



Prófanir á TOUGH

Tómas Jóhannesson

Greinargerð TJ-90-02

PRÓFANIR Á TOUGH

1. INNGANGUR

Síðastliðið sumar lentu Helga Tulinius og Grímur Björnsson í ýmsum erfiðleikum við notkun hermiforrítsins TOUGH á HP9000 tölvum OS. Erfiðleikarnir þessir koma (stundum) fram þegar TOUGH er notað við stór þrívíð vandamál (fjöldi búta ("elements") um og yfir 500 eða fjöldi tenginga ("connections") um of yfir 1000). Erfiðleikarnir fólust í því að keyrslur ýmist hrundu eða þurftu óeðlilega langan reiknitíma.

Hér að neðan er lýst niðurstöðum prófana á TOUGH sem gerðar voru með það fyrir augum að komast fyrir orsök þessara erfiðleika. Niðurstöður prófananna voru sendar til Karsten Pruess höfundar TOUGH með tölvupósti og keyrði hann sömu dæmi á Cray tölvu sína í Bandaríkjunum. Hann lenti í hliðstæðum erfiðleikum og lagði til ákveðnar breytingar á inntaksskrám TOUGH sem gera kleyft að komast hjá vandræðum.

Prófanirnar leiddu í ljós athyglisverðar upplýsingar um keyrslutíma TOUGH fyrir ýmis vandamál á nokkrum gerðum tölva og koma þær fram hér að neðan. Einnig er gerð grein fyrir þeim möguleikum sem nú standa Orkustofnun til boða við keyrslu á mjög stórum vandamálum.

2. UPPSETNING OG BREYTINGAR Á TOUGH

TOUGH var sett upp á HP tölvum OS haustið 1989. Útgáfan sem sett var upp er ættuð frá NESC (National Energy Software Center) í Bandaríkjunum. Hún miðast við vandamál með færri en 500 bútum og 1000 tengingum. Þessi útgáfa var prófuð og reyndist reikna rétt fyrir ýmis vandamál sem NESC dreifir með útgáfu sinni af TOUGH. Einnig bar TOUGH sæmilega saman við

SHAFT79 þegar gamalt vandamál sem leyst hafði verið með SHAFT79 á Orkustofnun var keyrt með TOUGH (mismunurinn sem fram kom virðist stafa af villum í SHAFT79 en ekki TOUGH).

Upprunaleg útgáfa TOUGH reyndist of lítil til þess að ráða við stór þrívíð reikninet sem skilgreind voru fyrir Hvíthólasvæðið í Kröflu síðsumars 1990. Keyrslur byrjuðu að hrynja þegar fjöldi tenginga fór að nálgast 1000. Virðist því skilgreindur hámarksfjöldi búta og tenginga í TOUGH þurfa að vera nokkru hærri en fjöldi búta og tenginga í vandamálum sem keyrð eru.

Útbúin var ný útgáfa af TOUGH sem ræður við allt að 1000 bútum og 2000 tengingum. Fyrir þessa útgáfu var TOUGH breytt lítilliga þannig að fjöldi búta og tenginga o.fl. er lesinn úr svokölluðum "include"-skrá. Þetta gerir það að verkum að lítið verk verður að fjölga bútum eða tengingum í framtíðinni, gerist þess þörf.

Þegar þessi nýja útgáfa TOUGH var keyrð fyrir Hvíthóla tóku keyrslur mjög langan tíma en niðurstöður, sem mönnum entist þolinmæði til þess að bíða eftir, virtust réttar.

2.1 PRÓFANIR

Fyrst var athugað hvort TOUGH hefði eitthvað laskast við stækkun og breytingar frá upprunalegri útgáfu. Sýnidæmi NESC (6 talsins) voru keyrð með báðum útgáfum TOUGH og voru niðurstöður eðlilegar og bar vel saman milli útgáfanna tveggja. Einnig var reiknitími útgáfanna sambærilegur. HP tölvur OS reiknuðu 10-20 sinnum hægar en Cray X-MP/14 tölva NESC og er það eðlilegur munur. Það var því eitthvað sérstakt við Hvíthólareikninetið sem TOUGH átti í erfiðleikum með.

