

Nyfikenhet och otillräcklighet

Gymnasielärares erfarenheter av att undervisa matematik med programmering

En lärartanke: Det är konstigt att för att få undervisa i programmeringskurser krävs det 90 högskolepoäng men för att tillämpa programmering i matematikkurserna krävs ingen kunskap i programmering.

Programmering har varit ett ämne för flera artiklar i Nämnaren redan från 1980-talet. De flesta artiklarna har diskuterat programmering ur ett problemlösningsperspektiv. Men det finns även andra tillämpningsområden för programmering i matematik, såsom algebra, vilket också syns i kursplaner för den svenska grundskolan.

I två artiklar rapporterar Kilhamn, Bråting och Rolandsson om ett projekt som syftade till att undersöka både hur programmering som en del av matematikämnet formas i praktiken och hur implementering av programmering eventuellt har berikat eller försvårat den existerande matematikundervisningen i grundskolan. Genom att analysera bland annat dokumentationen från 32 lesson studies som lärare genomfört inom ramen för ämnet matematik i grundskolan identifierade forskarna fyra olika roller som programmering gavs i lektionerna. Den vanligaste rollen beskrev relationen där matematik endast utgjorde en kontext där undervisningen syftade till att elever skulle lära sig programmering. Men det fanns också lektioner där deltagande lärare tänkte att programmering kan vara ett verktyg för att undervisa matematik. Programmering sågs ibland som ett användbart verktyg för att effektivisera beräkningar inom olika kunskapsområden och kunde även fungera väl som ett verktyg för att utforska till exempel samband och förändring eller geometriska objekt.

Ett förvånande resultat i Kilhamns, Bråtings och Rolandssons projekt var att matematisk problemlösning inte identifierades som ett lämpligt kunskapsområde för programmering. När deltagande lärare pratade om problemlösning var problemen oftast formulerade utifrån programmeringen, medan matematiken som skulle ingå i programmet var nästan trivial.

En kritisk diskussion

I en tidigare artikel, *Programmering på gymnasiet*, berättade vi om ett tillvägagångssätt att använda programmering för att stödja elever att utveckla förståelse av matematiska begrepp. Ett argument för användbarheten av programmering för detta ändamål är att kodning kan synliggöra de steg som eleven enligt APOS-teorin ska ta för att utveckla fördjupad förståelse av matematiska objekt. Samtidigt vet vi att utveckling av förtrogenhet med grundläggande matematiska begrepp och metoder och deras användbarhet är ett av de svåraste målen i matematikundervisning på alla nivåer. Därför vill vi i denna artikel kritiskt diskutera vilka möjligheter och utmaningar vi ser med att undervisa matematik med programmering. Vår diskussion är baserad på ett forskningsmaterial som består av Teams-inspelningar och enkät-svar från en grupp gymnasieelever samt en diskussion med deltagande lärare.

Också David Taub har i artikeln *Vad hände med programmeringen?* reflekterat över delvis motsvarande frågor som vi gör här. I sin artikel diskuterar han om det är möjligt att använda programmering i matematikundervisning så att elever verkligen utvecklar djupare förståelse för matematik. En av hans slutsatser är att elever ska vara ganska bekväma med programmering innan de kan dra nytta av den när de lär sig ny matematik.

Lite om oss och bakgrunden till vår studie

Under läsåret 2018–19 fick två av oss, Alf Julin och Anna Wedestig, tillsammans med övriga matematiklärare vid naturvetenskaps- och teknikprogrammen vid Luleå gymnasieskola utbildning i programmering ledd av kollegan Claes Johansson som är lärare både i matematik och programmering. Utbildningen var en timma varannan vecka och följde ett fortbildningskompendium, *Matematisk programmering i Python*, från Malmö.

Timo Tossavainen är professor i matematikdidaktik vid Luleå tekniska universitet. Han har läst några kurser i programmering på 1990-talet och har emellanåt undervisat programmering i olika kurser i ämneslärarutbildningen. Laborationen som genomfördes under projektet var en del av en pilotstudie i Timos forskning varför han deltog vid laborationen och även senare under projektet.

