```
> restart;
  with(PDEtools):
  with(plots):
  with(LinearAlgebra):
```

Finite difference solution of the advection equation

The purpose of this worksheet is to discuss the various finite difference stencils one can use to solve the diffusion equation

$$\frac{\partial}{\partial t}u(t,x) + c\frac{\partial}{\partial x}u(t,x) = 0.$$

Here, c is the speed of propagation. In this equation, t plays the role of a time variable and x is a spatial variable; that is, we are going to solve the PDE as an initial value problem.

Finite difference stencils

```
> pde := diff(u(t,x),t)+c*diff(u(t,x),x);
```

$$pde := \frac{\partial}{\partial t}u(t,x) + c\left(\frac{\partial}{\partial x}u(t,x)\right) \quad \textbf{(1.1}$$

We will make use of the `GenerateStencil` procedure derived elsewhere to find our finite difference stencils of **(1.1)**:

```
> GenerateStencil := proc(F,N,{orientation:=center,stepsize:=h,showorder:=true,showerror:=
  false})
      local vars, f, ii, Degree, stencil, Error, unknowns, Indets, ans, Phi, r, n, phi;

      Phi := convert(F,D);
      vars := op(Phi);
      n := PDEtools[difforder](Phi);
      f := op(1,op(0,Phi));
      if (nops([vars])<>1) then:
        r := op(1,op(0,op(0,Phi)));
      else:
        r := 1;
      fi:
      phi := f(vars);
      if (orientation=center) then:
        if (type(N,odd)) then:
          ii := [seq(i,i=-(N-1)/2..(N-1)/2)];
```

```
        else:
          ii := [seq(i,i=-(N-1)..(N-1),2)];
        fi;
      elif (orientation=left) then:
        ii := [seq(i,i=-N+1..0)]:
      elif (orientation=right) then:
        ii := [seq(i,i=0..N-1)]:
      fi;
      stencil := add(a[ii[i]]*subsop(r=op(r,phi)+ii[i]*stepsize,phi),i=1..N);
      Error := D[r$n](f)(vars) - stencil;
      Error := convert(series(Error,stepsize,N),polynom);
      unknowns := {seq(a[ii[i]],i=1..N)};
      Indets := indets(Error) minus {vars} minus unknowns minus {stepsize};
      Error := collect(Error,Indets,'distributed');
      ans := solve({coeffs(Error,Indets)},unknowns);
      if (ans=NULL) then:
         print(`Failure: try increasing the number of points in the stencil`);
         return NULL;
      fi:
      stencil := subs(ans,stencil);
      Error := convert(series(`leadterm`(D[r$n](f)(vars) - stencil),stepsize,N+20),polynom);
      Degree := degree(Error,stepsize);
      if (showorder) then:
         print(cat(`This stencil is of order `,Degree));
      fi:
      if (showerror) then:
         print(cat(`This leading order term in the error is `,Error));
      fi:
      convert(D[r$n](f)(vars) = stencil,diff);

  end proc:
```

Here are the particular sub-stencils we will need:

```
> forward_time := GenerateStencil(diff(u(t,x),t),2,orientation=right,stepsize=s);
  backward_time := GenerateStencil(diff(u(t,x),t),2,orientation=left,stepsize=s);
  centered_time := GenerateStencil(diff(u(t,x),t),2,orientation=center,stepsize=s);
  centered_space := GenerateStencil(diff(u(t,x),x),2,orientation=center,stepsize=h);
  right_space := GenerateStencil(diff(u(t,x),x),2,orientation=left,stepsize=h);
```

This stencil is of order 1

$$forward\_time := \frac{\partial}{\partial t} u(t,x) = -\frac{u(t,x)}{s} + \frac{u(t+s,x)}{s}$$

This stencil is of order 1

$$backward\_time := \frac{\partial}{\partial t} u(t,x) = -\frac{u(t-s,x)}{s} + \frac{u(t,x)}{s}$$

This stencil is of order 2

$$centered\_time := \frac{\partial}{\partial t} u(t,x) = -\frac{1}{2}\frac{u(t-s,x)}{s} + \frac{1}{2}\frac{u(t+s,x)}{s}$$

This stencil is of order 2

$$centered\_space := \frac{\partial}{\partial x} u(t,x) = -\frac{1}{2}\frac{u(t,x-h)}{h} + \frac{1}{2}\frac{u(t,x+h)}{h}$$

This stencil is of order 1

$$right\_space := \frac{\partial}{\partial x} u(t,x) = -\frac{u(t,x-h)}{h} + \frac{u(t,x)}{h} \quad (1.2$$

The FTCS stencil is constructed using the forward time and centered space stencil:

```
> Label[1] := `FTCS`;
  stencil[1] := subs(forward_time,centered_space,pde);
```

$$Label_1 := FTCS$$

$$stencil_1 := -\frac{u(t,x)}{s} + \frac{u(t+s,x)}{s} + c\left(-\frac{1}{2}\frac{u(t,x-h)}{h} + \frac{1}{2}\frac{u(t,x+h)}{h}\right) \quad (1.3$$

The BTCS stencil is constructed using the backward time and centered space stencil:

```
> Label[2] := `BTCS`;
  stencil[2] := subs(backward_time,centered_space,t=t+s,pde);
```

$$Label_2 := BTCS$$

$$stencil_2 := -\frac{u(t,x)}{s} + \frac{u(t+s,x)}{s} + c\left(-\frac{1}{2}\frac{u(t+s,x-h)}{h} + \frac{1}{2}\frac{u(t+s,x+h)}{h}\right) \quad (1.4$$

As usual, the Crank-Nicholson method is the average of the above two:

```
> Label[3] := `Crank-Nicholson`;
  stencil[3] := (stencil[1]+stencil[2])/2;
```

$$Label_3 := Crank\text{-}Nicholson$$

$$stencil_3 := -\frac{u(t,x)}{s} + \frac{u(t+s,x)}{s} + \frac{1}{2}c\left(-\frac{1}{2}\frac{u(t,x-h)}{h} + \frac{1}{2}\frac{u(t,x+h)}{h}\right) + \frac{1}{2}c\left(-\frac{1}{2}\frac{u(t+s,x-h)}{h}\right. \quad (1.5$$

$$+\frac{1}{2}\frac{u(t+s,x+h)}{h}\Bigg)$$

The upwind stencil uses one-sided approximations for both derivatives:

```
> Label[4] := `upwind`;
  stencil[4] := subs(forward_time,right_space,pde);
```

$$Label_4 := upwind$$

$$stencil_4 := -\frac{u(t,x)}{s}+\frac{u(t+s,x)}{s}+c\left(-\frac{u(t,x-h)}{h}+\frac{u(t,x)}{h}\right) \quad \textbf{(1.6}$$

The Lax stencil involves making the replacement $u(t,x) \mapsto \frac{1}{2}(u(t,x+h)+u(t,x-h))$ in the FTCS stencil (we'll see why in the next section):

```
> Label[5] := `Lax`;
  stencil[5] := subs(u(t,x)=1/2*(u(t,x+h)+u(t,x-h)),stencil[1]);
```

$$Label_5 := Lax$$

$$stencil_5 := -\frac{\frac{1}{2}u(t,x+h)+\frac{1}{2}u(t,x-h)}{s}+\frac{u(t+s,x)}{s}+c\left(-\frac{1}{2}\frac{u(t,x-h)}{h}+\frac{1}{2}\frac{u(t,x+h)}{h}\right) \quad \textbf{(1.7}$$

Finally, the leapfrog stencil involves using centered approximations for both derivatives:

```
> Label[6] := `leapfrog`;
  stencil[6] := subs(centered_time,centered_space,pde);
```

$$Label_6 := leapfrog$$

$$stencil_6 := -\frac{1}{2}\frac{u(t-s,x)}{s}+\frac{1}{2}\frac{u(t+s,x)}{s}+c\left(-\frac{1}{2}\frac{u(t,x-h)}{h}+\frac{1}{2}\frac{u(t,x+h)}{h}\right) \quad \textbf{(1.8}$$

Some comments:

- The BTCS and Crank-Nicholson stencils are implicit, the others are all explicit.
- The leapfrog stencil involves 3 different timesteps, while the others all involve two timesteps. You can think of the leapfrog method as the PDE version of the midpoint method to solve $y'(x) = f(x, y(x))$.