Til þess að skoða þetta nánar var sett upp einfalt dæmi sem reiknar Theislausnina fyrir flæði vatns að borholu (þ.e. einfasaflæði) í gropnu bergi. Dæmið var sett upp með 1, 2, 3, 4 og 5 nákvæmlega eins lögum. Þegar þessi dæmi voru keyrð kom í ljós að reiknitími fór upp úr öllu valdi þegar lögnum fjölgaði með mjög svipuðum hætti og í Hvíthóladæminu (sjá töflu hér að neðan). Reiknitími meira en 150 faldaðist við það að fjölga lögum úr 1 í 5. Sömu dæmi voru prófuð á VAX tölvu OS, en hún keyrir 3-6 sinnum hægar en HP tölvurnar. Reiknitíminn á VAX óx með sama hraða og á HP tölvunum eftir því sem lögum var fjölgað.

Niðurstöður í Theisdæmunum reyndust eins sama hversu mörg lög voru notuð og í ágætu samræmi við Theislausnina (sjá meðfylgjandi mynd, punktar eru niðurstöður TOUGH, samfelld lína er þekkt lausn, frávik punktanna frá línunni stafa af ónákvæmum gildum á seigju og fjaðurstuðli vatns í þekktu lausninni).

Lýsing á Theisreikningunum var send til Karsten Pruess höfundar TOUGH við LBL með tölvupósti (sjá meðfylgjandi ljósrít). Hann keyrði sömu dæmin á Cray tölvu sína og lenti í hliðstæðum erfiðleikum með reiknitíma (sjá meðfylgjandi ljósrít af svörum Pruess).

Pruess prófaði síðan að breyta ákveðinni stærð í inntaki TOUGH sem tilgreinir hvort nota á "skölun" við að leysa jöfnuhneppi ($MOP(17)=1$ í stað 0) og gengu reikningarnir þá vandræðalaust. Sjálfgefið er að nota skölun ($MOP(17)=0$) og gefur það alla jafna nákvæmstu niðurstöðu. Í sumum tilfellum getur skölun hins vegar leitt til vandræða í reikningum og þarf þá að keyra án skölunar ($MOP(17)=1$). Fram kemur í svari Pruess að ekki sé hægt með einföldum hætti að segja fyrir um það hvenær rétt sé að nota skölun og hvenær ekki, heldur verði notendur að prófa sig áfram.

Þegar þessi breyting Pruess var prófuð keyrðu bæði Theisdæmin og einnig dæmi frá Hvíthólum eðlilega á HP tölvunum.

Öll ofangreind dæmi voru jafnframt keyrð á nýrri IBM RS6320 vinnustöð á Reiknistofnun Háskólans með TOUGH útgáfu sem sett var upp þar til þessara prófana. Niðurstöður voru í góðu samræmi við niðurstöður á HP tölvum OS en reikningar gengu 5-10 sinnum hraðar fyrir sig! Yfirleitt voru allir 4 útprentaðir stafir í niðurstöðum eins. Stundum munaði 1-2 í síðasta aukastaf en munur gat orðið meiri fyrir mjög smáar stærðir sem eru því sem næst 0. Í sýnidæmi 6 frá TOUGH kom einu sinni fyrir að munur sást á því hvernig lengd tímabreps þróaðist með tíma. Það stafaði af öllum líkindum af því að stærð sem ræður því hvort tímabrep er lengt lenti nærri viðmiðunarmörkum. Að öðru leyti voru niðurstöður eðlilegar.

Það er að sjá af prófununum að TOUGH keyri rétt á HP tölvum OS, en nauðsynlegt sé að notendur prófi sig áfram með einföld dæmi og mismundandi gildi á inntaksbreytum áður en stærri keyrslur eru settar af stað til þess að forðast vandræði sem ella geta komið upp.

3. REIKNITÍMI

Eftirfarandi tafla sýnir reiknitíma TOUGH fyrir öll dæmi sem nefnd voru hér að ofan.

Dæmi	Fj. búta	Fj. teng.	Fj. t.pr.	Reiknitími (s)			
				HP 9800		IBM RS6320	
				M=0	M=1	M=0	M=1
Sam 1	22	3	4	12	10	2	2
Sam 2	81	80	20	500	404	62	51
Sam 3	164	289	14	224	160	29	21
Sam 4	50	49	50	141	101	18	12
Sam 5	174	305	75	4423	4341	506	482
Sam 6	91	90	100	2086	2004	236	224
Theis 1	100	99	10	41	34	7	5
Theis 2	200	298	10	151	76	24	12
Theis 3	300	497	10	950	129	225	18
Theis 4	400	696	10	2824	186	551	26
Theis 5	500	895	10	6164	241	1865	34
Htul 1	290	993	1	636	354	168	81
Htul 2	290	993	5	5944	1447	1262	314

