Lineární algebra

```
>
```

Při importu knihovny **linalg** hned zjistíme, co vše "umí"

```
> with(linalg);
Warning, new definition for norm
Warning, new definition for trace
```

[*BlockDiagonal*, *GramSchmidt*, *JordanBlock*, *LUdecomp*, *QRdecomp*, *Wronskian*, *addcol*, *addrow*, *adj*, *adjoint*, *angle*, *augment*, *backsub*, *band*, *basis*, *bezout*, *blockmatrix*, *charmat*, *charpoly*, *cholesky*, *col*, *coldim*, *colspace*, *colspan*, *companion*, *concat*, *cond*, *copyinto*, *crossprod*, *curl*, *definite*, *delcols*, *delrows*, *det*, *diag*, *diverge*, *dotprod*, *eigenvals*, *eigenvalues*, *eigenvectors*, *eigenvects*, *entermatrix*, *equal*, *exponential*, *extend*, *ffgausselim*, *fibonacci*, *forwardsub*, *frobenius*, *gausselim*, *gaussjord*, *geneqns*, *genmatrix*, *grad*, *hadamard*, *hermite*, *hessian*, *hilbert*, *htranspose*, *ihermite*, *indexfunc*, *innerprod*, *intbasis*, *inverse*, *ismith*, *issimilar*, *iszero*, *jacobian*, *jordan*, *kernel*, *laplacian*, *leastsqrs*, *linsolve*, *matadd*, *matrix*, *minor*, *minpoly*, *mulcol*, *mulrow*, *multiply*, *norm*, *normalize*, *nullspace*, *orthog*, *permanent*, *pivot*, *potential*, *randmatrix*, *randvector*, *rank*, *ratform*, *row*, *rowdim*, *rowspace*, *rowspan*, *rref*, *scalarmul*, *singularvals*, *smith*, *stackmatrix*, *submatrix*, *subvector*, *sumbasis*, *swapcol*, *swaprow*, *sylvester*, *toeplitz*, *trace*, *transpose*, *vandermonde*, *vecpotent*, *vectdim*, *vector*, *wronskian*]

Vektorová analýza

```
> curl([0,0,1/r*sin(theta)], [r,theta,phi], coords=spherical);
```

$$\left[2\frac{\cos(\theta)}{r^2}, 0, 0\right]$$

Totéž dokážeme explicitním uvedením Lameových koeficientů

```
> curl([0,0,1/r*sin(theta)], [r,theta,phi], [1,r,r*sin(theta)]);
```

$$\left[2\frac{\cos(\theta)}{r^2}, 0, 0\right]$$

```
> diverge(%,[r,theta,phi], coords=spherical);
```

$$0$$

```
> orthopoly[P](3,x);
```

$$\frac{5}{2}x^3-\frac{3}{2}x$$

```
> laplacian(orthopoly[P](3,cos(theta))/r^4,[r,theta,phi],
  coords=spherical):
  simplify(%);
```

$$0$$

Neumí ale laplacian na vektor, ve třech dimezích si ale můžeme pomoci takto:

```
> vec_laplacian:=proc(V,C,O)
    local O2;

    if nargs<3 then O2:=coords='rectangular'; else O2 := O; fi;
    RETURN(evalm(grad(
  diverge(V,C,O2),C,O2)-curl(curl(V,C,O2),C,O2)));
  end:
```

```
> grad(cos(theta)/r^3,[r,theta,phi],coords=spherical);
```

$$\left[-3\,\frac{\cos(\theta)}{r^4}, -\frac{\sin(\theta)}{r^4}, 0\right]$$

```
> vec_laplacian(%,[r,theta,phi],coords=spherical);
```

$$\left[-20\,\frac{\cos(\theta)}{r^6}, -4\,\frac{\sin(\theta)}{r^6}, 0\right]$$

```
>
```

Matice

Vytvořit můžeme matici několika způsoby

```
> A:=matrix([[0,1],[1,0]]);
```

$$A := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

```
> B:=matrix(2,2,[1,0,0,-1]);
```

$$B := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

```
> matrix(2,2,(i,j)->abs(i-j));
```

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

```
> stackmatrix(%,%%);
```

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & -1 \end{bmatrix}$$

```
> jacobian([y,x],[x,y]);
```

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

```
> band([1,-2,1],13);
```

$$\begin{bmatrix}
-2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2
\end{bmatrix}$$

```
> diag(1,0,-1);
  diag(1$5);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> diag( matrix([[0,1],[1,0]]), 1,-1) ;
```

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Operace s maticemi a vektory

Aby mohlo probíhat zjednodušování i na úrovni symbolů, musíme explicitně používat **evalm**

```
> A+B;
```

$$A+B$$

```
> evalm(A+B);
```

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

```
> evalm(A&*B)-evalm(B&*A);
  evalm(%);
```

