Numerisk lösning av PDE: Comsol Multiphysics

I denna lab ska du bekanta dig med programmet Comsol Multiphysics för numerisk lösning av PDE med finita element. Programmet har många faciliteter och vi kommer bara att använda en liten del.

Hjälp

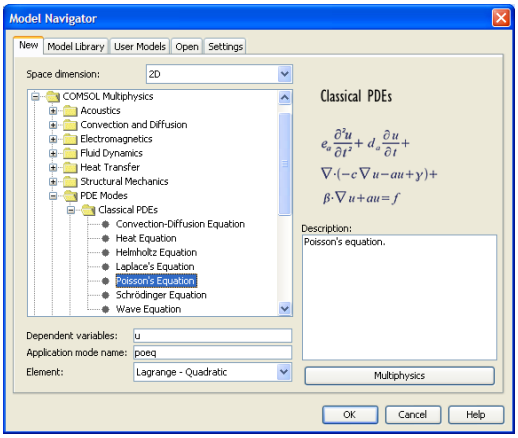
I menyraden överst finns längst till höger en hjälp-knapp ? som ger tillgång till all dokumentation (om man installerat alla filer). Lämpligt att starta med *Quick Start*, men den är ganska omfattande, och GUI-funktionerna är förhoppningsvis ganska intuitiva.

Exempel 1, Poissons ekvation på en ellips.

Starta comsol.

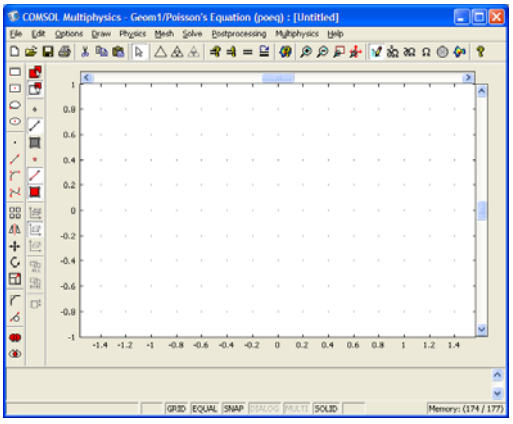
I den första skärmbilden **`ModelNavigator`** väljer vi 2D och `ApplicationModes>ComsolMultiphysics>PDEModes>ClassicalPDEs>Poisson's Equation`.

Här syns också att elementtypen är **`Lagrange P2`**-trianglar, dvs., lösningen approximeras med styckevis andragradspolynom, ett över varje triangel. Det går att ändra på sedan.

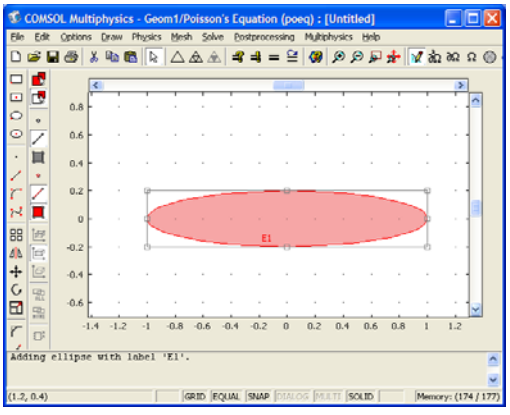


Nästa skärmbild visar arbetsytan och man arbetar oftast från vänster till höger i menyraden:

1. Skapa geometri (**`Draw`**)
2. Definiera PDEn (**`Physics`**)
3. Sätt randvillkor (**`Physics`**)
4. Gör elementnät (**`Mesh`**)
5. Lös (**`Solve`**)
6. Plotta (**`Postprocessing`**)



Under **`File`** finns operationer som **`open, save, save as`**, etc. **`Options`** ger faciliteter som att definiera konstanter och uttryck. Praktiskt att ha namn på värdena på modellparametrar som värmeledningstal, specifikt värme, etc.



Geometri

Man bygger geometrin i ett rit-program från elementar-kroppar, i 2D rektanglar och ellipser, eller ytor som begränsas av Bezier-kurvor. Delkropparna sätts ihop med mängdoperationer. Vi tar en ellips, centrum i origo och halvaxlar 1 och 0.2. Observera att objekten snäpps mot rutnätet, ändra

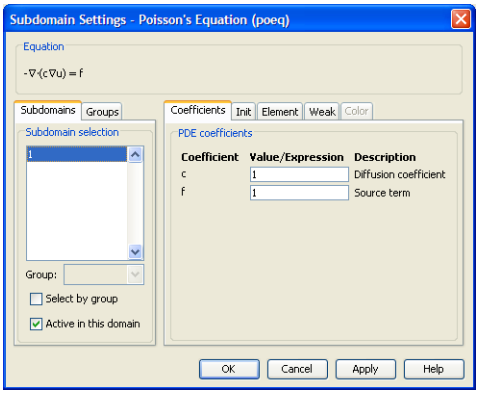
rutnätet under **`Options>GridSettings`** om det behövs. Om objektet som här består av en enda delkropp kan vi direkt gå vidare.