Reiknitíminn var mældur á HP9800 tölvum OS og á IBM6320 tölvu RHÍ bæði með og án skölunar ($MOP(17)=0$ og 1). Í töflunni kemur jafnframt fram fjöldi búta, tenginga og tímabrepa fyrir hvert dæmi. "Sam 1-6" eru sýnidæmi NESC. "Theis 1-5" eru Theisdæmin með mismunandi mörgum lögum. "Htul 1-2" eru dæmi frá Hvíthólum annað

með 1 tímaþrepi en hitt með 5. M í töfluhausnum er skammstöfun á MOP(17).

Á Cray tölvu Pruess þurfti dæmi Theis 1 hins vegar 3.2 sekúndur en dæmi Theis 4 16.2 sekúndur.

Í öllum sýnidæmum frá NESC (nema e.t.v. dæmi 3) er reiknitíminn svipaður með og án skölunar. Þar er því eðlilegt að nota skölun.

Í Theisdæmunum með mörgum lögum sést hvernig reiknitíminn með skölun er miklu lengri en án skölunar.

Sjá má að reiknitíminn þegar allt er með felldu í raunverulegum dæmum (þ.e. að Theis dæmunum slepptum, þar eru reikningar óeðlilega einfaldir og fljótlegir) er ekki fjarri því að vera í réttu hlutfalli við margfeldið af fjölda tenginga og fjölda tímaþrepa. Þ.e. fyrir HP tölvurnar er reiknitíminn

$$cpu_{hp} \approx n_c n_t / (2-5).$$

Fyrir IBM vinnustöð RHÍ er hins vegar

$$cpu_{ibm} \approx n_c n_t / (10-20),$$

þar sem cpu_{hp} og cpu_{ibm} er reiknitími í sekúndum, n_c er fjöldi tenginga og n_t er fjöldi tímaþrepa. Þessar tölur er rétt að hafa í huga þegar TOUGH er keyrt. Ef reikningar taka margfalt lengri tíma en búast má við samkvæmt formúlunum hér að ofan þá er líklega eitthvað að.

4. RÁÐLEGGINGAR FRÁ PRUESS

Í svari Pruess kemur fram að hann sé vanur að prófa nokkra möguleika á gildum ýmissa breyta í inntaksskrá TOUGH áður en hann setur af stað stórar keyrslur. Í þessu prófunum er ekki þörf að keyra nema 1 tímaþrep og þurfa þær því ekki að taka langan tíma.

Auk MOP(17) sýnist mér rétt að huga að RE1 og e.t.v. RE2. Í keyrslu á Theisdæmunum kom fyrir að sjálfgefið gildi RE1=1.0E-5 reyndist of stórt og er ekki ólíklegt að oft þurfi að minnka það. Ef erfíðleikar koma upp má líta á U, WUP, WNR og DFAC. Rétt er að keyra þessar prófanir með fullri villuútprentun ($MOP(i)=1,$

$i=1,6$) til þess að hægt sé að sjá hvaða áhrif breytingar hafa. Í þessu sambandi er hægt að hafa hliðsjón af því hvað reikningarnir ættu taka langan tíma samkvæmt niðurstöðum um reiknitíma hér að ofan.