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -2 \\ 2 & 0 \end{bmatrix}$$

```
> evalm(A&*B&*inverse(A));
  evalm(A&*B&*(A)^(-1));
```

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
> N:=2;
  matrix(N,N,(i,j)->1/(1+i*x+j*y));
  evalm(inverse(%));
```

$$N := 2$$

$$\begin{bmatrix} \frac{1}{1+x+y} & \frac{1}{1+x+2y} \\ \frac{1}{1+2x+y} & \frac{1}{1+2x+2y} \end{bmatrix}$$

$$\begin{bmatrix} \frac{(1+x+y)(1+x+2y)(1+2x+y)}{xy} & -\frac{(1+x+y)(1+2x+2y)(1+2x+y)}{xy} \\ -\frac{(1+x+y)(1+2x+2y)(1+x+2y)}{xy} & \frac{(1+2x+2y)(1+x+2y)(1+2x+y)}{xy} \end{bmatrix}$$

```
> det(%);
```

$$\frac{(1+x+y)(1+2x+2y)(1+x+2y)(1+2x+y)}{x\,y}$$

```
> iMx:=matrix(3,3,[0,0,0,0,0,1,0,-1,0]);
  iMy:=matrix(3,3,[0,0,-1,0,0,0,1,0,0]);
  iMz:=matrix(3,3,[0,1,0,-1,0,0,0,0,0]);
```

$$iMx := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$iMy := \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$iMz := \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

```
>
> exponential(-vx*iMx-vy*iMy-vz*iMz):
  subs(exp(-sqrt(-vx^2-vz^2-vy^2))=cos(v)-I*sin(v),
       exp(+sqrt(-vx^2-vz^2-vy^2))=cos(v)+I*sin(v),%):
  subs(1/sqrt(-vx^2-vz^2-vy^2)=1/(I*v),sqrt(-vx^2-vz^2-vy^2)=(I*v)
  ,(vx^2+vz^2+vy^2)=v^2,%):
  simplify(%);
```

$$\left[\frac{vx^2+vz^2\cos(v)+vy^2\cos(v)}{v^2},\right.$$

$$-\frac{vz\,vx^2\sin(v)+vz^3\sin(v)+vz\,vy^2\sin(v)+vy\,vx\,v\cos(v)-vy\,vx\,v}{v^3},$$

$$\left.-\frac{vz\,vx\,v\cos(v)-vz\,vx\,v-vy\,vx^2\sin(v)-vy\,vz^2\sin(v)-vy^3\sin(v)}{v^3}\right]$$

$$\left[-\frac{-vz\,vx^2\sin(v)-vz^3\sin(v)-vz\,vy^2\sin(v)+vy\,vx\,v\cos(v)-vy\,vx\,v}{v^3},\right.$$

$$\frac{vy^2+vz^2\cos(v)+vx^2\cos(v)}{v^2},$$

$$\left.-\frac{vz\,vy\,v\cos(v)-vz\,vy\,v+vx^3\sin(v)+vx\,vz^2\sin(v)+vx\,vy^2\sin(v)}{v^3}\right]$$

$$\left[-\frac{vy\,vx^2\sin(v)+vy\,vz^2\sin(v)+vy^3\sin(v)+vz\,vx\,v\cos(v)-vz\,vx\,v}{v^3},\right.$$

$$-\frac{vz\,vy\,v\cos(v)-vz\,vy\,v-vx^3\sin(v)-vx\,vz^2\sin(v)-vx\,vy^2\sin(v)}{v^3},$$

$$\left.\frac{vz^2 + vy^2\cos(v) + vx^2\cos(v)}{v^2}\right]$$

Především lze snadno počítat řady s maticemi

```
> evalm(exponential(u*iMx+u*iMy)-exponential(u*iMx)&*exponential(u
*iMy)):
map(series,%,u,3);
map(convert,%,polynom)=evalm(1/2*u^2*iMz);
```

$$\begin{bmatrix} O(u^3) & \frac{1}{2}u^2 + O(u^3) & O(u^3) \\ -\frac{1}{2}u^2 + O(u^3) & O(u^3) & O(u^3) \\ O(u^3) & O(u^3) & O(u^3) \end{bmatrix}$$

$$\begin{bmatrix} 0 & \frac{1}{2}u^2 & 0 \\ -\frac{1}{2}u^2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2}u^2 & 0 \\ -\frac{1}{2}u^2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

```
>
```

Příklad na použití lineární algebry:

Časový vývoj stavu v 1D kvantovém oscilátoru

Metoda nijak nepředpkládá zrovna harmonický oscilátor, takže můžete experimentovat

```
> restart;
> Digits:=14;
```

$$Digits := 14$$

```
> with(linalg):with(plots):
Warning, new definition for norm
Warning, new definition for trace
```

Velmi naivní volba jak zajistit aby se v limitě pro N velké pokryla celá osa nekonečně malými intervaly