von Neumann stability analysis

We now perform a von Neumann stability analysis of each of the stencils obtained in the previous section. We first re-write them in terms of the numeric solution $u_{i,j}$:

```
> Subs := [seq(seq(u(t+ii*s,x+jj*h)=u[i+ii,j+jj],ii=-1..1),jj=-1..1)]:

  for n from 1 to 6 do:
      _stencil[n] := subs(Subs,stencil[n]);
  od;
```

$$\_stencil_1 := -\frac{u_{i,j}}{s} + \frac{u_{i+1,j}}{s} + c\left(-\frac{1}{2}\frac{u_{i,j-1}}{h} + \frac{1}{2}\frac{u_{i,j+1}}{h}\right)$$

$$\_stencil_2 := -\frac{u_{i,j}}{s} + \frac{u_{i+1,j}}{s} + c\left(-\frac{1}{2}\frac{u_{i+1,j-1}}{h} + \frac{1}{2}\frac{u_{i+1,j+1}}{h}\right)$$

$$\_stencil_3 := -\frac{u_{i,j}}{s} + \frac{u_{i+1,j}}{s} + \frac{1}{2}c\left(-\frac{1}{2}\frac{u_{i,j-1}}{h} + \frac{1}{2}\frac{u_{i,j+1}}{h}\right) + \frac{1}{2}c\left(-\frac{1}{2}\frac{u_{i+1,j-1}}{h} + \frac{1}{2}\frac{u_{i+1,j+1}}{h}\right)$$

$$\_stencil_4 := -\frac{u_{i,j}}{s} + \frac{u_{i+1,j}}{s} + c\left(-\frac{u_{i,j-1}}{h} + \frac{u_{i,j}}{h}\right)$$

$$\_stencil_5 := -\frac{\frac{1}{2}u_{i,j+1} + \frac{1}{2}u_{i,j-1}}{s} + \frac{u_{i+1,j}}{s} + c\left(-\frac{1}{2}\frac{u_{i,j-1}}{h} + \frac{1}{2}\frac{u_{i,j+1}}{h}\right)$$

$$\_stencil_6 := -\frac{1}{2}\frac{u_{i-1,j}}{s} + \frac{1}{2}\frac{u_{i+1,j}}{s} + c\left(-\frac{1}{2}\frac{u_{i,j-1}}{h} + \frac{1}{2}\frac{u_{i,j+1}}{h}\right) \quad \textbf{(2.1}$$

The equation of motion for the errors are obtained by subtracting the elements of **stencil** and **_stencil** and defining $E_{i,j} = u_{i,j} - u(t_i, x_j)$. This just amounts to making the switch $u \mapsto E$ in the above.

```
> for n from 1 to 6 do:
      EOM[n] := subs(u=E,_stencil[n]);
  od;
```

$$EOM_1 := -\frac{E_{i,j}}{s} + \frac{E_{i+1,j}}{s} + c\left(-\frac{1}{2}\frac{E_{i,j-1}}{h} + \frac{1}{2}\frac{E_{i,j+1}}{h}\right)$$

$$EOM_2 := -\frac{E_{i,j}}{s} + \frac{E_{i+1,j}}{s} + c\left(-\frac{1}{2}\frac{E_{i+1,j-1}}{h} + \frac{1}{2}\frac{E_{i+1,j+1}}{h}\right)$$

$$EOM_3 := -\frac{E_{i,j}}{s} + \frac{E_{i+1,j}}{s} + \frac{1}{2}c\left(-\frac{1}{2}\frac{E_{i,j-1}}{h} + \frac{1}{2}\frac{E_{i,j+1}}{h}\right) + \frac{1}{2}c\left(-\frac{1}{2}\frac{E_{i+1,j-1}}{h} + \frac{1}{2}\frac{E_{i+1,j+1}}{h}\right)$$

$$EOM_4 := -\frac{E_{i,j}}{s} + \frac{E_{i+1,j}}{s} + c\left(-\frac{E_{i,j-1}}{h} + \frac{E_{i,j}}{h}\right)$$

$$EOM_5 := -\frac{\frac{1}{2}E_{i,j+1} + \frac{1}{2}E_{i,j-1}}{s} + \frac{E_{i+1,j}}{s} + c\left(-\frac{1}{2}\frac{E_{i,j-1}}{h} + \frac{1}{2}\frac{E_{i,j+1}}{h}\right)$$

$$EOM_6 := -\frac{1}{2}\frac{E_{i-1,j}}{s} + \frac{1}{2}\frac{E_{i+1,j}}{s} + c\left(-\frac{1}{2}\frac{E_{i,j-1}}{h} + \frac{1}{2}\frac{E_{i,j+1}}{h}\right) \quad (2.2$$

The von Neumann *ansatz* for the errors is $E_{i,j} = \xi^i \exp ikx_j$. We sub this into the equations of motion:

```
> ansatz := E[i,j] = xi^i*exp(I*k*x[j]);
  Subs := subs(x[j+1]=x[j]+h,x[j-1]=x[j]-h,[seq(seq(eval(ansatz,[i=i+ii,j=j+jj]),ii=-1..1),
  jj=-1..1)]):

  for n from 1 to 6 do:
     EOM[n] := subs(Subs,EOM[n]);
  od;
```

$$ansatz := E_{i,j} = \xi^i e^{I k x_j}$$

$$EOM_1 := -\frac{\xi^i e^{I k x_j}}{s} + \frac{\xi^{i+1} e^{I k x_j}}{s} + c\left(-\frac{1}{2}\frac{\xi^i e^{I k (x_j - h)}}{h} + \frac{1}{2}\frac{\xi^i e^{I k (x_j + h)}}{h}\right)$$

$$EOM_2 := -\frac{\xi^i e^{I k x_j}}{s} + \frac{\xi^{i+1} e^{I k x_j}}{s} + c\left(-\frac{1}{2}\frac{\xi^{i+1} e^{I k (x_j - h)}}{h} + \frac{1}{2}\frac{\xi^{i+1} e^{I k (x_j + h)}}{h}\right)$$

$$EOM_3 := -\frac{\xi^i e^{I k x_j}}{s} + \frac{\xi^{i+1} e^{I k x_j}}{s} + \frac{1}{2}c\left(-\frac{1}{2}\frac{\xi^i e^{I k (x_j - h)}}{h} + \frac{1}{2}\frac{\xi^i e^{I k (x_j + h)}}{h}\right) + \frac{1}{2}c\left(-\frac{1}{2}\frac{\xi^{i+1} e^{I k (x_j - h)}}{h} + \frac{1}{2}\frac{\xi^{i+1} e^{I k (x_j + h)}}{h}\right)$$

$$EOM_4 := -\frac{\xi^i e^{I k x_j}}{s} + \frac{\xi^{i+1} e^{I k x_j}}{s} + c\left(-\frac{\xi^i e^{I k (x_j - h)}}{h} + \frac{\xi^i e^{I k x_j}}{h}\right)$$

$$EOM_5 := -\frac{\frac{1}{2}\xi^i e^{I k (x_j + h)} + \frac{1}{2}\xi^i e^{I k (x_j - h)}}{s} + \frac{\xi^{i+1} e^{I k x_j}}{s} + c\left(-\frac{1}{2}\frac{\xi^i e^{I k (x_j - h)}}{h} + \frac{1}{2}\frac{\xi^i e^{I k (x_j + h)}}{h}\right)$$

$$EOM_6 := -\frac{1}{2}\frac{\xi^{i-1} e^{I k x_j}}{s} + \frac{1}{2}\frac{\xi^{i+1} e^{I k x_j}}{s} + c\left(-\frac{1}{2}\frac{\xi^i e^{I k (x_j - h)}}{h} + \frac{1}{2}\frac{\xi^i e^{I k (x_j + h)}}{h}\right) \quad \textbf{(2.3}$$

Each of the above can be solved for ξ:

```
> for n from 1 to 6 do:
     xi_sol[n] := {solve(simplify(expand(EOM[n])/exp(I*k*x[j]),exp),xi)} minus {0};
  od;
```

$$xi\_sol_1 := \left\{-\frac{I c \sin(k h) s - h}{h}\right\}$$

$$xi\_sol_2 := \left\{\frac{h}{I c \sin(k h) s + h}\right\}$$

$$xi\_sol_3 := \left\{-\frac{I c \sin(k h) s - 2 h}{I c \sin(k h) s + 2 h}\right\}$$

$$xi\_sol_4 := \left\{\frac{h + c e^{-I k h} s - c s}{h}\right\}$$

$$xi\_sol_5 := \left\{\frac{1}{2}\frac{e^{I k h} h + e^{-I k h} h + c e^{-I k h} s - c e^{I k h} s}{h}\right\}$$

$$xi\_sol_6 := \left\{-\frac{I\left(c \sin(k h) s - \sqrt{c^2 \sin(k h)^2 s^2 - h^2}\right)}{h}, -\frac{I\left(c \sin(k h) s + \sqrt{c^2 \sin(k h)^2 s^2 - h^2}\right)}{h}\right\} \quad \textbf{(2.4}$$

