.21333, -0.83076)^t.$$

The components u_1, u_2 of $\mathbf{u}$ denote respectively humidity content of the solid and the gas. The components u_3, u_4 are the density of energy transfer. They are related to the temperature T_S of the of the solid and the temperature T_G of the gas by the formulas:

$$T_S = \frac{u_3}{\sigma_S + u_1} - \tau_S \qquad \text{and} \qquad T_G = \frac{u_4 + \lambda u_2 - \lambda\zeta}{\sigma_G - u_1} - \tau_G$$

where τ_S and τ_G are two constants. For the present study case, their value will be :

$$\tau_S = \tau_G = 0.17778.$$

We wish to observe the evolution in time and space of the variables u_1, u_2, T_S, T_G.

Tous les schémas qui sont présentés peuvent s'écrire sous la forme compacte $(P_{k,h}v)_j^n = 0$, $j \in \mathbb{Z}$, $n \in \mathbb{N}$, *relativement à la résolution du problème continu* $Pu = 0$.

Définition *(Consistance). Le schéma aux différences finies* $(P_{k,h}v)_j^n = 0$ *est dit consistant avec le problème* $Pu = 0$ *si pour toute fonction régulière* $\phi := \phi(t, x)$,

$$P\phi - P_{k,h}\phi \to 0$$

si $k \to 0$ *et* $h \to 0$*, la convergence étant au sens ponctuel en chacun des points de discrétisation.*

Définition *(Ordre). Le schéma aux différences finies* $(P_{k,h}v)_j^n = 0$ *consistant avec le problème* $Pu = 0$ *est dit précis à l'ordre* p *en temps et* q *en espace si pour toute fonction régulière* $\phi := \phi(t, x)$ *telle que* $P\phi = 0$*, on a*

$$P_{k,h}\phi = \mathcal{O}(k^p + h^q).$$

Définition *(Stabilité). On dit que le schéma aux différences finies* $P_{k,h}v = 0$ *est stable au sens* l^2 *si, en posant* $T = Nk$ *le temps final prescrit,*

$$\|v^n\| \leq C_T \|v^0\|, \quad \forall n \neq N,$$

où C_T *est une constante qui ne dépend éventuellement que de* T *et*

$$\|v\| = \left(\sum_{j=-\infty}^{+\infty} |v_j|^2 \right)^{1/2}.$$

Définition*(Norme dans* $l^2(\mathbb{Z})$*). Pour des suites dans* $l^2(\mathbb{Z})$*, on introduit les normes*

$$\|v\|_{l^2(\mathbb{Z})} = \|v\|_2 = \left(\sum_{j\in\mathbb{Z}} |v_j|^2 \right)^{1/2}, \qquad \|v\|_{l^2(h\mathbb{Z})} = \|v\|_h = \left(h \sum_{j\in\mathbb{Z}} |v_j|^2 \right)^{1/2} = \sqrt{h}\|v\|_2.$$

Définition *(Transformée de Fourier). Soit* $\widehat{\cdot}$ *l'application définie par*

$$\begin{array}{cccc} \widehat{\cdot} \; : & l^2(h\mathbb{Z}) & \to & L^2\left(\left]-\frac{\pi}{h}, \frac{\pi}{h}\right[\right) \\ & v & \mapsto & \widehat{v}, \end{array}$$

avec

$$\widehat{v}(\xi) = \sum_{j\in\mathbb{Z}} e^{-ijh\xi} v_j h, \quad \xi \in \left[-\frac{\pi}{h}, \frac{\pi}{h}\right].$$

Il est également possible de reconstruire la suite v *si* $\widehat{v}$ *est connue sur* $[-\pi/h, \pi/h]$ *: si* $j \in \mathbb{Z}$ *est fixé, on multiplie en effet l'équation par* $e^{ijh\xi}$ *et on intègre sur l'intervalle* $[-\pi/h, \pi/h]$*. On obtient*

$$\int_{-\pi/h}^{\pi/h} \widehat{v}(\xi)\, \delta\xi = h \int_{-\pi/h}^{\pi/h} v_j\, \delta\xi + h \sum_{k\neq j} \int_{-\pi/h}^{\pi/h} e^{-i(j-k)h\xi} v_k\, \delta\xi.$$

Par périodicité, on obtient

$$v_j = \frac{1}{2\pi} \int_{-\pi/h}^{\pi/h} e^{-ijh\xi} \widehat{v}(\xi)\, \delta\xi.$$

Proposition *(Formule de Parseval). Pour* $v \in l^2(h\mathbb{Z})$*, on a*

$$\frac{1}{2\pi}\int_{-\pi/h}^{\pi/h} |\widehat{v}(\xi)|^2\, \delta\xi = \|v\|_h^2.$$

Proposition *(Transformée de Fourier et suite translatée). Le paramètre* $l \in \mathbb{Z}$ *étant fixé, la suite* $\tau_l v$ *définie par* $(\tau_l v)_j := v_{j+l}$ *pour* $v \in l(h\mathbb{Z})$ *est telle que*

$$\widehat{\tau_l v}(\xi) = e^{ilh\xi}\widehat{v}(\xi), \xi \in \left[-\frac{\pi}{h}, \frac{\pi}{h}\right].$$

Théorème *(CNS de stabilité (1)). Un schéma aux différences finies est stable au sens* l^2 *si et seulement s'il existe une constante* C *telle que le coefficient d'amplification* g *vérifie* $|g(h\xi)| \leq 1 + Ck$*,* $\forall \xi \in [-\pi/h, \pi/h]$*.*

Théorème *(CNS de stabilité (2)). Si* g *ne dépend pas explicitement de* k *et* h*, alors une CNS pour assurer la stabilté au sens* l^2 *du schéma aux différences finies est* $|g(h\xi)| \leq 1$*,* $\forall \xi \in [-\pi/h, \pi/h]$*.*