Spetsutbildning i matematik

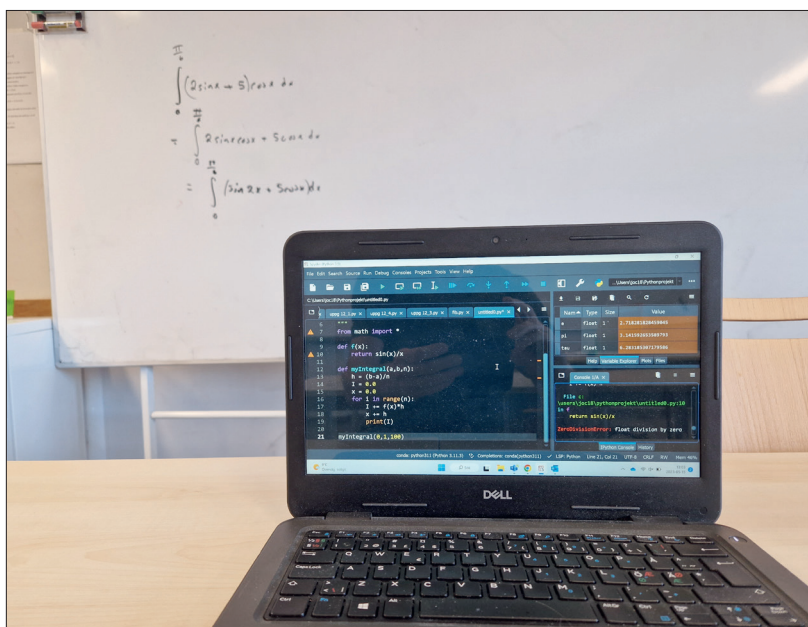
Luleå gymnasieskola har en spetsutbildning i matematik. På vårterminen 2022 genomförde vi en laboration som handlade om numeriska integraler. De elever som deltog läste andra året på spetsutbildningen. De hade jobbat med integraler både analytiskt och numeriskt innan den lektion som vi rapporterar om. Eleverna hade också läst cirka 15 timmar programmering som en del av en tidigare kurs.

För datainsamling indelades eleverna i grupper om tre och under lektionen spelade vi in gruppernas arbete via Teams så att vi i efterhand kunde analysera deras diskussioner och se deras skärm och hur de jobbade. Någon grupp jobbade på mer än en dator och då missade vi möjligheten att se vad som hände på andra datorer men överlag fick vi en bra bild av gruppernas verksamhet.

Laboration om integraler

Vid den inspelade laborationen fick eleverna se en kod som numeriskt beräknar bestämda integraler med en vänster rektangelsumma:

```
def f(x):  
    return cos(x)  
  
def minIntegral(a,b,n):  
    h = (b-a)/n  
    I = 0.0  
    x = 0.0  
    for i in range(n):  
        I += f(x)*h  
        x += h  
    print(I)  
  
minIntegral(0,1,100)
```



Eleverna skulle med hjälp av koden studera några integraler och reflektera över hur numerisk beräkning av integraler fungerar.

Syftet med laborationen var till exempel att undersöka hur värdet på en rektangelsumma påverkas av indelning av intervallet och om funktionen är växande eller avtagande. Lektionen var utformad under Kleindagarna i juni 2018.

Det tar tid att lära sig dra nytta av programmering

De flesta grupper stötte på problem när de skulle beräkna $\int_0^1 \frac{\sin x}{x} dx$ då det med den kod de fick ledde till division med 0. Efter mycket funderande löste en del av dem problemet på lite olika sätt.