Notice that the first five stencils yield one solution for the amplification factor ξ, but the leapfrog stencil yields two solutions. This is directly related to the fact that this stencil involves three timesteps. The stability condition is that $|\xi| \leq 1$ (for the leapfrog stencil, we need both solutions to satisfy $|\xi| \leq 1$). Here are the absolute values:

```
> for n from 1 to 6 do:
     abs_xi_sol[n] := map(x->abs(x),xi_sol[n]) assuming(k,real,h>0,c>0,s>0);
  od;
```

$$abs\_xi\_sol_1 := \left\{\frac{\sqrt{h^2 + c^2 \sin(k h)^2 s^2}}{h}\right\}$$

$$abs\_xi\_sol_2 := \left\{ \frac{h}{\sqrt{h^2 + c^2 \sin(k\,h)^2 s^2}} \right\}$$

$$abs\_xi\_sol_3 := \{1\}$$

$$abs\_xi\_sol_4 := \left\{ \frac{\sqrt{(h + c\,s\cos(k\,h) - c\,s)^2 + c^2 \sin(k\,h)^2 s^2}}{h} \right\}$$

$$abs\_xi\_sol_5 := \left\{ \frac{\sqrt{h^2 \cos(k\,h)^2 + c^2 \sin(k\,h)^2 s^2}}{h} \right\}$$

$$abs\_xi\_sol_6 := \left\{ \frac{\left| c\sin(k\,h)\,s - \sqrt{c^2 \sin(k\,h)^2 s^2 - h^2} \right|}{h}, \frac{\left| c\sin(k\,h)\,s + \sqrt{c^2 \sin(k\,h)^2 s^2 - h^2} \right|}{h} \right\} \quad (2.5$$

These expression are more simply written in terms of the parameters $\theta = kh$ and $\alpha = \frac{cs}{h}$:

```
> theta_def := theta = k*h;
  alpha_def := alpha = c*s/h;

  for n from 1 to 6 do:
     abs_xi_sol[n] := map(x->subs(isolate(theta_def,k),isolate(alpha_def,c),x),abs_xi_sol[n]
  );
     abs_xi_sol[n] := map(x->simplify(x),abs_xi_sol[n]) assuming(h>0,theta,real);
     print(Label[n],'abs(xi)'=abs_xi_sol[n]);
  od:
```

$$theta\_def := \theta = k\,h$$

$$alpha\_def := \alpha = \frac{c\,s}{h}$$

$$FTCS, |\xi| = \left\{ \sqrt{1 + \alpha^2 - \alpha^2 \cos(\theta)^2} \right\}$$

$$BTCS, |\xi| = \left\{ \frac{1}{\sqrt{1 + \alpha^2 - \alpha^2 \cos(\theta)^2}} \right\}$$

$$Crank\text{-}Nicholson, |\xi| = \{1\}$$

$$upwind, |\xi| = \left\{ \sqrt{1 + 2\,\alpha\cos(\theta) - 2\,\alpha + 2\,\alpha^2 - 2\,\alpha^2\cos(\theta)} \right\}$$

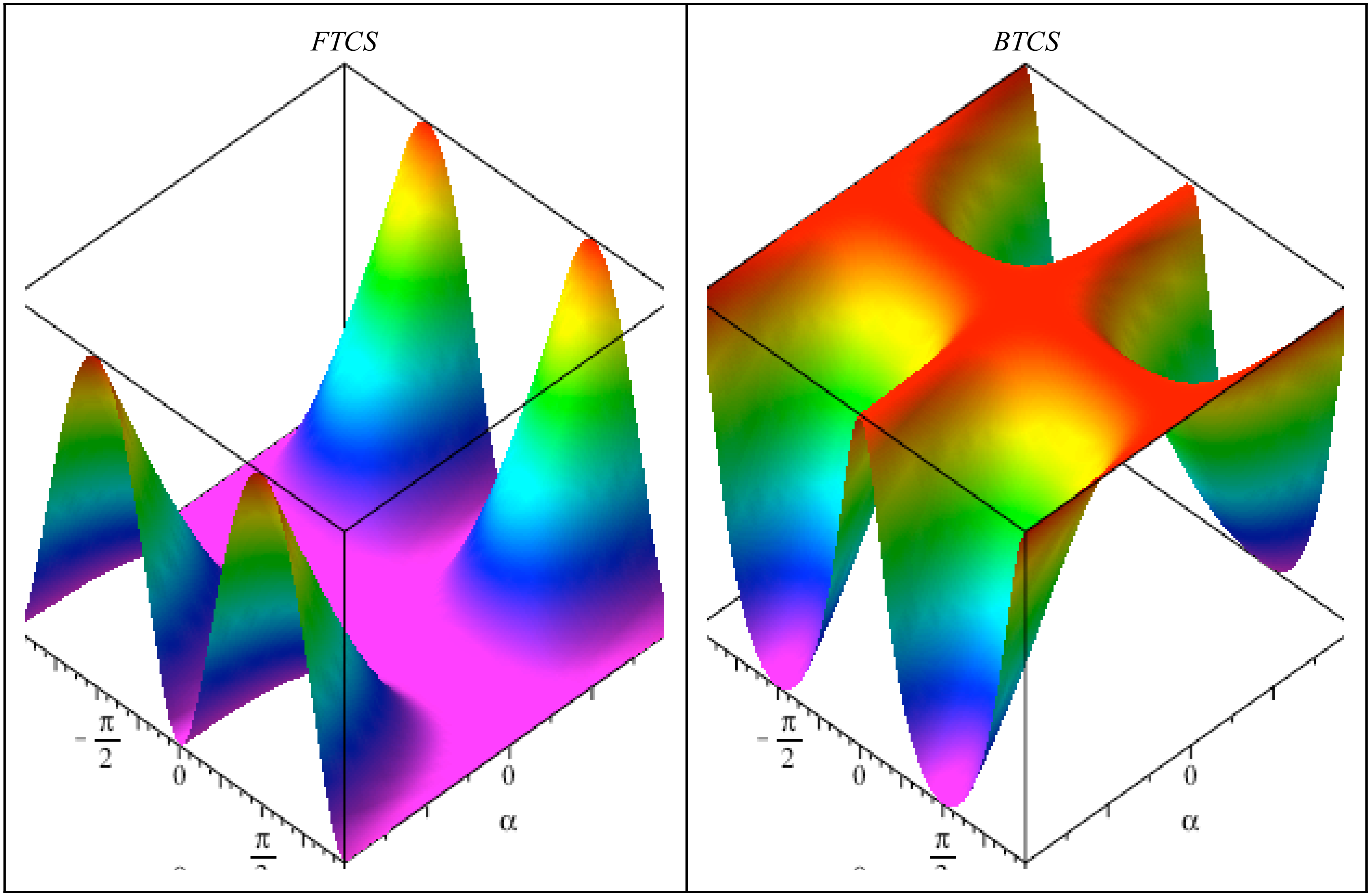
$$Lax, |\xi| = \left\{\sqrt{\cos(\theta)^2 + \alpha^2 - \alpha^2 \cos(\theta)^2}\right\}$$

$$leapfrog, |\xi| = \left\{\left|\alpha \sin(\theta) - \sqrt{-1 + \alpha^2 - \alpha^2 \cos(\theta)^2}\right|, \left|\alpha \sin(\theta) + \sqrt{-1 + \alpha^2 - \alpha^2 \cos(\theta)^2}\right|\right\} \quad (2.6$$