```
> N:=139;
dx:=1/sqrt(N);
```

$$N := 139$$

$$dx := \frac{1}{139}\sqrt{139}$$

Numerická druhá derivace

```
> T:=band([-1,2,-1]/dx^2, N):
```

Potenciál je diagonální

```
> x:=(i)->(i-(N+1)/2)*dx;
v:=(x)->x^2;
V:=evalf(diag(seq(v(x(i)),i=1..N))):
```

$$x := i \rightarrow \left(i - \frac{1}{2}N - \frac{1}{2}\right) dx$$

$$v := x \rightarrow x^2$$

```
> H:=evalm(T+V):
```

Kontrola, zda náhodou spektrum nevyšlo degenerované

```
>
> sort([eigenvalues(H)]):
  N=nops(%);
```

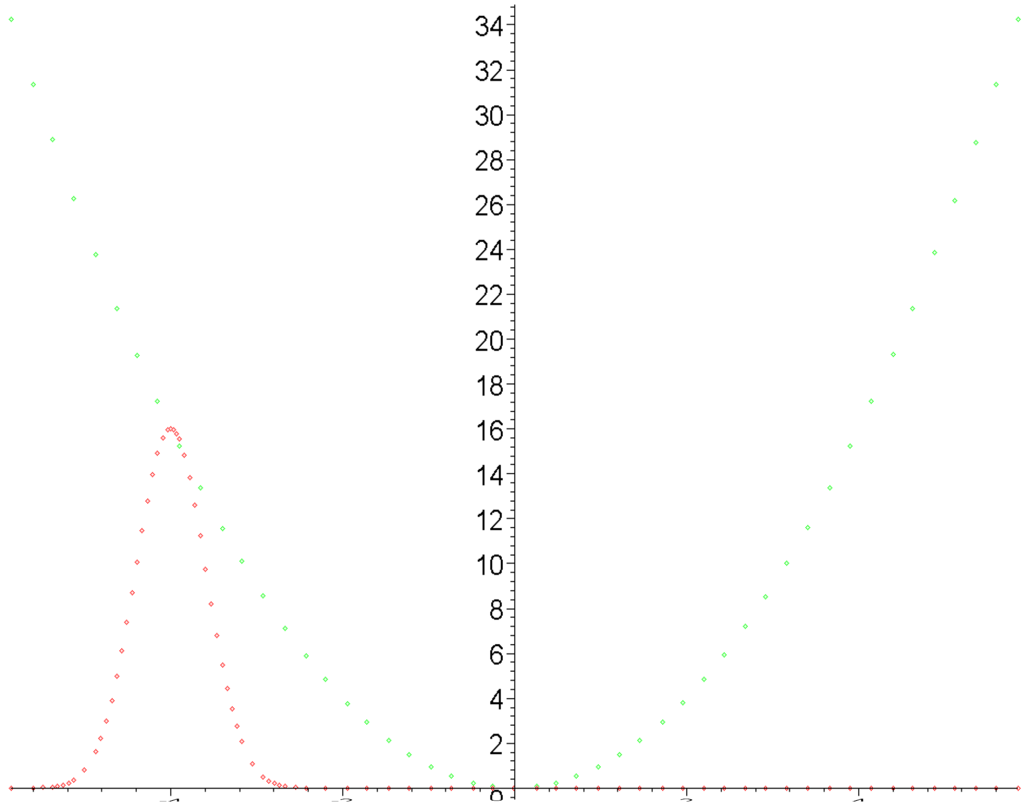
$$139 = 139$$

Počáteční stav zvolíme rozumně lokalizovaný okolo úvrati

```
> x0:=-4;
  psi0:=x->exp(-3*(x-x0)^2);
  Psi0:=evalf(vector([seq(psi0(x(i)),i=1..N)])):
  plot([psi0(x)*v(x0),v(x)],x=x(1)..x(N),style=point);
```

$$x0 := -4$$

$$\psi 0 := x \rightarrow e^{(-3\,(x - x0)^2)}$$




```
>
```

Pro delší numerické výpočty se procedura Jordan může nahradit následujícím pokusem

```
> myjordan:=proc(A,Q)
  local ev;

  ev:=[eigenvectors(A)];
  Q:=matrix(nops(ev),nops(ev),(i,j)->ev[j][3][1][i]);
  RETURN (diag(seq(ev[i][1],i=1..nops(ev))));
  end;
```

$myjordan := \textbf{proc}(A, Q)$

local *ev*;

$ev := [\text{eigenvectors}(A)];$

$Q := \text{matrix}(\text{nops}(ev), \text{nops}(ev), (i, j) \to ev[j][3][1][i]);$

$\text{RETURN}(\text{diag}(\text{seq}(ev[i][1], i = 1 .. \text{nops}(ev))))$

end