De flesta svårigheter och diskussioner rörde programmeringstekniska problem och det uppstod få diskussioner om matematiska problem. Vissa grupper fick helt orimliga svar men gick utan större eftertanke vidare. Till exempel skrev en grupp av koden fel och fick integralen $\int_0^1 x dx$ till först 9 sedan 99 och till sist 999 – de skrev ut antalet rektanglar i stället för arean, men de skrev glatt ner resultatet och gick vidare. Om de hade tänkt efter vad det var de beräknade så skulle alla i gruppen ha kunnat beräkna den integralen med huvudräkning. Detta är förvånande med tanke på att de var mattespetselever. Enligt vår tolkning berodde elevernas felaktiga slutsatser kring ganska enkla matematiska problem framförallt på deras ringa erfarenhet från att ha studerat matematik med programmering. Eleverna fokuserade så intensivt på att förstå koden och åtgärda de fel som de gjorde när de kopierade koden från klassrummets skärm att deras uppmärksamhet inte räckte till att upptäcka matematiken i koden.

Resultatet kan jämföras med när vi vid ett senare tillfälle under vårterminen 2023 genomförde samma uppgift i en grupp med motsvarande matematiska bakgrund men med skillnaden att de även hade läst nästan hela gymnasiekursen *Programmering 1*. Då framgick tydligt av elevernas aktiviteter att diskussionerna fokuserade mer på det matematiska. Koden hade de ganska fort analyserat och visste hur den fungerade till skillnad från första gruppen som bara delvis förstod koden. Också Taub rapporterade om motsvarande slutsatser.

Elevernas tankar

När eleverna hade läst en stor del av kursen *Programmering 1* genomfördes en enkät där de fick beskriva vad de tyckte om programmering i allmänhet samt om programmering i matematikkurser. Här redovisas en del svar från den enkäten.

Cirka två tredjedelar av eleverna upplevde inte att deras matematiska kunskaper stärkts av att de läst programmering. Detta resultat motsvarar väl vad många tidigare studier har rapporterat om elevers erfarenheter från att ha använt programmering (eller IKT över huvud taget) i matematik. De flesta elever ser ingen större koppling mellan programmeringen och matematik – som någon påpekade känns det fortfarande som två olika ämnen. Endast en elev menade att de matematiska kunskaperna stödjer programmeringen.

Man förstär algoritmer bättre. För att datorn ska förstå vad du menar så måste man själv förstå vad man gör steg för steg.

Känner inte att programmeringen har stärkt mitt matematiska tänkande, utan snarare mitt logiska tänkande och förmåga att lösa problemen i programmering.

Programmering har hjälpt mig att bryta ner ett problem till mindre bitar.

Skulle vilja påstå att mina matematiska kunskaper stöder programmering å andra sidan, men skulle inte vilja säga att programmeringen hjälpt speciellt mycket, med det har inte heller påverkat negativt.

Intressant nog var elevernas motivation att läsa en fristående kurs i programmering, efter att de fått testa programmering i sin specialiseringskurs, endast svagt eller inte alls kopplad till matematik. Detta trots att en stor del av deltagarna efter laborationen valde att läsa kursen *Programmering I* under nästkommande läsår tillsammans med elever från en annan årskurs. Exempel på typiska motiveringar till att göra så är:

Att få grunder i programmering tror jag kommer vara väldigt viktigt i framtiden och borde vara del av grundundervisning precis som matematik och svenska. Utöver att bara lära sig hur man programmerar hade man även kunna lära sig hur man skyddar sig på nätet och skyddar sin dator samt informationen på den. Ser inget negativt med att läsa programmering.

Jag tycker att det är roligt att få lära sig lite om programmering, eftersom det är något som jag inte har testat på tidigare. Jag tror också att alla kan ha nytta av att ha grundläggande kunskaper i programmering, vilket man får genom att läsa programmering nu på gymnasiet.

Matematiklärares tankar

I resten av denna artikel ska vi reflektera över matematiklärares erfarenheter från egen kompetensutveckling i programmering. Alf och Anna, och ett dussin andra matematiklärare som deltog i projektet, har fått bekanta sig med programmering på ett ganska vanligt förekommande sätt: de har fått sin utbildning i programmering av en kollega. Från början ställde sig lärarna positiva till att lära sig programmering för att kunna använda det i matematikundervisningen, men den ambitionen kom på skam. Att lära sig att programmera "lite grann" räcker inte för att använda programmering i matematikkurser.