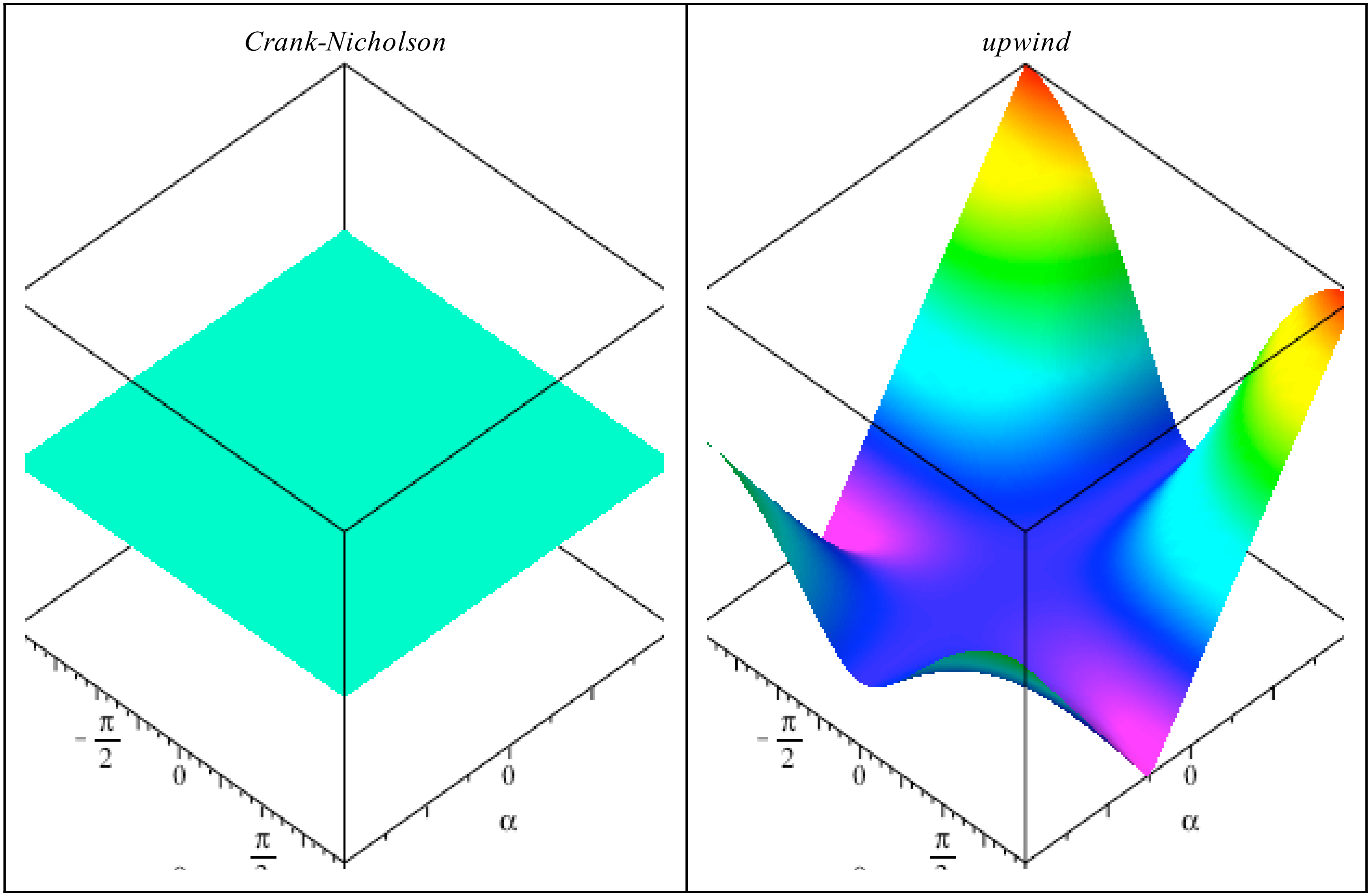
We can get a good idea of the stability properties of each stencil by plotting $|\xi|$ as a function of α and θ. (Note that each $|\xi|$ is periodic in θ.)

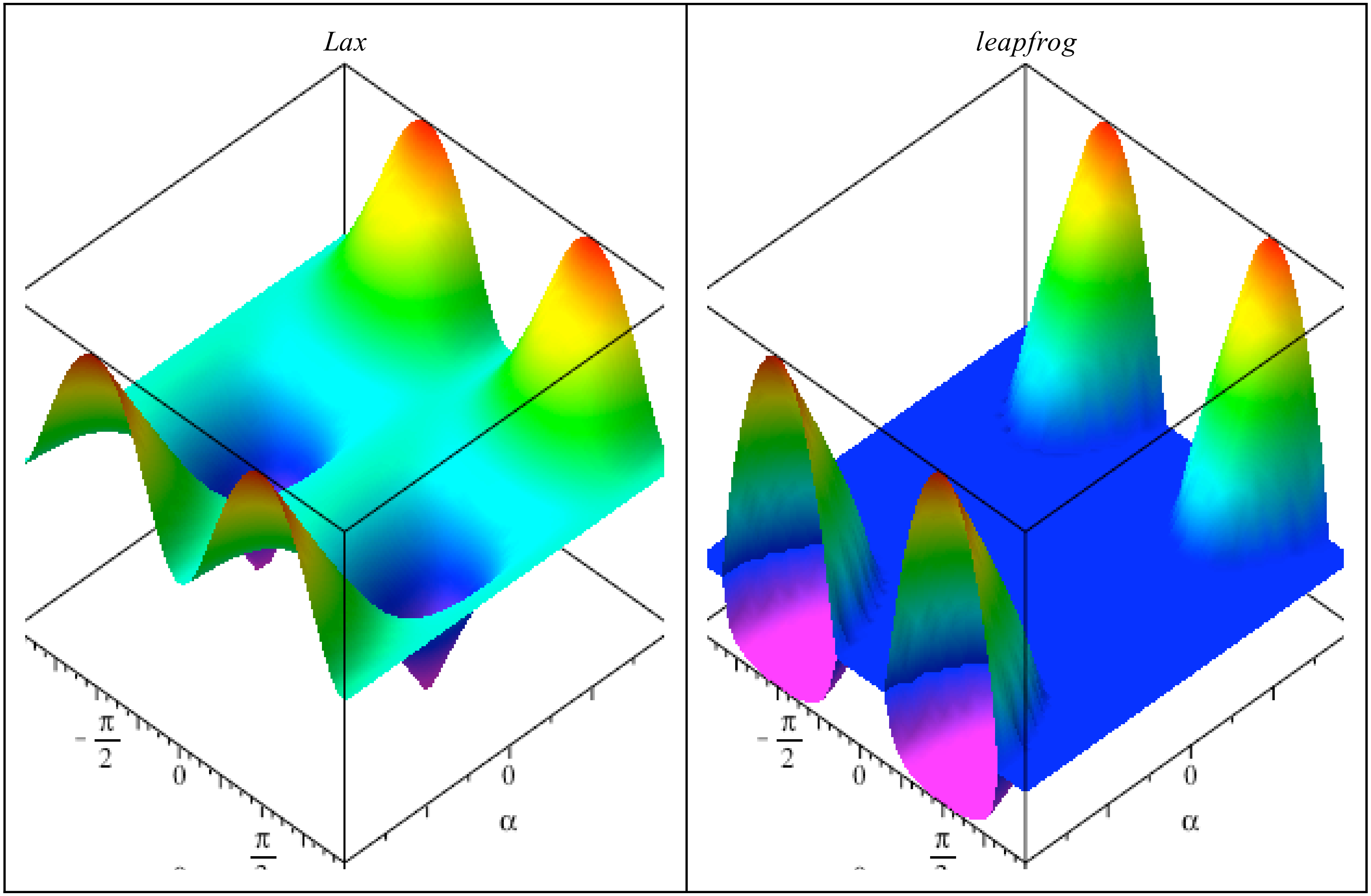
```
> p := Array(1..3,1..2):
  for i from 1 to 3 do:
     for j from 1 to 2 do:
        m := j+2*(i-1);
        p[i,j] := plot3d(abs_xi_sol[m],alpha=-2..2,theta=-Pi..Pi,axes=boxed,grid=[50,50],
  shading=zhue,style=patchnogrid,title=Label[m],lightmodel=light4,labels=[alpha,theta,abs(xi
  (alpha,theta))]);
     od:
  od;

  display(p);
```



FTCS
BTCS
$-\frac{\pi}{2}$
0
$\frac{\pi}{2}$
0
α
$-\frac{\pi}{2}$
0
$\frac{\pi}{2}$
0
α


Crank-Nicholson
upwind
$-\frac{\pi}{2}$
0
$\frac{\pi}{2}$
0
α
$-\frac{\pi}{2}$
0
$\frac{\pi}{2}$
0
α



From the plots we can infer:

- the FTCS stencil is unconditionally unstable
- the BTCS stencil is unconditionally stable
- the Crank-Nicholson stencil is unconditionally stable with $|\xi| = 1$

• the upwind method is stable for $\frac{cs}{h} \in [0, 1]$

• the Lax method is stable for $\frac{cs}{h} \in [-1, 1]$

• the leapfrog method is stable for $\frac{cs}{h} \in [-1, 1]$ with $|\xi| = 1$

Numeric solutions

We are now going to use the stencils from the above section to construct numeric solutions of the advection equation. We will seek a solution for $x \in [x_L, x_R]$. The boundaries of the interval will be stored in a Maple "range" variable **x_range = xL..xR**. We will assume a periodic solution with period $T = x_R - x_L$ such that $u(t, x+T) = u(t, x)$. The stencils defined above **(2.1)** all involve two discrete time levels (i and $i+1$), except for the leapfrog method which involves three time levels ($i-1$, i, and $i+1$). The former stencils are known as "one-step" methods, while the latter is a "two-step" method. Due to the fundemental difference in structure, we will construct different procedures for each type of stencil.