```
> DH:=myjordan(H,'Q'):
  Q1:=evalm(Q^(-1)):
  #evalm(H)=evalm(Q&*DH&*Q1):
  #evalf(%,3);
```

```
> #U:=(t)-> evalm(Q &* diag(seq(exp(-I*t*DH[i,i]),i=1..N)) &*Q1
  );
  #
```

```
>
```

```
> Psi_plot:=proc(psi);
    plot([seq([x(i),abs(psi[i])^2],i=1..N)]);
  end;
```

$$Psi\_plot := \textbf{proc}(\psi)\ \text{plot}([\text{seq}([x(i), \text{abs}(\psi[i])\text{^}2], i = 1 .. N)])\ \textbf{end}$$

```
>
```

Stav v pozdějším čase pak spočteme aplikací evolučního operátoru na počáteční stav

```
> Psi1:=(t)-> evalm(Q &* evalm(diag(seq(exp(-I*t*DH[i,i]),i=1..N))
  &*evalm(Q1&*Psi0)) );
```

$$\Psi 1 := t \to \text{evalm}(Q\ \text{`\&*`}\ \text{evalm}(\text{diag}(\text{seq}(\mathbf{e}^{(-I\, t\, DH_{i,i})}, i = 1 .. N))\ \text{`\&*`}\ \text{evalm}(Q1\ \text{`\&*`}\ \Psi 0)))$$

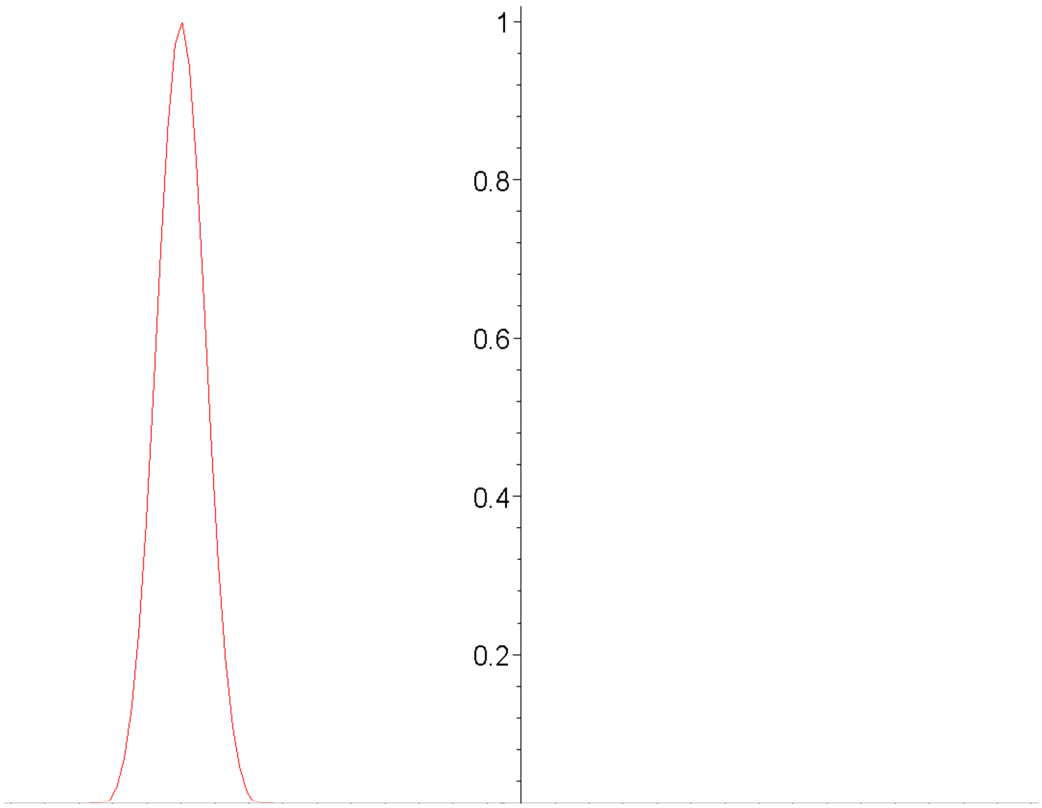
Vypočteme teď všecny fáze animace a uložíme do tabulky

139

```
> for i from 0 to 40 do
   graf[i]:=Psi_plot(Psi1(3.141*i/20));
  od:
```

```
>
```

```
> display(seq(graf[j],j=0..40),insequence=true);
```



Výslednou animaci můžeme uložit do animovaného grafického souboru *plot..gif*

```
> plotsetup(gif,plotoutput=`D:\\plot.gif`);
  display(seq(graf[j],j=0..40),insequence=true);
  plotsetup(default);
>
>
```