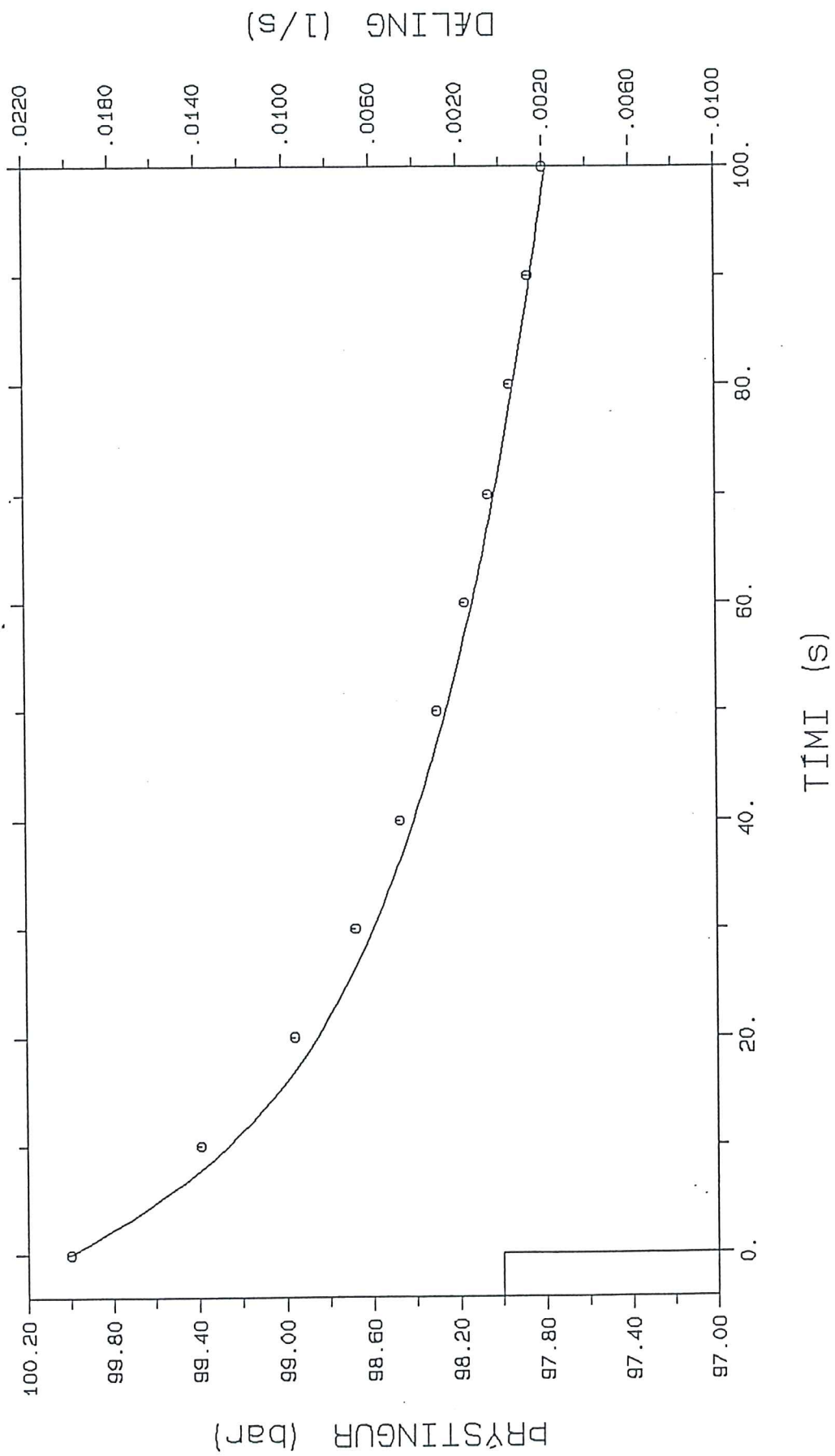
5. REIKNINGAR MEÐ TOUGH OS Í FRAMTÍÐINI

Reikningar með TOUGH á hinna nýju IBM vinnustöð RHÍ eru óverulega hægari en á Cray (u.þ.b. 50%). Þegar RHÍ fær enn afkastameiri IBM RS6000 tölvu síðar í haust þá verða reikniafköst við reikninga af þessu tagi á tölvum RHÍ fyllilega sambærileg við afköst á Cray tölvum þeim sem notaðar hafa verið við TOUGH reikninga í Bandaríkjunum. Í sambandi við prófanirnar sem hér er lýst var TOUGH sett upp á IBM vinnustöð RHÍ og er TOUGH því til reiðu þar ef á þarf að halda án frekari vinnu við uppsetningu. Tiltölulega auðvelt er að flytja dæmi sem sett hafa verið upp á tölvum OS yfir á tölvur RHÍ. Það er ekkert því til fyrirstöðu að reikna stærstu vandamál sem sett hafa verið upp fyrir TOUGH erlendis á nýju tölvur RHÍ (gegn greiðslu fyrir cpu- og disknotkun). IBM RS6000 tölvur eru tiltölulega ódýrar og er ekki ósennilegt slík tölva verði keypt til OS einhvern tíma í framtíðinni. Tölvuafköst við reikninga ættu því ekki að verða takmarkandi fyrir útreikninga með TOUGH á OS í framtíðinni.