First, we deal with the one-step methods. Each of these stencils can be more compactly written if we denote the future field values by $\phi_j = u_{i+1,j}$ and the past field values as $\psi_j = u_{i,j}$:

```
> i := 'i':
  j := 'j':
  Subs := seq(u[i+1,j+jj]=phi[j+jj],jj=-1..1),seq(u[i,j+jj]=psi[j+jj],jj=-1..1);
  for n from 1 to 5 do:
     master_stencil[n] := unapply(expand(isolate(subs(Subs,_stencil[n]),phi)),psi,phi,j,c,s,
  h);
  od;
```

$$Subs := u_{i+1,j-1} = \phi_{j-1}, u_{i+1,j} = \phi_j, u_{i+1,j+1} = \phi_{j+1}, u_{i,j-1} = \psi_{j-1}, u_{i,j} = \psi_j, u_{i,j+1} = \psi_{j+1}$$

$$master\_stencil_1 := (\psi, \phi, j, c, s, h) \rightarrow \phi_j = \psi_j + \frac{1}{2}\frac{s\,c\,\psi_{j-1}}{h} - \frac{1}{2}\frac{s\,c\,\psi_{j+1}}{h}$$

$$master\_stencil_2 := (\psi, \phi, j, c, s, h) \rightarrow -2\,\phi_j\,h + c\,s\,\phi_{j-1} - c\,s\,\phi_{j+1} = -2\,\psi_j\,h$$

$$master\_stencil_3 := (\psi, \phi, j, c, s, h) \rightarrow -4\,\phi_j\,h + c\,s\,\phi_{j-1} - c\,s\,\phi_{j+1} = -4\,\psi_j\,h - s\,c\,\psi_{j-1} + s\,c\,\psi_{j+1}$$

$$master\_stencil_4 := (\psi, \phi, j, c, s, h) \rightarrow \phi_j = \psi_j + \frac{s\,c\,\psi_{j-1}}{h} - \frac{s\,c\,\psi_j}{h}$$

$$master\_stencil_5 := (\psi, \phi, j, c, s, h) \rightarrow \phi_j = \frac{1}{2}\,\psi_{j+1} + \frac{1}{2}\,\psi_{j-1} + \frac{1}{2}\,\frac{s\,c\,\psi_{j-1}}{h} - \frac{1}{2}\,\frac{s\,c\,\psi_{j+1}}{h} \qquad \textbf{(3.1}$$

Our spatial lattice will be defined by

$$x_j = x_{\mathrm{L}} + \frac{(j-1)}{M}\left(x_{\mathrm{R}} - x_{\mathrm{L}}\right), \quad h = x_{j+1} - x_j = \frac{x_{\mathrm{R}} - x_{\mathrm{L}}}{M}$$

This implies $x_1 = x_{\mathrm{L}}$ and $x_{M+1} = x_{\mathrm{R}}$. Furthermore, the assumed periodicity of the solution $u(t, x+T) = u(t,x)$ implies $u_{i,j} = u_{i,j+M}$. O more concretely:

$$\phi_0 = \phi_M, \quad \phi_{M+1} = \phi_1, \quad \psi_0 = \psi_M, \quad \psi_{M+1} = \psi_1.$$

Using these boundary conditions, each of the one-step stencils can be written in the form

$$\mathbf{P}_L\phi = \mathbf{P}_R\psi, \quad \phi = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_M \end{bmatrix}, \quad \psi = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_M \end{bmatrix}.$$

For an explicit stencil, we expect $\mathbf{P}_L = \mathbf{I}$. This procedure calculates the form of the square matrices $\mathbf{P}_L$ and $\mathbf{P}_R$ for each of the one-step stencils:

```
> _P := proc(M,c,s,h,choice)
       local phi, psi, sys, P:

       phi[0]  := phi[M]:
       phi[M+1]  := phi[1]:
       psi[0]  := psi[M]:
       psi[M+1]  := psi[1]:

       sys := [seq(master_stencil[choice](psi,phi,j,c,s,h),j=1..M)];
       P[L]  := GenerateMatrix(sys,[seq(phi[j],j=1..M)])[1];
       P[R]  := -GenerateMatrix(sys,[seq(psi[j],j=1..M)])[1];

       P[L],P[R];
```

```
end proc:
```

The output is `P[L],P[R]`. Here is an example for the Lax stencil:

```
> _P(5,c,s,h,5);
```

$$\left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right], \left[\begin{array}{ccccc} 0 & -\frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} & 0 & 0 & \frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} \\ \frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} & 0 & -\frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} & 0 & -\frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} & 0 & -\frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} \\ -\frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} & 0 & 0 & \frac{1}{2}\frac{c\,s}{h}+\frac{1}{2} & 0 \end{array}\right] \quad (3.2$$

As expected, the lefthand matrix is the identity since we are considering an explicit stencil. Notice the entries in the top-right and bottom-lef corners, these are the result of the periodic boundary conditions. Here is the form for the Crank-Nicholson stencil:

```
> _P(5,c,s,h,3);
```

$$\left[\begin{array}{ccccc} -4\,h & -c\,s & 0 & 0 & c\,s \\ c\,s & -4\,h & -c\,s & 0 & 0 \\ 0 & c\,s & -4\,h & -c\,s & 0 \\ 0 & 0 & c\,s & -4\,h & -c\,s \\ -c\,s & 0 & 0 & c\,s & -4\,h \end{array}\right], \left[\begin{array}{ccccc} -4\,h & c\,s & 0 & 0 & -c\,s \\ -c\,s & -4\,h & c\,s & 0 & 0 \\ 0 & -c\,s & -4\,h & c\,s & 0 \\ 0 & 0 & -c\,s & -4\,h & c\,s \\ c\,s & 0 & 0 & -c\,s & -4\,h \end{array}\right] \quad (3.3$$

This code utilizes these matrices to solve the advection equation for a given choice of one-step stencil:

```
> one_step := proc(f,x_range,tau,N,M,c,choice)
    local x, t, h, s, X, u_past, p, P, i, u_future:

    x := j -> evalf(lhs(x_range) + j/(M+1)*(rhs(x_range)-lhs(x_range)));
    t := i -> evalf(tau/N*i):
    h := x(1)-x(0);
    s := t(1)-t(0);

    X := Vector([seq(x(j),j=1..M)]):
```

```
    u_past := map(z->f(z),X):

    p[0] := Frame(X,u_past,s,h,c,t(0),choice):

    P[1],P[2] := _P(M,c,s,h,choice):

    for i from 1 to N do:
       u_future := LinearSolve(P[1],P[2].u_past):
       u_past := LinearAlgebra[Copy](u_future):
       p[i] := Frame[1](X,u_past,s,h,c,t(i),choice):
    od:

    display(convert(p,list),insequence=true);

 end proc:

 Color := [red,green,blue,magenta,violet,plum]:

 Frame := proc(xdata,ydata,s,h,c,T,choice) local Title, PlotOptions:

        Title := typeset(`timestep = `,evalf[2](s),`, spacestep = `,evalf[2](h),`, `,
 alpha=evalf[2](c*s/h),`, `,t=evalf[2](T));
        PlotOptions := axes=boxed, labels=[x,u(t,x)], legend=[Label[choice]], color=Color
 [choice];
        plot(Matrix([xdata,ydata]),title=Title,PlotOptions);

 end proc:
```

For the leapfrog stencil, we define $\phi_j = u_{i+1,j}$ and $\psi_j = u_{i,j}$ as before, but we also write $\chi_j = u_{i-1,j}$:

```
> i := 'i':
  j := 'j':
  Subs := seq(u[i+1,j+jj]=phi[j+jj],jj=-1..1),seq(u[i,j+jj]=psi[j+jj],jj=-1..1),seq(u[i-1,j+
  jj]=chi[j+jj],jj=-1..1);
  master_stencil[6] := unapply(expand(isolate(subs(Subs,_stencil[n]),phi)),chi,psi,phi,j,c,
  s,h);
```

$$Subs := u_{i+1,j-1} = \phi_{j-1}, u_{i+1,j} = \phi_j, u_{i+1,j+1} = \phi_{j+1}, u_{i,j-1} = \psi_{j-1}, u_{i,j} = \psi_j, u_{i,j+1} = \psi_{j+1}, u_{i-1,j-1} = \chi_{j-1}, u_{i-1,j} = \chi_j, u_{i-1,j+1} = \chi_{j+1} \quad (3.4$$

$$master\_stencil_6 := (\chi, \psi, \phi, j, c, s, h) \rightarrow \phi_j = \chi_j + \frac{s\, c\, \psi_{j-1}}{h} - \frac{s\, c\, \psi_{j+1}}{h} \tag{3.4}$$

This stencil is of the form

$$\phi = \chi + \mathbf{Q}\psi.$$

Here is a procedure that yields the **Q** matrix:

```
> _Q := proc(M,c,s,h)
      local phi, psi, chi, sys, P:

      phi[0] := phi[M]:
      phi[M+1] := phi[1]:
      psi[0] := psi[M]:
      psi[M+1] := psi[1]:
      chi[0] := chi[M]:
      chi[M+1] := chi[1]:
      sys := [seq(master_stencil[6](chi,psi,phi,j,c,s,h),j=1..M)];
      -GenerateMatrix(sys,[seq(psi[j],j=1..M)])[1];

  end proc:

  _Q(5,c,s,h);
```

$$\begin{bmatrix} 0 & -\frac{c\,s}{h} & 0 & 0 & \frac{c\,s}{h} \\ \frac{c\,s}{h} & 0 & -\frac{c\,s}{h} & 0 & 0 \\ 0 & \frac{c\,s}{h} & 0 & -\frac{c\,s}{h} & 0 \\ 0 & 0 & \frac{c\,s}{h} & 0 & -\frac{c\,s}{h} \\ -\frac{c\,s}{h} & 0 & 0 & \frac{c\,s}{h} & 0 \end{bmatrix} \tag{3.5}$$

