Ťahák ALGEBRA – Jupyter + Sage
Predmety ZMF, DGS – Jozef Hanč

Podľa ref. Robert Beezer, Sage Quick reference: Linear Algebra
http://wiki.sagemath.org/quickref

Vektory – zadávanie

Pozor! prvý prvok vektora má poradové číslo 0
`u = vector(QQ,[1,3/2,-1])` 3-zložkový vektor zlomkov
`v = vector(ZZ,[1,8,-2])` 3-zložkový vektor celých čísel
`z = vector(QQ,{2:4,95:4,210:0})` *pozri riedke vektory*
 211 zložiek, nenulová 3. a 96. zložka s hodnotou 4

Vektory – operácie

`v.column()` stlpcový vektor z vektora **v**
`2*u - 3*v` lineárna kombinácia vektorov **u**, **v**
`u.dot_product(v)` alebo `u*v` skalárny súčin **u**·**v**
`u.cross_product(v)` vektorový súčin **u**×**v**
`u.pairwise_product(v)` súčin po zodp. zložkách
`u.norm() == u.norm(2)` euklidovská norma vektora
`u.norm(1)` súčet abs. hodnôt zložiek vektora
`u.norm(oo)` zložka vektora s max. abs. hodnotou

Matice – zadávanie

Pozor! číslovanie riadkov a stĺpcov sa začína od 0
`A = matrix(ZZ, [[1,2],[3,4],[5,6]])`
 3×2 matica celých čísel
`B = matrix(QQ, 2, [1,2,3,4,5,6])`
 vytvorí 2 riadky zo zoznamu; 2×3 matica rac. čísel
`C = matrix(CDF, 2, 2, [[5*I, 4*I], [I, 6]])`
 2×2 matica komplexných čísel; 53-bitová presnosť
`Z = matrix(QQ, 2, 2, 0)` 2×2 matica núl
`D = matrix(QQ, 2, 2, 8)` prvky hl. diag. sú 8, ostat. 0
`H = diagonal_matrix([1,2,3])` s hl. diagonálou $1, 2, 3$
`T = matrix.toeplitz([1,2,3], [4,5])`
 prvky hl. diag. sú 1, na diagonálach pod $2, 3$ a nad $4, 5$
`E = block_matrix([[P,0],[1,R]])`, bloková matica
`II = identity_matrix(5)` 5×5 jednotková matica
 `I` $\equiv \sqrt{-1}$, preto maticiam nedávajte meno `I`
`J = jordan_block(-2,3)` 3×3 matica
 s -2 na hl. diagonále, s 1 na diagonále nad ňou
`var('x y z'); K = matrix(SR, [[x,y+z],[0,x^2*z]])`
 2×2 symbolická matica
`L = matrix(ZZ, 20, 80, {(5,9):30, (15,77):-6})`
 20×80, dva nenulové prvky s danými pozíciami

Maticové priestory – zadávanie a vlastnosti

`M = MatrixSpace(QQ, 3, 4)` priestor matíc 3×4
`A = M([1,2,3,4,5,6,7,8,9,10,11,12])`
 zmení zoznam na prvok `M`, maticu 3×4 nad `QQ`
`M.basis(), M.dimension()` báza a dimenzia priestoru M
`M.zero_matrix()` nulová matica z priestoru M

Násobenie matíc

`u = vector(QQ, [1,2,3]),  v = vector(QQ, [1,2])`
`A = matrix(QQ, [[1,2,3],[4,5,6]])`
`B = matrix(QQ, [[1,2],[3,4]])`
`u*A,  A*v,  B*A,  B^6,  B^(-3)` možné súčiny
`A.elementwise_product(B)` súčin po zodp. zložkách
`B.iterates(v, 6)` postupnosť $vB^0, vB^1, \ldots, vB^5$
`rows = False` dáva násobenie matíc zprava vektorom `v`
`f(x)=x^2+5*x+3`, tak `f(B)` je hodnota funkcie v B
`B.exp()=exp(B)` e na maticový exponent, t.j. $\sum_{k=0}^{\infty} \frac{1}{k!} B^k$