HP tölvur OS geta ráðið við minni og meðalstór vandamál sem taka upp í 0.5 til 3 sólarhringa í keyrslu. Það svarar nokkur hundruð (jafnvel allt að 1000) tímaþrepa fyrir dæmi með um 1000 tengingum. Hvort sem HP eða IBM tölvur verða notaðar við stærstu keyrslur er ljóst að mestu tími notenda fer í uppsetningu vandamála og prófanir til þess skilgreina sem best þær spurningar sem svara á með stóru keyrslunum og tryggja að þær gangi vel. Þessi undirbúningsvinna krefst engra stórra keyrslna og ráða HP tölvurnar því vel við hana.

Pruess nefnir í svörum sínum að nokkrir TOUGH notendur hafi prófað með góðum árangri að skipta um undirforrit sem sjá um að leysa jöfnuhneppi í TOUGH (MA28) og

sett í staðinn nýttískulegri undirforrit og ætlar hann að senda OS upplýsingar um slíkar breytingar. Ljóst er að villan sem kom upp í Theis- og Hvíthóladæmunum stafar af galla á MA28 eða notkun hans í TOUGH og væri vafalaust til bóta að nota eitthvað annað í staðinn. Þegar reynsla af TOUGH og hermireikningum er orðin meiri á OS kemur til greina að prófa að nota undirforrit úr SPARSPAK (sem til er á OS) í þessum tilgangi. Það krefst hins vegar nokkurrar forritunar.



Reykjavik, 11.09.1990

Dear Dr. Pruess;

Thank you for your interest in looking at the problem with TOUGH that I described to you on the phone yesterday.

I installed TOUGH last year based on the code we got from NESC. In order to get the code to run on our HP9000 computers (32 bit machines with 64 bit representation of FORTRAN double precision variables) I had to change all definitions of REAL variables to DOUBLE PRECISION and make a few changes in the format of output. I also inserted a routine for measuring CPU time. Apart from this no (intentional!) code changes have been made in the version of TOUGH which is used in most of the examples below. This TOUGH version can handle up to 500 elements and 1000 connections. This is not enough for the largest problems (TOUGH aborts with an error message if the elements more than 500. TOUGH crashes in many cases, if the elements are close to 500 or if the connections are close to 1000. I have found the reason for the crashes to be that an index in a work array in ma28 goes out of bounds). I have installed another version of TOUGH for the largest problems. This version handles up to 1000 elements and 2000 connections. Both versions agree on every problem where they can both be run and they both use approximately the same amount of CPU time.

I have run all 6 sample problems from NESC successfully with both versions of TOUGH getting execution times which are between 10 and 20 times longer than the execution time reported by NESC on a Cray X/MP.

I have noticed that some problems seem to be very tough for our version of TOUGH. I am not sure whether this is because of an error in our implementation of TOUGH, some limitation in our computers or perhaps some problem with the ma28 software.

I have been testing TOUGH in order to shed some light on this problem lately and I have simplified the input as much as possible in order to isolate the cause. The output of the test below is supposed to be similar to the Theis solution which makes it possible to compare the numerical results to an analytical solution.

The model grid has 100 radial blocks starting at radius 0.25m with uniform radial thickness of 0.25m. The grid has 1, 2, 3, 4 og 5 layers with uniform vertical thicknesses adjusted such that the the total vertical thickness of the model is 1m. The initial pressure is 100 Bars and the initial temperature is 25 degrees C. There is pumping of 0.01 kg/s equally distributed among the innermost elements. The grid is generated with your grid generator rz2d. Output is generated at 10,20, ... 100 seconds.

The input file for rz2d for the 1 layer model was the following:

```
***** beginning
radii
1
0.25
equid
50 0.25
layer
1
1.0
***** end
```

for the 2,3,4 and 5 layer models the last three lines were:

```
***** beginning
layer
2
0.5 0.5
*****
layer
3
0.333333333d0 0.333333333d0 0.333333333d0
*****
layer
4
0.25 0.25 0.25 0.25
```



```
*****
layer
5
0.2 0.2 0.2 0.2 0.2
***** end
```

