6A Tehtävä M2

$with(LinearAlgebra)$

[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, CompressedSparseForm, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm, FromSplitForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, ProjectionMatrix, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, SplitForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip] **(1)**

$with(VectorCalculus)$

[&x, \`*\`, \`+\`, \`-\`, \`.\`, <,>, <|>, About, AddCoordinates, ArcLength, BasisFormat, Binormal, ConvertVector, CrossProduct, Curl, Curvature, D, Del, DirectionalDiff, Divergence, DotProduct, Flux, GetCoordinateParameters, GetCoordinates, GetNames, GetPVDescription, GetRootPoint, GetSpace, Gradient, Hessian, IsPositionVector, IsRootedVector, IsVectorField, Jacobian, Laplacian, LineInt, MapToBasis, ∇, Norm, Normalize, PathInt, PlotPositionVector, PlotVector, PositionVector, PrincipalNormal, RadiusOfCurvature, RootedVector, ScalarPotential, SetCoordinateParameters, SetCoordinates, SpaceCurve, SurfaceInt, TNBFrame, TangentLine, TangentPlane, TangentVector, Torsion, Vector, VectorField, VectorPotential, VectorSpace, Wronskian, diff, eval, evalVF, int, limit, series] **(2)**

Ratkaistaan pintojen todelliset leikkauspisteet, niin keksitään alkuarvaus Newtonin menetelmälle :D

$$solve(\{y^2 + z^2 = 4, x^2 + z^2 = 3, x^2 - z = 1\}, \{x, y, z\})$$

$$\{x = RootOf(\_Z^2 - 2), y = RootOf(\_Z^2 - 3), z = 1\}, \{x = RootOf(\_Z^2 + 1), y = 0, z = -2\} \quad (3)$$

$$evalf(\%)$$

$$\{x = 1.414213562, y = 1.732050808, z = 1.\}, \{x = I, y = 0., z = -2.\} \quad (4)$$

Saatiin siis ratkaistua yhtälöryhmälle kaksi ratkaisua. Näistä ensimmäinen on se, mitä etsitään, koska haluttiin leikkauspiste 1. oktantissa (kaikki koordinaatit positiivisia).

Kokeillaan nyt Newtonin menetelmää alkuarvauksella (1, 1, 1).

Määritetään Jacobin determinantti:

$$J := Jacobian([y^2 + z^2 - 4, x^2 + z^2 - 3, x^2 - z - 1], [x, y, z])$$

$$J := \begin{bmatrix} 0 & 2\,y & 2\,z \\ 2\,x & 0 & 2\,z \\ 2\,x & 0 & -1 \end{bmatrix} \quad \textbf{(5)}$$

Ja vektori f(x):

$f := Vector\left(\left[y^2 + z^2 - 4,\, x^2 + z^2 - 3,\, x^2 - z - 1\right]\right)$

$$f := \left(y^2 + z^2 - 4\right)e_x + \left(x^2 + z^2 - 3\right)e_y + \left(x^2 - z - 1\right)e_z \quad \textbf{(6)}$$

Ensimmäinen iteraatio:

$x\_1 := Vector([1, 1, 1]) - eval(MatrixInverse(J), \{x = 1, y = 1, z = 1\}) \cdot eval(f, \{x = 1, y = 1, z = 1\})$

$$x\_1 := \left(\frac{3}{2}\right)e_x + (2)e_y + (1)e_z \quad \textbf{(7)}$$

Toinen iteraatio:

$x\_2 := x\_1 - eval(MatrixInverse(J), \{x = x\_1[1], y = x\_1[2], z = x\_1[3]\}) \cdot eval(f, \{x = x\_1[1], y = x\_1[2], z = x\_1[3]\})$

$$x\_2 := \left(\frac{17}{12}\right)e_x + \left(\frac{7}{4}\right)e_y + (1)e_z \quad \textbf{(8)}$$

Kolmas iteraatio:

$x\_3 := x\_2 - eval(MatrixInverse(J), \{x = x\_2[1], y = x\_2[2], z = x\_2[3]\}) \cdot eval(f, \{x = x\_2[1], y = x\_2[2], z = x\_2[3]\})$

$$x\_3 := \left(\frac{577}{408}\right)e_x + \left(\frac{97}{56}\right)e_y + (1)e_z \quad \textbf{(9)}$$

Neljäs iteraatio:

$x\_4 := x\_3 - eval(MatrixInverse(J), \{x = x\_3[1], y = x\_3[2], z = x\_3[3]\}) \cdot eval(f, \{x = x\_3[1], y = x\_3[2], z = x\_3[3]\})$

$$x\_4 := \left(\frac{665857}{470832}\right)e_x + \left(\frac{18817}{10864}\right)e_y + (1)e_z \quad \textbf{(10)}$$

$evalf(x\_4)$

$$(1.414213562)e_x + (1.732050810)e_y + (1.)e_z \quad \textbf{(11)}$$

Alussa laskettiin todelliseksi leikkauspisteeksi

$\{x = 1.414213562, y = 1.732050808, z = 1.\}$, eli jo 4 iteraatiolla päästiin lähes täydellisesti oikeaan pisteeseen.