Ďalšie operácie s maticami

`5*A+2*B` lineárna kombinácia
`A.inverse(), A^(-1)` inverzná matica; pre neinvertibilnú (singulárnu) maticu vracia chybu `ZeroDivisionError`
`A.transpose()` transponovaná matica
`A.conjugate()` matica komplexne združených zložiek
`A.conjugate_transpose()` transp. `A.conjugate()`
`A.antitranspose()` transponovaná s opačným poradím
`A.adjoint()` adjungovaná matica = matica kofaktorov
`A.restrict(V)` reštrikcia na invariantný podpriestor `V`

Riadkové a stĺpcové operácie v maticiach

Dané riadkové (stĺpcové) operácie menia danú maticu.
Pozor! prvý riadok má poradové číslo 0
`A.rescale_row(i,a)` `a*`(riadok `i`)
`A.add_multiple_of_row(i,j,a)` `a*`(riadok `j`)+riadok `i`
`A.swap_rows(i,j)` výmena riadkov `i` a `j`
Každá operácia má zodp. stĺpcovú obdobu, `row`→`col`
Ak nechceš zmeniť pôv. maticu, použi uloženie do pamäte
`B = A.with_rescaled_row(i,a)`

Gaussova eliminácia – horná trojuholníková forma

`A.rref()` horná trojuholníková (pod diagonálou sú 0)
`A.echelon_form()` horná trojuholníková s 1 na diagonále
`A.echelonize()` zmení A na hor. troj. s 1 na diagonále
Pozn. `rref()` zmení automaticky na maticu rac. čísel
`A = matrix(ZZ,[[4,2,1],[6,3,2]]);`

`A;`	`A.rref();`	`A.echelon_form()`
$\begin{pmatrix} 4 & 2 & 1 \\ 6 & 3 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$

`A.pivots()` indexy nezávislých stĺpcov (pivotov)
`A.pivot_rows()` to isté pre riadky

Časti matíc a submatice

Pozor! číslovanie riadkov a stĺpcov sa začína od 0
`A.nrows(), A.ncols()` počet riadkov a stĺpcov A
`A[i,j]` prvok matice v riadku i a stĺpci j
`A[i]` riadok `i` ako pythonovská n-tica
`A.row(i)` vráti riadok `i` ako Sage vektor
`A.column(j)` vráti stĺpec `j` ako Sage vektor
`A.list()` zmení na zoznam podľa riadkov (aj pre vektor)
`A.matrix_from_columns([8,2,8])`
 nová matica zo stĺpcov v zozname, môžu sa opakovať
`A.matrix_from_rows([2,5,1])`
 nová matica z riadkov v zozname, nemusia byť v poradí
`A.matrix_from_rows_and_columns([2,4,2],[3,1])`
 sučasne z daných riadkov a stĺpcov
`A.rows()` všetky riadky ako zoznam n-tíc
`A.columns()` všetky stĺpce ako zoznam m-tíc
`A.submatrix(i,j,nr,nc)`
 začne na prvku `(i,j)`, použije `nr` riadkov, `nc` stĺpcov
`A[2:4,1:7]` matica z riadkov 2 až 4 a stĺpcov 1 až 7
`A[0:8:2,3::-1]` pythonovský výber riadkov a stĺpcov

Spájanie matíc

`A.augment(B)` `A` rozšíri zprava o `B`; `B` môže byť vektor
`A.stack(B)` `A` rozšíri zdola o `B`; `B` môže byť vektor
`A.block_sum(B)` vráti maticu s `A` a `B` na diagonále
`A.tensor_product(B)` tenzorový súčin `A` a `B`

Skalárne funkcie matíc

`A.rank()` hodnosť = počet nezávislých stĺpcov (riadkov)
`A.right_nullity()` počet závislých stĺpcov
`A.left_nullity() == A.nullity()` počet záv. riadkov
`A.determinant() == A.det()` determinant matice
`A.permanent(),  A.trace()` stopa matice
`A.norm() == A.norm(2)` l_2 (spektrálna) norma
`A.norm(1)` najväčšia stĺpcová suma abs. hodnôt
`A.norm(oo)` najväčšia riadková suma abs. hodnôt
`A.norm('frob')` Frobéniova (Euklidovská) norma