Definiera PDE

`Physics>SubdomainSettings` (det finns boundary också, strax…)
Här står PDEn:

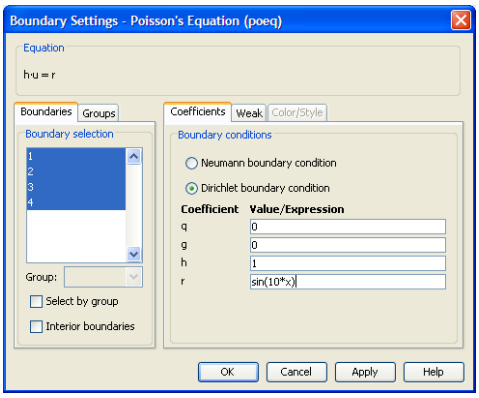
$$-\text{div}\,(c\,\text{grad}u) = f$$

och vi ska definiera koefficienten c och högerledet f. Man kan ge funktioner av x,y, u, ux, uy och (om tidsberoende) t. Som synes finns bara en subdomän, ellipsen. Välj den och acceptera $c = f = 1$ genom OK.



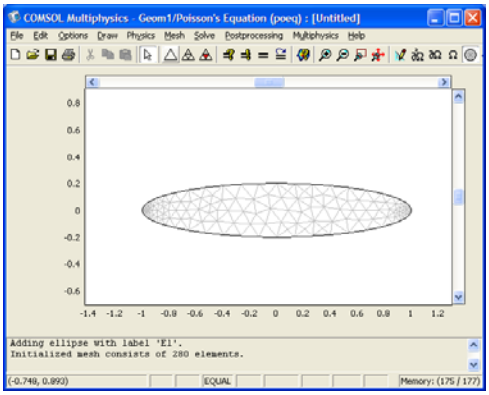
Randvillkor

`Physics>BoundarySettings`
Varje ellips ger fyra ränder. Vi väljer alla med ctrl-a; Det går att peka på dem också på arbetsytan. Default-randvillkoret är uppenbarligen Dirichlet: $h\,u = r$ med $h = 1$ och $r = 0$ och vi byter till $r = \sin(10x)$ för att få lite variation, OK.



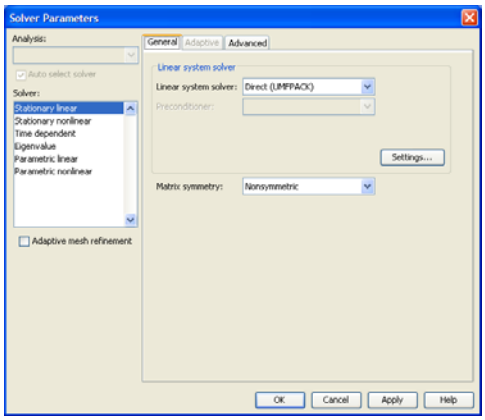
Elementnät

Ikonen triangel ger ett nät. Det ser bra ut, OK. Man kan noga styra element-genereringen med parametrar som sätts under **`Mesh>MeshParameters`** .



Lösning

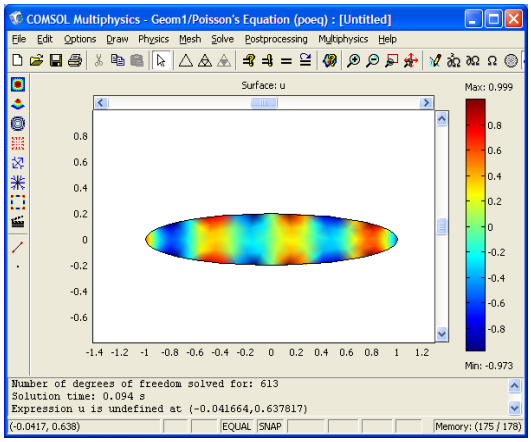
Man kan lösa de genererade ekvationerna med ikonen =. Vi går först in under **`Solve>SolverParameters`** och ser vilka möjligheter som bjuds. Med vårt val av c, f och randvillkor är problemet linjärt, så förvalet **`stationary linear`** är bra, OK. Detaljer som kan vara nog så viktiga för stora problem är t ex val av numerisk lösare för de linjära ekvationssystemen kan styras här. Visserligen står det **`Unsymmetric`** om



matrisen (och den är symmetrisk) men vi accepterar det, OK, och löser med =.