Förutom undervisning fylls en lärares vardag av lektioner som ska förberedas, prov som ska rättas, utvecklingssamtal som ska genomföras och så klart möten. Mellan programmeringslektionerna som äger rum varannan vecka blir det sällan av att sätta sig ned och programmera. Följande citat beskriver väl de omständigheter där verksamma matematiklärare försöker utveckla sin kompetens i programmering:

Kollegan som var vår lärare beskrev det som att börja om från början varje gång. Vi försökte igen tillsammans med spets eleverna när de fick programmering inlagt i sin specialiseringskurs. Så drabbade en pandemi landet, vi undervisade på distans eller delvis på distans. Allt för ofta upptäckte jag att jag befann mig i ett breakout room med en kollega där vi avhandlade andra saker, jobbrelaterade men inte kopplade till programmering.

Men lärarna säger också att de har lärt sig lite om programmering. Den första tidens panikkänsla är borta. De berättar att de kan följa små exempel som finns i böckerna, men de känner sig inte kunniga nog att uppträda som bollplank för sina elever eller svara på frågor om varför en elevs program inte fungerar. De ställer sig gärna vid sidan av och hoppas att eleverna klarar det själva, vilket är en obehaglig känsla för en lärare. En lärare säger:

Vi har lagt ner stora delar av vår fortbildning under ett par år på programmering, utan att jag känner mig trygg i att programmera. Troligtvis hade det funnits bättre fortbildning för mig. Det är konstigt att för att få undervisa i programmeringskurser krävs det 90 högskolepoäng men för att tillämpa programmering i matematikkurserna krävs ingen kunskap i programmering.

Programmeringens potential och införlivande i matematikämnet

Trots en viss osäkerhet gällande sina kunskaper i programmering ser lärarna att programmering kan vara ett värdefullt redskap i matematikundervisningen:

Det blev en intressant och bra diskussion om högersumma respektive vänstersumma i en funktion som är växande. Eleverna kommer fram till att en högersumma blir en underdrift och att en vänstersumma blir en överdrift. Men sedan när dom försöker lösa integralen med ett program ser dom att både höger- och vänstersumman ger en underdrift. Utan någon vidare kommentar så lämnar dom problemet där. Ingen reflektion om att dom eventuellt gjort något fel vid programmeringen. Jag misstänker att deras osäkerhet i programmering gör att dom inte vill göra om något i koden, när dom ändå fått ett rimligt svar. Hade jag nu varit en matematiklärare som dessutom var helt trygg i programmering och om eleverna också kunde programmera så skulle nu detta varit en perfekt uppgift att gå vidare med. Men nu känns det som min och deras osäkerhet i själva programmeringen blir ett hinder.

En annan lärare fortsätter:

Om man skulle göra motsvarande aktivitet i en klass som inte läst någon programmering alls på gymnasiet så skulle en mycket stor andel av tiden gå åt till att få eleverna att kunna köra koden i någon utvecklingsmiljö (idé- eller webbaserad). Sedan till att få dem att förstå hur iterationer, funktioner med mera fungerar skulle ta mycket tid i anspråk utan att ge så mycket tillbaka i form av matematiska kunskaper. Om de däremot redan läst en kurs i programmering så kan man ha möjlighet att använda programmering som en del i matematikundervisningen.

Hur borde programmering införlivas i matematikundervisningen? Lärarna tänkte att om man ska använda programmering i matematikkurser så skulle användning av exempelkoder som eleverna kan utgå från skapa en möjlighet att lägga fokus på matematiken. Då blir programmet för de allra flesta en "black box" som gör något. Enligt lärarnas åsikt finns det väldigt få elever som kan eller ger sig tid till att analysera koden och förstå vad den gör. Resultatet av den rapporterade laborationen visar att den utdata som programmet ger accepteras utan någon större eftertanke. Om eleverna ser att det är något fel så testar de att ändra något i koden tills det blir rätt, sedan går de vidare utan att tänka så mycket mer. Ibland leder resultaten till diskussioner av matematisk karaktär vilket leder till djupare insikt i ämnet, men oftare är det problem med koden och syntaxen som de fastnar i. Exempelkoder bör nog ses som labbinstruktioner, som kräver genomgångar innan och efter samt mer tid och lärarstyrning för att leda till lärande.