Matice – vlastnosti

`.is_zero()`; `.is_symmetric()`; `.is_hermitian()`;
`.is_square()`; `.is_orthogonal()`; `.is_unitary()`;
`.is_scalar()`; `.is_singular()`; `.is_invertible()`;
`.is_one()`; `.is_nilpotent()`; `.is_diagonalizable()`

Vlastné čísla a vlastné vektory matíc

Pozn. eigenvalue(vector) = vlastná hodnota(vektor)
`A.charpoly('t')`, bez premennej automaticky berie `x`
`A.characteristic_polynomial() == A.charpoly()`
`A.fcp('t')` charakteristický polynóm rozložený na súčin
`A.minpoly()` minimálny polynóm
`A.minimal_polynomial() == A.minpoly()`
`A.eigenvalues()` netriedený zoznam aj s násobnosťou
`A.eigenvectors_right()` vlastné vektory zprava;
vracia pre každú vlastnú hodnotu trojicu:
`e`: vlastnú hodnotu;
`V`: zoznam bázových vlast. vektorov; `n`: násobnosť
`A.eigenmatrix_right()` vlastné vektory zprava;
vracia dvojicu: `D`: diagonálna matica vlast. hodnôt
`P`: vlastné vektory ako stĺpce
vracia nulové stĺpce pri nediagonalizovateľnosti
Pozn. každému príkazu `_right` zodpovedá príkaz `_left`

Rozklady matíc

Pozn. dostupnosť závisí na zvolenom okruhu matíc;
použi `RDF`,`CDF` pre numerické výpočty; `QQ`,`AA` exaktné
unitárnosť = ortogonalita pre mat. s prvkami v $\mathbb{R}$
`A.jordan_form(transformation=True)`
vráti dvojicu matíc: pre `A == P^(-1)*J*P`
`J`: matica Jordan. blokov pre vlastné hodnoty
`P`: nesingulárna matica
`A.smith_form()` trojicu: pre `D == U*A*V`
`D`: elementárne delitele na diagonále
`U, V`: s jednotkovým determinantom
`A.LU()` trojica: pre `P*A == L*U`
`P`: permutačná matica
`L`: dolná trojuholníková, `U`: horná trojuholníková
`A.QR()` dvojica: pre `A == Q*R`
`Q`: unitárna matica, `R`: horná trojuholníková
`A.SVD()` trojica: pre `A == U*S*(V-conj-transpose)`
`U`: unitárna matica
`S`: diagonálna s rozmermi `A`
`V`: unitárna matica
`A.schur()` dvojica: pre `A == Q*T*(Q-conj-transpose)`
`Q`: unitárna matica
`T`: horná-trojuh. matica; môže byť tvar 2×2 diag. bloky
`A.rational_form()` Frobéniova forma
`A.symplectic_form()` symplektická forma
`A.hessenberg_form()` Hessenbergova forma
`A.cholesky()` Choleského rozklad

Sústavy rovníc – riešenie cez matice

`A.solve_right(B)`, odbobne príkaz s `_left`
je riešenie sústavy `A*X = B`, `X` vektor **či** matica neznámych
`A = matrix(QQ, [[1,2],[3,4]])`
`b = vector(QQ, [3,4])`; `A\b` dáva riešenie `(-2, 5/2)`

Vektorové priestory – zadávanie

`VectorSpace(QQ, 4)` 4-rozmerný nad poľom $\mathbb{Q}$
`VectorSpace(RR, 4)` 4-rozmerný nad poľom $\mathbb{R}$
`VectorSpace(RealField(200), 4)`
pole R má 200-bitovú presnosť (štandardne 53)
`CC^4` vektorový priestor 4-rozmerný nad poľom $\mathbb{C}$
`Y = VectorSpace(GF(7), 4)` konečný priestor, ktorý
má $7^4 = 2401$ prvkov, uložených v `Y.list()`