The input file for TOUGH for the 1 layer model was the following:

```
***** beginning
*THEI* Debugging of TOUGH
ROCKS
LAY 1      2650.0    0.05  1.0E-14  1.0E-14  1.0E-14    0.0    0.0
LAY 2      2650.0    0.05  1.0E-14  1.0E-14  1.0E-14    0.0    0.0
LAY 3      2650.0    0.05  1.0E-14  1.0E-14  1.0E-14    0.0    0.0
LAY 4      2650.0    0.05  1.0E-14  1.0E-14  1.0E-14    0.0    0.0
LAY 5      2650.0    0.05  1.0E-14  1.0E-14  1.0E-14    0.0    0.0
```

```
START
PARAM
0 0 10 0 01111110000000000000000000000000
0.0E+00 0.0E+00 10.0 AA 1
```

```
1.0E7 25.0 0.0
RPCAP
1 .5000E+00 .5000E+00 .1000E+01 .1000E+01
1 0 0 1
***** ELEME and COMME blocks from rz2d were inserted here *****
```

```
GENER
AA 1PRO 1 MASS -1.0E-2
```

INCON

```
ENDCY
***** end
```

for the 2,3,4 and 5 layer models the GENER data blocks were the following:

```
***** beginning
GENER
AA 1PRO 1 MASS -0.5E-2
BA 1PRO 2 MASS -0.5E-2
```

```
*****
GENER
AA 1PRO 1 MASS -3.3333E-3
BA 1PRO 2 MASS -3.3333E-3
CA 1PRO 3 MASS -3.3333E-3
```

```
*****
GENER
AA 1PRO 1 MASS -0.25E-2
BA 1PRO 2 MASS -0.25E-2
CA 1PRO 3 MASS -0.25E-2
DA 1PRO 4 MASS -0.25E-2
```

```
*****
GENER
AA 1PRO 1 MASS -0.2E-2
BA 1PRO 2 MASS -0.2E-2
CA 1PRO 3 MASS -0.2E-2
DA 1PRO 4 MASS -0.2E-2
EA 1PRO 5 MASS -0.2E-2
***** end
```

With the default value of 1.0e-5 of the convergence criterion RE1, I noticed that the changes are so slow that here is no change in the output after the first 5 time steps. Therefore I lowered RE1 to 1.0e-7 and 1.0e-9. With the lower valued of RE1 the agreement with the analytical This solution is good. The agreement is also good for the default value of RE1 for the first 5 timesteps.

The CPU times required for 15 model runs for the 5 different number of layers and 3 different values of RE1 are shown in the following table:

CPU time in seconds on HP9000:

# of layers	# of elem	# of conne	RE1= 1.0e-5	RE1= 1.0e-7	RE1= 1.0e-9
1	100	99	27	28	51

2	200	298	67	105	187
3	300	497	310	793	1265
4	400	696	750	2310	3038
5	500	895	1360	~10000	15000

The number of iterations for the 10 timesteps was 2,2,2,4,2,1,1,1,1,1 for the 5 layer model with RE1=1.0e-5 and 4,5,4,3,5,4,4,4,6,4 for the 5 layer model with RE1=1.0e-9. The time step was never reduced.

The output was identical for the different number of layers for each of the values of RE1, which means that there is no flow between the layers. However there is an incredible increase in the the CPU when the number of layers is increased, especially for the lower values of RE1 (the last run for RE1=1.0e-7 has not completed when I send this mail to you!).

I wonder whether this behavior may occur for the default value of RE1=1.0e-5 for other more complex problems where we have observed excessive CPU requirements.

It would be interesting to see if you see the same behavior on your computer and also to have some comments from you on this problem if you have some idea of a possible explanation.

I do not send any output. If you want some specific part of my output I can e-mail it to you.

Sincerely,

Tomas Johannesson
National Energy Authority
Grensasvegi 9
108 Reykjavik, Iceland

e-mail address: tj@os.is

From rhi.hi.is!Csa2.LBL.Gov!pruess Mon Sep 17 20:34 EDT 1990
Received: from rhi.hi.is with uucp; Mon, 17 Sep 90 20:34:34
Received: by os.is (smail2.5)
id AA06071; 17 Sep 90 20:34:34 EDT (Mon)
Received: from isgate.is
by rhi.hi.is (15.11/smail2.5/03-10-88); Tue, 18 Sep 90 00:21:01 gmt
Received: from [128.3.254.197] by isgate.is (5.61++/smail2.5/03-30-89); Mon, 17 Sep 90 17:05:34 GMT
Date: Mon, 17 Sep 90 10:10:35 PDT
From: pruess@Csa2.LBL.Gov (Karsten Pruess)
Message-Id: <900917100919.2581046f@Csa2.LBL.Gov>
Subject: TOUGH-Theis
To: tj@os.is
Status: RO

Dear Dr. Johannesson:

I received your email with the information on the Theis problem (you apparently mailed this twice). I understand what the problem is, but have no immediate idea of why it arises and what to do about it. I will set the problem up here and will get back to you after my test runs. - One question: You mention that your model grid has 100 radial blocks, starting at $r = 0.25$ m, with uniform increments of $\Delta r = 0.25$ m; however, your R22D input file shows 50 blocks of 0.5 m starting at $r = 0.5$ m. Perhaps you could clarify this point.