Efter 0.1 sekunder har systemet med 613 obekanta ställts upp och lösts och lösningen målats som färgade iso-ytor på arbetsytan, och log-raden säger

```
Number of degrees of freedom
solved for: 613
Solution time: 0.093 s
```



Konvergensstudie

Vi vill se hur lösningen konvergerar och beräknar *u* i centrum . Välj
`Postprocessing>DataDisplay>Subdomain`
där vi ger koordinaterna för centrum och ser

```
Value: 0.020201, Expression: u, Position: (0,0)
```

i log-raden.

Förfina nätet reguljärt med fyrtriangelikonen, lös igen, skriv ut u(centrum) igen:

```
Number of degrees of freedom solved for: 2345
Solution time: 0.188 s
Value: 0.019069, Expression: u, Position: (0,0)
```

och igen, och igen, ... Prova upp till ca en miljon frihetsgrader, sedan tar minnet slut.

Konvergens:

u(0,0)	**diff.**
1201	
69	-1132
231	162
234	3
231	-3

Konvergensordningen ska för reguljära lösningar bli 2 i energi-norm och 3 i max-norm. Här tycks det i alla fall gå fort, tills avrundningsfelen stör bilden. I allmänhet kan vi inte vänta oss samma mycket regelbundna konvergens som man ser med t ex trapetsregeln för kvadratur på snälla funktioner, eftersom elementnätet är så oregelbundet. Inte heller blir det alltid *precis* samma nät vid två likadana körningar! I nätgenereringsprocessen ingår val som styrs av (pseudo-)slumptal.

Prova nu med linjära ansatsfunktioner istället. **`Lagrange P1`** element väljs under
`Physics>SubdomainSettings>Element`

Experimentet upprepas: Långsammare konvergens, ordning runt 2.

#dof	tid	u(0.0)	diff.
613	0.094	0.019103	
2345	0.156	0.019115	12
9169	0.547	0.019205	90
36257	2.172	0.019224	19
144193	11.204	0.019230	6

Exempel 2 – Poissons ekvation på en fyrklöver
Starta ny modell genom att välja **`File>New,`** välj PDE som ovan. Vi vill lösa samma PDE men på en fyrklöver:

Geometri
Skapa en cirkel. Kopiera den med ctrl-c (som windows), klistra in med ctrl-v, ge translationen i dialogrutan; klistra in två gånger till med lämpliga translationer. Skapa unionen av de fyra cirklarna i **`Draw>CreateCompositeObject,`** t.ex ”**`select all`**” (union är default-operation, som man ser i formel-rutan), OK. Objektet döptes till **`CO1`** (Composite Object 1).

Definiera PDE
`Physics>SubdomainSettings`
Som synes finns bara en subdomän, unionen; vi hade kunnat ha kvar alla inre begränsningskurvor som blir av de fyra cirklarna, men vi hade omedvetet (default!) valt att ta bort inre ränder, så nu är de borta. Välj den enda domänen och acceptera $c = f = 1$ genom OK.

Randvillkor
`Physics>BoundarySettings`
Det blev många ränder, 8 stycken; varje cirkel ger fyra och de två inre har tagits bort. Vi väljer alla med ctrl-a; default-randvillkoret är uppenbarligen Dirichlet: $r = \sin(10x)$ som ovan, OK.

Elementnät
Triangel-ikonen ger ett nät. Det ser bra ut, OK.

Lösning
Lös med =. Efter 0.2 sekunder har systemet med 1290 obekanta ställts upp och lösts och målats som färgade iso-ytor på arbetsytan.
Konvergensstudie som ovan (möjligen har du andra koordinater)

```
#dof     tid     u(0,0)    diff.
  5057  0.422   -0.253078
 20033  1.469   -0.253353  275
 79745  7.218   -0.253463  110
318209 39.953   -0.253507   44
```

Konvergensordningen någonstans mellan 1 och 2. Kan det vara så att lösningen inte är så reguljär som krävs? De spetsiga inåtgående hörnen är uppenbara misstänkta. Plotta |gradu| så syns det,
`PlotParameters>Surface>SurfaceData`, välj **`|gradu|`**, OK
och kanske ännu tydligare om man ritar 3D-graf**,**
`PlotParameters>Surface>HeightData,` välj **`|gradu|`**, klicka i boxen **`HeightData`**, OK.

Man kan visa med variabelseparation att singulariteten är av typ $r^{\frac{\alpha}{\pi}}$ då yttervinkeln är α (här $\pi/2$). Det räcker för att försämra konvergensordningen märkbart.