Vektorové priestory – vlastnosti

`V.dimension()` dimenzia priestoru V
`V.basis()` báza priestoru V
`V.echelonized_basis()` báza po Gaussovej eliminácii
`V.has_user_basis()` má V nekanonickú bázu?
`V.is_subspace(W)` pravda =`True`, ak `W` je podpriestor `V`
`V.is_full()` rovná sa hodnosť stupňu modulu?
`Y = GF(7)^4`, `T = Y.subspaces(2)`
`T` je generátor 2D podpriestorov v `Y`
`[U for U in T]` je zoznam 2850 2D podpiestorov v `Y`,
použi `T.next()` na prechádzanie cez podpriestory

Vektorové podpriestory – zadávanie

`span([v1,v2,v3], QQ)` obálka vektorov nad zvol.poľom

Pre maticu `A`, dostávame:
vektorové priestory, ak okruhom je pole
moduly, ak okruhom je len okruh
`A.left_kernel() == A.kernel()` jadro; obd. `right_`
`A.row_space()==A.row_module()` riadkový priestor
`A.column_space()==A.column_module()` stĺpc. priestor
`A.eigenspaces_right()` vektory zprava; obd. `_left`
Dvojice: vlastné hodnoty a vektory zprava
`A.eigenspaces_right(format='galois')`
1 vlast. priestor na ireducibilný činiteľ char. polynómu

Ak `V` a `W` sú podpiestory, tak
`V.quotient(W)` faktorový priestor
`V.intersection(W)` prienik `V` a `W`
`V.direct_sum(W)` direktný súčet `V` a `W`
`V.subspace([v1,v2,v3])` podpriestor V z vektorov

Plné a riedke vektory, matice

Pozn. algoritmy závisia na zvol. reprezetácii;
vektory a matice majú dve reprezentácie:
plná (dense): používame zoznamy
riedka (sparse): používame slovníky
`.is_dense()`, `.is_sparse()` zistí reprezentáciu
`A.sparse_matrix()` vracia riedku verziu `A`
`A.dense_rows()` vracia plné riadkové vektory z `A`
Niektoré príkazy obsahujú `sparse` ako boolovský výraz.

Okruhy a ich vlastnosti

Pozn. Algoritmy zvyčajne závisia od zvol. okruhu
`<object>.base_ring(R)` pre vektory, matice, ...
zisti zvolený okruh
`<object>.change_ring(R)` pre vektory, matice, ...
zmeň okruh (pole) na okruh (pole) `R`
`R.is_ring()`, `R.is_field()`, `R.is_exact()`
Hlavné vstavané okruhy a polia v systéme Sage
`ZZ` okruh celých čísel
`QQ` pole racionálnych čísel
`AA`, `QQbar` polia algebraických čísel (nekon. presnosť)
`RDF` pole reálnych čísel, 53 bitov (pre rýchle výpočty)
`CDF` pole kompl. čísel, 53 bitov (pre rýchle výpočty)
`RR` pole reálnych čísel, 53 bitov
`RealField(400)` 400-bitová presnosť
`CC`, `ComplexField(400)` 400-bitová presnosť
`RIF` pole reálnych intervalov
`IntegerModRing(4)` okruh zvyškových tried mod 4
`GF(2)` pole zvyškových tried mod 2
`GF(p)==FiniteField(p)` zvyš. triedy mod p(prvočíslo)
`Integers(6)` okruh zvyš. tried mod 6
`CyclotomicField(7)` rac. čísla so 7. jedn. koreňom
`QuadraticField(-5, 'x')` rac. čísla s `x=`$\sqrt{-5}$
`SR` okruh symbolických premenných a výrazov

Poznámka k vektor. priestorom a modulom

Modul "je" vektor. priestor nad okruhom (nie nutne poľom).
Mnoho príkazov z vekt. priestorov možno aplikovať na moduly.
Niektoré "vektory" sú v skutočnosti prvky modulov.

Kde hľadať pomoc - Jupyter

"kláves Tab" dokončenie príkazu, vlastností, parametrov
"stlačenie Shift+Tab" zobrazenie vlastností a popisu objektu
`<command>?+Shift+Enter` = popis príkazu a pomoc s príkladmi
`<command>??+Shift+Enter` = pythonovský kód príkazu