



LUNDS UNIVERSITET  
Lunds Tekniska Högskola

Kursplan för läsåret 2011/2012  
(Genererad 2011-08-31.)

---

## AVANCERAD KURS I NUMERISKA ALGORITMER FMNN25 MED PYTHON/SCIPY

Advanced Course in Numerical Algorithms with Python/SciPy

**Antal högskolepoäng:** 7,5. **Betygsskala:** UG. **Nivå:** A (Avancerad nivå). **Huvudområde:** Teknik. **Undervisningsspråk:** Kursen ges på begäran på engelska. **Valfri för:** D4, E4, E4pv, F4, F4bs, Pi4. **Kursansvarig:** Claus Führer, Claus.Fuhrer@na.lu.se, Numerisk analys. **Förutsatta förkunskaper:** Grundkurs i numerisk analys. Kunskap i några av programmeringsspråken Java, C, C++, Fortran, Python samt MATLAB. **Prestationsbedömning:** Veckovisa programmeringsprojekt. Ett större programmeringsprojekt i grupp, med tillhörande skriftlig rapport som redovisas muntligen inför övriga kursdeltagare. Opposition på någon annan grupps rapport. **Obligatorisk närvaro** vid alla redovisningar. **Hemsida:** <http://www.maths.lth.se/na/courses/>.

### Syfte

Kursen är tänkt som ett algoritmororienterat komplement till de mer på metodanalys inriktade grund- och specialkurserna i numerisk analys. Den betonar kopplingen mellan komplexa numeriska algoritmer och moderna programmeringsspråk.

### Mål

#### *Kunskap och förståelse*

För godkänd kurs skall studenten

- ha fått en förståelse för beräkningsalgoritmers grundprinciper.
- ha fördjupat sin kännedom om ett antal viktiga beräkningsproblem, och sätt att angripa dem.

#### *Färdighet och förmåga*

För godkänd kurs skall studenten

- ha förvärvat numerisk programmeringsfärdighet på hög nivå.
- ha lärt sig att koda, testa och resultatsbedöma komplexa numeriska algoritmer, med utnyttjande av beprövade programbibliotek.
- kunna genomföra ett programmeringsprojekt i grupp, inklusive identifikation av och

uppdelning i delproblem, och eget ansvar för lösandet av ett delproblem.

- kunna redogöra för ett beräkningsprojekt, såväl i en muntlig presentation som i en skriftlig rapport.

### **Innehåll**

Introduktion till Python utgående från programmeringskunskap i andra språk/verktyg. Objektorienterad programmeringsstil i beräkningsteknik. Scipy/Numpy datastrukturer.

Exempel på komplexa numeriska algoritmer från olika områden inom numerisk analys.

Koppling till beräkningsbibliotek i C och Fortran (Netlib).

Automatiserade test i beräkningsprogrammering. Grafisk representation av numeriska resultat (animering). Python för att styra systemprocesser.

Innehållet kan kompletteras med specialkunskap från eventuella gästlärare

### **Litteratur**

Führer, C, Solem, J.E., Verdier, O: Scientific Computing with Python. Matematikcentrum 2011.