We just had a two-day workshop on TOUGH last Thursday and Friday. Several people reported major code speed-ups for 3-D problems from replacing MA28 with other solvers. Excessive computing time on a 2-D problem is something new to me. I am curious to "see what I will see" when running your multi-layer Theis problem.

Sincerely,

Karsten Pruess

From rhi.hi.is!Csa1.LBL.Gov!pruess Sun Sep 30 14:25 GMT 1990
Received: from rhi.hi.is with uucp; Sun, 30 Sep 90 14:25:05
Received: by os.is (smail2.5)
id AA28753; 30 Sep 90 14:25:05 GMT (Sun)
Received: from isgate.is
by rhi.hi.is (15.11/smail2.5/03-10-88); Sun, 30 Sep 90 14:21:05 gmt
Received: from sunic.sunet.se by isgate.is (5.61++/smail2.5/03-30-89); Sun, 30 Sep 90 14:17:59 GMT
Received: from csa1.lbl.gov by sunic.sunet.se (5.61+IDA/KTH/LTH/1.159)
id AASunic28786; Sun, 30 Sep 90 04:50:01 +0100
Date: Sat, 29 Sep 90 20:48:39 PDT
From: pruess@Csa1.LBL.Gov (Karsten Pruess).
Message-Id: <900929204757.26812a2d@Csa1.LBL.Gov>
Subject: TOUGH Theis
To: tj@os.is
Status: RO

Berkeley, September 29, 1990

Dear Dr. Johannesson:

Thank you for your two emails of September 22nd and 24th. I have in the meantime run the Theis problem you described on the Cray X-MP computer that I normally use for simulations. I used $RE1 = 1.e-9$ and ran the problem with either 1 or 4 layers. The 1 layer problem performs without problem, CPU time is 3.223 seconds on the Cray. For the 4 layer problem I find that TOUGH goes into a terrible agony. During iteration on the first time step it obtains enormously large increments for certain grid blocks, so that subsequently primary variables go out of bounds in EOS, and time step is repeatedly reduced. After burning up 3 minutes of CPU time it still had not completed two time steps, and I stopped execution.

It is my experience that the appearance of huge increments during the Newton-Raphson iteration process is usually due to severe loss of accuracy in the linear equation solution, because of near-cancellation of almost equal numbers. Although I did not "dig" into the details of the Jacobian matrix, some pathological numerical properties might be expected for the multi-layer problem with identical behavior in the layers.

Following up on the suspicion that the problem is in loss of accuracy in the linear equation solution, I made a small change in the input deck. I changed MOP(17) from 0 to 1, i.e., I selected the "no scaling" option for the Jacobian matrix. ("Scaling" computes a set of multipliers for each row and each column of the Jacobian matrix, in such a way that the range of non-zero numerical entries in the Jacobian is "compressed".) With the choice MOP(17)=1, the 4-layer problem ran without difficulty, taking a CPU time of 16.162 seconds for the 10 time steps.

I have never been able to figure out simple rules when scaling is preferable and when it isn't. All it does is change the numerical conditioning of the matrix. Prior to "production runs" for a certain problem, I usually run a few time steps both with and without scaling, and then select whatever works best.

Under separate cover I will send you a set of extended abstracts of the papers that were presented at the recent TOUGH workshop. Several of them deal with improvements in the linear equation solution in TOUGH, and you may wish to contact their authors directly for further information or possible code exchange. I am not aware of anyone who has tried SPARSPAK with TOUGH.

A final comment: The possible variety of two-phase nonisothermal flow problems with water and air is very large. It has been my experience that for any given problem, slight modifications in the choice of parameters and options may produce significant gains in simulation efficiency and/or accuracy.

I hope you will find these comments helpful.

Sincerely,

Karsten Pruess