This procedure calculates the actual leapfrog solution. Notice how since we are using a two-step method, we need to retain the values of

$u(t, x)$ at two past timesteps at all times. Also note how we use the Lax stencil in the first step to generate $u_{1,j}$, which is required (along with $u_{0,j}$) by the leapfrog stencil to generate $u_{2,j}$ in the second step.

```
> leapfrog := proc(f,x_range,tau,N,M,c)
     local x, t, h, s, X, u_past, p, P, i, u_future, u_farpast, Q:

     x := j -> evalf(lhs(x_range) + j/(M+1)*(rhs(x_range)-lhs(x_range)));
     t := i -> evalf(tau/N*i):
     h := x(1)-x(0);
     s := t(1)-t(0);

     X := Vector([seq(x(j),j=1..M)]):
     u_past := map(z->f(z),X):

     p[0] := Frame(X,u_past,s,h,c,t(0),6):

     P[1],P[2] := _P(M,c,s,h,5):
     u_future := LinearSolve(P[1],P[2].u_past):
     u_farpast := LinearAlgebra[Copy](u_past):
     u_past := LinearAlgebra[Copy](u_future):
     p[1] := Frame(X,u_past,s,h,c,t(1),6):

     Q := _Q(M,c,s,h):
     for i from 2 to N do:
        u_future := Q.u_past + u_farpast:
        u_farpast := LinearAlgebra[Copy](u_past):
        u_past := LinearAlgebra[Copy](u_future):
        p[i] := Frame(X,u_past,s,h,c,t(i),6):
     od:
     display(convert(p,list),insequence=true);

  end proc:
```

Now let's turn our attention to actual simulation results. For concreteness, let's fix some of the parameters and initial data a follows:

```
> m := 'm':
  f := x -> exp(-x^2);
  L := 10;
  x_range := -L .. L;
  tau := 20;
  c := 1;
```

$$f := x \to e^{-x^2}$$

$$L := 10$$

$$x\_range := -10..10$$

$$\tau := 20$$

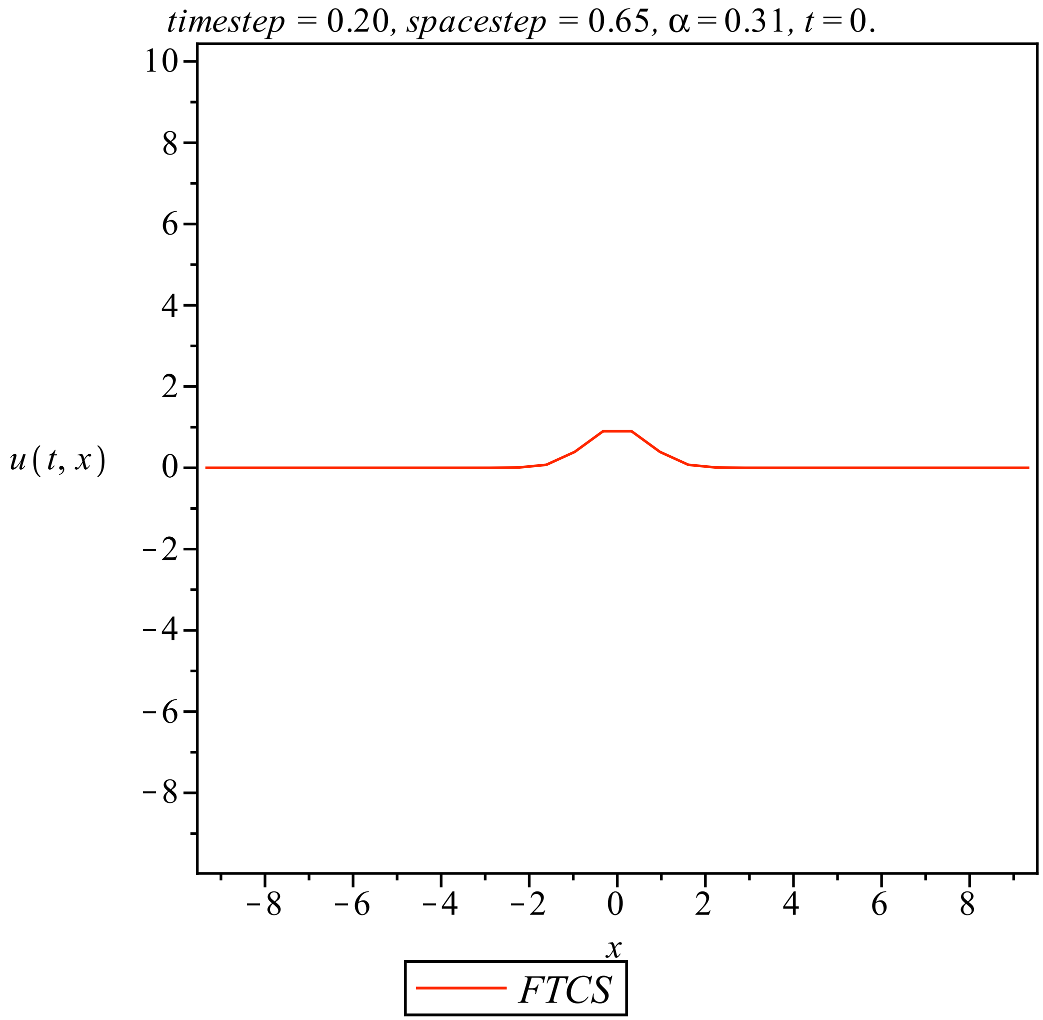
$$c := 1 \qquad (3.6$$

We first confirm that the FTCS stencil is unstable no matter what parameters are used:

```
> N := 100;
  M := 30;
  one_step(f,x_range,tau,N,M,c,1);
```

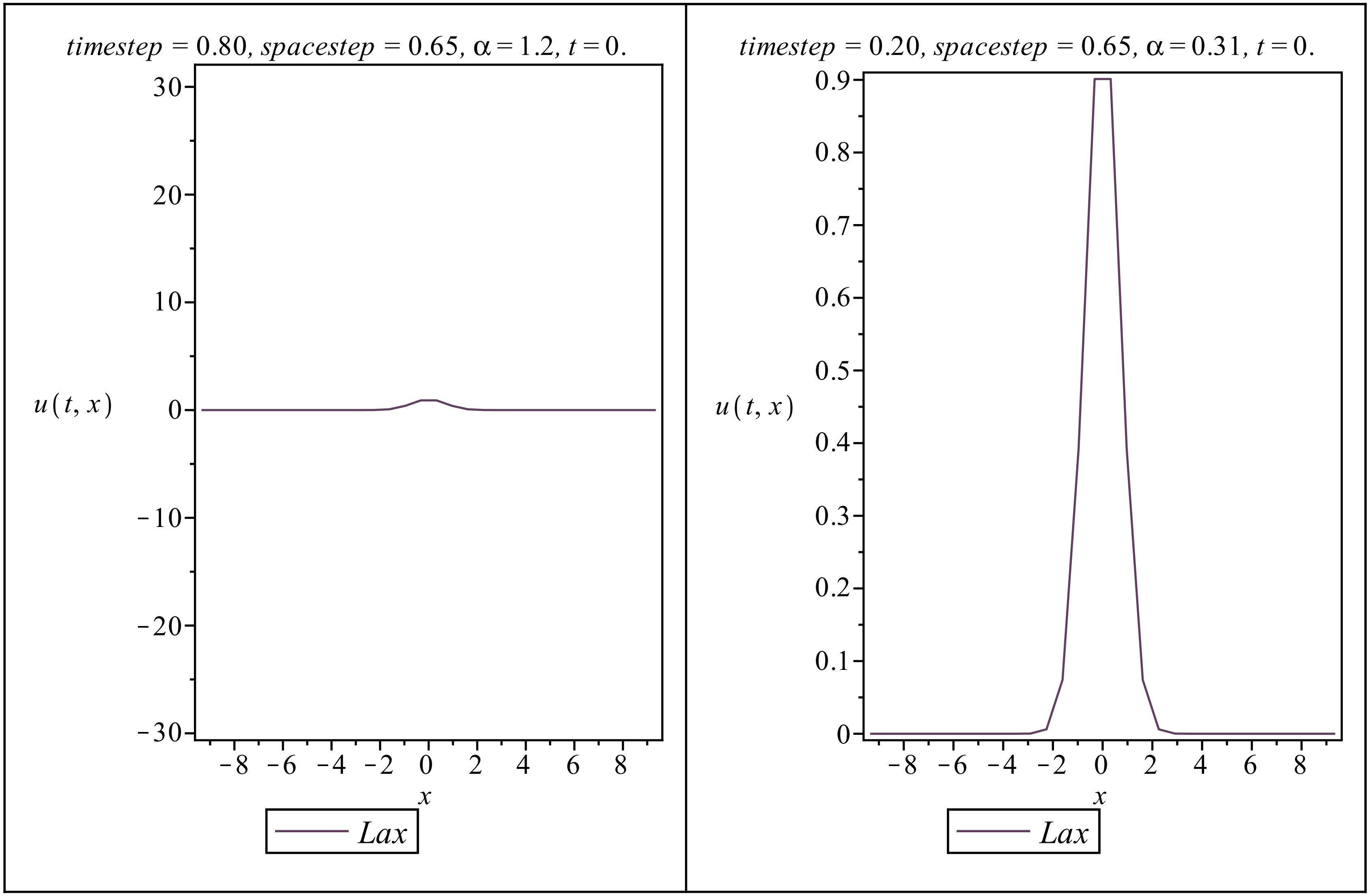
$$N := 100$$

$$M := 30$$



Here is an example of the conditional stability of the Lax method. The right panel has $|\alpha| = |c| s h^{-1} < 1$, while the lefthand panel has $|\alpha| > 1$:

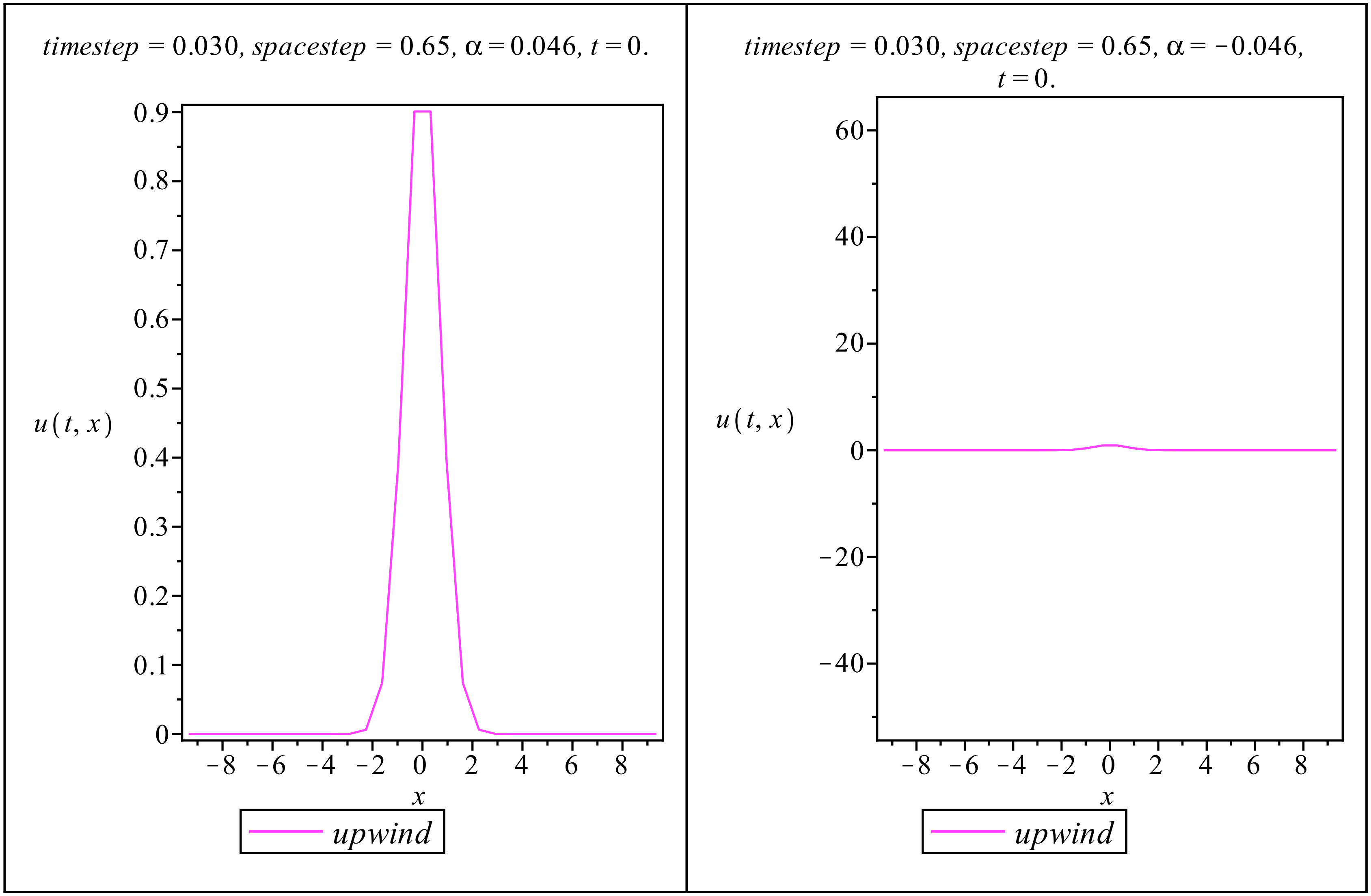
```
> m := 'm':
  m[1] := one_step(f,x_range,tau,N/4,M,c,5):
  m[2] := one_step(f,x_range,tau,N,M,c,5):
  display(Array([m[1],m[2]]));
```



Here is an example of how the upwind method is only stable for a rightgoing pulse. The lefthand panel has positive velocity $c > 0$ while the righthand panel has negative velocity $c < 0$.

```
> m := 'm':
  N := 100:
  tau := 3:
  m[1] := one_step(f,x_range,tau,N,M,c,4):
```

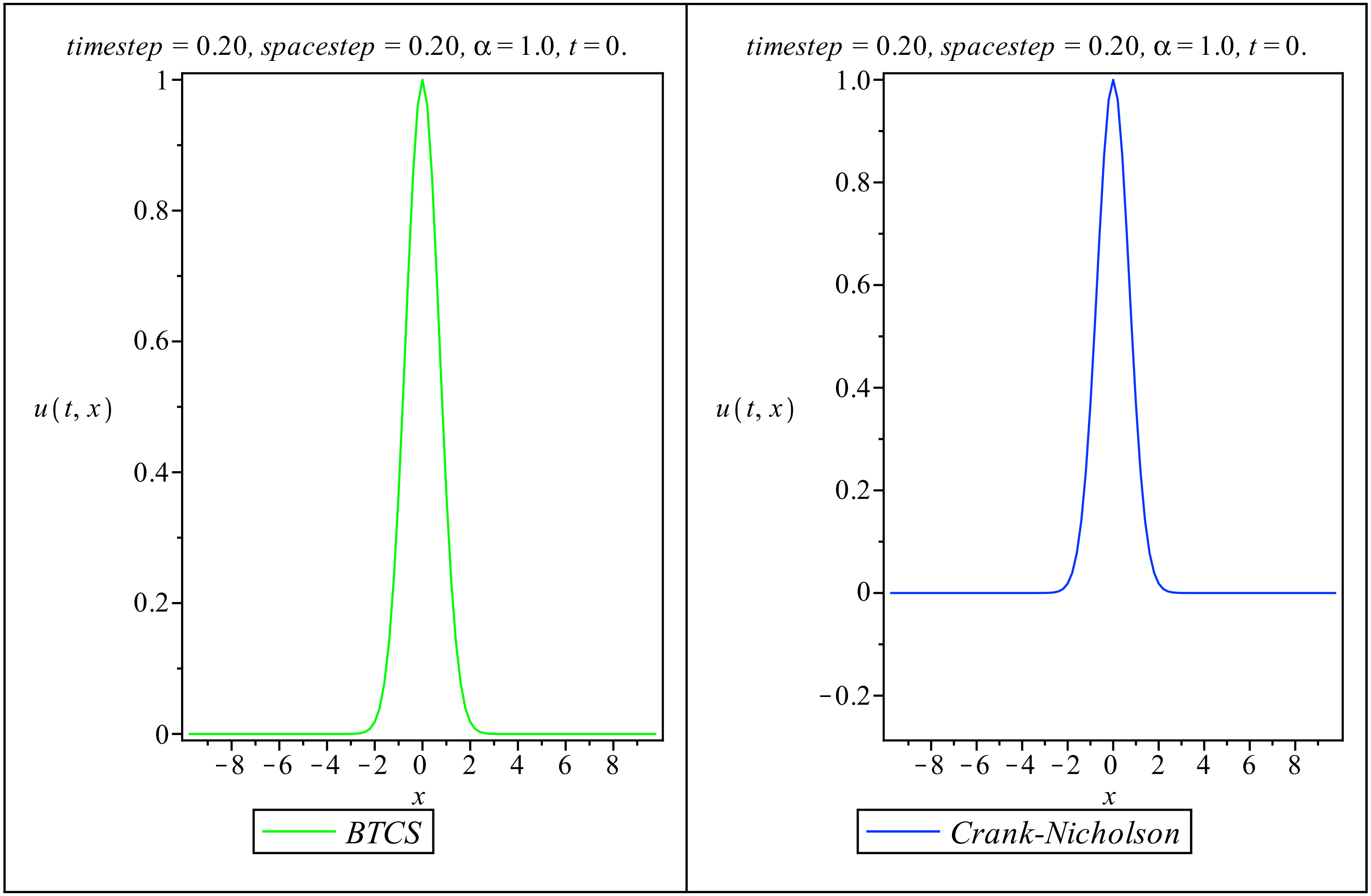
```
m[2] := one_step(f,x_range,tau,N,M,-c,4):
display(Array([m[1],m[2]]));
```