Laborationer i programmering

Lärarna uttryckte att det finns en övertro på att programmering kan effektivisera matematikinläring. Enligt deras åsikter ger programmering dock variation i matematikundervisningen. Programmering kan på många sätt liknas vid de laborationer man har i fysik, kemi och biologi. En skillnad är att laborationerna har varit med från början och haft en självklar plats i de ämnena, och att lärarna redan i sin grundutbildning har förberett sig på att bedriva laborationer både teoretiskt och praktiskt, på universitetet och på VFU. Vanligt är också, på grund av labbsalarnas utformning, att man bedriver laborationerna i halvklass. Under de förhållandena har läraren möjlighet

att på ett professionellt sätt stötta eleverna. Trots detta är det inte självklart att laborationerna leder till en ökad förståelse av ämnet. Många skulle snarare säga att laborationerna syftar till att förbereda eleverna inför högre studier, genom att de lär sig genomföra laborationer och skriva labbrapporter.

När det gäller programmering saknar de flesta lärarna utbildning både i programmering och dess didaktik. Det finns inte tid avsatt för laborationer i ämnet matematik där programmering utöver den ”vanliga skolmatematiken” också konkurrerar med att lära sig handha andra digitala hjälpmedel. Två ganska typiska citat som beskriver utmaningar och möjligheter som lärarna erfar är:

Det man hoppas är att matematiken skall stärkas av programmeringen och vice versa, min erfarenhet är nog snarare tvärtom.

Eleverna vill i första hand få bra betyg, allt som ligger lite utanför tar tid från lektionerna och det skapar en stress hos både elever och lärare. Det är omöjligt att klämma in programmering utan att stryka andra moment eller lägga ut mer tid i kurserna.

Det verkar vara vanligt bland lärarna att de känner sig otillräckliga i interaktionen med sina elever, till exempel att de inte kan utmana elevernas tänkande och inte kan svara på deras frågor. En lärare uttrycker det så här: ”Jag har inget språk för att kommunicera programmering.”

För att sammanfatta: lärarnas erfarenheter motsvarar ganska väl det som tidigare har rapporterats av såväl Taub som Bråting, Rolandsson och Kilhamn. En betydande skillnad mellan studierna är att vi har samlat in material från elever med särskilt intresse för matematik. Trots detta hade eleverna svårigheter med att använda programmering som ett redskap för att stödja eller förstärka sitt lärande i matematik. Även elever med mer erfarenhet av programmering såg matematik och programmering som olika slags kunskapsområden fastän de tyckte att båda till exempel handlar om problemlösning. Om läget är sådant för elever på en spetsutbildning i matematik, väcks frågan om matematik och programmering borde undervisas som separata skolämnen för alla elever.

LITTERATUR

- Bjerner, B. (1984–85). *Kommer datalogin att påverka matematikutbildningen?* Nämnaren 1984–85:2.
- Bråting, K., Rolandsson, L. & Kilhamn, C. (2023). *Programmering och skolmatematik*. Nämnaren 2023:1.
- Forss, M. (2020). *Programmering som verktyg för problemlösning*. Nämnaren 2020:1.
- Johansson, A. (2018). *Ekvationen $x^y=y^x$. Exempel på problemlösning med hjälp av programmering*. Nämnaren 2018:3.
- Johansson, C., Juhlin, A., Tossavainen, T. & Wedestig, A. (2021). *Programmering på gymnasiet*. Nämnaren 2021:4.
- Kilhamn, C., Bråting, K. & Rolandsson, L. (2021). *Programmering i skolmatematiken?* Nämnaren 2021:4.
- Taub, D. (2023). *Vad hände med programmeringen?* Nämnaren 2023:1.
- Länk till Kleindagarna: www.kleindagarna.se
- Länk till kompendium från Malmö: sites.google.com/skola.malmo.se/programmeringsfortbildning/start sida?pli=1