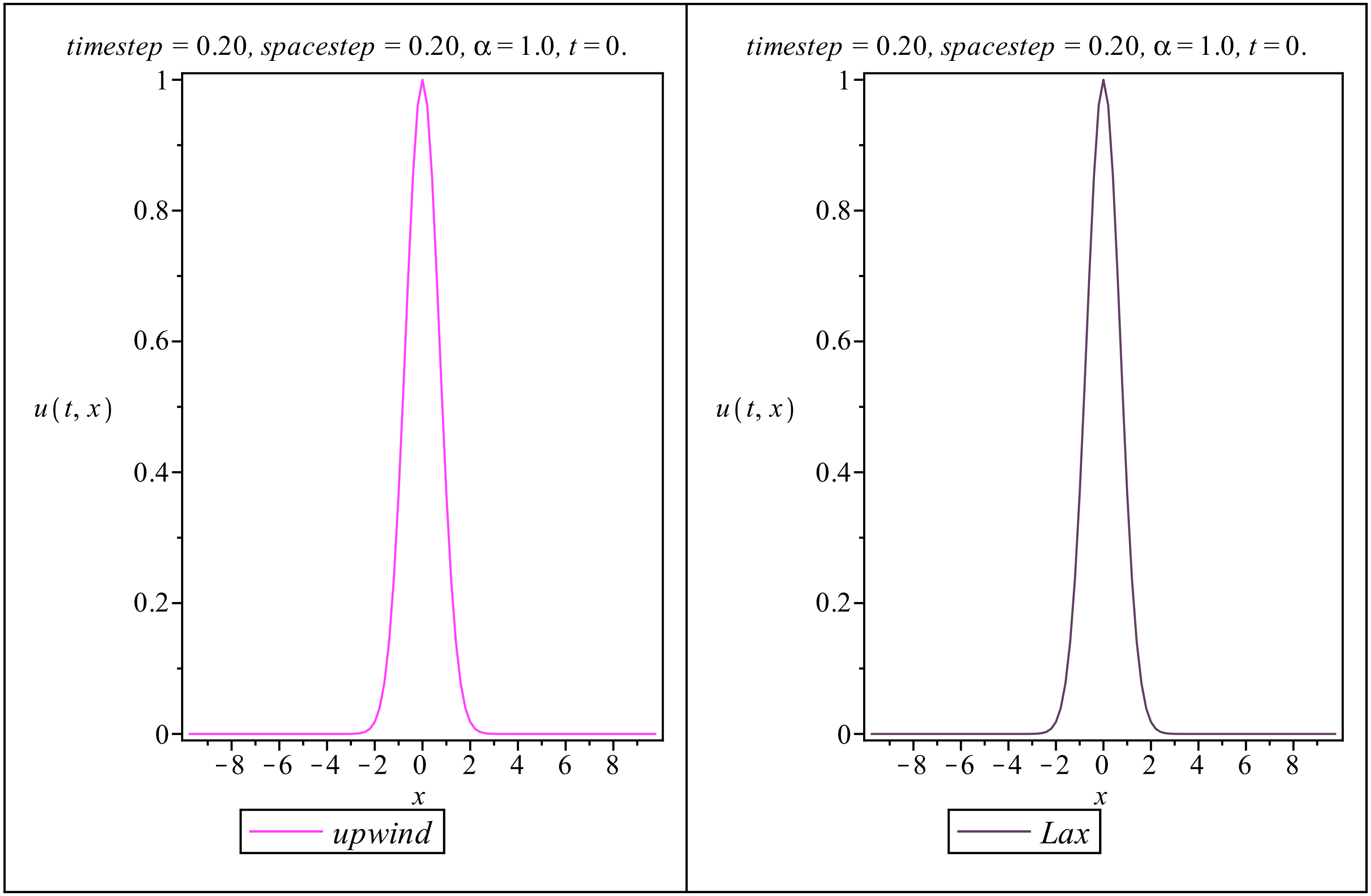


Here is an example of the behaviour of all the stable stencils when $\alpha = 1$. Notice how the explicit stencils (upwind, Lax, and leapfrog) all perform well here:

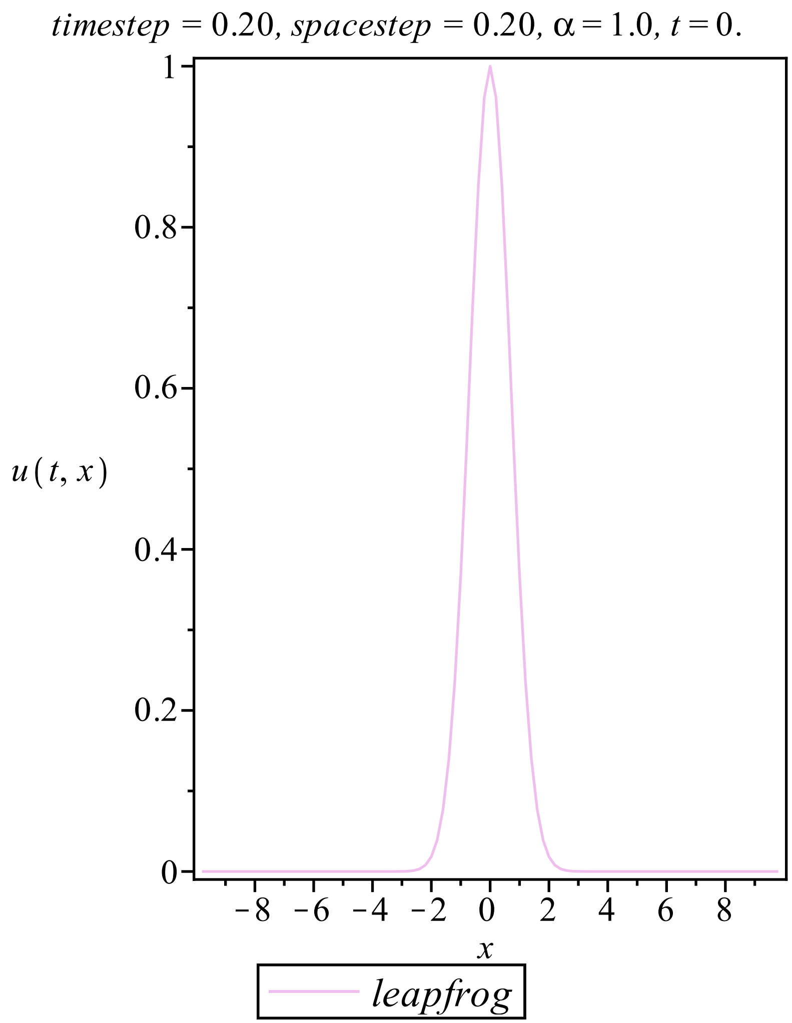
```
> m := 'm':
```

```
tau := 20:
N := 100:
M := 99:
for n from 1 to 4 do:
   m[n] := one_step(f,x_range,tau,N,M,c,n+1);
od:
m[5] := leapfrog(f,x_range,tau,N,M,c,n+1):
display(Array([ [m[1],m[2]],[m[3],m[4]],[m[5],plot(x->NULL,axes=none)]]));
```


timestep = 0.20, spacestep = 0.20, α = 1.0, t = 0.
u(t, x)
x
BTCS
timestep = 0.20, spacestep = 0.20, α = 1.0, t = 0.
u(t, x)
x
Crank-Nicholson



timestep = 0.20, spacestep = 0.20, α = 1.0, t = 0.
1
0.8
0.6
u(t, x)
0.4
0.2
0
−8 −6 −4 −2 0 2 4 6 8
x
upwind
timestep = 0.20, spacestep = 0.20, α = 1.0, t = 0.
1
0.8
0.6
u(t, x)
0.4
0.2
0
−8 −6 −4 −2 0 2 4 6 8
x
Lax



timestep = 0.20, spacestep = 0.20, α = 1.0, t = 0.
u(t, x)
1
0.8
0.6
0.4
0.2
0
−8 −6 −4 −2 0 2 4 6 8
x
leapfrog